

デンソーロボット

プログラミングマニュアルⅡ
PACライブラリ

Copyright © 2009-2011 DENSO WAVE INCORPORATED
All rights reserved.

この取扱説明書の著作権は、株式会社デンソーウェーブにあります。
本書に掲載されている会社名や製品は、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

■ RC7 型コントローラ搭載

- 垂直多関節型ロボット V *シリーズ
- 水平多関節型ロボット H * -G シリーズ
- 直角座標型ロボット XYC-4G シリーズ
- 組込型ロボット XR-G シリーズ

■ オプション品

- 視覚装置 μ Vision シリーズ

注1：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」のソフトウェア 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

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

本書の構成

本書の構成は、以下のようになっております。

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 III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。

目次

アルファベット順コマンド一覧

機能別コマンド一覧

第 1 章	プログラムライブラリの使い方	1-1
第 2 章	プログラムライブラリの分類	2-1
第 3 章	従来言語のライブラリ	3-1
第 4 章	パレタイジングのライブラリ	4-1
第 5 章	ツール操作のライブラリ	5-1
第 6 章	入出力のライブラリ	6-1
第 7 章	アーム動作のライブラリ	7-1
第 8 章	TP 簡易操作盤画面作成	8-1
8.1	操作盤画面作成サンプルプログラムの使い方	8-1
8.2	操作盤画面作成サンプルプログラム	8-1
8.3	操作盤画面作成用のライブラリ	8-9
第 9 章	視覚のライブラリ	9-1

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			7-49
ArchMoveV	アーチモーションを実施します。		V3.2		7-50
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		8-12
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		8-12
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) (注1) ですべて指定します。	◎	◎		3-1
aspChange	最適可搬質量設定モードの内部モードを選択します。	◎	◎		3-2
C					
ClearCollisionForce	外力最大値を初期化します。		V2.7		7-65
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		7-55
D					
dioConstantDistanceIoOut	軸が一定の角度 (Z 軸、直動軸などは距離)、動作する度に I/O の出力状態を反転します	V3.0	V3.0		6-2
dioSync	DIO に接続された外部機器 (シーケンサなど) との同期	◎	◎		6-1
G					
GetCollisionForce	外力最大値を取得します。		V2.7		7-64
M					
make_LABEL	タイトル (ラベル) を作成します。	V1.7	V1.7		8-10
make_LED	LED ボタンを作成します。	V1.7	V1.7		8-9
make_PARAM_BOX	変数ボタン (入力&表示 BOX) を作成します。	V1.7	V1.7		8-10
make_PB	P B ボタンを作成します。	V1.7	V1.7		8-9
MotionComp	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)	V1.5	V1.5		7-48
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		7-47
mvResetPulseWidth	停止時許容パルス幅をデフォルト値にします。	◎	◎		7-1
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		7-2
mvResetTimeOut	動作終了タイムアウト値をデフォルト値にします。	◎	◎		7-3
mvReverseFlip	4 軸形態反転	◎	◎		7-3
mvSetPulseWidth	停止時許容パルス幅設定	◎	◎		7-4
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		7-5
mvSetTimeOut	動作終了タイムアウト値設定	◎	◎		7-6
N					
ndApra	ツール座標系指定の絶対動作 (4 軸ロボット専用)	◎			3-14
ndDepa	ツール座標系指定の絶対動作 (4 軸ロボット専用)	◎			3-15
ndInb	指定ポートの入力を 2 進数とみなして 10 進数に変換	◎	◎		3-3
ndJf	外部機器から OK/NG を受信し、条件分岐 (RS232C 入出力)	◎	◎		3-4
ndOnb	10 進数を 2 進数に変換して指定ポートより出力	◎	◎		3-5

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
ndOnbl	10 進数を 2 進数に変換して指定ポートより出力	◎	◎		3-6
ndTc	T C 時間設定	◎	V1.2		3-16
ndTs	T S 時間、スロー速度設定	◎	V1.2		3-16
ndVcom	外部機器との通信動作 (カーネル)。(RS232C 入出力)	◎	◎		3-13
ndVdt	外部機器から転送された変数を記憶 (RS232C 入出力)	◎	◎		3-7
ndVis	外部機器へ指定した 2 桁の整数を転送 (RS232C 入出力)	◎	◎		3-8
ndVput	外部機器へ位置姿勢を転送 (RS232C 入出力)	◎	◎		3-9
ndVrst	外部機器の初期化 (RS232C 入出力)	◎	◎		3-10
ndVset	外部機器からデータ受信 (RS232C 入出力)	◎	◎		3-11
ndVType	外部機器との通信プロトコルを指定します。(RS232C 入出力)	◎	◎		3-12

O

OffPWM	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)	◎			7-19
OffSrvLock	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)	◎			7-17
OnPWM	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)	◎			7-18
OnSrvLock	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)	◎			7-16

P

pltDecCnt	パレタイジング総カウンタのデクリメント	◎	◎		4-1
pltGetCnt	パレタイジング総カウンタの取得	◎	◎		4-1
pltGetK	パレタイジング設定値 K の取得	◎	◎		4-2
pltGetK1	パレタイジングカウンタ K1 の取得	◎	◎		4-2
pltGetM	パレタイジング設定値 M の取得	◎	◎		4-3
pltGetM1	パレタイジングカウンタ M1 の取得	◎	◎		4-3
pltGetN	パレタイジング設定値 N の取得	◎	◎		4-4
pltGetN1	パレタイジングカウンタ N1 の取得	◎	◎		4-4
pltGetNextPos	次ポジションの取得	◎	◎		4-5
pltGetPLT1END	パレタイジング 1 段終了フラグの取得	◎	◎		4-5
pltGetPLTEND	パレタイジング全段終了フラグの取得	◎	◎		4-6
pltIncCnt	パレタイジング総カウンタのインクリメント	◎	◎		4-6
pltInit1	パレタイジング初期化テンプレート 1	◎	◎		4-7
pltInitialize	パレタイジング初期化	◎	◎		4-8
pltKernel	パレタイジング動作 (カーネル)	◎	◎		4-9
pltLetCnt	パレタイジング総カウンタの設定	◎	◎		4-10
pltLetK1	パレタイジングカウンタ K1 の設定	◎	◎		4-10
pltLetM1	パレタイジングカウンタ M1 の設定	◎	◎		4-11
pltLetN1	パレタイジングカウンタ N1 の設定	◎	◎		4-11
pltMain1	パレタイジング テンプレート 1	◎	◎		4-12
pltMain2	パレタイジング テンプレート 2	◎	◎		4-12
pltMove	標準パレタイジング テンプレート 1	◎	◎		4-13
pltMove0	標準パレタイジング動作 1	◎	◎		4-14

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

コマンド	機能	4 軸	6 軸	視覚装置	説明ページ
pltResetAll	パレタイジングカウンタすべてのリセット	◎	◎		4-15
pltResetPLT1END	パレタイジング1段終了フラグのリセット	◎	◎		4-15
pltResetPLTEND	パレタイジング全段終了フラグのリセット	◎	◎		4-16
R					
ResetCollisionJnt	指定した軸の衝突検出を無効にします。		V2.7		7-63
ResetCompControl	力制限機能を無効にします。(6軸専用命令)		V1.4		7-29
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。(6軸専用命令)		V1.4		7-42
ResetCompJLimit	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)		V1.4		7-38
ResetCompRate	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)		V1.4		7-34
ResetCompVMode	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)		V1.4		7-40
ResetCurLmt	指定した軸のモータ電流制限を解除します。	◎	V1.2		7-13
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		7-22
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		7-23
ResetDampRate	力制限時の粘性割合を初期化します。(6軸専用命令)		V1.4		7-44
ResetEralw	指定した軸の偏差許容値をデフォルト値に戻します。	◎	V1.2		7-15
ResetFrcAssist	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)		V1.4		7-36
ResetFrcLimit	力制限割合を初期化します。(6軸専用命令)		V1.4		7-32
ResetGravity	重力バランスを無効にします。		V1.2		7-8
ResetGrvOffset	重力補償値の補正を無効にします。		V1.2		7-10
ResetHighPathAccuracy	CP動作(直線、円弧、自由曲線)時の動作軌跡を向上する高軌跡制御機能を無効にします。	V2.61	V2.61		7-60
ResetVibControl	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。	V1.4	V1.4		7-46
S					
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		8-11
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			7-51
SetCollisionJnt	指定した軸の衝突検出を有効にします。		V2.7		7-62
SetCollisionLevel	指定した軸の衝突検出レベルを設定します。		V2.7		7-66
SetCompControl	力制限機能を有効にします。(6軸専用命令)		V1.4		7-27
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。(6軸専用命令)		V1.4		7-41
SetCompFControl	力制限機能を有効にします。(6軸専用命令)		V1.4		7-28
SetCompJLimit	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)		V1.4		7-37
SetCompRate	力制限時の柔らかさの割合を設定します。(6軸専用命令)		V1.4		7-33
SetCompVMode	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)		V1.4		7-39
SetCPSpdMode	CP動作時のTCP速度設定を変更する。	V1.8	V1.8		7-24
SetCurLmt	指定した軸のモータ電流値を制限します。	○	V1.2		7-11

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

コマンド	機能	4 軸	6 軸	視覚装置	説明ページ
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		7-20
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		7-21
SetDampRate	力制限時の粘性割合を設定します。(6軸専用命令)		V1.4		7-43
SetEralw	指定した軸の偏差許容値を変更します。	◎	V1.2		7-14
SetExtForceDetect	外力検出の有効、無効を設定します。		V2.7		7-67
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。	◎			7-26
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。	◎			7-25
SetFrcAssist	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)		V1.4		7-35
SetFrcCoord	力制限設定座標系を選択します。(6軸専用命令)		V1.4		7-30
SetFrcLimit	力制限割合を設定します。(6軸専用命令)		V1.4		7-31
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		7-7
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		7-9
SetHighPathAccuracy	CP 動作(直線、円弧、自由曲線)時の動作軌跡を向上する高軌跡制御機能に移行します。	V2.61	V2.61		7-59
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。(Ver. 1.5 以降対応)	V1.5	V1.5		7-52
SetSingularAvoid	特異点回避機能を有効化、または無効化します。(6軸専用)		V2.61		7-61
SetVibControl	残留振動低減制御モードにします。	V1.4	V1.4		7-45
single_button_set	ボタンを一つだけ作成します。	V1.7	V1.7		8-11
StartSrvMonitor	サーボ単軸データモニタを開始します。(Ver. 1.5 以降対応)	V1.5	V1.5		7-53
StopSrvMonitor	サーボ単軸データモニタを終了します。(Ver. 1.5 以降対応)	V1.5	V1.5		7-54
T					
tolChange	ツールチェンジ	◎	◎		5-1
tolInit1	ツールチェンジ初期化テンプレート 1	◎	◎		5-1
tolInitialize	ツールチェンジ初期化	◎	◎		5-2
tolKernel	ツールチェンジ動作(カーネル)	◎	◎		5-2
tolMain1	ツールチェンジテンプレート 1	◎	◎		5-3
V					
viTran6	視覚座標をロボット座標に変換(6軸)	◎	◎	◎	9-1
viTran6S	視覚座標をロボット座標に変換(6軸)				9-2
X					
xdSPLClrTakeArm	TakeArm 時に実行する自由曲線の通過点消去処理の有効、無効を変更する。	V2.3	V2.3		7-58
xdSPLPASSNUM	自由曲線が通過済の通過点番号を取得する。	V2.3	V2.3		7-57
xdWAITSPLINE	自由曲線が指定した通過点を通過するのを待つ。	V2.3	V2.3		7-56

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
従来言語のライブラリ						
	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) (注 1) ですべて指定します。	◎	◎		3-1
	aspChange	最適可搬質量設定モードの内部モードを選択します。	◎	◎		3-2
	ndInb	指定ポートの入力を 2 進数とみなして 10 進数に変換	◎	◎		3-3
	ndJf	外部機器から OK/NG を受信し、条件分岐 (RS232C 入出力)	◎	◎		3-4
	ndOnb	10 進数を 2 進数に変換して指定ポートより出力	◎	◎		3-5
	ndOnbI	10 進数を 2 進数に変換して指定ポートより出力	◎	◎		3-6
	ndVdt	外部機器から転送された変数を記憶 (RS232C 入出力)	◎	◎		3-7
	ndVis	外部機器へ指定した 2 桁の整数を転送 (RS232C 入出力)	◎	◎		3-8
	ndVput	外部機器へ位置姿勢を転送 (RS232C 入出力)	◎	◎		3-9
	ndVrst	外部機器の初期化 (RS232C 入出力)	◎	◎		3-10
	ndVset	外部機器からデータ受信 (RS232C 入出力)	◎	◎		3-11
	ndVType	外部機器との通信プロトコルを指定します。(RS232C 入出力)	◎	◎		3-12
	ndVcom	外部機器との通信動作 (カーネル)。(RS232C 入出力)	◎	◎		3-13
	ndApra	ツール座標系指定の絶対動作 (4 軸ロボット専用)	◎			3-14
	ndDepa	ツール座標系指定の絶対動作 (4 軸ロボット専用)	◎			3-15
	ndTc	T C 時間設定	◎	V1.2		3-16
	ndTs	T S 時間、スロー速度設定	◎	V1.2		3-16
パレタイジングのライブラリ						
	pltDecCnt	パレタイジング総カウンタのデクリメント	◎	◎		4-1
	pltGetCnt	パレタイジング総カウンタの取得	◎	◎		4-1
	pltGetK	パレタイジング設定値 K の取得	◎	◎		4-2
	pltGetK1	パレタイジングカウンタ K1 の取得	◎	◎		4-2
	pltGetM	パレタイジング設定値 M の取得	◎	◎		4-3
	pltGetM1	パレタイジングカウンタ M1 の取得	◎	◎		4-3
	pltGetN	パレタイジング設定値 N の取得	◎	◎		4-4
	pltGetN1	パレタイジングカウンタ N1 の取得	◎	◎		4-4
	pltGetNextPos	次ポジションの取得	◎	◎		4-5
	pltGetPLT1END	パレタイジング 1 段終了フラグの取得	◎	◎		4-5

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
	pltGetPLTEND	パレタイジング全段終了フラグの取得	◎	◎		4-6
	pltIncCnt	パレタイジング総カウンタのインクリメント	◎	◎		4-6
	pltInit1	パレタイジング初期化テンプレート 1	◎	◎		4-7
	pltInitialize	パレタイジング初期化	◎	◎		4-8
	pltKernel	パレタイジング動作 (カーネル)	◎	◎		4-9
	pltLetCnt	パレタイジング総カウンタの設定	◎	◎		4-10
	pltLetK1	パレタイジングカウンタ K1 の設定	◎	◎		4-10
	pltLetM1	パレタイジングカウンタ M1 の設定	◎	◎		4-11
	pltLetN1	パレタイジングカウンタ N1 の設定	◎	◎		4-11
	pltMain1	パレタイジング テンプレート 1	◎	◎		4-12
	pltMain2	パレタイジング テンプレート 2	◎	◎		4-12
	pltMove	標準パレタイジング テンプレート 1	◎	◎		4-13
	pltMove0	標準パレタイジング動作 1	◎	◎		4-14
	pltResetAll	パレタイジングカウンタすべてのリセット	◎	◎		4-15
	pltResetPLT1END	パレタイジング 1 段終了フラグのリセット	◎	◎		4-15
	pltResetPLTEND	パレタイジング全段終了フラグのリセット	◎	◎		4-16
ツール操作のライブラリ						
	tolChange	ツールチェンジ	◎	◎		5-1
	tolInit1	ツールチェンジ初期化テンプレート 1	◎	◎		5-1
	tolInitialize	ツールチェンジ初期化	◎	◎		5-2
	tolKernel	ツールチェンジ動作 (カーネル)	◎	◎		5-2
	tolMain1	ツールチェンジテンプレート 1	◎	◎		5-3
入出力のライブラリ						
	dioSync	DIO に接続された外部機器 (シーケンサなど) との同期	◎	◎		6-1
	dioConstantDistanceIoOut	軸が一定の角度 (Z 軸、直動軸などは距離)、動作する度に I/O の出力状態を反転します	V3.0	V3.0		6-2
アーム動作のライブラリ						
	mvResetPulseWidth	停止時許容パルス幅をデフォルト値にします。	◎	◎		7-1
	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		7-2
	mvResetTimeOut	動作終了タイムアウト値をデフォルト値にします。	◎	◎		7-3
	mvReverseFlip	4 軸形態反転	◎	◎		7-3
	mvSetPulseWidth	停止時許容パルス幅設定	◎	◎		7-4
	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		7-5
	mvSetTimeOut	動作終了タイムアウト値設定	◎	◎		7-6
	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		7-7
	ResetGravity	重力バランスを無効にします。		V1.2		7-8

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

機能区分	コマンド	機能	4 軸	6 軸	視覚装置	説明ページ
	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		7-9
	ResetGrvOffset	重力補償値の補正を無効にします。		V1.2		7-10
	SetCurLmt	指定した軸のモータ電流値を制限します。	○	V1.2		7-11
	ResetCurLmt	指定した軸のモータ電流制限を解除します。	◎	V1.2		7-13
	SetEralw	指定した軸の偏差許容値を変更します。	◎	V1.2		7-14
	ResetEralw	指定した軸の偏差許容値をデフォルト値に戻します。	◎	V1.2		7-15
	OnSrvLock	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)	◎			7-16
	OffSrvLock	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)	◎			7-17
	OnPWM	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)	◎			7-18
	OffPWM	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)	◎			7-19
	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		7-20
	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		7-21
	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		7-22
	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		7-23
	SetCPSpdMode	CP 動作時の TCP 速度設定を変更する。	V1.8	V1.8		7-24
	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。	◎			7-25
	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。	◎			7-26
	SetCompControl	力制限機能を有効にします。(6軸専用命令)		V1.4		7-27
	SetCompFControl	力制限機能を有効にします。(6軸専用命令)		V1.4		7-28
	ResetCompControl	力制限機能を無効にします。(6軸専用命令)		V1.4		7-29
	SetFrcCoord	力制限設定座標系を選択します。(6軸専用命令)		V1.4		7-30
	SetFrcLimit	力制限割合を設定します。(6軸専用命令)		V1.4		7-31
	ResetFrcLimit	力制限割合を初期化します。(6軸専用命令)		V1.4		7-32

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
	SetCompRate	力制限時の柔らかさの割合を設定します。(6軸専用命令)		V1.4		7-33
	ResetCompRate	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)		V1.4		7-34
	SetFrcAssist	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)		V1.4		7-35
	ResetFrcAssist	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)		V1.4		7-36
	SetCompJLimit	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)		V1.4		7-37
	ResetCompJLimit	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)		V1.4		7-38
	SetCompVMode	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)		V1.4		7-39
	ResetCompVMode	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)		V1.4		7-40
	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。(6軸専用命令)		V1.4		7-41
	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。(6軸専用命令)		V1.4		7-42
	SetDampRate	力制限時の粘性割合を設定します。(6軸専用命令)		V1.4		7-43
	ResetDampRate	力制限時の粘性割合を初期化します。(6軸専用命令)		V1.4		7-44
	SetVibControl	残留振動低減制御モードにします。	V1.4	V1.4		7-45
	ResetVibControl	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。	V1.4	V1.4		7-46
	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		7-47
	MotionComp	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)	V1.5	V1.5		7-48
	ArchMove	アーチモーションを実施します。	V1.9			7-49
	ArchMoveV	アーチモーションを実施します。		V3.2		7-50
	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			7-51
	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。(Ver. 1.5以降対応)	V1.5	V1.5		7-52
	StartSrvMonitor	サーボ単軸データモニタを開始します。(Ver. 1.5以降対応)	V1.5	V1.5		7-53
	StopSrvMonitor	サーボ単軸データモニタを終了します。(Ver. 1.5以降対応)	V1.5	V1.5		7-54

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
	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		7-55
	xdWAITSPLINE	自由曲線が指定した通過点を通過するのを待つ。	V2.3	V2.3		7-56
	xdSPLPASSNUM	自由曲線が通過済の通過点番号を取得する。	V2.3	V2.3		7-57
	xdSPLClrTakeArm	TakeArm 時に実行する自由曲線の通過点消去処理の有効、無効を変更する。	V2.3	V2.3		7-58
	SetHighPathAccuracy	CP 動作 (直線、円弧、自由曲線) 時の動作軌跡を向上する高軌跡制御機能に移行します。	V2.61	V2.61		7-59
	ResetHighPathAccuracy	CP 動作 (直線、円弧、自由曲線) 時の動作軌跡を向上する高軌跡制御機能を無効にします。	V2.61	V2.61		7-60
	SetSingularAvoid	特異点回避機能を有効化、または無効化します。(6 軸専用)		V2.61		7-61
	SetCollisionJnt	指定した軸の衝突検出を有効にします。		V2.7		7-62
	ResetCollisionJnt	指定した軸の衝突検出を無効にします。		V2.7		7-63
	GetCollisionForce	外力最大値を取得します。		V2.7		7-64
	ClearCollisionForce	外力最大値を初期化します。		V2.7		7-65
	SetCollisionLevel	指定した軸の衝突検出レベルを設定します。		V2.7		7-66
	SetExtForceDetect	外力検出の有効、無効を設定します。		V2.7		7-67
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		8-9
	make_LED	LED ボタンを作成します。	V1.7	V1.7		8-9
	make_LABEL	タイトル (ラベル) を作成します。	V1.7	V1.7		8-10
	make_PARAM_BOX	変数ボタン (入力&表示 BOX) を作成します。	V1.7	V1.7		8-10
	single_button_set	ボタンを一つだけ作成します。	V1.7	V1.7		8-11
	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		8-11
	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		8-12
	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		8-12
視覚のライブラリ						
	viTran6	視覚座標をロボット座標に変換 (6 軸)	◎	◎	◎	9-1
	viTran6S	視覚座標をロボット座標に変換 (6 軸)				9-2

第 1 章 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 IIIのプログラムバンクを起動し、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 IIIガイドを参照してください。

第2章 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 III 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
!DW00 : 従来言語	従来言語のコマンドと同等機能を提供します。
!DW01 : パレタイジング	パレタイジング機能を提供します。
!DW02 : ツール操作	ツール操作関連の機能を提供します。
!DW03 : 入出力	DIO、RS232C 入出力関連の機能を提供します。
!DW04 : Arm_ コンプライアンス	アーム動作関連のコンプライアンス機能を提供します。
!DW05 : Arm_ コンベアトラッキング	アーム動作関連のコンベアトラッキング機能を提供します。
!DW06 : Arm_ 衝突検出	アーム動作関連の衝突検出機能を提供します。
!DW07 : Arm_ 自由曲線	アーム動作関連の自由曲線機能を提供します。
!DW08 : Arm_ 機能 On/Off	アーム動作関連のその他機能の ON/OFF に関するプログラムを提供します。
!DW09 : Arm_ その他	アーム動作関連のその他の機能を提供します。
!DW10 : 視覚	視覚動作関連の機能を提供します。
!DW11 : その他	旧バージョンに対応する機能などを提供します。

第 3 章 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

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。

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。

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。

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ент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ент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。

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。

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。

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」

注意事項

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

第3章 従来言語のライブラリ

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。

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」

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。

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。

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 >, < 最下位桁入力ポート番号 >, < 最上位桁入力ポート番号 >)

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

ndOnb、ndOnbI

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(15,104,119)

’出力ポート 104 ~ 119 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。

ndOnbl (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

ndOnbl (<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(1,104,119)

’出力ポート104～119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
    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)
  CALL ndVrst
  CALL ndVset(0)
  CALL ndVis(3)
  CALL ndJf(3, JF_VAL)
  IF JF_VAL = TRUE THEN
    CALL ndVset(3)
    CALL ndVdt(pacPOS, 1)
    CALL ndVdt(pacJNT, 1)
    CALL ndVdt(pacTRN, 1)
  END IF
END
```

- ’ プロトコルを設定 (旧 =0/ 新 =1)
- ’ 外部機器の初期化
- ’ 受信データ (VDT) を 0 にクリア
- ’ 外部機器を起動 (03 を送信)
- ’ 外部機器の応答結果取得 (03 を送信)
- ’ 応答が OK(TRUE) ならばデータ受信
- ’ 外部機器より 10 個のデータを受信 (03 を送信)
- ’ 外部機器より受信したデータを変数 (P1) に代入
- ’ 外部機器より受信したデータを変数 (J1) に代入
- ’ 外部機器より受信したデータを変数 (T1) に代入

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ウェアバージョンが 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。

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。

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 PTP 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400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400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 400 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 400 の座標に移動
  GIVEARM
END
```

ndTc (ライブラリ)【Ver.1.2 以降】

機能

TC 時間設定

書式

ndTc (<TC 時間 >)

説明

従来言語の TC 命令と同等な機能を提供します。

TC 時間を設定します。

0 秒～600 秒の間で設定できます。工場出荷状態時点では60 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

TS 時間、スロー速度設定

書式

ndTS (<TS 時間 >, <スロー速度 >)

説明

従来言語の TS 命令と同等な機能を提供します。

TS 時間、スロー速度を設定します。

TS 時間は3 秒～30 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 %～10 %の間で設定できます。工場出荷時点では10 %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

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。

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段階終了フラグの取得

書式

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

説明

パレタイジング1段階終了フラグをI[<1段階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

第4章 パレタイジングのライブラリ

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。
```


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:IO[40]、アンチャック:IO[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 IO の先頭番号

関連項目

tolChange、tolInitialize、tolKernel

dioConstantDistanceIoOut (ライブラリ) 【Ver.3.0 以降】

機能

軸が一定の角度 (Z 軸、直動軸などは距離)、動作する度に I/O の出力状態を反転します

書式

dioConstantDistanceIoOut(< 検出周期 >)

説明

あらかじめ各軸の角度 (距離) と I/O 出力ポート番号を設定しておき、< 検出周期 > を 1 以上にしてこのライブラリを実行すると、設定した角度 (距離) 分、動作する度に設定した I/O 出力ポートの出力状態を反転します。

< 検出周期 > を 0 にしてこのライブラリを実行すると、I/O の出力状態を反転しなくなります。

角度 (距離) と I/O 出力ポート番号の設定は、各軸毎に設定できます。

ティーチングペンダントまたは WINCAPS III で行います。

- 操作経路：ティーチングペンダント
[F4 I/O] - [F6 補助機能] - [F1 ハード設定]
- 操作経路：WINCAPS III
「プロジェクト」メニューから「パラメータ」を選択し、「I/O」タブをクリックします。

項目	No. 注1	パラメータ名称	説明
1 軸から 8 軸の I/O 出力ポート番号	85	位置出力 出力 IO 番号 (J1)	出力 I/O のポート番号を入力してください。
	86	位置出力 出力 IO 番号 (J2)	
	:	:	
	91	位置出力 出力 IO 番号 (J7)	
	92	位置出力 出力 IO 番号 (J8)	
1 軸から 8 軸の角度 (距離)	93	位置出力 検出角度 (J1 deg*10 ⁵)	指定の角度に 100000 を掛算した値を入力してください。0 にすると、その軸の出力 I/O ポートは反転しません。
	94	位置出力 検出角度 (J2 deg*10 ⁵)	
	:	:	
	99	位置出力 検出角度 (J7 deg*10 ⁵)	
	100	位置出力 検出角度 (J8 deg*10 ⁵)	

注 1：WINCAPS III では 1 足した番号になります。

< 検出周期 > に設定した値は下記計算式により実際の周期に換算し、その周期で各軸の角度 (距離) を検出します。

また、< 検出周期 > の設定範囲は 0 ~ 32 です。

$$\text{周期} = 250 \mu \text{sec} \times \text{< 検出周期 >}$$

マクロ定義

<pacman.h> ファイルが必要です。

注意事項

< 検出周期 > を 1 以上にしてこのライブラリを実行すると、設定した I/O 出力ポートを OFF にしてから検出を開始します。

この機能はオプションです。機能ライセンスを購入してください。

第 6 章 入出力のライブラリ

用 例

```
TAKEARM
DRIVEA @E (4, -180)           '4 軸を初期位置に戻します。
CALL dioConstantDistanceIoOut(1) '検出開始 (250usec 周期)
DRIVEA @E (4, 180)           '4 軸を 360 度動作させます。
CALL dioConstantDistanceIoOut(0) '検出停止
```

第 7 章 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。
```


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。

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。

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。
```

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ー」となります。

マクロ定義

<pacman.h> が必要です。

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ー」が発生する場合があります。

用例

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

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。

関連項目

SetCurLmt、ResetGravity、SetGrvOffset

注意事項

- 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。
- 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。
- 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。
- 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

```
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。

関連項目

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。

関連項目

SetCurLmt、SetGravity、ResetGrvOffset

注意事項

- 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。
- 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。
- 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。
- 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値が 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。

関連項目

SetGrvOffset

注意事項

- 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。
- 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。

関連項目

ResetCurLmt、SetGravity、SetGrvOffset、SetEralw

注意事項

- 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。
- 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。
- 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。
- 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。
- 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

- 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。
- 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。
- 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。
- 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。

第7章 アーム動作のライブラリ

4 軸、HS-E

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

- 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。
- 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
```

[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。

関連項目

SetCurLmt、ResetEralw

注意事項

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。

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。

関連項目

OffSrvLock

注意事項

- 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。
- 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 (<指定軸>)

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。

関連項目

OnSrvLock

注意事項

- 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。
- 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。
- 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。
- 使用条件の "25: 電流制限リセット" 設定値の 2 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 "+2" を設定して下さい。
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 OffSrvLock(1) ; 1軸のサーボロック解除

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。

関連項目

OffPWM

注意事項

- 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。
- 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。

関連項目

OnSrvLock

注意事項

- 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。
- 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。
- 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。
- 使用条件の "25: 電流制限リセット" 設定値の 3 bit 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。
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

- 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。
- 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。
- 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。
- 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。
- 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。

関連項目

Setcycloid

注意事項

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 CP 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

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。

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。

マクロ定義

<pacman.h> ファイルが必要です。

関連項目

SetForce_HC、SetCurLmt、ResetCurLmt

注意事項

- 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。
- 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。
- 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。
- 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。
- 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 "OffSrvlock" の注意事項を参照してください。
- 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。
- 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

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。

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。

マクロ定義

<pacman.h> ファイルが必要です。

関連項目

SetForce_HM、SetCurLmt、ResetCurLmt

注意事項

- 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。
- 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。
- 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。
- 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。
- 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 "OffSrvlock" の注意事項を参照してください。
- 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。
- 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。

関連項目

SetFrcLimit、SetCompRate、SetFrcCoord、ResetCompControl、SetCompFControl

注意事項

- 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。
- 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。
- 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。
- 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。
- 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。
- 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。
- SetCompControl 実行後、ロボットの姿勢が大きく変化した場合は、重力補償の補正值に誤差が生じ、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。

関連項目

SetCompControl

注意事項

- 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。
- 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。
- 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。
- 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。
- 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(100,0,100,100,100,100)
                             ' 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。

関連項目

SetCompControl

注意事項

- 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。
- 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。
- 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。
- 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 \D× 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。
- 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軸専用命令応)

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。

関連項目

SetFrcLimit、SetCompRate、SetFrcCoord、ResetCompControl

注意事項

- 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。
- 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。
- <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、力制限有効設定 (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。
- <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、力制限有効設定 (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。
- 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。

関連項目

ResetFrcLimit、SetFrcCoord、SetCompControl

注意事項

- 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。
- 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。
- 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(100,0,100,100,100,100)
                                ' 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。

関連項目

SetFrcLimit

注意事項

- 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。
- 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

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。

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。

マクロ定義

<pacman.h> ファイルが必要です。

関連項目

ResetCompRate、SetFrcCoord、SetCompControl

注意事項

- 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。
- 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。
- 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 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。

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。

関連項目

SetCompRate

注意事項

- 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。
- 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。

関連項目

ResetFrcAssist、SetFrcCoord、SetCompControl

注意事項

- 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。
- 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。
- 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(-30,0,0,0,0,0)
                                '-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。

関連項目

SetFrcAssist

注意事項

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 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。

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

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。
設定値は 0 ~ 100 の範囲で設定します。

マクロ定義

<pacman.h> ファイルが必要です。

関連項目

ResetCompJLimit、SetCompControl

注意事項

- 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。
- 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。
- 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

```
CALL SetFrcCoord(1)           ' 力制限座標系をツール座標に設定する。
CALL SetCompJLimit(30,0,0,0,0,0)
                                ' 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。
```

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に設定します。

マクロ定義

<pacman.h> ファイルが必要です。

関連項目

SetCompJLimit、SetCompControl

注意事項

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 ResetCompControl	・力機能を無効にする。
CALL ResetCompJLimit	・電流制限値を初期化する。

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。

関連項目

ResetCompVMode

注意事項

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 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。

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。

関連項目

SetCompVMode

注意事項

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。

関連項目

ResetCompEralw、SetFrcCoord

注意事項

- 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。
- 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。
- 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。
- 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。

関連項目

SetCompEralw

注意事項

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

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。

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。

マクロ定義

<pacman.h> ファイルが必要です。

関連項目

ResetDampRate、SetFrcCoord、SetCompRate

注意事項

- 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。
- 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。
- 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。
- 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。
- 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

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。

ResetDampRate (ライブラリ)【Ver.1.4 以降】

機能

力制限時の粘性割合を初期化します。(6軸専用命令)

書式

ResetDampRate

説明

X軸方向、Y軸方向、Z軸方向、X軸回り、Y軸回り、Z軸回りの粘性割合を初期化し、すべて100にします。

マクロ定義

<pacman.h> ファイルが必要です。

関連項目

SetDampRate

注意事項

- 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。
- 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 ResetDampRate ’粘性割合を初期化する。

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。

関連項目

ResetVibControl

注意事項

- 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。
- 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せん。再度 SetEralw にて角度偏差許容値を再設定してください。
- 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

- ・ 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。
- ・ 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
defsnrg 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

- ・ 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。
- ・ 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
- ・ 瞬時停止中は、動作中と判断します。
- ・ < 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

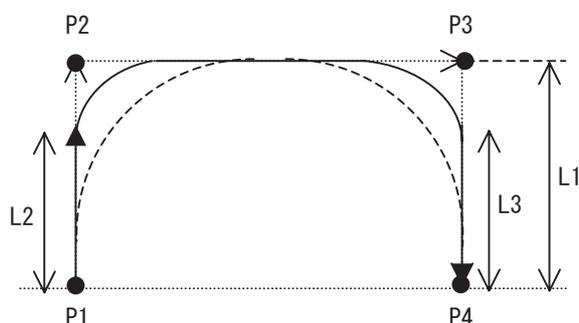
ArchMove (< 目標位置 (P 型変数) >, < Z 軸動作量 >)

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。

関連項目

SetArchParam

注意事項

- 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。
- 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。
- 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。
- 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。

ArchMoveV (ライブラリ) 【Ver.3.2 以降の6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

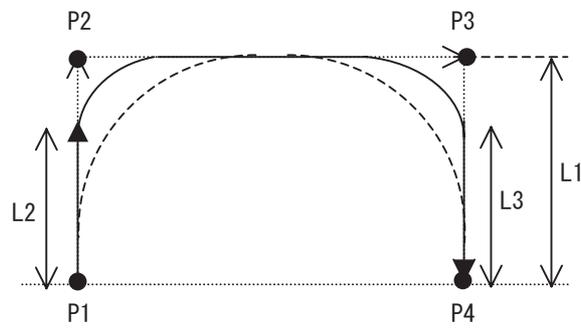
ArchMoveV (<目標位置 (P 型変数)>、<アプローチ長>)

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<アプローチ長>(L1) 上昇させた位置となります。経由点 P2 は現在位置からアプローチ長 (L1) だけ上昇させた位置となります。

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。

関連項目

SetArchParam

注意事項

- 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。
- 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。
- 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。
- 動作条件によっては、速度異常、加速度異常が発生し、動作できない場合があります。その際は、速度を下げるなど、動作条件を見直してください。

用例

CALL ArchMoveV (P10,100) ' 目標位置 P10、アプローチ長 100mm で動作します。

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

- ArchMove を実行する場合のアーチの形状を設定できます。
- 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。
- 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。
- 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。

関連項目

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。
.
.
.
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

- 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。
- 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

- 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。
- 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。
- 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

- 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。
- 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タを初期化します。(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タを初期化し、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。

xdWAITSPLINE (ライブラリ) 【Ver.2.3 以降】

機能

自由曲線が指定した通過点を通過するのを待つ。

書式

xdWAITSPLINE (<通過点番号>、<待ち条件>)

説明

実行中の自由曲線動作に対し、<通過点番号>で指定したポイントを通過するまで待ちます。

<待ち条件>に0を指定した場合、指令値が指定した通過点を通過するまで待ちます。

0以外を指定した場合、エンコーダ値が指定した通過点を通過するまで待ちます。

関連項目

MOVE、SETSPLINEPOINT

注意事項

自由曲線未実行状態で xdWAITSPLINE を CALL した場合、待ち状態になりません。

また、<通過点番号>で指定した通過点を通過後に xdWAITSPLINE を CALL した場合、待ち状態になりません。

用例

```
PROGRAM PRO1
TAKEARM
CLRSPLINEPOINT 5
SETSPLINEPOINT 5, P4
SETSPLINEPOINT 5, P1
SETSPLINEPOINT 5, J5
MOVE S, 5, NEXT
CALL xdWAITSPLINE(1,1)
SET IO[240]
CALL xdWAITSPLINE(2,1)
RESET IO[240]
```

- ’ 軌道番号5番の自由曲線で通過点をすべてクリアする。
- ’ 軌道番号5番の自由曲線で P4 を1番目の通過点とする。
- ’ 軌道番号5番の自由曲線で P1 を2番目の通過点とする。
- ’ 軌道番号5番の自由曲線で J5 を3番目の通過点とする。
- ’ P4,P1 を通過し J5 へ移動する自由曲線動作を実行する。
- ’ ロボットが1番目の通過点 (P4) を通過するのを待つ。
- ’ ポート240をONします。
- ’ ロボットが2番目の通過点 (P1) を通過するのを待つ。
- ’ ポート240をOFFします。

xdSPLPASSNUM (ライブラリ)【Ver.2.3 以降】

機能

自由曲線が通過済の通過点番号を取得する。

書式

xdSPLPASSNUM (<通過点番号>)

説明

実行中の自由曲線動作に対し、指令値が通過済の通過点番号を<通過点番号>にセットします。

関連項目

MOVE、SETSPLINEPOINT

用例

PROGRAM PRO1

TAKEARM

CLRSPLINEPOINT 5

SETSPLINEPOINT 5, P4

SETSPLINEPOINT 5, P1

SETSPLINEPOINT 5, J5

MOVE S, 5, NEXT

DELAY 500

CALL xdSPLPASSNUM(I1)

- ’ 軌道番号 5 番の自由曲線で通過点をすべてクリアする。
- ’ 軌道番号 5 番の自由曲線で P4 を 1 番目の通過点とする。
- ’ 軌道番号 5 番の自由曲線で P1 を 2 番目の通過点とする。
- ’ 軌道番号 5 番の自由曲線で J5 を 3 番目の通過点とする。
- ’ P4,P1 を通過し J5 へ移動する自由曲線動作を実行する。
- ’ 通過済の通過点番号を I1 にセットする。

xdSPLClrTakeArm (ライブラリ) 【Ver.2.3 以降】

機能

TakeArm 時に実行する自由曲線の通過点消去処理の有効、無効を変更する。

書式

xdSPLClrTakeArm (<設定値>)

説明

<設定値>が 0 の時、TakeArm 時に自由曲線の通過点を消去しない。

<設定値>が 0 以外の時、TakeArm 時に自由曲線の通過点を消去する。

関連項目

CLRSPLINEPOINT、SETSPLINEPOINT

注意事項

コントローラ電源投入後の初期状態は、TakeArm 時に自由曲線の通過点を消去します。よって、自由曲線動作プログラムとは別のプログラムで自由曲線の通過点を登録する（初期化プログラムで登録する）場合は、CALL xdSPLClrTakeArm(0) にて、TakeArm 時に自由曲線の通過点を消去しないように設定してください。

用例

```
PROGRAM INITIAL
```

```
CLRSPLINEPOINT 5
```

```
SETSPLINEPOINT 5, P4
```

```
SETSPLINEPOINT 5, P1
```

```
SETSPLINEPOINT 5, J5
```

```
CALL xdSPLClrTakeArm(0)
```

’軌道番号 5 番の自由曲線で通過点をすべてクリアする。

’軌道番号 5 番の自由曲線で P4 を 1 番目の通過点とする。

’軌道番号 5 番の自由曲線で P1 を 2 番目の通過点とする。

’軌道番号 5 番の自由曲線で J5 を 3 番目の通過点とする。

’TakeArm 時に自由曲線の通過点を消去しない。

SetHighPathAccuracy (ライブラリ)【Ver.2.61 以降】

機能

CP 動作（直線、円弧、自由曲線）時の動作軌跡を向上する高軌跡制御機能に移行します。

書式

SetHighPathAccuracy

説明

- (1) 高軌跡制御機能では、動作軌跡を向上させて動作させることができます。
- (2) SetHighPathAccuracy 直後の動作命令より、高軌跡制御機能が有効となり、ResetHighPathAccuracy で高軌跡制御機能が解除されます。

マクロ定義

<pacman.h> ファイルが必要です。

関連項目

ResetHighPathAccuracy

注意事項

- 動作軌跡を向上させたい動作のみ、高軌跡制御機能を有効にしてください。
- 動作条件、負荷条件によっては振動しながらの動作になる場合があります。その場合、速度や加速度を下げたり、機能を無効にしたりして下さい。
- コントローラ電源オフまたはモーターオフで高軌跡制御機能は無効になります。
- 付加軸オプションの軸（7, 8 軸）には対応していません。
- 力制限機能有効時には高軌跡制御機能を有効にすることはできません。エラーが発生します。
- 動作中（パス動作中など）に高軌跡制御機能を有効にすることはできません。
- 負荷設定（先端負荷質量、先端負荷重心位置など）を正確に設定してください。正確に設定されていない場合、精度が向上しない原因となります。
- 高軌跡制御機能は旧エンジンボードでは機能しません。

用例

```
PROGRAM PRO1
TAKEARM
MOVE P, @E P1
CALL SetHighPathAccuracy
MOVE S, 5
CALL ResetHighPathAccuracy
MOVE P, @0 P5
END
```

’アームグループの制御権を取得します。
’P1 へ PTP 動作で移動します。
’高軌跡制御機能を有効にします。
’軌道番号 5 を自由曲線動作で高軌跡移動します。
’高軌跡制御機能を無効にします。
’P4 から P5 へ PTP 動作で移動します。
’プログラム終了します。

ResetHighPathAccuracy (ライブラリ)【Ver.2.61以降】

機能

CP動作（直線、円弧、自由曲線）時の動作軌跡を向上する高軌跡制御機能を無効にします。

書式

ResetHighPathAccuracy

説明

- (1) 高軌跡制御機能では、動作軌跡を向上させて動作させることができます。
- (2) SetHighPathAccuracy直後の動作命令より、高軌跡制御機能が有効となり、ResetHighPathAccuracyで高軌跡制御機能が解除されます。

マクロ定義

<pacman.h> ファイルが必要です。

関連項目

SetHighPathAccuracy

注意事項

- 動作軌跡を向上させたい動作のみ、高軌跡制御機能を有効にしてください。
- 動作条件、負荷条件によっては振動しながらの動作になる場合があります。その場合、速度や加速度を下げたり、機能を無効にしたりして下さい。
- コントローラ電源オフまたはモーターオフで高軌跡制御機能は無効になります。
- 付加軸オプションの軸（7, 8軸）には対応していません。
- 力制限機能有効時には高軌跡制御機能を有効にすることはできません。エラーが発生します。
- 動作中（パス動作中など）に高軌跡制御機能を有効にすることはできません。
- 負荷設定（先端負荷質量、先端負荷重心位置など）を正確に設定してください。正確に設定されていない場合、精度が向上しない原因となります。
- 高軌跡制御機能は旧エンジンボードでは機能しません。

用例

```
PROGRAM PRO1
TAKEARM                                'アームグループの制御権を取得します。
MOVE P, @E P1                          'P1へPTP動作で移動します。
CALL SetHighPathAccuracy                '高軌跡制御機能を有効にします。
MOVE S, 5                               '軌道番号5を自由曲線動作で高軌跡移動します。
CALL ResetHighPathAccuracy              '高軌跡制御機能を無効にします。
MOVE P, @0 P5                           'P4からP5へPTP動作で移動します。
END                                       'プログラム終了します。
```

SetSingularAvoid (ライブラリ)【Ver.2.61 以降】

機能

特異点回避機能を有効化、または無効化します。(6軸専用)

書式

SetSingularAvoid(<有効・無効設定>)

説明

<有効・無効設定>で1を指定した場合、特異点回避機能が有効となり、0を指定した場合、無効となります。特異点回避機能を有効にすると直線、円弧、自由曲線等のCP補間動作中に5軸が0度近くを通過する時に4軸が大きく回転する挙動を抑制することができます。

マクロ定義

<pacman.h>ファイルが必要です。

注意事項

- 4軸ロボットおよび5軸ロボットでは使用できません。
- PTP補間動作では機能しません。
- 機能有効時、動作条件によっては動作中の軌道が大きくずれる場合があります。
- 特異点回避機能は旧エンジンボードでは機能しません。

用例

```
PROGRAM PRO1
TakeArm           ' 制御権を取得
MOVE P,P1        ' P1にPTP補間で移動する。
CALL SetSingularAvoid (1) ' 特異点回避機能を有効にする。
MOVE L,P2        ' P2に直線補間で移動する。
CALL SetSingularAvoid (0) ' 特異点回避機能を無効にする。
END
```

SetCollisionJnt (ライブラリ) 【Ver.2.61 以降】

機能

指定した軸の衝突検出を有効にします。

書式

SetCollisionJnt(< 軸番号 >)

説明

指定した軸の衝突検出を有効にします。

マクロ定義

< pacman.h >ファイルが必要です。

関連項目

ResetCollisionJnt

注意事項

- 軸番号は、ロボット軸を指定してください。ロボット軸以外を指定した場合、「指定軸が範囲外です」のエラーが表示されます。
- 力制限機能有効時は実行できません。
- 衝突検出機能は下記のコントローラバージョンで機能します。
下記コントローラバージョンより古いプロジェクトデータをご使用する場合はプログラミングマニュアル I の「衝突検出機能」を参照してください。
 - VS-G シリーズ : Ver.2.61 以降
 - VS-*** シリーズ : Ver.3.20 以降
 - VM-G、VP-G シリーズ : Ver.2.70 以降
 - HM-G、HS-G シリーズ : Ver.2.80 以降
- 衝突検出機能は旧エンジンボードでは機能しません。

用例

TAKEARM	’ロボット制御権を取得します。
CALL SetCollisionJnt(2)	’2軸の衝突検出を有効にします。
MOVE P, P1,P2	’2軸の衝突検出が有効の状態です。P1点からP2点へPTP移動します。
CALL ResetCollisionJnt(2)	’2軸の衝突検出を無効にします。
MOVE P, P2,P3	’2軸の衝突検出が無効の状態です。P2点からP3点へPTP移動します。

ResetCollisionJnt (ライブラリ) 【Ver.2.61 以降】

機能

指定した軸の衝突検出を無効にします。

書式

ResetCollisionJnt(< 軸番号 >)

説明

指定した軸の衝突検出を無効にします。

マクロ定義

< pacman.h >ファイルが必要です。

関連項目

SetCollisionJnt

注意事項

- 軸番号は、ロボット軸を指定してください。ロボット軸以外を指定した場合、「指定軸が範囲外です」のエラーが表示されます。
- 衝突検出機能は下記のコントローラバージョンで機能します。
下記コントローラバージョンより古いプロジェクトデータをご使用の場合はプログラミングマニュアル I の「衝突検出機能」を参照してください。
 - VS-G シリーズ : Ver.2.61 以降
 - VS-*** シリーズ : Ver.3.20 以降
 - VM-G、VP-G シリーズ : Ver.2.70 以降
 - HM-G、HS-G シリーズ : Ver.2.80 以降
- 衝突検出機能は旧エンジンボードでは機能しません。

用例

TAKEARM	'ロボット制御権を取得します。
CALL SetCollisionJnt(2)	'2軸の衝突検出を有効にします。
MOVE P, P1,P2	'2軸の衝突検出が有効の状態です。P1点からP2点へPTP移動します。
CALL ResetCollisionJnt(2)	'2軸の衝突検出を無効にします。
MOVE P, P2,P3	'2軸の衝突検出が無効の状態です。P2点からP3点へPTP移動します。

GetCollisionForce (ライブラリ) 【Ver.2.61 以降】

機能

外力最大値を取得します。

書式

GetCollisionForce(<J型変数>)

説明

外力最大値を <J型変数> にセットします。

関連項目

ClearCollisionForce

注意事項

- 外力最大値は、モータ ON 状態の衝突検出有効設定軸のみ、手動モードおよび自動モード毎に計算されます。
- 自動モード時は、自動モード時の外力最大値がセットされ、手動モード時、ティーチチェックモード時は、手動モード時の外力最大値がセットされます。
- 衝突検出機能は下記のコントローラバージョンで機能します。
下記コントローラバージョンより古いプロジェクトデータをご使用する場合はプログラミングマニュアル I の「衝突検出機能」を参照してください。
 - VS-G シリーズ : Ver.2.61 以降
 - VS-*** シリーズ : Ver.3.20 以降
 - VM-G、VP-G シリーズ : Ver.2.70 以降
 - HM-G、HS-G シリーズ : Ver.2.80 以降
- 衝突検出機能は旧エンジンボードでは機能しません。

用例

TAKEARM	’ ロボット制御権を取得します。
CALL ClearCollisionForce	’ 外力最大値を初期化します。
CALL SetCollisionJnt(2)	’ 2 軸の衝突検出を有効にします。
MOVE P, P1,P2	’ P1 点から P2 点へ PTP 移動します。
CALL GetCollisionForce(J1)	’ P1 点から P2 点に移動した際の外力最大値を J1 に ’ セットします。
CALL ClearCollisionForce	’ 外力最大値を初期化します。
MOVE P, P2,P3	’ P2 点から P3 点へ PTP 移動します。
CALL GetCollisionForce(J2)	’ P2 点から P3 点に移動した際の外力最大値を J2 に ’ セットします。

ClearCollisionForce (ライブラリ) 【Ver.2.61 以降】

機能

外力最大値を初期化します。

書式

ClearCollisionForce

説明

外力最大値を 0 (初期値) に戻します。

関連項目

GetCollisionForce

注意事項

- 本命令は、全対象軸の、手動モード時および自動モード時の外力最大値とともに初期化します。
- 衝突検出機能は下記のコントローラバージョンで機能します。
下記コントローラバージョンより古いプロジェクトデータをご使用する場合はプログラミングマニュアル I の「衝突検出機能」を参照してください。
 - VS-G シリーズ : Ver.2.61 以降
 - VS-*** シリーズ : Ver.3.20 以降
 - VM-G、VP-G シリーズ : Ver.2.70 以降
 - HM-G、HS-G シリーズ : Ver.2.80 以降
- 衝突検出機能は旧エンジンボードでは機能しません。

用例

TAKEARM	’ ロボット制御権を取得します。
CALL SetCollisionJnt (2)	’ 2 軸の衝突検出を有効にします。
MOVE P, P1, P2	’ P1 点から P2 点へ PTP 移動します。
CALL ClearCollisionForce	’ 外力最大値を初期化します。
MOVE P, P2, P3	’ P2 点から P3 点へ PTP 移動します。
CALL GetCollisionForce (J2)	’ P2 点から P3 点に移動した際の外力最大値を J2 に ’ セットします。

SetCollisionLevel (ライブラリ)【Ver.2.61 以降】

機能

指定した軸の衝突検出レベルを設定します。

書式

SetCollisionLevel(< 軸番号, レベル >)

説明

指定した軸の衝突検出レベルを設定します。標準レベルは 100 です。レベルを小さく設定した場合、衝突検出感度が高くなります。レベルを大きく設定した場合、衝突検出感度が低くなります。

マクロ定義

< pacman.h >ファイルが必要です。

関連項目

SetCollisionJnt、ResetCollisionJnt

注意事項

- 軸番号は、ロボット軸を指定してください。ロボット軸以外を指定した場合、「指定軸が範囲外です」のエラーが表示されます。
- 標準レベルは通常作業時に衝突誤検出しないレベルになります。ロボットの負荷設定値を正しく設定していただくと、レベルを小さくし、衝突検出感度を上げることができます。
- 衝突を誤検出する場合、レベルを大きくし、衝突検出感度を下げてください。
- 衝突検出機能は下記のコントローラバージョンで機能します。
下記コントローラバージョンより古いプロジェクトデータをご使用の場合はプログラミングマニュアル I の「衝突検出機能」を参照してください。
 - VS-G シリーズ : Ver.2.61 以降
 - VS-*** シリーズ : Ver.3.20 以降
 - VM-G、VP-G シリーズ : Ver.2.70 以降
 - HM-G、HS-G シリーズ : Ver.2.80 以降
- 衝突検出機能は旧エンジンボードでは機能しません。

用例

```
TAKEARM                ' ロボット制御権を取得します。
CALL SetCollisionJnt(3) ' 3 軸の衝突検出を有効にします。
CALL SetCollisionLevel(3,80)
                        ' 3 軸の衝突検出レベルを 80 に設定します。
MOVE P, P1,P2          ' 3 軸の衝突検出が敏感な状態で P1 点から
                        ' P2 点へ PTP 移動します。
CALL SetCollisionLevel(3,120)
                        ' 3 軸の衝突検出レベルを 120 に設定します。
MOVE P, P2,P3          ' 3 軸の衝突検出が鈍い状態で P2 点から
                        ' P3 点へ PTP 移動します。
```

SetExtForceDetect (ライブラリ) 【Ver.2.61 以降】

機能

外力検出の有効、無効を設定します。

書式

SetExtForceDetect(< 設定値 >)

説明

< 設定値 > が 1 の場合、外力検出が有効になります。0 の場合、無効になります。

マクロ定義

< pacman.h > ファイルが必要です。

関連項目

SetCollisionJnt、GetCollisionForce

注意事項

- 設定値は、0 または 1 にしてください。これ以外の場合はエラーになります。
- デフォルト値は 1 (有効) です。無効に変更しても手動・自動のモード切替え時、または電源再投入後には 1 に戻ります。
- 本命令はロボットが記憶している外力最大値を更新するかどうかを指定するものであり、衝突検出の有効・無効を設定するものではありません。無効に設定した場合、記憶している外力最大値よりも大きな外力をロボットが検出しても、外力最大値は更新されません。特定区間の外力最大値を確認する場合などにご使用ください。
- 衝突検出機能は下記のコントローラバージョンで機能します。
下記コントローラバージョンより古いプロジェクトデータをご使用する場合はプログラミングマニュアル I の「衝突検出機能」を参照してください。
 - VS-G シリーズ : Ver.2.61 以降
 - VS-*** シリーズ : Ver.3.20 以降
 - VM-G、VP-G シリーズ : Ver.2.70 以降
 - HM-G、HS-G シリーズ : Ver.2.80 以降
- 衝突検出機能は旧エンジンボードでは機能しません。

用例

TAKEARM	’ ロボット制御権を取得します。
CALL SetExtForceDetect (0)	’ 外力検出を無効にします。
CALL ClearCollisionForce	’ 外力最大値を初期化します。
CALL SetCollisionJnt (3)	’ 3 軸の衝突検出を有効にします。
MOVE P, P1, P2	’ P1 点から P2 点へ PTP 移動します。
CALL SetExtForceDetect (1)	’ 外力検出を有効にします。
MOVE P, P2, P3	’ P2 点から P3 点へ PTP 移動します。
CALL GetCollisionForce (J2)	’ P2 点から P3 点へ移動した際の外力最大値が J2 にセットされます。

第 8 章 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 III のインストールディスク内の Samples フォルダに収録されています。

これをインポートするには、プログラム (P) → プログラムのインポート (I) で必要なサンプルプログラムを選択してください。

8.2 操作盤画面作成サンプルプログラム

WINCAPS III 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_btms、Lamp_variable_btms、Lamp_pb_btms

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_btms

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 P B 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 P B 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 P B 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 P B 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 P B 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 P B 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 P B 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

```
call make_PB(sysprm(),pb_title(),PB_NUM)
```

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

```
call make_LED(sysprm(),pl_title(),MAX_PL)
```

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

```
call make_LABEL(sysprm(),pl_title(),PL_NUM)
```

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

make_PARAM_BOX

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

```
call make_PARAM_BOX(sysprm(),VBTN_NUM)
```

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

```
single_button_set
```

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

```
call single_button_set(btn_no,prm())
```

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)
```

第9章 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

MOVE P, P10

P11 = P10

CALL viTran6 (0, F1, F2, P11)

APPROACH P, P11, 100

MOVE L, P11

DEPART L, 100

CHANGETOOL 0

・視覚 CAL を行なったツールに設定

・待機位置に移動

・待機位置の姿勢データをコピー

・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)

・視覚で計測した位置に移動

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。

デンソーロボット

プログラミングマニュアルⅡ PACライブラリ

初版 2009年 4月
第3版 2011年 1月
第4版 2011年 4月

株式会社デンソーウェーブ

4N**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が、万一ご不審の点や誤り、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。

