

デンソーロボット

垂直多関節型

V* -D/-E シリーズ

水平多関節型

H* -D/-E シリーズ

直角座標型

XYC-4D シリーズ

視覚装置

μ Vision シリーズ

プログラミングマニュアルⅡ

PAC ライブラリ

(Ver. 1.98)

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。

本書が扱う対象製品

■対象製品

- ・垂直多関節型ロボット V*-D/-E シリーズ
- ・水平多関節型ロボット H*-D/-E シリーズ
- ・直角座標型ロボット XYC-4D シリーズ
- ・視覚装置 μ Vision シリーズ

■ロボットコントローラのバージョン（注）

- ・RC5型コントローラのVer. 1.9**までが対象
-

注：ロボットコントローラのバージョンはコントローラの上面に貼られている「コントローラ設定表」のメインソフトVer. 欄に記されています。
またティーチングペンダントからは、[基本画面]-[F6設定]-[F6保守]-[F2バージョン]で表示されるROMバージョン欄から確認できます。

お願い

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

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	デンス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 II 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 II 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 (I)、(II) (本書)	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 インターフェース説明書	RC5コントローラの概要、外部機器とのインターフェース、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 II 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。

本書の構成

本書の構成は、以下のようになっております。

コマンド一覧表 -アルファベット順コマンド一覧
-機能別コマンド一覧

PAC ライブラリ

WINCAPS II に標準で付属する、プログラムライブラリについて説明しています。

索引

目次

はじめに.....	i
取扱説明書の構成.....	ii
アルファベット順コマンド一覧	
機能別コマンド一覧	
1 プログラムライブラリの使い方.....	1
2 プログラムライブラリの分類.....	1
3 従来言語のライブラリ.....	2
4 パレタイジングのライブラリ.....	17
5 ツール操作のライブラリ.....	30
6 入出力のライブラリ.....	32
7 アーム動作のライブラリ.....	33
8 TP 簡易操作盤画面作成.....	80
8.1 操作盤画面作成サンプルプログラムの使い方.....	80
8.2 操作盤画面作成サンプルプログラム.....	82
8.3 操作盤画面作成用のライブラリ.....	89
9 視覚のライブラリ.....	93

アルファベット順コマンド一覧

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

コマンド	機能	4 軸	6 軸	視覚装置	説明 ページ
A					
ArchMove	アーチモーションを実施します。		V1.9		75
arrange_button_pos	画面上のボタン配置（位置・大きさなど）を指定します。		V1.7	V1.7	92
arrange_button_size	画面上のボタン配置（位置・大きさなど）を指定します。		V1.7	V1.7	91
aspACLD	内部負荷条件値を変更します。負荷条件値は、先端負荷質量（g）、負荷重心位置（mm）ですべて指定します。	◎	◎		2
aspChange	最適可搬質量設定モードの内部モードを選択します。	◎	◎		3
C					
ClearSrvMonitor	サーボ単軸データモニタの取得データのポインタを初期化します。（Ver. 1.5以降対応）		V1.5	V1.5	79
D					
dioSync	DIOに接続された外部機器（シーケンサなど）との同期	◎	◎		32
M					
make_LABEL	タイトル（ラベル）を作成します。		V1.7	V1.7	89
make_LED	LED ボタンを作成します。		V1.7	V1.7	89
make_PARAM_BOX	変数ボタン（入力&表示BOX）を作成します。		V1.7	V1.7	90
make_PB	P B ボタンを作成します。		V1.7	V1.7	89
MotionComp	動作命令実行が完了したかどうかを判断します。		V1.7	V1.7	74
MotionSkip	実行中の動作命令を中断します。（Ver. 1.5以降対応）		V1.5	V1.5	73
mvResetPulseWidth	停止時許容パルス幅をデフォルト値にします。	◎	◎		33
mvResetPulseWidthJnt	指定した付加軸の、停止時許容パルス幅をデフォルト値に戻します。		V1.5	V1.5	33
mvResetTimeOut	動作終了タイムアウト値をデフォルト値にします。	◎	◎		34
mvReverseFlip	4 軸形態反転	◎	◎		34
mvSetPulseWidth	停止時許容パルス幅設定	◎	◎		35
mvSetPulseWidthJnt	指定した付加軸の、停止時許容パルス幅を設定します。				35
mvSetTimeOut	動作終了タイムアウト値設定	◎	◎		36
N					
ndApra	ツール座標系指定の絶対動作（4軸ロボット専用）	◎			14
ndDepa	ツール座標系指定の絶対動作（4軸ロボット専用）	◎			15
ndInb	指定ポートの入力を2進数とみなして10進数に変換	◎	◎		4
ndJf	外部機器からOK/NGを受信し、条件分岐（RS232C入出力）	◎	◎		5
nd0nb	10進数を2進数に変換して指定ポートより出力	◎	◎		6
nd0nbI	10進数を2進数に変換して指定ポートより出力	◎	◎		6
ndTC	TC時間を設定します。	◎	V1.2		16
ndTS	TS時間、スロー速度を設定します。	◎	V1.2		16
ndVcom	外部機器との通信動作（カーネル）。（RS232C入出力）	◎	◎		13
ndVdt	外部機器から転送された変数を記憶（RS232C入出力）	◎	◎		7

4軸 6軸 視覚装置

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

コマンド	機能	4軸	6軸	視覚装置	説明 ページ
ndVis	外部機器へ指定した2桁の整数を転送 (RS232C 入出力)	◎	◎		8
ndVput	外部機器へ位置姿勢を転送 (RS232C 入出力)	◎	◎		9
ndVrst	外部機器の初期化 (RS232C 入出力)	◎	◎		10
ndVset	外部機器からデータ受信 (RS232C 入出力)	◎	◎		11
ndVType	外部機器との通信プロトコルを指定します。(RS232C 入出力)	◎	◎		12
O					
OffPWM	指定した軸のPWMスイッチング制御を解除します。	◎			48
OffSrvLock	指定した軸のサーボロックを解除します。	◎			46
OnPWM	指定した軸をPWMスイッチング制御します。	◎			47
OnSrvLock	指定した軸をサーボロック状態にします。	◎			45
P					
pltDecCnt	パレタイジング総カウンタのデクリメント	◎	◎		17
pltGetCnt	パレタイジング総カウンタの取得	◎	◎		17
pltGetK	パレタイジング設定値Kの取得	◎	◎		18
pltGetK1	パレタイジングカウンタ K1 の取得	◎	◎		18
pltGetM	パレタイジング設定値 M の取得	◎	◎		18
pltGetM1	パレタイジングカウンタ M1 の取得	◎	◎		19
pltGetN	パレタイジング設定値 N の取得	◎	◎		19
pltGetN1	パレタイジングカウンタ N1 の取得	◎	◎		19
pltGetNextPos	次ポジションの取得	◎	◎		20
pltGetPLT1END	パレタイジング1段終了フラグの取得	◎	◎		20
pltGetPLTEND	パレタイジング全段終了フラグの取得	◎	◎		20
pltIncCnt	パレタイジング総カウンタのインクリメント	◎	◎		21
pltInit1	パレタイジング初期化テンプレート1	◎	◎		21
pltInitialize	パレタイジング初期化	◎	◎		22
pltKernel	パレタイジング動作(カーネル)	◎	◎		23
pltLetCnt	パレタイジング総カウンタの設定	◎	◎		24
pltLetK1	パレタイジングカウンタ K1 の設定	◎	◎		24
pltLetM1	パレタイジングカウンタ M1 の設定	◎	◎		25
pltLetN1	パレタイジングカウンタ N1 の設定	◎	◎		25
pltMain1	パレタイジング テンプレート1	◎	◎		26
pltMain2	パレタイジング テンプレート2	◎	◎		26
pltMove	標準パレタイジング テンプレート1	◎	◎		27
pltMove0	標準パレタイジング動作1	◎	◎		28
pltResetAll	パレタイジングカウンタすべてのリセット	◎	◎		29
pltResetPLT1END	パレタイジング1段終了フラグのリセット	◎	◎		29
pltResetPLTEND	パレタイジング全段終了フラグのリセット	◎	◎		29
R					
ResetCompControl	力制限機能を無効にします。			V1.4	57
ResetCompEralw	力制限時のツール端の位置、姿勢偏差許容値を初期化します。			V1.4	69
ResetCompJLimit	力制限時の電流制限値を初期化します。			V1.4	66
ResetCompRate	力制限時の柔らかさの割合を初期化します。			V1.4	62
ResetCompVMode	力制限時の速度制御モードを無効にします。			V1.4	67
ResetCurlmt	指定した軸のモータ電流制限を解除します。	◎	V1.2		42
Resetcycloid	サイクロイド動作モードから通常モードに移行します。	◎	V1.4		51
ResetDampRate	力制限時の粘性割合を初期化します。			V1.4	71
ResetEralw	指定した軸の偏差許容値を初期値に変更します。	◎	V1.2		44

4軸 6軸 視覚装置

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

コマンド	機能	4軸	6軸	視覚装置	説明ページ
ResetFrcAssist	力制限時のオフセット力を初期化します。		V1.4		64
ResetFrcLimit	力制限割合を初期化します。		V1.4		60
ResetGravity	重力バランスを無効にします。		V1.2		38
ResetGrvOffset	重力補償値の補正を無効にします。		V1.2		40
ResetVibControl	残留振動低減制御モードから通常制御モードへ戻します。	V1.9	V1.4		72
S					
SetArchParam	アーチモーションにおける、上昇動作中に横方向動作を開始する位置（アーチ開始位置）および下降動作中に横方向動作を終了する位置（アーチ完了位置）の設定。	V1.9			76
set_button_param	ボタン属性（タイプ・色・形など）を指定します。				91
SetCompControl	力制限機能を有効にします。		V1.4		55
SetCompEralw	力制限時のツール端の位置、姿勢偏差許容値を設定します。		V1.4		68
SetCompFControl	力制限機能を有効にします。ただし、重力補償補正処理は実行されません。		V1.4		56
SetCompJLimit	力制限時の電流制限値を設定します。		V1.4		65
SetCompRate	力制限時の柔らかさの割合を設定します。		V1.4		61
SetCompVMode	力制限時の速度制御モードを設定します。		V1.4		66
SetCPSpdMode	CP動作時のTCP速度設定を変更する。	V1.8	V1.8		52
SetCurLmt	指定した軸のモータ電流値を制限します。	○	V1.2		40
Setcycloid	P T P動作エンド動作時のオーバーシュート量および残留振動を抑えるサイクロイド動作モードに移行します。	◎	V1.4		49
SetCycloidJnt	付加軸のエンド動作時のオーバーシュート量および残留振動を抑えるサイクロイド動作モードに移行します。	V1.5	V1.5		50
SetDampRate	力制限時の粘性割合を設定します。		V1.4		70
SetEralw	指定した軸の偏差許容値を変更します。	◎	V1.2		43
SetForce_HC	HCロボットにてZ軸の推力（単位：N）を指定した電流制限ライブラリです。	◎			54
SetForce_HM	HM/HSロボットにてZ軸の推力（単位：N）を指定した電流制限ライブラリです。	◎			53
SetFrcAssist	力制限時のオフセット力を設定します。		V1.4		63
SetFrcCoord	力制限設定座標系を選択します。		V1.4		76
SetFrcLimit	力制限割合を設定します。		V1.4		58
SetGravity	各関節の静荷重（重力トルク）を補正し、重力バランスを設定します。		V1.2		37
SetGrvOffset	各関節の重力トルクより重力補償値を補正します。		V1.2		39
SetMonitorCond	サーボ単軸データモニタ機能のモニタ条件を設定します。	V1.5	V1.5		77
SetVibControl	残留振動低減制御モードにします。	V1.9	V1.4		71
single_button_set	ボタンを一つだけ作成します。	V1.7	V1.7		90
StartSrvMonitor	サーボ単軸データモニタを開始します。	V1.5	V1.5		78
StopSrvMonitor	サーボ単軸データモニタを終了します。	V1.5	V1.5		78
T					
tolChange	ツールチェンジ	◎	◎		30
tolInit1	ツールチェンジ初期化テンプレート1	◎	◎		30
tolInitialize	ツールチェンジ初期化	◎	◎		31
tolKernel	ツールチェンジ動作(カーネル)	◎	◎		31
tolMain1	ツールチェンジ テンプレート1	◎	◎		31
V					
viTran6	視覚座標をロボット座標に変換（6軸）	◎	◎		93

機能別コマンド一覧

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

機能区分	コマンド	機能	4 軸	6 軸	視覚装置	説明ページ
PAC ライブラリ						
従来言語	aspACLD	内部負荷条件値を変更します。負荷条件値は、先端負荷質量 (g)、負荷重心位置 (mm) ですべて指定します。	◎	◎		2
	aspChange	最適可搬質量設定モードの内部モードを選択します。	◎	◎		3
	ndInb	指定ポートの入力を 2 進数とみなして 10 進数に変換	◎	◎		4
	ndJf	外部機器から OK/NG を受信し、条件分岐 (RS232C 入出力)	◎	◎		5
	ndOnb	10 進数を 2 進数に変換して指定ポートより出力	◎	◎		6
	ndOnbI	10 進数を 2 進数に変換して指定ポートより出力	◎	◎		6
	ndVdt	外部機器から転送された変数を記憶 (RS232C 入出力)	◎	◎		7
	ndVis	外部機器へ指定した 2 桁の整数を転送 (RS232C 入出力)	◎	◎		8
	ndVput	外部機器へ位置姿勢を転送 (RS232C 入出力)	◎	◎		9
	ndVrst	外部機器の初期化 (RS232C 入出力)	◎	◎		10
	ndVset	外部機器からデータ受信 (RS232C 入出力)	◎	◎		11
	ndVType	外部機器との通信プロトコルを指定します。(RS232C 入出力)	◎	◎		12
	ndVcom	外部機器との通信動作 (カーネル)。(RS232C 入出力)	◎	◎		13
	ndApra	ツール座標系指定の絶対動作 (4 軸ロボット専用)	◎			14
	ndDepa	ツール座標系指定の絶対動作 (4 軸ロボット専用)	◎			15
	ndTC	TC 時間を設定します。	◎	V1.2		16
	ndTS	TS 時間、スロー速度を設定します。	◎	V1.2		16
パレタイジング	pltDecCnt	パレタイジング総カウンタのデクリメント	◎	◎		17
	pltGetCnt	パレタイジング総カウンタの取得	◎	◎		17
	pltGetK	パレタイジング設定値 K の取得	◎	◎		18
	pltGetK1	パレタイジングカウンタ K1 の取得	◎	◎		18
	pltGetM	パレタイジング設定値 M の取得	◎	◎		18

4軸 6軸 視覚装置

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

機能区分	コマンド	機能	4軸	6軸	視覚装置	説明ページ
	pltGetM1	パレタイジングカウンタ M1 の取得	◎	◎		19
	pltGetN	パレタイジング設定値 N の取得	◎	◎		19
	pltGetN1	パレタイジングカウンタ N1 の取得	◎	◎		19
	pltGetNextPos	次ポジションの取得	◎	◎		20
	pltGetPLT1END	パレタイジング 1 段終了フラグの取得	◎	◎		20
	pltGetPLTEND	パレタイジング全段終了フラグの取得	◎	◎		20
	pltIncCnt	パレタイジング総カウンタのインクリメント	◎	◎		21
	pltInit1	パレタイジング初期化テンプレート 1	◎	◎		21
	pltInitialize	パレタイジング初期化	◎	◎		22
	pltKernel	パレタイジング動作(カーネル)	◎	◎		23
	pltLetCnt	パレタイジング総カウンタの設定	◎	◎		24
	pltLetK1	パレタイジングカウンタ K1 の設定	◎	◎		24
	pltLetM1	パレタイジングカウンタ M1 の設定	◎	◎		25
	pltLetN1	パレタイジングカウンタ N1 の設定	◎	◎		25
	pltMain1	パレタイジング テンプレート 1	◎	◎		26
	pltMain2	パレタイジング テンプレート 2	◎	◎		26
	pltMove	標準パレタイジング テンプレート 1	◎	◎		27
	pltMove0	標準パレタイジング動作 1	◎	◎		28
	pltResetAll	パレタイジングカウンタすべてのリセット	◎	◎		29
	pltResetPLT1END	パレタイジング 1 段終了フラグのリセット	◎	◎		29
	pltResetPLTEND	パレタイジング全段終了フラグのリセット	◎	◎		29
ツール操作	tolChange	ツールチェンジ	◎	◎		30
	tolInit1	ツールチェンジ初期化テンプレート 1	◎	◎		30
	tolInitialize	ツールチェンジ初期化	◎	◎		31
	tolKernel	ツールチェンジ動作(カーネル)	◎	◎		31
	tolMain1	ツールチェンジ テンプレート 1	◎	◎		31
入出力	dioSync	DIO に接続された外部機器(シーケンサなど)との同期	◎	◎		32
アーム動作	mvResetPulseWidth	停止時許容パルス幅をデフォルト値にします。	◎	◎		33
	mvResetPulseWidthJnt	指定した付加軸の、停止時許容パルス幅を設定します。	V1.5	V1.5		33
	mvResetTimeOut	動作終了タイムアウト値をデフォルト値にします。	◎	◎		34
	mvReverseFlip	4 軸形態反転	◎	◎		34
	mvSetPulseWidth	停止時許容パルス幅設定	◎	◎		35
	mvSetPulseWidthJnt	指定した付加軸の、停止時許容パルス幅を設定します。	V1.5	V1.5		35
	mvSetTimeOut	動作終了タイムアウト値設定	◎	◎		36
	SetGravity	各関節の静荷重(重力トルク)を補正し、重力バランスを設定します。		V1.2		37
	ResetGravity	重力バランスを無効にします。		V1.2		38
	SetGrvOffset	各関節の重力トルクより重力補償値を補正します。		V1.2		39
	ResetGrvOffset	重力補償値の補正を無効にします。		V1.2		40
	SetCurLmt	指定した軸のモータ電流値を制限します。	○	V1.2		40

4軸 6軸 視覚装置

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

機能区分	コマンド	機能	4軸	6軸	視覚装置	説明 ページ
	ResetCurLmt	指定した軸のモータ電流制限を解除します。	◎	V1.2		42
	SetEralw	指定した軸の偏差許容値を変更します。	◎	V1.2		43
	ResetEralw	指定した軸の偏差許容値を初期値に変更します。	◎	V1.2		44
	OnSrvLock	指定した軸をサーボロック状態にします。	◎			45
	OffSrvLock	指定した軸のサーボロックを解除します。	◎			46
	OnPWM	指定した軸を PWM スイッチング制御します。	◎			47
	OffPWM	指定した軸の PWM スイッチング制御を解除します。	◎			48
	Setcycloid	P T P 動作エンド動作時のオーバーシュート量および残留振動を押さえるサイクロイド動作モードに移行します。	◎	V1.4		49
	SetCycloidJnt	付加軸のエンド動作時のオーバーシュート量および残留振動を抑えるサイクロイド動作モードに移行します。	V1.5	V1.5		50
	Resetcycloid	サイクロイド動作モードから通常モードに移行します。	◎	V1.4		51
	ResetCycloidJnt	サイクロイド動作モードを解除し、通常モードに移行します。	V1.5	V1.5		52
	SetCPSpdMode	CP 動作時の TCP 速度設定を変更する。	V1.8	V1.8		52
	SetForce_HM	HM/HS ロボットにて Z 軸の推力 (単位: N) を指定した電流制限ライブラリです。	◎			53
	SetForce_HC	HC ロボットにて Z 軸の推力 (単位: N) を指定した電流制限ライブラリです。	◎			54
	SetCompControl	力制限機能を有効にします。		V1.4		55
	SetCompFControl	力制限機能を有効にします。ただし、重力補償補正処理は実行されません。		V1.4		56
	ResetCompControl	力制限機能を無効にします。		V1.4		57
	SetFrcCoord	力制限設定座標系を選択します。		V1.4		58
	SetFrcLimit	力制限割合を設定します。		V1.4		59
	ResetFrcLimit	力制限割合を初期化します。		V1.4		60
	SetCompRate	力制限時の柔らかさの割合を設定します。		V1.4		61
	ResetCompRate	力制限時の柔らかさの割合を初期化します。		V1.4		62
	SetFrcAssist	力制限時のオフセット力を設定します。		V1.4		63
	ResetFrcAssist	力制限時のオフセット力を初期化します。		V1.4		64
	SetCompJLimit	力制限時の電流制限値を設定します。		V1.4		65
	ResetCompJLimit	力制限時の電流制限値を初期化します。		V1.4		66
	SetCompVMode	力制限時の速度制御モードを設定します。		V1.4		66
	ResetCompVMode	力制限時の速度制御モードを無効にします。		V1.4		67
	SetCompEralw	力制限時のツール端の位置、姿勢偏差許容値を設定します。		V1.4		68

4軸 6軸 視覚装置

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

機能区分	コマンド	機能	4軸 6軸 視覚装置	説明 ページ
	ResetCompEralw	力制限時のツール端の位置、姿勢偏差許容値を初期化します。	V1.4	69
	SetDampRate	力制限時の粘性割合を設定します。	V1.4	70
	ResetDampRate	力制限時の粘性割合を初期化します。	V1.4	71
	SetVibControl	残留振動低減制御モードにします。	V1.9 V1.4	71
	ResetVibControl	残留振動低減制御モードから通常制御モードへ戻します。	V1.9 V1.4	72
	MotionSkip	実行中の動作命令を中断します。	V1.5 V1.5	73
	MotionComp	動作命令実行が完了したかどうかを判断します。	V1.5 V1.5	74
	archMove	アーチモーションを実施します。	V1.9	75
	SetArchParam	アーチモーションにおける、上昇動作中に横方向動作を開始する位置（アーチ開始位置）および下降動作中に横方向動作を終了する位置（アーチ完了位置）の設定。	V1.9	76
サーボ単軸データモニタ	SetMonitorCond	サーボ単軸データモニタ機能のモニタ条件を設定します。	V1.5 V1.5	77
	StartSrvMonitor	サーボ単軸データモニタを開始します。	V1.5 V1.5	78
	StopSrvMonitor	サーボ単軸データモニタを終了します。	V1.5 V1.5	78
	ClearSrvMonitor	サーボ単軸データモニタの取得データのポイントを初期化します。（Ver. 1.5以降対応）	V1.5 V1.5	79
操作盤画面作成用のライブラリ	make_PB	P Bボタンを作成します。	V1.7 V1.7	89
	make_LED	LED ボタンを作成します。	V1.7 V1.7	89
	make_LABEL	タイトル（ラベル）を作成します。	V1.7 V1.7	89
	make_PARAM_BOX	変数ボタン（入力&表示BOX）を作成します。	V1.7 V1.7	90
	single_button_set	ボタンを一つだけ作成します。	V1.7 V1.7	90
	set_button_param	ボタン属性（タイプ・色・形など）を指定します。	V1.7 V1.7	91
	arrange_button_size	画面上のボタン配置（位置・大きさなど）を指定します。	V1.7 V1.7	91
	arrange_button_pos	画面上のボタン配置（位置・大きさなど）を指定します。	V1.7 V1.7	92
視覚	viTran6	視覚座標をロボット座標に変換（6軸）	◎ ◎	93

1 プログラムライブラリの使い方

プログラムライブラリを使用するには、WINCAPS II 中のPACマネージャからプログラムバンク（ツール）を起動し、必要なライブラリを追加する必要があります。

プログラムバンクの操作法に関しては、WINCAPS II ガイドを参照してください。

2 プログラムライブラリの分類

標準プログラムライブラリはWINCAPS II のプログラムバンクに下表のような分類で提供されています。

分類（クラス名）	内 容
従来言語	従来言語のコマンドと同等機能を提供します。
パレタイジング	パレタイジング機能を提供します。
ツール操作	ツール操作関連の機能を提供します。
入出力	DIO、RS232C入出力関連の機能を提供します。
アーム動作	アーム動作関連の機能を提供します。
視覚	視覚動作関連の機能を提供します。
Ver. 1.2 互換	コントローラのソフトバージョンが Ver. 1.2*以前で使用可能なライブラリを提供します。 Ver. 1.2*以前のロボットで下記のライブラリを使用する場合、「Ver. 1.2 互換」クラスに提供されているものを使用してください。 「Ver. 1.2 互換」クラスのライブラリ ndVcom、pltMove、pltMoveO、ResetCurLmt、ResetEralw 注：このライブラリを「Ver. 1.2 互換」以外のクラスに提供されているものから使用するとコンパイル異常が発生します。
μ Vision-21	μ Vision-21 動作関連の機能を提供します。
操作盤	TP 簡易操作盤画面作成用の機能を提供します。 注：操作盤については、標準プログラムライブラリに加えて、「サンプルプログラム」も準備されています。サンプルプログラムについては、8.1 項を参照してください。
移動ロボット	移動ロボット動作関連の機能を提供します。 移動ロボットのライブラリについては、「移動ロボット取扱説明書 追補版」を参照してください。

3 従来言語のライブラリ

RC3コントローラ搭載ロボットで使用していた従来言語のコマンドと同等機能を提供するためのライブラリです。

aspACLD (ライブラリ)

機能	内部負荷条件値を変更します。負荷条件値は、先端負荷質量 (g)、負荷重心位置 (mm) (注 1) ですべて指定します。
書式	aspACLD (<先端負荷質量>, <先端負荷重心位置 X 座標>, <先端負荷重心位置 Y 座標>, <先端負荷重心位置 Z 座標>) (注 1) 注 1: 4 軸ロボットの Ver. 1.9 以降の場合 aspACLD (<先端負荷質量>, <先端負荷重心位置 X 座標>, <先端負荷重心位置 Y 座標>, <先端負荷イナーシャ>)
説明	<p>先端負荷質量は、ロボットの第 6 軸に取り付く負荷 (ツール+ワーク) の質量です。単位は (g) で指定します。</p> <p>負荷重心位置は、負荷の重心位置をツール 0 座標系で指定します。単位は (mm) です。</p> <p>負荷イナーシャは、負荷の重心位置を原点とした Z 軸まわりのイナーシャです。単位は kgcm^2 です。</p> <p>6 軸ロボットの例で説明すると、ツール 0 座標系の原点は、6 軸フランジ中心、Y 成分はフランジ中心から $\Phi 5H7$ ($\Phi 6H7$) ピン穴方向 (オリエントベクトル方向)、Z 成分は、フランジ中心を通りフランジ面に垂直な方向 (アプローチベクトル方向)、X 成分は、オリエントベクトルを Y 軸、アプローチベクトルを Z 軸としたときの右手座標系における X 軸方向 (ノーマルベクトル方向) になります。プログラミングマニュアル I の「4.7 「使用条件」における最適可搬質量設定機能」を参照してください。</p> <p>先端負荷質量、負荷重心位置 X、負荷重心位置 Y、負荷重心位置 Z (負荷イナーシャ) の 4 つの値の中の 1 つだけ変更する場合でも、4 つの値をすべて記述してください。</p> <p>負荷条件値の切り替えには、約 0.1 秒の実行時間がかかります。頻繁に負荷条件を切り替えると動作時間遅れの要因になります。また、パス動作中に負荷条件値を切り替える場合、エンド動作になりますので、障害物近傍のパス動作中にモードを変更しないようご注意ください。</p>
マクロ定義	<pacman.h>ファイルが必要です。
関連項目	プログラミングマニュアル I の「4.7 「使用条件」における最適可搬質量設定機能」
用例	CALL aspACLD(8500, -50, 100, 80) ' 内部先端負荷条件値を先端負荷質量 8500 (g)、 ' 負荷重心位置 X 成分 -50 (mm)、Y 成分 100 ' (mm)、Z 成分 80 (mm) に設定します。

注意事項

- (1) 先端負荷質量はロボット毎に設定された範囲内の整数で指定してください。それ以外の数値を指定すると、「60d2 先端負荷設定値が許容値を超えました」が発生します。
- (2) 先端負荷重心位置は、ロボット毎に設定された範囲を満たすように入力してください。範囲を満たさない場合、「60d2 先端負荷設定値が許容値を超えました」が発生します。
- (3) 内部負荷条件値は、外部負荷条件値に対し、以下の範囲に設定してください。範囲を満たさない場合、「60d2 先端負荷設定値が許容値を超えました」が発生します。

$$0.5 \times \text{外部負荷条件値} \leq \text{内部負荷条件値} \leq \text{外部負荷条件値}$$

aspChange (ライブラリ)

機能 最適可搬質量設定モードの内部モードを選択します。

書式 aspChange (<モード>)

説明 最適可搬質量設定モードを切り替えます。

<設定値>

- 0 → 無効
- 1 → PTP のみ有効
- 2 → CP のみ有効
- 3 → PTP、CP ともに有効

モード切り替えには、約 0.1 秒の実行時間がかかります。頻繁にモードを切り替えると動作遅れの要因になります。また、パス動作中にモードを切り替える場合、エンド動作になります。障害物近傍のパス動作中にモードを変更しないようにしてください。

マクロ定義 <pacman.h>ファイルが必要です。

関連項目 プログラミングマニュアル I の「4.7 「使用条件」における最適可搬質量設定機能」

用例

CALL aspChange(1) '最適可搬質量設定モードの内部モードを 1 にします。

注意事項 <モード>は 0~3 の整数で指定してください。それ以外の数値を指定すると、「6003 有効な数値範囲を超えた」が発生します。

ndInb (ライブラリ)

機能 指定ポートの入力を 2 進数とみなして 10 進数に変換

書式 ndInb (<整数変数番号>, <最下位桁出力ポート番号>, <最上位桁出力ポート番号>)

説明 従来言語の INB 命令と同等な機能を提供します。
指定した入力ポートの信号の状態を読み取り、それを 2 進数とみなして 10 進数に変換します。
変換した値は整数変数に代入します。

関連項目 [nd0nb](#)、[nd0nbI](#)

備考 PAC 言語では同等の機能を DEFIO と IN 命令で記述できます。
その方法の方が、このライブラリより効率的ですので DEFIO と IN 命令を使ってみてください。
入力ポートは連続した 16 ポート以内としてください。16 ポート以上を指定した場合、処理されません。

用例

CALL ndInb(1, 552, 567) ’入力ポート 552~567 を 16 ビットの 2 進数とみなして
 ’ 10 進数に変換し結果を I[1]に代入します。

ndJf (ライブラリ)

機能 外部機器から OK/NG を受信し、条件分岐 (RS232C 入出力)

書式 ndJf (<2 桁の整数>, <判定引数>)

説明 従来言語の JF 命令と同等な機能を提供します。
指定された 2 桁の整数を外部機器に転送した後、外部機器からの応答結果を判定し、プログラム中で条件分岐を行ないます。
外部機器からの応答が OK ならば次のステップへ進み、NG の場合はラベル先へ進みます。

関連項目 [ndVcom](#)、[_ndVType](#)

用例

```
#include <pacman.h>
PROGRAM PRO1
DEFINT JF_VAL = 0
  FLUSH
  CALL ndVType(1)          ' プロトコルを設定 (旧=0/新=1)
  CALL ndVrst             ' 外部機器の初期化
  CALL ndVset(0)          ' 受信データ (VDT) を 0 にクリア
  CALL ndVis(3)           ' 外部機器を起動 (03 を送信)
  CALL ndJf(3, JF_VAL)    ' 外部機器の応答結果取得 (03 を送信)
  IF JF_VAL = TRUE THEN  ' 応答が OK (TRUE) ならばデータ受信
    CALL ndVset(3)        ' 外部機器より 10 個のデータを受信 (03 を送信)
    CALL ndVdt(pacPOS, 1) ' 外部機器より受信したデータを変数 (P1) に代入
    CALL ndVdt(pacJNT, 1) ' 外部機器より受信したデータを変数 (J1) に代入
    CALL ndVdt(pacTRN, 1) ' 外部機器より受信したデータを変数 (T1) に代入
  END IF
END
```

ndOnb (ライブラリ)

機能	10 進数を 2 進数に変換して指定ポートより出力
書式	ndOnb (<整数値>, <最下位桁出力ポート番号>, <最上位桁出力ポート番号>)
説明	従来言語の ONB 命令と同等な機能を提供します。 整数を 2 進数変換し、指定のポートから出力します。 I 型変数値を出力する形式の ONB 命令は ndOnbI ライブラリを使います。

関連項目 [ndInb](#)、[ndOnbI](#)

備考 出力ポートは連続した 16 ポート以内としてください。16 ポートを超える指定をした場合、下位ポート>上位ポートの場合、出力不可のポートを指定した場合は処理されません。

用例

```
CALL ndOnb(15, 769, 784)      '出力ポート 769~784 を 16 ビットの 2 進数とみなして  
                              ' 10 進数 15 を変換出力します。
```

ndOnbI (ライブラリ)

機能	10 進数を 2 進数に変換して指定ポートより出力
書式	ndOnbI (<整数変数番号>, <最下位桁出力ポート番号>, <最上位桁出力ポート番号>)
説明	従来言語の ONB 命令と同等な機能を提供します。 整数変数値を 2 進数変換し、指定のポートから出力します。 整数値を出力する形式の ONB 命令は ndOnb ライブラリを使います。

関連項目 [ndInb](#)、[ndOnb](#)

備考 出力ポートは連続した 16 ポート以内としてください。16 ポートを超える指定をした場合、下位ポート>上位ポートの場合、出力不可のポートを指定した場合は処理されません。

用例

```
CALL ndOnbI(1, 769, 784)    '出力ポート 769~784 を 16 ビットの 2 進数とみなして、  
                              ' I[1]の値を変換し出力します。
```

ndVdt (ライブラリ)

機能 外部機器から転送された変数を記憶 (RS232C 入出力)

書式 ndVdt (<格納変数型>, <格納変数番号>)

説明 従来言語の VDT 命令と同等な機能を提供します。
外部機器より転送されたデータを指定された格納変数型 (P/J/T) の格納変数番号に代入します。

マクロ定義 <pacman.h>が必要です。

関連項目 [ndVcom](#)、[ndVset](#)

用例

```
#include <pacman.h>
PROGRAM PRO1
DEFINT JF_VAL = 0
  FLUSH
  CALL ndVType(1)      'プロトコルを設定 (旧=0/新=1)
  CALL ndVrst         '外部機器の初期化
  CALL ndVset(0)      '受信データ (VDT) を 0 にクリア
  CALL ndVis(3)       '外部機器を起動 (03 を送信)
  CALL ndJf(3, JF_VAL) '外部機器の応答結果取得 (03 を送信)
  IF JF_VAL = TRUE THEN '応答が OK (TRUE) ならばデータ受信
    CALL ndVset(3)     '外部機器より 10 個のデータを受信 (03 を送信)
    CALL ndVdt(pacPOS, 1) '外部機器より受信したデータを変数 (P1) に代入
    CALL ndVdt(pacJNT, 1) '外部機器より受信したデータを変数 (J1) に代入
    CALL ndVdt(pacTRN, 1) '外部機器より受信したデータを変数 (T1) に代入
  END IF
END
```

ndVis (ライブラリ)

機能 外部機器へ指定した 2 桁の整数を転送 (RS232C 入出力)

書式 ndVis (<2 桁の整数>)

説明 従来言語の VIS 命令と同等な機能を提供します。
外部機器の準備状態を確認した後、ロボットから外部機器へ指定した 2 桁の整数を転送します。

関連項目 [ndVcom](#)、[ndVType](#)

用例

```
#include <pacman.h>
PROGRAM PRO1
DEFINT JF_VAL = 0
  FLUSH
  CALL ndVType(1)          'プロトコルを設定 (旧=0/新=1)
  CALL ndVrst             '外部機器の初期化
  CALL ndVset(0)          '受信データ (VDT) を 0 にクリア
  CALL ndVis(3)           '外部機器を起動 (03 を送信)
  CALL ndJf(3, JF_VAL)    '外部機器の応答結果取得 (03 を送信)
  IF JF_VAL = TRUE THEN  '応答が OK (TRUE) ならばデータ受信
    CALL ndVset(3)        '外部機器より 10 個のデータを受信 (03 を送信)
    CALL ndVdt(pacPOS, 1) '外部機器より受信したデータを変数 (P1) に代入
    CALL ndVdt(pacJNT, 1) '外部機器より受信したデータを変数 (J1) に代入
    CALL ndVdt(pacTRN, 1) '外部機器より受信したデータを変数 (T1) に代入
  END IF
END
```

ndVput (ライブラリ)

機能 外部機器へ位置姿勢を転送 (RS232C 入出力)

書式 ndVput (<格納変数型>, <格納変数番号>)

説明 従来言語の VPUT 命令と同等な機能を提供します。

<位置変数番号>が負の場合 :

ロボットの現在位置座標 (ツール定義をしているときは、ツール座標系の原点位置座標) と姿勢または位置変数の内容を外部機器へ転送します。

<位置変数番号>が正の場合 :

<格納変数型>と<格納変数番号>で指定された位置変数の内容を外部機器へ転送します。

マクロ定義 <pacman. h>が必要です。

関連項目 [ndVcom](#)、[ndVType](#)

用例

```
#include <pacman. h>
PROGRAM PRO65
  FLUSH
  CALL dVType(1)          'プロトコルを設定 (旧=0/新=1)
  CALL ndVrst            '外部機器の初期化
  CALL ndVis(4)          '外部機器を起動(04 を送信)
  P1 = (1, 2, 3, 4, 5, 6, 7)
  J1 = (11, 12, 13, 14, 15, 16)
  T1 = (21, 22, 23, 24, 25, 26, 27, 28, 29, 30)
  CALL ndVput(pacPOS, 1) '変数 "P1" のデータを外部機器に送信
  CALL ndVput(pacJNT, 1) '変数 "J1" のデータを外部機器に送信
  CALL ndVput(pacTRN, 1) '変数 "T1" のデータを外部機器に送信
END
```

ndVrst (ライブラリ)

機能 外部機器の初期化 (RS232C 入出力)

書式 ndVrst

説明 従来言語の VRST 命令と同等な機能を提供します。
外部機器に対して、初期化を指示します。

関連項目 [ndVcom](#)、[ndVType](#)

用例

```
#include <pacman.h>
PROGRAM PRO1
DEFINT JF_VAL = 0
  FLUSH
  CALL ndVType(1)          'プロトコルを設定 (旧=0/新=1)
  CALL ndVrst             '外部機器の初期化
  CALL ndVset(0)          '受信データ (VDT) を 0 にクリア
  CALL ndVis(3)           '外部機器を起動 (03 を送信)
  CALL ndJf(3, JF_VAL)    '外部機器の応答結果取得 (03 を送信)
  IF JF_VAL = TRUE THEN   '応答が OK (TRUE) ならばデータ受信
    CALL ndVset(3)        '外部機器より 10 個のデータを受信 (03 を送信)
    CALL ndVdt(pacPOS, 1) '外部機器より受信したデータを変数 (P1) に代入
    CALL ndVdt(pacJNT, 1) '外部機器より受信したデータを変数 (J1) に代入
    CALL ndVdt(pacTRN, 1) '外部機器より受信したデータを変数 (T1) に代入
  END IF
END
```

ndVset (ライブラリ)

機能 外部機器からデータ受信 (RS232C 入出力)

書式 ndVset (<2桁の整数>)

説明 従来言語の VSET 命令と同等な機能を提供します。

ロボットから外部機器へ指定した 2 桁の整数を転送した後、外部機器よりデータを受け取り、内部変数への加算 (代入) を行いません。<2 桁の整数>を “0” に指定すると内部変数を “0” で初期化します。

関連項目 [ndVcom](#)、[ndVType](#)

用例

```
#include <pacman.h>
PROGRAM PRO1
DEFINT JF_VAL = 0
  FLUSH
  CALL ndVType(1)          'プロトコルを設定 (旧=0/新=1)
  CALL ndVrst             '外部機器の初期化
  CALL ndVset(0)          '受信データ (VDT) を 0 にクリア
  CALL ndVis(3)           '外部機器を起動 (03 を送信)
  CALL ndJf(3, JF_VAL)    '外部機器の応答結果取得 (03 を送信)
  IF JF_VAL = TRUE THEN  '応答が OK (TRUE) ならばデータ受信
    CALL ndVset(3)        '外部機器より 10 個のデータを受信 (03 を送信)
    CALL ndVdt(pacPOS, 1) '外部機器より受信したデータを変数 (P1) に代入
    CALL ndVdt(pacJNT, 1) '外部機器より受信したデータを変数 (J1) に代入
    CALL ndVdt(pacTRN, 1) '外部機器より受信したデータを変数 (T1) に代入
  END IF
END
```

ndVType (ライブラリ)

機能 外部機器との通信プロトコルを指定します。(RS232C 入出力)

書式 ndVType (<2 桁の整数>)

説明 ndVis、ndJf、ndVset、ndVrst、ndVput、ndVcom の通信プロトコルを指定します。各ライブラリの初期値は新プロトコルに設定されています。

(旧プロトコル=0/新プロトコル=1)

関連項目 [ndVis](#)、[ndJf](#)、[ndVset](#)、[ndVrst](#)、[ndVput](#)、[ndVcom](#)

用例

```
#include <pacman.h>
PROGRAM PRO1
DEFINT JF_VAL = 0
  FLUSH
  CALL ndVType(1)          'プロトコルを設定 (旧=0/新=1)
  CALL ndVrst             '外部機器の初期化
  CALL ndVset(0)          '受信データ(VDT)を0にクリア
  CALL ndVis(3)           '外部機器を起動(03を送信)
  CALL ndJf(3、JF_VAL)    '外部機器の応答結果取得(03を送信)
  IF JF_VAL = TRUE THEN  '応答がOK(TRUE)ならばデータ受信
    CALL ndVset(3)        '外部機器より10個のデータを受信(03を送信)
    CALL ndVdt(pacPOS、1)'外部機器より受信したデータを変数(P1)に代入
    CALL ndVdt(pacJNT、1)'外部機器より受信したデータを変数(J1)に代入
    CALL ndVdt(pacTRN、1)'外部機器より受信したデータを変数(T1)に代入
  END IF
END
```

ndVcom (ライブラリ)

機能 外部機器との通信動作 (カーネル)。(RS232C 入出力)

書式 ndVcom (<機能コード>, <配列変数>)

説明 外部機器との通信動作の核プログラムです。通信ライブラリ (ndVis、ndJf、ndVset、ndVrst、ndVput、ndVType) を使用するのに、このライブラリが必要です。コントローラのソフトバージョンが Ver. 1.2* 以前の場合は Ver. 1.2 互換のクラスのものを使用してください。従来言語のクラスのものを使用するとコンパイル異常が発生します。コントローラソフトのバージョン確認方法は、操作ガイド「5.7 [設定 (メイン)] ウィンドウの表示、各モジュールのバージョン情報の表示、[F6 設定]—[F6 保守.]—[F2 バージョン]」を参照してください。

<p>注意：このプログラムは複数タスクで実行されることを想定していません。 もし、複数タスクで実行するのであれば、セマフォなどで同期を取るように変更する必要があります。</p>
--

関連項目 [ndVis](#)、[ndJf](#)、[ndVset](#)、[ndVrst](#)、[ndVput](#)、[ndVdt](#)、[ndVType](#)

ndApra (ライブラリ)

機能 ツール座標系指定の絶対動作（4軸ロボット専用）

書式 ndApra (<基準位置>, <Z座標>, <停止精度>)

説明 従来言語の APRA 命令と同等な機能を提供します。

基準位置の Z 座標のみを指定して P T P 制御で移動します。停止精度の指定により下記のような違いがあります。

0 : エンド動作

1 : パス動作

2 : エンコーダ値による目標位置への到達確認

従来言語の APRT 命令と同等な機能を用いたい場合はこのライブラリを参考に並進偏差計算（プログラム 1 の p. 7-19 「7.9.7 ポジション演算」参照）にて、ポジション型の値を計算し、MOVE 命令を使用してください

関連項目 [ndDepa](#)

用例

```
PROGRAM PRO1
  TAKEARM
  CALL ndApra((P0), 400.0, 1) ' P 0 の Z 座標が 4 0 0 の座標に移動
  MOVE P, @0 P0 ' P 0 へ移動
  CALL ndDepa(400.0, 0) ' P 0 の Z 座標が 4 0 0 の座標に移動
  GIVEARM
END
```

ndDepa (ライブラリ)

機能 ツール座標系指定の絶対動作（4軸ロボット専用）

書式 ndDepa (<Z座標>, <停止精度>)

説明 従来言語の DEPA 命令と同等な機能を提供します。
現在位置から Z 座標のみを指定して PTP 制御で移動します。停止精度の違いにより下記のような違いがあります。

- 0 : エンド動作
- 1 : パス動作
- 2 : エンコーダ値による目標位置への到達確認

従来言語の DRET 命令と同等な機能を用いたい場合はこのライブラリを参考に並進偏差計算（プログラム 1 の p. 7-19 「7.9.7 ポジション演算」参照）にて、ポジション型の値を計算し、MOVE 命令を使用してください

関連項目 [ndApra](#)

用例

```
PROGRAM PRO1
  TAKEARM
  CALL ndApra((P0), 400.0, 1) ' P 0 の Z 座標が 4 0 0 の座標に移動
  MOVE P, @0 P0 ' P 0 へ移動
  CALL ndDepa(400.0, 0) ' P 0 の Z 座標が 4 0 0 の座標に移動
  GIVEARM
END
```

ndTc (ライブラリ) [Ver. 1.2 以降]

機能	T C 時間設定
書式	ndTc (<T C 時間>)
説明	従来言語の T C 命令と同等な機能を提供します。 T C 時間を設定します。 0 秒～6 0 0 秒の間で設定できます。工場出荷状態時点では 6 0 秒に設定されています。 0 秒を設定すると従来言語の TC OFF と同等の機能を提供します
マクロ定義	<pacman. h>ファイルが必要です。
関連項目	ndTS

ndTs (ライブラリ) [Ver. 1.2 以降]

機能	T S 時間、スロー速度設定
書式	ndTS (<T S 時間>, <スロー速度>)
説明	従来言語の T S 命令と同等な機能を提供します。 T S 時間、スロー速度を設定します。 T S 時間は 3 秒～3 0 秒の間で設定できます。工場出荷状態時点では 5 秒に設定されています。 スロー速度は 1 %～1 0 %の間で設定できます。工場出荷時点では 1 0 %に設定されています。
マクロ定義	<pacman. h>ファイルが必要です。
関連項目	ndTC

4 パレタイジングのライブラリ

パレタイジング機能のライブラリを以下に説明します。

pltDecCnt (ライブラリ)

機能 パレタイジング総カウンタのデクリメント

書式 pltDecCnt (<パレタイジング番号>)

説明 指定されたパレタイジング番号のパレタイジング総カウンタを現在値から-1 します。
パレタイジングカウンタを操作することで、千鳥型や歯抜け型のパレタイジングを作
ることができます。

関連項目 [pltGetCnt](#)、[pltIncCnt](#)、[pltKernel](#)、[pltLetCnt](#)

用例

CALL pltDecCnt(1) 'パレタイジング番号 1 番の総カウンタから-1 します。

pltGetCnt (ライブラリ)

機能 パレタイジング総カウンタの取得

書式 pltGetCnt (<パレタイジング番号>, <Cnt 保存整数変数番号>)

説明 指定されたパレタイジング番号のパレタイジング総カウンタを取得します。

関連項目 [pltDecCnt](#)、[pltIncCnt](#)、[pltKernel](#)、[pltLetCnt](#)

用例

CALL pltGetCnt(1, 0) '現在のカウンタを I[0] に代入します。
CALL pltLetCnt(1, I[0] - 1) 'カウンタを 1 つ戻します。

pltGetK (ライブラリ)

機能 パレタイジング設定値 K の取得

書式 pltGetK (<パレタイジング番号>, <K 保存整数変数番号>)

説明 指定されたパレタイジング番号の設定値 K を取得します。

関連項目 [pltGetK1](#)、[pltLetK1](#)、[pltKernel](#)

用例

CALL pltGetK(1, 10) 'パレタイジング番号 1 番のパレタイジング設定値 K を
' I[10]へ代入します。

pltGetK1 (ライブラリ)

機能 パレタイジングカウンタ K1 の取得

書式 pltGetK1 (<パレタイジング番号>, <K1 保存整数変数番号>)

説明 指定されたパレタイジング番号のカウンタ K1 を取得します。

関連項目 [pltGetK](#)、[pltLetK1](#)、[pltKernel](#)

用例

CALL pltGetK1(1, 10) 'パレタイジング番号 1 番のパレタイジングカウンタ K1
' を I[10]へ代入します。

pltGetM (ライブラリ)

機能 パレタイジング設定値 M の取得

書式 pltGetM (<パレタイジング番号>, <M 保存整数変数番号>)

説明 指定されたパレタイジング番号の設定値 M を取得します。

関連項目 [pltGetM1](#)、[pltLetM1](#)、[pltKernel](#)

用例

CALL pltGetM(1, 10) 'パレタイジング番号 1 番のパレタイジング設定 M を
' I[10]へ代入します。

pltGetM1 (ライブラリ)

機能 パレタイジングカウンタ M1 の取得

書式 pltGetM1 (<パレタイジング番号>, <M1 保存整数変数番号>)

説明 指定されたパレタイジング番号のカウンタ M1 を取得します。

関連項目 [pltGetM](#)、[pltLetM1](#)、[pltKernel](#)

用例

CALL pltGetM1(1, 10) 'パレタイジング番号1番のパレタイジングカウンタを
' I[10]へ代入します。

pltGetN (ライブラリ)

機能 パレタイジング設定値 N の取得

書式 pltGetN (<パレタイジング番号>, <N 保存整数変数番号>)

説明 指定されたパレタイジング番号の設定値 N を取得します。

関連項目 [pltGetN1](#)、[pltLetN1](#)、[pltKernel](#)

用例

CALL pltGetN(1, 10) 'パレタイジング番号1番のパレタイジング設定値 N を
' I[10]に代入します。

pltGetN1 (ライブラリ)

機能 パレタイジングカウンタ N1 の取得

書式 pltGetN1 (<パレタイジング番号>, <N1 保存整数変数番号>)

説明 指定されたパレタイジング番号のカウンタ N1 を取得します。

関連項目 [pltGetN](#)、[pltLetN1](#)、[pltKernel](#)

用例

CALL pltGetN1(1, 10) 'パレタイジング番号1番のパレタイジングカウンタ N1
' を I[10]に代入します。

pltGetNextPos (ライブラリ)

機能 次ポジションの取得

書式 pltGetNextPos (<パレタイジング番号>, <次位置の取得ポジション型変数番号>)

説明 次ポジションを取得します。また、このライブラリを実行した時点でパレタイジングカウンタはインクリメントされます。

したがって、取得だけして動作しなかった場合などは、pltDecCnt 等でデクリメントする必要があります。

関連項目 [pltMove](#)、[pltKernel](#)

用例

```
CALL pltGetNextPos(1, 10)      'パレタイジング番号 1 番の次位置を I[10]に代入しま  
                               'す。
```

pltGetPLT1END (ライブラリ)

機能 パレタイジング 1 段終了フラグの取得

書式 pltGetPLT1END (<パレタイジング番号>, <1 段終了フラグ保存整数番号>)

説明 パレタイジング 1 段終了フラグを I[<1 段終了フラグ保存整数番号>]へ格納します。

関連項目 [pltGetPLTEND](#)、[pltResetPLT1END](#)、[pltKernel](#)

用例

```
CALL pltGetPLT1END(1, 10)     'パレタイジング番号 1 番の 1 段階終了フラグを I[10]  
                               'に代入します。
```

pltGetPLTEND (ライブラリ)

機能 パレタイジング全段終了フラグの取得

書式 pltGetPLTEND (<パレタイジング番号>, <全段終了フラグ保存整数番号>)

説明 パレタイジング全段終了フラグを I[<全段終了フラグ保存整数番号>]へ格納します。

関連項目 [pltGetPLT1END](#)、[pltResetPLTEND](#)、[pltKernel](#)

用例

```
CALL pltGetPLTEND(1, 10)     'パレタイジング番号 1 番の全段終了フラグを I[10]に  
                               '代入します。
```

pltIncCnt (ライブラリ)

機能 パレタイジング総カウンタのインクリメント

書式 pltIncCnt (<パレタイジング番号>)

説明 指定されたパレタイジング番号のパレタイジング総カウンタを現在値から+1 します。
パレタイジングカウンタを操作することで、千鳥型や歯抜け型のパレタイジングを作
ることができます。

関連項目 [pltDecCnt](#)、[pltGetCnt](#)、[pltKernel](#)、[pltLetCnt](#)

用例

CALL pltIncCnt(1) 'パレタイジング番号 1 番のパレタイジング総カウンタ
 'を+1 します。

pltInit1 (ライブラリ)

機能 パレタイジング初期化テンプレート 1

書式 pltInit1

説明 これは pltInitialize の使用例です。
引数の意味は pltInitialize の引数情報を参照してください。

関連項目 [pltInitialize](#)

pltInitialize (ライブラリ)

機能 パレタイジング初期化

書式 pltInitialize (<パレタイジング番号>, <横分割数>, <縦分割数>, <段積数>, <アプローチ長>, <デパート長>, <パレット高さ>, <パレット 4 隅 1P 型変数番号>, <パレット 4 隅 2P 型変数番号>, <パレット 4 隅 3P 型変数番号>, <パレット 4 隅 4P 型変数番号>)

説明 パレタイジング動作を定義する初期化プログラムです。
パレタイジングカウンタのみの初期化の場合はpltResetAll ライブラリを使った方が効率的です。

注意 : pltInitializeを実行しないと、パレタイジング動作を実行することができません。

マクロ定義 mcApprVal : アプローチ長を代入する F 型変数番号
 mcDepVal : デパート長を代入する F 型変数番号
 mcPltH : パレット高さを代入する F 型変数番号

注意 : これらの定義がユーザ使用の変数番号と競合する場合はマクロ定義し直してください。

関連項目 [pltKernel](#)

用例

```
CALL pltInitialize(0, 4, 3, 1, 50, 50, 50, 52, 53, 54, 55)
' パレタイジング番号:0、横分割数: 4、縦分割数:3、段積数:1
' アプローチ長:50mm、デパート長:50mm、パレット高さ:50mm
' パレット 4 隅 1:P[52]、2:P[53]、3:P[54]、4:P[55]で初期
' 化します。
```

pltKernel (ライブラリ)

機能 パレタイジング動作(カーネル)

書式 pltKernel (<パレタイジング番号>, <アクション番号>, <アクション引数>, <エラー番号>)

説明 パレタイジング動作の核プログラムです。その他のすべてのパレタイジングライブラリを使用するのに、このライブラリが必要です。

注意 : このプログラムは複数タスクで実行されることを想定していません。もし、複数タスクで実行するのであれば、セマフォなどで同期を取るように変更する必要があります。

マクロ定義 mcPaltMax : 最大パレット数
このマクロを定義し直すだけで簡単にパレット数を増やせます。

注意 : パレット数を増やすとその分ローカル変数領域が消費されます。

関連項目 [pltInitialize](#)

備考

- 基本的に、このライブラリは他のライブラリから呼ばれるだけで、ユーザが直接呼ぶことはありません。
- 直接使用する場合は、プログラムの内容を理解した上で使用してください。

pltLetM1 (ライブラリ)

機能 パレタイジングカウンタ M1 の設定

書式 pltLetM1 (<パレタイジング番号>, <M1 設定値>)

説明 指定されたパレタイジング番号のカウンタ M1 を設定できます。
パレタイジングカウンタを操作することで、千鳥型や歯抜け型のパレタイジングを作ることができます。

関連項目 [pltGetM1](#)、[pltKernel](#)

用例

CALL pltLetM1(1, 4) 'パレタイジング番号1番のパレタイジングカウンタ M1
'を 4 に設定します。

pltLetN1 (ライブラリ)

機能 パレタイジングカウンタ N1 の設定

書式 pltLetN1 (<パレタイジング番号>, <N1 設定値>)

説明 指定されたパレタイジング番号のカウンタ N1 を設定できます。
パレタイジングカウンタを操作することで、千鳥型や歯抜け型のパレタイジングを作ることができます。

関連項目 [pltGetN1](#)、[pltKernel](#)

用例

CALL pltLetN1(1, 3) 'パレタイジング番号1番のパレタイジングカウンタ N1
'を 3 に設定します。

pltMain1 (ライブラリ)

機能	パレタイジング テンプレート 1
書式	pltMain1
説明	パレットから取り出し、組み付け位置へ移動する基本作業です。
マクロ定義	pltIndex : パレタイジング番号
関連項目	pltMain2
備考	必要に応じ修正して使用して下さい。

pltMain2 (ライブラリ)

機能	パレタイジング テンプレート 2
書式	pltMain2
説明	パレットから取り出し、組み付け位置へ移動する基本作業です。 取り出す前に前回の取り出しが正常に行なわれたか検査します。
マクロ定義	pltIndex : パレタイジング番号 ChuckNG : 取り出し NG 信号 (DIO 番号)
関連項目	pltMain1
備考	必要に応じ修正して使用して下さい。

pltMove0 (ライブラリ)

機能 標準パレタイジング動作 1

書式 pltMove0

説明 pltMove(Index)の Index=0 に特化したプログラムです。
パレットごとに動作を変更したい場合はこのプログラムを追加し、#DEFINE pltIndex 0 の行をそのパレット番号に変更し、プログラム名を変更してください。
コントローラのソフトバージョンが Ver. 1.2*以前の場合は Ver. 1.2 互換のクラスのものを使用してください。パレタイジングのクラスのものを使用するとコンパイル異常が発生します。
コントローラソフトのバージョン確認方法は操作ガイド、操作ガイド「5.7 [設定 (メイン)] ウィンドウの表示、各モジュールのバージョン情報の表示、[F6 設定]—[F6 保守.]—[F2 バージョン]」を参照してください。

マクロ定義 mcNextPos : 次ポジションを格納する P 型変数番号
mcApprLen : アプローチ長を格納する F 型変数番号
mcDepLen : デパート長を格納する F 型変数番号

注意：これらの定義がユーザ使用の番号と競合する場合はマクロ定義し直してください。

pltIndex : パレット番号

関連項目 [pltInitialize](#)、[pltMove](#)、[pltKernel](#)

備考 次ポジションを取得した時点でパレタイジングカウンタはインクリメントされます。
したがって、取得だけして動作しなかった場合などは、pltDecCntなどでデクリメントする必要があります。

用例 CALL pltMove0 'パレタイジング番号 0 番のパレタイジング動作を実行
' します。

pltResetAll (ライブラリ)

機能 パレタイジングカウンタすべてのリセット

書式 pltResetAll (<パレタイジング番号>)

説明 パレタイジングカウンタすべてをリセットします。
pltLetCnt(<パレタイジング番号>, 0) とほぼ同等の意味を持ちますが、このライブラリは段の終了フラグもリセットします。
パレタイジング動作の再定義が必要ではなく、かつカウンタだけをリセットするのであれば、pltInitialize ではなくこのライブラリを使います。

関連項目 [pltInitialize](#)、[pltResetPLTEND](#)、[pltKernel](#)

用例

```
CALL pltResetAll(1)           'パレタイジング番号1番のパレタイジングカウンタを  
                              '全てリセットします。
```

pltResetPLT1END (ライブラリ)

機能 パレタイジング1段終了フラグのリセット

書式 pltResetPLT1END (<パレタイジング番号>)

説明 パレタイジング1段終了フラグをリセットします。

関連項目 [pltGetPLT1END](#)、[pltResetPLTEND](#)、[pltKernel](#)

用例

```
CALL pltResetPLT1END(1)       'パレタイジング番号1番の1段終了フラグをリセット  
                              'します。
```

pltResetPLTEND (ライブラリ)

機能 パレタイジング全段終了フラグのリセット

書式 pltResetPLTEND (<パレタイジング番号>)

説明 パレタイジング全段終了フラグをリセットします。

関連項目 [pltGetPLTEND](#)、[pltResetPLT1END](#)、[pltKernel](#)

用例

```
CALL pltResetPLTEND (1)       'パレタイジング番号1番の全段終了フラグをリセット  
                              'します。
```

5 ツール操作のライブラリ

ツール操作関連機能のライブラリを以下に説明します。

tolChange (ライブラリ)

機能	ツールチェンジ
書式	tolChange (<ツール装着番号>, <ツール番号>)
説明	このライブラリはtolInitializeで定義された標準的なツール切り替え動作を実行します。
関連項目	tolInitialize 、 tolKernel
備考	PACのツール座標系切り替え命令(ChangeTool)は、ツール座標系を切り替えるだけで実際のツールを付け替える命令ではありません。
用例	CALL tolChange(1,2) 'ツール装着番号1に2番のツールを割り当てます。

tolInit1 (ライブラリ)

機能	ツールチェンジ初期化テンプレート1
書式	tolInit1
説明	これはtolInitializeの使用例です。 引数の意味はtolInitializeの引数情報を参照してください。
関連項目	tolInitialize 、 tolKernel

tolInitialize (ライブラリ)

機能 ツールチェンジ初期化

書式 tolInitialize (<ツール装着番号>, <ツール番号>, <チャック点P型変数番号>, <アプローチ長>, <チャック DIO 番号>, <アンチャック DIO 番号>)

説明 標準的なツールチェンジ動作を定義します。

関連項目 [tolInit1](#)、[tolKernel](#)

用例

```
CALL tolInitialize(0, 1, 50, 50, 40, 41)' ツール装着番号:0、ツール番号:1  
'チャック点:P [50]、アプローチ長:50mm  
'チャック:I0[40]、アンチャック:I0[41]
```

tolKernel (ライブラリ)

機能 ツールチェンジ動作(カーネル)

書式 tolKernel (<ツール装着番号>, <ツール番号>, <アクション番号>, <アクション引数>)

説明 ツールチェンジの核プログラムです。その他のすべてのツールチェンジライブラリを使用するのに、このライブラリが必要です。

マクロ定義 mcToolMax : ツールの最大数
mcTools : 多肢ハンドの場合の最大装着可能数

関連項目 [tolInitialize](#)

tolMain1 (ライブラリ)

機能 ツールチェンジ テンプレート 1

書式 tolMain1

説明 ツールチェンジの使用例です。

ioINB から 8 ビットでツール番号が指定されたとき、その指定されたツールに切り替えます。

マクロ定義 ioINB : ツール番号指定I0の先頭番号
関連項目 [tolChange](#)、[tolInitialize](#)、[tolKernel](#)

6 入出力のライブラリ

DIO、RS232C入出力関連機能のライブラリです。

dioSync (ライブラリ)

機能	DIO に接続された外部機器 (シーケンサなど) との同期
書式	dioSync (<データ受信可能信号番号>, <データ受信完了信号番号>)
説明	DIO に接続された外部機器 (シーケンサなど) と同期を取る標準的な手続きです。
用例	CALL dioSync (220. 221) 'データ受信可能信号を IO[220]、データ受信完了信号 'を IO[221]に割り当てて同期を取ります。

7 アーム動作のライブラリ

アーム動作関連機能のライブラリを以下に説明します。

mvResetPulseWidth (ライブラリ)

機能	停止時許容パルス幅をデフォルト値にします。
書式	mvResetPulseWidth
説明	停止時許容パルス幅の設定値をデフォルト値の 20 に全軸設定します。 停止時許容パルス幅とは、エンコーダ値確認動作 (@E指定) 時に停止とみなす誤差パルスです。
マクロ定義	<pacman.h>が必要です。
用例	CALL mvResetPulseWidth '停止時許容パルス幅をデフォルト値にします。

mvResetPulseWidthJnt (ライブラリ) [Ver. 1.7 以降]

機能	指定した付加軸の、停止時許容パルス幅をデフォルト値に戻します。
書式	mvResetPulseWidthJnt(<軸番号>)
説明	<軸番号>で指定した付加軸の停止時許容パルス幅をデフォルト値の 20 に戻します。 停止時許容パルス幅とは、エンコーダ値確認動作 (@E指定) 時に停止とみなす誤差パルスです。
マクロ定義	<pacman.h>が必要です。
関連項目	mvResetPulseWidth 、 mvSetPulseWidthJnt
注意事項	mvResetPulseWidthJnt は付加軸専用です。ロボット軸には mvResetPulseWidth を使用してください。
用例	CALL mvResetPulseWidthJnt(7) '7 軸の停止時許容パルス幅をデフォルト値に戻します。

mvResetTimeOut (ライブラリ)

機能 動作終了タイムアウト値をデフォルト値にします。

書式 mvResetTimeOut

説明 動作終了タイムアウトの設定値をデフォルト値の 5600 (ms) (5.6 秒) に設定します。
動作終了タイムアウトとは、エンコーダ値確認動作 (@E指定) 時に停止時許容パルス幅に入るまでの時間のリミット値であり、リミット値を超えてもパルス幅に入らなかった場合、エラー「6651 チェック命令タイムオーバ」となります。

マクロ定義 <pacman.h>が必要です。

用例

CALL mvResetTimeOut '動作終了タイムアウト値をデフォルト値にします。'

mvReverseFlip (ライブラリ)

機能 4 軸形態反転

書式 mvReverseFlip (<基準ポジション>, <非検査ポジション>, <判定基準値>)

説明 <基準ポジション>と<非検査ポジション>の 4 軸値を比較して<判定基準値>を超えていたら FLIP/NONFLIP を反転します。

用例

CALL mvReverseFlip(P0,P1,90) '基準位置 P[0]と非検査位置 P[1]を比較し、絶対値で 90 を超えた場合形態を反転します。'

mvSetPulseWidth (ライブラリ)

機能	停止時許容パルス幅設定
書式	mvSetPulseWidth (<停止時許容パルス幅 J1>, <停止時許容パルス幅 J2>, <停止時許容パルス幅 J3>, <停止時許容パルス幅 J4>, <停止時許容パルス幅 J5>, <停止時許容パルス幅 J6>)
説明	停止時許容パルス幅を設定します。 停止時許容パルス幅とは、エンコーダ値確定動作 (@E 指定) 時に停止とみなす誤差パルスです。
マクロ定義	<pacman.h>が必要です。
用例	CALL mvSetPulseWidth (10, 10, 10, 10, 10, 10) ' 停止時許容パルス幅を全軸 10 パルスにします。
注意事項	パルス幅は 1 以上を設定してください。0 を指定した場合、デフォルト値(20)になり、負の数を指定した場合、エラー「6003 有効な数値範囲を超えた」が発生します。また、小さい値を設定すると、エラー「6651 チェック命令タイムオーバ」が発生する場合があります。

mvSetPulseWidthJnt (ライブラリ) [Ver. 1.7 以降]

機能	指定した付加軸の、停止時許容パルス幅を設定します。
書式	mvSetPulseWidthJnt (<軸番号>, <停止時許容パルス幅>)
説明	<軸番号> で指定した付加軸の停止時許容パルス幅を<停止時許容パルス幅>に設定します。停止時許容パルス幅とは、エンコーダ値確認動作 (@E 指定) 時に停止とみなす誤差パルスです。
マクロ定義	<pacman.h>が必要です。
関連項目	mvResetPulseWidthJnt 、 mvSetPulseWidth
注意事項	mvSetPulseWidthJnt は付加軸専用です。ロボット軸には mvSetPulseWidth を使用してください。
用例	CALL mvSetPulseWidthJnt (7, 10) ' 7 軸の停止時許容パルス幅を 10 値にします。

mvSetTimeout (ライブラリ)

機能 動作終了タイムアウト値設定

書式 mvSetTimeout (<動作終了タイムアウト>)

説明 動作終了タイムアウト値を設定します。(単位: ms)

動作終了タイムアウトとは、エンコーダ値確認動作 (@E指定) 時に停止時許容パルス幅に入るまでの時間のリミット値であり、リミット値を超えてもパルス幅に入らなかった場合、エラー「6651 チェック命令タイムオーバ」となります。

マクロ定義 <pacman.h>が必要です。

用例

CALL mvSetTimeout (3000) 'タイムアウト時間を3秒に設定します。'

注意事項 タイムアウト値は1以上を設定してください。0を指定した場合、デフォルト値(5600)になり、負の数を指定した場合、エラー「6003 有効な数値範囲を超えた」が発生します。また、小さい値を設定すると、エラー「6651 チェック命令タイムオーバ」が発生する場合があります。

SetGravity (ライブラリ) [Ver. 1.2 以降]

機能	各関節の静荷重（重力トルク）を補正し、重力バランスを設定します。						
書式	SetGravity						
説明	ロボットの各関節は、重力により、下向きの静荷重（重力トルク）を受けます。重力トルクは、ツールやワークの質量、重心位置、ロボットの姿勢により変化します。電流制限を設定し、モータのトルクを制限した場合、制限トルクが重力トルク以下になれば重力方向に動作してしまいます。その為、制限トルクから重力トルク分を補正し、電流制限設定時でも重力方向に動作しない様に重力バランスを設定する機能です。						
マクロ定義	<pacman.h>ファイルが必要です。						
関連事項	SetCurLmt, ResetGravity, SetGrvOffset						
注意事項	<ol style="list-style-type: none">(1) ロボット制御権を取得(TAKEARM)したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。(2) 先端負荷質量と負荷重心位置を正確に設定して下さい。正確に設定されていない状態で、電流制限設定値を小さく(30 以下程度)設定すると重力落下する場合がありますのでご注意ください。先端負荷質量と負荷重心位置設定は、プログラム 1 の p. 4-15「4.7「使用条件」における最適可搬質量設定機能」を参照ください。(3) 先端負荷質量と負荷重心位置が正確に分からない場合、正確に設定しても電流制限設定時に重力落下する場合は、重力補償補正機能(SetGrvOffset)を使用し、重力補償値の補正する事が可能です。重力補償補正ライブラリを参照ください。(4) 使用条件の重力補償有効無効設定をペンダントで1に設定した場合、重力補償が有効になります。ペンダントで有効設定した場合、コントローラ電源を立ち上げ、キャリブレーション実行直後から、重力補償が有効になります。						
用例	<table><tr><td>CALL SetGravity</td><td>: 重力補償を有効。</td></tr><tr><td>Delay 100</td><td>: 重力補償が反映されるのを待つ。</td></tr><tr><td>CALL SetCurLmt (2, 30)</td><td>: 2 軸の電流制限値を 30%に設定する。</td></tr></table>	CALL SetGravity	: 重力補償を有効。	Delay 100	: 重力補償が反映されるのを待つ。	CALL SetCurLmt (2, 30)	: 2 軸の電流制限値を 30%に設定する。
CALL SetGravity	: 重力補償を有効。						
Delay 100	: 重力補償が反映されるのを待つ。						
CALL SetCurLmt (2, 30)	: 2 軸の電流制限値を 30%に設定する。						

ResetGravity (ライブラリ) [Ver. 1.2 以降]

機能 重力バランスを無効にします。

書式 ResetGravity

説明 重力バランスを無効にします。

マクロ定義 <pacman.h>ファイルが必要です。

関連事項 ResetCurLmt, SetEralw

注意事項

- (1) ロボット制御権を取得(TAKEARM)したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。
- (2) 電流制限中に命令実行した場合、エラー「665b 重力補償を無効にできません」が発生します。電流制限を解除した後、再度実行してください。
- (3) 使用条件の重力補償有効無効設定をペンダントで 0 に設定した場合でも、重力補償が無効になります。ただし、(2)同様、電流制限中は、設定できません。

用例

CALL ResetGravity

SetGrvOffset (ライブラリ) [Ver. 1.2 以降]

機能 各関節の重力トルクより重力補償値を補正します。

書式 SetGrvOffset

説明 ロボットの各関節は、重力により、下向きの静荷重（重力トルク）を受けます。重力補償ライブラリ (SetGravity) により重力バランスを設定できますが、先端負荷質量設定値と実際の負荷質量とのずれ等により、重力バランスにずれが生じる場合があります。本機能は、ロボットが停止した状態で重力トルクを推定し、重力補償値を補正します。

マクロ定義 <pacman.h>ファイルが必要です。

関連事項 SetCurLmt, SetGravity, ResetGrvOffset

- 注意事項**
- (1) ロボット制御権を取得 (TAKEARM) したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。
 - (2) モータ電源が ON でロボットが停止した状態で実行してください。モータ電源 OFF 状態で実行した場合、エラー「6006 モータ電源がオフです」が発生します。ロボットが動作中に実行した場合、エラー「600B ロボット動作中です」が発生します。
 - (3) 重力補償補正を実行した姿勢から大きく変化した場合は、補正にずれが生じます。その場合は再度補正を実施して下さい。
 - (4) 使用条件の電流制限リセット設定値が 1、3、5、7 以外の場合は、モータ電源投入時に補正值が 0 にリセットされます。

用例

```
CALL SetGrvOffset
```

ResetGrvOffset (ライブラリ) [Ver. 1.2 以降]

機能	重力補償値の補正を無効にします。
書式	ResetGrvOffset
説明	重力補償値の補正を無効にします。
マクロ定義	<pacman.h>ファイルが必要です。
関連事項	SetGrvOffset
注意事項	(1) ロボット制御権を取得(TAKEARM)したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。 (2) ロボットが停止した状態で実行してください。ロボットが動作中に実行した場合、エラー「600B ロボット動作中です」が発生します。モータ電源 OFF 中でも実行可能です。
用例	CALL ResetGrvOffset

SetCurLmt (ライブラリ) [Ver. 1.2 以降]

機能	指定した軸のモータ電流値を制限します。
書式	SetCurLmt (<軸番号>、<設定値>)
説明	<軸番号>で指定した軸のモータ電流値（トルク）を設定値に制限します。挿入作業、突き当て作業時にワークに加わる力を制限したい場合に使用します。 <設定値>は、モータ定格電流値が 100 になります。各軸の許容値を超える設定をした場合は、許容値に制限されます。 1 以上の値を設定してください。0 以下の値を設定した場合は、エラー6003「有効な数値範囲を越えました」が発生します。
マクロ定義	<pacman.h>ファイルが必要です。
関連事項	ResetCurLmt, SetGravity, SetGrvOffset, SetEralw

注意事項

- (1) 電流制限設定中は、モータ電流が制限される為、最高速度、最高加速度で動作できません。電流制限は、必要なステップのみ使用してください。また、電流制限を使用する際は、加速度を下げてください。
- (2) 電流制限にて推力を制限しても高速でワークが衝突すると、ワークとハンドと軸の慣性で衝撃力が発生します。電流制限はワークが接触する直前から SET し、かつ速度を下げてください。
- (3) 電流制限は、ロボット停止状態で SET してください。パス動作中に SET すると、エラーが発生する事があります。
- (4) ロボット制御権を取得(TAKEARM)したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。
- (5) 使用条件の電流制限リセット設定値が 1、3、5、7 以外の場合は、モータ電源投入時に電流制限がリセットされます。
モータ電源投入直後から電流制限を有効にする場合は、電流制限リセット設定値の最下位ビットを 1 に設定してください。

6 軸

- (6) 電流制限設定する際は、必ず重力補償機能を有効にしてください。重力補償機能が無効の状態では電流制限を設定すると、エラー「665a 電流制限設定できません」が発生します。重力補償は、SetGravity を参照ください。

6 軸

- (7) 先端負荷質量と負荷重心位置を正確に設定して下さい。正確に設定されていない状態で、設定値を小さく(30 以下程度)設定すると重力落下する場合がありますのでご注意ください。先端負荷質量と負荷重心位置設定は、p. 4-15「4.7「使用条件」における最適可搬質量設定機能」を参照ください。

6 軸

- (8) 電流制限リセット設定値の最下位ビットが 1 に設定されている場合、モータ電源投入直後に重力落下する場合があります。モータ電源 OFF 状態で電流制限を RESET (ResetCurLmt を実行) し、モータ電源を入れてください。

6 軸

- (9) 先端負荷質量と負荷重心位置が正確に分からない場合、正確に設定しても重力落下する場合は、重力補償補正機能(SetGrvOffset)を使用し、重力補償値の補正を行ってください。

4 軸、HS-E

- (10) エアバランスのない 4 軸ロボットの Z 軸 T 軸に電流制限を設定する場合、制限値を小さくすると Z 軸が落下したり、T 軸が回転する場合があります。Z 軸 T 軸重力補償値設定機能(st_SetZBalance)を実行した後電流制限を実行してください。

用例

6 軸

- | | |
|-----------------------|-------------------------|
| CALL SetGravity | ' 重力補償を有効にする。 |
| CALL SetGrvOffset | ' 重力補償値を補正する。 |
| CALL SetEralw(2, 20) | ' 2 軸の偏差許容値を 20 度に設定する。 |
| CALL SetCurLmt(2, 30) | ' 2 軸の電流制限値を 30%に設定する。 |

4 軸

- | | |
|-----------------------|-------------------------|
| CALL SetEralw(2, 20) | ' 2 軸の偏差許容値を 20 度に設定する。 |
| CALL SetCurLmt(2, 30) | ' 2 軸の電流制限値を 30%に設定する。 |

4 軸、HS-E

- | | |
|-----------------------|---------------------------|
| st_SetZBalance | ' Z 軸重力補償値を設定する。 |
| CALL SetEralw(3, 100) | ' Z 軸の偏差許容値を 100mm に設定する。 |
| CALL SetEralw(4, 30) | ' T 軸の偏差許容値を 30 度に設定する。 |
| CALL SetCurLmt(3, 10) | ' Z 軸の電流制限値を 10%に設定する。 |
| CALL SetCurLmt(4, 10) | ' T 軸の電流制限値を 10%に設定する。 |

ResetCurLmt (ライブラリ) [Ver. 1.2 以降]

機能 指定した軸のモータ電流制限を解除します。

書式 ResetCurLmt (<軸番号>)

説明 <軸番号>で指定した軸のモータ電流制限を解除します。解除した後、電流制限値と偏差許容値はデフォルト値に戻ります。

[Ver 1.4 以前] 軸番号に 0 を指定すると、全軸の電流制限を解除します。

[Ver 1.5 以降] 軸番号に 0 を指定すると、ResetCurLmt 実行タスクのセマフォ取得中の軸の全部の電流制限を解除します。

マクロ定義 <pacman.h>ファイルが必要です。

関連項目 [SetCurLmt](#)、[ResetEralw](#)

注意事項 (1) 電流制限解除時に偏差除去処理を実施します。外力にて角度偏差がある場合、設定速度、加速度により偏差除去処理時間が変化します。偏差除去処理時間を短縮するには、速度、加速度を高く設定してください。

(2) モータ電源 OFF 状態でも実行可能です。モータ電源 OFF 状態で、電流制限を解除する場合は、アームセマフォ取得中のタスクを終了した後、以下のプログラムを実行してください。

```
PRO999  
TAKEARM  
CALL ResetCurLmt(0)  
END
```

(3) [Ver. 1.4 以前] ロボット制御権を取得(TAKEARM)したタスクにて実行してください。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。

[Ver. 1.5 以降] 制御権を取得(TAKEARM)したタスクにて実行してください。制御権未取得の軸を指定した場合は、エラー「27D* *軸セマフォを取得できません」が発生します。

用例

```
CALL ResetCurLmt(0)           ’ 全軸の電流制限を解除  
CALL ResetGrvOffset           ’ 重力補償補正值をリセット
```

SetEralw (ライブラリ) [Ver. 1.2 以降]

機能 指定した軸の偏差許容値を変更します。

書式 SetEralw(<軸番号>、<設定値>)

説明 <軸番号>で指定した軸の偏差許容値を設定します。電流制限時に外力にて角度偏差が生じる場合に使用します。

<設定値>は、回転関節の場合はアーム関節角度(度)で、直動関節の場合はアーム関節変位(mm)で指定します。

偏差許容値とは、安全の為サーボ偏差が一定値以上になると発生するエラー「611* J* 偏差過大」の許容範囲です。電流制限を使用して組立てを行う場合、外力によりサーボ偏差が発生し、このエラーが発生する事があります。それを回避するため、一時的に偏差許容値を大きくする場合に使用します。また、電流制限調整中、重力落下を素早く検出するため、偏差許容値を小さく設定する場合にも使用します。

マクロ定義 <pacman.h>ファイルが必要です。

関連事項 SetCurLmt, ResetEralw

注意事項 (1) ロボット制御権を取得(TAKEARM)したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。

用例

CALL SetEralw(2, 20) ; 2軸の偏差許容値を20度に設定する。

ResetEralw (ライブラリ) [Ver. 1.2以降]

機能	指定した軸の偏差許容値をデフォルト値に戻します。
書式	ResetEralw(<軸番号>)
説明	<p><軸番号>で指定した軸の偏差許容値をデフォルト値に戻します。</p> <p>[Ver. 1.4以前] 軸番号に0を指定すると、全軸の偏差許容値をデフォルト値に戻します。</p> <p>[Ver. 1.5以降] 軸番号0を指定すると、ResetEralw実行タスクのセマフォ取得中の軸の全部の偏差許容値をデフォルト値に戻します。</p>
マクロ定義	<pacman.h>ファイルが必要です。
関連項目	ResetCurLmt 、 _SetEralw
注意事項	<p>(1) [Ver. 1.4以前] ロボット制御権を取得(TAKEARM)したタスクにて実行してください。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。</p> <p>[Ver. 1.5以降] 制御権を取得(TAKEARM)したタスクにて実行してください。制御権未取得の軸を指定した場合は、エラー「27D* *軸セマフォを取得できません」が発生します。</p> <p>(2) 電流制限解除命令(ResetCurLmt)実行時も同様に偏差許容値がデフォルト値になります。</p>
用例	CALL ResetEralw(0) ; 全軸の偏差許容値をデフォルト値に戻す。

OnSrvLock (ライブラリ)

機能	指定した軸をサーボロック状態にします。(4軸ロボット専用)
書式	OnSrvLock (<指定軸>)
説明	従来言語の ON SVLOCK 命令と同等な機能を提供します。 サーボロックとはロボットのアームが制御され、その位置が保たれている状態をいいます。
マクロ定義	<pacman.h>ファイルが必要です。
関連事項	OffSrvLock
注意事項	(1) サーボロックは、ロボット停止状態で SET してください。パス動作中に SET すると、エラーが発生する事があります。 (2) ロボット制御件を取得(TAKEARM)したタスクにて実行下さい。ロボット制御件未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。
用例	CALL OnSrvLock(1) ' 1軸のサーボロック CALL OnSrvLock(0) ' 全軸のサーボロック

OffSrvLock (ライブラリ)

機能 指定した軸のサーボロックを解除します。(4軸ロボット専用)

書式 OffSrvLock

説明 従来言語の OFF SVLOCK 命令と同等な機能を提供します。
サーボロックとはロボットのアームが制御され、その位置が保たれている状態をいいます。サーボロックを解除するとロボットのアームは位置が保たれないため、外力が加わると位置がずれます。

マクロ定義 <pacman.h>ファイルが必要です。

関連事項 OnSrvLock

- 注意事項**
- (1) サーボロックが解除状態にある軸は、動作コマンドを実行できません。
 - (2) サーボロック解除は、ロボット停止状態で SET してください。パス動作中に SET すると、エラーが発生する事があります。
 - (3) ロボット制御件を取得 (TAKEARM) したタスクにて実行下さい。ロボット制御件未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。
 - (4) 使用条件の“25:電流制限リセット”設定値の2bit目が“0” (初期値) の場合は、モータ電源入時にサーボロック解除がリセット(サーボロック)されます。モータ電源投入直後からサーボロック解除を有効にする場合は、電流制限リセット設定値に“+2”を設定して下さい。

注：使用条件“25：電流制限リセット”設定例

	PWM	SVLock	CurLmt	
	*	*	*	
SVLock のみ有効の場合	0	1	0	= 2
全て有効の場合	1	1	1	= 7

用例

CALL OnSrvLock(1) ; 1軸のサーボロック

OnPWM (ライブラリ)

機能 指定した軸を PWM スイッチング制御します。(4 軸ロボット専用)

書式 OnPWM (<指定軸>)

説明 PWM スイッチング状態とはロボットのアームが制御され、その位置が保たれている状態をいいます。

マクロ定義 <pacman.h>ファイルが必要です。

関連事項 OffPWM

注意事項

- (1) PWM スイッチングは、ロボット停止状態で SET してください。パス動作中に SET すると、エラーが発生する事があります。
- (2) ロボット制御件を取得(TAKEARM)したタスクにて実行下さい。ロボット制御件未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。

用例

CALL OnPWM(1) ' 1 軸の PWM スイッチング
CALL OnPWM(0) ' 全軸の PWM スイッチング

OffPWM (ライブラリ)

機能 指定した軸の PWM スイッチング制御を解除します。(4 軸ロボット専用)

書式 OffPWM (<指定軸>)

説明 PWM スイッチングを解除するとロボットのアームは位置が保たれないため、外力が加わると位置がずれます。ロボットは完全なフリー状態になります。

マクロ定義 <pacman.h>ファイルが必要です。

関連事項 OnSrvLock

- 注意事項**
- (1) PWM スイッチングが解除状態にある軸は、動作コマンドを実行できません。
 - (2) PWM スイッチング解除は、ロボット停止状態で SET してください。パス動作中に SET すると、エラーが発生する事があります。
 - (3) ロボット制御件を取得(TAKEARM)したタスクにて実行下さい。ロボット制御件未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。
 - (4) 使用条件の"25:電流制限リセット"設定値の3bit目が"0"(初期値)の場合は、モータ電源入時にPWMスイッチング解除がリセット(PWMスイッチング)されます。モータ電源投入直後から PWM スイッチング解除を有効にする場合は、電流制限リセット設定値に"+4"を設定して下さい。

注：使用条件"25：電流制限リセット"設定例

	PWM	SVLock	Curlmt	
	*	*	*	
PWMのみ有効の場合	1	0	0	= 4
全て有効の場合	1	1	1	= 7

用例

CALL OffPWM(1) ; 1 軸の PWM スイッチング解除

SetCycloidJnt (ライブラリ) [Ver. 1.7 以降]

機能	付加軸のエンド動作時のオーバーシュート量および残留振動を抑えるサイクロイド動作モードに移行します。
書式	SetCycloidJnt(<軸番号>)
説明	<ol style="list-style-type: none">(1) サイクロイド動作モードでは、減速時の速度変化を滑らかにすることができます。そのため、停止時のオーバーシュート量、および残留振動が低減できます。(2) サイクロイド動作モード使用時は動作時間が増加します。サイクルタイムに余裕があることを確認の上使用してください。(3) SetCycloidJnt を実行後、設定した軸を別のプログラムで動作させた場合、本モードは有効になっています。解除する場合は ResetCycloidJnt を実行してください。(4) ロボットとの同期動作では、この設定は参照されません。ロボットの設定に従って動作します。(5) 複数の付加軸を同期させて動作させる場合は、動作する軸の1つがサイクロイド動作モードに設定されていると同期動作する全ての軸に設定が適用されます。軸毎に設定を変えることはできません。
マクロ定義	<pacman.h>ファイルが必要です。
関連項目	ResetCycloidJnt
注意事項	指定する軸の制御権を取得 (TAKEARM) していないと実行できません。 SetCycloidJnt は付加軸専用です。ロボット軸には SetCycloid を使用してください。
用例	CALL SetCycloidJnt (8) ' 8 軸をサイクロイド動作モードに移行

Resetcycloid (ライブラリ)

機能	サイクロイド動作モードから通常モードに移行します。
書式	Resetcycloid
説明	サイクロイド動作モードから通常動作モードに移行します。再度サイクロイド動作モードにしたい場合は、Setcycloid を実行してください。
マクロ定義	<pacman.h>ファイルが必要です。
関連事項	setcycloid
注意事項	(1) ロボット制御権を取得 (TAKEARM) したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。
用例	CALL Resetcycloid ’サイクロイド動作モードの解除

ResetCycloidJnt (ライブラリ) [Ver. 1.7 以降]

機能	サイクロイド動作モードを解除し、通常モードに移行します。
書式	ResetCycloidJnt (<軸番号>)
説明	サイクロイド動作モードから通常動作モードに移行します。再度、サイクロイド動作モードにしたい場合は SetCycloidJnt を実行してください。
マクロ定義	<pacman.h>ファイルが必要です。
関連項目	SetCycloidJnt
注意事項	指定する軸の制御権を取得 (TAKEARM) していないと実行できません。 ResetCycloidJnt は付加軸専用です。ロボット軸には ResetCycloid を使用してください。
用例	CALL ResetCycloidJnt(8) ’ 8軸のサイクロイド動作モードを解除

SetCPSpdMode (ライブラリ) [Ver. 1.8 以降]

機能	CP 動作時の TCP 速度設定を変更する。
書式	SetCPSpdMode (<設定値>)
説明	デフォルト設定では、手先の回転動作を含んだ CO 動作を行った場合は回転動作量に応じて TCP 速度が減速されます。設定を 1 にすることにより、回転動作量が一定量以下であれば TCP 速度を一定に保つことができます。
注意事項	設定 1 の場合、TCP 速度を一定に保つため動作内容によって回転速度が変化します。設定された回転速度の限界値を超えるような動作を実行しようとした場合は警告が表示されるとともに TCP 速度が減速されて動作が実行されます。
用例	CALL SetCPSpdMode(0) ’ デフォルトの速度設定にします CALL SetCPSpdMode(1) ’ TCP 速度を 1 にします

SetForce_HM (ライブラリ)

機能	HM/HS ロボットにてZ軸の推力 (単位: N) を指定した電流制限ライブラリです。
書式	SetForce_HM (<推力 (単位: N) >)
説明	<p>Z軸のモータ電流を制限することにより、推力を制限します。単位はN (ニュートン) です。</p> <p>周辺機器との接触時に生じる衝撃力により最大推力は大きく異なりますが、この指定できる推力は衝突時の衝撃力がほぼ"0"の時の推力です。(極低速で衝突した場合) またメカの摩擦によりZ軸の位置によっても推力は違ってきますのであくまでも目安と考えてください。(保証値ではありません) また電流制限ライブラリ (SetCurLmt) と同様に許容値を超える設定をした場合は許容値に制限されます。</p>
マクロ定義	<pacman.h>ファイルが必要です。
関連事項	SetForce_HC, SetCurLmt, ResetCurLmt
注意事項	<ol style="list-style-type: none">(1) 電流制限実施中は、モータ電流が制限されるため、最高速度、最高加速度で動作できません。電流制限は必要なステップのみ使用してください。また電流制限を使用する際は、加速度を下げてください。(2) 電流制限にて推力を制限しても高速でワークが衝突すると、ワークとハンドの慣性で衝撃力が発生します。電流制限はワークが接触する前からセットし、かつ速度を下げてください。(3) 電流制限は、ロボット停止状態でセットしてください。パス動作中にセットすると、エラーが発生することがあります。(4) ロボット制御権を取得 (TAKEARM) したタスクにて実行してください。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。(5) 使用条件の"25:電流制限リセット"設定値の1bit目が"0" (初期値) の場合は、モータ電源入時にサーボロック解除がリセット (サーボロック) されます。モータ電源投入直後からサーボロック解除を有効にする場合は、電流制限リセット設定値に"+1"を設定してください。設定方法は"OffSrvlock"の注意事項を参照してください。(6) このコマンドを解除するときは"ResetCurLmt (3)"を実行してください。(7) 細かい設定を行っても摩擦等の関係から指定した推力が出力されない場合があります。

用例

CALL SetForce_HM(10.0) ' 10.0(N) の推力を指定する

SetForce_HC (ライブラリ)

機能	HC ロボットにて Z 軸の推力 (単位 : N) を指定した電流制限ライブラリです。
書式	SetForce_HC (<推力 (単位 : N) >)
説明	<p>Z 軸のモータ電流を制限することにより、推力を制限します。単位は N (ニュートン) です。</p> <p>周辺機器との接触時に生じる衝撃力により最大推力は大きく異なりますが、この指定できる推力は衝突時の衝撃力がほぼ "0" の時の推力です。(極低速で衝突した場合) またメカの摩擦により Z 軸の位置によっても推力は違ってきますのであくまでも目安と考えてください。(保証値ではありません) また電流制限ライブラリ (SetCurLmt) と同様に許容値を超える設定をした場合は許容値に制限されます。</p>
マクロ定義	<pacman.h>ファイルが必要です。
関連事項	SetForce_HM, SetCurLmt, ResetCurLmt
注意事項	<ol style="list-style-type: none">(1) 電流制限実施中は、モータ電流が制限されるため、最高速度、最高加速度で動作できません。電流制限は必要なステップのみ使用してください。また電流制限を使用する際は、加速度を下げてください。(2) 電流制限にて推力を制限しても高速でワークが衝突すると、ワークとハンドの慣性で衝撃力が発生します。電流制限はワークが接触する前からセットし、かつ速度を下げてください。(3) 電流制限は、ロボット停止状態でセットしてください。パス同 s あ中にセットすると、エラーが発生することがあります。(4) ロボット制御権を取得 (TAKEARM) したタスクにて実行してください。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。(5) 使用条件の "25:電流制限リセット" 設定値の 1 bit 目が "0" (初期値) の場合は、モータ電源入時にサーボロック解除がリセット (サーボロック) されます。モータ電源投入直後からサーボロック解除を有効にする場合は、電流制限リセット設定値に "+ 1" を設定してください。設定方法は "DffSrvlock" の注意事項を参照してください。(6) このコマンドを解除するときは "ResetCurLmt (3)" を実行してください。(7) 細かい設定を行っても摩擦等の関係から指定した推力が出力されない場合があります。

用例

CALL SetForce_HC(10.0) ' 10(N) の推力を指定する

SetCompControl (ライブラリ) [Ver. 1.4 以降]

機能	力制限機能を有効にします。(6軸専用命令)
書式	SetCompControl
説明	力制限機能を有効にします。SetFrcLimit, SetCompRate, SetFrcCoord 等で設定された力制限条件を有効にします。
マクロ定義	<pacman.h>ファイルが必要です。
関連事項	SetFrcLimit, SetCompRate, SetFrcCoord, ResetCompControl SetCompFControl
注意事項	<ol style="list-style-type: none">(1) 重力補償無効時、電流制限有効時に本ライブラリを実行した場合、エラー「60f5 力制限実行できません」が発生します。電流制限を無効、重力補償を有効にした後、再度、実行してください。電流制限無効化は、ResetCurLmt、重力補償有効化は、SetGravity を参照ください。(2) モータ電源 OFF 状態で実行しても、力制限は有効になりません。また、力制限中にモータ電源 OFF した場合は、力制限は無効になります。(3) ロボット制御権を取得(TAKEARM)したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。(4) ロボット停止状態で実行してください。パス動作中に本ライブラリを実行した場合は、エンド動作となります。また、ロボット動作中に本ライブラリを実行し、エラー「600b ロボット動作中です」が発生する場合、Delay 命令等でロボット停止を待ってから実行してください。(5) 力制限中にロボットが外力等で動作する場合、エラー「611* 偏差過大」が発生する場合があります。エラー発生する場合は、偏差許容値を変更してください。偏差許容値変更は、SetEralw を参照ください。(6) ロボットが接触状態などロボットに力が加わった状態で本ライブラリを実行しないでください。力が加わった状態で力制限機能を有効にする場合は、SetCompFControl を使用してください。(7) SetCompControl 実行後、ロボットの姿勢が大きく変化した場合は、重力補償の補正値に誤差が生じ、ロボットが重力方向に動作する場合があります。力制限中に姿勢が大きく変化する場合は、途中で ResetCompControl にて力制限を無効にした後、再度 SetCompControl にて力制限を有効にしてください。

用例

```
CALL SetFrcCoord(1)           ' 力制限座標系を設定する。
CALL SetFrcLimit(100, 0, 100, 100, 100, 100)
                                ' 力制限割合を設定する。
CALL SetCompControl           ' 力制限機能を有効にする。
CALL SetEralw(1, 90)         ' 偏差許容値を設定する。
```

SetCompFControl (ライブラリ) [Ver. 1.4 以降]

機能	力制限機能を有効にします。(6軸専用命令)
書式	SetCompFControl
説明	SetCompControlと同様、力制限機能を有効にします。ただし、SetCompControlにて実行される重力補償補正処理は実行されません。
マクロ定義	<pacman.h>ファイルが必要です。
関連事項	SetCompControl
注意事項	<ol style="list-style-type: none">(1) 重力補償無効時、電流制限有効時に本ライブラリを実行した場合、エラー「60f5 力制限実行できません」が発生します。電流制限を無効、重力補償を有効にした後、再度、実行してください。(2) モータ電源 OFF 状態で実行しても、力制限は有効になりません。また、力制限中にモータ電源 OFF した場合は、力制限は無効になります。(3) ロボット制御権を取得(TAKEARM)したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。(4) ロボット停止状態で実行してください。パス動作中に本ライブラリを実行した場合は、エンド動作となります。また、ロボット動作中に本ライブラリを実行し、エラー「600b ロボット動作中です」が発生する場合、Delay 命令等でロボット停止を待ってから実行してください。(5) 先端負荷設定を正確に設定してください。先端負荷設定値と実際の先端負荷が異なる場合、重力方向に落下する場合があります。また、SetGrvOffsetにて重力補償補正を実行して頂くと重力落下を防止できます。

用例

```
CALL SetGrvOffset          ' 重力補償補正值を計算する。
CALL SetFrcCoord(1)       ' 力制限座標系を設定する。
CALL SetFrcLimit(100, 0, 100, 100, 100) ' 力制限割合を設定する。
CALL SetCompFControl      ' 力制限を有効化する。
```

ResetCompControl (ライブラリ) [Ver. 1.4 以降]

機能	力制限機能を無効にします。(6軸専用命令)
書式	ResetCompControl
説明	力制限機能を無効にします。
マクロ定義	<pacman.h>ファイルが必要です。
関連事項	SetCompControl
注意事項	<ol style="list-style-type: none">(1) ロボット制御権を取得(TAKEARM)したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。(2) ロボット停止状態で実行してください。パス動作中に本ライブラリを実行した場合は、エンド動作となります。また、ロボット動作中に本ライブラリを実行し、ロボットが急停止したり、エラー「612* 過電流」が発生する場合、<設定値>を1に設定するか、Delay 命令等でロボット停止を待ってから実行してください。(3) 本ライブラリ実行中に瞬時停止し、ステップバックやプログラムリセット操作した場合、エラー「60f9 力制限無効操作異常です」が発生します。(4) 力制限有効時に、SetEralwにて偏差許容値を変更した場合、偏差許容値は、初期値に戻ります。しかし、ResetCompControl 実行時にエラーが発生した場合、初期値に戻らない場合がありますので、力制限無効後、ResetEralwにて偏差許容値を初期値に戻してください。(5) エラー「608* J*指令速度限界オーバ」が発生する場合があります。その場合、ResetCompControl 実行前に aspChangeにて最適可搬質量モードを2又は3に変更し、実行後に最適可搬質量設定モードを元の値に戻してください。

用例

CALL ResetCompControl	' 力制限機能を無効化する。
CALL ResetEralw(0)	' 偏差許容値を初期化する。

SetFrcCoord (ライブラリ) [Ver. 1.4 以降]

機能 力制限設定座標系を選択します。(6軸専用命令)

書式 SetFrcCoord(<設定値>)

説明 SetFrcLimit, SetCompRate にて設定する力制限値の座標系を選択します。
設定値が0の場合、ロボットのベース座標を選択し、設定値が1の場合は、ツール座標、設定値が2の場合は、ワーク座標を選択します。

マクロ定義 <pacman.h>ファイルが必要です。

関連事項 SetFrcLimit, SetCompRate, SetFrcCoord, ResetCompControl

- 注意事項**
- (1) ロボット制御権を取得(TAKEARM)したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。
 - (2) 力制限中は、実行できません。力制限中に本ライブラリを実行すると、エラー「60fa 力制限中です」が発生します。
 - (3) <設定値>を1とし、ツール座標を選択した場合、ツール座標は、力制限有功設定(SetCompControl 実行)時のツール座標となります。力制限中に changetool 命令にてツール座標を変更しても、力制限座標は変化しません。
 - (4) <設定値>を2とし、ワーク座標を選択した場合、ワーク座標は、力制限有功設定(SetCompControl 実行)時のワーク座標となります。力制限中に changework 命令にてワーク座標を変更しても、力制限座標は変化しません。
 - (5) コントローラ電源立ち上げ直後は、設定値は初期化され、0 (ベース座標) となります。

用例

```
CALL SetFrcCoord(1)           ' 力制限座標系をツール座標に設定する。
Changetool 2                  ' ツール座標をツール2に設定する。
CALL SetFrcLimit(100, 0, 100, 100, 100, 100)
                              ' 力制限割合を設定する。
CALL SetCompControl           ' ツール2座標系のY方向の力制限割合を0%に設定し、
                              ' 力機能を有効にする。
```

SetFrcLimit (ライブラリ) [Ver. 1.4 以降]

機能	力制限割合を設定します。(6軸専用命令)
書式	SetFrcLimit(<X方向制限割合>, <Y方向制限割合>, <Z方向制限割合>, <X回り制限割合>, <Y回り制限割合>, <Z回り制限割合>)
説明	SetFrcCoordにて設定された座標系のX軸方向、Y軸方向、Z軸方向、X軸回り、Y軸回り、Z軸回りの力制限割合を設定します。 設定値は0~100で設定します。小数点2桁まで有効です。
マクロ定義	<pacman.h>ファイルが必要です。
関連事項	ResetFrcLimit, SetFrcCoord, SetCompControl
注意事項	(1) ロボット制御権を取得(TAKEARM)したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。 (2) 力制限中は、実行できません。力制限中に本ライブラリを実行すると、エラー「60fa 力制限中です」が発生します。 (3) コントローラ電源立ち上げ直後は、設定値は初期化され、X軸方向、Y軸方向、Z軸方向、X軸回り、Y軸回り、Z軸回りすべて100となります。
用例	CALL SetFrcCoord(1) ' 力制限座標系をツール座標に設定する。 CALL SetFrcLimit(100, 0, 100, 100, 100) ' 力制限割合を設定する。 CALL SetCompControl ' ツール座標系のY方向の力制限割合を0%に設定し、力 ' 機能を有効にする。

ResetFrcLimit (ライブラリ) [Ver. 1.4 以降]

機能 力制限割合を初期化します。(6軸専用命令)

書式 ResetFrcLimit

説明 力制限割合を初期化します。X軸方向, Y軸方向, Z軸方向, X軸回り, Y軸回り, Z軸回りすべて100%となります。

マクロ定義 <pacman.h>ファイルが必要です。

関連事項 SetFrcLimit

注意事項

- (1) ロボット制御権を取得(TAKEARM)したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。
- (2) 力制限中は、実行できません。力制限中に本ライブラリを実行すると、エラー「60fa 力制限中です」が発生します。

用例

```
CALL ResetCompControl      ' 力制限を無効にします。
CALL ResetFrcLimit         ' 力制限割合を初期化します。
```

SetCompRate (ライブラリ) [Ver. 1.4 以降]

機能	力制限時の柔らかさの割合を設定します。(6軸専用命令)
書式	SetCompRate(<X方向柔らかさ>, <Y方向柔らかさ>, <Z方向柔らかさ>, <X回り柔らかさ>, <Y回り柔らかさ>, <Z回り柔らかさ>)
説明	<p>SetFrcCoordにて設定された座標系のX軸方向、Y軸方向、Z軸方向、X軸回り、Y軸回り、Z軸回りの柔らかさの割合を設定します。</p> <p>設定値は0~100の範囲で設定します。0の時最も柔らかくなります。小数点2桁まで有効です。</p>
マクロ定義	<pacman.h>ファイルが必要です。
関連事項	ResetCompRate, SetFrcCoord, SetCompControl
注意事項	<ol style="list-style-type: none">(1) ロボット制御権を取得(TAKEARM)したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。(2) 力制限中は、実行できません。力制限中に本ライブラリを実行すると、エラー「60fa 力制限中です」が発生します。(3) コントローラ電源立ち上げ直後は、設定値は初期化され、X軸方向、Y軸方向、Z軸方向、X軸回り、Y軸回り、Z軸回りすべて100となります。
用例	<pre>CALL SetFrcCoord(1) ' 力制限座標系をツール座標に設定する。 CALL SetCompRate(100, 0, 100, 100, 100, 100) ' 柔らかさ割合を設定する。 CALL SetCompControl ' ツール座標系のY方向の柔らかさ割合を0%に設定し、 ' 力機能を有効にする。</pre>

ResetCompRate (ライブラリ) [Ver. 1.4 以降]

機能 力制限時の柔らかさの割合を初期化します。(6軸専用命令)

書式 ResetCompRate

説明 X軸方向、Y軸方向、Z軸方向、X軸回り、Y軸回り、Z軸回りの柔らかさの割合を初期化し、すべて100とします。

マクロ定義 <pacman.h>ファイルが必要です。

関連事項 SetCompRate

注意事項

- (1) ロボット制御権を取得(TAKEARM)したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。
- (2) 力制限中は、実行できません。力制限中に本ライブラリを実行すると、エラー「60fa 力制限中です」が発生します。

用例

```
CALL ResetCompControl      ' 力機能を無効にする。
CALL ResetCompRate         ' 柔らかさ割合を初期化する。
```

SetFrcAssist (ライブラリ) [Ver. 1.4 以降]

機能 力制限時のオフセット力を設定します。(力制限特殊機能ライブラリ)(6軸専用命令)

書式 SetFrcAssist(<X 方向オフセット力>,<Y 方向オフセット力>,<Z 方向オフセット力>,<X 回りオフセットモーメント>,<Y 回りオフセットモーメント>,<Z 回りオフセットモーメント>)

説明 SetFrcCoord にて設定された座標系の X 軸方向、Y 軸方向、Z 軸方向、X 軸回り、Y 軸回り、Z 軸回りのオフセット力、モーメントを設定します。力制限最大値の 10% が設定最大値となります。

力設定値の単位は [N] です。モーメント設定値の単位は [Nm] です。小数点 1 桁まで有効です。

マクロ定義 <pacman.h>ファイルが必要です。

関連事項 ResetFrcAssist, SetFrcCoord, SetCompControl

- 注意事項**
- (1) ロボット制御権を取得(TAKEARM)したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。
 - (2) オフセット力、モーメントを加える方向にロボットが動作する場合があります。動作する場合は、設定値を下げてください。
 - (3) コントローラ電源立ち上げ直後は、設定値は初期化され、X 軸方向、Y 軸方向、Z 軸方向、X 軸回り、Y 軸回り、Z 軸回りすべて 0 となります。

用例

```
CALL SetFrcCoord(1)           ' 力制限座標系をツール座標に設定する。
CALL SetFrcAssist(-30, 0, 0, 0, 0)
                                ' -X 方向に 30 [N] のオフセット力を設定する。
CALL SetFrcLimit(0, 100, 100, 100, 100, 100)
                                ' 力制限割合を設定する。
CALL SetCompControl           ' 力機能を有効にする。X 方向は 0 % に力制限され、-X
                                ' 方向に 30 [N] の力を加える。
```

ResetFrcAssist (ライブラリ) [Ver. 1.4 以降]

機能	力制限時のオフセット力を初期化します。(力制限特殊機能ライブラリ) (6 軸専用命令)
書式	ResetFrcAssist
説明	SetFrcCoord にて設定された座標系の X 軸方向、Y 軸方向、Z 軸方向、X 軸回り、Y 軸回り、Z 軸回りのオフセット力、モーメントを 0 にします。
マクロ定義	<pacman.h>ファイルが必要です。
関連事項	SetFrcAssist
注意事項	(1) ロボット制御権を取得 (TAKEARM) したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。
用例	<pre>CALL ResetCompControl ' 力制限を無効にする。 CALL ResetFrcAssist ' オフセット力、モーメントを 0 に設定する。</pre>

SetCompJLimit (ライブラリ) [Ver. 1.4 以降]

機能	力制限時の電流制限値を設定します。(力制限特殊機能ライブラリ) (6軸専用命令)
書式	SetCompJLimit (<J1 電流制限値>, <J2 電流制限値>, <J3 電流制限値>, <J4 電流制限値>, <J5 電流制限値>, <J6 電流制限値>)
説明	<p>力制限時の電流制限値を設定します。モータ定格電流値が 100 になります。SetFrcLimit にて全方向を 0 に設定した場合、モータ電流は、設定値以下に制限されます。</p> <p>設定値は 0~100 の範囲で設定します。</p>
マクロ定義	<pacman.h>ファイルが必要です。
関連事項	ResetCompJLimit, SetCompControl
注意事項	<ol style="list-style-type: none">(1) ロボット制御権を取得(TAKEARM)したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。(2) コントローラ電源立ち上げ直後は、設定値は初期化され、電流制限値は、すべて 0 となります。(3) SetCompJLimit に比較的大きな値を設定した場合、ロボットが発振的になりエラーが発生する場合があります。その場合、力制限の柔らかさを SetCompRate, SetDumpRate にて調整してください。
用例	<pre>CALL SetFrcCoord(1) ' 力制限座標系をツール座標に設定する。 CALL SetCompJLimit(30, 0, 0, 0, 0) ' J1 の電流制限値を 30%に設定する。 CALL SetFrcLimit(100, 0, 100, 100, 100, 100) ' 力制限割合を設定する。 CALL SetCompControl ' 力機能を有効にする。</pre>

ResetCompJLimit (ライブラリ) [Ver. 1.4 以降]

機能	力制限時の電流制限値を初期化します。(力制限特殊機能ライブラリ) (6軸専用命令)
書式	ResetCompJLimit
説明	力制限時の電流制限値を初期化し、全軸0に設定ます。
マクロ定義	<pacman.h>ファイルが必要です。
関連事項	SetCompJLimit, SetCompControl
注意事項	(1) ロボット制御権を取得(TAKEARM)したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。
用例	<pre>CALL ResetCompControl ' 力機能を無効にする。 CALL ResetCompJLimit ' 電流制限値を初期化する。</pre>

SetCompVMode (ライブラリ) [Ver. 1.4 以降]

機能	力制限時の速度制御モードを設定します。(力制限特殊機能ライブラリ) (6軸専用命令)
書式	SetCompVMode
説明	SetCompControl 実行時の力制限速度制御モードを有効にします。
マクロ定義	<pacman.h>ファイルが必要です。
関連事項	ResetCompVMode
注意事項	(1) ロボット制御権を取得(TAKEARM)したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。
用例	<pre>CALL SetCompVMode ' 力制限速度制御モードを有効にする。 CALL SetCompControl ' 力制限を有効にする。</pre>

ResetCompVMode (ライブラリ) [Ver. 1.4 以降]

機能	力制限時の速度制御モードを無効にします。(力制限特殊機能ライブラリ) (6 軸専用命令)
書式	ResetCompVMode
説明	SetCompControl 実行時の力制限速度制御モードを無効にします。
マクロ定義	<pacman.h>ファイルが必要です。
関連事項	SetCompVMode
注意事項	(1) ロボット制御権を取得 (TAKEARM) したタスクにて実行下さい。ロボット制御権未 取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。
用例	CALL ResetCompVMode ' 力制限速度制御モードを無効にする。 CALL SetCompControl ' 力制限を有効にする。

SetCompEralw (ライブラリ) [Ver. 1.4 以降]

機能	力制限時のツール端の位置、姿勢偏差許容値を設定します。(6軸専用命令)
書式	SetCompEralw(<X 方向偏差許容値>, <Y 方向偏差許容値>, <Z 方向偏差許容値>, <X 回り偏差許容値>, <Y 回り偏差許容値>, <Z 回り偏差許容値>)
説明	力制限時のツール端の位置、姿勢偏差許容値を設定します。X, Y, Z 方向の偏差許容値の単位は、(mm)、X, Y, Z 回りの偏差許容値の単位は、(度)です。小数点1桁まで有効です。
マクロ定義	<pacman.h>ファイルが必要です。
関連事項	ResetCompEralw、SetFrcCoord
注意事項	<ol style="list-style-type: none">(1) ロボット制御権を取得(TAKEARM)したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。(2) コントローラ電源立ち上げ直後の偏差許容値の初期値は、X, Y, Z 方向が 100(mm)、X, Y, Z 回りが 30(度)です。X, Y, Z 周りは、最大値が 175(度)です。最大値を超える設定をした場合、エラー「6003 有効な数値範囲を超えた」が発生します。(3) ツール端の位置、姿勢偏差が設定した許容値を超えた場合、エラー「60f8 力制限時位置偏差過大異常」が発生します。(4) 偏差設定座標系は、SetFrcCoord にて設定された座標系となります。例えば SetFrcCoord(1) と設定した場合は、SetCompEralw の設定座標系は、ツール座標系となります。

用例

```
CALL SetFrcCoord(2)          '力制限座標系をワーク座標にします。
CALL SetFrcLimit(100, 0, 100, 100, 100, 100) 'ワーク座標のY方向の力制限割合を0%
                                         にします。
CALL SetCompEralw(10, 150, 10, 5, 5, 5) 'ワーク座標X, Z方向の偏差許容値を10(mm), Y方
                                         向の偏差許容値を150(mm), X, Y, Z回りの偏差許容値を5
                                         (度)にします。
```

ResetCompEralw (ライブラリ) [Ver. 1.4 以降]

機能	力制限時のツール端の位置、姿勢偏差許容値を初期化します。(6軸専用命令)
書式	ResetCompEralw
説明	力制限時のツール端の位置、姿勢偏差許容値を初期化します。X, Y, Z 方向の偏差許容値の初期値は、100(mm)、X, Y, Z 回りの偏差許容値の初期値は、30(度)です。
マクロ定義	<pacman.h>ファイルが必要です。
関連事項	SetCompEralw
注意事項	(1) ロボット制御権を取得(TAKEARM)したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。
用例	CALL ResetCompEralw

SetDampRate (ライブラリ) [Ver. 1.4 以降]

機能	力制限時の粘性割合を設定します。(6軸専用命令)
書式	SetDampRate(<X方向粘性割合>, <Y方向粘性割合>, <Z方向粘性割合>, <X回り粘性割合>, <Y回り粘性割合>, <Z回り粘性割合>)
説明	<p>SetFrcCoordにて設定された座標系のX軸方向、Y軸方向、Z軸方向、X軸回り、Y軸回り、Z軸回りの粘性割合を設定します。</p> <p>設定値は0~100の範囲で設定します。0の時最も粘性が小さくなります。小数点2桁まで有効です。</p>
マクロ定義	<pacman.h>ファイルが必要です。
関連事項	ResetDampRate, SetFrcCoord, SetCompRate
注意事項	<ol style="list-style-type: none">(1) ロボット制御権を取得(TAKEARM)したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。(2) 力制限中は、実行できません。力制限中に本ライブラリを実行すると、エラー「60fa 力制限中です」が発生します。(3) コントローラ電源立ち上げ直後は、設定値は初期化され、X軸方向、Y軸方向、Z軸方向、X軸回り、Y軸回り、Z軸回りすべて100となります。(4) SetCompRateにて設定した柔らかさ割合より小さい値を設定しないでください。柔らかさ割合より小さい値を設定した場合、ロボットが振動し、エラー停止する場合があります。(5) SetDampRate実行後、SetCompRateを設定すると、粘性割合は、SetCompRateの設定値となります。
用例	<pre>CALL SetFrcCoord(1) ' 力制限座標系をツール座標に設定する。 CALL SetCompRate(100, 0, 100, 100, 100, 100) ' 柔らかさ割合を設定する。 CALL SetDampRate(100, 20, 100, 100, 100, 100) ' 粘性割合を設定する。 CALL SetCompControl ' ツール座標系のY方向の柔らかさ割合を0%、粘性割合を20%に設定し、力機能を有効にする。</pre>

ResetVibControl (ライブラリ) [Ver. 1.4 以降] [4 軸 : Ver. 1.9 以降]

機能 残留振動低減制御モードから通常制御モードに戻します。

書式 ResetVibControl

説明 残留振動低減制御モードから通常制御モードへ戻します。

マクロ定義 <pacman.h>ファイルが必要です。

関連事項 ResetVibControl

- 注意事項**
- (1) ロボット制御権を取得 (TAKEARM) したタスクにて実行下さい。ロボット制御権未取得の場合は、エラー「21F7 アームセマフォを取得できません」が発生します。
 - (2) SetEralw にて角度偏差許容値変更中に ResetVibControl を実行した場合、角度偏差許容値は初期化されます。再度 SetEralw にて角度偏差許容値を再設定してください。
 - (3) パス動作中に ResetVibControl を実行した場合、動作停止を待って実行されます。よってエンド動作になります。

用例

CALL ResetVibControl ’通常制御モードに設定する。

MotionSkip (ライブラリ) [Ver. 1.5 以降]

機能	実行中の動作命令を中断します。(Ver. 1.5 以降対応)
書式	MotionSkip
説明	MotionSkip 命令を実行したタスクで実行中の動作命令のみを中断します。別タスクの動作は中断しません。
マクロ定義	
関連項目	GetJntData、GetSrvData
注意事項	<ol style="list-style-type: none">(1) 制御権を取得(TAKEARM)したタスクにて実行ください。制御権未取得の場合は、エラー「実行できません」が発生します。(2) ロボット動作タスクにて MotionSkip 命令を実行した場合、ロボット動作命令を中断します。付加軸動作タスクにて MotionSkip 命令を実行した場合は付加軸動作を中断します。 ロボットと付加軸を含むアームグループの動作タスクにて MotionSkip 命令を実行した場合、ロボット、付加軸とも動作を中断します。

用例

```
defjnt lj1
defsg lf1
move p, P1, next
lj1=GetSrvData(2)      '各軸の偏差を取り込む
lf1=ABS(JOINT(2, lj1)) '2軸の偏差を抽出する
if lf1 > 10000 then
    CALL MotionSkip    '2軸の偏差が 10000(Pulse)を超えたら動作命令を中断する
endif
```

MotionComp (ライブラリ) [Ver. 1.5 以降]

機能 動作命令実行が完了したかどうかを判断します。(Ver. 1.5 以降対応)

書式 MotionComp(<動作命令完了状態>)

説明 動作命令実行完了の時、1 を<動作命令完了状態>で返します。

MotionComp 命令を実行したタスクで実行中の動作命令の実行完了を判断します。別タスクの動作は判断しません。

命令がエンコーダ値確認動作の場合、エンコーダ値収束にて動作完了と判断します。それ以外の場合は、動作指令値が停止した時動作完了と判断します。

マクロ定義

関連項目 GetSrvData、GetJntData、[MotionSkip](#)

- 注意事項**
- (1) 制御権を取得(TAKEARM)したタスクにて実行ください。制御権未取得の場合は、エラー「実行できません」が発生します。
 - (2) ロボット動作タスクにて MotionComp 命令を実行した場合、ロボット動作命令の完了を判断します。付加軸動作タスクにて MotionComp 命令を実行した場合は付加軸動作の完了を判断します。
ロボットと付加軸を含むアームグループの動作タスクにて MotionComp 命令を実行した場合、ロボット、付加軸同期動作を完了と判断します
 - (3) 瞬時停止中は、動作中と判断します。
 - (4) <動作命令完了状態>にローカル変数を使用する場合、ローカル変数はあらかじめ 0 に初期化しておいてください。

用例

```
defint comp=0          ' 動作命令完了状態の初期化
defjnt lj1
defsnrg lf1
move p, P1, next
DO
  lj1=GetSrvData(2)    ' 各軸の偏差を取り込む
  lf1=ABS(JOINT(2, lj1)) ' 2 軸の偏差を抽出する
  if lf1 > 10000 then
    CALL MotionSkip    ' 2 軸の偏差が 10000 (Pulse) を超えたら動作命令を中断し、
                      ' ループを終了します。
  EXIT DO
endif
CALL MotionComp(comp)
LOOP UNTIL comp=1     ' 動作命令が完了するまでループします。
```

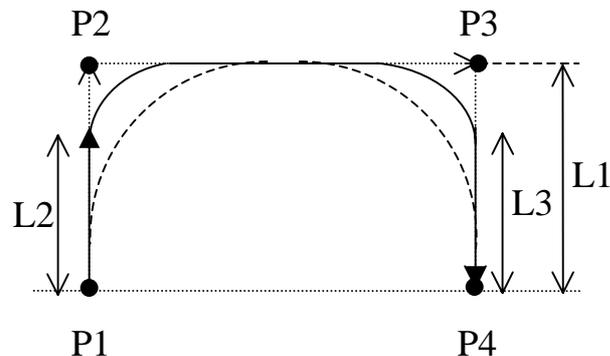
ArchMove (ライブラリ) [Ver.1.9 以降の 4 軸ロボットのみ]

機能 アーチモーションを実施します。

書式 ArchMove (<目標位置 (P型変数)>、<Z軸動作量>)

説明

現在位置 P1 と目標位置 P4 および経由点 P2、P3 の 4 点にて P&P 動作を行いません。経由点 P3 は目標位置に対して指定した<Z 軸動作量>(L1)上昇させた位置となります。経由点 P2 は現在位置の Z 軸延長上で経由点 P3 の Z 軸値と同一となる位置となります。図中のアーチ開始位置 (L2)、アーチ完了位置 (L3) は SetArchParam にて設定します。



マクロ定義 <pacman.h>ファイルが必要です。

関連事項 SetArchParam

注意事項

- (1) ロボット制御権を取得 (TAKEARM) したタスクにて実行してください。
- (2) この動作は位置制御を行っている訳ではありません。動作速度によって動作経路が変化しますので、動作の際は周辺設備との干渉に気を付けてください。
- (3) 現在位置と目標位置の Z 値が同一でない場合は正常に機能しません。Z 方向の位置を同一にした状態で使用してください。
- (4) ワーク座標系には対応していません。ベース座標系で使用してください。

用例

CALL ArchMove (P10,100) ‘目標位置 P10、Z 軸動作量 100mm で動作します。
なお、アーチ開始位置、アーチ完了位置は現在の設定値が適用されます。

SetArchParam (ライブラリ) [Ver.1.9 以降の 4 軸ロボットのみ]

機能 アーチモーションにおける、上昇動作中に横方向動作を開始する位置（アーチ開始位置）および下降動作中に横方向動作を終了する位置（アーチ完了位置）の設定

書式 SetArchParam (<アーチ開始位置>、<アーチ完了位置>)

説明

1. ArchMove を実行する場合のアーチの形状を設定できます。
2. アーチ開始位置、アーチ完了位置はそれぞれ現在位置、目標位置からの Z 軸方向の距離[mm]で指定します。
3. 設定値は 0 ～（Z 軸方向の最大動作可能距離）で、1mm 刻みで設定可です。
4. ロボットコントローラの電源投入後の初期値はアーチ開始位置、アーチ完了位置とも 0mm になっています。また、SetArchParam で設定した値は、ロボットコントローラの電源を切るまで有効です。

マクロ定義 <pacman.h>ファイルが必要です。

関連事項 ArchMove

注意事項 無効な数値が設定された場合、即時エラーとはなりません。ArchMove 実行時にエラーが発生します。

用例

CALL SetArchParam (10,20) ‘10mm 上昇後に横方向動作を開始し、下降完了 20mm 手前で横方向動作が完了するように設定。

：
<途中省略>

CALL ArchMove (P10,100) ‘目標位置 P10、Z 軸動作量 100mm で動作します。

SetMonitorCond (ライブラリ) [Ver. 1.5 以降]

機能 サーボ単軸データモニタ機能のモニタ条件を設定します。(Ver. 1.5 以降対応)

書式 SetMonitorCond(<軸番号>, <モニタデータ 1 選択>, <モニタデータ 2 選択>, <サンプリング間隔>)

説明 サーボ単軸データモニタ機能のモニタ条件として、モニタする軸番号、モニタするデータ (2 種類)、サンプリング間隔(ms)を設定します。モニタ可能なデータは以下の 5 種類で、<モニタデータ 1 選択>, <モニタデータ 2 選択>の指定により一度に 2 種類のデータを選択できます。

<モニタデータ 1 選択> <モニタデータ 2 選択>	モニタするデータ
0	モータ速度指令値 [r/min]
1	モータ速度現在値 (実速度) [r/min]
2	モータトルク指令値 (トルクオフセットを除いた値) [定格比 %]
3	モータ角度偏差 (モータ角度指令値-モータ実角度) [Pulse]
4	モータ電流絶対値 (モータ 3 相の電流検出値の絶対値のうち、最大となる値) [定格比 %]

<サンプリング間隔>は、1 ~ 8 ms の 8 刻みで設定します。

マクロ定義

関連項目 [ClearSrvMonitor](#)、[StartSrvMonitor](#)、[StopSrvMonitor](#)

- 注意事項**
- (1) モニタ開始命令 (StartSrvMonitor) 実行後、本命令を実行すると、エラー「6001 実行できません」が発生します。必ずモニタ開始前に条件を設定してください。
 - (2) 軸番号、データ種類、サンプリング間隔に誤った数値を設定した場合、エラー「範囲外です」が発生します。エラー発生時は指定値を確認ください。

用例

```
CALL SetMonitorCond(7, 0, 3, 4)    '7 軸の速度指令値、角度偏差、を 4ms 毎に取り込む設定を  
                                  'します。  
CALL StartSrvMonitor              'データモニタを開始します。
```

StartSrvMonitor (ライブラリ) [Ver. 1.5 以降]

機能	サーボ単軸データモニタを開始します。(Ver. 1.5 以降対応)
書式	StartSrvMonitor
説明	サーボ単軸データモニタを開始します。StartSrvMonitor を実行後 StopSrvMonitor を実行するまで、最大 1250 個のサーボデータを取得します。
マクロ定義	
関連項目	ClearSrvMonitor 、 SetMonitorCond 、 StopSrvMonitor
注意事項	<ol style="list-style-type: none">(1) StartSrvMonitor 実行 (モニタ開始) から StopSrvMonitor 実行 (モニタ終了) までのデータ数が 1250 個未満の場合は、開始から終了までのデータをモニタします。1250 個を超える場合は、終了以前 1250 個のデータをモニタします。(2) モニタ開始から終了までの間にエラーなどが発生しモータ電源が OFF した場合は、モータ OFF 後 400 個とモータ OFF 前、最大 850 個のデータをモニタします。(3) モータ OFF 中はデータモニタできません。モータ ON 状態で実行してください。

用例

```
CALL SetMonitorCond(7, 0, 3, 4)    ' 7 軸の速度指令値、角度偏差、を 4ms 毎に取り込む設定を  
                                   ' します。  
CALL StartSrvMonitor              ' データモニタを開始します。
```

StopSrvMonitor (ライブラリ) [Ver. 1.5 以降]

機能	サーボ単軸データモニタを終了します。(Ver. 1.5 以降対応)
書式	StopSrvMonitor
説明	サーボ単軸データモニタを終了します。StartSrvMonitor を実行後 StopSrvMonitor を実行するまで最大 1250 個のサーボデータを取得します。
マクロ定義	
関連項目	ClearSrvMonitor 、 SetMonitorCond 、 StartSrvMonitor
注意事項	<ol style="list-style-type: none">(1) StartSrvMonitor 実行 (モニタ開始) から StopSrvMonitor 実行 (モニタ終了) までのデータ数が 1250 個未満の場合は、開始から終了までのデータをモニタします。1250 個を超える場合は、終了以前 1250 個のデータをモニタします。(2) モニタ開始から終了までの間にエラー等が発生しモータ電源が OFF した場合は、モータ OFF 後 400 個とモータ OFF 前、最大 850 個のデータをモニタします。

用例

```
CALL StopSrvMonitor                ' データモニタを終了します。
```

ClearSrvMonitor (ライブラリ) [Ver. 1.5 以降]

機能 サーボ単軸データモニタの取得データのポイントを初期化します。(Ver. 1.5 以降対応)

書式 ClearSrvMonitor

説明 サーボ単軸データモニタの取得データのポイントを初期化し、新たに 1250 個のデータ取得を開始します。

マクロ定義

関連項目 [ClearSrvMonitor](#)、[SetMonitorCond](#)、[StartSrvMonitor](#)

注意事項 ClearSrvMonitor 実行 (クリア) から StopSrvMonitor 実行 (モニタ終了) までのデータ数が 1250 個未満の場合は、クリアから終了までのデータをモニタします。1250 個を超える場合は、終了以前 1250 個のデータをモニタします。

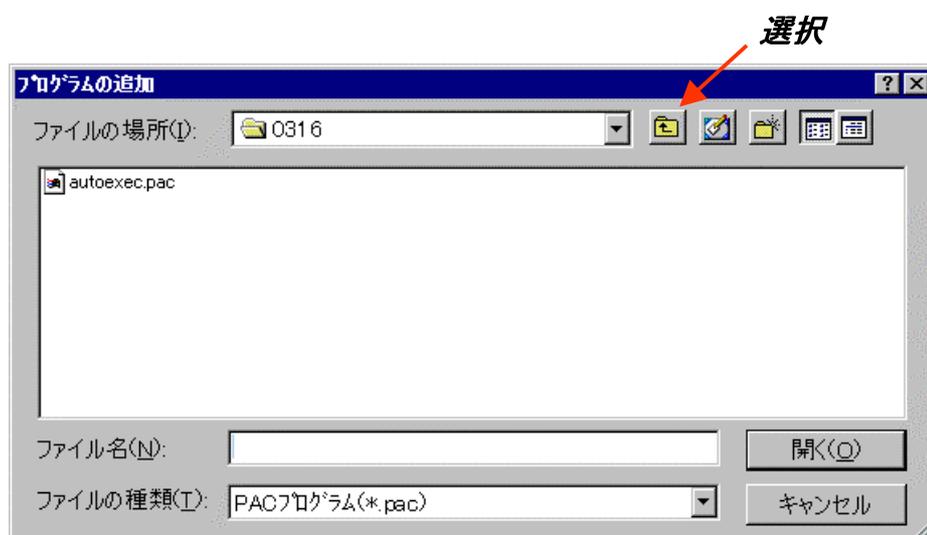
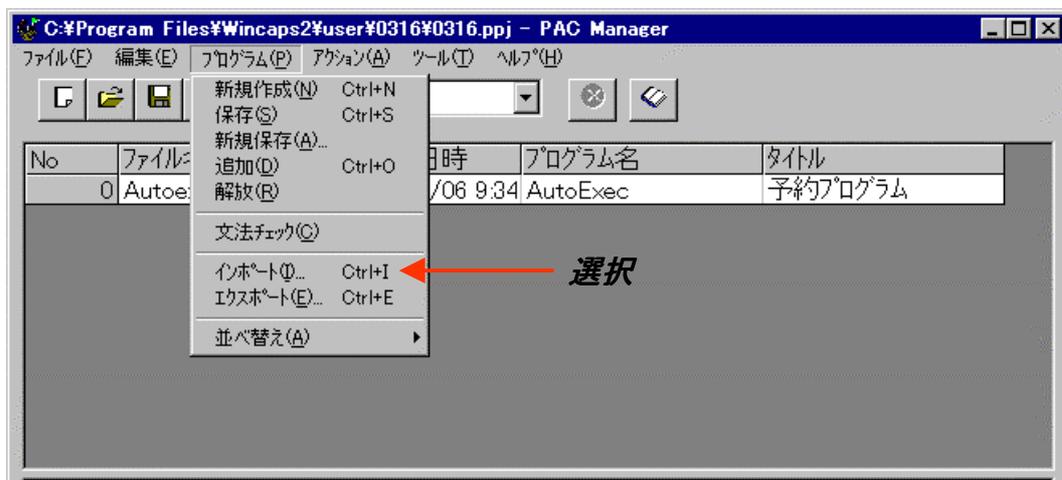
用例

```
CALL StartSrvMonitor                    'データモニタを開始します。
:
:
CALL ClearSrvMonitor                    'StartSrvMonitor 以降のデータをクリアします。
:
:
CALL StopSrvMonitor                    'データモニタを終了します。
                                         'ClearSrvMonitor から StopSrvMonitor 間のデータを
                                         'モニタします。
```

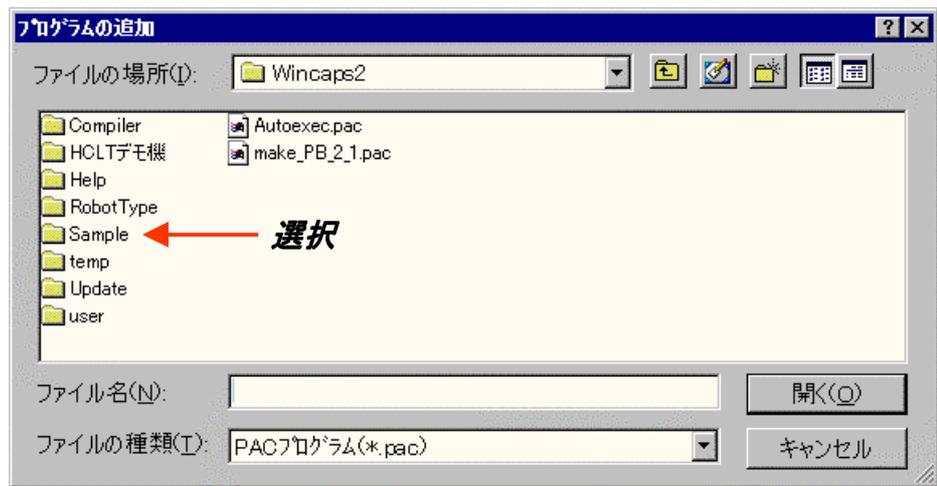
8 TP 簡易操作盤画面作成

8.1 操作盤画面作成サンプルプログラムの使い方

操作盤画面作成において、雛形となるサンプルプログラムがWINCAPS II内に登録されています。場所は¥Wincaps 2 ¥Sample ¥make_screen内にあります。これをPACマネージャにインポートするには、プログラム→インポート→“プログラムの追加”で上記フォルダへ移行し、必要なサンプルプログラムを選択してください。



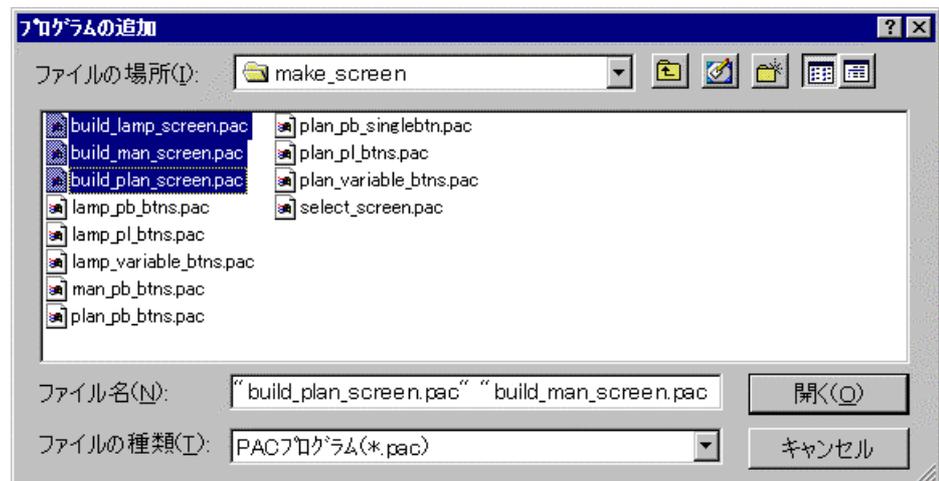
Sampleフォルダを選択します。



make_screenフォルダを選択します。



インポートしたいプログラムを選択します。



8.2 操作盤画面作成サンプルプログラム

WINCAPS II に準備されているサンプルプログラムを以下に説明します。

Select_Screen (サンプルプログラム) [Ver. 1.7 以降]

機能 汎用操作盤画面切替処理の使用例です。

説明 このプログラムは、汎用操作盤機能における画面の切替えを制御します。

各操作盤画面にて定義されている画面切替用ボタンの I/O 番号を監視し、I/O が ON した場合該当する操作盤画面を表示します。

なお、このプログラムはサンプルプログラムですので、実際の使用にあたってはプログラムを変更する必要があります。

マクロ定義 <button.h>が必要です。

関連項目 [Build_Lamp_Screen](#)、[Build_Man_Screen](#)、[Build_Plan_Screen](#)

Build_Lamp_Screen (サンプルプログラム) [Ver. 1.7 以降]

機能 下記のランプ画面を作成します。(引数として画面番号が必要です)

説明 このプログラムは、各種ボタン作成プログラムをコールして下記のランプ画面を作成します。
なお、このプログラムはサンプルプログラムですので、実際の使用にあたってはプログラムを変更する必要があります。

マクロ定義 <button.h>が必要です。

関連項目 [Lamp_pl_btns](#)、[Lamp_variable_btns](#)、[Lamp_pb_btns](#)



Build_Man_Screen (サンプルプログラム) [Ver. 1.7以降]

機能 下記の各個画面を作成します。(引数として画面番号が必要です)

説明 このプログラムは、PBボタン作成プログラムをコールして下記の各個画面を作成します。

なお、このプログラムはサンプルプログラムですので、実際の使用にあたってはプログラムを変更する必要があります。

マクロ定義 <button.h>が必要です。

関連項目 [Man_pb_btns](#)



Build_Plan_Screen (サンプルプログラム) [Ver. 1.7 以降]

機能 下記の段取画面を作成します。(引数として画面番号が必要です)

説明 このプログラムは、各種ボタン作成プログラムをコールして下記の段取画面を作成します。なお、このプログラムはサンプルプログラムですので、実際の使用にあたってはプログラムを変更する必要があります。

マクロ定義 <button.h>が必要です。

関連項目 [Plan_variable_btns](#)、[Plan_pb_btns](#)、[Plan_pl_btns](#)、[Plan_pb_singlebtn](#)



Lamp_pl_btns (サンプルプログラム) [Ver. 1.7 以降]

機能	指定画面にランプ(LED)を作成します。(引数として画面番号が必要です)
説明	ボタン作成に必要な各パラメータを入力する事によりランプボタンを生成します。 なお、このプログラムはサンプルプログラムですので、実際の使用にあたってはプログラムを変更する必要があります。
マクロ定義	<button.h>が必要です。
関連項目	make_LED

Lamp_variable_btns (サンプルプログラム) [Ver. 1.7 以降]

機能	指定画面に変数ボタン(データ表示BOX)を作成します。(引数として画面番号が必要です)
説明	ボタン作成に必要な各パラメータを入力する事により変数ボタンを生成します。 なお、このプログラムはサンプルプログラムですので、実際の使用にあたってはプログラムを変更する必要があります。
マクロ定義	<button.h>が必要です。
関連項目	make_PARAM_BOX 、 make_LABEL

Lamp_pb_btns (サンプルプログラム) [Ver. 1.7 以降]

機能	指定画面にPBボタン(画面切替ボタン)を作成します。(引数として画面番号が必要です)
説明	ボタン作成に必要な各パラメータを入力する事によりPBボタンを生成します。 なお、このプログラムはサンプルプログラムですので、実際の使用にあたってはプログラムを変更する必要があります。
マクロ定義	<button.h>が必要です。
関連項目	make_PB

Man_pb_btns (サンプルプログラム) [Ver. 1.7 以降]

機能	指定画面にPBボタンを作成します。(引数として画面番号が必要です)
説明	ボタン作成に必要な各パラメータを入力する事によりPBボタンを生成します。 なお、このプログラムはサンプルプログラムですので、実際の使用にあたってはプログラムを変更する必要があります。
マクロ定義	<button.h>が必要です。
関連項目	make_PB

Plan_variable_btns (サンプルプログラム) [Ver. 1.7 以降]

機能	指定画面に変数入力ボタン(段取パラメータ入力BOX)を作成します。(引数として画面番号が必要です)
説明	ボタン作成に必要な各パラメータを入力する事により変数ボタンを生成します。 なお、このプログラムはサンプルプログラムですので、実際の使用にあたってはプログラムを変更する必要があります。
マクロ定義	<button.h>が必要です。
関連項目	make_PARAM_BOX 、 make_LABEL

Plan_pb_btns (サンプルプログラム) [Ver. 1.7 以降]

機能	指定画面にPBボタン(データ登録処理ボタン)を作成します。(引数として画面番号が必要です)
説明	ボタン作成に必要な各パラメータを入力する事によりPBボタンを生成します。 なお、このプログラムはサンプルプログラムですので、実際の使用にあたってはプログラムを変更する必要があります。
マクロ定義	<button.h>が必要です。
関連項目	make_PB

Plan_pl_btns (サンプルプログラム) [Ver. 1.7 以降]

機能 指定画面にランプボタンを作成します。(引数として画面番号が必要です)

説明 ボタン作成に必要な各パラメータを入力する事によりランプボタンを生成します。

なお、このプログラムはサンプルプログラムですので、実際の使用にあたってはプログラムを変更する必要があります。

マクロ定義 <button.h>が必要です。

関連項目 [make_LED](#)、[make_LABEL](#)

Plan_pb_singlebtn (サンプルプログラム) [Ver. 1.7 以降]

機能 指定画面に一つだけボタンを作成します。(引数として画面番号が必要です)

説明 ボタン作成に必要な各パラメータを入力する事によりPBボタンを生成します。

なお、このプログラムはサンプルプログラムですので、実際の使用にあたってはプログラム内のパラメータを変更する必要があります。

マクロ定義 <button.h>が必要です。

関連項目 [single_button_set](#)

8.3 操作盤画面作成用のライブラリ

ティーチングペンダントのスクリーンに操作盤画面を作成するためのライブラリを以下に説明します。

make_PB (ライブラリ) [Ver. 1.7 以降]

機能	P B ボタンを作成します。
書式	make_PB
説明	指定されたパラメータ・表示文字・ボタン数を元に画面にP B ボタンを設定します。
マクロ定義	<button.h>が必要です。
用例	<code>call make_PB(sysprm(),pb_title(),PB_NUM)</code>

make_LED (ライブラリ) [Ver. 1.7 以降]

機能	LED ボタンを作成します。
書式	make_LED
説明	指定されたパラメータ・表示文字・ボタン数を元に画面にL E D ボタンを設定します。
マクロ定義	<button.h>が必要です。
用例	<code>call make_LED(sysprm(),pl_title(),MAX_PL)</code>

make_LABEL (ライブラリ) [Ver. 1.7 以降]

機能	タイトル (ラベル) を作成します。
書式	make_LABEL
説明	指定されたパラメータ・表示文字・ボタン数を元に画面にタイトルを設定します。
マクロ定義	<button.h>が必要です。
用例	<code>call make_LABEL(sysprm(),pl_title(),PL_NUM)</code>

make_PARAM_BOX (ライブラリ) [Ver. 1.7 以降]

機能	変数ボタン (入力&表示BOX) を作成します。
書式	<code>make_PARAM_BOX</code>
説明	指定されたパラメータ・ボタン数を元に画面に変数ボタンを設定します。
マクロ定義	<button.h>が必要です。
用例	<code>call make_PARAM_BOX(sysprm(), VBTN_NUM)</code>

single_button_set (ライブラリ) [Ver. 1.7 以降]

機能	ボタンを一つだけ作成します。
書式	<code>single_button_set</code>
説明	指定されたボタン番号にパラメータを設定する事によりボタンを一つ設定します。
マクロ定義	<button.h>が必要です。
用例	<code>call single_button_set(btn_no, prm())</code>

set_button_param (ライブラリ) [Ver. 1.7 以降]

- 機能** ボタン属性 (タイプ・色・形など) を指定します。
- 書式** set_button_param(<表示ボタン数>, <先頭ボタン数>, <表示ボタンタイプ>, <ボタン文字色>, <ボタン背景色>, <ボタン使用フラグ>, <ボタン可視フラグ>, <ボタン対応 I 変数番号>, <ボタン対応 I/O 番号>, <ボタン表示パネル番号>, <ボタン文字位置>)
- 説明** このプログラムは、ボタン表示に必要なパラメータを指定するプログラムです。ここでは、計 11 個のパラメータを指定する必要があります。
- マクロ定義** <button.h>が必要です。
- 用例**
- ```
call set_button_param(BUTTON_NUM, B_START_NUM, BUTTON_TYPE,
 CHAR_COLOR, BKGRND_COLOR, BUTTON_USE, BUTTON_VIS, B_VARIABLE_NUM,
 B_ASSIGN_IO, PANEL_NO, CAP_POS)
```

## arrange\_button\_size (ライブラリ) [Ver. 1.7 以降]

- 機能** 画面上のボタン配置 (位置・大きさなど) を指定します。
- 書式** arrange\_button\_size(<表示ボタン数>, <ボタン表示開始 X 座標>, <ボタン表示開始 Y 座標>, <ボタン横幅>, <ボタン高さ>, <先頭ボタン番号>, <ボタン横ギャップ>, <ボタン縦ギャップ>, <ボタン表示パネル番号>)
- 説明** このプログラムは、ボタン配置に必要なパラメータを引数とし、これを元に画面上のボタン位置を決定するプログラムです。  
arrange\_button\_pos とほぼ同様の処理ですが、ボタン作成用引数パラメータの中でボタンサイズを指定するタイプの処理です。
- マクロ定義** <button.h>が必要です。
- 用例**
- ```
call arrange_button_size(BUTTON_NUM, ORIGIN_PX, ORIGIN_PY, BUTTON_W,
    BUTTON_H, B_START_NUM, H_GAP, V_GAP, PANEL_NO)
```

arrange_button_pos (ライブラリ) [Ver. 1.7 以降]

機能

画面上のボタン配置 (位置・大きさなど) を指定します。

書式

arrange_button_pos (<表示ボタン数>, <ボタン表示開始 X 座標>, <ボタン表示開始 Y 座標>, <画面横方向のボタン数>, <画面縦方向のボタン数>, <先頭ボタン番号>, <ボタン横ギャップ>, <ボタン縦ギャップ>, <ボタン表示パネル番号>)

説明

このプログラムは、ボタン配置に必要なパラメータを引数とし、これを元に画面上のボタン位置を決定するプログラムです。

set_button_size とほぼ同様の処理ですが、ボタン作成用引数パラメータの中で、画面縦横方向におけるボタン数を指定するタイプの処理です。

マクロ定義

<button.h>が必要です。

用例

```
call arrange_button_size (BUTTON_NUM, ORIGIN_PX, ORIGIN_PY, PANEL_M,  
PANEL_N, , B_START_NUM, H_GAP, V_GAP, PANEL_NO)
```

9 視覚のライブラリ

視覚機能のライブラリを以下に説明します。

viTran6 (ライブラリ) (μ Vision ボード用)

機能 視覚座標をロボット座標に変換 (6 軸)

書式 viTran6 (<CAL 番号>, <視覚座標 X>, <視覚座標 Y>, <ロボット座標 (P 型)>)

説明 視覚の座標 (X, Y) をロボットの絶対座標 (X, Y, Z) に変換します。
このライブラリを実行するためには、事前に視覚 CAL を実行し、視覚 CAL データをコントローラに転送する必要があります。
CAL 番号には視覚 CAL を実行した番号を指定します。
この座標変換では絶対座標 (X, Y, Z) のみが求まるため、MOVE 命令を実行するためには姿勢データを別途設定する必要があります。

用例

```
' F1 には視覚座標 X を格納
' F2 には視覚座標 Y を格納
TAKEARM
CHANGETOOL 1          ' 視覚 CAL を行なったツールに設定
MOVE P, P10           ' 待機位置に移動
P11 = P10             ' 待機位置の姿勢データをコピー
CALL viTran6 (0, F1, F2, P11) ' 視覚座標をロボット座標に変換 (視覚 CAL
                          ' データの "0" を使用)

APPROACH P, P11, 100
MOVE L, P11          ' 視覚で計測した位置に移動
DEPART L, 100
CHANGETOOL 0
```

viTran6S (ライブラリ) (μ Vision-21 用)

機能 視覚座標をロボット座標に変換 (6 軸)

書式 viTran6S (<CAL 番号>, <視覚座標 X>, <視覚座標 Y>, <ロボット座標 X>, <ロボット座標 Y>, <ロボット座標 Z>)

説明 視覚の座標 (X, Y) をロボットの絶対座標 (X, Y, Z) に変換します。
このライブラリを実行するためには、事前に視覚 CAL を実行し、視覚 CAL データをコントローラに転送する必要があります。
CAL 番号には視覚 CAL を実行した番号を指定します。
この座標変換では絶対座標 (X, Y, Z) のみが求まるため、MOVE 命令を実行するためには姿勢データを別途設定する必要があります。

索引

あ

アーム動作 [33](#)

し

視覚 [93](#)

従来言語 [2](#)

つ

ツール操作 [30](#)

に

入出力 [32](#)

は

パレタイジング [17](#)

ふ

プログラムライブラリ [1](#)

デンソーロボット

垂直多関節型 V*-D/-E シリーズ
水平多関節型 H*-D/-E シリーズ
直角座標型 XYC-4D シリーズ
視覚装置 μ Vision シリーズ

プログラミングマニュアル PAC ライブラリ (Ver. 1.98)

初 版 2002 年 2 月
第 2 版 2002 年 9 月

株式会社デンソーウェーブ FA 事業部

6E**C

- この取扱説明書の一部または全部を無断で複製・転載することはお断りします。
- この説明書の内容は将来予告なしに変更することがあります。
- 本書の内容については、万全を期して作成いたしました。万が一不審の点や誤り、記載もれなど、お気づきの点がありましたら、ご連絡ください。
- 運用した結果の影響については、上項にかかわらず責任を負いかねますのでご了承ください。

