

デンソーロボット

垂直多関節型

V* -D/-E シリーズ

入門編

Copyright © 2001 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・高密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。

本書が扱うロボットシリーズ／モデル

- ・垂直多関節型ロボット V*-D, Eシリーズ

お願い

ご使用前に、1ページ～9ページの「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。

取扱説明書の構成

取扱説明書の構成

本製品に関する取扱説明書は、以下のように構成されています。

本製品を初めて導入された場合は、すべての取扱説明書をお読みにになり、よく理解してから使用してください。

ロボット概要書	ロボットの仕様および構成について説明します。
設置・保守ガイド	ロボット構成機器の設置、仕様変更および保守点検について説明します。
入門編（本書）	デンソーロボットの概要から、ティーチングペンダントを使って操作する方法およびWINCAPS を使ってプログラムを作成する方法まで、具体的な設備事例を取り上げて説明しています。ロボットの基本的な使い方を習得したい場合にお使いください。
操作ガイド	ティーチングペンダント、オペレーティングパネルおよびミニペンダントによる、ロボットの基本操作と補助機能について説明します。
WINCAPS ガイド	ロボットおよびロボットコントローラにパソコンを接続して、プログラムの開発と管理を行なう、パソコン教示システムの使用方法について説明します。
プログラミングマニュアル （1）（ ）	プログラム言語であるPACについて、そしてPACによるプログラムの作成方法、コマンド仕様について説明します。
RC5 コントローラ インタフェース説明書	RC5コントローラの概要、外部機器とのインタフェース、汎用・専用入出力信号、および入出力回路について説明します。
エラーコード表	ロボットやWINCAPS でエラーが発生した際、ティーチングペンダント、オペレーティングパネルまたはパソコン画面に表示されるエラーコードの一覧です。その解説・処置方法もまとめてあります。
オプション機器説明書	ロボットのオプション機器の仕様や操作について説明します。

本書（入門編）の構成

本書の構成は、以下のようになっています。

安全にご使用いただくために

ロボットを安全にご使用いただくための注意事項をまとめてあります。ご使用前に、必ずお読みください。

第1部 ティーチングペンダントを使って操作する

ティーチングペンダントを使って手動モードでロボットを動かしたり、簡単なプログラムを作って確認する方法を説明しています。

第2部 WINCAPS II を使ってパソコンでプログラムを作成する

WINCAPS II をパソコン上で起動し、プログラムを作成、コンパイルし実行形式にします。作成したプログラムをロボットコントローラにアップロードします。

シミュレーション動作を第3部で行うために、ロボットコントローラをマシンロック状態にしておきます。

第3部 作成したプログラムを使ってパソコンで動作を確認する

パソコン上でシミュレータを使ってプログラム動作を確認します。

第4部 プログラムを使ってロボットを動かす

ロボットを実際に動かします。パレタイジングプログラムについてサンプルを見ながら学習します。ライブラリの活用方法についても説明します。

第5部 デンソーロボットの特徴的機能

力制限機能など、デンソーロボットの特徴的な機能について説明しています。

付録

用語の解説や、コントローラおよびティーチングペンダント各部の名称を説明しています。また機能メニューの一覧を掲載しています。

安全上のご注意

安全にご使用いただくために、以下の注意事項は必ずお守りください。

警告・注意表示は、デンソーロボットを安全に正しくお使いいただき、操作者や他の作業者を含む人への危害あるいは他の設備への物的損害を未然に防ぐために守らなければならない事項を示しています。

これらの表示レベルと意味は次のようになっています。内容をよく理解してから本文をお読みください。

 警告	この表示を無視して誤った取扱いをすると、死亡または重傷を負う可能性が想定される内容を示しています。
 注意	この表示を無視して誤った取扱いをすると、傷害を負う可能性が想定される内容および物的損害の発生が想定される内容を示しています。

用語と定義

最大可動範囲 (Maximum space): エンドエフェクタ、ワークピース、アタッチメントなどロボットを構成するすべての部位の移動範囲について、設計上考えられる最大空間を指します。(Quoted from the RIA* Committee Draft.)

可動制限範囲 (Restricted space): 機械的なストッパ等の移動範囲限定装置によりロボットの移動範囲が制限された空間を指します。その限定装置を有効にしたときロボット本体、エンドエフェクタ、およびワークピースが移動できる最大距離が、このロボットの可動制限範囲の境界を決めることとなります。(Quoted from the RIA Committee Draft.)

可動範囲 (Motion space): ソフトウェア的手段によって制限された、ロボットの可動空間を指します。ソフトウェア的手段が設定されたときロボット本体、エンドエフェクタ、およびワークピースが移動できる最大距離が、このロボットの可動範囲の境界を決めることとなります。(The "motion space" is Denso-proprietary terminology.)

動作範囲 (Operating space): ロボットをタスクプログラムによって実際に操作するとき、そのロボットの制限動作範囲をいいます。(Quoted from the RIA Committee Draft.)

タスクプログラム (Task program): ロボットに目的の移動あるいはそれに伴う機能を行わせるための命令の集合、つまり(アプリケーション)プログラムをいいます。(Quoted from the RIA Committee Draft.)

(*RIA: Robotic Industries Association)

1 産業用ロボットの 「特別教育」の受講

産業用ロボットのティーチング・点検・調整・修理等に従事する作業者は「労働安全衛生法第59条および関連省令等」に定める産業用ロボットの「特別教育」の受講が義務づけられていますので、必ずこの「特別教育」を受講してください。

2 設置上の注意

2.1 適切な設置環境の確保

■ 標準タイプ

標準タイプは、防爆・防塵・防滴等の仕様にはなっていないので、次のような場所に設置することはできません。

- (1) 可燃性ガス・引火性液体等の雰囲気
- (2) 金属加工の削りクズ等導電性物質が飛散している雰囲気
- (3) 酸・アルカリ等の腐食性ガスの雰囲気
- (4) 切削液・研削液等のミスト雰囲気
- (5) イオウ含有の切削液・研削液等のミスト雰囲気
- (6) 大型のインバータ、大出力の高周波発信器、大型のコンタクタ、溶接機などの電気ノイズ源の近傍

■ 防塵防滴タイプ

防塵防滴タイプは、JIS B8438、IP54相当の防塵・防滴構造になっています。(VM-D-W型、VS-E-W型ロボットの手首部はIP65相当)ただし、ロボットコントローラは、防塵・防滴構造ではありません。

ミスト雰囲気等の環境で使用する場合は、ロボットコントローラ保護ボックス（オプション設定）をご使用ください。

防塵防滴タイプは、防爆構造ではありませんので、次のような場所に設置することはできません。

- (1) 可燃性ガス・引火性液体等の雰囲気
- (2) 酸・アルカリ等の腐食性ガスの雰囲気
- (3) 大型のインバータ、大出力の高周波発信器、大型のコンタクタ、溶接機などの電気ノイズ源の近傍
- (4) 液体に没する場所
- (5) 研削加工等、小さい削りクズの発生する雰囲気
- (6) 弊社推奨切削油以外での雰囲気
弊社推奨切削油：ユシロンオイルNo. 4C（不水溶性）
- (7) イオウ含有の切削液・研削液等のミスト雰囲気

2.2 作業空間の確保

ロボット本体および周辺機器は、ティーチング・保守点検等の作業を安全に行なうための作業空間を、十分に確保して、設置してください。

2.3 制御装置はロボット可動制限範囲の外へ設置

ロボットコントローラ・オペレーティングパネルおよびティーチングペンダントの設置場所は、ロボットの可動制限範囲の外で、かつロボットの作業が見渡せる場所で操作できる場所に設置してください。

2.4 計器類の設置

圧力計・油圧計その他の計器は、作業者の見やすい場所に設置してください。

2.5 電気配線・油空圧配管の保護

電気配線・油空圧配管が、損傷を受けるおそれのある場合は、覆い等を設け保護してください。

2.6 D種接地の確保

ロボット用電源の電源アースはD種接地(接地抵抗100Ω以下)としてください。

2.7 非常停止スイッチの設置

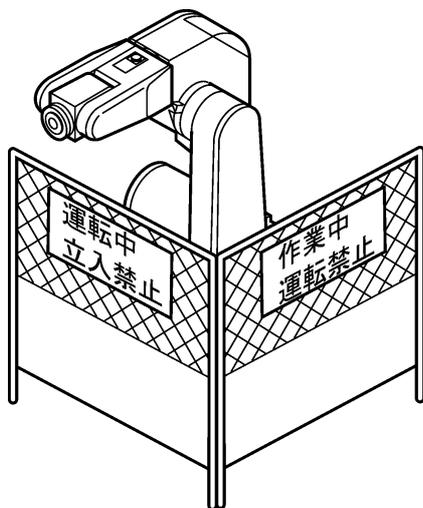
非常の際に、ただちにロボットの運転を停止できるよう、作業者が容易に操作できる位置に非常停止スイッチを設置してください。

- (1) 非常停止スイッチは、赤色にしてください。
- (2) 非常停止の機能は、作動したあと自動的に復帰せず、また他の作業者が不用意に復帰させることができないようにしてください。
- (3) 非常停止スイッチは、電源スイッチとは別個に設けてください。

2.8 運転状態表示灯の設置

ロボットが単に一時停止しているのか、非常・異常停止しているのかが、作業者に判るように、見やすい位置に表示灯を設置してください。

2.9 安全柵または囲いの設置



作業者および第三者が安易にロボットの可動制限範囲内に立ち入らないよう、必ず安全柵または囲いを設置するか、2.10項の措置を実施してください。安全柵または囲いは、以下の条件を守って設置してください。

- (1) 柵または囲いは、容易に移動できない構造にしてください。
- (2) 柵または囲いは、運転中に外力によって、容易に破損や変形しない構造にしてください。
- (3) 柵または囲いは、出入口を定め、これ以外の箇所から作業者および第三者が、乗り越えて侵入できないなど容易に入れない構造にしてください。
- (4) 柵または囲いは、手など身体の一部が入らない構造にしてください。
- (5) 柵または囲いの出入口には、次のいずれかの措置を講じてください。
 - ① 柵または囲いの出入口には、扉・ロープ・鎖等を設け、これらを開け、または外した場合に非常停止装置が自動的に作動するインターロック機構を設けてください。
 - ② 柵または囲いの出入口に「**運転中立入禁止**」および「**作業中運転禁止**」などの旨の表示を行ない、作業者にその趣旨の徹底を図ってください。

柵または囲いの設置前に試運転等でロボットを動作させる場合には、可動制限範囲内に作業者を立ち入らせないように、可動制限範囲外で、かつロボットの作動を見渡せる位置に監視人を配置し、監視業務に専念させてください。

2.10 ロープまたは鎖の設置

2.9項の措置が取れない場合、ロープまたは鎖を可動制限範囲の外側に張り、作業者および第三者が安易に可動制限範囲内に立ち入れないようにしてください。

- (1) 支柱は容易に動かないものにしてください。
- (2) ロープまたは鎖の存在が、周囲から容易に識別できるものにしてください。
- (3) 見やすい位置に「**運転中立入禁止**」および「**作業中運転禁止**」などの旨の表示を行ない、作業者にその趣旨の徹底を図ってください。
- (4) 出入口を定めて、出入口には2.9項の(5)に示す措置を講じてください。

2.11 ロボットの可動範囲の設定

ロボットがその作業を行なうのに必要な領域を動作範囲といいます。

ロボットの可動範囲が動作範囲より大きい場合、他の装置との衝突を防止するために、可動範囲を狭く設定することをお勧めします。

【参照】設置保守

2.12 ロボットの改造禁止

ロボット本体・ロボットコントローラおよびティーチングペンダント等の改造は絶対に行なわないでください。

2.13 作業工具の清掃等の措置

溶接ガン・塗装用ノズル等の作業工具を先端部に有するロボットで、作業工具の清掃等を行なう必要のあるものについては、当該作業が自動的に行なわれるようにすることが望まれます。

2.14 照度の確保

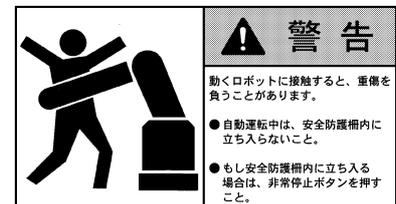
作業を安全に行なうために必要な照度を確保してください。

2.15 把持した物の飛来等の防止

ロボットが把持した物の飛来・落下等によって作業者に危険を及ぼすおそれがあるときは、物の大きさ・重量・温度・化学的性質等を勘案し、適切な防護措置を講じてください。

2.16 警告シールの貼り付け

ロボットの構成品として同梱されている「警告シール」を、安全柵の出入口等の見やすい位置に貼り付けてください。



3 作業上の注意



警告： 動作中のロボットに接触すると重傷を負う恐れがありますので、必ず以下のことを守り、3.1以降の注意に従って作業を行なってください。



警告

動くロボットに接触すると、重傷を負うことがあります。

●自動運転中は、安全防護柵内に立ち入らないこと。

●もし安全防護柵内に立ち入る場合は、非常停止ボタンを押すこと。

- ① ロボット運転中およびモータ電源が入っているときは、絶対にロボットの可動制限範囲に入らないでください。
- ② 異常処置等のため、ロボットの可動制限範囲に立ち入る場合は、非常停止装置を作動させる等により、ロボットのモータ電源を必ず切ってください。
- ③ ティーチングや保守点検等のためやむを得ずロボットの可動制限範囲内で、運転を伴う作業を行なう場合、必ず「3.3可動制限範囲内で作業を行なう作業者の安全確保」に示す措置を講じてください。

3.1 「作業規定」の作成と作業者への徹底

ティーチングや保守点検などのために、ロボットの可動制限範囲内で作業を行なう場合は以下の事項について「作業規定」を定め、作業者に徹底を図ってください。

- (1) 起動方法・スイッチの取扱方法等の作業において必要となるロボットの操作の手順
- (2) ティーチングなどの作業を行なう場合のロボットの速度
- (3) 複数の作業者に作業を行なわせる場合の合図の方法
- (4) 異常時に作業者がとるべき異常の内容に応じた措置
- (5) 非常停止装置等が作動しロボットの運転が停止したあと、これを再起動させるために必要な異常事態の解除の確認・安全の確認等の措置。
- (6) 上記以外に、ロボットの不意の作動による危険または、ロボットの誤操作による危険を防止するために必要な次に掲げる措置
 - ① 操作盤への表示（次ページの3.2項参照）
 - ② 可動制限範囲内で作業を行なう作業者の安全確保（次ページの3.3項参照）
 - ③ 作業位置・姿勢の徹底
ロボットの動きが常時確認でき、かつ異常時にすぐ退避できる位置および姿勢

- ④ ノイズ防止対策の実施
- ⑤ 関連機器の操作者との合図の方法
- ⑥ 異常の種類および判別方法

「作業規定」はロボットの種類・設置場所・作業内容に応じた適切なものとしてください。

「作業規定」の作成にあたっては、関係作業員・設備メーカーの技術者・労働安全コンサルタント等の意見を取り入れるように努めてください。

3.2 操作盤への表示

作業中は、当作業に従事している作業員以外の者が起動スイッチ・切り替えスイッチ等を不用意に操作することを防止するため、オペレーティングパネル・ティーチングペンダントおよび操作盤に、作業中である旨のわかりやすい表示をしてください。場合によっては、操作盤のカバーに施錠する等の措置を講じてください。

3.3 可動制限範囲内で作業を行なう作業員の安全確保

ロボットの可動制限範囲内で作業を行なうときは、異常時にただちにロボットの運転を停止することができるように、次のいずれかの措置を講じてください。

- (1) ロボットの可動制限範囲外でかつロボットの作動を見わたせる位置に監視人を配置し、監視業務に専念させて次の事項を行なわせてください。
 - ① 異常の際にただちに非常停止装置を作動させる。
 - ② 作業従事者以外の者をロボットの可動制限範囲内に立ち入らせない。
- (2) 非常停止スイッチ（ティーチングペンダントではロボット停止ボタン）をすぐ押せるように可動制限範囲内の作業員に携帯させてください。

3.4 ティーチング等の作業開始前の点検

ティーチング等の作業を開始する前に次の事項を点検し、異常を認めたときは、ただちに補修その他必要な措置を講じてください。

- (1) 外部電線の被覆または外装の損傷の有無
- (2) ロボットの作動の異状の有無（作動時に異常な音、振動がないか）
- (3) 非常停止装置の機能
- (4) 配管からの空気または油漏れの有無
- (5) ロボットの可動制限範囲内またはその付近の障害物の有無

3.5 残圧の開放

空気系統部分の分解・部品交換等の作業を行なうときは、あらかじめ駆動用シリンダ内の残圧を開放してください。

3.6 確認運転時の注意

確認運転を行なう場合は、作業者はできる限り可動制限範囲の外に出て、行なってください。

3.7 自動運転時の注意

(1) 起動時の措置

ロボットを起動させるときは、あらかじめ次の事項を確認するとともに一定の合図を定め、関係作業者に対し合図を行なってください。

- ① ロボットの可動制限範囲内に人がいないこと。
- ② ティーチングペンダント・工具等が所定の位置にあること。
- ③ ロボットまたは関連機器の異常を示すランプ等による異常表示がされていないこと。

(2) 自動運転時の確認ランプ等による自動運転中であることを示す表示がされていることを確認してください。

(3) 異常発生時の措置

ロボットまたは関連機器に異常が発生し応急処置のため可動制限範囲内に立ち入るときは、非常停止装置を作動させる等によりロボットの運転を停止させ、起動スイッチに作業中である旨の表示をする等、作業者以外の者がロボットを操作することを防止するための措置を講じてください。

3.8 修理時の注意

(1) 定められた範囲以外の修理は行なわないでください。

(2) いかなる場合においても、インターロック機構を取りはずさないでください。

(3) 電池の交換等のためにロボットコントローラの蓋を開くときは、必ずロボットコントローラのパワースイッチを切って、電源ケーブルを取りはずしてください。

(4) 補修用の部品は必ず当社指定のものをご使用ください。

4 日常点検・定期点検の実施

- (1) 日常点検および定期的な点検は必ず実施し、作業の前にロボットおよび関連機器に異常が無いことを確認してください。異常を認めた場合はただちに補修その他必要な措置を講じてください。
- (2) 定期的な点検または補修等を行なったときは、その内容を記録し、3年以上保存してください。

5 フロッピーディスクの管理

- (1) ロボットの構成品として、同梱されている「初期設定フロッピーディスク」は、大切に保管してください。そのロボット固有のデータが記録されています。
- (2) ティーチング終了時および変更後には、プログラム等のデータは必ずフロッピーディスクにセーブする習慣をつけてください。ロボットコントローラ内のデータが、バックアップ電池の寿命等で消失した場合にも、復旧が容易にできます。
- (3) ロボットの作動プログラムが記憶されているフロッピーディスクには、その内容を表示してください。間違ったフロッピーディスクを選択しないよう、必要な措置を講じてください。
- (4) フロッピーディスクは、ほこり・湿度・磁力線等の影響をうけて、誤動作することのないように、管理してください。

目次

はじめに	i
取扱説明書の構成	ii
取扱説明書の構成	ii
本書（入門編）の構成 本書の構成は、以下のようになっています。	iii
安全上のご注意	1
第1部 ティーチングペンダントを使って操作する	1
レッスン1 手動モードでロボットを動かす	3
1.1 ティーチングペンダント操作の基本	3
1.2 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す	5
レッスン2 簡単なプログラムでロボットを動かす	14
2.1 ティーチングペンダントから簡単なプログラムを作る	15
2.2 ティーチング	24
2.3 ティーチチェック	29
2.4 ロボットを自動運転で動かす	34
第2部 WINCAPSIIを使ってパソコンでプログラムを作成する	39
レッスン3 ティーチングペンダントを使ってロボットコントローラの準備を行う	40
3.1 キャリブレーション（CAL）を実行する	40
3.2 マシンロック状態にする	43
3.3 通信ポートを設定する	44
レッスン4 WINCAPSIIを起動し、システムプロジェクトを作る	48
4.1 システムマネージャを起動する	48
4.2 システムプロジェクトを新規登録する	50
4.3 パソコンの通信ポートを設定する	51
レッスン5 マクロを定義する	53

レッスン 6 プログラムを入力・編集する	55
6.1 プログラムサンプル	55
6.2 プログラム編集ウィンドウを開く	56
6.3 プログラムを入力する.....	57
6.4 コマンドビルダを使用する	58
6.5 プログラムを保存する.....	60
レッスン 7 プログラムをコンパイルし実行形式にする	61
レッスン 8 プログラムをアップロードする (パソコン→ロボットコントローラ).....	62
第 3 部 作成したプログラムを使って パソコンで動作を確認する	65
レッスン 9 パソコン側の用意をする	67
9.1 アームマネージャを起動する.....	67
9.2 アームマネージャの通信を接続する	67
レッスン 10 位置変数を入力する.....	68
10.1 ロボットの動きを手動でシミュレートする	68
10.2 現在位置を取り込む	70
10.3 位置変数を編集する	76
レッスン 11 プログラムを試運転する.....	77
11.1 プログラムをロードする	77
11.2 プログラムを起動する.....	79
レッスン 12 I/O の観測と操作を行う	80
12.1 DIO マネージャの起動し、通信を接続する	80
12.2 I/O を観測する	81
12.3 I/O の擬似 SW を ON/OFF する.....	82
レッスン 13 変数の観測と操作を行う	83
13.1 変数マネージャを起動し、通信を接続する	83
13.2 変数を観測する	84
レッスン 14 連続起動試験を行う	85
14.1 連続起動する	85
14.2 I/O を連続観測する	86
14.3 プログラムを停止させる	88

第4部	プログラムを使ってロボットを動かす	89
レッスン 15	ロボットを実際に動かす	90
15.1	マシンロックを解除する	90
15.2	速度・加速度を設定する	91
15.3	モータの電源を入れる	92
15.4	ティーチング	93
15.5	プログラムを起動する	96
15.6	速度を変更する	97
15.7	ロボットの連続運転を停止する	98
レッスン 16	パレタイジング	102
16.1	パレタイジングの説明	102
16.2	簡易パレタイジング	110
レッスン 17	ライブラリを活用する	118
17.1	プログラムバンク	119
17.2	プログラムのインポート	121
レッスン 18	作業を終了する	123
18.1	WINCAPS II を終了し、パソコンの電源を切る	123
18.2	ロボットコントローラの電源を切る	123
第5部	デンソーロボットの特徴的機能	125
レッスン 19	力制限機能	126
19.1	各軸電流制限機能	126
19.2	先端力制限機能	127
レッスン 20	その他の機能	128
付録		129
付録 1	用語	130
付録 2	ロボットコントローラ各部の名称	141
付録 3	ティーチングペンダント各部の名称	142
付録 4	ティーチングペンダントの機能メニュー	144

第1部 ティーチングペンダントを使って操作する

第1部で学習すること

ティーチングペンダントの取扱い方法および操作を学びます。

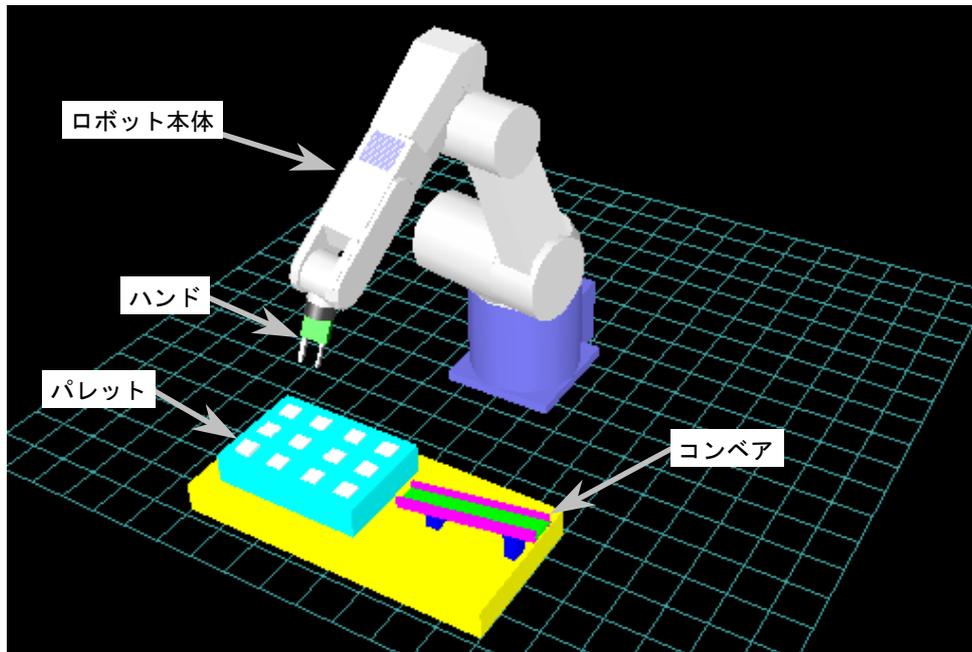
ティーチングペンダントを使用して、以下の作業を習得します。

- ・手動動作（各軸、X-Y、ツール）を安全かつ確実に行う
- ・キャリブレーション操作（CAL）を行う
- ・プログラムの作成、編集をする
- ・教示作業を安全に実施する
- ・ティーチチェックの方法を理解し、安全に操作する
- ・プログラムの起動と停止の方法を理解し、安全に操作する

▶ レッスン1 手動モードでロボットを動かす	3
1.1 ティーチングペンダント操作の基本	3
1.2 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す	5
▶ レッスン2 簡単なプログラムでロボットを動かす	14
2.1 ティーチングペンダントから簡単なプログラムを作る	15
2.2 ティーチング	24
2.3 ティーチチェック	29
2.4 ロボットを自動運転で動かす	34
■ 第1部の練習問題	38

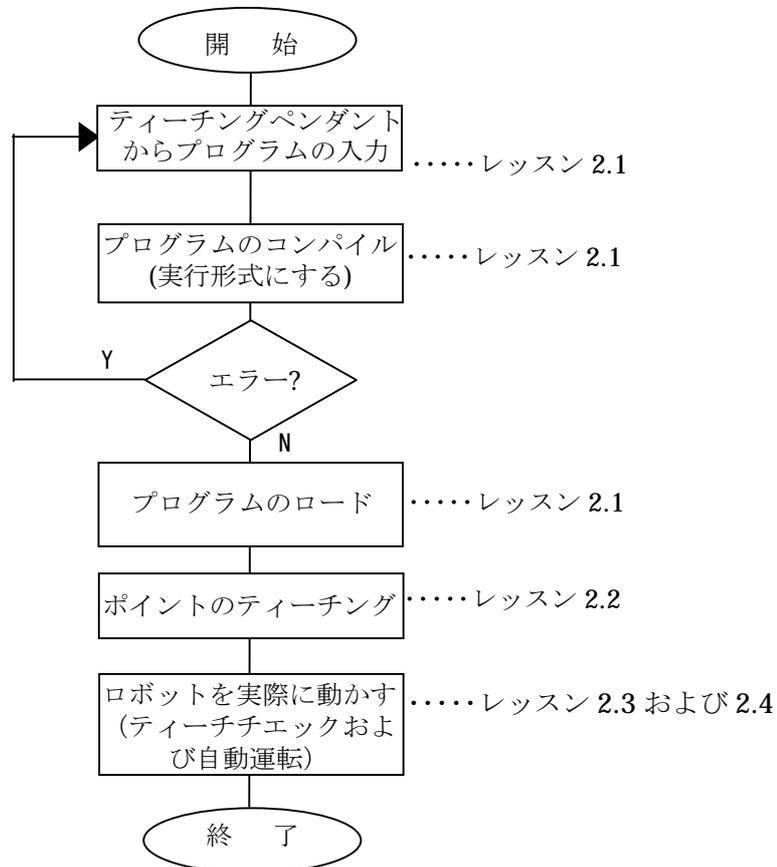
設備事例

下図はロボットを製造ラインに組み込んだ設備事例で、このロボットはパレタイジング作業を行います。



プログラム作成からロボット動作確認までの流れ

ティーチングペンダントを使用したプログラムの作成、ロボットの動作確認を行なうまでのフローを下に示します。



レッスン1 手動モードでロボットを動かす

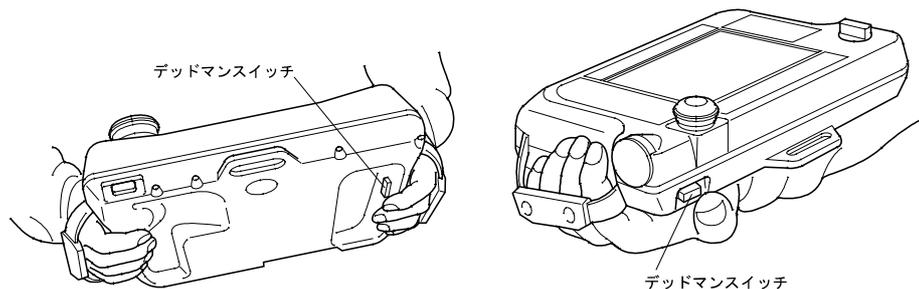
1.1 ティーチングペンダント操作の基本

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。

デッドマンスイッチは2箇所についていますので、下図に示すように2通りの持ち方が可能です。

確実に操作できるように、しっかりと持って操作してください。



ティーチングペンダントの持ち方

★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最中に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。

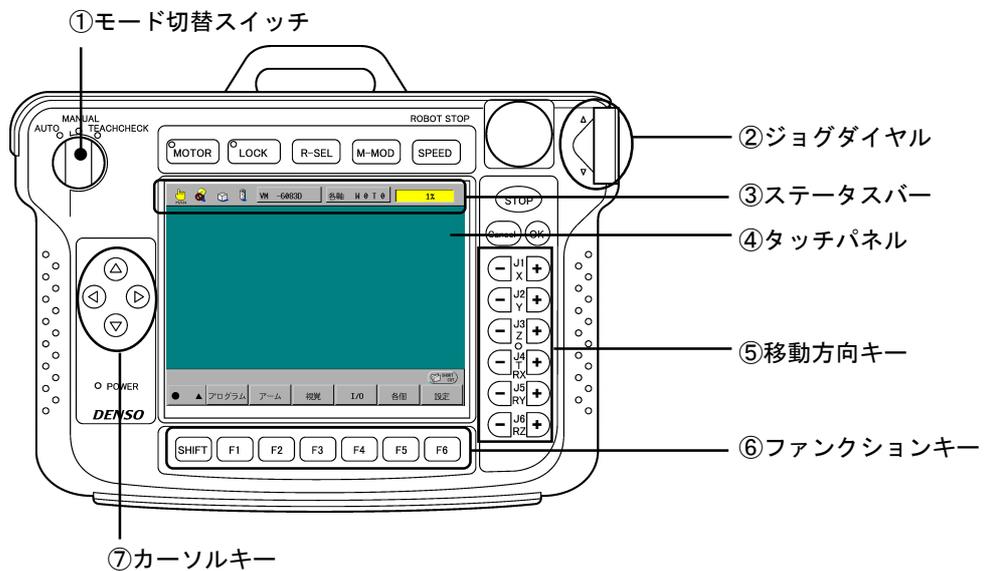
- 1) 押していないか、押す力が弱いとき → スイッチはOFF
- 2) 押す力が正常なとき → スイッチはON
- 3) 押す力が強すぎるとき → スイッチはOFF

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

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。

ティーチングペンダントの構成要素

コントローラの電源を入れると、ティーチングペンダントには下図のように基本画面が表示されます。



ティーチングペンダント基本画面

①モード切替スイッチ

動作モード（自動モード、手動モード、ティーチチェックモード）の切り替えを行います。

②ジョグダイヤル

数値調整を容易にします。

③ステータスバー

ロボットの動作モードや状態を常時表示します。

④タッチパネル

ティーチングペンダントの液晶表示画面は、タッチパネルにもなっています。画面に表示されるボタンや、データを入力する欄は、画面に直接タッチすることによって、操作したり選択したりできます。

注意：画面には、指でタッチしてください。ペン先など鋭利なものでつつくと故障の原因となります。

⑤移動方向キー

ロボットを手動で指定方向に動作させます。デッドマンスイッチを同時押下する必要があります。

⑥ファンクションキー

F1～F6の機能が画面に通常表示されます。必要に応じてSHIFTキーを押すと、F7～F12表示に切り替わります。

⑦カーソルキー

表示画面、入力画面でカーソルを移動させます。

ティーチングペンダント各部の詳細は付録3をご覧ください。

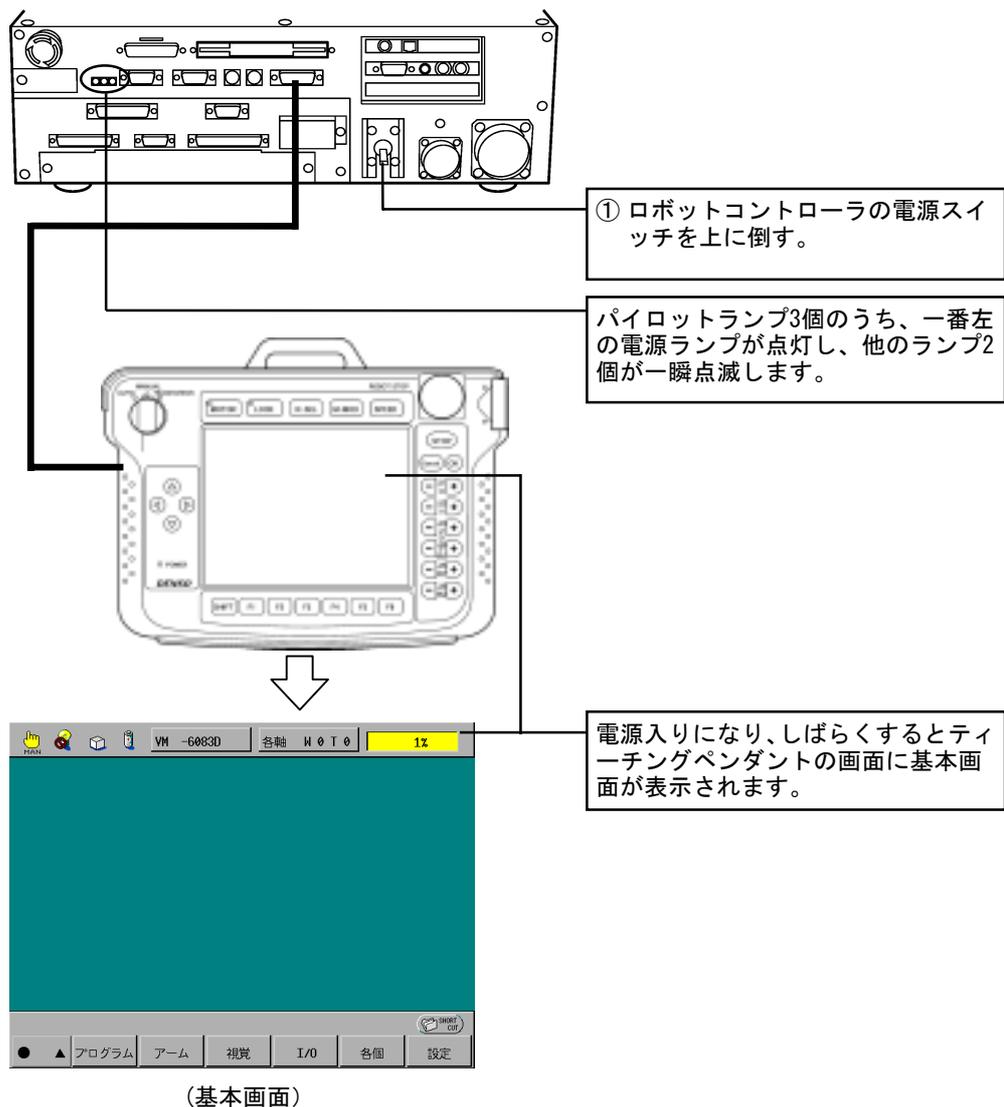
1.2 ティーチングペンダントを使ってロボットを手動で動かす

ここではまず、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てみましょう。

ステップ1 安全を確認する

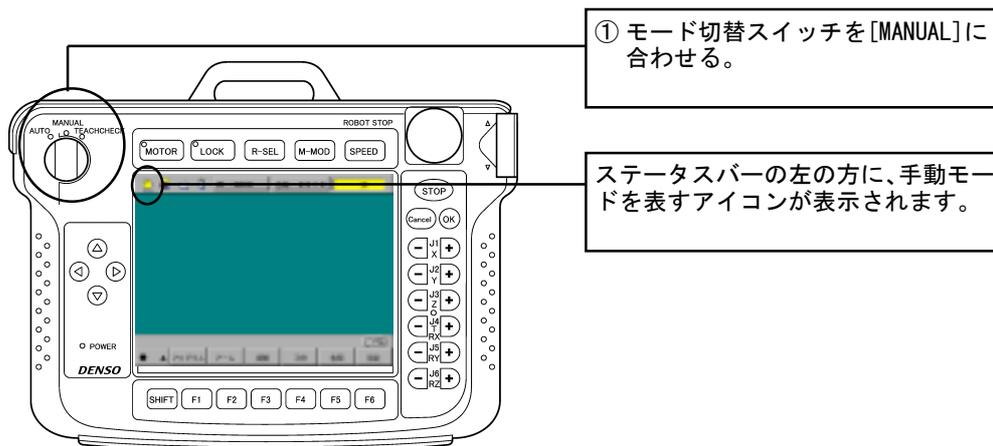
- ・ロボットの設置状況が正常であること。
- ・ロボットの可動範囲に人がいないこと。

ステップ2 ロボットコントローラの電源を入れる



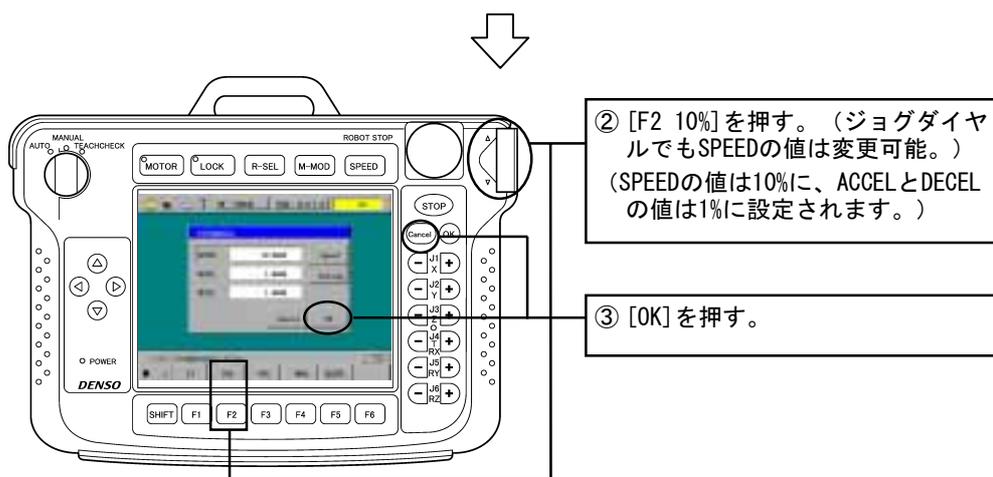
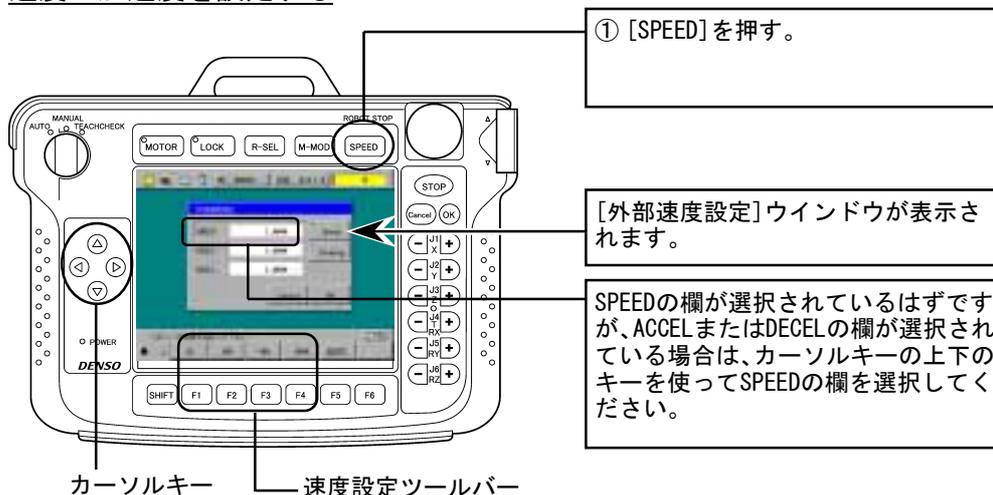
ステップ3

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★

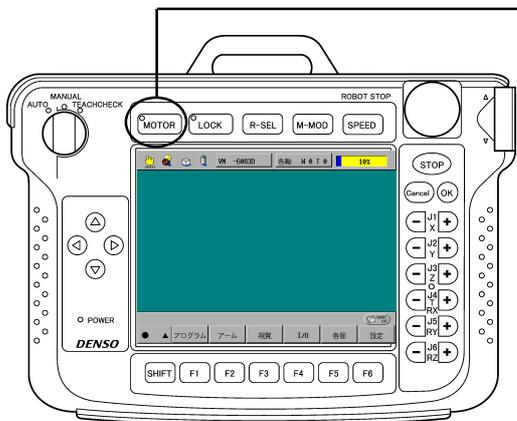
初めのうちは、安全のためにゆっくりと動かすので、この位の値にしておきます。あとで、様子がわかったところでこの値を入れ直すことができます。



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。

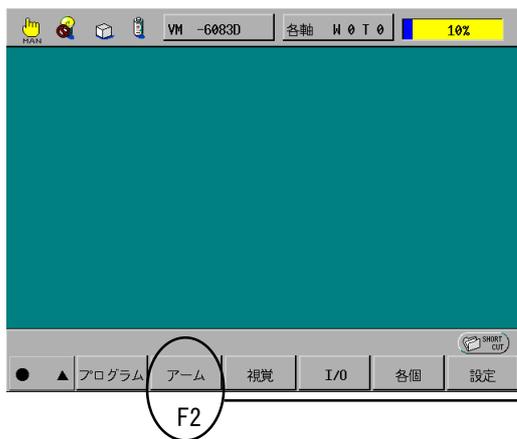
ステップ6

手動でロボットの各アームを実際に動作させる



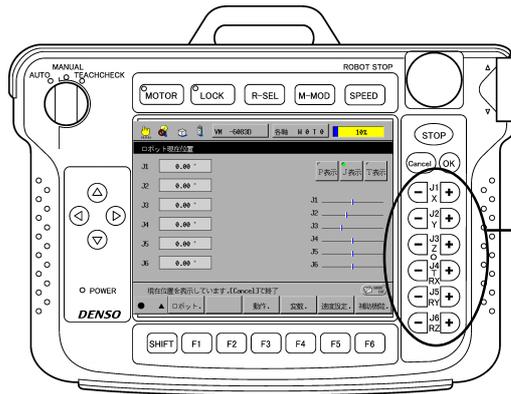
注意

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可動範囲から出てください。



① [F2 アーム] を押す。





② ロボットを見ながら、デッドマンスイッチと移動方向キーを押す。

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]ウインドウには、各軸の角度が表示されます。

注意：キャリブレーション(CAL)を行なうまでは、各軸モードでしかロボットを動作させられません。表示される各軸座標値も不正確です。

★注意★

実際に、正確な値でロボットを動作させるには、CALが必要です

ステップ7

CAL (キャリブレーション) を実行する

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。



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]ウインドウが表示されている状態で、[F6 補助機能]を押す。





② [F12 CAL実行]を押す。

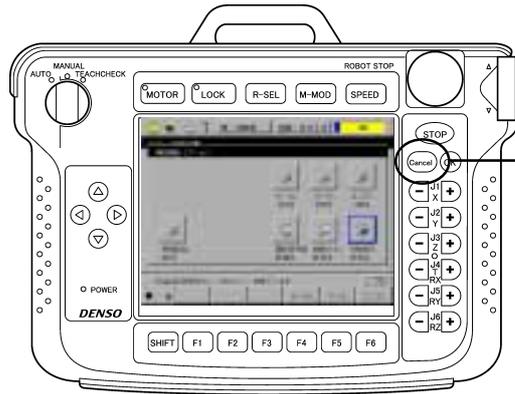


③ システムメッセージを確認し、[OK]を押す。



④ システムメッセージ「CAL成功しました!」を確認し、[OK]を押す。

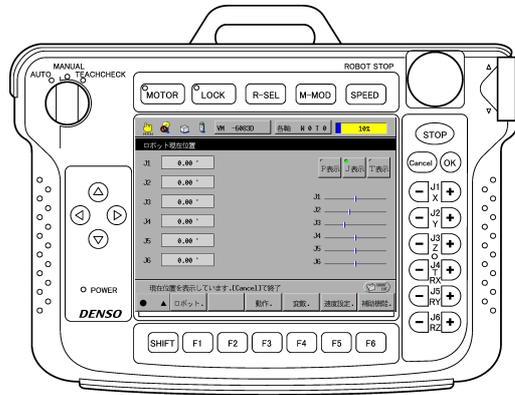




⑤ [補助機能(アーム)]ウインドウの表示に戻ったら、[Cancel]を押す。

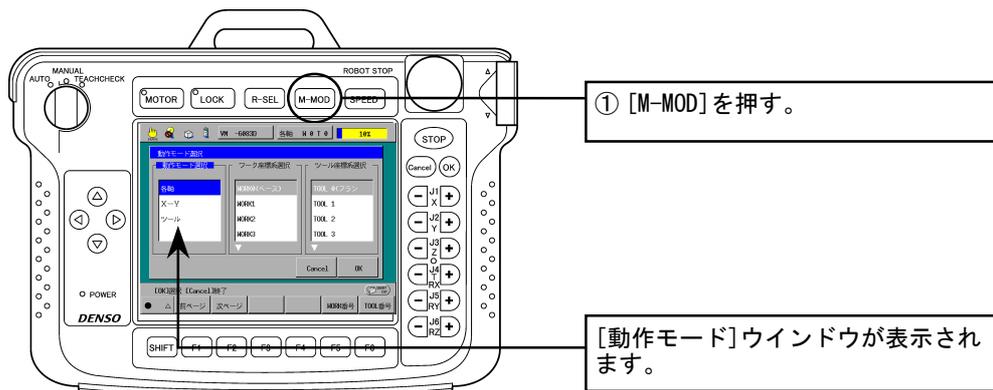


[ロボット現在位置]ウインドウの表示に戻ります。これで、CALは終了し、ロボット動作が可能となります。



ステップ 8

動作モードを選択し、手動運転する



★ポイント★

ロボットの手動動作では、各軸モード、X-Yモード、ツールモードの3動作モードが選べます。

〈各軸モード〉

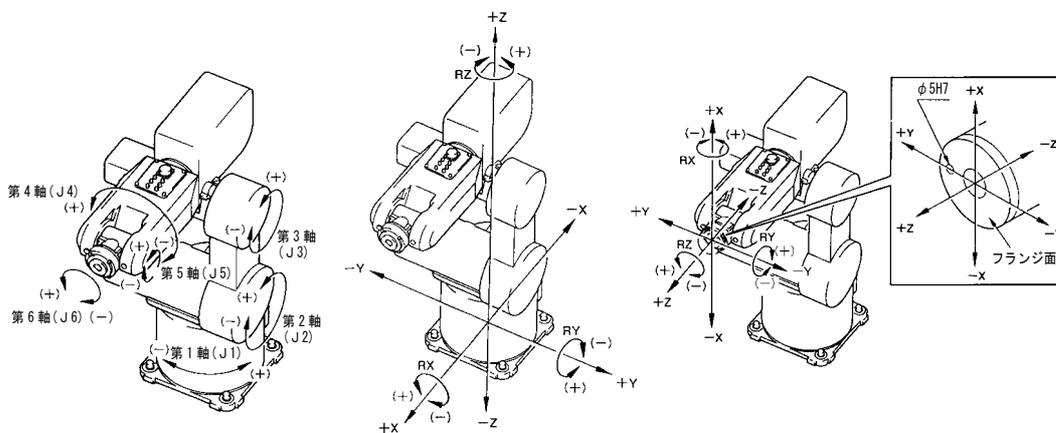
各アームごとの動作。

〈X-Yモード〉

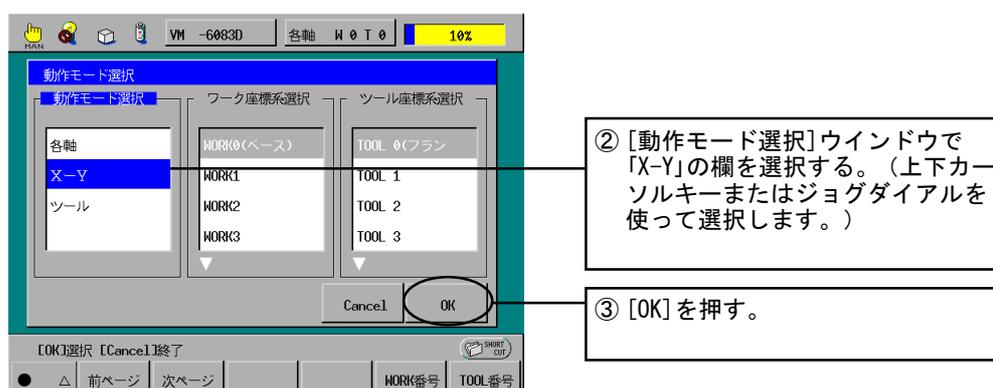
直交座標に沿って直線動作します。[RX][RY][RZ]の各キーを操作すると、フランジ面の中心の位置は変化せず、フランジ面中心に仮想的にもってきたワーク座標まわりの回転動作をします。

〈ツールモード〉

フランジ面上の直交座標に沿って直線動作します。[RX][RY][RZ]の各キーを操作すると、ツール座標の各軸まわりの回転動作をします。



ここではX-Yモードでロボットを運転してみましょう。



ステータスバーにX-Yと表示されます。

④ [F2アーム] を押す。



[ロボット現在位置]ウィンドウが表示されます。

⑤ 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。

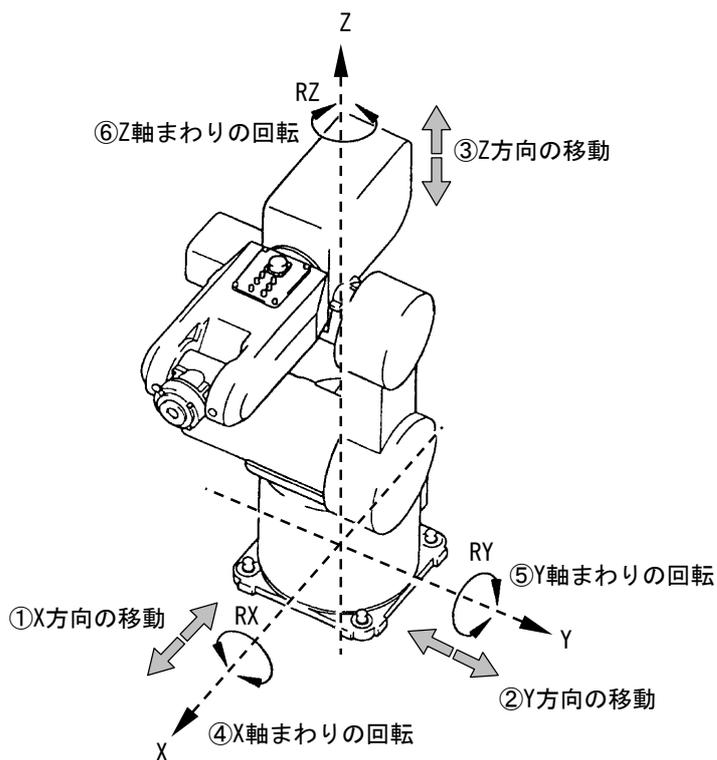
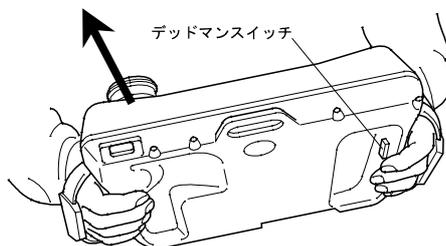


P表示が点灯し、現在位置の表示形式が[P表示]になります。



⑥ 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 RXの回転
 - ⑤ RYの回転
 - ⑥ RZの回転



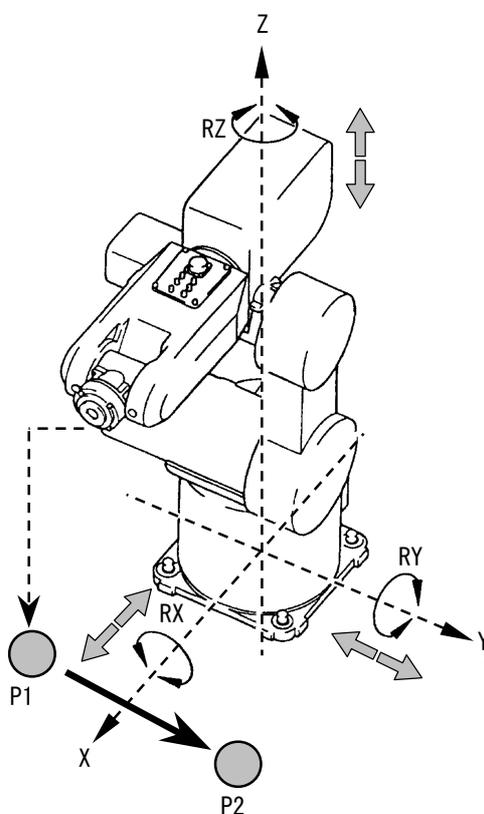
レッスン2 簡単なプログラムでロボットを動かす

ロボットに決まった動作をさせるには、プログラムを作成し、動作させたい位置をティーチングさせるといった作業が必要です。

ここでは、下図に示すようにロボットをP1からP2へ移動動作させてみましょう。

- 2.1 ティーチングペンダントから簡単なプログラムを作る
- 2.2 ティーチング (P1とP2のティーチング)
- 2.3 ティーチチェック
- 2.4 ロボットを自動運転で動かす

の順に説明します。



2.1 ティーチングペンダントから簡単なプログラムを作る

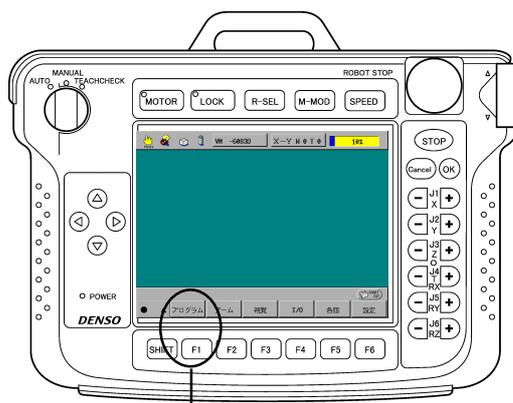
ティーチングペンダントを使って簡単なプログラムを実際に入力してみましょう。

プログラムの作成と編集は、手動モードで行ないます。前節から続けて操作する場合は、そのまま操作を続けてください。このレッスン2.1から操作を始める場合は、手動モードにしてから操作ステップに従って操作してください。

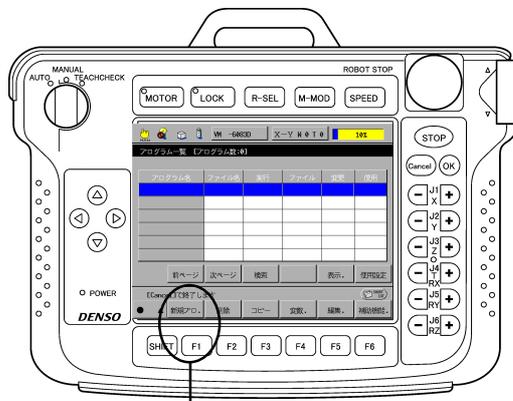
プログラムの作り方についての詳しい説明は、プログラミングマニュアルを参照してください。

ステップ1 新規プログラム編集画面を開く

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。

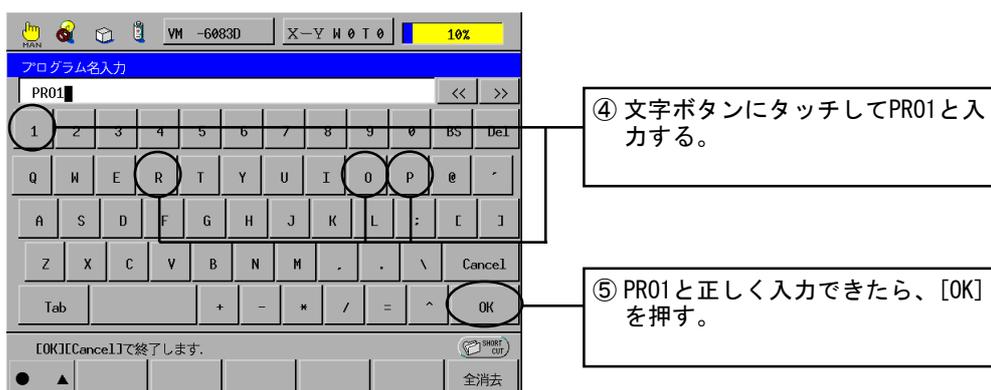


② [F1 新規プロ.]を押す。



③ [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。



ステップ 2

プログラムを入力する

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	‘ロボット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
GIVEARM	‘ロボット制御権を解放します
END	



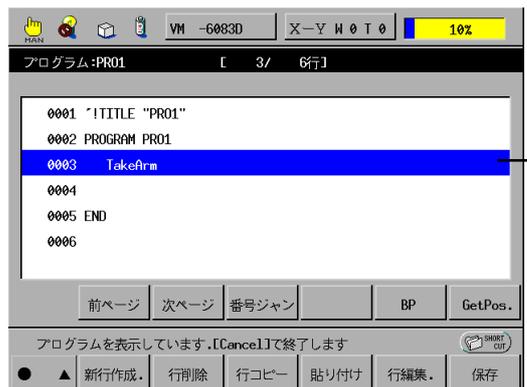
① カーソルキーまたはジョグダイヤルを使って、プログラム編集ウィンドウ [プログラム: PRO1] の3行目にカーソルを合わせる。

② [F5 行編集] を押す。



③ カーソルキーと[Del]を使って、行頭にある「]」(クォーテーションマーク)を取り除く。

④ [OK]を押す。



プログラム編集ウィンドウ [プログラム: PRO1] になり、修正された3行目が表示されます。





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:PRO1]になり、4行目に「SPEED 100」が表示されます。





⑧ 「SPEED 100」の入力と同様に、p. 16のプログラムコードをすべて入力する。

⑨ プログラムをすべて入力したら、[F6 保存] を押す。



⑩ [OK] を押し、入力したプログラムを保存する。

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。

ステップ3

プログラムをコンパイルし実行形式にする。

プログラムの編集ができたなら、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、プログラムをロードしたり、実行することができません。



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 [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。

④ [OK] を押す。





⑤ [OK]を押す。
コンパイルが始まります。



コンパイルが終了すると、画面は[プログラム一覧]ウインドウの表示に戻ります。

★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]ウインドウに戻ります。
- 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]ウインドウで、[F6 補助機能]を押すと、[補助機能(プログラム)]ウインドウが表示されます。
[補助機能(プログラム)]ウインドウに現れる[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。
この方法では、プログラムを実行形式に変換したあと、プログラムのロードも続いて行なえます。



F6

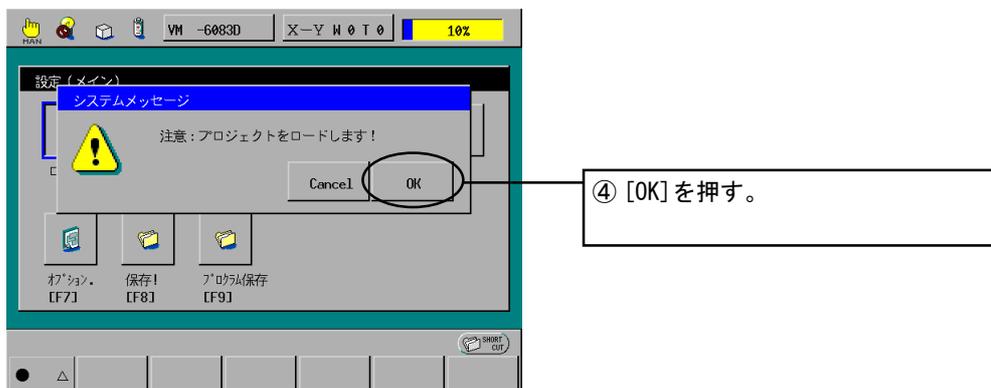
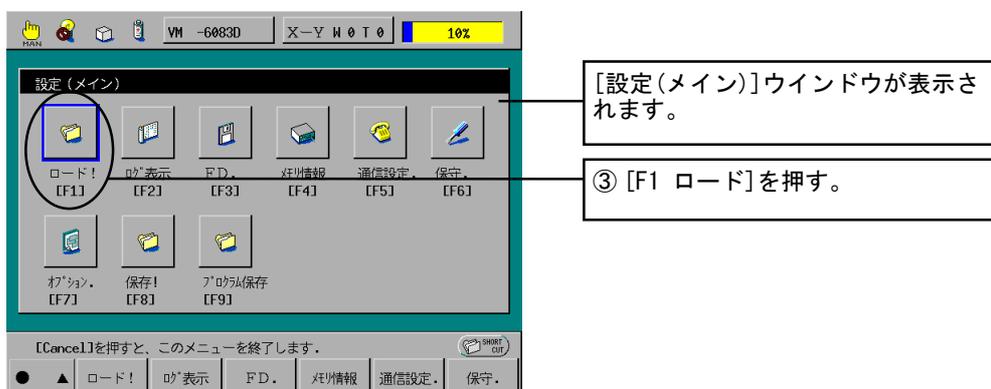
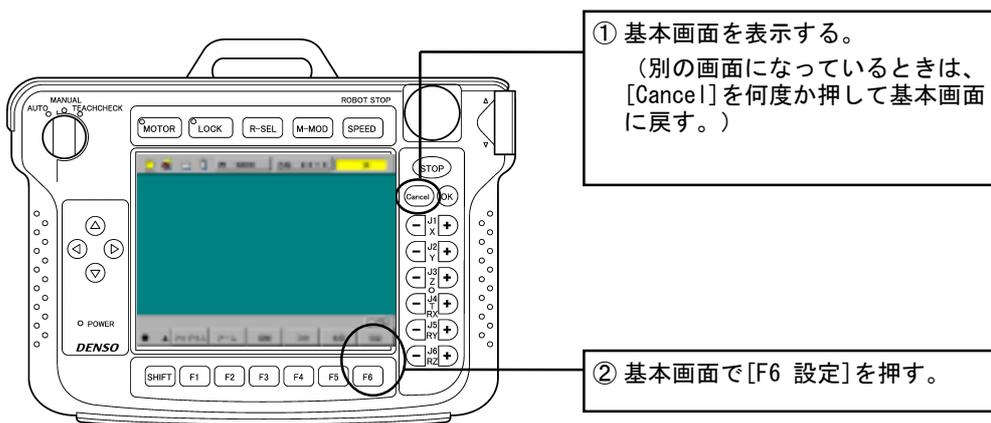
ステップ4

プログラムをロードする

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。





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。

2.2 ティーチング

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、ティーチングで位置変数の具体的な値を指定します。

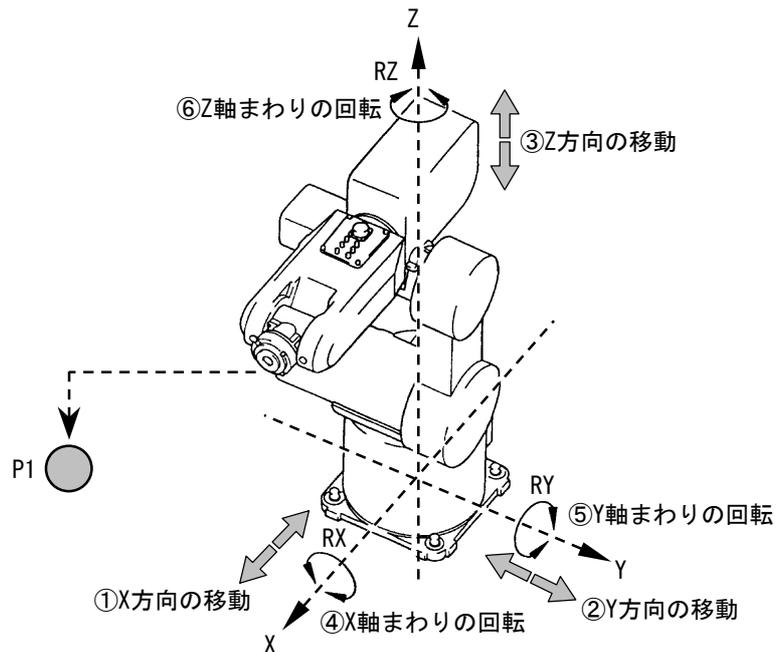
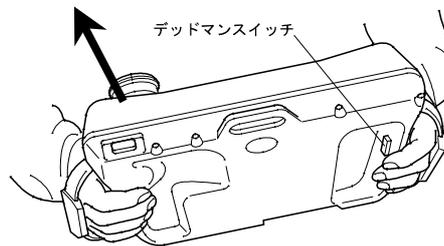
ここでは、レッスン2.1のステップ2で記述した位置変数P1とP2の値をティーチングしてみましょう。

ステップ1 ロボットの位置（P1）をティーチングする



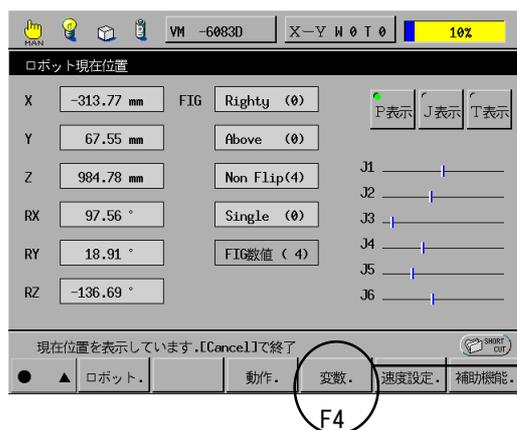
① デッドマンスイッチを押しながら、移動方向キーを適宜押して、P1に設定したい位置までロボットを動かす。

- 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 RXの回転
 - ⑤ RYの回転
 - ⑥ RZの回転



ステップ2

ティーチングした値を「変数名P1」に保存する



★ポイント★

「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。変数には以下のタイプがあります。

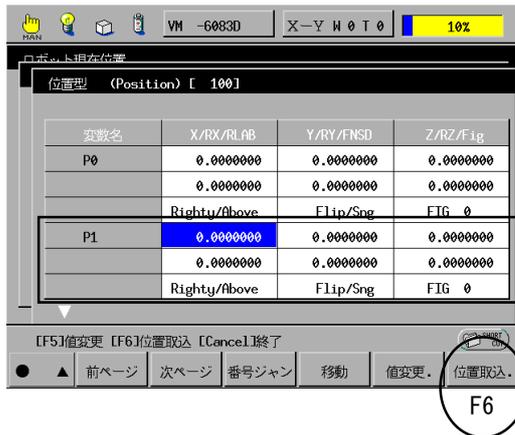
- I. 整数： 整数型変数（範囲：-2147483648～+2147483647）
- F. 実数： 単精度実数型変数（範囲：-3.402823E+38～+3.402823E+38）
- D. 実数： 倍精度実数型変数（範囲：-1.7976931348623157D+308～1.7976931348623157D+308）
- V. Vector： ベクトル型（X, Y, Z）
- P. 位置： 位置型変数（X, Y, Z, RX, RY, RZ, FIG）
- J. 軸値： ジョイント（軸）型変数（J1, J2, J3, J5, J6）
- T. 同次： 同次変換型変数（Px, Py, Pz, Ox, Oy, Oz, Ax, Ay, Az, FIG）
- S. 文字： 文字列型変数（最大で247文字までの文字列を含むことが可能）



[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。

[位置型] ウィンドウには、一つの変数について7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、7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。

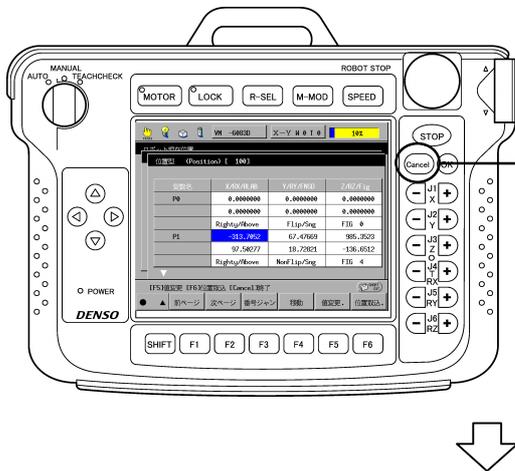




現在位置が変数P1の値として取り込まれています。

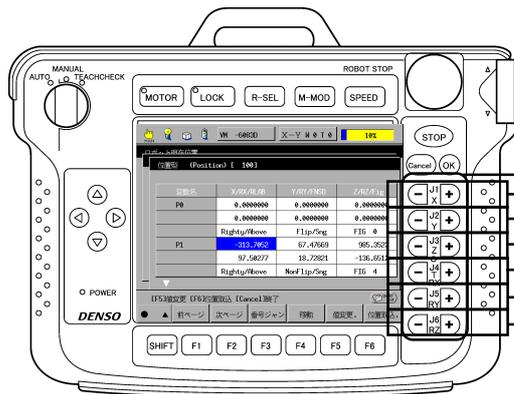
ステップ3

ロボットの位置 (P 2) のティーチングおよび [変数名P2] への保存



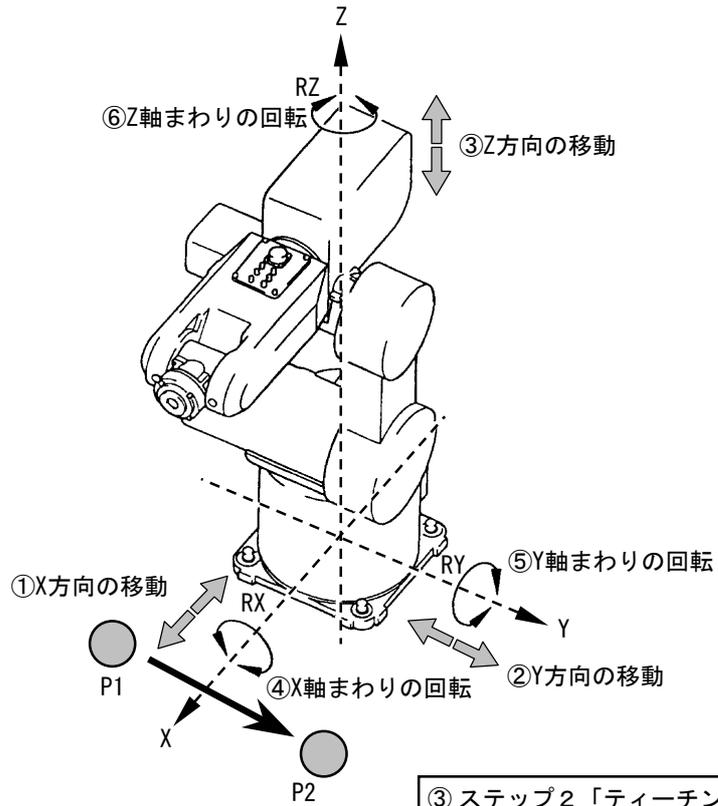
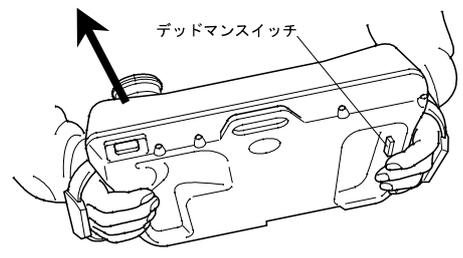
① [Cancel] を 2 回押して、[ロボット現在位置] ウィンドウに戻す。





② デッドマンスイッチを押しながら、移動方向キーを適宜押して、P2に設定したい位置までロボットを動かす。

- 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 RXの回転
- ⑤ RYの回転
- ⑥ RZの回転



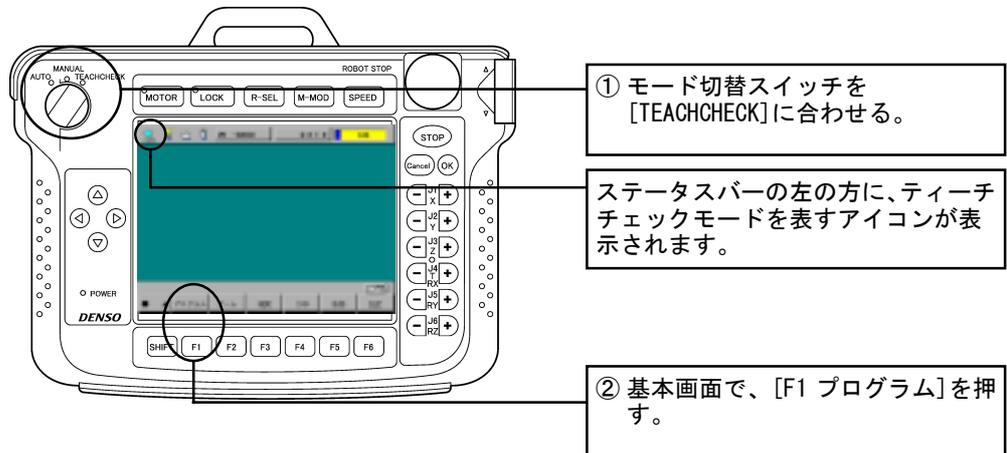
③ ステップ2「ティーチングした値を[変数名P1]に保存する」の手順に従って、P2の値を[変数名P2]に保存する。

これで、P1とP2のティーチングが完了しました。

2.3 ティーチチェック

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。

ステップ1 ティーチチェックモードにする



[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>

ステップ2

ステップ送りチェックを行う



① [プログラム一覧] ウィンドウで「PRO 1」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] ウィンドウのPRO1のプログラムコードが表示されます。



③ [F6 STEP送] を押す。
(ステップ送りは右カーソルでも可能。)



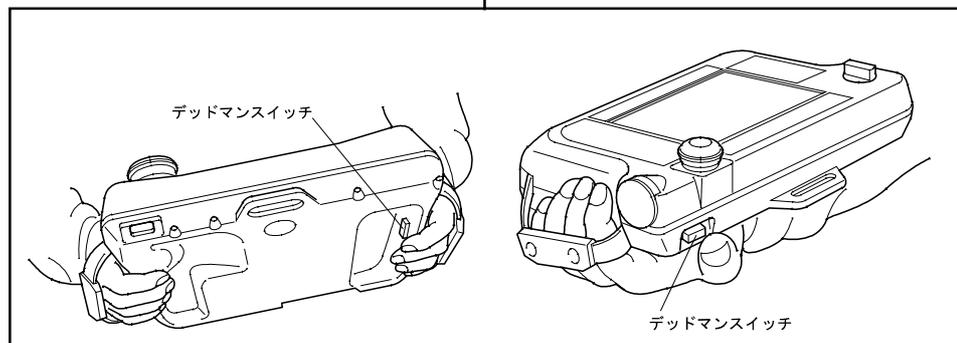
システムメッセージが表示されます。

★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。



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。

ステップ2

サイクルチェックを行う

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。



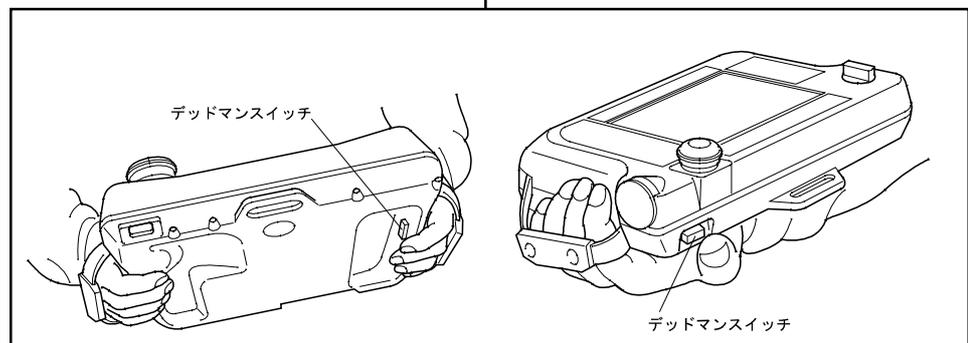
システムメッセージが表示されます。

★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。
(サイクルチェックを中止するには、[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。



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進みます。

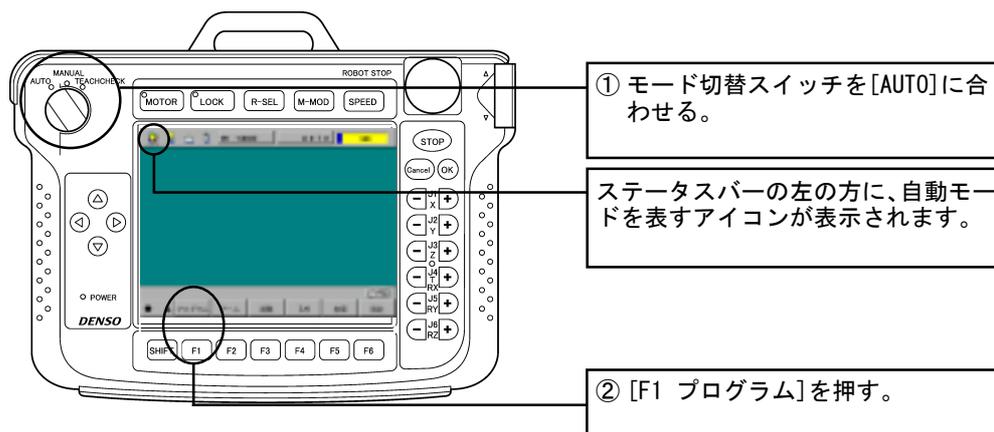
プログラムエンドまで実行すると止まります。

2.4 ロボットを自動運転で動かす

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よう。前節で編集したプログラムPR01を使います。

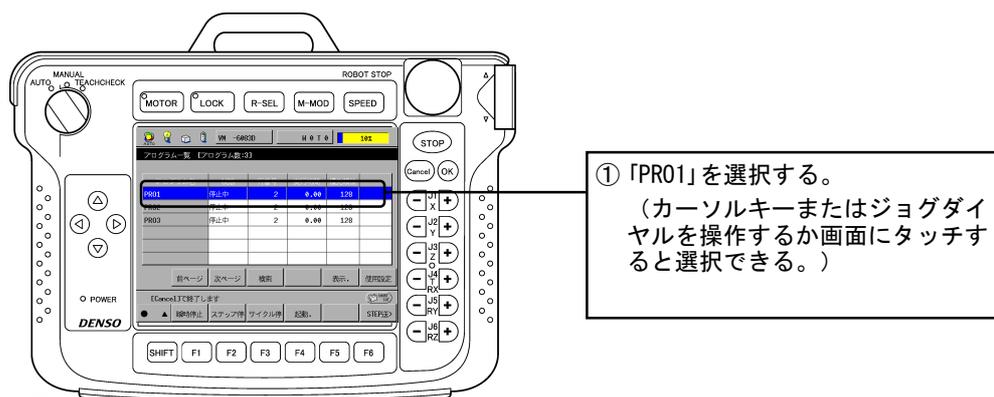
注意：初めて自動モードで運転するプログラムは、速度設定を10%程度にして、運転を始めてください。手動モードやティーチチェックモードでは、自動モードの10%の速度で動くようになっていますが、自動運転では最高速度で動かすことができます。

ステップ1 自動モードにする



ステップ2 プログラムを選択する

[プログラム一覧]ウインドウで、自動運転させるプログラムを選択しましょう。

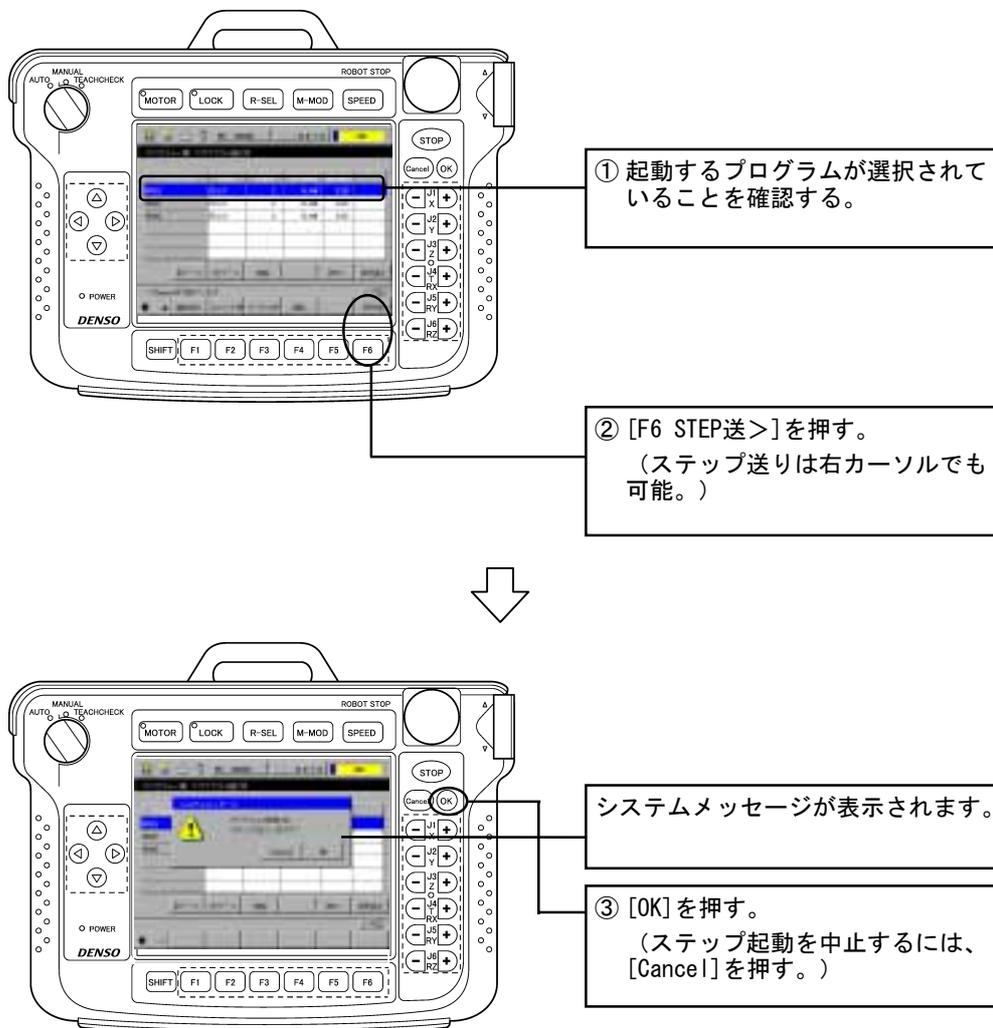


ステップ3

ステップ起動する

★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★

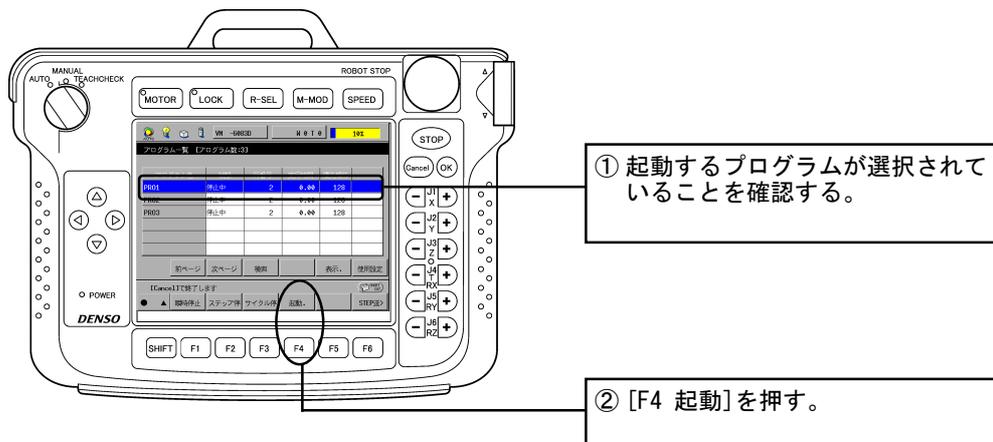
プログラム確認時はいつでも [STOP] を押せるようにして運転してください。

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。

ステップ 4

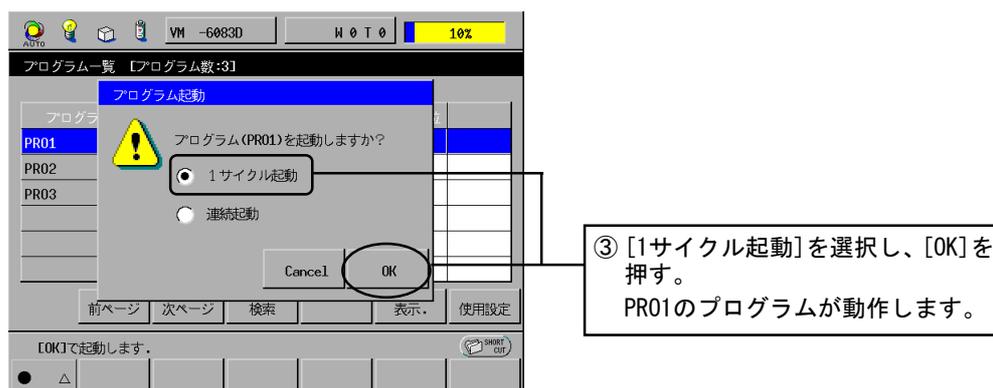
サイクル起動する

ステップ起動を確認したら、サイクル起動させましょう。



★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。



プログラムエンドまで実行すると止まります。

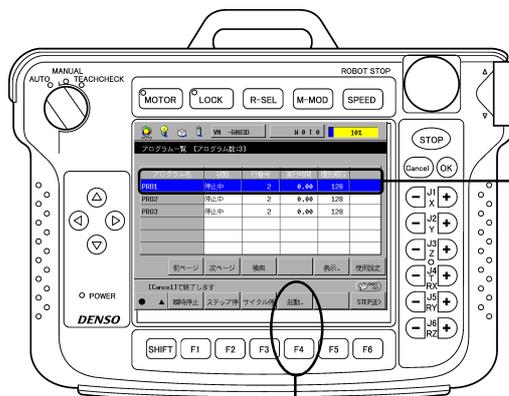
★注意★

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。

ステップ5

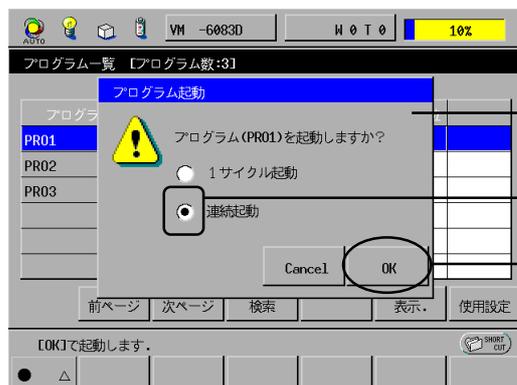
連続起動する

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」を選択する。

④ [OK]を押す。
PR01プログラムが連続動作する。
(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★

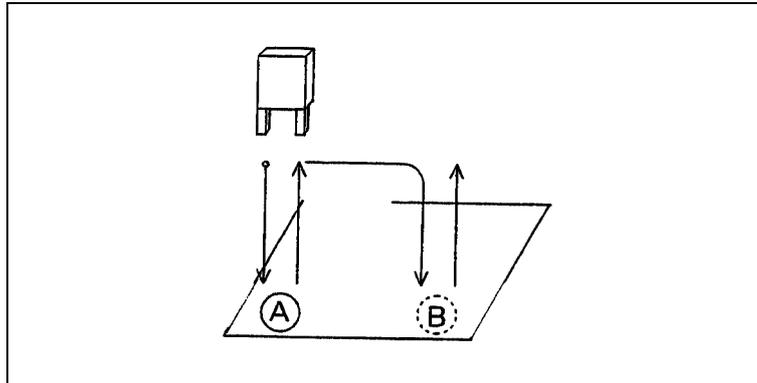
プログラム確認時はいつでも[STOP]を押せるようにして運転してください。



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。

第 1 部の練習問題

課題 ①のワークを②へ移動させるプログラムを作成し、動作確認を行ってください。



- ・ I/O割付け 番号64 (専用出力) …ハンド閉
番号65 (専用出力) …ハンド開
- ・ ワークを取りに行く時と置く時のスピードは30%で、その他は80%とする。
- ・ アプローチ長とデパート長は、全て50mmとする。
- ・ 補間方法は、全てPTP制御とする。

解答

```
0001 ' !TITLE "P&P"
0002 PROGRAM PR010
0003     TAKEARM
0004     APPROACH P , P5 , 50 , S=80
0005     MOVE P , P5 , S=30
0006     DELAY 500
0007     RESET IO[65]
0008     SET IO[64]
0009     DEPART P , 50 , S=80
0010     APPROACH P , P6 , 50 , S=80
0011     MOVE P , P6 , S=30
0012     DELAY 500
0013     RESET IO[64]
0014     SET IO[65]
0015     DEPART P , 50 , S=80
0016     GIVEARM
0017 END
```

第2部 WINCAPS II を使ってパソコンで プログラムを作成する

第2部で学習すること

パソコン教示システムWINCAPS IIをパソコン上で起動し、実際にプログラムを作成、コンパイルします。次にプログラムをパソコンからロボットコントローラにアップロードします。

また、第2部では、ロボットコントローラをマシンロック状態にすることも行います。これは、第3部で、作成したプログラムを使って、ロボットを実際には動かさずに、パソコン画面上でシミュレーション動作を行うために、ロボットコントローラをマシンロック状態にしておく必要があるからです。

▶ レッスン3	ティーチングペンダントを使ってロボットコントローラの準備を行う	40
3.1	キャリブレーション (CAL) を実行する	40
3.2	マシンロック状態にする	43
3.3	通信ポートを設定する	44
▶ レッスン4	WINCAPS IIを起動し、システムプロジェクトを作る	48
4.1	システムマネージャを起動する	48
4.2	システムプロジェクトを新規登録する	50
4.3	パソコンの通信ポートを設定する	51
▶ レッスン5	マクロを定義する	53
▶ レッスン6	プログラムを入力・編集する	55
6.1	プログラムサンプル	55
6.2	プログラム編集ウィンドウを開く	56
6.3	プログラムを入力する	57
6.4	コマンドビルダを使用する	58
6.5	プログラムを保存する	60
▶ レッスン7	プログラムをコンパイルし実行形式にする	61
▶ レッスン8	プログラムをアップロードする (パソコン→ロボットコントローラ)	62

レッスン3 ティーチングペンダントを使ってロボット コントローラの準備を行う

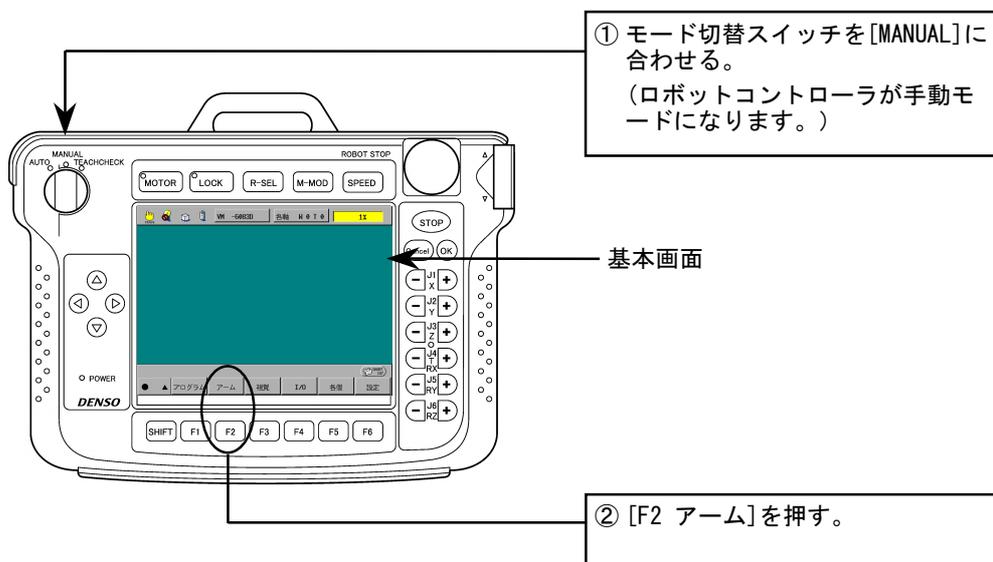
3.1 キャリブレーション (CAL) を実行する

ロボットの現在位置を正確に把握するために、キャリブレーション (CAL) が必要です。CALは、すべての軸を微小動作させます。



CALを行うとロボットが動作します。作業者はロボットの可動制限範囲から出て
ください。ロボットの可動制限範囲に障害物がないことを確認してから、CALを
行ってください。

ステップ1 手動モードにする



ステップ2

CALを開始する



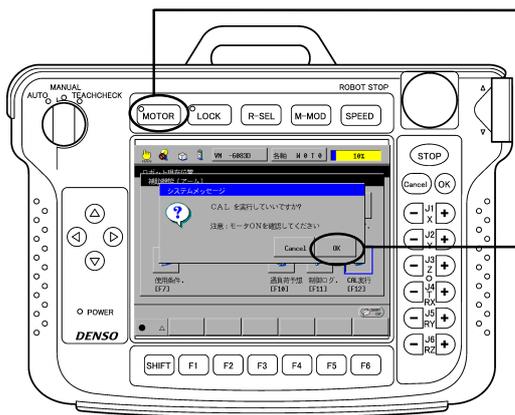
① [F6 補助機能]を押す。



② [F12 CAL実行]を押す。

ステップ3

モータ電源を入れる



① [MOTOR]を押す。
モータ電源がONになり、[MOTOR]ランプが点灯します。

② [OK]を押す。

ステップ4

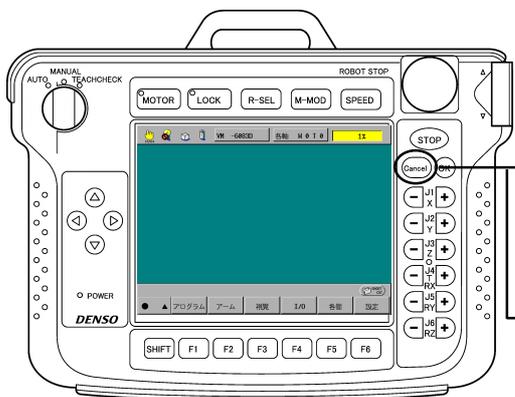
CALを終了する



① [OK]を押す。



これでCALは完了しました。



② [CANCEL]を押す。
([ロボット現在位置]ウインドウの表示に戻ります。)

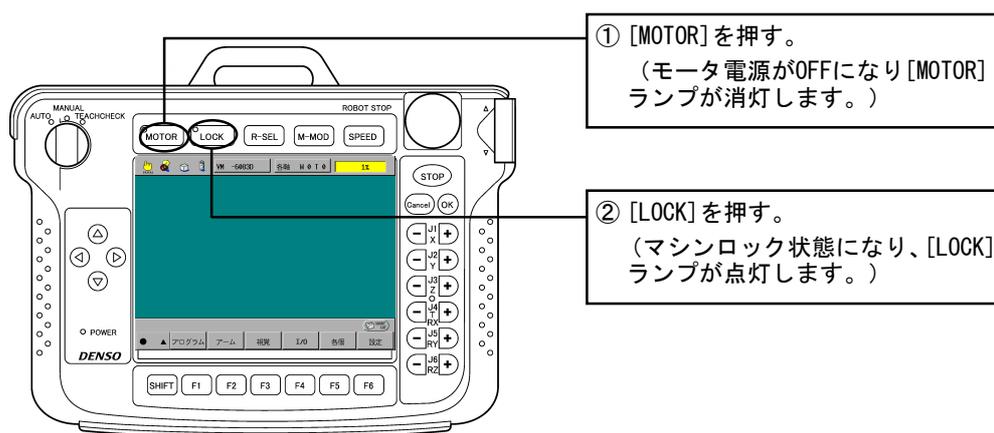
③ [CANCEL]を押す。
(基本画面に戻ります。)

3.2 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、第3部で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



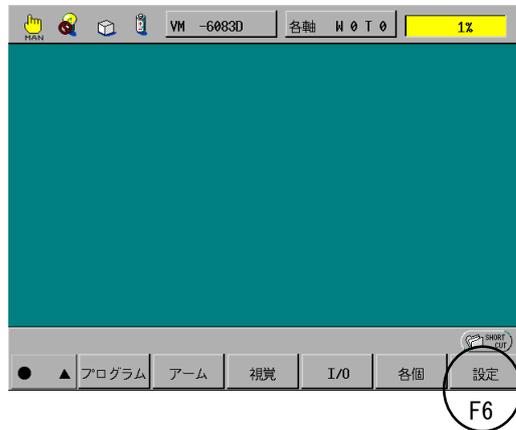
: I/O出力制限あり

3.3 通信ポートを設定する

ロボットコントローラがパソコンと通信を行えるように、通信ポートの設定を行います。ここでは、RS-232Cを使った最も一般的な接続方法を取り上げます。

ステップ1

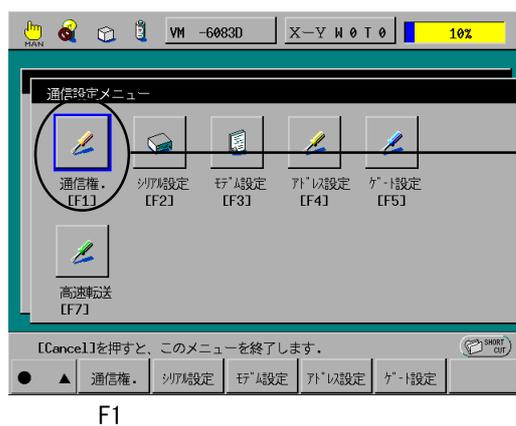
通信権を設定する



① [F6 設定]を押す。



② [F5 通信設定]を押す。



③ [F1 通信権.]を押す。



④ [COM2 (RS232C)]の欄を選ぶ。

⑤ [F5 設定変更.]を押す。



⑥ [読込/書込可]を選択する。

⑦ [OK]を押す。



[COM2 (RS232C)]の欄が、[読込/書込可]になります。

⑧ [OK]を押す。

ステップ2

通信速度を設定する



F2



F5



⑤ [OK]を押す。





⑦ [Cancel] を押す。
 ([設定 (メイン)] ウィンドウの
 表示に戻ります。)

⑧ [Cancel] を押す。
 (基本画面に戻ります。)

レッスン4 WINCAPS を起動し、システムプロジェクトを作る

パソコンでWINCAPS を起動し、システムプロジェクトを新規登録します。これは、プログラムを入力、編集、検証するために必要です。また、パソコンの通信ポートを設定します。

4.1 システムマネージャを起動する

WINCAPS は以下の機能モジュールから成り立っています。

- PACプログラムマネージャ
- 変数マネージャ
- DIOマネージャ
- アームマネージャ
- 視覚マネージャ
- ログマネージャ
- 通信設定マネージャ

これらの機能モジュール群を統合管理する役目を持っているのが、システムマネージャと呼ばれるプログラムです。WINCAPS のすべての機能は、システムマネージャから呼び出して使用します。パソコン教示システムを利用するためには、まずシステムマネージャを起動します。

ステップ 1 System Managerを選択する



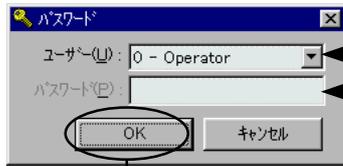
[スタート] から、WINCAPS のフォルダ内の[System Manager]を起動する。

注意

- インストール後、初めて起動する場合は、プログラムバンクのファイル名を指定する必要があります。必要に応じて [プログラムバンク新規作成] ダイアログが表示されますので、その場合は、WINCAPS II ガイド「5.6.2.3 プログラムバンクの更新」を参照してください。
- システムマネージャを初めて起動する場合は、システムプロジェクトがまだ定義されていないために、[プロジェクトの新規作成] ダイアログが表示されます。その場合は、「4.2 システムプロジェクトを新規登録する」を行ってから、次のステップ2に進んでください。

ステップ 2

ユーザレベルを選択する



ユーザレベルをポップアップメニューで選ぶ。

必要に応じてパスワードを入力する。

[OK]を押す。



[いいえ]をクリックする。

ポイント

ツールバー上の各機能モジュール起動ボタン



システムマネージャが起動し、[System Manager]ウィンドウが表示されます。アイコンのあるボタンは、各機能モジュールを起動するためのものです

- 通信設定ボタン
- ログマネージャボタン
- 視覚マネージャボタン
- アームマネージャボタン
- DIOマネージャボタン
- 変数マネージャボタン
- PACプログラムマネージャボタン

4.2 システムプロジェクトを新規登録する

WINCAPS II では、プロジェクトという単位で、複数のロボットプログラムを管理します。1台のロボットを動かすためには、いくつかのロボットプログラムを組み合わせることで使えます。1台のロボットのための一揃いのロボットプログラムを、一つのプロジェクトとして管理すると便利です。

ロボットプログラムを作るに先立って、プロジェクトを新しく登録してみましょう。

注意： 初めてシステムマネージャを起動した場合は、[プロジェクトの新規作成] ダイアログが表示されますので、「新規プロジェクトの登録」を行ってから、「4.1 システムマネージャを起動する」のステップ2を行ってください。

ステップ1 [ファイル]メニューから「プロジェクトの新規作成 (N)」を選択する



① ここをクリックする。

ステップ2 新規プロジェクトを登録する



① VM/VS/VC を選ぶ。

② VM-60B1D を選ぶ。

③ 標準 を選ぶ

④ プロジェクト名を入力する。
(ここではVM-60B1Dと入力する。)

⑤ フォルダ名を指定する。

⑥ 実行プログラムのバージョンを
指定する

⑦ 0-標準 を選ぶ。

⑧ [OK]をクリックする。

4.3 パソコンの通信ポートを設定する

WINCAPS IIの通信設定を、ロボットコントローラの通信設定と合わせましょう。

ステップ1 [ROBOTalk Manager]ダイアログを表示する



① [通信設定] ボタンをクリックする。

★注意★ パスワードをまだ入力していない場合には、[パスワード] ダイアログが表示されますので、ユーザレベルを選択して、パスワードを入力してください。「4.1 システムマネージャを起動する」のステップ2を参照してください。

ステップ2 通信デバイスと設定オプションを設定する



② [ROBOTalk] タブをクリックする。

③ ここをクリックしてRS232Cを選択する。

④ これらスイッチで設定オプションを設定する。
ここでは、タイムアウトを4000msec、リトライ回数を5回、通信リトライ回数を5回に設定する。

★注意★ データ転送時にタイムアウトが発生するようであれば、上記のタイムアウト時間を長くして時間を調整してください。

ステップ3 RS232C通信オプションを設定する



⑤ [RS232C] タブをクリックする。

⑥ [標準] をクリックする。
[設定オプション]の欄が、標準設定になります。

⑦ [OK] をクリックする。
設定内容が有効になり、[ROBOTalk Manager]ダイアログは閉じます。

★注意★ 標準設定以外の設定をする場合は、ロボットコントローラや、ご使用のパソコンの仕様に合わせて、設定内容を決定してください。

★注意★ [OK]が無効になっている場合は、すべてのマネージャの[接続]ボタンをOFFしてください。いずれかのマネージャが接続状態にある場合は、通信設定を変更することはできません。

WINCAPS IIの通信設定は一度行なえば、次に変更するまで有効です。WINCAPS IIを起動するたびに、再設定する必要はありません。WINCAPS IIを起動したときに出る [DensoROBOTalk] ダイアログメッセージ「接続を復帰しますか？」に「はい」と答えると、前回の設定のままになります。「いいえ」と答えると、WINCAPS IIの通信設定は行ないません。



レッスン5 マクロを定義する

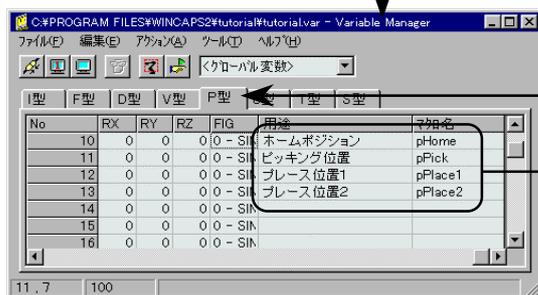
変数とI/Oのマクロ名と用途を定義し、マクロ定義ファイルを作成しましょう。

ステップ1

変数マクロ定義ファイルを作る



① 変数マネージャボタンをクリックして、変数マネージャを起動する。



[Variable Manager] ウィンドウが表示されます。

② [P型] タブをクリックする。

③ 入力する欄をダブルクリックし、各欄に変数の用途とマクロ名を入力する。
ここではP型変数P10~P13を入力する。



④ [マクロ定義ファイルの作成] をクリックする。



⑤ [OK] ボタンを押す。
これでマクロ定義ファイルができました。



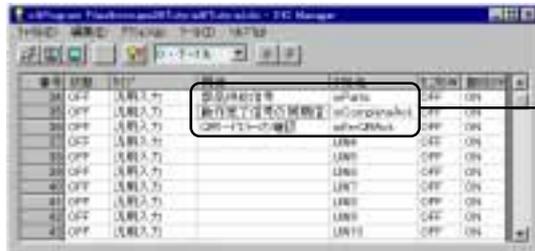
⑥ ここをクリックして変数マネージャを終了する。

ステップ2

I/Oマクロ定義ファイルを作る



① DIOマネージャボタンをクリックして、DIOマネージャを起動する。



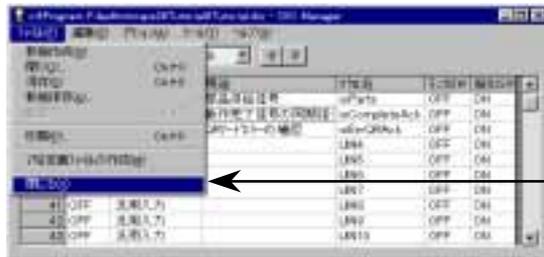
② 入力する欄をダブルクリックし、各欄にI/Oの用途とマクロ名を入力する。



③ [マクロ定義ファイルの作成]をクリックする。



④ [OK]ボタンを押す。
これでマクロ定義ファイルができました。



⑤ [閉じる]をクリックして、DIOマネージャを終了する。

レッスン6 プログラムを入力・編集する

6.1 プログラムサンプル

プログラムの入力手順に入る前に、コーディングリストのサンプルを見てみましょう。動作を理解するために、まず良くお読みください。コメントを参考にしてください。

コーディングリスト「PR01」

```
'!TITLE “ピック&ブレース”
#INCLUDE “dio_tab.h”           ' DIO マクロ定義ファイルの読み込み
#INCLUDE “var_tab.h”           ' 変数マクロ定義ファイルの読み込み

PROGRAM pro1
  TAKEARM                       ' アームセマフォ取得
  SET IO[ioComplate]
  MOVE P, P[pHome], S=50        ' ホームポジションへ内部速度を 50%で PTP 移動
  SPEED 100                     ' 内部速度を 100%に
  APPROACH P,P[pPick],200
  MOVE P,P[pPick]
  GOSUB *ChuckItem
  DEPART P,200
  APPROACH P,P[pPlace1],200
  MOVE P,P[pPlace1]
  GOSUB *UnchuckItem
  DEPART P,200
  SET IO[ioComplate]           ' 動作完了信号出力
  GIVEARM                       ' アームセマフォ解放
  END

' ===== 部品チャック =====
*ChuckItem:
RESET IO[ioUnChuck]
SET IO[ioChuck]
RETURN

' ===== 部品アンチャック =====
*UnchuckItem:
RESET IO[ioChuck]
SET IO[ioUnChuck]
RETURN
```

6.2 プログラム編集ウィンドウを開く

ロボットプログラムの入力と編集はプログラムマネージャで行ないます。プログラムマネージャはシステムマネージャから呼び出して起動します。

ステップ1 PACプログラムマネージャを起動する

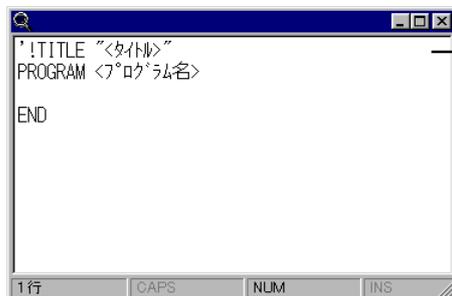


① PACプログラムマネージャボタンをクリックして、PACプログラムマネージャを起動する。

ステップ2 プログラム編集ウィンドウを開く



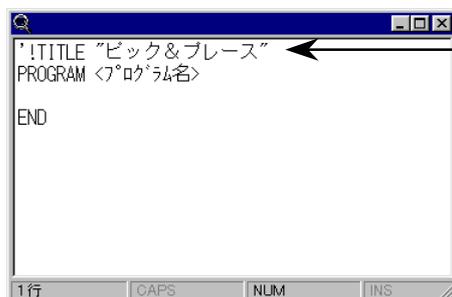
② [新規プログラム]ボタンをクリックする。



プログラムを編集するための新しい編集ウィンドウが表示されます。

6.3 プログラムを入力する

ステップ1 プログラムタイトルを入力する



```
!TITLE "ピック&ブレース"  
PROGRAM <プログラム名>  
  
END
```

① プログラムのタイトルを入力する。(ここでは「ピック&ブレース」というタイトルを入力する。)

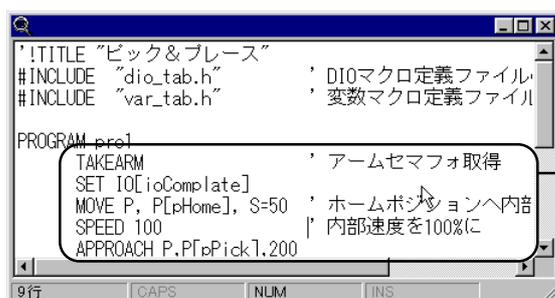
ステップ2 プログラム名を入力する



```
!TITLE "ピック&ブレース"  
PROGRAM pro1  
  
END
```

② プログラム名を入力する。(ここではPR01と入力する。)

ステップ3 プログラムコードを入力する



```
!TITLE "ピック&ブレース"  
#INCLUDE "dio_tab.h" ' DIOマクロ定義ファイル  
#INCLUDE "var_tab.h" ' 変数マクロ定義ファイル  
  
PROGRAM pro1  
  TAKEARM ' アームセマフォ取得  
  SET IO[ioComplate]  
  MOVE P, P[pHome], S=50 ' ホームポジションへ内部  
  SPEED 100 ' 内部速度を100%に  
  APPROACH P,P[pPick1],200
```

③ 「ピック&ブレースのプログラムコードを入力する。

6.4 コマンドビルダを使用する

プログラム編集ウィンドウへの入力には、もちろんキーボードを使ってワードプロセッサのように入力できますが、もう一つの方法として、プログラムマネージャには簡単にコマンドを入力するための、コマンドビルダという機能が備わっています。ここでは、コマンドビルダを使って入力してみましょう。

ステップ1

コマンドビルダを選ぶ



① [ツール (T)]メニューから[コマンドビルダ (C)]を選ぶ。

ステップ2

コマンドビルダを使う



② [コマンドビルダ]ウィンドウの[クラス選択]リストをクリックして、リストから「入出力制御文」を選ぶ。

コマンドリスト



③ 入出力制御文のコマンドリストのスクロールバーを操作して、[RESET]コマンドをリストに表示する。

④ [RESET]コマンドをクリックする。

⑤ [I/O変数]の設定値欄をダブルクリックする。



⑥ [I/O変数]の設定値欄に「ioUnChuck」とキーボードから入力する。



```
' ===== 部品チャック =====
*ChuckItem:
  SET IO[ioChuck]
  RETURN
' ===== 部品アンチャック =====
*UnchuckItem:
  RESET IO[ioChuck]
  SET IO[inChuck]
```

⑦ プログラム編集ウインドウのコメントを入力したい位置をクリックする。



⑧ [貼り付け] ボタン  をクリックする。

ステップ3

見やすく編集する



```
!TITLE "ピック&ブレース"
#include "dio_tab.h"
#include "var_tab.h"
PROGRAM pro1
  TAKEARM
  SET IO[ioComplete]
  MOVE P, P[pHome], S=50
  SPEED 100
  APPROACH P, P[pPick], 200
  MOVE P, P[pPick]
  GRSR *ChuckItem
```

★注意★

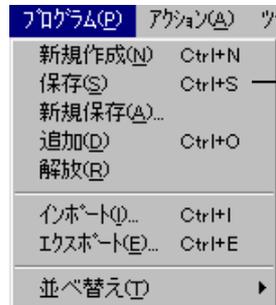
プログラムを保存するために、次節「6.5 プログラムを保存する」を続けて行ってください。

6.5 プログラムを保存する

前節「6.4 コマンドビルダを使用する」で作成したプログラムpro1を保存しましょう。

ステップ1

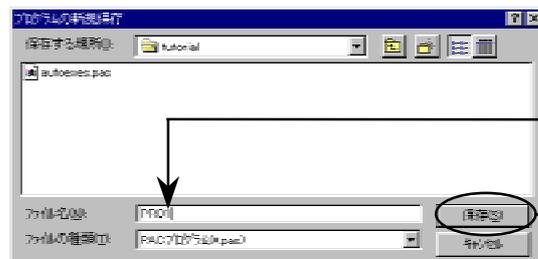
[保存(S)]を選択する



① [プログラム]メニューから[保存(S)]を選択する。

ステップ2

ファイル名を入力する



② [ファイル名(N)]の欄に、ファイル名を入力する。
ここでは、ファイル名に、「pro1」と入力する。
(拡張子は自動的に付加される。)

③ [保存]をクリックする。プログラムソースファイル「pro1.pac」が保存されます。

[PAC Manager]ウィンドウのファイル名が「pro1.pac」に変わります。これで、プログラム「pro1」の入力と保存が完了しました。

レッスン7 プログラムをコンパイルし実行形式にする

PAC言語は、入力したプログラムを実行するために、あらかじめ機械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



① 「実行プログラムの作成」ボタン
をクリックする。

現在選ばれているプロジェクトに含まれるすべてのプログラムが、実行プログラムに変換されます。実行プログラム作成中の履歴が、[DensoPacManager]ウィンドウのメッセージペインに表示されます。

ステップ2 エラーのないことを確認する



② 「エラー数：0」であることを確認
して、[OK]をクリックする。

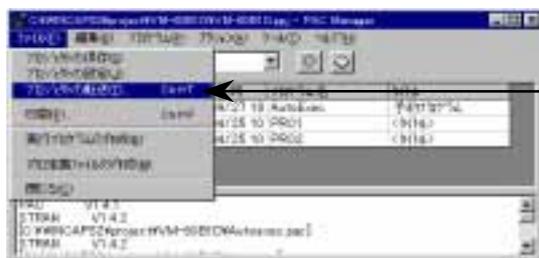
エラーが出たときは、「レッスン6 プログラムを入力・編集する」に戻って、入力に誤りがないか確認してください。

レッスン8 プログラムをアップロードする (パソコン→ロボットコントローラ)

できあがったプログラムのデータは、まだパソコンの中にあります。プログラムを実行するには、このデータを、パソコンからロボットコントローラに転送（アップロード）する必要があります。

レッスン3.3とレッスン4.3で通信設定が済んでいますので、プログラムのデータを、パソコンからロボットコントローラにアップロードしてみましょう。

ステップ1 プログラムマネージャとロボットコントローラの通信を確立する



① [ファイル (F)] メニューから [プロジェクトの転送 (T)] を選ぶ。

ステップ2 アップロードするプログラムを選択する



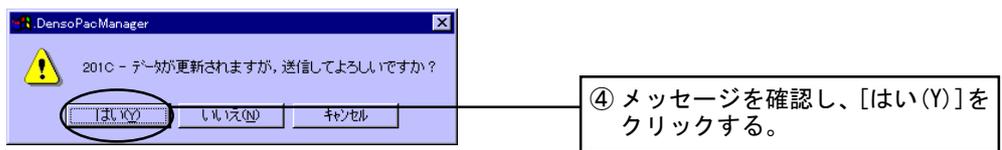
ローカルはパソコン側、リモートはロボットコントローラ側を意味します。

② [全選択 (A)] をクリックする。



ステップ3

プログラムデータをアップロードする



プログラムデータがロボットコントローラにアップロードされます。

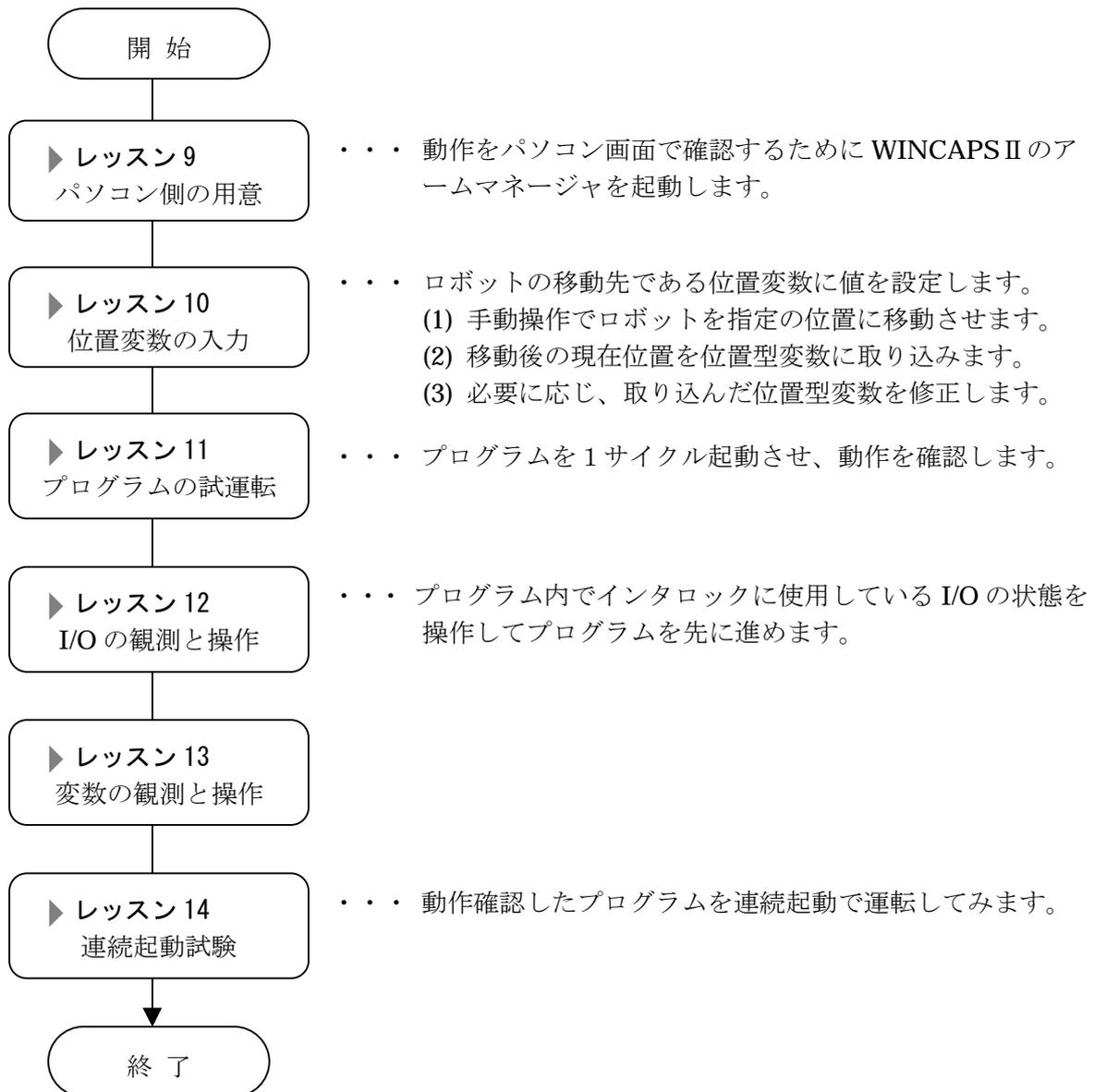
第3部 作成したプログラムを使って パソコンで動作を確認する

第3部で学習すること

第3部では、パソコン上で作成しロボットコントローラへアップロードしたプログラムをマシンロック状態で起動し、パソコンでシミュレート画像を見てプログラム動作をチェックします。

ロボット本体を実際に動作させる前に、ロボットプログラムを検証できるので、安全性とプログラム開発効率の向上に役立ちます。

▶ レッスン 9 パソコン側の用意をする	67
9.1 アームマネージャを起動する	67
9.2 アームマネージャの通信を接続する	67
▶ レッスン 10 位置変数を入力する	68
10.1 ロボットの動きを手動でシミュレートする	68
10.2 現在位置を取り込む	70
10.3 位置変数を編集する	76
▶ レッスン 11 プログラムを試運転する	77
11.1 プログラムをロードする	77
11.2 プログラムを起動する	79
▶ レッスン 12 I/O の観測と操作を行う	80
12.1 DIOマネージャの起動し、通信を接続する	80
12.2 I/Oを観測する	81
12.3 I/Oの擬似SWをON/OFFする	82
▶ レッスン 13 変数の観測と操作を行う	83
13.1 変数マネージャを起動し、通信を接続する	83
13.2 変数を観測する	84
▶ レッスン 14 連続起動試験を行う	85
14.1 連続起動する	85
14.2 I/Oを連続観測する	86
14.3 プログラムを停止させる	88



レッスン9 パソコン側の用意をする

前章で作成したロボットプログラムの動作を、パソコンのシミュレーション映像を見ながらテストしてみましょう。

9.1 アームマネージャを起動する

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マネージャを起動します。アームマネージャは、システムマネージャから呼び出します。

ステップ1

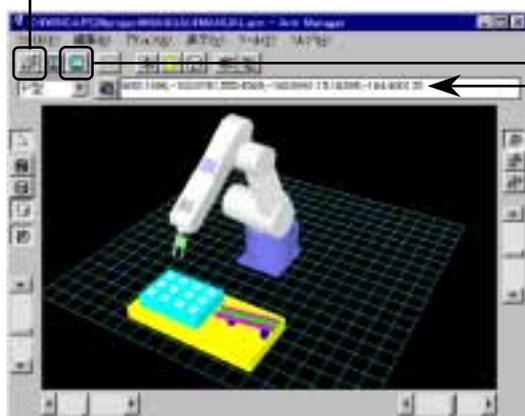


① [アーム]ボタンをクリックする。

9.2 アームマネージャの通信を接続する

アームマネージャとロボットコントローラの間で、データを常時やり取りするよう、通信を接続します。

ステップ1



① [接続]ボタンをクリックする。
([接続]ボタンが押し込まれたように表示されます。)

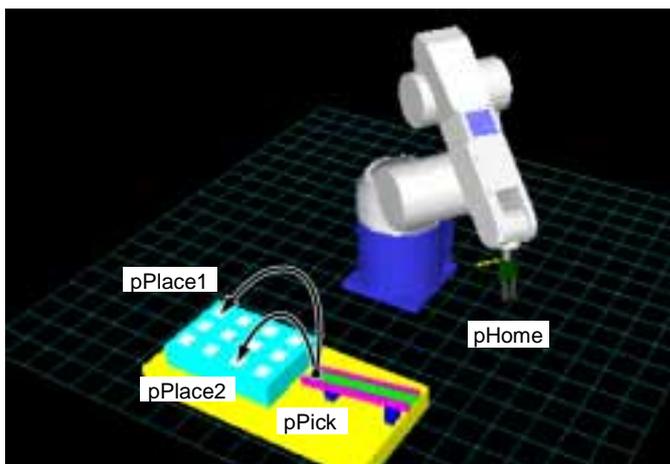
② [連続モニタ]ボタンをクリックする。
([連続モニタ]ボタンが押し込まれたように表示されます。)

以上で、アームマネージャとロボットコントローラの間でのデータ通信が常時行なわれるようになります。
[現在位置]ウィンドウには、ロボットの現在位置が表示されます。

レッスン10 位置変数を入力する

プログラムを動かす前に、第2部「WINCAPS IIを使ってパソコンでプログラムを作成する」の「レッスン6 プログラムを入力・編集する」で作成したプログラムの「pHome」、「pPick」、「pPlace1」、「pPlace2」の各位置変数の値を決めておく必要があります。

ここでは、「ポイント教示」と呼ばれる方法を使ってティーチングペンダントから入力してみましよう。

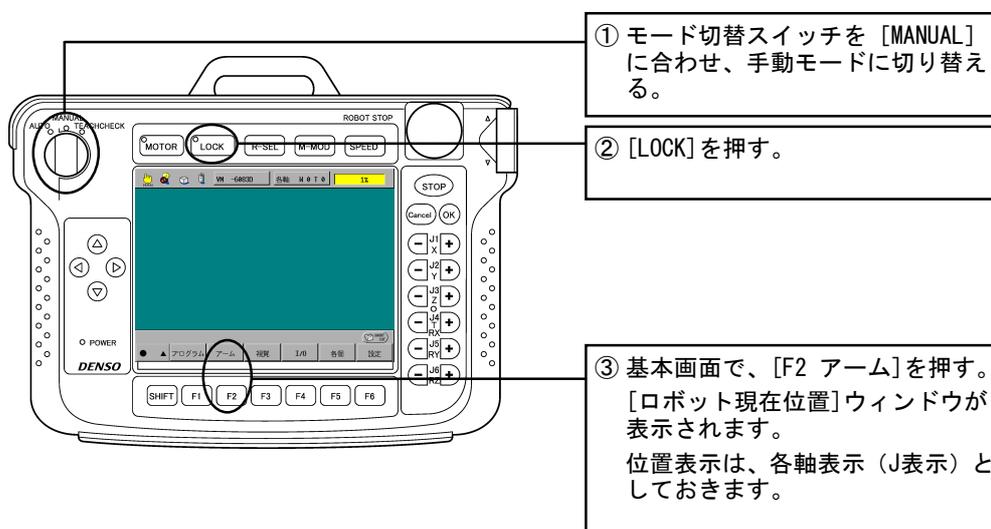


★注意★ 「pHome」、「pPick」、「pPlace1」、「pPlace2」の各位置変数番号は、「レッスン5 マクロを定義する」を参照してください。

10.1 ロボットの動きを手動でシミュレートする

アームマネージャに表示されるロボットのシミュレーション画像を見ながら、位置変数「pHome」とする位置まで手動動作でロボットを動かしてみましよう。以下に説明する手順で行います。

ステップ1 手動モードにして、ロボット現在位置を表示させる



ステップ2

ロボットを移動させる



④ ロボットのシミュレーション画像を見ながら、デッドマンスイッチと移動方向キーを使って、「pHome」とする位置まで動かす。

ここでは、各軸の値が以下のようにになるようする。

- ・ J1 : 約40°
- ・ J2 : 0 (動かさない)
- ・ J3 : 約100°
- ・ J4 : 0 (動かさない)
- ・ J5 : 約70°
- ・ J6 : 0 (動かさない)

ステップ3

現在位置を確認する



⑤ [ロボット現在位置]ウィンドウで、各軸の位置を確認する。

10.2 現在位置を取り込む

アームマネージャで移動させた位置を位置型変数に取り込みましょう。

位置型変数に現在位置を取り込むには、大きく分けて次の3つの方法があります。

- [1] ティーチングペンダントの位置型変数表示画面にて取り込む方法
- [2] ティーチングペンダントのプログラム表示画面にてプログラム中で指定されている変数に取り込む方法
- [3] オペレーティングパネルを用いる方法

ここでは、以上3つの方法で現在位置を変数に取り込む手順について説明します。

[1] ペンダントの位置型変数表示画面にて取り込む

ステップ1

位置型変数を表示させる



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、[F4 変数]を押す。



② [変数タイプ選択]ウィンドウが表示されたら、[F4 P.位置]を押す。



[位置型]ウィンドウが表示されます。

ステップ2

現在位置を取り込む



③ ジョグダイヤルまたはカーソルキーを使って、「P10」の欄を選択する。

④ [F6 位置取込.]を押す。



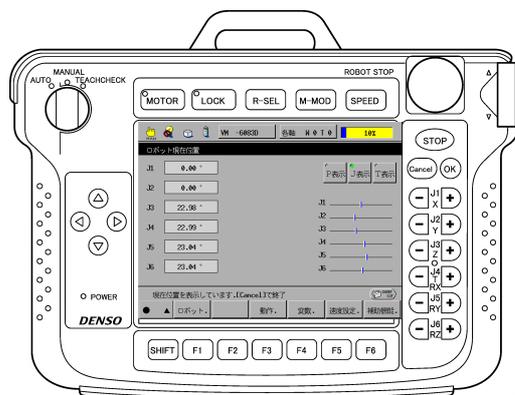
⑤ [OK]を押す。
位置型変数P[10]に現在位置が取り込まれます。

⑥ [Cancel]を2回押す。
[ロボット現在位置]ウィンドウの表示に戻ります。

⑦ レッスン10.1, ステップ2「ロボットを移動させる」からの操作を繰り返す。

ステップ3

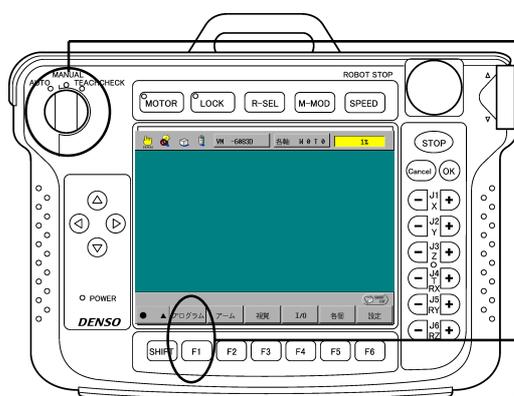
他の位置を取り込む



ステップ1、2と同様の方法で他の位置を取り込む。

[2] ティーチングペンダントのプログラム表示画面にてプログラム中で指定されている変数に取り込む

ステップ1 プログラム一覧表示ウインドウを表示します



① モード切替スイッチを[MANUAL]に合わせる。

② 基本画面で[F1 プログラム]を押し、[プログラム一覧]ウインドウを表示します。

ステップ2 対象のプログラムを選択、表示します



③ 対象プログラムを選択します。

④ [表示.]もしくは[F5 編集]を押し、プログラムの内容を表示します。

ステップ3 取り込み対象となる変数が含まれる行を選択します



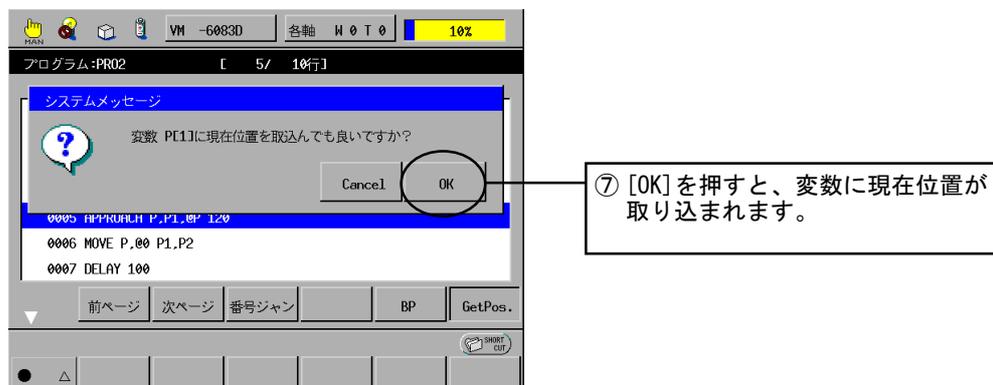
⑤ 対象となる行を選択します。

⑥ [GetPos.]を押します。

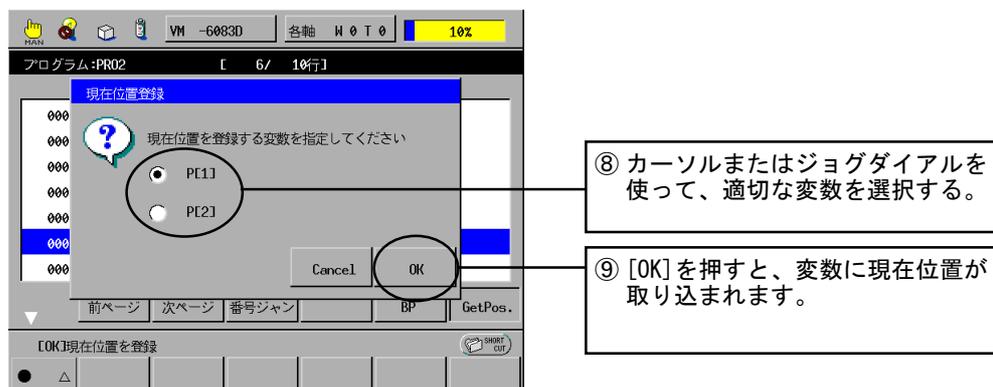
ステップ4

行に含まれている変数に現在位置を取り込みます

変数の候補が一つの場合には以下の様な画面が表示されます。



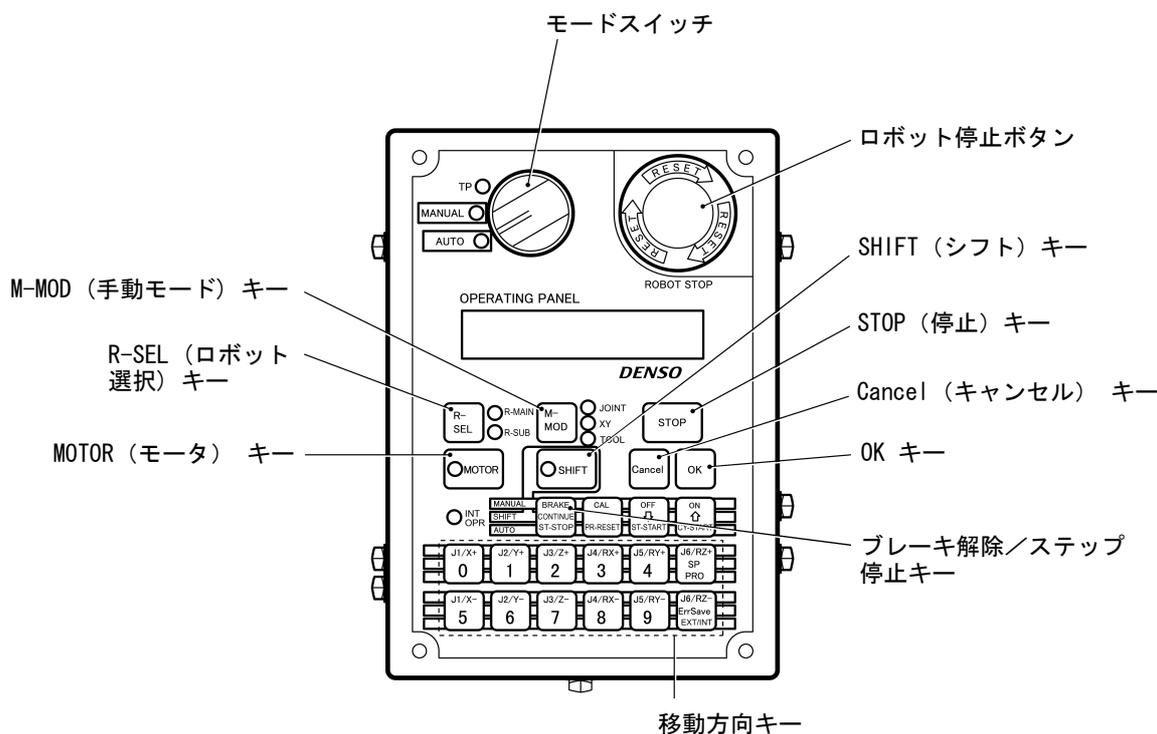
変数の候補が複数ある場合には以下の様な画面が表示されます。適切な変数を選択してください。



プログラム行に含まれている変数が、適切でない場合は候補の対象外となり、教示対象としては表示されません。

また、候補として表示される変数は、プログラム行中の先頭3つまでの変数です。

[3] オペレーティングパネルを用いて現在位置を変数に取り込む



ステップ 1 手動モード

オペレーティングパネルを手動モードにします。この時、オペレーティングパネルの手動モードLEDが点灯していることを確認してください。

ステップ 2 機能選択モードに入る

[SHIFT]キーを押してください。この時[SHIFT]キーのLEDが点灯することを確認してください。その後、[ブレーキ解除/ステップ停止]キーを押すと機能選択モードになります。

ステップ 3 位置取り込み機能選択を行う

[↑][↓]キーを押して、様々な機能選択をすることができます。ここで、[OK]キーを押すと、機能が選択されます。

P変数に現在位置を取り込む場合は[F 7:Set CurPos P]を、J変数、T変数の場合は[F 8:Set CurPos J]、[F 9:Set CurPos T]を選択後、[OK]キーを押してください。

ステップ4 位置取り込み変数番号選択を行う

ここでオペレーティングパネルの数字キーを押すと、変数番号の入力ができます。[OK]キーを押すと変数番号が確定します。

また、ここで[Cancel]キーを押すと、ステップ3へ戻ります。

ステップ5 位置取り込み実行

この時点で、オペレーティングパネルのLCDには[SetCurrentPos?]と表示されます。[OK]キーを押します。

また、ここで[Cancel]キーを押すと、ステップ4へ戻ります。

位置取り込みに成功すると、オペレーティングパネルのLCDに[SetVarVal OK]と表示されます。

10.3 位置変数を編集する

取り込んだ位置変数は、必要に応じて編集することができます。ここでは、様子を知るために簡単な数値の入れ替えを試してみましょう。

ティーチングペンダントには、[位置型] ウィンドウが表示されています。

ステップ 1

位置型変数を選択する



① ジョグダイヤルまたはカーソルキーを使って、位置型変数「P13」の[X]の欄を選択する。

② [F5 値変更.]を押す。
[値変更]ウィンドウが表示されます。

ステップ 2

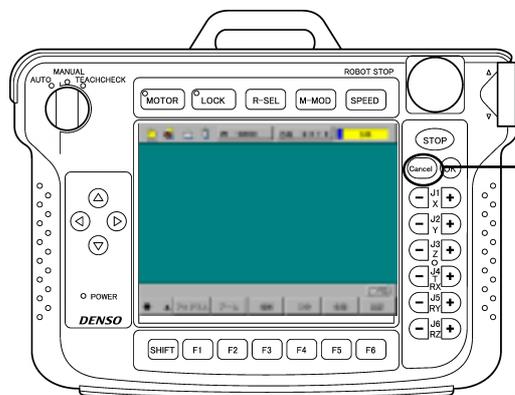
値を変更する



① [値変更] ウィンドウのテンキーを使って、新しい値を入力する。
あまりかけ離れた値を入れるとエラーになるので、ここでは、元の値に近い「319」を入力する。

② [OK]を押す。
位置型変数「P13」の[X]の欄に「319」が入力されます。

基本画面に戻す



③ [Cancel]を3回押して、基本画面に戻す。
これで、位置変数の編集ができました。

レッスン11 プログラムを試運転する

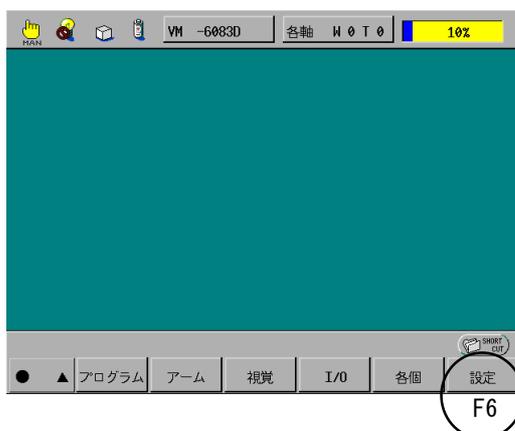
ここでは、プログラム「PR01」をサイクル起動してみましょう。

11.1 プログラムをロードする

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。ロボットコントローラに転送されたデータを、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。

ステップ1

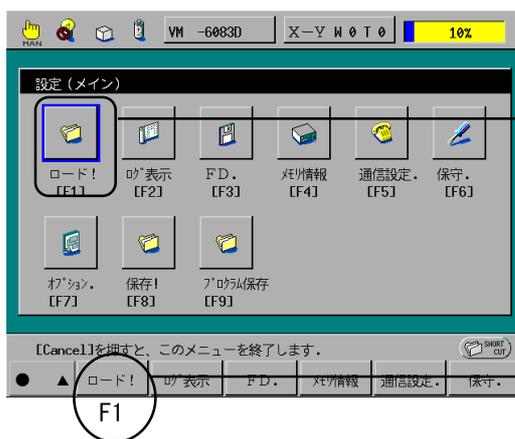
ロード画面を表示する



① 基本画面で[F6 設定]を押す。

ステップ2

プロジェクトをロードする



② [F1 ロード!]または[ロード!]ボタンを押す。

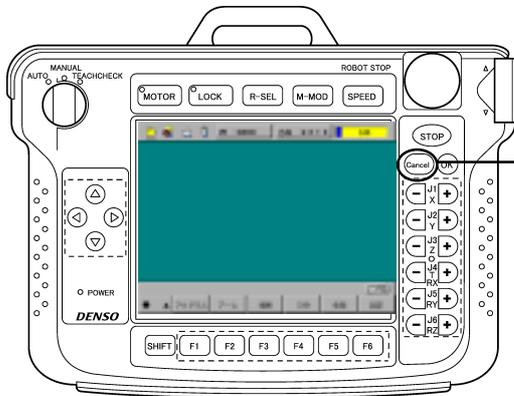


③ システムメッセージを確認して、
[OK]を押す。



ロードが完了すると、設定(メイン)
ウィンドウに戻ります。

基本画面に戻る



④ [Cancel]を押して、基本画面に戻す。

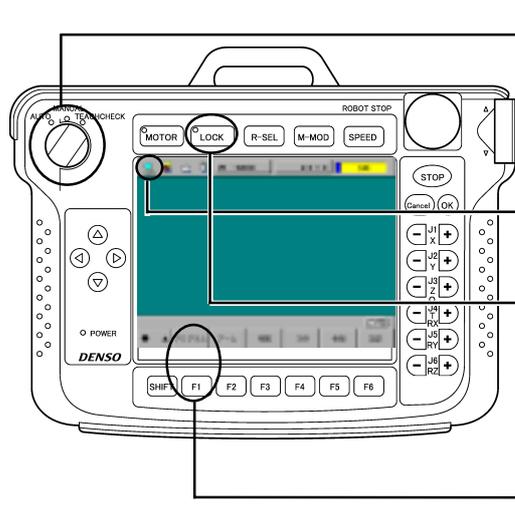
11.2 プログラムを起動する

ロードしたプログラムをパソコン上のシミュレーション映像で1サイクル起動させて、動作を確認してみましょう。実際のロボットは、動作しないようにマシンロック状態にしてください。

途中で、部品供給確認信号のI/O待ちのために、プログラムは「待機中」になりますが、レッスン12でI/Oを操作しますので、そのまま先に進んでください。

ステップ1

ティーチチェックモードでプログラム一覧ウィンドウを表示する



① モード切替スイッチを [TEACHCHECK] に合わせる。
(ロボットコントローラがティーチチェックモードになります。)

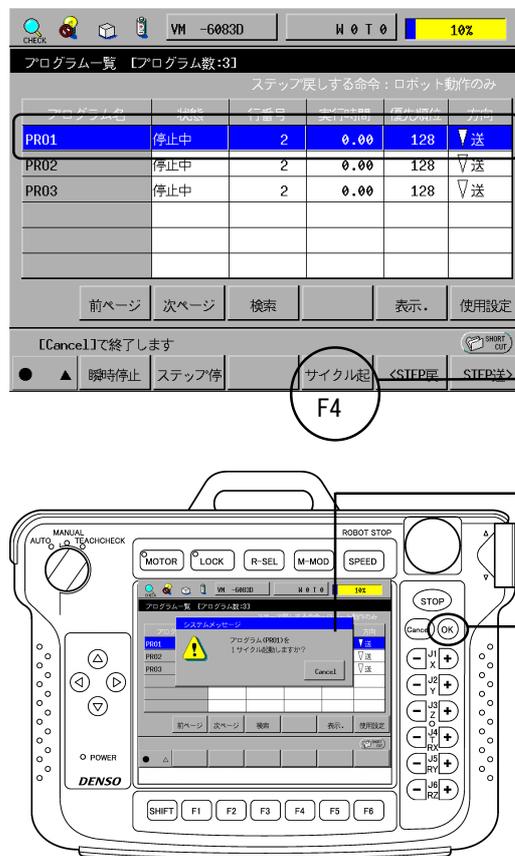
② 画面左上のモードアイコンが  に切り替わったことを確認する。

③ [LOCK] を押しロボットをマシンロック状態にする。[LOCK] LEDが点灯したことを確認する。

④ [F1 プログラム] を押し、[プログラム一覧] ウィンドウを表示する。

ステップ2

プログラムを選択、起動する



プログラム名	状態	実行回	実行時間	残り回数	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

⑤ 「PR01」の行を選択する。

⑥ [F4 サイクル起] を押す。

⑦ システムメッセージを確認する。

⑧ [OK] を押す。
「PR01」がサイクル起動します。

レッスン12 I/O の観測と操作を行う

サイクル起動したプログラム「PR01」は、部品供給確認のために、「ioParts」というマクロ名の付いたI/Oの入力待ちになります。

ここでは、プログラムの試験のために、DIOマネージャでI/Oの状態を観測、操作し、プログラムを先に進めてみましょう。

注意：「ioParts」等のI/O番号は、「レッスン5 マクロを定義する」を参照してください。

12.1 DIO マネージャの起動し、通信を接続する

DIOマネージャを起動して、ロボットコントローラ間でデータを常時やり取りするよう、通信を接続します。

ステップ1 DIOマネージャを起動する



① [DENSO System Manager]ウィンドウの[I/O]ボタンをクリックする。
[DIO Manager]ウィンドウが表示されます。

ステップ2 ロボットコントローラと接続し、連続モニタを開始する



② [DIO Manager]ウィンドウの[接続]ボタンをクリックする。
[接続]ボタンが、押し込まれたように表示されます。

③ [連続モニタ]ボタンをクリックする。
DIOマネージャとロボットコントローラの間でのデータ通信が常時行なわれるようになります。
[連続モニタ]ボタンが、押し込まれたように表示されます。

12.2 I/O を観測する

DI0マネージャでI/Oの状態を観測しましょう。

DI0マネージャでのI/Oの表示は、テーブル型、オシロスコープ型、パネル型の3種類の表示形態が選べます。ここでは、テーブル型の表示を使って、I/034番の「ioParts」を調べてみます。

テーブル型のI/Oの状態表示は、[DI0 Manager] ウィンドウを開いたときに最初に現れるリストのような表示形式です。テーブル型以外の表示になっている場合は、[変数スコープ] のリストからテーブル型を選択してください。

ステップ1

I/Oを表示する



- ① [DI0 Manager] ウィンドウをスクロールさせて、34番のI/Oを表示する。
(スクロールは、ウィンドウ右側のスクロールバーを動かして行なう。)

ステップ2

モニタリングを設定する



- ② [DI0 Manager] ウィンドウの[連続モニタ] ボタンをクリックして、連続モニタをOFFにする。(連続モニタ中は設定内容を変更できないため。)

- ③ [DI0 Manager] ウィンドウに表示されている34番のI/Oの[モニタSW]の欄をダブルクリックする。
[モニタSW]の欄が「OFF」から「ON」に変わり、34番のI/Oをモニタできるようになります。

- ④ [連続モニタ] ボタンをクリックして、連続モニタをONにする。
34番のI/Oの[状態]の欄が、I/Oの変化に応じて、「ON」または「OFF」と表示されます。
ここでは、I/034番の「ioParts」は「OFF」と表示されている。

12.3 I/O の擬似 SW を ON/OFF する

DIOマネージャでは、I/Oの状態を擬似的にON/OFFすることができます。

プログラムPRO1「ピック&プレース」は、I/034番の「ioParts」がONになるのを待っていますが、このままではプログラムステップが次に進みません。そこで、I/Oの状態を擬似的にON/OFFすることによって、動作試験を行なえるようにします。

ステップ 1

モニタリングを停止する



番号	状態	出力	用途	マクロ名	モニタSW	擬似SW
34	OFF	汎用入力	部品供給信号	ioParts	ON	ON
35	OFF	汎用入力	動作完了信号の同期信	ioCompleteAck	OFF	ON
36	OFF	汎用入力	RCQエラーの確認	ioErrQRAck	OFF	ON
37	OFF	汎用入力		LIN4	OFF	ON
38	OFF	汎用入力		LIN5	OFF	ON
39	OFF	汎用入力		LIN6	OFF	ON
40	OFF	汎用入力		LIN7	OFF	ON
41	OFF	汎用入力		LIN8	OFF	ON
42	OFF	汎用入力		LIN9	OFF	ON
43	OFF	汎用入力		LIN10	OFF	ON

[DIO Manager]ウィンドウの34番のI/Oの[モニタSW]の欄が「ON」になっていることを確認する。

[DIO Manager]ウィンドウの[連続モニタ]を「OFF」にする。

ステップ 2

擬似SWを設定する



番号	状態	出力	用途	マクロ名	モニタSW	擬似SW
34	OFF	汎用入力	部品供給信号	ioParts	ON	ON
35	OFF	汎用入力	動作完了信号の同期信	ioCompleteAck	OFF	ON
36	OFF	汎用入力	RCQエラーの確認	ioErrQRAck	OFF	ON
37	OFF	汎用入力		LIN4	OFF	ON
38	OFF	汎用入力		LIN5	OFF	ON
39	OFF	汎用入力		LIN6	OFF	ON
40	OFF	汎用入力		LIN7	OFF	ON
41	OFF	汎用入力		LIN8	OFF	ON
42	OFF	汎用入力		LIN9	OFF	ON
43	OFF	汎用入力		LIN10	OFF	ON

[DIO Manager]ウィンドウの34番のI/Oの[擬似SW]の欄をダブルクリックし、ONにする。

擬似I/Oの有効化
[DIO Manager]ウィンドウの[擬似入出力]ボタンをクリックする。
[擬似入出力]ボタンが押し込まれたように表示されます。

[DIO Manager]ウィンドウの34番のI/Oの[状態]の欄をダブルクリックし、「ON」にする。

I/034番の「ioParts」は、プログラムから「ON」状態と認識され、プログラムは次のステップへと進み、問題がなければプログラムを最後まで実行した後に停止します。

レッスン13 変数の観測と操作を行う

変数マネージャで、変数を調べることができます。

ここでは、動作カウントに使われている変数、マクロ名「iPartsId」のI[0]を調べてみましょう。

注意：「iPartsId」等の変数0番号は、「レッスン5 マクロを定義する」を参照してください。

13.1 変数マネージャを起動し、通信を接続する

変数マネージャを起動して、ロボットコントローラとの間でデータを常時やり取りするよう、通信を接続します。

ステップ1 変数マネージャを起動する



① [DENSO System Manager] ウィンドウの[変数] ボタンをクリックする。

ステップ2



[Variable Manager] ウィンドウが表示されます。

② [接続] ボタンをクリックして、ロボットコントローラと接続する。
[接続] ボタンが、押し込まれたように表示されます。

③ [連続モニタ] ボタンをクリックし、連続モニタを開始する。
[連続モニタ] ボタンが、押し込まれたように表示されます。

以上で変数マネージャとロボットコントローラの間でのデータ通信が常時行なわれるようになります。

13.2 変数を観測する

ロボットコントローラと接続されたことにより、ロボットコントローラの変数の状態を見ることができます。

ステップ1

変数を表示する



① [Variable Manager] ウィンドウをスクロールさせて、変数I[0]を表示する。
スクロールは、ウィンドウ右側のスクロールバーを動かして行なう。

変数I[0]の [値] の欄が、変数の変化に応じて表示されます。

レッスン14 連続起動試験を行う

前レッスンまでで、「PR01」の動作をサイクル運転で確認できました。

連続起動で運転すると、同じ動作を繰り返します。I/034番の「ioParts」を擬似SWで「ON」のままにしておけば、実際にI/034番の入力がなくても、連続起動運転が可能です。

実際にロボットを動かす前に、マシンロックしたままで、アームマネージャのロボット画像を見ながら連続運転の試験をしてみましょう。

I/034番の「ioParts」は、前レッスンから引き続き「ON」になったままとします。

14.1 連続起動する

変数マネージャを起動してロボットコントローラとの間で、データを常時やり取りするよう、通信を接続します。

ステップ1 自動モードでプログラム一覧ウィンドウを表示する

① モード切替スイッチを[AUTO]に合わせる。

② 画面左上のモードアイコンがに切り替わったことを確認する。

③ 基本画面で[F1 プログラム]を押し、[プログラム一覧]ウィンドウを表示する。

ステップ2 プログラムを選択、起動する

④ 「PR01」の行を選択する。

プログラム名	モード	実行回数	実行時間	残数
PR01	停止中	2	0.00	128
PR02	停止中	2	0.00	128
PR03	停止中	2	0.00	128

⑤ [F4 起動] を押す。

ステップ3

連続起動する



「1サイクル起動」と「連続起動」の
選択画面が表示されます。

⑤ [連続起動]を選択し、[OK]を押す。
「PRO1」が起動します。

14.2 I/Oを連続観測する

連続起動運転中に、DIOマネージャでI/Oの状態を観測してみましょう。

DIOマネージャでのI/Oの表示は、テーブル型、オシロスコープ型、パネル型の3種類の表示形態が選べます。

[連続モニタ]ボタンが押されていると、[モニタSW]欄が「ON」になっているI/Oの状態を、連続してリアルタイムで表示します。

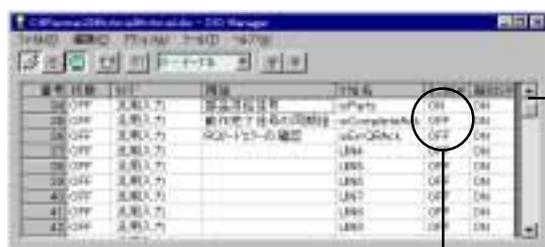
ここでは、プログラムで使用しているI/Oを連続モニタで見てください。

「レッスン13 変数の観測と操作」で開いたDIOマネージャが開いたままのはずですが、もし閉じていたら、システムマネージャの[変数]ボタンをクリックして起動してください。

テーブル型

テーブル型のI/Oの状態表示は、[DIO Manager] ウィンドウを開いたときに最初に現れるリストのような表示形式です。

もしテーブル型以外の表示になっていたら、[表示切り替え]のリストからテーブル型を選択してください。



① [DIO Manager] ウィンドウをスクロールさせて、34番～36番のI/Oを表示する。
(スクロールは、ウィンドウ右側のスクロールバーを動かして行なう。)

② [DIO Manager] ウィンドウに表示されている34番～36番のI/Oの[モニタSW]の欄をダブルクリックし、「ON」にする。
[状態]の欄が、I/Oの変化に応じて「ON」または「OFF」と表示されます。

オシロスコープ型

[DIO Manager]ウィンドウのテーブル型表示で、[モニタSW]を「ON」にしたI/Oの状態を、オシロスコープのように視覚的に観察することができます。

前レッスンでの操作に引き続いて、次の操作を行なってみてください。



① [表示切り替え]リストの  ボタンをクリックし、表示されるリストから、[1-オシロスコープ]を選択する。
34~36番のI/Oの状態が、オシロスコープのように表示されます。

② オシロスコープ型の表示では、I/Oの状態変化を時間の経過と合わせて調べるのに適しています。
各I/Oが、プログラムに従って、ON/OFFするのを確認する。

パネル型

DIOマネージャでは、I/Oの状態をパネル型で表示することもできます。

テーブル型表示で [モニタSW] が「ON」になっているI/Oを表示します。



① [表示切り替え]リストの  ボタンをクリックし、表示されるリストから、[2-パネル]を選択する。
34~36番のI/Oの状態が、オシロスコープのように表示されます。

② パネル型の表示では、多数のI/Oの状態変化を同時に観察するのに適しています。
各I/Oが、プログラムに従ってON/OFFするのを確認する。

14.3 プログラムを停止させる

ここまでで、「PR01」の連続運転を確認できました。第4部に進む前に連続起動運転を続けているプログラムを停止させましょう。



① [STOP]を押す。
プログラムが停止し、[状態]欄の表示が「コンティ停」になります。

② [SHIFT]を押してから、[F7 プロリセット.]を押す。



③ [プログラム リセット]ウィンドウが表示されます。「このプログラムのみ」を選択し、[OK]を押す。



[状態]欄の表示が「停止中」になります。これで、プログラム運転は停止しました。

第4部 プログラムを使ってロボットを動かす

第4部で学習すること

ここでは、ロボットをプログラムによって操作する方法、主要応用分野であるパレタイジングと関連プログラムの編集などについて学習します。また、ロボットタスクプログラムの作成効率を飛躍的に向上させるPACライブラリの活用方法についても学習します。

▶ レッスン 15 ロボットを実際に動かす	90
15.1 マシンロックを解除する	90
15.2 速度・加速度を設定する	91
15.3 モータの電源を入れる	92
15.4 ティーチング	93
15.5 プログラムを起動する	96
15.6 速度を変更する	97
15.7 ロボットの連続運転を停止する	98
■ プログラムを「コンティニュー停止」し、「コンティニュー起動」させる	98
■ プログラムを完全に停止（「プログラムリセット」）する	100
▶ レッスン 16 パレタイジング	102
16.1 パレタイジングの説明	102
16.2 簡易パレタイジング	110
▶ レッスン 17 ライブラリを活用する	118
17.1 プログラムバンク	119
17.2 プログラムのインポート	121
▶ レッスン 18 作業を終了する	123
18.1 WINCAPS II を終了し、パソコンの電源を切る	123
18.2 ロボットコントローラの電源を切る	123

レッスン15 ロボットを実際に動かす

第3部まででプログラムの動作は確認できました。レッスン15では実際にロボットを動かしてみましよう。

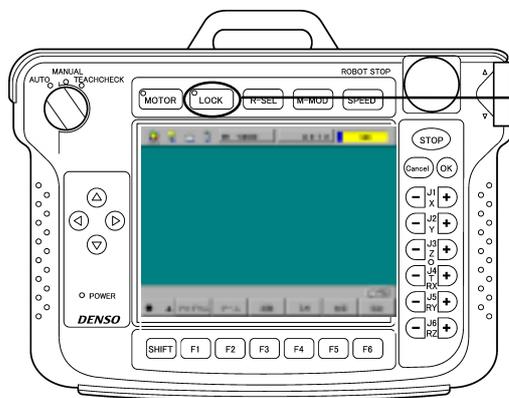
安全を良く確認して、始めは速度設定を低めに設定し、ゆっくりと動かしながら動作を確認します。

第3部までは、I/Oも擬似的な入力で行った動作試験をしましたが、ここからは実際のハードウェアを接続して検証してみてください。

ハードウェアの設定に関しては、設置・保守ガイドを参照してください。

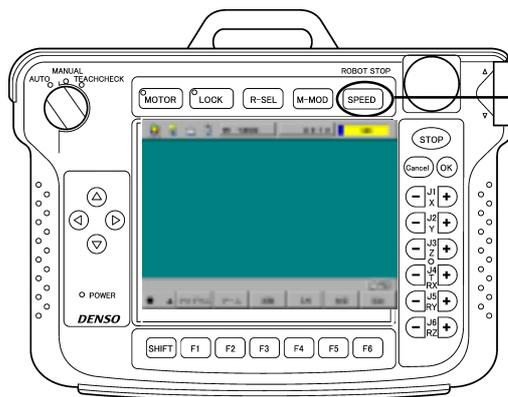
15.1 マシンロックを解除する

ステップ1



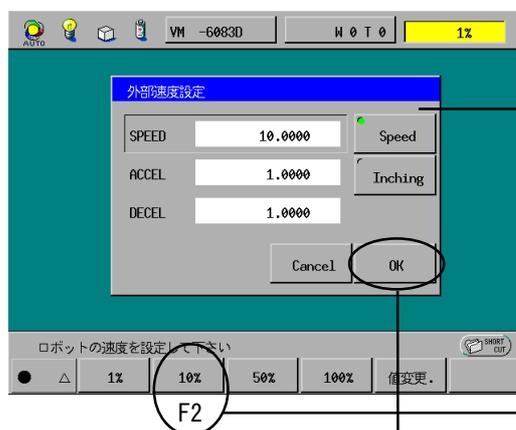
15.2 速度・加速度を設定する

ステップ1



① [SPEED] を押す。

ステップ2



[外部速度設定]ウィンドウが表示されます。

② [F2 10%] を押す。

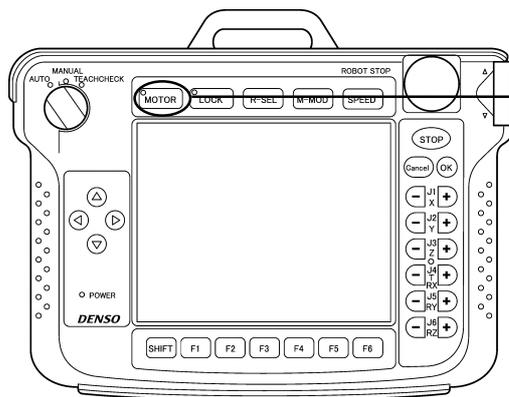
[SPEED]、[ACCEL]、[DECEL]の各欄に、数値が設定されます。

③ [OK] を押す。

速度と加速度が設定されました。

15.3 モータの電源を入れる

ステップ 1



- ① [MOTOR] を押す。
- ② [MOTOR] のランプが点灯したことを確認する。

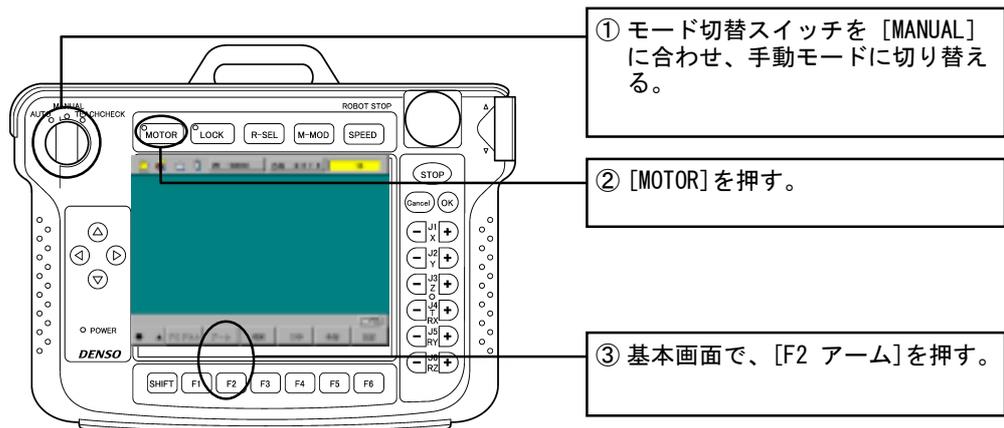
モータの電源が入りました。

15.4 ティーチング

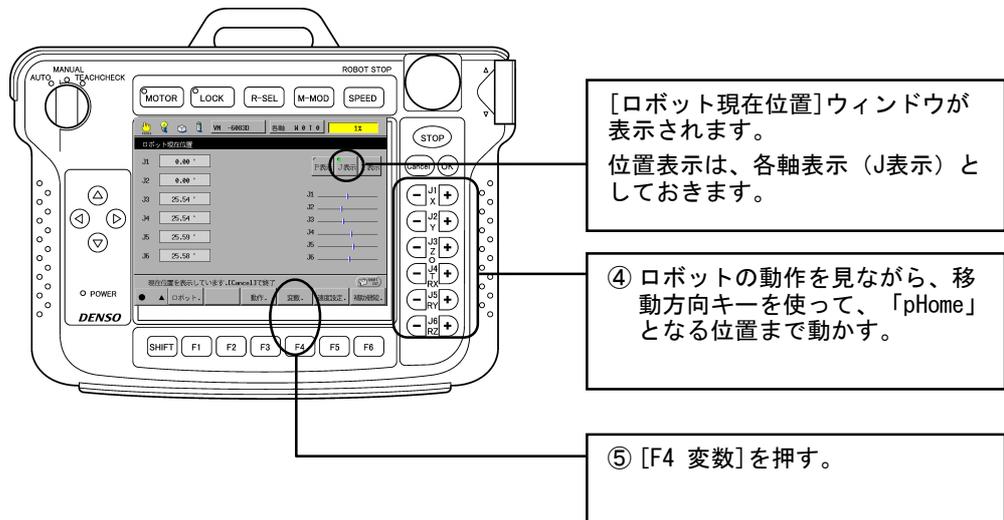
位置変数の値は、「レッスン10 位置変数を入力する」で済ませてありますが、アームマネージャ画面での位置決めはおおまかなものです。ここでもう一度、実機を使って位置決めをして、位置変数に正確な値を取り込んでみましょう。

手順は「レッスン10 位置変数を入力する」とほぼ同じですが、ここでは目標の数字に合う位置を探すのではなく、実際の位置にロボットの姿勢を確認しながら、合わせます。

ステップ1 手動モードにして、ロボット現在位置を表示させる

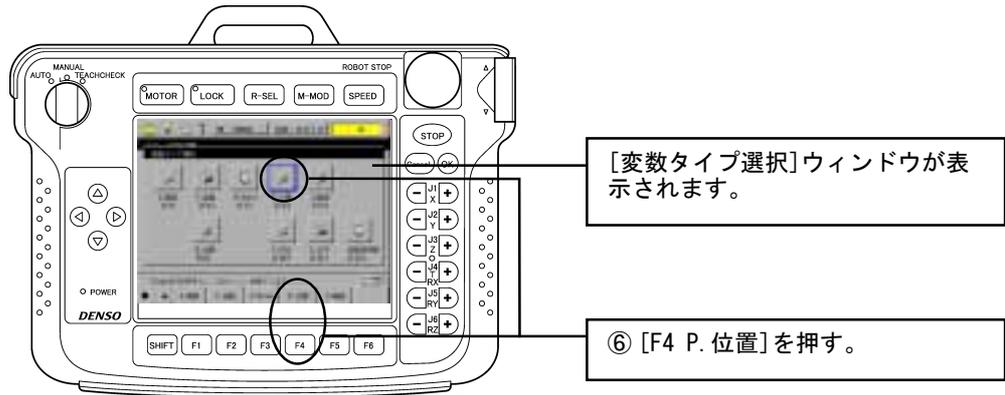


ステップ2 ロボットを移動させる



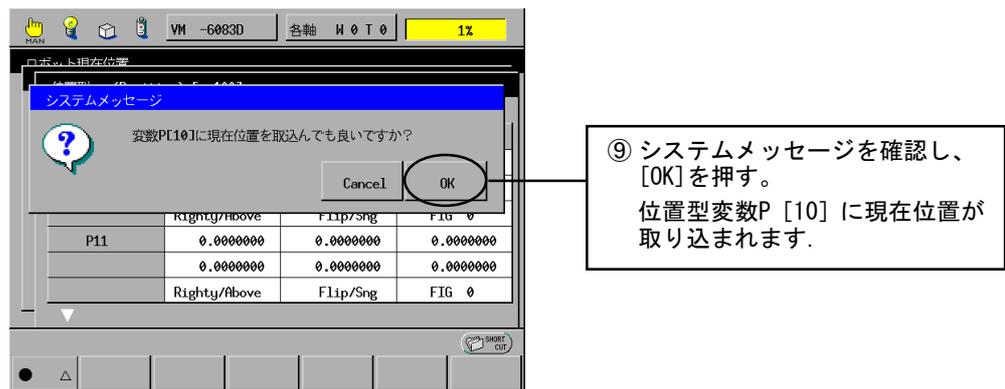
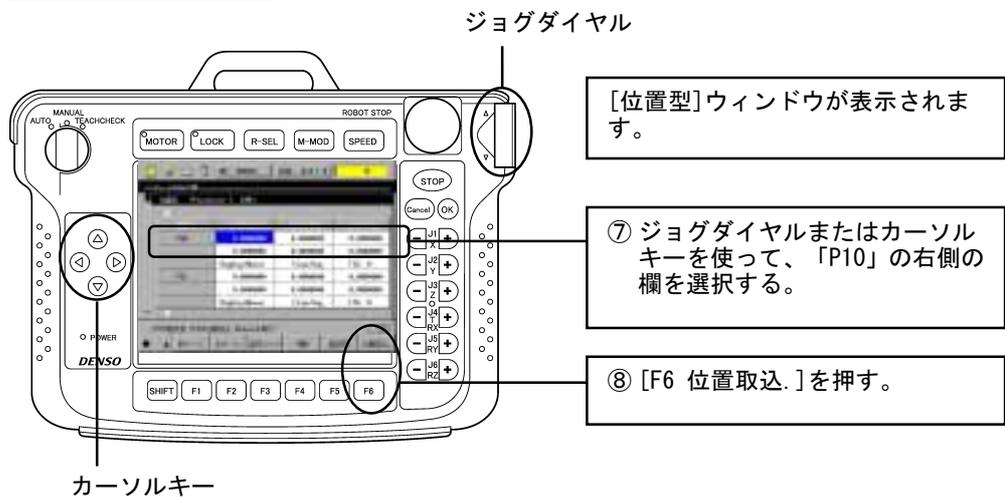
ステップ3

位置型変数を表示させる



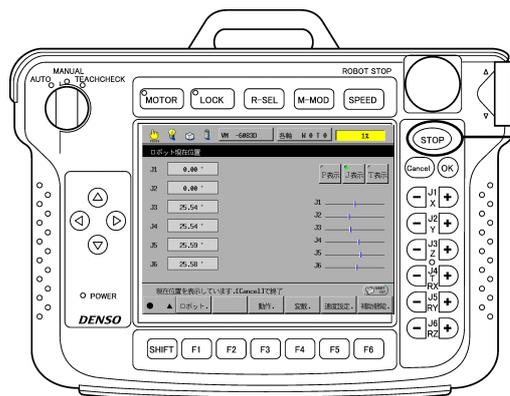
ステップ4

現在位置を取り込む



ステップ5

他の位置を取り込む



⑩ [Cancel]を2回押す。
[ロボット現在位置]ウィンドウの表示に戻ります。

⑪ ステップ2からステップ4の操作を繰り返して、他の3つの変数「pPick」、「pPlace1」、「pPlace2」についても、それぞれP [11]、P [12]、P [13] に位置データを取り込む。

ステップ6

基本画面に戻る



⑫ 全部の位置取り込みが終わったら、[Cancel]を押して、基本画面に戻る。

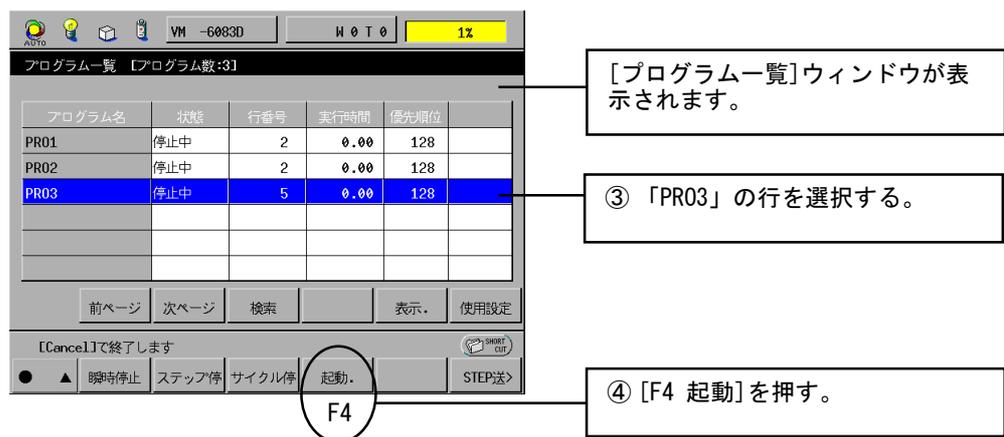
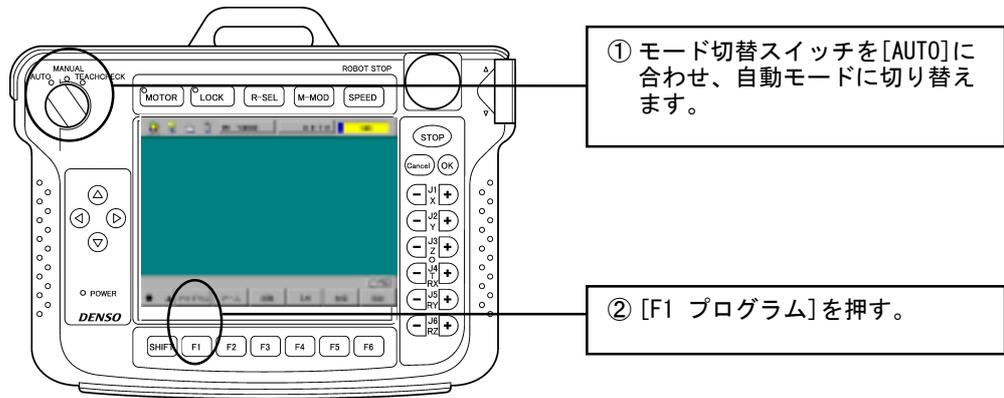
15.5 プログラムを起動する

ロボットをプログラムによって、実際に動かしてみましょう

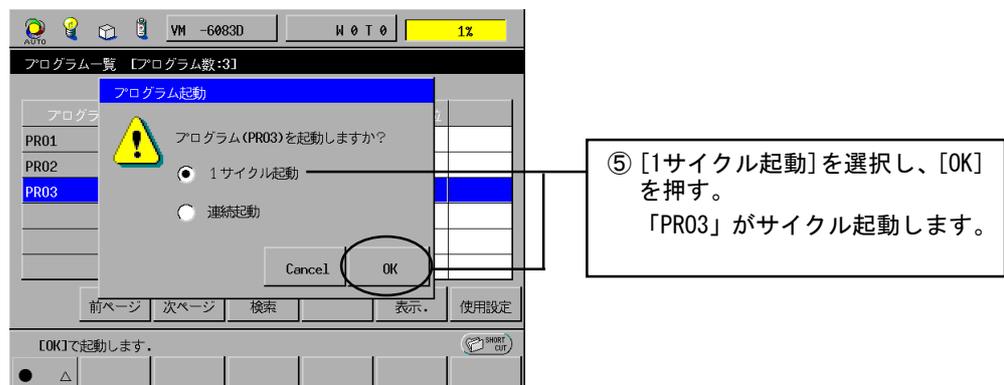
注意：ロボット動作中は、いつでもロボット停止ボタンを操作できるようにしてください。

ステップ 1

自動モードにして、プログラム一覧ウィンドウを表示する



ステップ 2



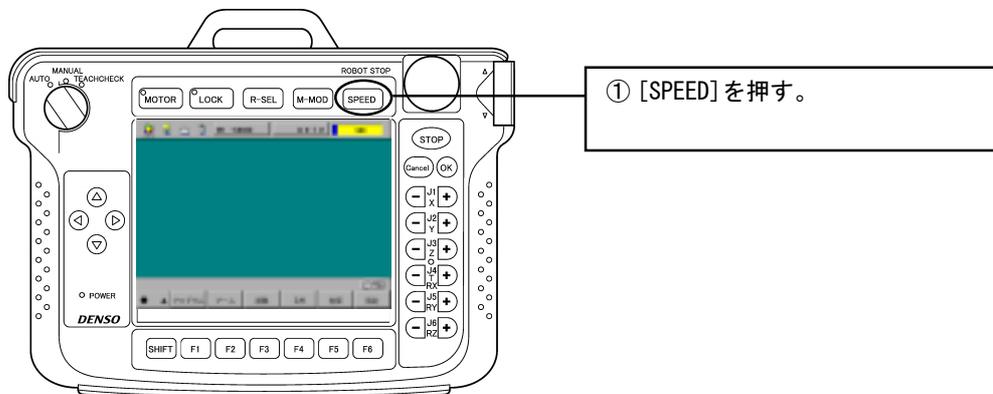
ステップ 3

ここで、[1サイクル起動]の代わりに[連続起動]を選べば、連続起動します。サイクル起動運転の結果が良ければ、連続起動も試してみてください。

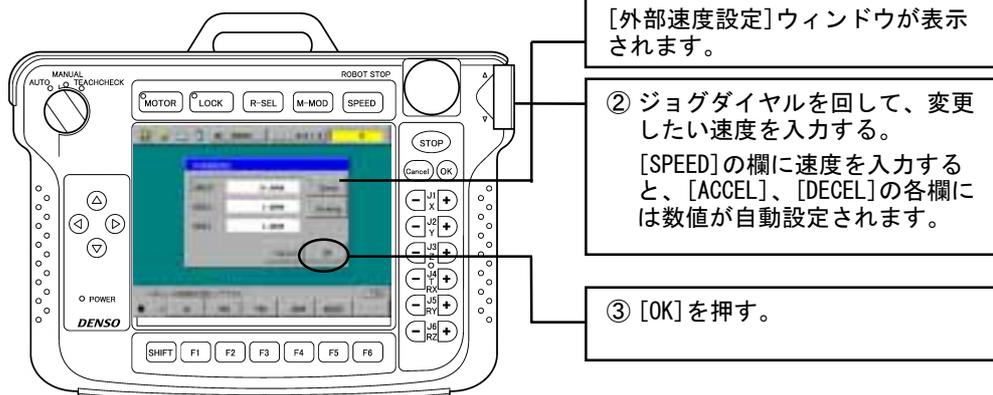
15.6 速度を変更する

サイクル起動運転と連続起動運転の結果が良ければ、速度設定を変更して、徐々に速く動かしてみましよう。

ステップ1



ステップ2



速度、加速度および減速度が設定されました。

15.7 ロボットの連続運転を停止する

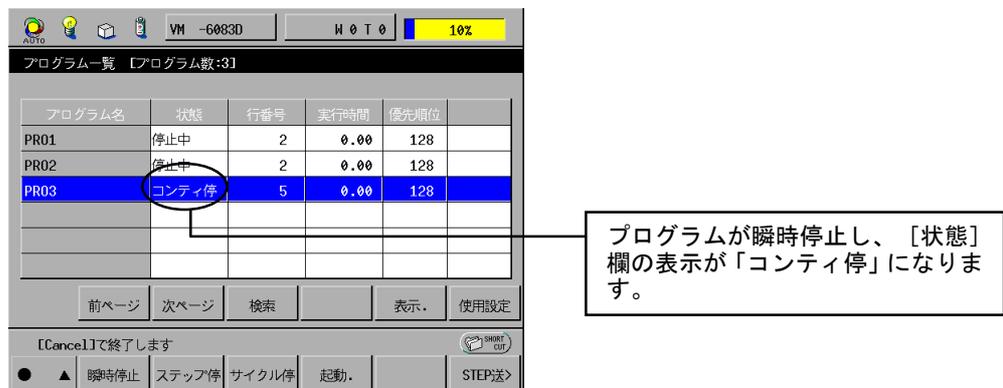
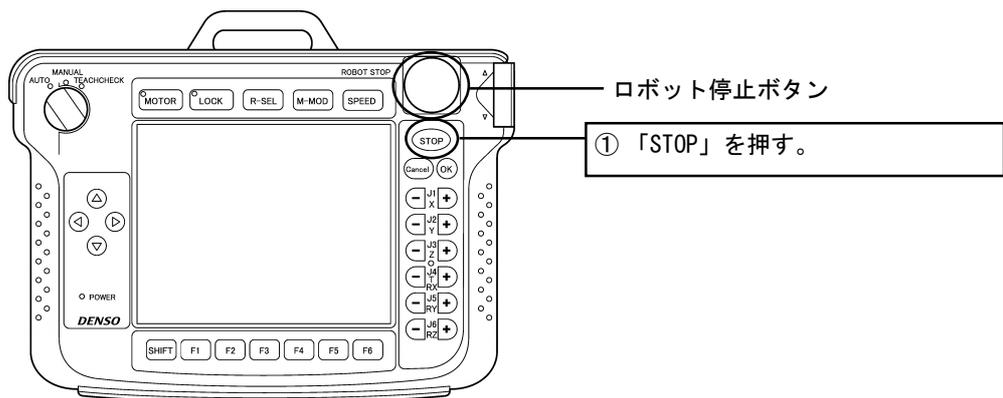
ロボットの連続動作を[STOP]を押して停止させると、「コンティニュー停止」状態となります。この状態のプログラムは停止した位置から再起動させることができます。

そのプログラムを完全に停止させるには、「プログラムリセット」を行ってください。

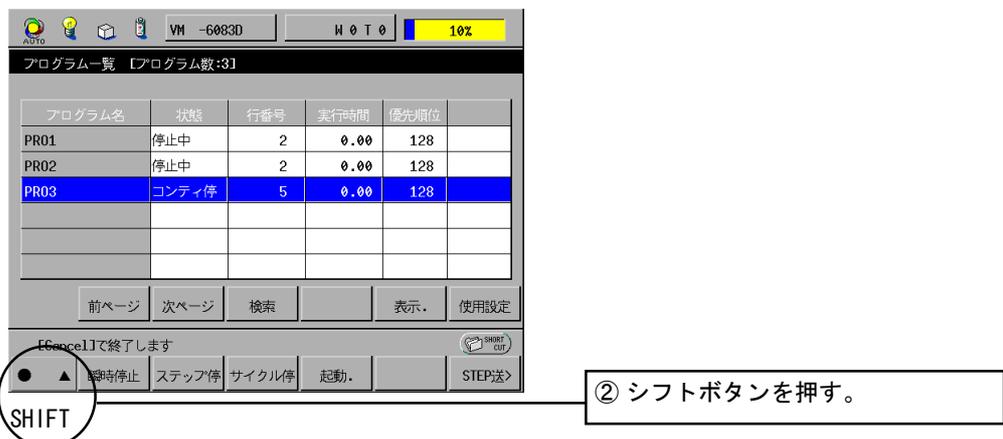
注意：緊急の場合には、ロボット停止ボタンを押してください。

■ プログラムを「コンティニュー停止」し、「コンティニュー起動」させる

ステップ1 「コンティニュー停止」させる



ステップ2 「コンティニュー起動」させる





③ 「F10 コンティ起動。」を押す。



④ 「OK」を押す。

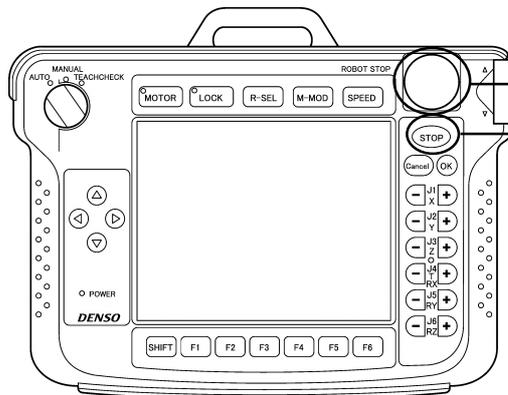


プログラムが実行開始し、状態の表示が「実行中」と表示されます。

これでプログラム実行再開の手順は完了です。

■ プログラムを完全に停止（「プログラムリセット」）する

ステップ1 「コンティニュー停止」させる



ロボット停止ボタン

① 「STOP」を押す。

The screenshot shows the software interface with a table of program statuses. The 'PR03' row is highlighted in blue, and its status is 'コンティ停' (Continuous Stop), which is circled in red. A callout line points from this status to a text box.

プログラム名	状態	行番号	実行時間	優先順位
PR01	停止中	2	0.00	128
PR02	停止中	2	0.00	128
PR03	コンティ停	5	0.00	128

プログラムが瞬時停止し、[状態]欄の表示が「コンティ停」になります。

ステップ2 「プログラムリセット」する

The screenshot is identical to the previous one, but now a callout line points from the 'SHIFT' button (located at the bottom left of the interface) to a text box. The 'PR03' row remains highlighted.

SHIFT

② シフトボタンを押す。





③ [F7 フォリセット.]を押す。



④ 停止させるプログラムを選択する。
「このプログラムのみ」を選択すると、選択されているプログラムのみ停止します。
「すべてのプログラム」を選択すると、すべてのプログラムが停止します。

⑤ 「OK」を押す。



プログラムが停止し、状態の表示が「停止中」と表示されます。

これで、プログラムの停止が完了しました。

レッスン16 パレタイジング

16.1 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。

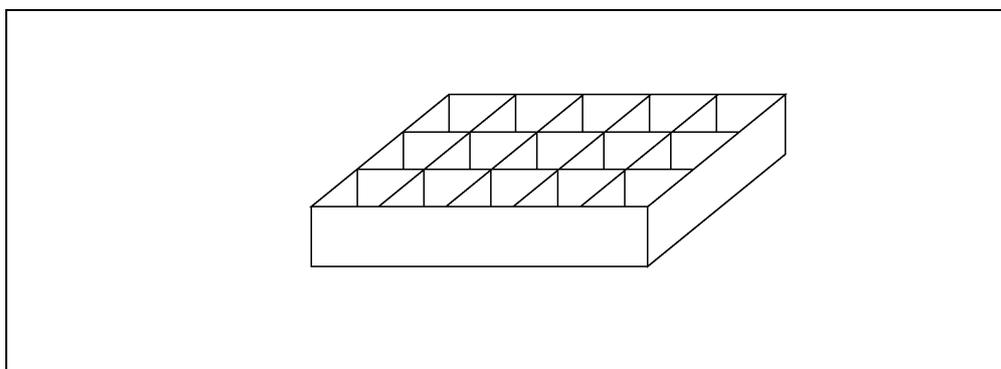


図16-1 仕切りのあるパレット

パレタイジングテンプレート2 「PR01」

実際のパレタイジングプログラムは、そのアプリケーションでの特殊な状況によってさまざまですが、ライブラリを使ってプログラムを組み立てる場合の標準的な手順を、タイトル「パレタイジングテンプレート2」のプログラム名「PR01」に用意してあります。

この「パレタイジングテンプレート2」を雛型にして、各アプリケーションの必要事項を書き足し、あるいは不要な部分を削除してプログラム開発に役立ててください。

このプログラム名「PR01」の「パレタイジングテンプレート2」の事例を示します。

この例では、以下のような設備を想定しています。

- ・ パレタイジングポイントはP50～P55とします。
- ・ 作成プログラムは、定位置P50に移動し、パレタイジングプログラム0番を実行して、組み付け位置P51に移動し、アンチャック動作を行ない、段終了を確認し、終了信号が出力されていれば必要に応じてパレット入れ替え動作を行ない、ワークがなくなると終了します。

プログラム名「PR01」の「パレタイジングテンプレート」と、プログラム名「PR02」の「パレタイジング初期化テンプレート1」は、WINCAPS IIで「プロジェクトの新規作成」を行なう際に、「プロジェクトの新規作成」ダイアログの「設備タイプ」の項目で「パレタイジング」を選ぶと、自動的に登録されます。

```
'!TITLE “パレタイジングテンプレート2”
'!AUTHOR “株式会社 デンソー”
#DEFINE pltIndex    0                (パレタイジングプログラム番号、
                                      0～30 までの任意の数値を選択可)
#DEFINE ChuckNG     40              (取り出し検査 IO の番号、任意の数値を選択可)

PROGRAM PRO3                          ' 適当な名前に変更して下さい。
  TAKEARM                              ' アームセマフォの取得
  MOVE P, P50                          ' 定位置 P50 へ移動。
                                      ' パレタイジングされた P50 に移動。
  IF IO[ChuckNG] = ON THEN             ' 前回の取り出し検査。
    CALL pltDecCnt(pltIndex)           ' 総カウンタを-1 する。
  END IF
  CALL pltMove(pltIndex)                ' パレタイジング番号 0 実行。
  MOVE P, P51                          ' 組み付け位置 P51 へ移動。
                                      ' パレタイジングされた P51 に移動。

'<--- アンチャック動作等を挿入 --->
CALL pltGetPLT1END(pltIndex,0)         ' I[0]に 1 段終了信号取得
                                      (2 個目の引数 “0” は次の行で判定される
                                      “I” の行列番号)
  IF I[0] THEN                          ' ON のとき(つまり <=>0 のとき)
    '<--- パレットチェンジ動作等を挿入 --->
    CALL pltResetPLT1END(pltIndex)     ' 1 段終了信号クリア
  END IF
  GIVEARM                              ' アームセマフォの開放
END
```

図16-2 プログラム名「PR01」、 「パレタイジングテンプレート2」

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、図16-3、図16-4、図16-5、表16-1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。

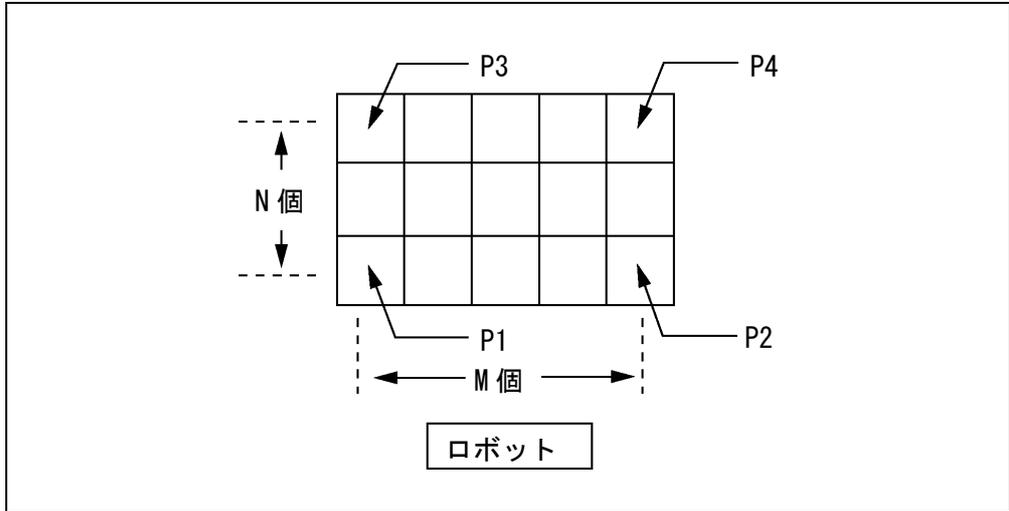


図16-3 パレットの上視図

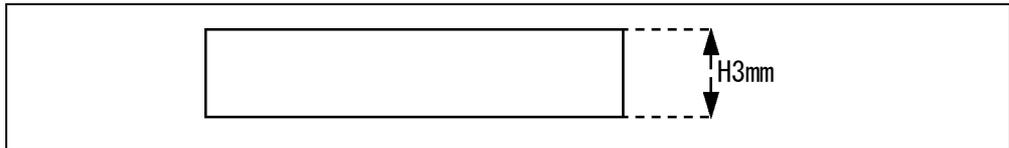


図16-4 パレットの横視図

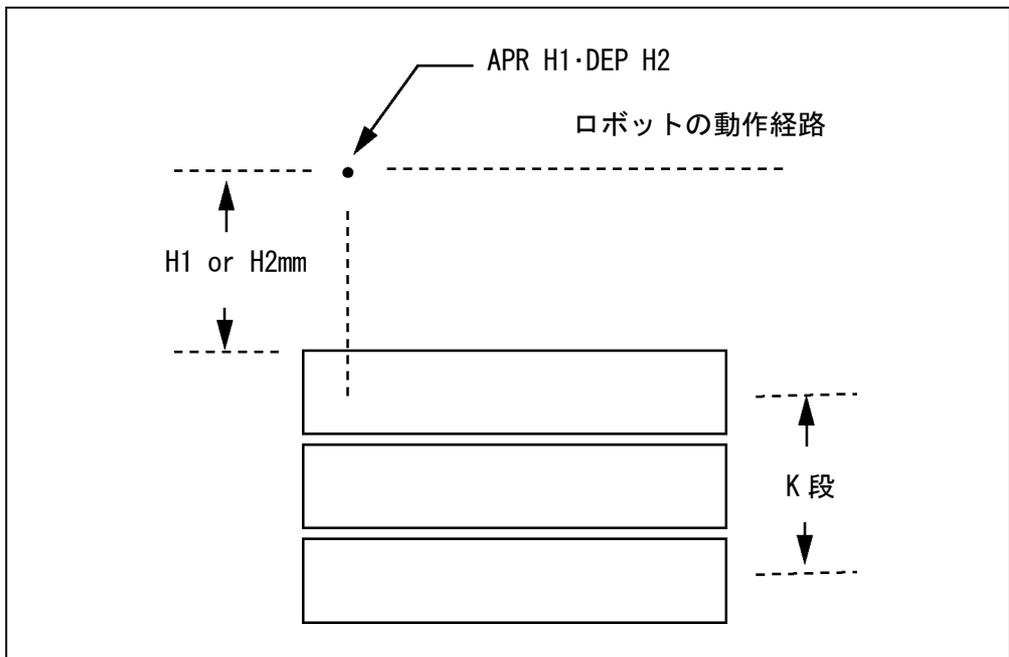


図16-5 パレットの段積図

表16-1 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)
	ただし、H1 と H2 は、次の条件を満たさなければなりません。 $H1 > \{H3 \times (K-1)\} + 5$ $H2 > \{H3 \times (K-1)\} + 5$		
P1 P2 P3 P4	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。		

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。

注意：H1とH2は、次の条件を満たさなければなりません。

$$H1 > \{H3 \times (K-1)\} + 5$$

$$H2 > \{H3 \times (K-1)\} + 5$$

この条件を満たさない場合には初期化時にエラーが表示されます。ロボットがパレットに衝突しないように、この条件があります。パレットの段数が最も多いときよりもさらに5mm高い点をアプローチ、デパートの位置にするためです。

段積みが増減しても、図16-6に示すように、アプローチ点、デパート点は同じです。

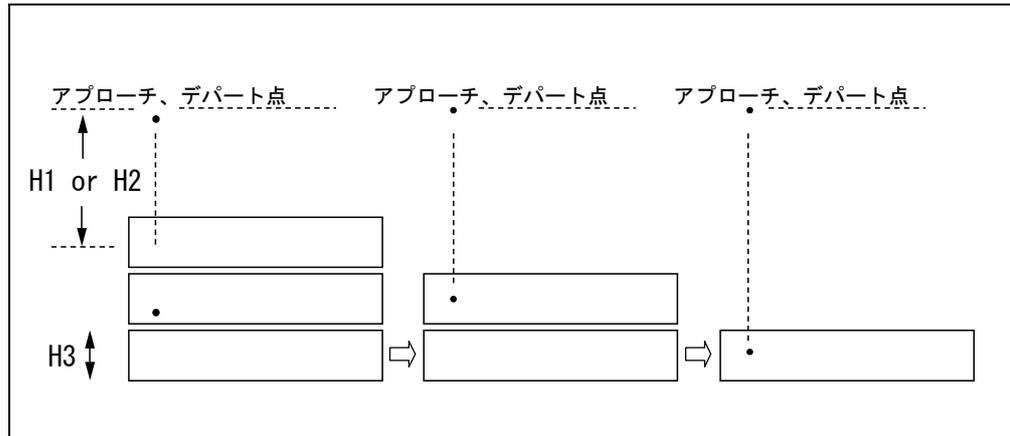


図16-6 段積みの変化とアプローチ点、デパート点

P1、P2、P3、P4 四隅の点

パレットの四隅の部品位置を示します。各点と実行順序の関係は、下図のとおりです。

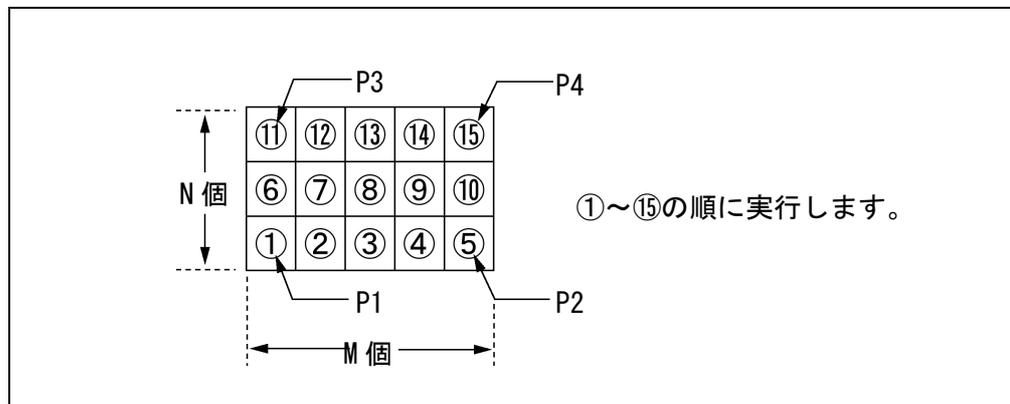


図16-7 パレタイジングの順番

パラメータ値の設定

パレタイジングにおける縦横の分割数や段積数など、パラメータ値の設定は、ライブラリプログラムの「pltInitialize」を呼び出すことによって行ないます。

「システムプロジェクトの新規作成」を行なうときに、[プロジェクトの新規作成] ダイアログボックスの[設備タイプ]の欄で、[1-パレタイジング]を選択すると、ライブラリの中に「パレタイジング初期化テンプレート1」というタイトルのプログラム「PRO2」が自動的に登録されます。このプログラムは、「pltInitialize」を呼び出すようになっていいますから、CALL文の引数の値を、適当な値に変更して使います。

```
"!TITLE "パレタイジング 初期化テンプレート 1"  
"!AUTHOR "株式会社デンソー"  
  
PROGRAM PRO2 "適当な名前に変更して下さい"  
  CALL pltInitialize(0,4,3,1,50,50,50,52,53,54,55) 'パレタイジング番号 0 の初期化  
END
```

図16-8 ライブラリプログラム「パレタイジング初期化テンプレート1」

CALL pltInitialize(0, 3, 5, 3, 50, 50, 10, 52, 53, 54, 55)のパラメータの説明

- 1 番目... パレタイジングプログラム番号 (0)
 - 2 番目... 横分割数 (3)
 - 3 番目... 縦分割数 (5)
 - 4 番目... 段積数 (3)
 - 5 番目... アプローチ長 (50mm)
 - 6 番目... デパート長 (50mm)
 - 7 番目... パレットの高さ (10mm)
 - 8 番目... P1 位置 (P52)
 - 9 番目... P2 位置 (P53)
 - 10 番目... P3 位置 (P54)
 - 11 番目... P4 位置 (P55)
- } P型変数の番号のみ指定します。

図16-9 「pltInitialize」のパラメータの説明

このパラメータの意味は、WINCAPS IIのPACマネージャの中にあるツールの「コマンドビルダ」でも見ることができます。

パレタイジングカウンタ

パレタイジングでは、動作中にパレットの仕切り数を数え、変数にカウント値を保持します。

カウンタは、横方向 (N)、縦方向 (M)、高さ方向 (K)、トータル (cnt) の4つがあります。これらのカウンタは、パレタイジング動作を制御する中核プログラム、「pltKern1」の中で定義されています。

ライブラリプログラムの「pltMove」は、パレットの位置に対して一つの作業を完了するごとに、トータルカウンタの値を1つ加算し、他のカウンタの値も調整します。

ライブラリプログラムの「pltDecCnt」は、呼び出すことで、トータルカウンタの値を1つ減算し、各カウンタの値を減算調整します。

パレタイジングプログラムは、デフォルト設定では30個まで作成可能です。したがって、パレタイジングカウンタも31セットあります。

カウント規則

パレタイジングカウンタは、「pltMove」が実行されるたびに、トータルカウンタの値を1つ加算し、他のカウンタの値を調整します。これにより、次のパレット位置が保証されます。

トータルカウンタが一つ加算されると、縦方向カウンタ (M) の位置が次へ1つ移動します。縦方向カウンタ (M) の位置が端まで到達して最大値になると、横方向カウンタ (N) の位置が次へ1つ移動し、縦方向カウンタ (M) の位置は設定された最小値になります。横方向カウンタ (N) の位置が端まで到達して最大値になると、高さ方向カウンタ (K) の位置が次へ1つ移動し、縦方向カウンタ (M) と横方向カウンタ (N) は、設定された最小値になります。

パレタイジングプログラムを、途中でプログラム停止して再開すると、カウンタ変数の値はカウントアップされているので次の位置へ動きます。

電源を切っても、パレタイジングカウンタの内容は保持されます。電源再投入後に、パレタイジングカウンタを初期化しなければ、前回終了時の続きからパレタイジング動作を行ないます。

注意：コンパイル後、実行プログラムをロードすると変数の値は初期化されます。

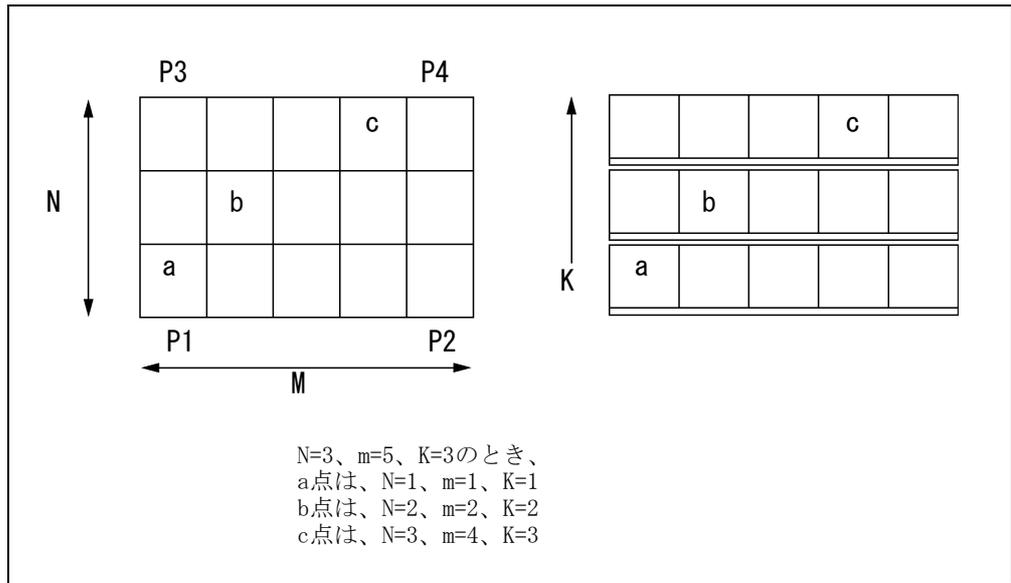


図16-10 パレタイジング位置とカウンタの関係

カウンタの初期化

パレットの入れ替えや、パレットのすべてのます目を使用しない場合に、カウンタを初期化します。

カウンタの初期化では、すべてのパレタイジングカウンタに「1」を代入します。

ライブラリプログラムの「pltResetAll」を使用すると、すべてのパレタイジングカウンタを一度に初期化できます。

たとえば、パレット番号1のすべてのカウンタを初期化するには、次のように記述します。

```
CALL pltResetAll(1)
```

各パレタイジングカウンタを個別に初期化するには、「pltLetN1」、「pltLetM1」、「pltLetK1」、「pltLetCnt」を使います。

たとえば、パレット番号1のNカウンタを初期化するには、次のように記述します。

```
CALL pltLetN1(1, 1)
```

注意：第2引数がカウンタに設定する値です。1以外の値に設定することもできません。

パレタイジングプログラムの終了信号

パレタイジングプログラムは、1段終了および全段終了すると、1段終了フラグおよび全段終了フラグをセットします。

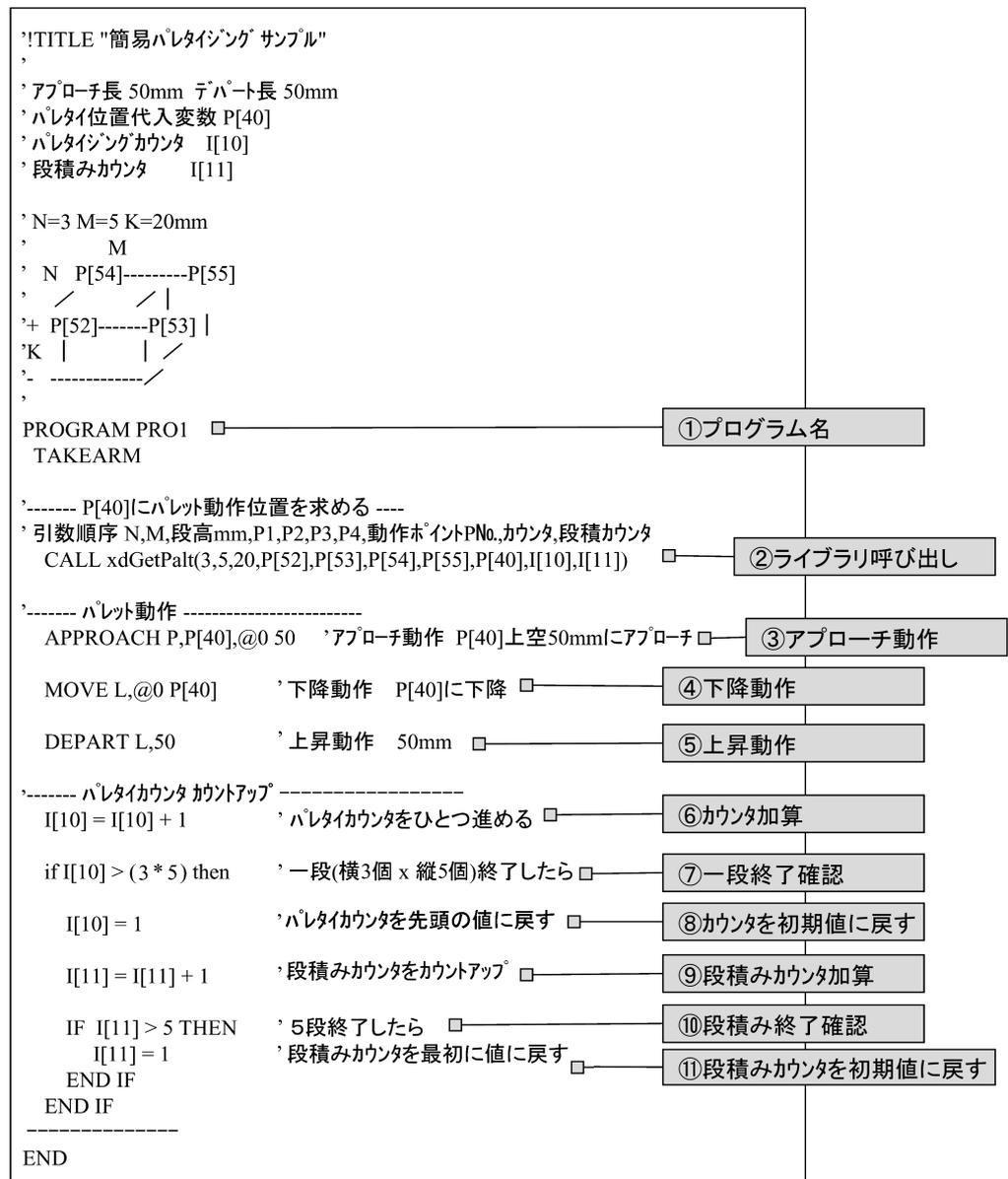
1段終了フラグの状態を取得するには、ライブラリプログラム「pltGetPLT1END」を使います。全段終了フラグの状態を取得するには、ライブラリプログラム「pltGetPLTEND」を使います。

1段終了フラグをリセット（0）するには、ライブラリプログラム「pltResetPLT1END」を使います。全段終了フラグをリセット（0）するには、ライブラリプログラム「pltResetPLTEND」を使います。

16.2 簡易パレタイジング

前項で、パレタイジングプログラムについて説明しましたが、それよりも単純なパレタイジングを行なうには、ここで紹介する簡易パレタイジングライブラリを使用した方が簡単な場合があります。

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

入門編のレッスン16で説明したパレタイジングでは、パレタイジングの実行に先立ち、必ずライブラリpltInitializeを実行する必要がありました。

しかし、この簡易パレタイジングプログラムでは、その必要は無く、PR01を実行するだけでパレタイジング動作が行われます。

パレタイジングカウンタや段積みカウンタは、ライブラリの内部で自動的に加算されていましたが、簡易パレタイジングでは、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(5,3,20,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以上である必要があります。
例では、3列になっています。
- 2 番目... 縦分割数M、1以上である必要があります。
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以下である必要があります。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以上である必要があります。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

```
'----- パレット動作 -----  
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

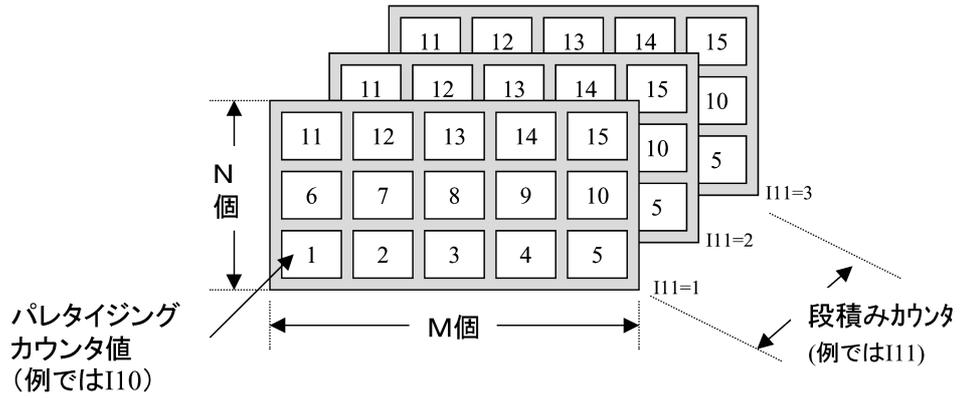
```
'----- パレタイカウンタ カウントアップ -----  
I[10] = I[10] + 1          'パレタイカウンタをひとつ進める  
  
if I[10] > (3 * 5) then    '一段(横3個 x 縦5個)終了したら  
  I[10] = 1                'パレタイカウンタを最初に値に戻す  
  I[11] = I[11] + 1        '段積みカウンタをカウントアップ  
  IF I[11] >= 5 THEN        '5段終了したら  
    I[10] = 1              '段積みカウン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

```

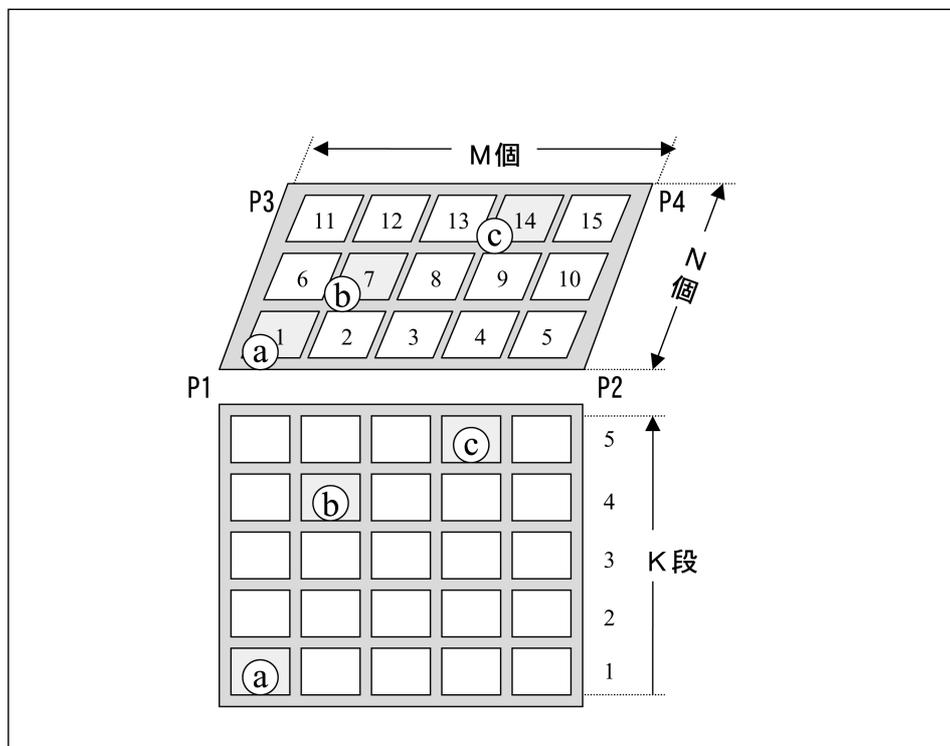
'----- パレタイカウンタ カウントアップ -----
I[10] = I[10] + 1      'パレタイカウンタをひとつ進める

if I[10] > (3 * 5) then
  I[10] = 1            '一段(横3個 x 縦5個)終了したら
                       '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

```

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



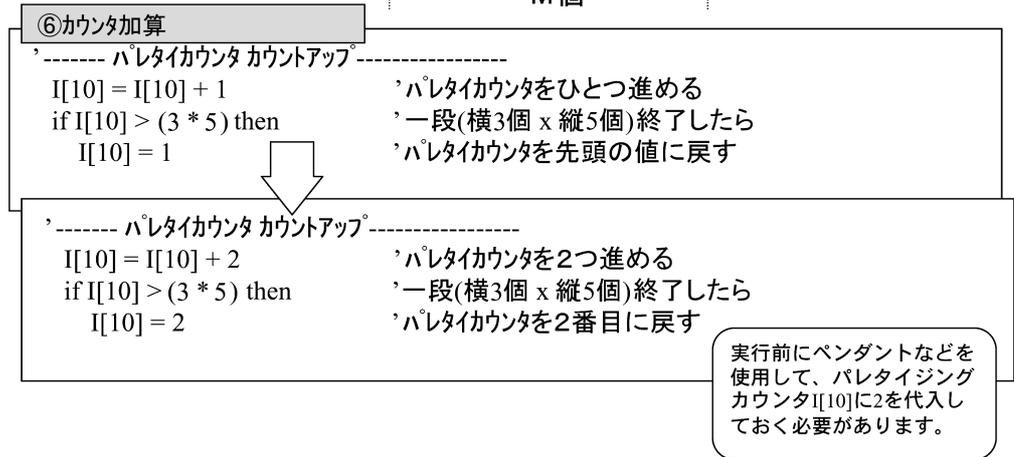
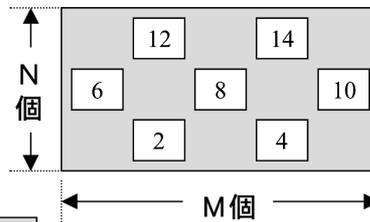
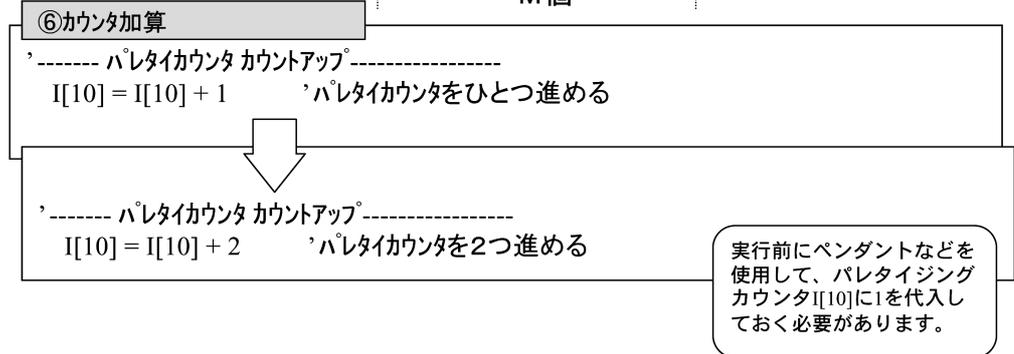
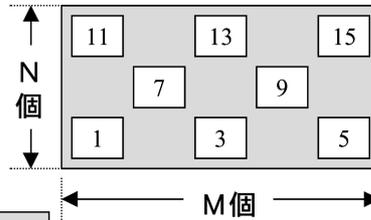
パレットが $N=3$, $M=5$ 、パレタイジングカウンタ I10、段積みカウンタ I11 の場合

- Ⓐ $I10 = 1$, $I11 = 1$
- Ⓑ $I10 = 7$, $I11 = 4$
- Ⓒ $I10 = 14$, $I11 = 5$

簡易パレタイジングプログラムの応用 – 特殊なパレタイジング例 –

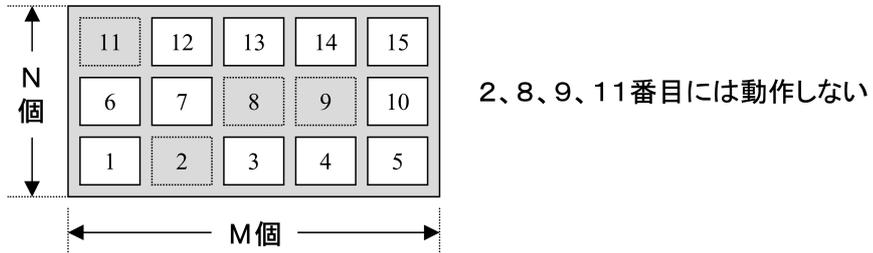
千鳥パレタイジング

パレタイジング動作で、一つ置きに動作する物を特に「千鳥パレタイジング」と呼んでいます。これを実現するのは非常に簡単です。



歯抜けパレタイジング

パレタイジング動作で、任意の場所をスキップする動作を特に「歯抜けパレタイジング」と呼んでいます。



このような複雑に見える動作であっても、結局ライブラリに渡すパレタイジングカウンタの値を変更すれば簡単に実現可能です。

```
PROGRAM PRO1
TAKEARM
```

’----- 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])
```

```
PROGRAM PRO1
```

```
TAKEARM
```

```
SELECT CASE I[10]
```

```
CASE 2      ’パレタイカウンタ I[10]の値が2の時、
  I[10] = 3  ’パレタイカウンタ I[10]を3にする
```

```
CASE 8,9    ’パレタイカウンタ I[10]の値が8,9の時、
  I[10] = 10 ’パレタイカウンタ I[10]を10にする
```

```
CASE 11     ’パレタイカウンタ I[10]の値が11の時、
  I[10] = 12 ’パレタイカウンタ I[10]を12にする
```

```
END SELECT
```

’----- 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])
```

レッスン17 ライブラリを活用する

すでに作成してあるプログラムを、現在作成中のプロジェクトに組み入れることができます。それには、次の3つの方法があります。

- ・ **プログラムバンク :**

プログラムバンクに登録されているプログラムを現在作成中のプロジェクトに追加します。

- ・ **プログラムのインポート :**

他のプログラムプロジェクトで作成してあるプログラムの複製を、現在作成中のプロジェクトフォルダに作成し、登録します。

- ・ **プログラムの追加 :**

他のプログラムプロジェクトで作成してあるプログラムを、現在作成中のプロジェクトにも登録します。一つのプログラムが同時に複数のプロジェクトに登録されることとなります。

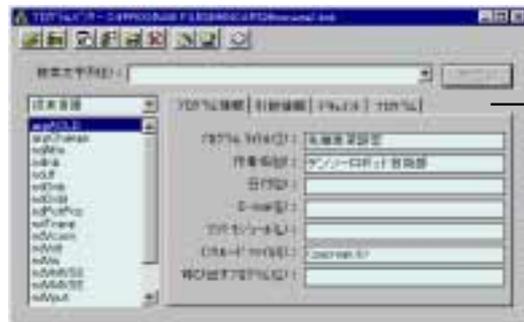
これらの方法を利用すると、今までに作成したプログラムを有効に活用して、新しいプロジェクトの作成効率を高めることができます。

17.1 プログラムバンク

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

プログラムバンクを参照する

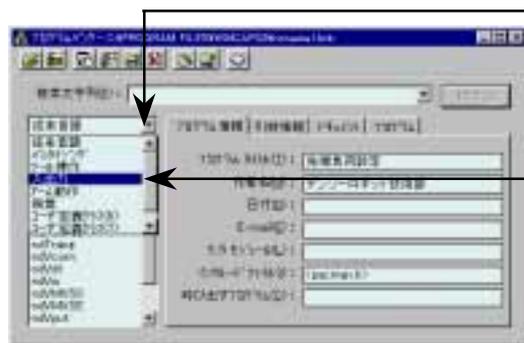


① [PAC Manager] ウィンドウの [ツール] メニューから [プログラムバンク] を選択する。

[プログラムバンク] ウィンドウが表示されます。

ステップ 2

クラスを選択する

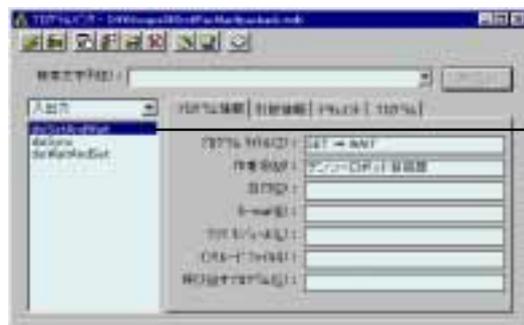


② クラス選択ボックスをクリックする。
クラス選択プルダウンメニューが表示されます。

③ [入出力] を選択する。

ステップ 3

追加したいプログラムを選ぶ



プログラム名リストボックスに、「入出力クラス」に登録されているプログラムのタイトルが表示されます。

④ プログラム名リストボックスに表示されている [dioSetAndWait] を選択する。

ステップ4

プロジェクトに追加する



```

TITLE "SET => WAIT"
AUTHOR "デンソーロボット技術部"
ARG 1, "同期出力信号", 1, "104"
ARG 2, "同期入力信号", 1, "34"

PROGRAM dioSetAndWait(ackIndex%, waitIndex%)
  SET IO(ackIndex) "同期信号ON"
  WAIT IO(waitIndex) = ON "信号ON待ち"
  RESET IO(ackIndex) "同期信号OFF"
END

```

⑤ [プロジェクトに追加]をクリックする。

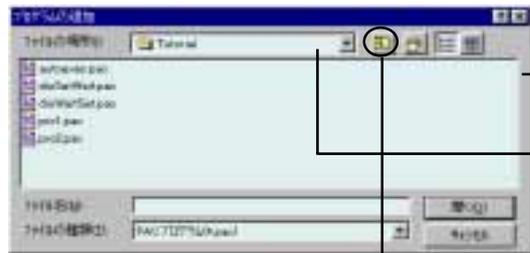
[dioSetAndWait]がプログラムプロジェクトに登録され、編集ウィンドウが表示されます

⑥ [プログラムバンク]ウィンドウのクローズボックスをクリックする。
[プログラムバンク]ウィンドウが閉じます。

17.2 プログラムのインポート

「pro1」、「pro2」の2つのプログラムをインポートしてみましょう。これらのプログラムは、WINCAPS II を標準インストールしてある場合、「C:\ProgramFiles\Wincaps2 \Sample\modelcase\」ディレクトリに用意されています。

ステップ 1



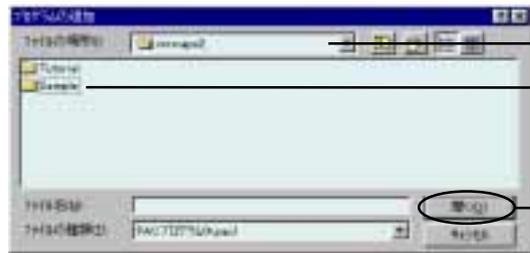
① [PAC Manager] ウィンドウの [プログラム] メニューから [インポート] を選択する。

[プログラムの追加] ダイアログウィンドウが表示されます。

現在のシステムプロジェクトフォルダである「tutorial」フォルダが表示されています

② [1つ上のフォルダへ] ボタン  をクリックする。

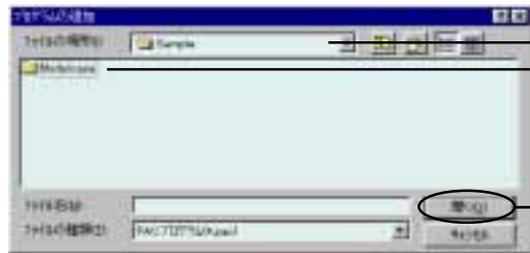
ステップ 2



[ファイルの場所 (I)] の欄の表示が、「wincaps2」フォルダになります。

③ 「Sample」フォルダを選び、[開く (O)] をクリックする。

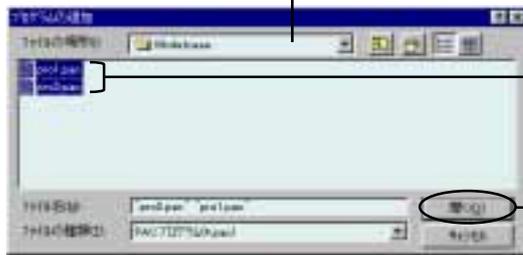
ステップ 3



[ファイルの場所 (I)] の欄の表示が、「Sample」フォルダになります。

④ 「Modelcase」フォルダを選び、[開く (O)] をクリックする。

ステップ4



[ファイルの場所 (I)] の欄の表示が、「Modelcase」フォルダになります。

⑤ 「PR01.pac」、「PR02.pac」の2つのファイルのアイコンを選び、[開く (O)] をクリックする。

ステップ5



[PAC Manager] ウィンドウのリストに、新たに2つのプログラムが追加表示されます。

⑥ [PAC Manager] ウィンドウのリストにある「PR01」をダブルクリックする。

★参考★

「PR01」をクリックして選択してから、[アクション (A)] メニューの [表示 (S)] を選択しても、編集ウィンドウを開くことができます。

ステップ6



「PR01.pac」の編集ウィンドウが表示されます

⑦ 編集ウィンドウの内容に間違いがないことを確認する。

⑧ もう一つのプログラム「PR02.pac」についても、ステップ4とステップ5の操作を繰り返して、内容が正しいか確認する。

⑨ クローズボックスをクリックして、各プログラムの編集ウィンドウを閉じる。

レッスン18 作業を終了する

パソコン教示システムを終了するために、パソコンとロボットコントローラの両方で終了操作をしましょう。

18.1 WINCAPS II を終了し、パソコンの電源を切る

ステップ1



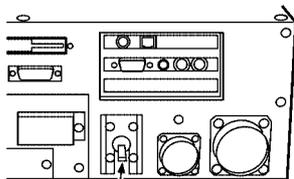
① [DENSO System Manager] ウィンドウの右上にあるクロー징ボックスをクリックする。
WINCAPS II のパソコン側のすべてのソフトウェアが終了します。

ステップ2

必要に応じ、Windows95など使用中のOSを終了し、パソコンの電源を切ってください。

18.2 ロボットコントローラの電源を切る

ステップ1



① ロボットコントローラの電源スイッチを下に倒して、電源を切る。

第5部 デンソーロボットの特徴的機能

第5部で学習すること

ここでは、力制限機能などデンソーロボットが持つ特徴的な機能について学習します。これらの機能は、設備コストの引き下げ、準備作業や実作業効率の改善に効果があります。

▶ レッスン 19 力制限機能	126
19.1 各軸電流制限機能	126
19.2 先端力制限機能	127
▶ レッスン 20 その他の機能	128

レッスン19 力制限機能

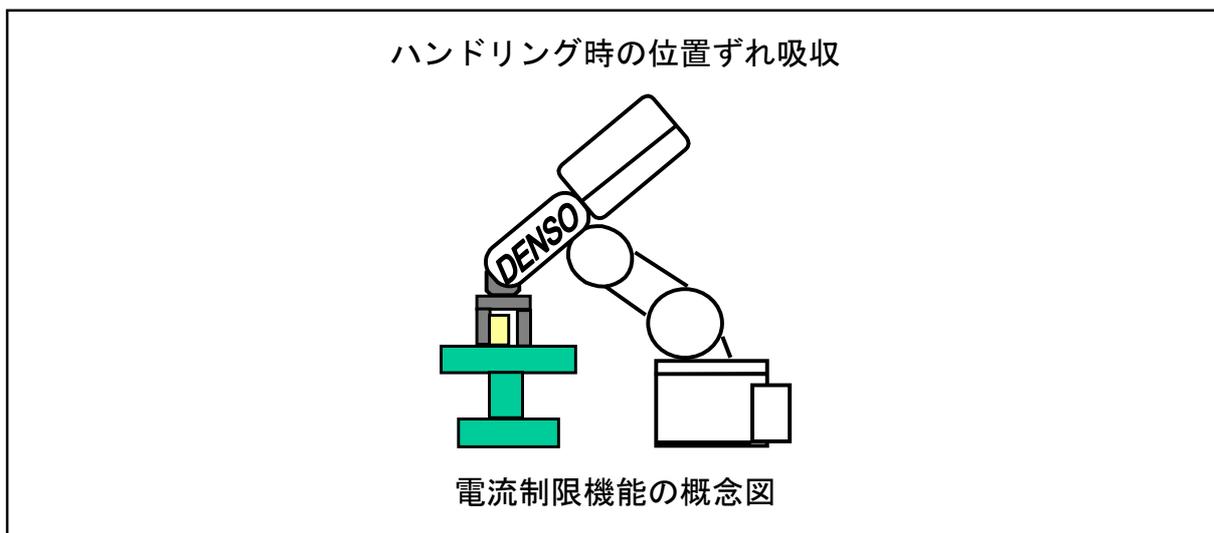
力制限機能は、ロボットにソフトウェアにて柔らかさを設定する機能です。

ロボットのハンドリング時の位置ずれを吸収したり、ロボットやワークに過大な力が加わるのを防止する事が可能です。本機能は、各関節毎に柔らかさを設定する機能（電流制限機能）とロボット先端の座標系の各要素毎に柔らかさを設定する機能（先端力制限機能）があります。

注： 電流制限機能は、Ver1.2以降、先端力制限機能は、V*-Dシリーズ（VM-6070Dは除く）のVer1.4以降で使用できます。

19.1 各軸電流制限機能

ロボットの各軸毎に柔らかさを設定する機能です。各軸モータのトルク（電流値）を制限する事で柔らかさを実現します。主にロボットやワークに過大な力が加わるのを防止したり、過負荷や過電流エラーにて停止するのを防止するために使用します。

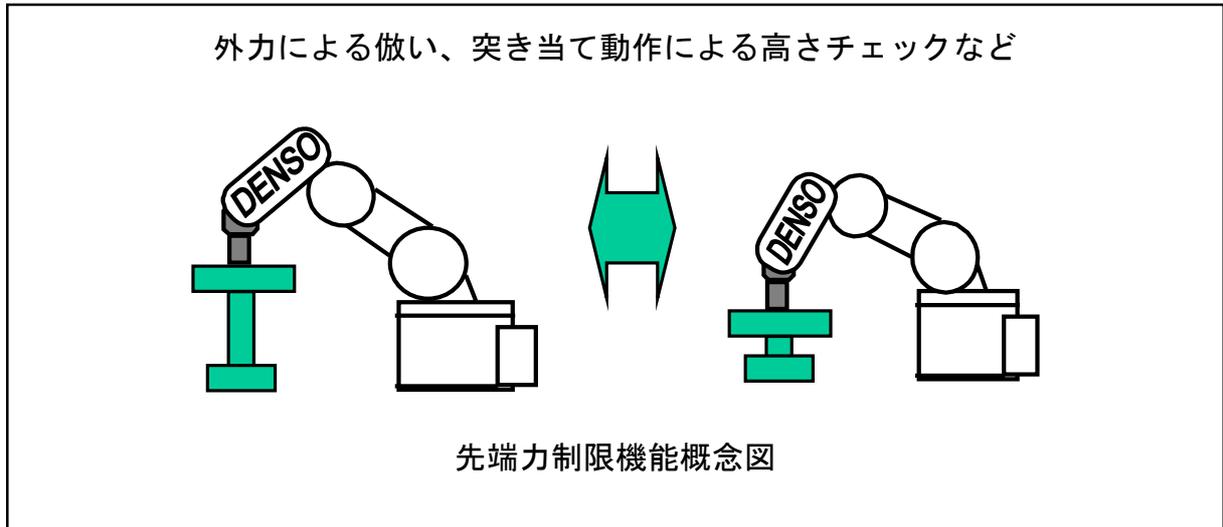


■ 電流制限機能の設定

ライブラリ機能を使用し、電流制限ライブラリ (SetCurLmt, ResetCurLmt) を実行することで機能を有効/無効に設定できます。使用方法の詳細については、プログラミングマニュアルを参照してください。

19.2 先端力制限機能

ロボット先端座標系の各要素毎に柔らかさを設定する機能です。先端の力制限値から各軸のモータトルクを制御する事で先端の柔らかさを実現します。座標系は、ベース座標、ツール座標、ワーク座標を選択できます。ロボットを、外力に対し特定方向に倣わせる場合や、突き当て動作による高さチェック等に使用します。



■ 先端力制限機能を使用可能にする設定

本機能は、拡張機能です。ティーチングペンダントにて力制限機能を使用可能に設定する必要があります。拡張機能は、一度使用可能に設定すれば、コントローラ電源OFF後も使用可能状態を保ちますので、再設定は不要です。

設定方法の詳細については、プログラミングマニュアルを参照してください。

■ 先端力制限機能の設定

ライブラリ機能を使用し、力制限ライブラリ (SetCompControl, ResetCompControl) を実行する事で機能を有効/無効に設定できます。

使用方法の詳細については、プログラミングマニュアルを参照してください。

レッスン20 その他の機能

力制限機能以外にデンソーロボットは下記のような特徴的機能を有しています。詳細はプログラミングマニュアルを参照してください。

機能名	機能と特徴の概略	関連コマンド
干渉エリア機能	ロボットのアーム先端が指令された領域にある間、指定したI/Oポートより信号を出力します。 原位置確認、作業位置確認に有効な命令です	AREA SETAREA RESETAREA
割り込み停止機能	専用割り込みポートの割り込み信号がONされたときに実行中の動作命令を終了し、次の動作命令に進みます。 外部入力によりロボット停止位置を変更したい場合に有効な命令です。 ただし、ロボットは減速停止するため停止するまでの距離はその動作命令の速度設定、実速度に依存します	INTERRUPT ON/OFF
動作／非動作コマンド並列実行機能	動作命令を実行中に非動作命令を並列に実行することができます。 サイクルタイムの短縮に有効な命令です。	IOBLOCK ON/OFF
マルチタスク機能	別プログラムを並列起動します。 外部インタロックの確認、視覚命令等を別プログラムで起動させるとサイクルタイムの短縮が可能になります。	RUN KILL 等
プログラムデバッグ機能	メッセージを表示、ブザーをならす事ができます。 ユーザプログラムのデバッグに有効です	PRINTMSG PRINTDBG BUZZER
最適可搬質量設定	ワーク、およびツールの質量に応じた加速度の設定が可能になります。 サイクルタイムの短縮が可能になります。	aspACLD aspChange

付録

▶ 付録1 用語	130
▶ 付録2 ロボットコントローラ各部の名称	141
▶ 付録3 ティーチングペンダント各部の名称	142
▶ 付録4 ティーチングペンダントの機能メニュー	144

付録1 用語

C

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

DIOマネージャ (DIO manager)

I/Oの状態をモニタしたり、I/Oの割付を管理するソフトウェア

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

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

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

J型変数 (J type variable)

各軸の値によって表される変数

N

NLIM (NLIM)

ソフトウェアリミットの負方向の端点の値 (⇔ PLIM)

O

Operator (operator)

WINCAPS II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

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

PACプログラムマネージャ (PAC program manager)

PACプログラムの開発を支援するソフトウェア。エディタ、コマンドビルダ、プログラムバンクなどの機能を持つ

PLIM (PLIM)

ソフトウェアリミットの正方向の端点の値
(⇔ NLIM)

Programmer (programmer)

WINCAPS II 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

R

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

S

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

T

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

X

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。アームマネージャが使用する

アームマネージャ (arm manager)

ロボットの動きをシミュレーションするソフトウェア

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。イーサネットでは通信するとき必要

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

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WINCAPS IIと通信するのに使用する

ウ

ウィンドウ (window)

画像を処理する範囲 (視覚用語)

エ

エッジ (edge)

輝度の変化点 (視覚用語)

エラーコード (error code)

ロボットあるいはWINCAPS II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。ティチングペンダントやWINCAPS II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

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。ティチング機能はない

オリент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

ク

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

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

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。システムマネージャが管理する

システムマネージャ (system manager)

WINCAPS IIの全情報の統括管理を行うソフトウェア

システム変数 (system variable)

プログラム内でシステムの状況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

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

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ー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

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



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)



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WINCAPS II (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)

オペレーティングパネルや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 (⇔ ソフトウェアリミット)

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

ユ

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

ヨー角 (yaw angle)

X軸回りの回転角

ラ

ライティール (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。WINCAPS II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

リモート操作 (remote operation)

アームマネージャ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

レフティール (LEFTY)

6軸ロボットの腕形態のひとつ (⇔ ライティール)

ロ

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イ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

ログマネージャ (log manager)

ロボットが持つ動作やエラーの記録を、パソコン上にコピーし管理するソフトウェア

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

い

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

う

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。ライティアー、レフティアーの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

き

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)

デンスーロボットで従来から使用されているロボット言語

け

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

こ

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

し

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やオペレーティングパネルの操作キーを使って、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

初期設定フロッピーディスク (initialization floppy disk)

工場出荷時のロボットの設定を記録したディスク。コントローラのメモリに異常が発生したとき、初期状態に復帰させるのに使用する

せ

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

そ

操作ログ (operation log)

ティーチングペンダントやオペレーティングパネル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

つ

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

て

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)

ベースを上にアームを下にして、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

と

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)

オペレーティングパネルや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

に

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

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。アバップとビローの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



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)



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



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

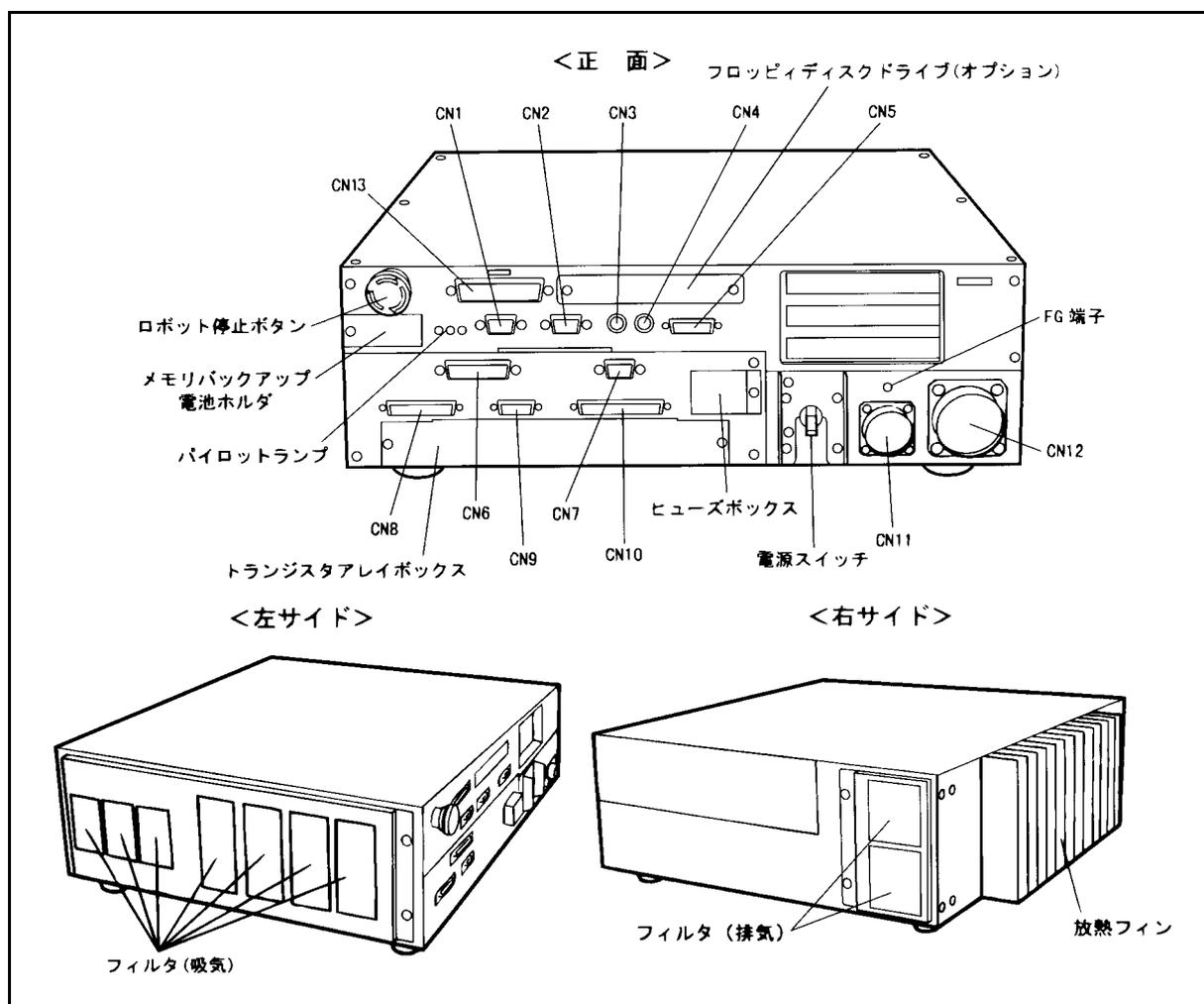


割り込みスキップ (interrupt skip)

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

付録2 ロボットコントローラ各部の名称

ロボットコントローラ各部の名称を、下図および下表に示します。



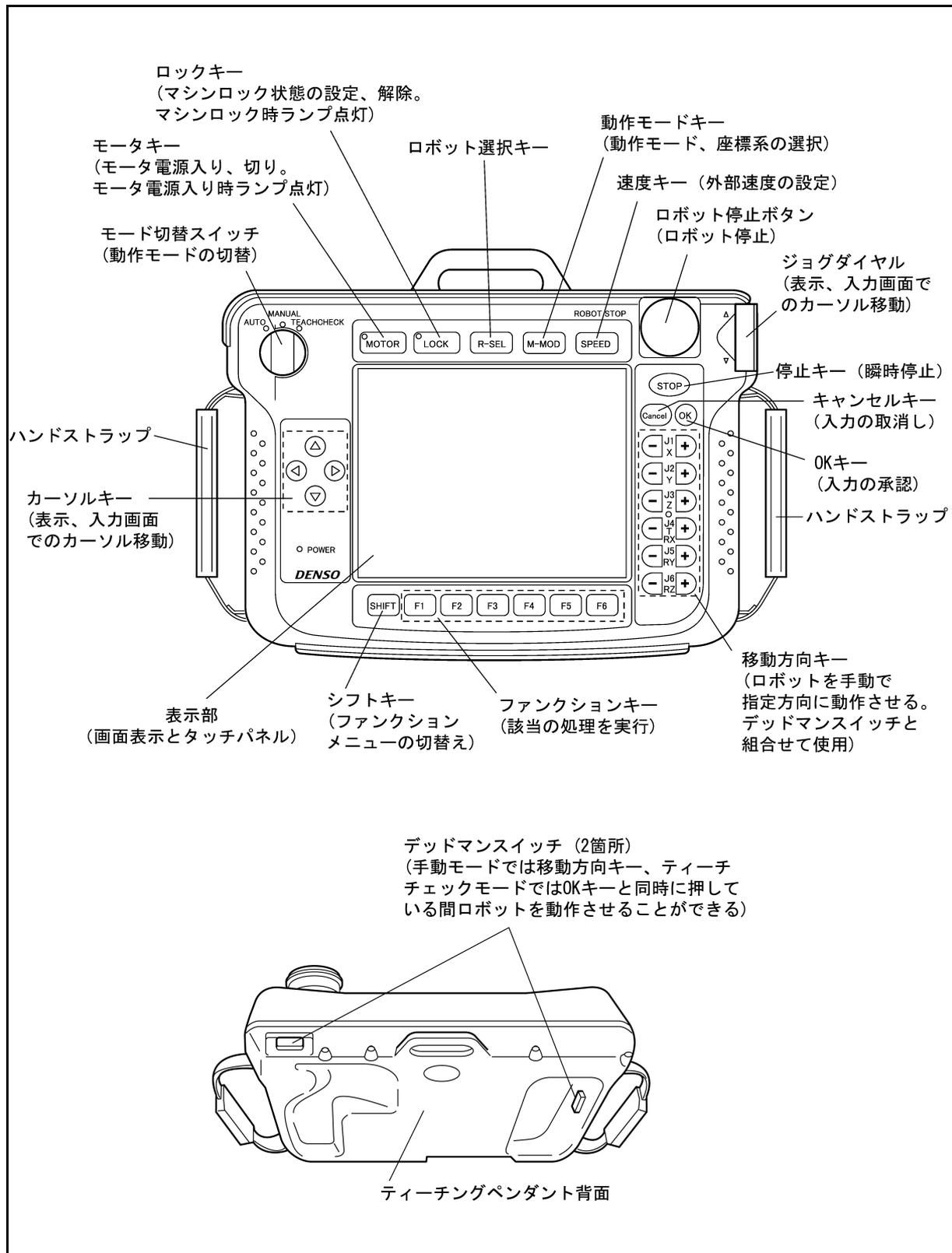
ロボットコントローラ各部の名称 (VS-Dシリーズ用の例)

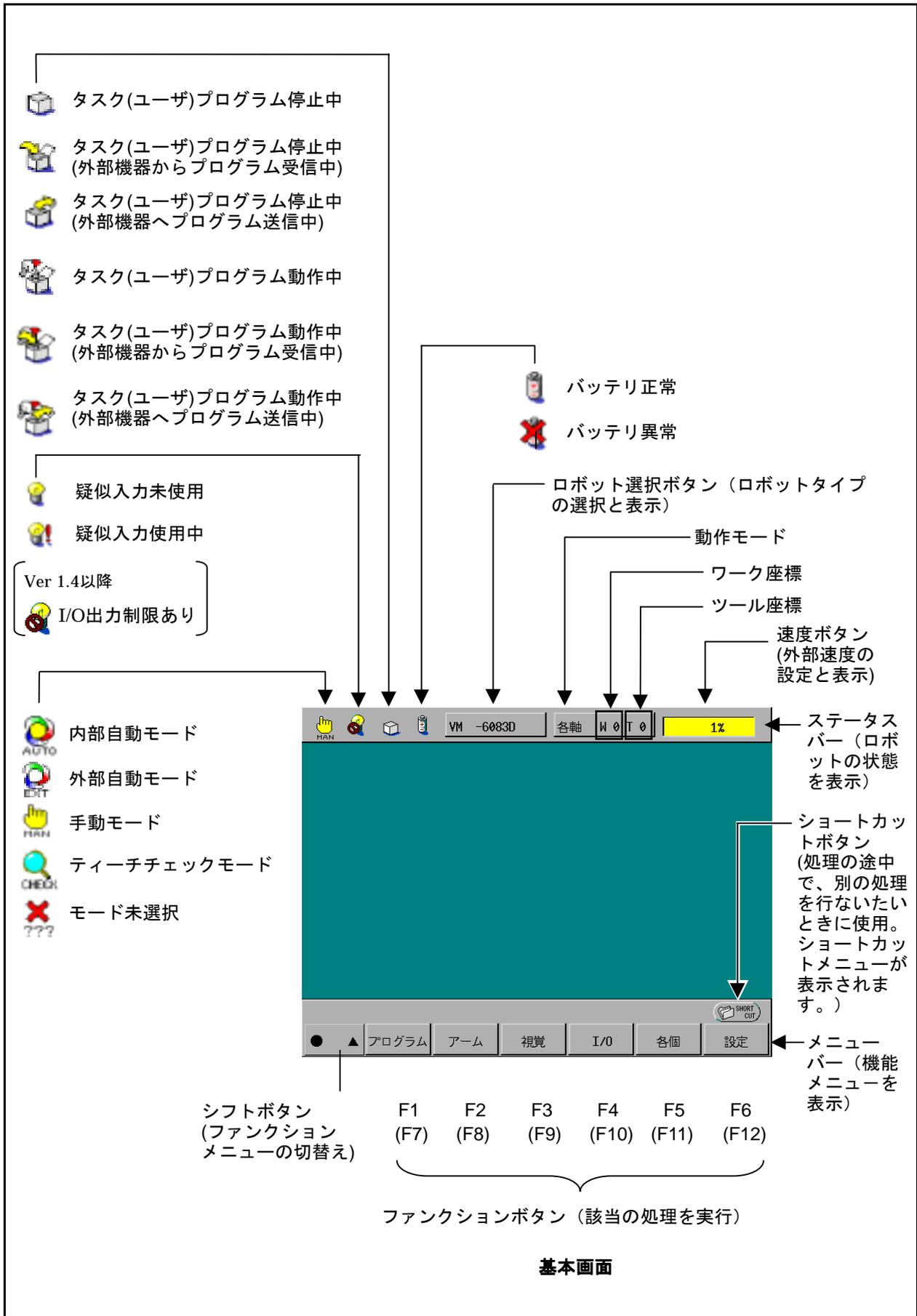
コネクタの名称 (VS-Dシリーズ用の例)

コネクタNo.	表示	名称	コネクタNo.	表示	名称
CN1	RS232C	シリアル用コネクタ	CN8	INPUT	汎用・専用入力用コネクタ
CN2	CRT	CRT用コネクタ	CN9	HAND I/O	ハンドI/O用コネクタ
CN3	KEYBD	キーボード用コネクタ	CN10	OUTPUT	汎用・専用出力用コネクタ
CN4	MOUSE	PS/2マウス用コネクタ	CN11	INPUT AC	電源コネクタ
CN5	PENDANT	ペンダント用コネクタ	CN12	MOTOR	モータコネクタ
CN6	PRINTER	プリンタ用コネクタ	CN13	ENCODER	エンコーダ用コネクタ
CN7	I/O POWER	I/O用電源コネクタ			

注意: ロボットコントローラのコネクタは、ビス止めまたはリング止めのロック機構になっています。コネクタは、しっかりとロックしてください。ロックしないと接触不良を起こし、エラーが発生する原因になります。
また、ロボットコントローラの電源スイッチを入れたまま電源コネクタ、モータコネクタを脱着すると、ロボットコントローラの内部回路が破損するおそれがあります。電源スイッチを切ってからコネクタの脱着を実施してください。

付録3 ティーチングペンダント各部の名称

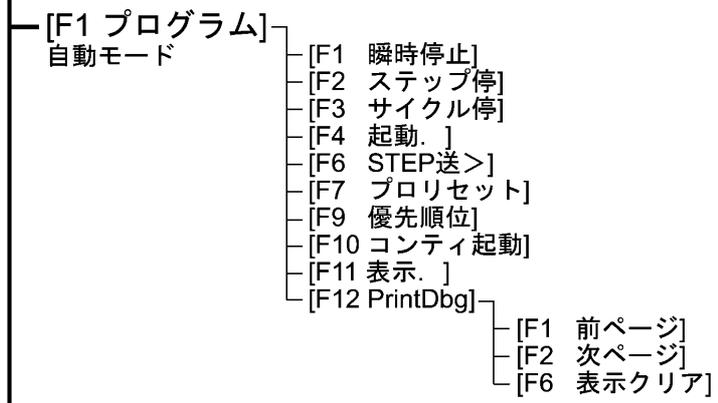
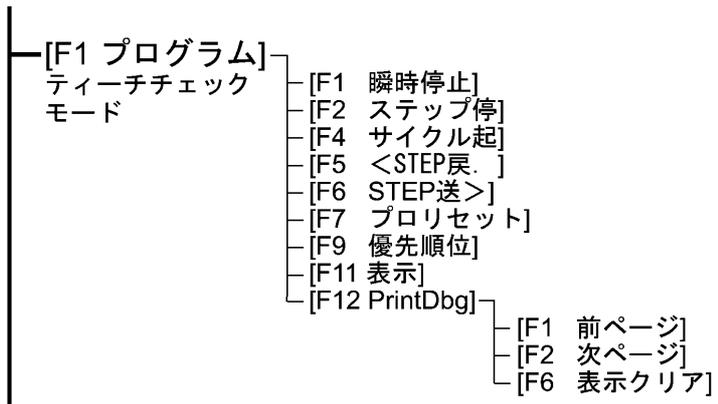




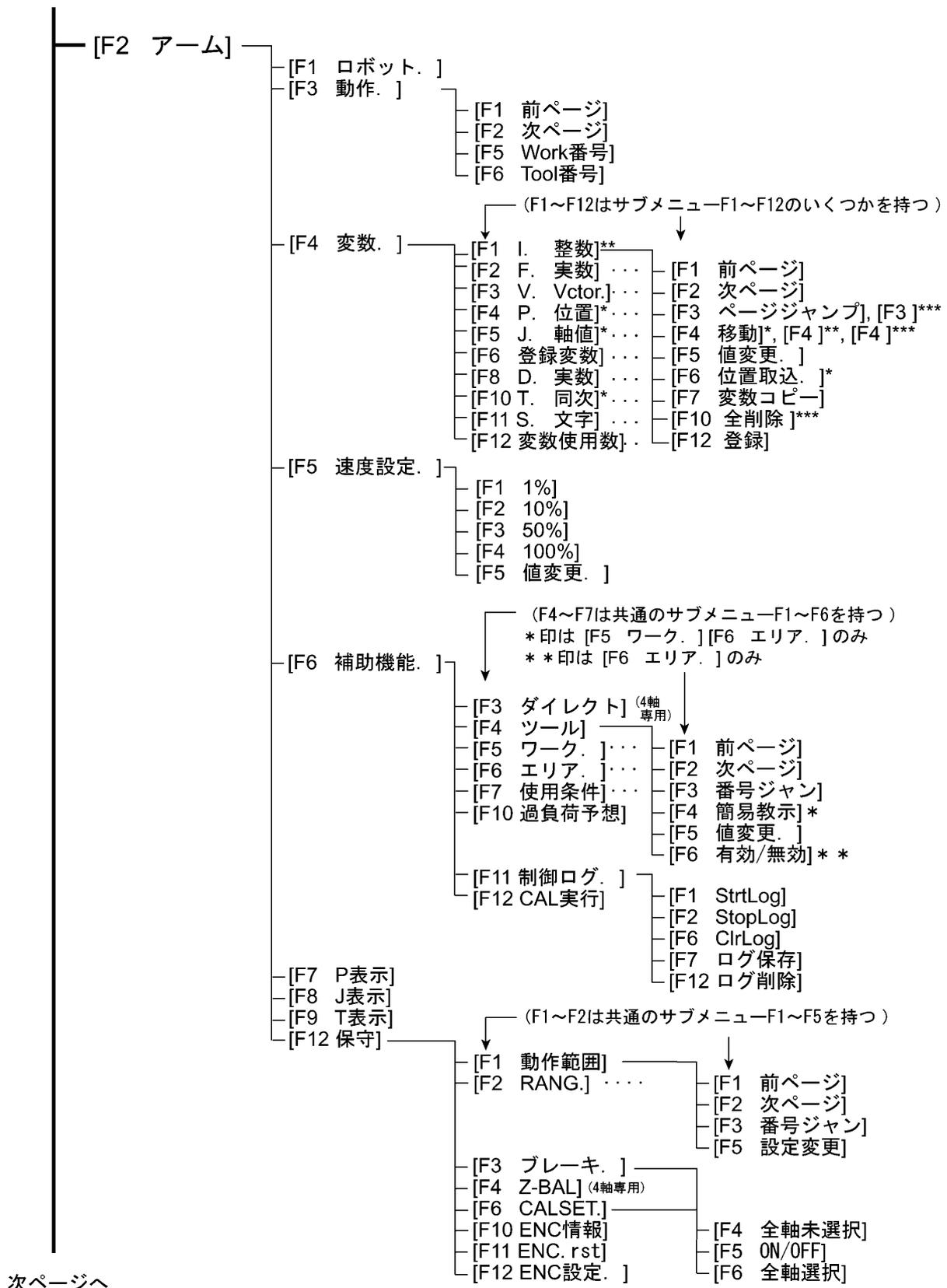
付録4 ティーチングペンダントの機能メニュー

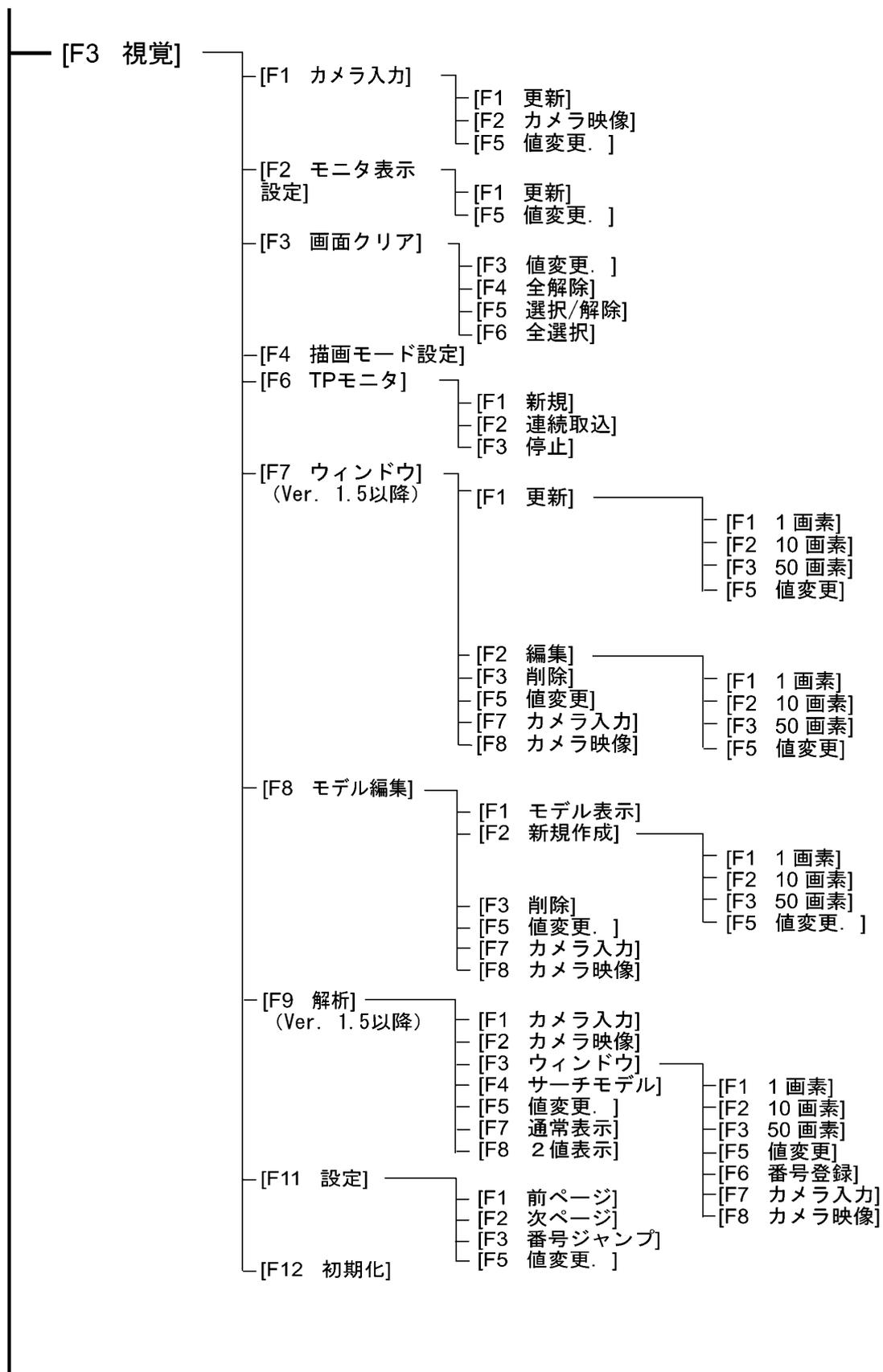


次ページへ

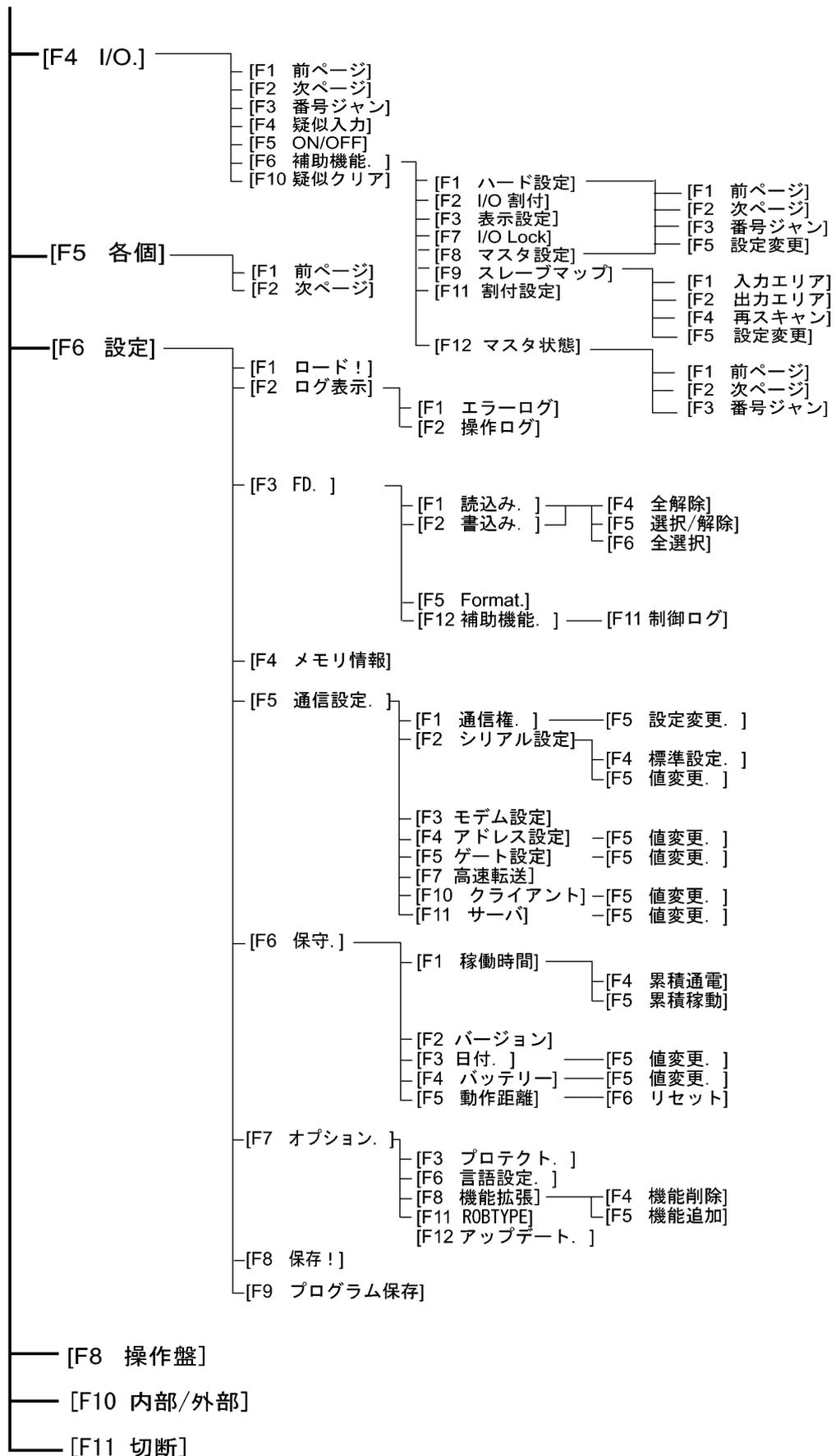


次ページへ





次ページへ



デンソーロボット
垂直多関節型 V* -D/-E シリーズ

入門編

初 版 2000 年 5 月
第 2 版 2001 年 4 月
第 3 版 2001 年 10 月

株式会社デンソーウェーブ FA 事業部

9D50C

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。

