

デンソーロボット

取扱説明書 追補版

Ver. 1.9 の新機能

Copyright © 2002 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

HS-E シリーズロボット新設に伴い、メインソフトウェアバージョンを Ver.1.8 から Ver.1.9 に更新しました。本書は取扱説明書の追補版として、このバージョン更新に伴って追加・変更された機能をまとめて説明してありますので、他の取扱説明書とあわせてご活用ください。

また、新設の HS-E シリーズについての説明は、「HS-E シリーズ 概要書」と「HS-E シリーズ 設置保守ガイド」を参照してください。

本書が扱う範囲

HS-E シリーズに搭載の RC5 コントローラ
— メインソフトウェアバージョン Ver.1.9 に追加・変更された新機能についての説明 —

目次

1. アーチモーション機能の追加[Ver.1.9以降、4軸ロボットのみ].....	1
1.1. アーチモーション機能とは.....	1
1.2. アーチモーション機能のライブラリ.....	2
ArchMove (ライブラリ) [Ver.1.9以降].....	2
SetArchParam (ライブラリ) [Ver.1.9以降].....	3
2. ローカル変数機能の追加.....	4
2.1. ローカル変数機能とは.....	4
2.2. 即表示機能.....	5
2.3. 登録表示機能.....	9
2.4. 引数付きのプログラム起動.....	10
2.5. WINCAPS IIによるローカル変数の参照.....	11
3. 簡単教示機能の適用範囲の拡大.....	13
3.1. 動作コマンド挿入方法.....	13
3.2. 動作コマンド編集方法.....	17
3.3. 動作コマンド実行方法.....	19
3.4. 簡単教示機能の補足説明.....	20
3.5. 各パラメータの編集,実行条件.....	22
3.6. 簡単教示機能の注意事項.....	25
3.7. 簡単教示用の使用条件.....	25
4. 操作ログ表示機能の追加.....	26
4.1. 操作経路.....	26
4.2. 操作ログ表示画面の内容.....	26
5. HS-E ロボットのダイレクトモードの操作方法.....	27
5.1. HS-E 以外の4軸ロボットの場合 (従来の方法).....	27
5.2. HS-E ロボットの場合.....	29
6. シリアル通信回線の追加 (イーサネットサーバ、クライアント対応).....	30
6.1. 回線番号.....	30
6.2. 通信コマンド.....	30
6.3. 通信バッファのクリア.....	30
6.4. ロボットコントローラのクライアントの設定.....	31
6.5. ロボットコントローラのサーバの設定.....	33
6.6. RS232Cポート、Ethernet (サーバ、クライアント) のコマンド.....	35
INPUT (ステートメント) 【SLIM 準拠】.....	35
LINEINPUT (ステートメント).....	36
PRINT (ステートメント) 【SLIM 準拠】.....	36
WRITE (ステートメント).....	37
FLUSH (ステートメント).....	38
7. シリアルバイナリ通信にイーサネットの使用が可能.....	39
7.1. 使用方法.....	39
7.2. シリアルバイナリ通信 (RS-232C および Ethernet ポート) のコマンド.....	40
printb.....	40
inputb.....	40
lprintb.....	41
linputb.....	42
com_encom.....	43
com_discom.....	44
com_state [Ver. 1.5以降].....	44
8. CC-Link リモートデバイスボード (オプション) の新設.....	45
8.1. 製品仕様.....	46
8.2. I/O データの割付け.....	50

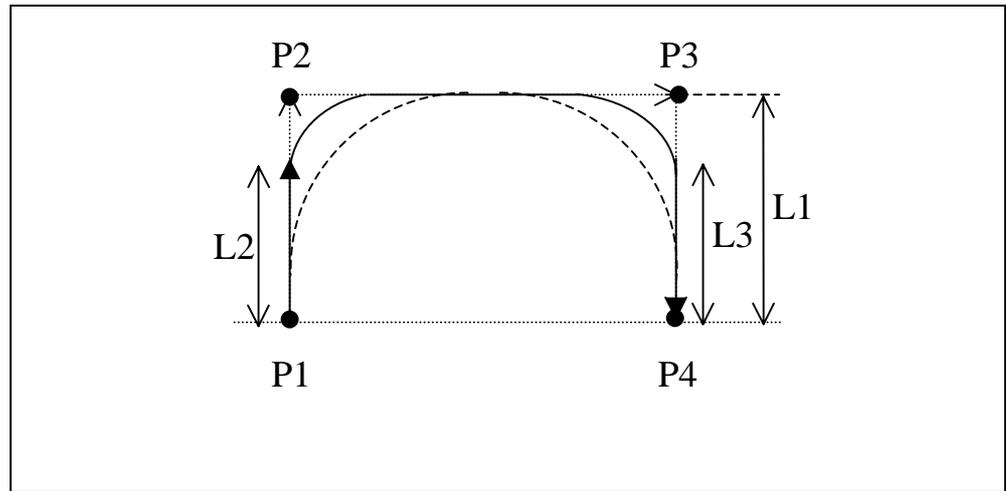
9. RS232C 増設ボード（推奨オプション）の設定.....	53
9.1. RS232C 増設ボード（推奨品）	53
9.2. 増設ボードの取り付け	53
9.3. 増設ボードのジャンパ・DIP スイッチ設定.....	54
9.4. RS232C 増設ボードのコネクタと回線番号.....	55
9.5. RS232C 増設ボードの通信設定	55
9.6. 用例（リトライ処理）	56
9.7. 保証の範囲	57
10. ペンダント延長ケーブル（オプション）の新設.....	58
11. WINCAPS II の機能追加	59
11.1. プログラムバンクに更新機能を追加	59
11.2. ローカル変数のモニタおよび変更機能の追加.....	59
11.3. コントローラの機能拡張機能の追加	60
12. コマンドの追加 [Ver.1.9 以降].....	61
ST_aspACLD（ステートメント） [Ver. 1.9 以降]	61
ST_aspChange（ステートメント） [Ver.1.9 以降]	62
ST_SetGravity（ステートメント） [Ver.1.9 以降].....	63
ST_ResetGravity（ステートメント） [Ver.1.9 以降]	64
ST_SetGrvOffset（ステートメント） [Ver.1.9 以降]	64
ST_ResetGrvOffset（ステートメント） [Ver.1.9 以降]	65
ST_SetCurLmt（ステートメント） [Ver.1.9 以降]	65
ST_ResetCurLmt（ステートメント） [Ver.1.9 以降]	67
ST_SetEralw（ステートメント） [Ver.1.9 以降]	68
ST_ResetEralw（ステートメント） [Ver.1.9 以降]	68
ST_OnSrvLock（ステートメント） [Ver.1.9 以降]	69
ST_OffSrvLock（ステートメント） [Ver.1.9 以降]	70
ST_SetCompControl（ステートメント） [Ver.1.9 以降]	71
ST_SetCompFControl（ステートメント） [Ver.1.9 以降]	72
ST_ResetCompControl（ステートメント） [Ver.1.9 以降]	73
ST_SetFrcCoord（ステートメント） [Ver.1.9 以降]	74
ST_SetFrcLimit（ステートメント） [Ver.1.9 以降].....	75
ST_ResetFrcLimit（ステートメント） [Ver.1.9 以降].....	76
ST_SetCompRate（ステートメント） [Ver.1.9 以降]	77
ST_ResetCompRate（ステートメント） [Ver.1.9 以降]	78
ST_SetFrcAssist（ステートメント） [Ver.1.9 以降].....	78
ST_ResetFrcAssist（ステートメント） [Ver.1.9 以降].....	79
ST_SetCompJLimit（ステートメント） [Ver.1.9 以降]	79
ST_ResetCompJLimit（ステートメント） [Ver.1.9 以降].....	80
ST_SetCompVMode（ステートメント） [Ver.1.9 以降].....	81
ST_ResetCompVMode（ステートメント） [Ver.1.9 以降]	81
ST_SetCompEralw（ステートメント） [Ver.1.9 以降].....	82
ST_ResetCompEralw（ステートメント） [Ver.1.9 以降].....	83
ST_SetDampRate（ステートメント） [Ver.1.9 以降].....	83
ST_ResetDampRate（ステートメント） [Ver.1.9 以降].....	84
ST_SetZBalance（ステートメント） [Ver. 1.9 以降]	85
ST_ResetZBalance（ステートメント） [Ver. 1.9 以降]	85
13. エラーコード表の追加・修正	86

1. アーチモーション機能の追加[Ver. 1.9 以降、4 軸ロボットのみ]

1.1. アーチモーション機能とは

【関連ページ】：「プログラミングマニュアル I」のP3-3

アーチモーション機能を使うと、効率のよいPick&Place (P & P)動作をさせることができます。



上図のように現在位置 (P 1) から目標位置 (P 4) へ動作する P & P 動作は、通常、経由する P 2, P 3 点を指定して実現します。P 2、P 3 付近でパス動作させることで動作時間を短縮させることができます。アーチモーションでは、パス動作と異なり Z 方向の動作中のどの位置からでも次動作に移る事ができますので、さらに効率のよい動作が可能です。

また、P 4 に対する Z 軸の動作量 (L 1)、アーチの開始位置 (P 1 からの距離 L 2)、アーチの完了位置 (P 4 からの距離 L 3) を指定するだけで、プログラム作成も簡単です。

ただし、アーチモーションでは位置制御を行なっていません。動作速度の変動により動作経路が変わりますので、動作の際は周辺設備との干渉に注意してください。なお、アーチモーション中の動作は全て P T P 補間となります。

1.2. アーチモーション機能のライブラリ

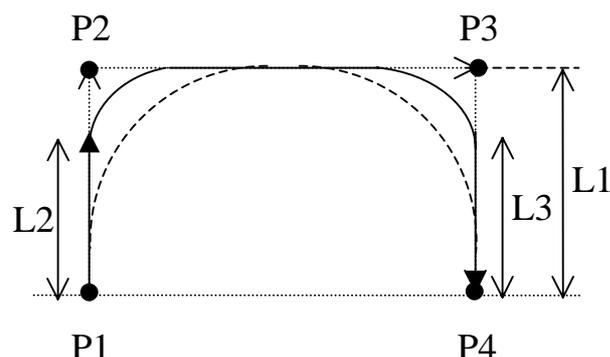
ArchMove (ライブラリ) [Ver.1.9 以降]

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]

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
                          :
                          <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。
```

2. ローカル変数機能の追加

2.1. ローカル変数機能とは

ローカル変数機能とは、プログラム内のローカル変数(I型、F型、D型、V型、P型、J型、T型、S型、DEFIO型)の参照および書き込みをする機能です。

ローカル変数機能として、下記の機能が追加されています。

(1) 即表示機能

行を指定することにより、プログラム内のローカル変数を参照します。

(2) 登録表示機能

登録された変数を参照します。

プログラム実行中で行が指定できない場合や、複数のプログラム変数を同時に参照したい場合に用います。

(3) 引数付きのプログラム起動

引数付きのプログラムを単独で起動することができます。

(4) WINCAPS II によるローカル変数の参照

WINCAPS II でローカル変数の読み込み、書き込みができます。

注意1：1から3の機能はティーチングペンダントでのみ実行できる機能です。

ミニペンダント、オペレーティングパネルでは実行できません。

注意2：プログラムの引数が配列変数である場合、ローカル変数機能は対応していません。

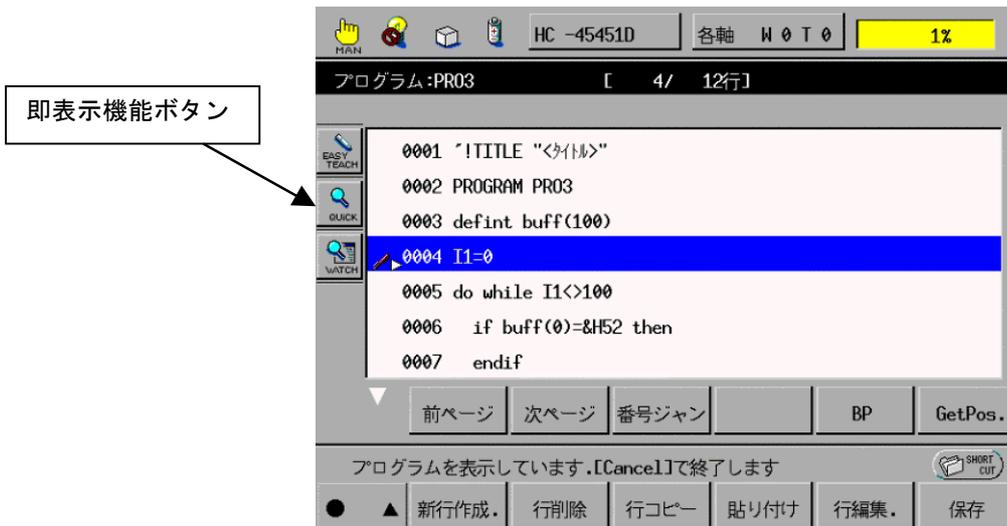
例. PROGRAM SUB1 (Ii%(10))のように引数が配列である場合。

2.2. 即表示機能

即表示機能とは、行を指定することによりプログラム内のローカル変数を参照する機能です。

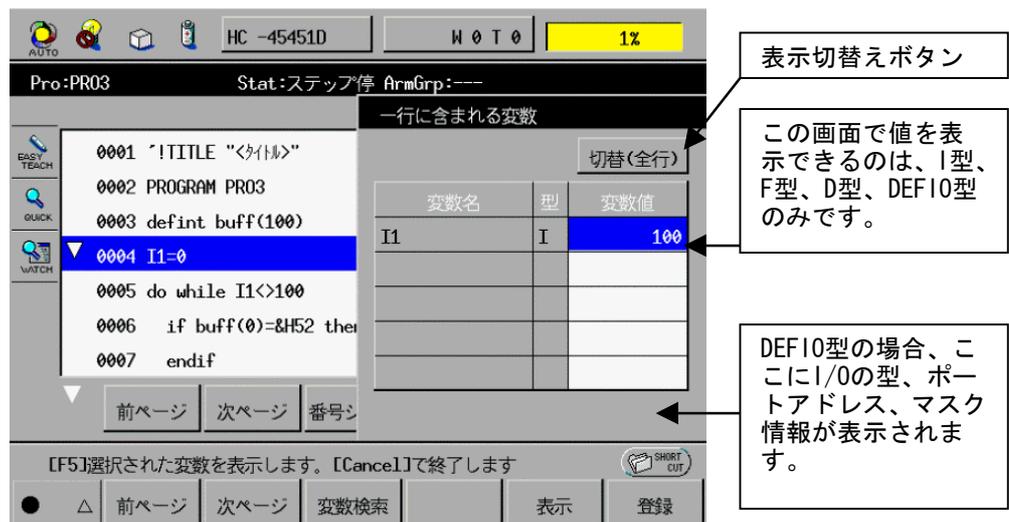
プログラムテキストが表示されている画面に、即表示機能ボタンが追加されています。

即表示ボタンを押すことにより、現在カーソル行のローカル変数、およびグローバル変数を参照します。



注意：カーソル行はマニュアルモードでしか変更できません。

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



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

例 1. I型変数の使用数が100であるのにI101を表示しようとした。

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

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

注意 2 : DEFIO型変数でポートアドレスが入出力範囲外の場合グレー表示になります。

注意 3 : 引数が配列である変数は表示されません。

例. PROGRAM SUB1(I%, I2%(10))などのプログラムである場合、引数が配列変数であるI2は表示されません。

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

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

表示切替えボタン

全行表示であるため変数名のINDEX部は?表示となります。

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

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

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

Ver. 1.9 の新機能

登録ボタンを押すことにより、変数を登録できます。登録された変数は登録表示機能で参照することができます。



注意1： 変数のINDEX部が?表示である場合、この画面で登録はできません。

注意2： DEFIO変数のポートが範囲外の場合、登録はできません。

表示ボタンを押すことにより、選択された変数が表示されます。



注意： DEFIO変数のポートが範囲外の場合、表示できません。

Ver. 1.9 の新機能

表示ボタンを押すことにより、選択された変数が表示されます。
次画面は、P型ローカル変数PX(3)を表示した例です。
この画面では値変更や変数移動など、グローバル変数同等の操作を実行することができます。この画面での変数の登録は「F12 登録」になります。



- 注意1：変数のINDEX部が?の場合、表示ボタンを押すとINDEXが0の変数が表示されます。目的のINDEXへカーソルを合わせてください。
- 注意2：DEF10変数の値を変更する場合は、通常のI/Oと同様にデッドマンスイッチを押す必要があります。
- 注意3：ローカル変数への位置取り込みはありません。

2.3. 登録表示機能

あらかじめ登録された変数を参照します。プログラム実行中で行が指定できない場合や、複数のプログラムの変数を参照したい場合に用います。変数の登録は、即表示機能の登録ボタンまたは変数を表示した画面の[F12 登録]で行なうことができます。（即表示機能の項目参照）登録できる最大数は50です。登録表示機能の画面は、テキストが表示される画面の登録表示ボタンまたは、変数選択画面での登録変数ボタンを押すことによって表示されます。



登録表示ボタンを押すことにより、登録されている変数の一覧が表示されます。次例は登録された変数の一覧表示されている画面です。



注意1： PRO名はグローバル変数である場合Globalと表示されます。

注意2： 登録されている変数で、プログラム名、変数名、次元数が変更されコンパイルされた場合グレー表示になります。

注意3： 登録されているDEFIO型変数で、プログラム名、変数名、I/O変数の型、ポート、マスク情報が変更されコンパイルされた場合グレー表示になります。

表示ボタンを押すと、選択された変数が表示されます。この画面では、値変更や変数移動などのグローバル変数と同等の操作を実行することができます。（即表示機能の章参照。）

2.4. 引数付きのプログラム起動

引数付きのプログラムを、引数の設定を行うことにより単独で起動することができます。

停止状態の引数付きプログラムを起動させる場合、引数の設定ウィンドウが表示されます。次例は PROGRAM SUB1 (1i%) を起動させる時の設定画面です。引数の値を設定する場合は表示ボタンを押します。値変更後、引数設定ウィンドウで、値を確認しOKを押します。

プログラム名	状態	行
AUTOEXEC	停止中	
LOCALTEST	停止中	
SUB1	停止中	
TSR1	停止中	
COMMAND	停止中	
ASPAULD	停止中	

変数名	型	変数値
LI	I	0

引数の値を変更する場合、表示ボタンを押します。表示されたウィンドウでの値変更の方法は、グローバル変数と同様です。

OKを押すと通常の起動と同じ画面が表示されます。

その後は通常の起動と同じです。次例はOKが押された時表示される画面です。

プログラム (SUB1) を起動しますか?

1 サイクル起動

連続起動

変数値
0

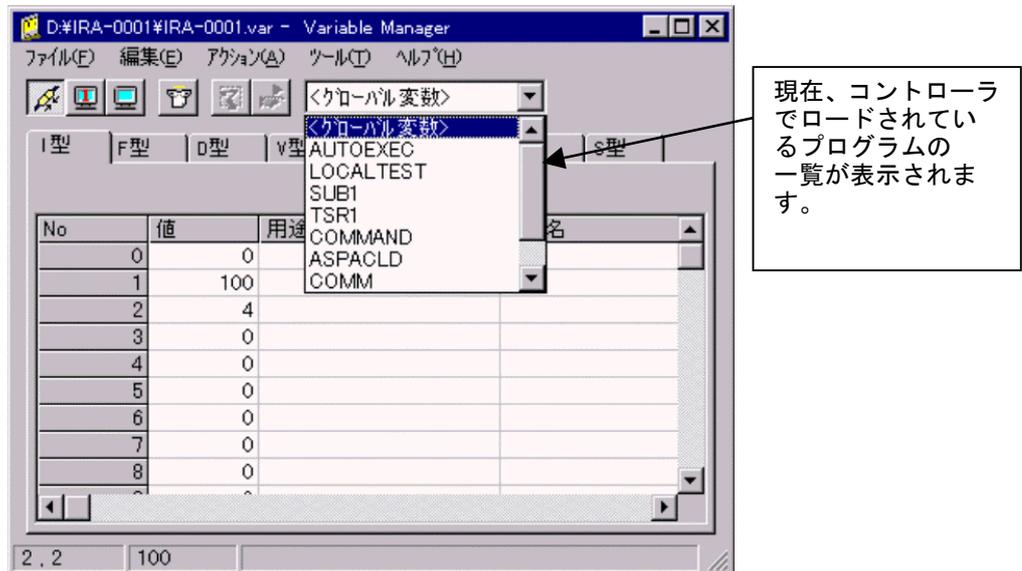
注意：引数に配列を含む場合は、引数設定ウィンドウは表示されず、実行時「736F 引数に配列のあるプログラムは起動できません」というエラーになります。

2.5. WINCAPS II によるローカル変数の参照

WINCAPS II でローカル変数の読み込み、書き込みができます。
WINCAPS II の変数マネージャを接続します。

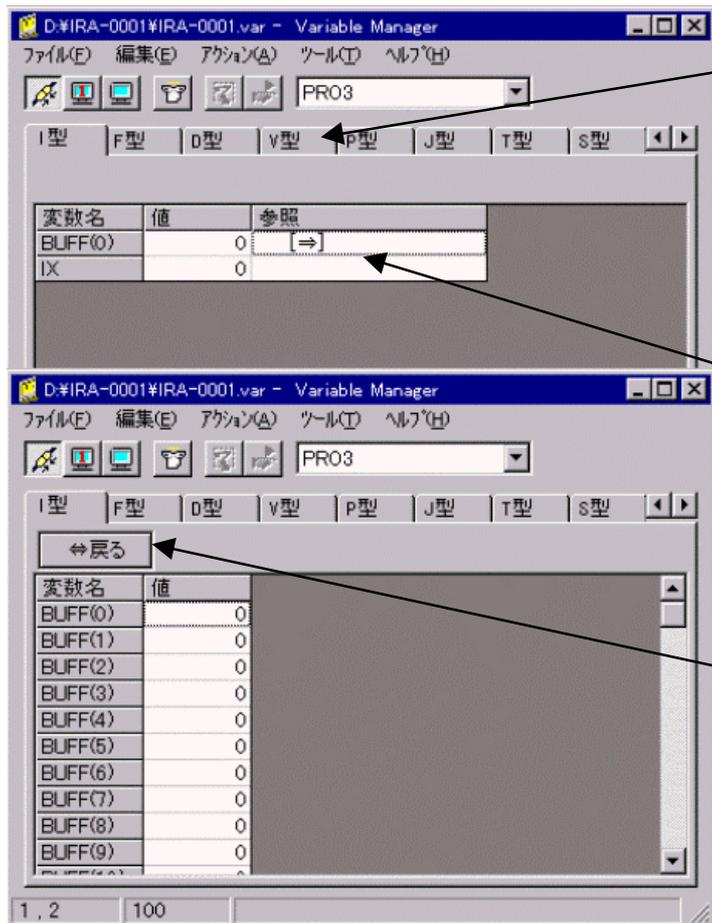


接続後、プログラムの一覧を表示させ、ローカル変数を表示したいプログラムを選択します。



Ver. 1.9 の新機能

次例は、プログラムのPR03のローカル変数と変数名BUFF配列の中身を表示した例です。



The image shows two screenshots of the 'Variable Manager' window. The top screenshot shows the 'P型' (P-type) tab selected, displaying a table with columns '変数名' (Variable Name), '値' (Value), and '参照' (Reference). The table contains two rows: 'BUFF(0)' with value '0' and a reference '⇒', and 'IX' with value '0'. The bottom screenshot shows the 'P型' tab selected, displaying a table with columns '変数名' (Variable Name) and '値' (Value). The table contains ten rows, all with 'BUFF' followed by a number in parentheses and the value '0'. A button labeled '⇒戻る' (Return) is visible above the table. Three callout boxes provide instructions: the first points to the 'P型' tab, the second points to the '⇒' symbol in the reference column, and the third points to the '⇒戻る' button.

表示したいローカル変数の型を選択します。この例では、プログラムPR03内のINT型ローカル変数を表示しています。

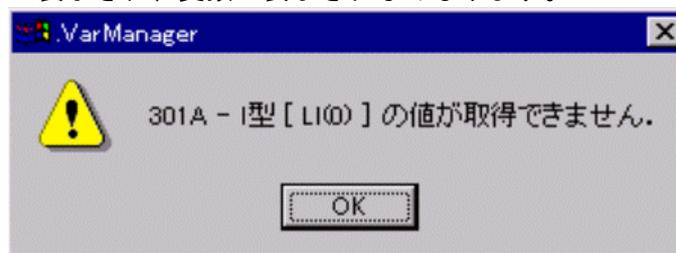
ローカル変数が配列の場合、⇒が表示されます。配列の中身がみたい場合はここをダブルクリックします。

ここをクリックすると上のウィンドウに戻ります。

値の変更方法はグローバル変数と同様です。

注意1： WINCAPS II からDEFIO型変数を変更する場合、ペンダントでデッドマンスイッチが押されている必要があります。

注意2： コントローラでロードの処理が行われた場合、次のようなメッセージが表示され、変数が表示されなくなります。



この場合、もう一度変数マネージャを接続し直してください。

3. 簡単教示機能の適用範囲の拡大

【関連ページ】：「操作ガイド」のP3-32

Ver. 1.9では、Ver. 1.8から追加された簡単教示機能に、以下の機能を追加しました。

- (1) グローバル変数の指定が可能
 - (2) APPROACH, DEPARTコマンドの指定が可能
- 以下、追加した機能を含めた全体の説明をします。

■ 簡単教示機能とは

簡単教示機能は、ティーチングペンダントを使用したプログラム編集・入力・位置教示を支援する機能です。

簡単教示機能を使用することにより、以下のようにプログラム入力・位置教示を容易に行なうことができます。

- (1) 動作コマンドおよびそのパラメータ（目標位置を含む）をワンタッチでプログラムに挿入できる。
- (2) 動作コマンドおよびそのパラメータをワンタッチで編集することができる。
- (3) コンパイルする前に作成した動作プログラムを確認できる。

3.1. 動作コマンド挿入方法

簡単教示機能を使用して動作コマンドを挿入する操作の流れを説明します。

■ ティーチングペンダントのみ操作可能

ステップ 1 モード切替スイッチを[MANUAL]に合わせます。

ステップ 2 [F1 プログラム]を押します。

プログラム一覧が表示されます。



F5

編集するプログラムを選択し、[F5 編集.]または[表示]（または[F1 新規プロ.]）を押します。

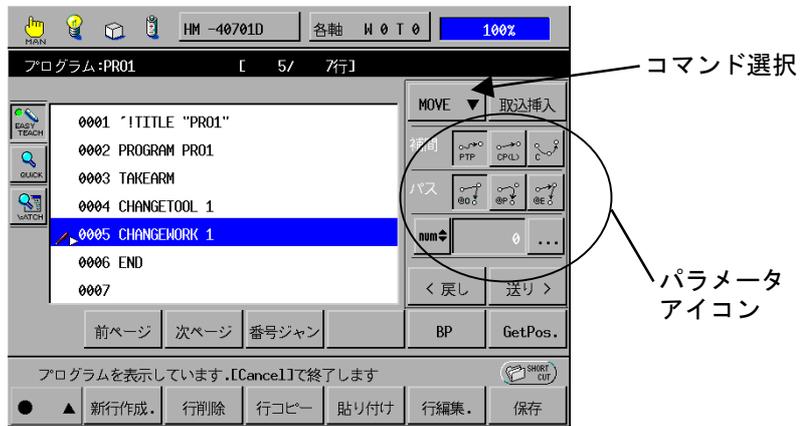
注：[F1 新規プロ.]は新規にプログラムを作成する場合に、[F5 編集.]または[表示]はプログラム内容の確認および編集する場合にご使用ください。

ステップ 3 プログラム表示画面が表示されます。



[簡単指示アイコン] を押し、簡単指示ウィンドウを表示させます。

ステップ 4 動作コマンドのパラメータをパラメータアイコンを押して設定してください。



注：コマンド選択ボタンは、現在は、何も反応しません。
これは、将来拡張用のボタンです。

補間方法 (注1)		パス開始変位 (注1)		目標位置	
	PTP制御 (書式 "P")		エンド動作(書式 "@0")		直値 (注2)
	CP制御 (書式 "L")		パス動作 (書式 "@P")		P型グローバル変数 (注2)
	円弧動作 (書式 "C")		エンコーダ値確認動作 (書式 "@E")		J型グローバル変数 (注2)
					T型グローバル変数 (注2)
					グローバル変数の番号。このボタンを押すと、10キーが現れ数値を変更する事が出来ます。

注1：押されているアイコンが有効となります。

注2：押すたびにアイコンが直値→P型→J型→T型と変わります。表示されているアイコンが有効となります。

ステップ 5

動作コマンドを挿入したい行の一行上にカーソルを移動し、[取込挿入]を押します。



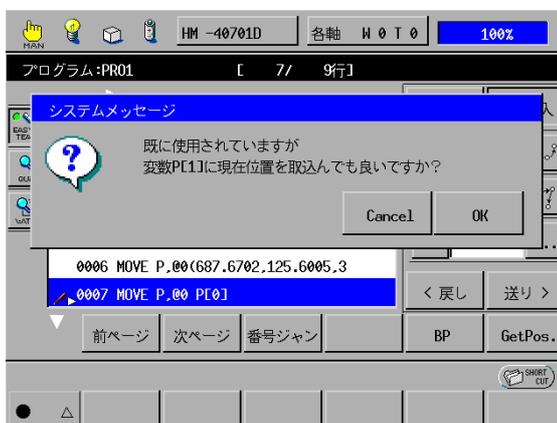
パラメータアイコンの内容から動作コマンドが作成され、選択したカーソルの一行下に動作コマンドが挿入されます。

目標位置に直値を指定した場合は、ロボットの現在位置が数値として動作コマンドに記述されます。

目標位置にグローバル変数を指定した場合は、ロボットの現在位置が指定したグローバル変数に格納されます。

ステップ 6

目標位置にグローバル変数を指定して[取込挿入]を押した時、指定したグローバル変数が使用済み（すでに値が入っている）の場合は、格納確認メッセージが表示されます。



上書きしても良い場合は[OK]を押してください。

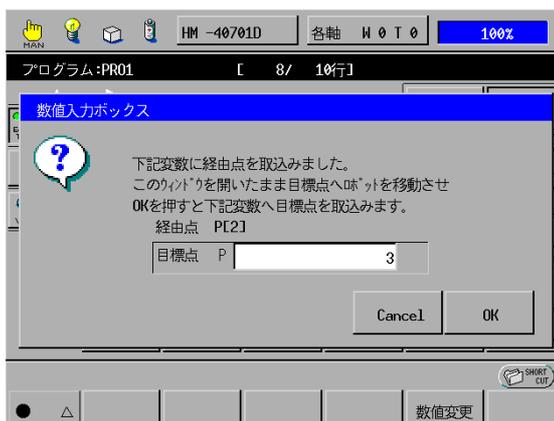
ステップ 7

挿入するコマンドが円弧(MOVE C)で目標位置が直値の場合、[取込挿入]を押すと下記のようなメッセージが表示されますので、ウィンドウが開いたままの状態
で目標点までロボットを移動させ[OK]を押すと、円弧動作のコマンドが挿入されま
す。



ステップ 8

挿入するコマンドが円弧(MOVE C)で目標位置がグローバル変数の場合、[取込挿入]を押すと下記のようなメッセージが表示されます。ウィンドウが開いたまま
の状態
で目標点までロボットを移動させ[OK]を押すと、指定のグローバル変数に目
標点が格納されます。(目標点の変数番号は経由点+1の値となります)



3.2. 動作コマンド編集方法

簡単教示機能を使用して動作コマンドを編集する操作の流れを説明します。

ステップ 1 「3.1 動作コマンド挿入方法」のステップ 1～3を行ない、簡単教示ウィンドウを表示させます。

ステップ 2 編集したい行に青いカーソルを合せ [F5 編集.] を押すと、その行の内容が簡単教示編集ウィンドウで編集可能(「3.5 各パラメータの編集, 実行条件」参照)である場合、簡単教示編集ウィンドウが表示されます。



この画面を開いた時、パラメータアイコンはその行の内容に合わせて押された状態になります。ここでパラメータ、目標点などを変更し、[OK]を押すと変更内容が反映されます。

パラメータを変更したい場合はパラメータアイコンで変更します。

目標点（または経由点）を変更したい場合は[F5 値変更]または[F6 GetPos]を押して変更します。

ステップ 3 [F5 値変更]を使用する場合は、上下左右キーまたはジョグダイヤルで変更したい要素にカーソルを合せ、その後[F5 値変更]を押すと、下記のようなテンキーが表示されるので、数値を入れてください。



注：動作コマンドの内容が円弧(MOVE C)の場合は、上下キーまたは、ジョグダイヤルにて経由点と目標点を選択することができます。

ステップ 4

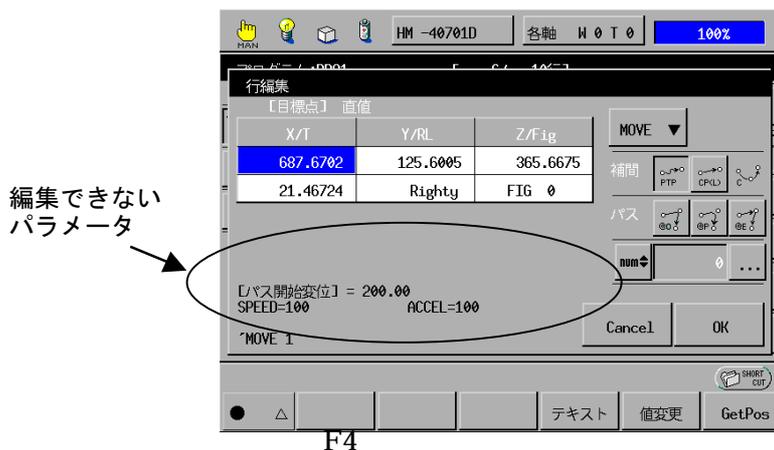
[F6 GetPos] を押すとロボットの現在位置を取込み、編集している行に上書きします。



注：動作コマンドの内容が円弧 (MOVE C) の場合は、上書きする点を経由点と目標点から選択するウィンドウが表示されますので、どちらかを選択してください。

ステップ 5

下記の丸で囲んだところには、この画面で編集できないパラメータが表示されています。そのようなパラメータを編集する時は[F4 テキスト]を押してください。



ステップ 6

下記のようにフルキーが表示され、テキスト編集を行なうことができます。



注：この画面で[Cancel]を押すと、簡単教示編集ウィンドウで変更した内容は取り消されます。

3.3. 動作コマンド実行方法

簡単教示機能を使用して動作コマンドを実行する操作の流れを説明します。

ステップ 1 「3.1 動作コマンド挿入方法」のステップ 1～3を行ない、簡単教示ウィンドウを表示させます。

ステップ 2 実行したい行に青いカーソルを合せ [〈戻し〉] または [送り] をおすと、その行の内容が簡単教示機能で実行可能 (「3.5 各パラメータの編集, 実行条件」参照) である場合、実行されます。

[送り] は青いカーソルの内容を実行します。

[〈戻し〉] は目標点以外は青いカーソルの内容で、目標点は青いカーソルの一つ前の動作命令の目標点で実行します。

[〈戻し〉] または [送り] を押すと実行確認ウィンドウが現れます。どちらの場合も、1ステップ起動、1サイクル起動を選択できます。1ステップ起動は1ステップ (1行) 実行すると、実行が終了します。1サイクル起動はプログラムのENDまで、または、簡単教示機能で実行できないコマンドまで実行すると、実行が終了します。

どちらの場合も実行中に停止操作 (非常停止など) を行なうことにより、途中で実行を中止できます。



実行確認ウィンドウで1ステップ起動または1サイクル起動を選択し、[デッドマンスイッチ] と [OK] を押しつづけると実行します。どちらかのボタンを離すと実行を中止します。

ステップ 3 簡単教示機能で実行できないコマンドや書式に誤りがあるコマンドを実行しようとするると下記のような、メッセージが表示されます。



ステップ 4

簡単教示ウィンドウ表示中にデッドマンスイッチを押すと、左右キーの振る舞いがプログラム表示画面横スクロールから [戻し] または [送り] ボタンと同じ機能に変わり、[戻し] または [送り] ボタンの表示が下記のように変わります。



3.4. 簡単教示機能の補足説明

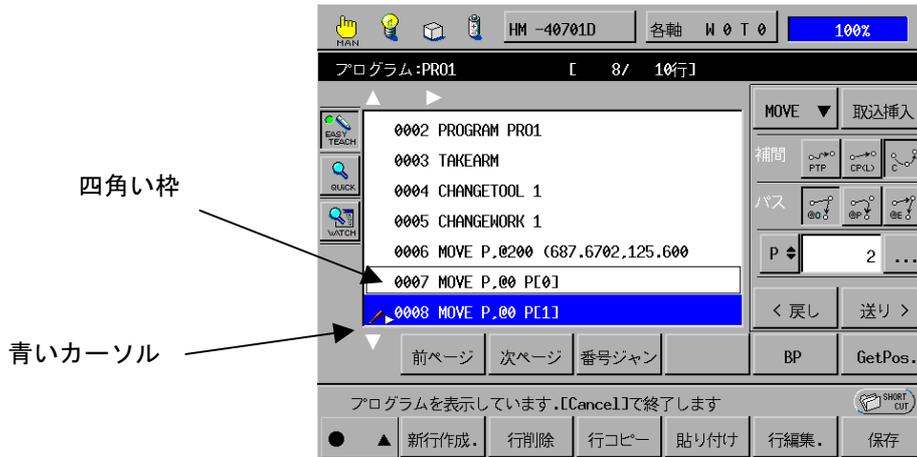
[1] 四角い枠について

簡単教示機能をもちいて動作コマンドを実行すると、プログラムリストに青いカーソルのほかに四角い枠が表示されます。これは、直前に実行したコマンドの目標点を示しています。

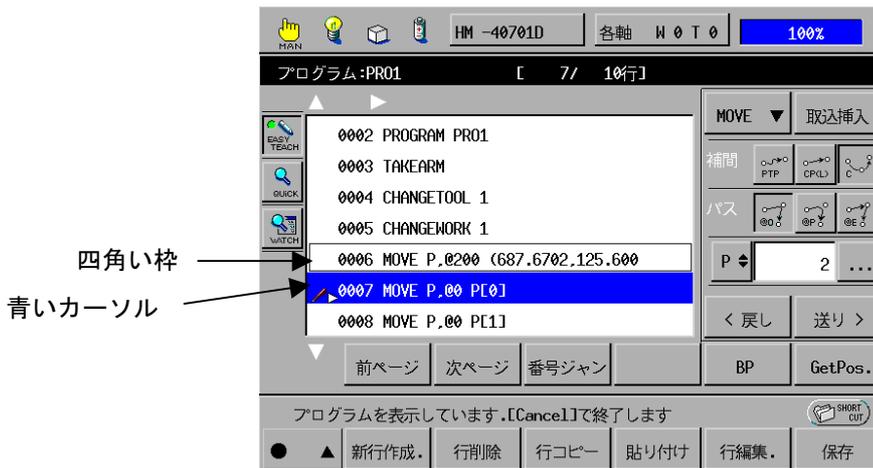
例えば、下記プログラムにおいて、0007行目を送り実行中に一時停止させたとしても、直前に実行した動作コマンドの目標点は0007行目に記述されていますので四角い枠は0007行目に合わされます。



また、0007行目を送り起動にて1ステップ起動した後は、青いカーソルは0008行目に移動しますが、直前に実行した動作コマンドの目標点はやはり0007行目に記述されていますので四角い枠は0007行目に合わされます。



戻し起動の場合は、目標点は青いカーソルの一つ前の動作コマンドの目標点なので、下記プログラムだと、0007行目を戻し起動にて1ステップ起動した時、一時停止または1ステップ起動後は四角い枠は、0006行目に合わされます。



[2] GetPos について

GetPos機能はプログラム表示画面と、簡単教示編集ウィンドウ表示時に使用できます。また、簡単教示ウィンドウが表示されている場合に限り、目標点が直値でも現在位置を取込むことができます。

注：直値とは下記のように目標点を変数で指定せず、各要素の数値を記述することです。

MOVE P,@0(1025.721,-354.7859,1026.708,-179.9987,65.01270,160.9215,5)

↑
直値

3.5. 各パラメータの編集, 実行条件

各パラメータの編集, 実行条件を下表に示します。

注意：

- (1) この表にないパラメータが含まれる場合、簡単教示編集ウィンドウは表示されません。
- (2) この表にないパラメータが含まれる場合、簡単教示機能での動作は行なえません。
- (3) 簡単教示編集ウィンドウを開いてOKを押すと、パラメータの中の余分なスペースは省かれます。また、目標点、パス開始変位量等に非常に小さな値を入力すると、指数表現となることがあります。

○→ 動作・編集可能 □→ 動作可能 表→ 編集時表示有り ×→ 動作・編集不可 △→ コンパイル後動作可能
送→ 送り動作のみ可能

コマンド	引数													
MOVE	補間方法	パス開始変位		ポーズ(目標位置)		動作オプション			その他					
	P	<input type="checkbox"/>	@0	<input type="checkbox"/>	直値	<input type="checkbox"/>	SPEED	直値	<input type="checkbox"/>	表	NEXT	<input type="checkbox"/>	×	表
	L	<input type="checkbox"/>	@P	<input type="checkbox"/>	グローバル変数: 位	<input type="checkbox"/>		グローバル変数: 値	<input type="checkbox"/>	表				
	C	<input type="checkbox"/>	@E	<input type="checkbox"/>	ローカル変数: 位	<input type="checkbox"/>		ローカル変数: 値	<input type="checkbox"/>	×				
			@数値	<input type="checkbox"/>	表		ACCEL	直値	<input type="checkbox"/>	表				
								グローバル変数: 値	<input type="checkbox"/>	表				
								ローカル変数: 値	<input type="checkbox"/>	×				
						DECEL	直値	<input type="checkbox"/>	表					
							グローバル変数: 値	<input type="checkbox"/>	表					
							ローカル変数: 値	<input type="checkbox"/>	×					

APPROACH	補間方法		基準位置		パス開始変位		アプローチ長		動作オプション			その他			
P	<input type="checkbox"/>		直値	<input type="checkbox"/>	@0	<input type="checkbox"/>	直値	<input type="checkbox"/>	SPEED	直値	<input type="checkbox"/>	表	NEXT	<input type="checkbox"/>	×
L	<input type="checkbox"/>		グローバル変数: 位	<input type="checkbox"/>	@P	<input type="checkbox"/>	グローバル変数: 値	<input type="checkbox"/>		グローバル変数: 値	<input type="checkbox"/>	表			
			ローカル変数: 位	<input type="checkbox"/>	@E	<input type="checkbox"/>	ローカル変数: 値	<input type="checkbox"/>		ローカル変数: 値	<input type="checkbox"/>	×			
			@数値	<input type="checkbox"/>					ACCEL	直値	<input type="checkbox"/>	表			
										グローバル変数: 値	<input type="checkbox"/>	表			
										ローカル変数: 値	<input type="checkbox"/>	×			
									DECEL	直値	<input type="checkbox"/>	表			
										グローバル変数: 値	<input type="checkbox"/>	表			
										ローカル変数: 値	<input type="checkbox"/>	×			

DEPART ^{※注2}	補間方法		パス開始変位		デパート長		動作オプション			その他			
P	<input type="checkbox"/>		@0	<input type="checkbox"/>	直値	<input type="checkbox"/>	SPEED	直値	<input type="checkbox"/>	表	NEXT	<input type="checkbox"/>	×
L	<input type="checkbox"/>		@P	<input type="checkbox"/>	グローバル変数: 値	<input type="checkbox"/>		グローバル変数: 値	<input type="checkbox"/>	表			
			@E	<input type="checkbox"/>	ローカル変数: 値	<input type="checkbox"/>		ローカル変数: 値	<input type="checkbox"/>	×			
			@数値	<input type="checkbox"/>			ACCEL	直値	<input type="checkbox"/>	表			
								グローバル変数: 値	<input type="checkbox"/>	表			
								ローカル変数: 値	<input type="checkbox"/>	×			
							DECEL	直値	<input type="checkbox"/>	表			
								グローバル変数: 値	<input type="checkbox"/>	表			
								ローカル変数: 値	<input type="checkbox"/>	×			

Ver. 1.9 の新機能

○→ 動作・編集可能 □→ 動作可能 表→ 編集時表示有り × 動作・編集不可 △→ コンパイル後動作可能
 送→ 送り動作のみ可能 →

コマンド	引数			
CHANGETOOL	ツール座標系番号			
	直値	<input type="checkbox"/>	送	
	グローバル変数: 値	<input type="checkbox"/>	送	
	ローカル変数: 値			×
CHANGEWORK	ユーザ座標系番号			
	直値	<input type="checkbox"/>	送	
	グローバル変数: 値	<input type="checkbox"/>	送	
	ローカル変数: 値			×
SPEED	移動速度			
	直値	<input type="checkbox"/>	送	
	グローバル変数: 値	<input type="checkbox"/>	送	
	ローカル変数: 値			×
JSPEED	移動速度			
	直値	<input type="checkbox"/>	送	
	グローバル変数: 値	<input type="checkbox"/>	送	
	ローカル変数: 値			×
ACCEL	加速度		減速度	
	直値	<input type="checkbox"/>	直値	<input type="checkbox"/>
	グローバル変数: 値	<input type="checkbox"/>	グローバル変数: 値	<input type="checkbox"/>
	ローカル変数: 値	×	ローカル変数: 値	×
JACCEL	軸加速度		軸減速度	
	直値	<input type="checkbox"/>	直値	<input type="checkbox"/>
	グローバル変数: 値	<input type="checkbox"/>	グローバル変数: 値	<input type="checkbox"/>
	ローカル変数: 値	×	ローカル変数: 値	×
DECEL	減速度			
	直値	<input type="checkbox"/>	送	
	グローバル変数: 値	<input type="checkbox"/>	送	
	ローカル変数: 値			×
JDECEL	軸減速度			
	直値	<input type="checkbox"/>	送	
	グローバル変数	<input type="checkbox"/>	送	
	ローカル変数: 値			×

Ver. 1.9 の新機能

○→ 動作・編集可能 □→ 動作可能 表→ 編集時表示有り ×→ 動作・編集不可 △→ コンパイル後動作可能
 送→ 送り動作のみ可能

コマンド	引数																																				
"グローバル変数・値"の関節参照範囲																																					
グローバル数値変数型	<table border="1"> <thead> <tr> <th colspan="2">配列番号</th> <th colspan="2">配列番号</th> </tr> </thead> <tbody> <tr> <td>直値</td> <td>○</td> <td></td> <td></td> </tr> <tr> <td>グローバル数値変数型</td> <td></td> <td>直値</td> <td>□</td> </tr> <tr> <td></td> <td></td> <td>グローバル変数</td> <td>×</td> </tr> <tr> <td></td> <td></td> <td>ローカル変数</td> <td>×</td> </tr> <tr> <td>ローカル数値変数</td> <td>×</td> <td></td> <td></td> </tr> <tr> <td>ローカル数値変数(配列)型</td> <td></td> <td>直値</td> <td>×</td> </tr> <tr> <td></td> <td></td> <td>グローバル変数</td> <td>×</td> </tr> <tr> <td></td> <td></td> <td>ローカル変数</td> <td>×</td> </tr> </tbody> </table>	配列番号		配列番号		直値	○			グローバル数値変数型		直値	□			グローバル変数	×			ローカル変数	×	ローカル数値変数	×			ローカル数値変数(配列)型		直値	×			グローバル変数	×			ローカル変数	×
	配列番号		配列番号																																		
	直値	○																																			
	グローバル数値変数型		直値	□																																	
			グローバル変数	×																																	
			ローカル変数	×																																	
	ローカル数値変数	×																																			
	ローカル数値変数(配列)型		直値	×																																	
			グローバル変数	×																																	
			ローカル変数	×																																	
ローカル数値変数	×																																				
ローカル数値変数(配列)型	×																																				
ローカル数値変数(配列)型	<table border="1"> <thead> <tr> <th colspan="2">直値</th> <th colspan="2">直値</th> </tr> </thead> <tbody> <tr> <td>グローバル数値変数型</td> <td>×</td> <td>直値</td> <td>×</td> </tr> <tr> <td></td> <td></td> <td>グローバル変数</td> <td>×</td> </tr> <tr> <td></td> <td></td> <td>ローカル変数</td> <td>×</td> </tr> <tr> <td>ローカル数値変数</td> <td>×</td> <td></td> <td></td> </tr> <tr> <td>ローカル数値変数(配列)型</td> <td></td> <td>直値</td> <td>×</td> </tr> <tr> <td></td> <td></td> <td>グローバル変数</td> <td>×</td> </tr> <tr> <td></td> <td></td> <td>ローカル変数</td> <td>×</td> </tr> </tbody> </table>	直値		直値		グローバル数値変数型	×	直値	×			グローバル変数	×			ローカル変数	×	ローカル数値変数	×			ローカル数値変数(配列)型		直値	×			グローバル変数	×			ローカル変数	×				
	直値		直値																																		
	グローバル数値変数型	×	直値	×																																	
			グローバル変数	×																																	
			ローカル変数	×																																	
	ローカル数値変数	×																																			
	ローカル数値変数(配列)型		直値	×																																	
			グローバル変数	×																																	
			ローカル変数	×																																	
	ローカル数値変数	×																																			
ローカル数値変数(配列)型	×																																				
グローバル変数:位	P,T,J																																				
ローカル変数:位	DEFPOS,DEFTRN,DEFJNT,DIM~POSITION,DIM~JOINT,DIM~TRANS																																				
グローバル数値変数型	I,F,D																																				
ローカル数値変数	DEFINT,DEFSNG,DEFDBL,%!,#.DIM~INTEGER,DIM~SINGLE,DIM~DOUBLE																																				
ローカル数値変数(配列)型																																					

例 P1,P[1] → P型グローバル変数の変数番号1
 例 P[1],P[[1]] → P型グローバル変数で、変数番号が1型グローバル変数の変数番号1の内容

変数の型の種類

※注1 : (1) 円弧動作中に一時停止を行なうと、起動方向(送り起動か戻し起動)が同じであれば途中から実行できますが、逆方向に起動させることはできません。
 また、一度逆方向に起動させると、「エラー27BE 現在位置では動作できません」が発生し、その後同じ起動方向に起動させようとしても起動できません。
 (2) 円弧動作は、現在位置がその行の一つ前の動作コマンドの目標点と同じ位置でないと実行できません。
 現在位置が一つ前の動作コマンドの目標点と異なる場合、「エラー027BE 現在位置では動作できません」が発生します。

※注2 : DEPART動作は、現在位置がその行の一つ前の動作コマンドの目標点と同じ位置でないと実行できません。
 現在位置が一つ前の動作コマンドの目標点と異なる場合、「エラー027BE 現在位置では動作できません」が発生します。

※注3 : ポーズの指定方法は1行につき1ポーズの場合にのみ簡単教示の編集および動作が可能です。
 従って、ポーズ列(1行に2つ以上のポーズを指定する書式)は簡単教示では扱えません。
 ポーズ列の例 MOVE P, P3,P4,P5 → 一行に3つのポーズ(P3,P4,P5)を指定している。

3.6. 簡単教示機能の注意事項

簡単教示における注意事項を記載します。

- (1) パス動作時の軌跡はティーチチェック、自動、外部の各のモードと若干異なる場合があります。
- (2) 姿勢フラグを -1 に指定した場合、送り起動と戻し起動で姿勢が異なる場合があります。
- (3) 簡単教示機能にて動作コマンドを実行すると、内部でアームセマフォを取得します。

このアームセマフォを開放するタイミングは

- ・ 1ステップ起動または1サイクル起動において、最終目標点までに到達したとき
- ・ プログラム表示画面を閉じたとき
- ・ ショートカットメニューの[F8 プロリセット]を実行したとき
- ・ ティーチチェック、自動、外部の各モードにおいて[F7 プロリセット]を実行したとき（このプログラムのみ、すべてのプログラムどちらを選択した場合でも同じ）
- ・ TP/OP/MP以外が原因のレベル3のエラーが発生した場合
- ・ レベル4以上のエラーが発生した場合
- ・ I/O標準モードにおける外部からのプログラムリセット（データ領域を負数にした場合のみ）
- ・ I/O互換モードにおける外部からのプログラムリセット+ 運転準備スタート時です。

1ステップ起動などで途中で一時停止を行なうと、手動動作及び動作モードの切替え（各軸、X-Y動作など）が行なえません。動作の途中で手動動作及び動作モードの切替えを行なう場合は、ショートカットメニューのプロリセットなどを行なってアームセマフォを開放してください。

3.7. 簡単教示用の使用条件

簡単教示用の使用条件を下記に示します。

番号	表示	出荷時値	電源投入時	内容	備考
199	簡単教示の円弧動作許容値	100	100	簡単教示時の円弧動作の位置ずれ許容値	通常は変更しないでください。

5. HS-E ロボットのダイレクトモードの操作方法

【関連ページ】：「操作ガイド」の P5-78

新設のHS-E ロボットは、Z軸のエアーバランスシリンダを持たないため、従来の4軸ロボットとダイレクトモードの操作方法が異なります。

■ダイレクトモードの操作経路：

[基本画面] — [F2 アーム] — [F6 補助機能.] — [F3 ダイレクト]

モータの電源が切れた状態で、(ティーチングペンダントを使用せずに)手でロボットを動かした後、ポーズ変数(J、P、T型変数)のティーチングができるようになります。

(変数に現在位置を教示する場合は、通常、モータの電源が入っている必要があります。)

5.1. HS-E 以外の4軸ロボットの場合 (従来の方法)

- (1) [保守機能 (アーム)] ウィンドウで[F3 ダイレクト]を押すと、[Z軸エアーバランス調整]ダイアログボックスが表示され、エアーバランスシリンダの調整を行うことができます。この「Z軸エアーバランス調整」は、コントローラの電源を入れた後、ダイレクトモードをはじめて設定する場合のみ実行が必要です。

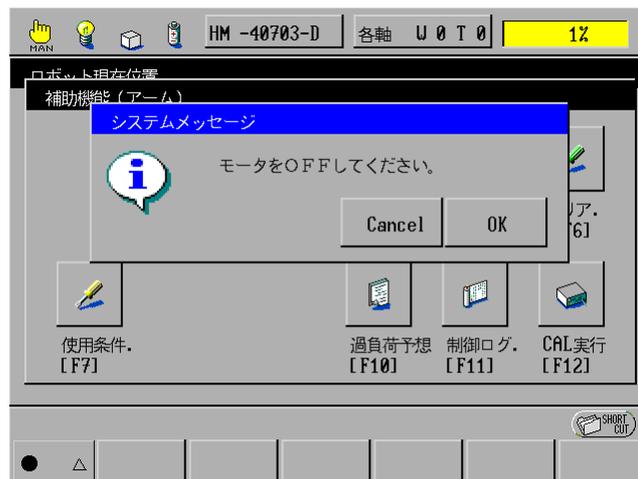


- (2) 指示に従って、エアーバランスの調整を行います。
「エアー圧力 OK!」となっていない場合は、ティーチングペンダントの指示に従ってエアーバランスの調整を行った後、[OK]ボタンを押します。
「エアー圧力 OK!」の場合は、[OK]を押します。

(3) 次のような [システムメッセージ] ダイアログボックスが表示されます。



[OK]を押すと、[システムメッセージ]ダイアログボックスが表示されるので、モータの電源をOFFし、[OK]を押します。

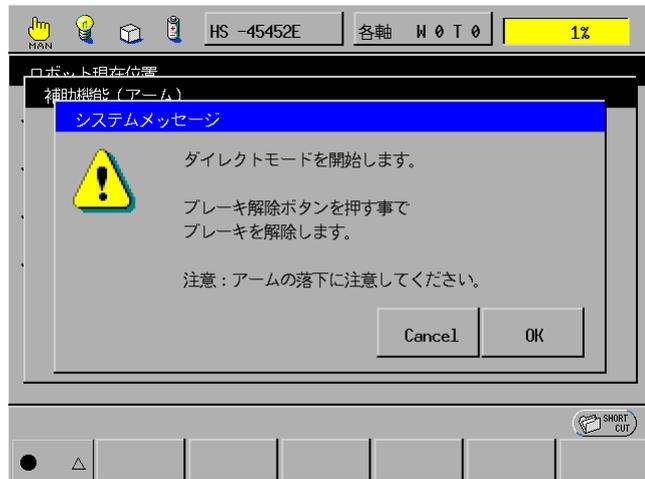


(4) ブレーキを解除するメッセージが表示されます。[OK]を押します。



5.2. HS-E ロボットの場合

- (1) [保守機能 (アーム)] ウィンドウで[F3 ダイレクト]を押すと、ダイレクトモード開始のダイアログが表示されます。ここで[OK]を押すとブレーキ解除状態になりますが、HS-Eロボットの場合は、ロボット上部のブレーキ解除ボタンを押すことでブレーキ解除が行なわれます。



- (2) [OK]でダイレクトモードが開始されます。



6. シリアル通信回線の追加 (イーサネットサーバ、クライアント対応)

【関連ページ】:「プログラミングマニュアル」の P2-8

RC5ロボットコントローラ用の通信回線用ポートにEthernetサーバポートおよびEthernetクライアントポートの増設を可能にしました。それに伴う追記内容を加えて説明します。

6.1. 回線番号

ロボットコントローラの回線番号とチャンネル番号の対応は以下のとおりです。

ロボットコントローラの回線番号

回線番号	チャンネル番号	用途
0	ch1	ティーチングペンダント接続ポート
1	ch2	汎用ポート
2	ch3	(予約)
3	ch4	(予約)
4 ~ 7 (注1)	ch5 ~ ch8	Ethernetサーバポート
8 ~ 15 (注1)	ch9 ~ ch16	Ethernetクライアントポート(注2)

注1: Ver. 1.9以降

注2: Ethernetクライアントポートを使用するときは、[7.2 シリアルバイナリ通信]の com_encom文、com_discom文を使用してポートのオープン、クローズを行なう必要があります。

注3: ロボットコントローラのクライアント (Client)、サーバ (Server) の設定は操作ガイド「5.7項」を参照してください。

6.2. 通信コマンド

RS232CおよびEthernetを介して通信するPAC言語のコマンドには、次の4つがあります。各コマンドの説明は「13.2項」を参照してください。

注: Ethernet使用時においても通信コマンドの形式に変更はありません。

- PRINT文(出力)
- WRITE文(出力)
- INPUT文(入力)
- LINEINPUT文(行入力)

PRINT文とWRITE文は出力形式が異なります。WRITE文の形式で出力された文字列は、INPUT文で正しく入力できます。

6.3. 通信バッファのクリア

FLUSH文によって、RS232CおよびEthernetの通信バッファを空にします。

通信を開始する前に必ず実施してください。

- FLUSH: 「13.2項」を参照。

6.4. ロボットコントローラのクライアントの設定

操作経路: [F6 設定]—[F5 通信設定.]—[F10 クライアント]

ロボットコントローラのクライアントを設定します。

- (1) [通信設定メニュー]で[F10 クライアント]を押すと、[クライアント通信設定]ウィンドウが表示されます。



- (2) 接続先のIPアドレス設定するにはIPアドレスを選択し、[F5 値変更]を押し、更に[F5 Edit]を押します。

次のようなテンキーが表示されます。



- (3) テンキーを使用して目的の値を入力し、[OK]ボタンを押すと、入力した新しい値が[クライアント通信設定]ウィンドウに表示されます。

- (4) 接続先のポート番号を設定するには、ポートを選択し、[F5 値変更]を押します。
次のようなテンキーが表示されます。



- (5) テンキーを使用して目的の値を入力し、[OK]ボタンを押すと、入力した新しい値が[クライアント通信設定]ウィンドウに表示されます。
- (6) デリミタを設定するには、デリミタを選択し、[F5 値変更]を押します。
次のような[デリミタ選択]ウィンドウが表示されます。



- (7) デリミタコードを選択し、[OK]ボタンを押すと、選択した新しい値が[クライアント通信設定]ウィンドウに表示されます。
- (8) 設定値を確認し、[OK]ボタンを押すと、新しい設定値が有効になります。
[OK]ボタンの代わりに[Cancel]ボタンを押すと、設定値の変更がキャンセルされます。[クライアント通信設定]ウィンドウは、[OK]ボタンか[Cancel]ボタンを押すと閉じます。

6.5. ロボットコントローラのサーバの設定

操作経路: [F6 設定]—[F5 通信設定.]—[F11 サーバ]

ロボットコントローラのサーバを設定します。

- (1) [通信設定メニュー]で[F11 サーバ]を押すと、[サーバ通信設定]ウィンドウが表示されます。



- (2) ポート番号を設定するには、ポートを選択し、[F5 値変更]を押します。
次のようなテンキーが表示されます。



テンキーを使用して目的の値を入力し、[OK]ボタンを押すと、入力した新しい値が[サーバ通信設定]ウィンドウに表示されます。

- (4) デリミタを設定するには、デリミタを選択し、[F5 値変更]を押します。
次のような[デリミタ選択]ウィンドウが表示されます。



- (5) デリミタコードを選択し、[OK]ボタンを押すと、選択した新しい値が[サーバ通信設定]ウィンドウに表示されます。
- (6) 設定値を確認し、[OK]ボタンを押すと、新しい設定値が有効になります。
[OK]ボタンの代わりに[Cancel]ボタンを押すと、設定値の変更がキャンセルされます。[サーバ通信設定]ウィンドウは、[OK]ボタンか[Cancel]ボタンを押すと閉じます。

6.6. RS232C ポート、Ethernet（サーバ、クライアント）のコマンド

INPUT (ステートメント) 【SLIM 準拠】

機能 RS232C および Ethernet ポートからのデータを得ます。

書式 INPUT [#<回線番号>,<変数名>[,<変数名>…]

説明 RS232C および Ethernet ポートに受信したデータを、<変数名>で指定した変数に格納します。

<回線番号>には、使用する RS232C および Ethernet ポートの回線番号を指定します。<回線番号>を省略するとデフォルト値の ch2 となります。ch1 はティーチングペンダント用を使用するため指定できません。（「6.1 回線番号」参照）

複数の情報を、それと同数の変数で受ける場合は、個々の変数間にカンマ(,)を付けます。

デリミタ (CR または CR+LF) の直前までのすべてを、指定した変数に格納します。デリミタは、変数には読み込まれません。

ボーレートは、システムパラメータで指定します。

注意①：データの受信に先立って、受信データの入力バッファに残っているデータをクリアするために、FLUSHコマンドを実行してください。

②：入力したデータが、代入する変数の型が表せる値の最大値を超えている場合、結果はその型の最大値となります。

関連項目 FLUSH、PRINT、WRITE

用例

```
DIM li1 As Integer
DEFSTR ls1, ls2, ls3, ls4
INPUT #1, ls1           ' ch2 からのデータを ls1 に書きます。
INPUT #1, li1, ls2, ls3 ' ch2 からのデータを li1, ls2, ls3 に書きます。
INPUT ls4              ' ch2 からのデータを ls4 に書きます。
```

LINEINPUT (ステートメント)

機能

RS232C および Ethernet ポートからデリミタまでのデータを読み込み、文字列型変数に代入します。

書式

LINEINPUT [#<回線番号>] <文字列型変数名>

説明

RS232C および Ethernet ポートに受信したデータから、デリミタ (CR または CR+LF) の直前までのすべての文字を、<文字列型変数名>で指定した変数に格納します。デリミタは、変数には読み込まれません。

<回線番号>には、使用する RS232C および Ethernet ポートの回線番号を指定します。<回線番号>を省略するとデフォルト値の ch2 となります。ch1 はティーチングペンダント用に使用するため指定できません。(「6.1 回線番号」参照)

<文字列変数名>には、文字列型変数の名前を指定します。

用例

```
DEFSTR 1s1, 1s2, 1s3
LINEINPUT #0, 1s1      ' ポート 1 から文字列を受信し、1s1 へ代入します。
LINEINPUT #1, 1s2      ' ポート 2 から文字列を受信し、1s2 へ代入します。
LINEINPUT 1s3          ' ポート 2 から文字列を受信し、1s3 へ代入します。
```

PRINT (ステートメント) 【SLIM 準拠】

機能

RS232C および Ethernet ポートからデータを出力します。

書式

PRINT [#<回線番号>] <メッセージ> [<セパレータ><メッセージ>...] [<セパレータ>]

説明

RS232C および Ethernet ポートから、<メッセージ>の値を外部へ出力します。

<回線番号>には、使用する RS232C および Ethernet ポートの回線番号を指定します。<回線番号>を省略するとデフォルト値の ch2 となります。ch1 はティーチングペンダント用に使用するため指定できません。(「6.1 回線番号」参照)

<セパレータ>には、カンマ(,)とセミコロン(;)が使用できます。

カンマ(,)の場合は、カンマと後ろの<メッセージ>の間に、スペース文字を入れます。

<メッセージ>の後にカンマ(,)を指定し、その後に<メッセージ>がなかった場合は、最後の<メッセージ>の後にデリミタ (CR または CR+LF) を送出して、出力を完了します。

セミコロン(;)の場合は、<メッセージ>との間にはスペース文字を入れません。

<メッセージ>の後にセミコロン(;)を指定し、その後に<メッセージ>がなかった場合は、最後の<メッセージ>の後にデリミタ (CR または CR+LF) を送出して、出力を完了します。

ポーズ型データを出力した場合は、各要素をスペース文字で区切って、出力されます。

関連項目 WRITE

用例

```
DEFINT li1, li2
DEFSTR ls1, ls2
PRINT #1, li1, ls1; ls2      ' li1, ls1, ls2 の値を ch2 から出力します。(li1 と ls1
                             ' の間はスペースが空き、ls1 と ls2 の間はスペースが空
                             ' きません。)
PRINT li2                   ' li2 の値を ch2 から出力します。
```

WRITE (ステートメント)

機能 RS232C および Ethernet ポートからデータを出力します。

書式 WRITE [#<回線番号>], <メッセージ>[, <メッセージ>…]

説明

RS232C および Ethernet ポートから、<メッセージ>の値を外部へ出力します。

<回線番号>には、使用する RS232C および Ethernet ポートの回線番号を指定します。<回線番号>を省略するとデフォルト値の ch2 となります。ch1 はティーチングペンダント用に使用するため指定できません。（「6.1 回線番号」参照）

複数の<メッセージ>を書き出す場合は、カンマ(,)で区切ります。

PRINT 文とは、以下の点が異なります。

- 文字列のデータは、二重引用符(")で囲まれて、出力されます。
- 出力情報の終わりに、デリミタ (CR または CR+LF) が付きます。
- <メッセージ>を区切っているカンマ(,)は、そのまま出力されます。
- ポーズ型データを出力した場合は、各要素をカンマ(,)で区切って、出力されます。

関連項目 PRINT

用例

```
DEFINT li1, li2
DEFSTR ls1, ls2
WRITE #1, li1, ls1, ls2     ' li1, ls1, ls2 の値を ch2 から出力します。
WRITE li2                   ' li2 の値を ch2 から出力します。
```

FLUSH (ステートメント)

機能 入力バッファをクリアします。

書式 FLUSH [#<回線番号>]

説明 RS232C および Ethernet ポートの入力バッファをクリアします。
<回線番号>には、入力バッファをクリアする RS232C および Ethernet ポートの回線番号を指定します。<回線番号>を省略するとデフォルト値の ch2 となります。ch1 はテーチングペンダント用に使用するため指定できません。(「6.1 回線番号」参照)
データの受信に先立って、受信データの入力バッファに残っているデータをクリアするために、FLUSH コマンドを実行してください。

関連項目 INPUT

用例

FLUSH #1	' ch2 の回線の入力バッファをクリアします。
FLUSH	' ch2 の回線の入力バッファをクリアします。

7. シリアルバイナリ通信にイーサネットの使用が可能

Ver. 1.4以前のコントローラでは、RS-232Cポートを利用した通信をASCIIコードに限っていました。Ver. 1.5以降から1バイト単位でのバイナリデータの通信をサポートするようになり、接続可能通信機器が拡充されました。

※Ver. 1.743以降からシリアル通信同様にEthernetの使用が可能になりました。

7.1. 使用方法

コントローラと外部機器をRS-232CケーブルまたはEthernetケーブルで接続し、以下のコマンドによりバイナリデータ通信を行ないます。

■データ入出力コマンド

- | | |
|---|------------|
| (1) 1バイトデータ出力 | printb |
| (2) 1バイトデータ入力 | inputb |
| (3) 複数バイトデータ出力 | lprintb |
| (4) 複数バイトデータ入力 | linputb |
| (5) COMポート占有および
クライアントポートのOPEN | com_encom |
| (6) COMポート開放および
クライアントポートのCLOSE | com_discom |
| (7) COMポート状態取得および
Ethernetポートの接続状態取得 | com_state |

※com_encom、com_discom、com_stateについてはEthernet対応にともない機能の拡張がされていますが、従来の機能、形式に変更はありません。

■通信方式

- (1) 通信方式 : RS-232C
- (2) 通信ポート : コントローラRS-232Cポート、 μ VisionボードRS-232Cポート
- (3) ピン配列 : 従来と同様
- (4) 通信条件 :

コントローラ RS-232C ポート	従来と同様で可変
μ Vision ボード RS-232C ポート	転送速度 : 9600bps
	ビット長 : 8
	パリティビット : なし
	ストップビット : 1
	フロー制御 : なし

7.2. シリアルバイナリ通信 (RS-232C および Ethernet ポート) のコマンド

printb

機能

1 バイトデータを RS-232C および Ethernet ポートに出力します。

書式

printb #<ポート番号>, <出力データ格納 I 変数>

<ポート番号> 出力ポート番号

(1 : コントローラ RS-232C ポート、-1 : μ Vision ボード RS-232C ポート
4~7 : Ethernet サーバポート、8~15 : Ethernet クライアントポート)

<出力データ格納 I 変数>

出力データが格納された I 型変数

説明

<出力データ格納 I 変数> で指定された I 型変数の下位バイトを、指定ポートに出力します。

用例

```
'!TITLE "<タイトル>"
PROGRAM sample
    .
    .
    printb #1, I10      ' I10 に格納されたデータの下位バイトを RS-232C ポートに
                        ' 出力
    .
    .
end
```

inputb

機能

1 バイトデータを RS-232C および Ethernet ポートから入力します。

書式

inputb #<ポート番号>, <入力データ格納 I 変数>

<ポート番号> 入力ポート番号

(1 : コントローラ RS-232C ポート、-1 : μ Vision ボード RS-232C ポート
4~7 : Ethernet サーバポート、8~15 : Ethernet クライアントポート)

<入力データ格納 I 変数>

入力データが格納される I 型変数の番号

説明

<入力データ格納 I 変数> で指定された I 型変数に、ポートから入力したデータを格納します。

注 : このコマンドでデータ入力を行なう場合、データがポートに存在しないとエラーが発生します。このコマンドを実行する前に外部からデータを送信してください。データの有無の確認には com_state コマンドを使用してください。

用例

```
'!TITLE "<タイトル>"
PROGRAM sample
  .
  .
  inputb #1, I10      ' I10 に RS-232C ポートから入力したデータを格納
  .
  .
end
```

lprintb

機能

複数バイトデータを RS-232C および Ethernet ポートに出力します。

書式

lprintb #<ポート番号>, <格納配列先頭要素>, <出力バイト数>

<ポート番号> 出力ポート番号

(1 : コントローラ RS-232C ポート、-1 : μ Vision ボード RS-232C ポート

4~7 : Ethernet サーバポート、8~15 : Ethernet クライアントポート)

<格納配列先頭要素>

出力データが格納された配列の先頭要素

<出力バイト数>

出力するデータのバイト数

説明

出力データが格納された配列の指定要素から指定出力バイト数分だけ指定ポートにデータを出力します。

用例

```
'!TITLE "<タイトル>"
PROGRAM sample
  .
  .
  lprintb #1, I64, 30      ' I64 から I93 までのデータの下位バイトを連続して
                          ' RS-232C ポートに出力
  .
  .
end
```

linputb

機能 複数バイトデータを RS-232C および Ethernet ポートから入力します。

書式 linputb #<ポート番号>, <格納配列先頭要素>, <入力バイト数>

<ポート番号> 入力ポート番号

(1 : コントローラ RS-232C ポート、-1 : μ Vision ボード RS-232C ポート

4~7 : Ethernet サーバポート、8~15 : Ethernet クライアントポート)

<格納配列先頭要素>

入力データが格納される配列の先頭要素

<入力バイト数>

入力するデータのバイト数

説明 配列の指定要素から、指定入力バイト数分だけ指定ポートからデータを入力します。

注：このコマンドでデータ入力を行なう場合、データがポートに存在しないとエラーが発生します。このコマンドを実行する前に外部からデータを送信してください。データの有無の確認には com_state コマンドを使用してください。

用例

```
'!TITLE "<タイトル>"
PROGRAM sample
    .
    .
    .
    linputb #1, I64, 30      ' I64 から I93 までにデータを連続して RS-232C ポート
                           ' から入力
    .
    .
    .
end
```

com_encom

機能

RS-232C ポートをバイナリ通信のために占有します。(COM ポート占有)
Ethernet クライアントポート使用時にクライアントポートのオープンをおこない接続を確立します。

書式

com_encom #<ポート番号>

説明

バイナリ通信中に WINCAPS II など PC \leftrightarrow コントローラ間通信を行なう際、通信データが交じり合わないようにします。

バイナリ通信が終了したら、com_discom コマンドで通信ポートを開放する必要があります。

注：このコマンドを実行したあとで、WINCAPSII などデータを送信してもバイナリ通信データとして取り扱われます。RS-232C ポートを閉じてしまうことではありませんのでご注意ください。

注：Ethernet 接続では、このコマンドを実行した後も WINCAPSII とのデータ通信に影響はありません。

用例

```
'!TITLE "<タイトル>"
PROGRAM sample
    .
    .
    .
    com_encom #1      ' ポートのバイナリ通信占有
    .
    .
    .
end
```

com_discom

機能 RS-232C ポートをバイナリ通信終了のため開放します。(COM ポート開放)
Ethernet クライアントポート使用時にクライアントポートのクローズをおこない接続を切断します。

書式 com_discom #<ポート番号>

説明 バイナリ通信を行なうために占有されていた通信ポートを、他の用途のために開放します。

注： このコマンドを実行するとデータの交じり合いを防ぐため、RS-232C ポートを一度クリアします。

注： Ethernet 使用時この処理を行うと未送信データは送信完了を待たずに切断する為、受信側において受信待ちし続けることがあります。

用例

```
'!TITLE "<タイトル>"
PROGRAM sample
    .
    .
    com_discom #1      'ポートのバイナリ通信からの解放
    .
    .
end
```

com_state [Ver. 1.5 以降]

機能 RS-232C および Ethernet ポートの状態を取得します。

書式 com_state #<ポート番号>, <状態格納 I 型変数>

説明 RS-232C および Ethernet 通信バッファの残りデータ数を取得します。ただし通信ポートエラー発生時は-1 を返します。

Ethernet 使用時は該当ポートの接続が確立していないときも-1 を返します。

用例

```
'!TITLE "<タイトル>"
PROGRAM sample
    .
    .
    com_state #1, I280      ' I280 に通信バッファに溜まっているデータ数を
                            ' 取得する
    .
    .
end
```

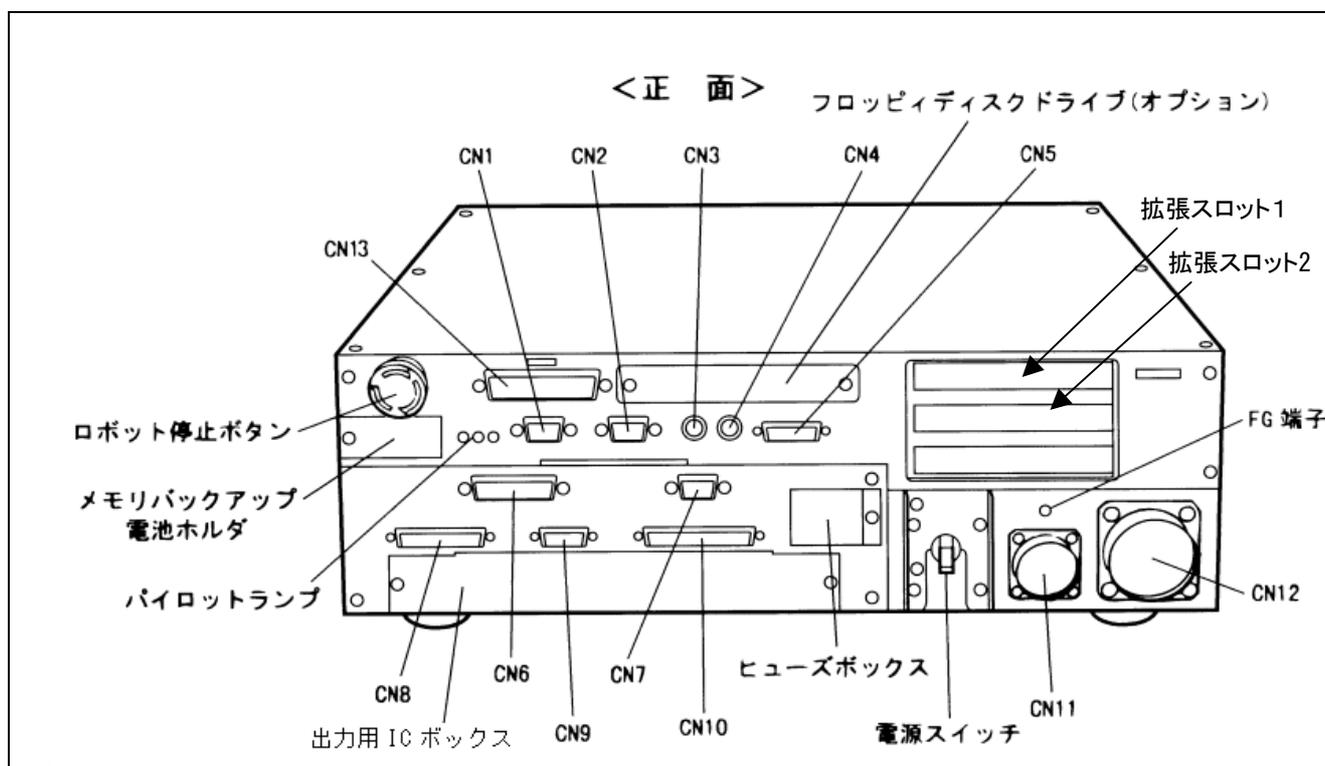
8. CC-Link リモートデバイスボード（オプション）の新設

ロボットコントローラにCC-Linkリモートデバイスボードを内蔵することで、CC-Linkに準拠するメーカ多機種の世界機器とI/Oデータの交換を容易に行なうことができます。

このとき、ロボットコントローラはオープンなネットワークであるCC-Linkに準拠したシリアル通信のリモートデバイスになります。

CC-Linkリモートデバイスボードはロボットコントローラの拡張スロット1またはスロット2に内蔵されます。

（オプション機器説明書 「増設ボードの取り付け」参照。）

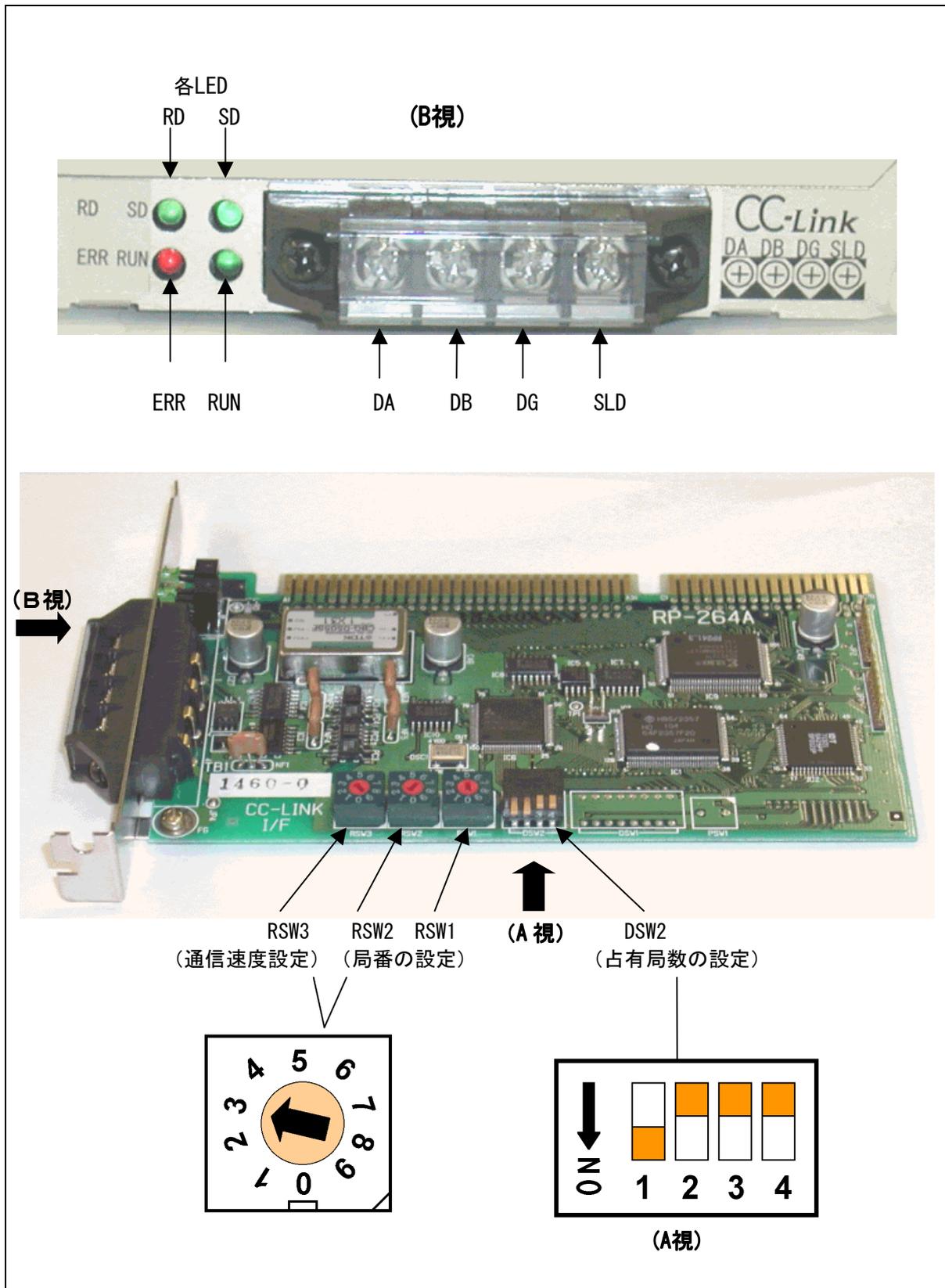


注意：CC-Link用コントローラまたはCC-Linkリモートデバイスボードには、付属品としてCN8 (INPUT)、CN9 (HAND I/O)、CN10 (OUTPUT)の平行I/O用のコネクタセットが準備されています。平行I/Oを使用する場合は、「RC5ロボットコントローラ インターフェース説明書」を参照して、この付属品のコネクタセットを活用してください。

8.1. 製品仕様

8.1.1. CC-Link リモートデバイスボード各部の名称

CC-Link リモートデバイスボード各部の名称を下図に示します。



8.1.2. 各部の機能とボードの設定

(1) LED 表示の意味

4 個の LED (RUN、ERR、SD、RD) の状態から下表のように CC-Link の動作状態を確認することができます。

LEDの状態				動作
RUN	ERR	SD	RD	
				正常交信しているが、ノイズで CRC エラーが時々発生している。
	0.4s 			リセット解除時のポーレート・局番設定からポーレートまたは局番設定が変化した。
				— (ありえない動作状態)
				受信データが CRC エラーとなり、応答できない。
				— (ありえない動作状態)
				正常交信
				— (ありえない動作状態)
				自局宛データがこない。
				— (ありえない動作状態)
				ポーリング応答はしているが、リフレッシュ受信が CRC エラー。
				— (ありえない動作状態)
				自局宛データが CRC エラー。
				— (ありえない動作状態)
				リンク起動されていない。
				— (ありえない動作状態)
				自局宛データが無いか、ノイズにより自局宛を受信不可。
				断線等でデータを受信できない。電源断または H/W セット中。
				ポーレート、局番設定不正

注意： ロボットコントローラは、シーケンサがRUN状態にならないとCC-Link通信を行いません。LEDの状態が正常交信を示しているのに、ティーチングペンダント等にCC-Linkの通信異常を知らせるメッセージが表示されている場合、シーケンサをRUN状態にすると通信異常が解除されます。

(2) 占有局数の設定

占有局数はDSW2で設定します。

ディップスイッチ

ON OFF OFF OFF

1 2 3 4

ディップスイッチを下にすると「ON」、上にすると「OFF」になります。

左図は占有局数を2局に設定した例

注意：設定は必ずコントローラの電源がOFFの状態で行なってください。

DSW2の設定

下表に従って、占有局数の設定を行なってください。

占有局数の設定

DSW2				占有局数
1	2	3	4	
OFF	OFF	OFF	OFF	設定不可
ON	OFF	OFF	OFF	2局
OFF	ON	OFF	OFF	3局
ON	ON	OFF	OFF	4局
OFF	OFF	ON	OFF	設定不可
...	
ON	ON	ON	ON	

(3) 局番の設定

局番の設定はRSW1、RSW2で下図のように設定します。

RSW2 (5の例)

RSW1 (3の例)

RSW2で1の位、RSW1で10の位を“中央の矢印を合わせて”設定します。

左図は局番35の例を示します。

局番の設定可能範囲：1～63

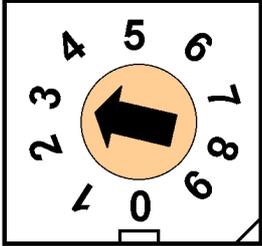
注意：設定は必ずコントローラの電源がOFFの状態で行なってください。

局番の設定

(4) 通信速度の設定

通信速度はRSW3で、下表に従って設定します。

通信速度の設定

RSW3の設定方法	RSW3	通信速度
 <p>RSW3 (3の例)</p> <p>注意： 設定は必ずコントローラの電源がOFFの状態で行なってください。</p>	0	156 kbps
	1	625 kbps
	2	2.5 Mbps
	3	5 Mbps
	4	10 Mbps
	5	設定不可
	...	
	...	
	...	
	...	
9		

8.2. I/O データの割付け

I/Oデータの割付けには、標準モード割付けと互換モード割付けの2種類があります。

8.2.1. 標準モード割付け（4局占有の場合）

入力データ(マスターリモートデバイス)			出力データ(リモートデバイス→マスター)		
I/Oポート番号	信号名	デバイスNo.(注1)	I/Oポート番号	信号名	デバイスNo.(注1)
512	ステップ停止(全タスク)	RYn0	768	-	RXn0
513	予約	RYn1	769	ロボット運転中	RXn1
514	瞬時停止(全タスク)	RYn2	770	ロボット異常	RXn2
515	スロープ信号	RYn3	771	サーボON中	RXn3
516	割り込みスキップ	RYn4	772	ロボット初期化完了	RXn4
517	-	RYn5	773	自動モード	RXn5
518	-	RYn6	774	外部モード	RXn6
519	コマンド・データ領域奇数パリティ	RYn7	775	バッテリー切れ警告	RXn7
520	データ領域1第0ビット	RYn8	776	ロボット警告	RXn8
521	データ領域1第1ビット	RYn9	777	コンティニュースタート許可	RXn9
522	データ領域1第2ビット	RYnA	778	SSモード出力	RXnA
523	データ領域1第3ビット	RYnB	779	予約	RXnB
524	データ領域1第4ビット	RYnC	780	予約	RXnC
525	データ領域1第5ビット	RYnD	781	予約	RXnD
526	データ領域1第6ビット	RYnE	782	コマンド処理完了	RXnE
527	データ領域1第7ビット	RYnF	783	ステータス領域奇数パリティ	RXnF
528	データ領域2第0ビット	RY(n+1)0	784	ステータス領域第0ビット	RX(n+1)0
529	データ領域2第1ビット	RY(n+1)1	785	ステータス領域第1ビット	RX(n+1)1
530	データ領域2第2ビット	RY(n+1)2	786	ステータス領域第2ビット	RX(n+1)2
531	データ領域2第3ビット	RY(n+1)3	787	ステータス領域第3ビット	RX(n+1)3
532	データ領域2第4ビット	RY(n+1)4	788	ステータス領域第4ビット	RX(n+1)4
533	データ領域2第5ビット	RY(n+1)5	789	ステータス領域第5ビット	RX(n+1)5
534	データ領域2第6ビット	RY(n+1)6	790	ステータス領域第6ビット	RX(n+1)6
535	データ領域2第7ビット	RY(n+1)7	791	ステータス領域第7ビット	RX(n+1)7
536	データ領域2第8ビット	RY(n+1)8	792	ステータス領域第8ビット	RX(n+1)8
537	データ領域2第9ビット	RY(n+1)9	793	ステータス領域第9ビット	RX(n+1)9
538	データ領域2第10ビット	RY(n+1)A	794	ステータス領域第10ビット	RX(n+1)A
539	データ領域2第11ビット	RY(n+1)B	795	ステータス領域第11ビット	RX(n+1)B
540	データ領域2第12ビット	RY(n+1)C	796	ステータス領域第12ビット	RX(n+1)C
541	データ領域2第13ビット	RY(n+1)D	797	ステータス領域第13ビット	RX(n+1)D
542	データ領域2第14ビット	RY(n+1)E	798	ステータス領域第14ビット	RX(n+1)E
543	データ領域2第15ビット	RY(n+1)F	799	ステータス領域第15ビット	RX(n+1)F
544	コマンド領域第0ビット	RY(n+2)0	800	汎用出力(OUTPUT m) (注2)	RX(n+2)0
545	コマンド領域第1ビット	RY(n+2)1	801		RX(n+2)1
546	コマンド領域第2ビット	RY(n+2)2	:		:
547	コマンド領域第3ビット	RY(n+2)3	:		:
548	予約	RY(n+2)4	:		:
549	予約	RY(n+2)5	:		:
550	予約	RY(n+2)6	:		:
551	予約	RY(n+2)7	:		:
552	汎用入力(INPUT m) (注2)	RY(n+2)8	:		:
553		RY(n+2)9	:		:
:		:	:		:
:		:	:		:
:		:	:		:
621		RY(n+6)D	877		RX(n+6)D
622		RY(n+6)E	878		RX(n+6)E
623		RY(n+6)F	879	RX(n+6)F	

注1： マスタ局から見た時の入出力名称（nは本ボードの先頭デバイス番号の二桁目。16進数表記）です。

注2： mにはI/Oポート番号が入ります。

8.2.2. 互換モード割付け（4局占有の場合）

入力データ(マスタ → リモートデバイス)			出力データ(リモートデバイス → マスタ)		
I/Oポート 番号	信号名	デバイスNo. (注1)	I/Oポート 番号	信号名	デバイスNo. (注1)
512	ステップ停止	RYn0	768	-	RXn0
513	コンティニュースタート	RYn1	769	ロボット運転中	RXn1
514	瞬時停止	RYn2	770	ロボット異常	RXn2
515	運転準備スタート	RYn3	771	自動モード	RXn3
516	割り込みスキップ	RYn4	772	外部モード	RXn4
517	プログラムスタート	RYn5	773	プログラムスタートリセット	RXn5
518	予約	RYn6	774	未使用	RXn6
519	予約	RYn7	775	未使用	RXn7
520	プログラム番号選択第0ビット	RYn8	776	ロボット電源入り完了	RXn8
521	プログラム番号選択第1ビット	RYn9	777	サーボON中	RXn9
522	プログラム番号選択第2ビット	RYnA	778	CAL完了	RXnA
523	プログラム番号選択第3ビット	RYnB	779	ティーチング中	RXnB
524	プログラム番号選択第4ビット	RYnC	780	1サイクル終了	RXnC
525	プログラム番号選択第5ビット	RYnD	781	バッテリー切れ警告	RXnD
526	プログラム番号選択第6ビット	RYnE	782	ロボット警告	RXnE
527	プログラム番号選択第7ビット	RYnF	783	コンティニュー許可	RXnF
528	モータ電源入り	RY(n+1)0	784	ERROR 1の位 2 ⁰	RX(n+1)0
529	CAL実行	RY(n+1)1	785	ERROR 1の位 2 ¹	RX(n+1)1
530	予約	RY(n+1)2	786	ERROR 1の位 2 ²	RX(n+1)2
531	SP100	RY(n+1)3	787	ERROR 1の位 2 ³	RX(n+1)3
532	外部モード切り替え	RY(n+1)4	788	ERROR 10の位 2 ⁰	RX(n+1)4
533	プログラムリセット	RY(n+1)5	789	ERROR 10の位 2 ¹	RX(n+1)5
534	ロボット異常クリア	RY(n+1)6	790	ERROR 10の位 2 ²	RX(n+1)6
535	予約	RY(n+1)7	791	ERROR 10の位 2 ³	RX(n+1)7
536	汎用入力(INPUT m) (注2)	RY(n+1)8	792	ERROR 100の位 2 ⁰	RX(n+1)8
537		RY(n+1)9	793	ERROR 100の位 2 ¹	RX(n+1)9
:		:	794	ERROR 100の位 2 ²	RX(n+1)A
:		:	795	ERROR 100の位 2 ³	RX(n+1)B
:		:	796	SSモード出力	RX(n+1)C
:		:	797	未使用	RX(n+1)D
:		:	798	未使用	RX(n+1)E
:		:	799	未使用	RX(n+1)F
:		:	800	汎用出力(OUTPUT m) (注2)	RX(n+2)0
:		:	801		RX(n+2)1
:	:	:	:		
:	:	:	:		
:	:	:	:		
:	:	:	:		
621	:	RY(n+6)D	877		RX(n+6)D
622	:	RY(n+6)E	878		RX(n+6)E
623	:	RY(n+6)F	879		RX(n+6)F

注1： マスタ局から見た時の入出力名称（nは本ボードの先頭デバイス番号の二桁目。16進数表記）です。

注2： mにはI/Oポート番号が入ります。

8.2.3. 占有局数に対する入出力範囲

占有局数に対する入出力範囲を以下の表に示します。

占有局数を変更することで汎用入出力の数が増減する形となります。

各占有局数に対する入出力範囲

占有局数	入力データ		出力データ	
	I/Oポート番号	デバイスNo. (注1)	I/Oポート番号	デバイスNo. (注1)
2局占有	512~559	RYn0~RY(n+2)F	768~815	RXn0~RX(n+2)F
3局占有	512~591	RYn0~RY(n+4)F	768~847	RXn0~RX(n+4)F
4局占有	512~623	RYn0~RY(n+6)F	768~879	RXn0~RX(n+6)F

注1: マスタ局から見た時の入出力名称(nは本ボードの先頭デバイス番号の二桁目。16進数表記)です。

8.2.4. システム領域

最終I/Oポート番号の次のポートから16点分はシステム領域となります。
また、当ボードで使用している信号は以下の表に示します。

システム領域詳細

入力データ(マスタ→リモートデバイス)			
I/Oポート番号 (注1)	信号名	デバイスNo. (注2)	使用状況
$\alpha+0$	システム領域 リザーブ(機能別に設定)	$RY(n+\beta)0$	未使用
:		:	
$\alpha+7$		$RY(n+\beta)7$	
$\alpha+8$	インシャルデータ処理完了フラグ	$RY(n+\beta)8$	
$\alpha+9$	インシャルデータ設定要求フラグ	$RY(n+\beta)9$	
$\alpha+10$	エラーリセット要求フラグ	$RY(n+\beta)A$	
$\alpha+11$	リザーブ	$RY(n+\beta)B$	
$\alpha+12$	リザーブ(予約済み)	$RY(n+\beta)C$	
$\alpha+13$		$RY(n+\beta)D$	
$\alpha+14$	OS定義	$RY(n+\beta)E$	
$\alpha+15$		$RY(n+\beta)F$	

出力データ(リモートデバイス→マスタ)				
I/Oポート番号 (注3)	信号名	デバイスNo. (注2)	使用状況	
$\gamma+0$	システム領域 リザーブ(機能別に設定)	$RX(n+\beta)0$	未使用	
:		:		
$\gamma+7$		$RX(n+\beta)7$		
$\gamma+8$	インシャルデータ処理要求フラグ	$RX(n+\beta)8$		
$\gamma+9$	インシャルデータ設定完了フラグ	$RX(n+\beta)9$		
$\gamma+10$	エラー状態フラグ	$RX(n+\beta)A$		使用 (注4)
$\gamma+11$	リモート局Ready	$RX(n+\beta)B$		
$\gamma+12$	リザーブ(予約済み)	$RX(n+\beta)C$		未使用
$\gamma+13$		$RX(n+\beta)D$		
$\gamma+14$	OS定義	$RX(n+\beta)E$		
$\gamma+15$		$RX(n+\beta)F$		

注1: α は各占有局数における最終I/O入力ポート番号に +1 した値です。

例えば4局占有の場合、入力システム領域は624~639までとなります。

注2: β は各占有局数における最終デバイスNoに +1 した値です。また、値は16進数表記です。

例えば4局占有の場合、入出力システム領域は次のようになります。

出力デバイス→ $RY(n+7)0\sim RY(n+7)F$, 入力デバイス→ $RX(n+7)0\sim RX(n+7)F$

注3: γ は各占有局数における最終I/O出力ポート番号に +1 した値です。

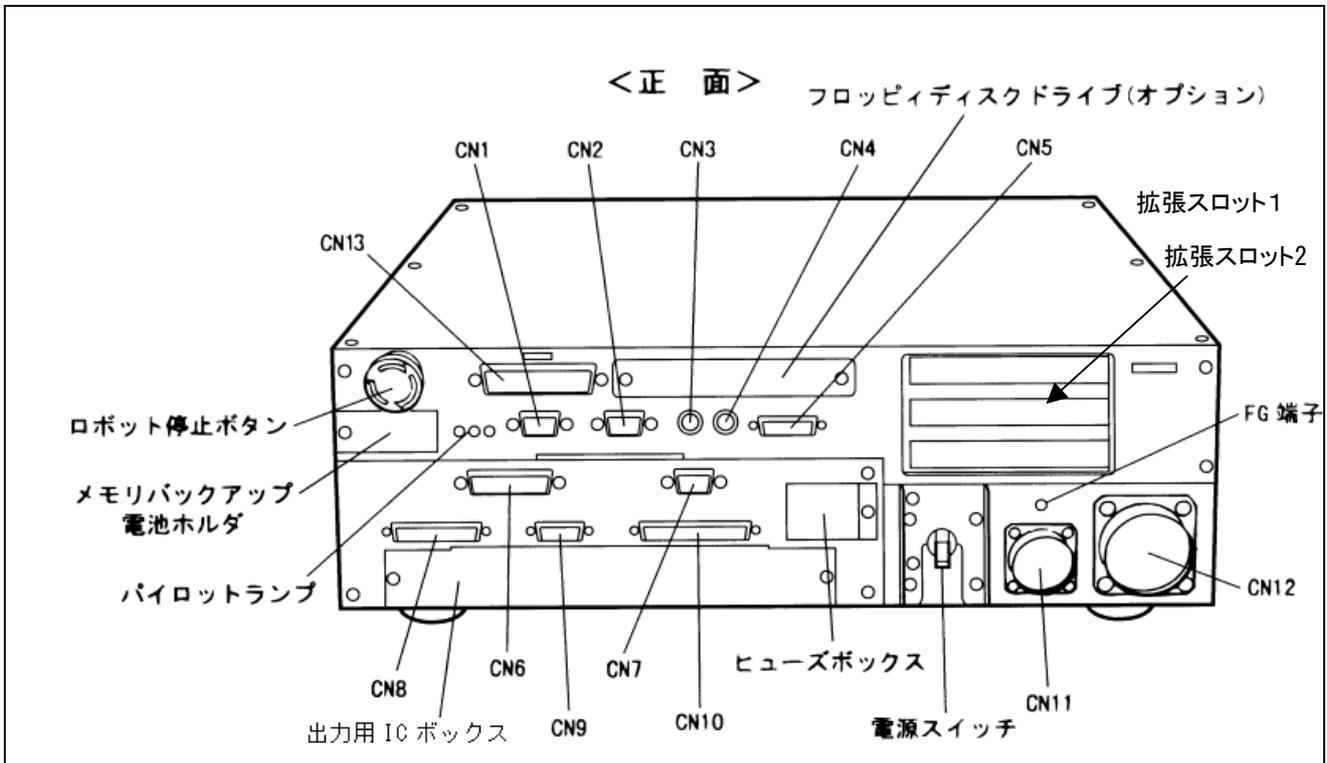
例えば4局占有の場合、出力システム領域は880~895までとなります。

注4: エラー状態フラグ、リモート局readyの作動条件を下記に示します。

	標準モード割付け	互換モード割付け
エラー状態フラグ	ロボット異常(I/Oポート番号770、デバイスNo RXn2)およびロボット警告(I/Oポート番号776、デバイスNo RXn8)と同じ。	ロボット異常(I/Oポート番号770、デバイスNo RXn2)およびロボット警告(I/Oポート番号782、デバイスNo RXnE)と同じ。
リモート局Ready	ロボット初期化完了(I/Oポート番号772、デバイスNo RXn4)と同じ。	ロボット電源入り完了(I/Oポート番号776、デバイスNo RXn8)と同じ。

9. RS232C 増設ボード（推奨オプション）の設定

RS232C増設ボードを増設することにより、ロボットコントローラはRS232C回線を3回線（標準1回線+増設2回線）使用することが可能になります。RS232C増設ボードはロボットコントローラの拡張スロット1または2に内蔵します。



9.1. RS232C 増設ボード（推奨品）

RS232C増設ボードは下表のものをお客様にてご準備をお願いします。

ボード名	COM-2(PC)F
メーカー名	CONTEC

注：増設ボードをコントローラに搭載する場合、コントローラ側にも対応処置が必要です。ご注文時に、「RS232C増設ボード対応コントローラ」と事前にご指定ください。

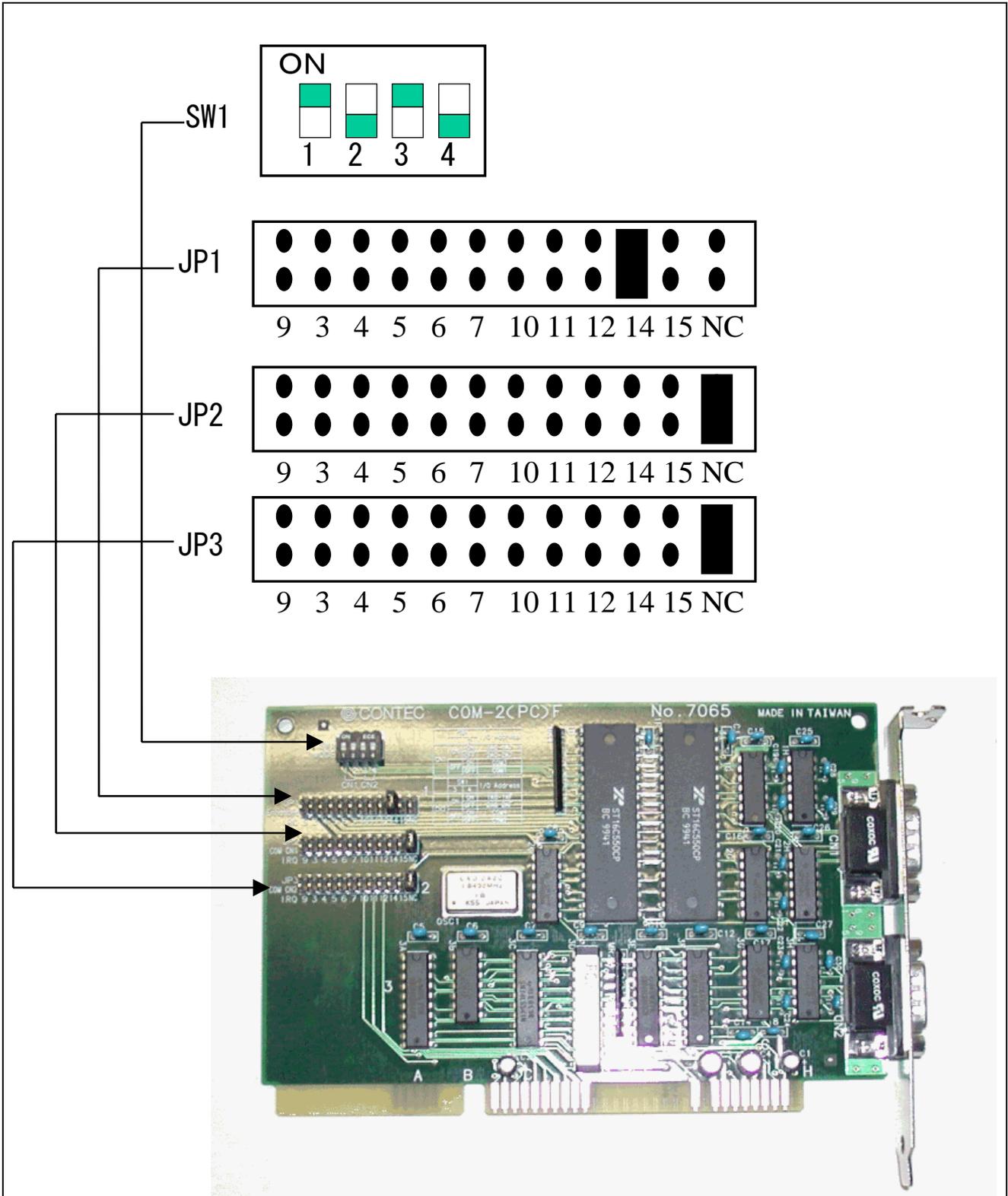
9.2. 増設ボードの取り付け

RC5コントローラへの増設ボードの取り付け方法は、「オプション機器説明書」の「増設ボードの取り付け」を参照してください。

9.3. 増設ボードのジャンパ・DIP スイッチ設定

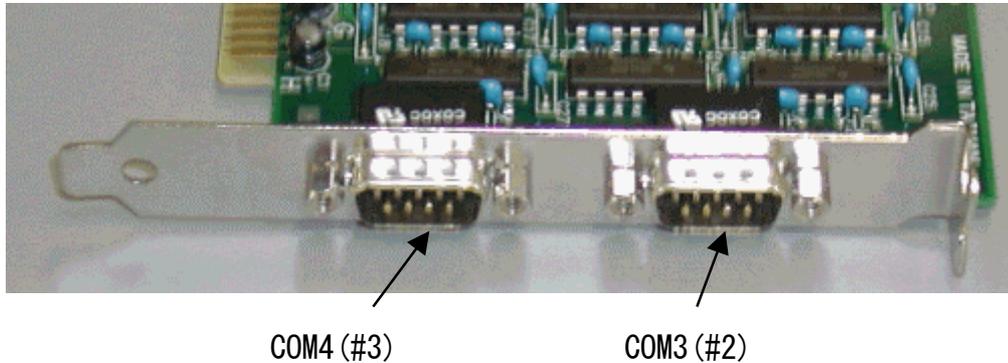
増設ボードのジャンパ設定とDIPスイッチ設定を下図のように行ってください。

設定箇所	SW1	JP1	JP2	JP3
設定	1、3をON	14にジャンパ・ピン装着	NCにジャンパ・ピン装着	NCにジャンパ・ピン装着



9.4. RS232C 増設ボードのコネクタと回線番号

RS232C増設ボードには下図のようにCOM3とCOM4のコネクタがあり、回線番号はCOM3が#2、COM4が#3になります。



9.5. RS232C 増設ボードの通信設定

RS232C増設ボードのCOM3, COM4の通信設定はティーチングペンダントから以下のように行なってください。

■通信権の設定について

操作経路：[F6 設定] - [F5 通信設定] - [F1 通信権]

注意：WINCAPS II との通信機能はCOM3, COM4ではサポートできませんので、通信権の設定では、両ポートとも「使用不可」のままにしてください。



■RS232C の設定

操作経路：[F6 設定] - [F5 通信設定] - [F2 シリアル設定]

COM3, COM4を下の画面で選択し、[F5 値変更] を押し、通信速度・パリティ等の設定を行ないます。



注意：RS232C 増設ボードでは使用できる通信速度は 38,400bps までとなり、デフォルトの通信速度は 19,200bps です。通信速度を 38,400bps に設定すると、通信エラーが発生しやすくなります。また通信速度 19,200bps においてもノイズ等の影響により通信エラーが発生する場合がありますので、次の用例のように「com_state」命令を使用してリトライ処理等を行なうようにしてください。

9.6. 用例（リトライ処理）

```
'!TITLE "<タイトル>"
PROGRAM sample
.
.
.
DEFPOS lp1(10)      'P型ローカル変数
DEFINT li1          'I型ローカル変数
.
.
.
li1 = 0             'li1 初期化
.
.
.
WHILE li1 < 10     '前判定反復を行います
.
.
.
INPUT #2, lp1(li1) '回線番号2より lp1(li1)にデータを
                   '取得する
```

```
com_state #2, I280      ' I280 に通信状態を取得する
IF I280 < 0 THEN        ' 通信エラーの場合-1が入っている
PRINT #2, "R"          ' リトライ指示を出力する
ELSE
PRINT #2, "A"          ' 正常受信を出力する
li1 = li1 + 1
END IF
.
.
.
.
WEND                    ' 10 回繰り返す
End
```

用例の説明

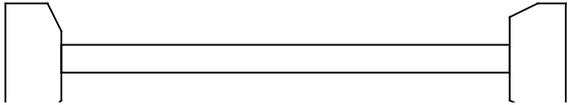
“R”は外部通信機器側のリトライ指示コマンドおよび“A”は正常受信コマンドになっているものとします。

9.7. 保証の範囲

当社は、このRS232C増設ボードを使用するための通信機能のみをコントローラから提供するものです。増設ボード自体のお問合せや製品についての保証はできませんのであらかじめ、ご承知おきをお願いします。

10. ペンダント延長ケーブル（オプション）の新設

オプション品として、ティーチングペンダント、ミニペンダント、オペレーティングパネルの延長ケーブルを設定しました。

品名	品番	備考
ペンダント延長ケーブル（4m）	410141-2390	
ペンダント延長ケーブル（4m）	410141-2400	

■ 取り扱い上の注意

1. 延長ケーブルを接続したケーブル全体の合計長は 12m 以下としてください。
2. 延長ケーブルは 2 本以上接続しないでください。
3. 延長ケーブルの中継接続コネクタには、ストレスや衝撃が加わらない処置をほどこしてください。

11. WINCAPS II の機能追加

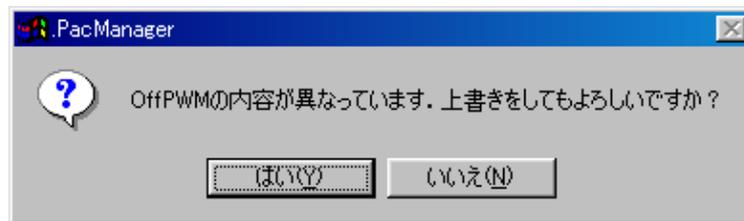
Ver. 1.9のWINCAPS IIに以下の機能を追加しました。

11.1. プログラムバンクに更新機能を追加

WINCAPSIIをバージョンアップした時は、旧バージョンでカスタマイズした内容を、新しいプログラムバンクにも反映させる必要があります。このプログラムバンクの更新を容易に行なえるように、WINCAPS IIに [自動更新] と [手動更新] の更新機能を追加しました。

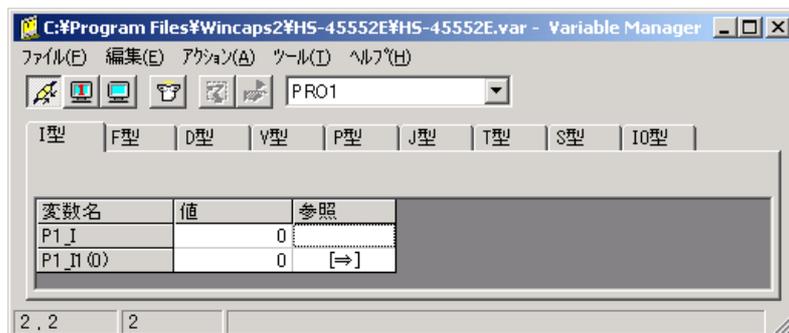
更新は、 [自動更新] または [手動更新] で行ない、どちらの方法でも現在使用中のプログラムバンクに対して、次の処理を実行します。

- ・旧バージョンで追加した、クラス/ライブラリを無条件に取込みます。
- ・同じ名前で内容の違うものは、どちらを使用するか選択します。



11.2. ローカル変数のモニタおよび変更機能の追加

コントローラと接続中は、変数スコープを切替えることによって、プログラム単位でローカル変数をモニタおよび変更することができます。



11.3. コントローラの機能拡張機能の追加

コントローラで使用するオプション機能の登録または削除をWINCAPS II からも行なえるようにしました。



12. コマンドの追加 [Ver. 1.9 以降]

従来、サーボ関係のPACライブラリとして使用していたものをコマンド（ステートメント）として追加し、使いやすさの向上をはかりました。

ST_aspACLD (ステートメント) [Ver. 1.9 以降]

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) ですべて指定します。
書式	ST_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>
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。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 Φ5H7 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「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 X₆、負荷重心位置 Y₆、負荷重心位置 Z₆ 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>
関連項目	プログラミングマニュアル「4.7 「使用条件」における最適可搬質量設定機能」
用例	<pre>ST_aspACLD 8500, -50, 100, 80</pre> <p>’ 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。</p>
注意事項	<ol style="list-style-type: none"> (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。 <p style="text-align: center;">$0.5 \times \text{外部負荷条件値} \leq \text{内部負荷条件値} \leq \text{外部負荷条件値}$</p>

ST_aspChange (ステートメント) [Ver.1.9 以降]

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 ST_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。

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

用例

ST_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。

ST_SetGravity (ステートメント) [Ver.1.9 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	ST_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。						
関連事項	ST_SetCurLmt, ST_ResetGravity, ST_SetGrvOffset						
注意事項	<p>(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。</p> <p>(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、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。</p> <p>(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(ST_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。</p> <p>(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。</p>						
用例	<table> <tr> <td>ST_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>ST_SetCurLmt 2, 30</td> <td>: 2 軸の電流制限値を 30%に設定する。</td> </tr> </table>	ST_SetGravity	: 重力補償を有効。	Delay 100	: 重力補償が反映されるのを待つ。	ST_SetCurLmt 2, 30	: 2 軸の電流制限値を 30%に設定する。
ST_SetGravity	: 重力補償を有効。						
Delay 100	: 重力補償が反映されるのを待つ。						
ST_SetCurLmt 2, 30	: 2 軸の電流制限値を 30%に設定する。						

ST_ResetGravity (ステートメント) [Ver.1.9 以降]

機能 重力バランスを無効にします。

書式 ST_ResetGravity

説明 重力バランスを無効にします。

関連事項 ST_ResetCurLmt, ST_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

ST_ResetGravity

ST_SetGrvOffset (ステートメント) [Ver.1.9 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 ST_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。重力補償ステートメント(ST_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。

関連事項 ST_SetCurLmt, ST_SetGravity, ST_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値が 0 にリセットされます。

用例

ST_SetGrvOffset

ST_ResetGrvOffset (ステートメント) [Ver.1.9 以降]

機能	重力補償値の補正を無効にします。
書式	ST_ResetGrvOffset
説明	重力補償値の補正を無効にします。
関連事項	ST_SetGrvOffset
注意事項	<p>(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。</p> <p>(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。</p>
用例	ST_ResetGrvOffset

ST_SetCurLmt (ステートメント) [Ver.1.9 以降]

機能	指定した軸のモータ電流値を制限します。
書式	ST_SetCurLmt <軸番号>、<設定値>
説明	<p><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。</p> <p><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。</p> <p>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。</p>
関連事項	ST_ResetCurLmt, ST_SetGravity, ST_SetGrvOffset, ST_SetEralw
注意事項	<p>(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。</p> <p>(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。</p> <p>(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。</p> <p>(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。</p>

(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、ST_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、プログラミングマニュアル「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 (ST_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 (ST_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) エアバランスの無い 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

<u>6 軸</u>	ST_SetGravity	’ 重力補償を有効にする。
	ST_SetGrvOffset	’ 重力補償値を補正する。
	ST_SetEralw 2, 20	’ 2 軸の偏差許容値を 20 度に設定する。
	ST_SetCurLmt 2, 30	’ 2 軸の電流制限値を 30% に設定する。
<u>4 軸</u>	ST_SetEralw 2, 20	’ 2 軸の偏差許容値を 20 度に設定する。
	ST_SetCurLmt 2, 30	’ 2 軸の電流制限値を 30% に設定する。
<u>4 軸 HS-E</u>	ST_SetZBalance	’ Z 軸重力補償値を設定する。
	ST_SetEralw 3, 100	’ Z 軸の偏差許容値を 100mm に設定する。
	ST_SetEralw 4, 30	’ T 軸の偏差許容値を 30 度に設定する。
	ST_SetCurLmt 3, 10	’ Z 軸の電流制限値を 10% に設定する。
	ST_SetCurLmt 4, 10	’ T 軸の電流制限値を 10% に設定する。

ST_ResetCurLmt (ステートメント) [Ver.1.9 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 ST_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、ST_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。

関連項目 ST_SetCurLmt, ST_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。

```
PR0999
TAKEARM
ST_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

```
ST_ResetCurLmt 0      ’ 全軸の電流制限を解除
ST_ResetGrvOffset     ’ 重力補償補正値をリセット
```


- 注意事項**
- (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。
[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。
 - (2) 電流制限解除命令 (ST_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。

用例

ST_ResetEralw 0 ' 全軸の偏差許容値をデフォルト値に戻す。

ST_OnSrvLock (ステートメント) [Ver.1.9 以降]

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 ST_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。

関連事項 ST_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

ST_OnSrvLock 1 ' 1 軸のサーボロック
ST_OnSrvLock 0 ' 全軸のサーボロック

ST_SetCompControl (ステートメント) [Ver.1.9 以降]

機能	力制限機能を有効にします。(6軸専用命令)
書式	ST_SetCompControl
説明	力制限機能を有効にします。ST_SetFrcLimit, ST_SetCompRate, ST_SetFrcCoord 等で設定された力制限条件を有効にします。
関連事項	ST_SetFrcLimit, ST_SetCompRate, ST_SetFrcCoord, ST_ResetCompControl ST_SetCompFControl

- 注意事項**
- (1) 重力補償無効時、電流制限有効時に本ステートメントを実行した場合、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、ST_ResetCurLmt、重力補償有効化は、ST_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。パス動作中に本ステートメントを実行した場合は、エンド動作となります。また、ロボット動作中に本ステートメントを実行し、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、ST_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ステートメント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、ST_SetCompFControl を使用してください。
 - (7) ST_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 ST_ResetCompControl にて力制限を無効にした後、再度 ST_SetCompControl にて力制限を有効にしてください。

用例

ST_SetFrcCoord 1	' 力制限座標系を設定する。
ST_SetFrcLimit 100, 0, 100, 100, 100, 100	' 力制限割合を設定する。
ST_SetCompControl	' 力制限機能を有効にする。
ST_SetEralw 1, 90	' 偏差許容値を設定する。

ST_SetCompFControl (ステートメント) [Ver.1.9 以降]

機能	力制限機能を有効にします。(6軸専用命令)
書式	ST_SetCompFControl
説明	ST_SetCompControl と同様、力制限機能を有効にします。ただし、ST_SetCompControl にて実行される重力補償補正処理は実行されません。
関連事項	ST_SetCompControl
注意事項	<p>(1) 重力補償無効時、電流制限有効時に本ステートメントを実行した場合、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。</p> <p>(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。</p> <p>(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。</p> <p>(4) ロボット停止状態で実行してください。パス動作中に本ステートメントを実行した場合は、エンド動作となります。また、ロボット動作中に本ステートメントを実行し、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。</p> <p>(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、ST_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。</p>

用例

ST_SetGrvOffset	' 重力補償補正值を計算する。
ST_SetFrcCoord 1	' 力制限座標系を設定する。
ST_SetFrcLimit 100, 0, 100, 100, 100, 100	' 力制限割合を設定する。
STSetCompFControl	' 力制限を有効化する。

ST_ResetCompControl (ステートメント) [Ver.1.9 以降]

機能 力制限機能を無効にします。(6軸専用命令)

書式 ST_ResetCompControl

説明 力制限機能を無効にします。

関連事項 ST_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) ロボット停止状態で実行してください。パス動作中に本ステートメントを実行した場合は、エンド動作となります。また、ロボット動作中に本ステートメントを実行し、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) 本ステートメント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、ST_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、ST_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、ST_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、ST_ResetCompControl 実行前に ST_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

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

ST_SetFrcCoord (ステートメント) [Ver.1.9 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 ST_SetFrcCoord <設定値>

説明 ST_SetFrcLimit, ST_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。

関連事項 ST_SetFrcLimit, ST_SetCompRate, ST_SetFrcCoord, ST_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。力制限中に本ステートメントを実行すると、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(ST_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(ST_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

ST_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。
ST_SetFrcLimit 100, 0, 100, 100, 100, 100	' 力制限割合を設定する。
ST_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。

ST_SetFrcLimit (ステートメント) [Ver.1.9 以降]

- 機能** 力制限割合を設定します。(6軸専用命令)
- 書式** ST_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** ST_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。
- 関連事項** ST_ResetFrcLimit, ST_SetFrcCoord, ST_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。力制限中に本ステートメントを実行すると、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

```
ST_SetFrcCoord 1          ' 力制限座標系をツール座標に設定する。
ST_SetFrcLimit 100, 0, 100, 100, 100, 100
                        ' 力制限割合を設定する。
ST_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。
```

ST_ResetFrcLimit (ステートメント) [Ver.1.9 以降]

機能 力制限割合を初期化します。(6軸専用命令)

書式 ST_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。

関連事項 ST_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。力制限中に本ステートメントを実行すると、エラー「60fa 力制限中です」が発生します。

用例

ST_ResetCompControl	'力制限を無効にします。
ST_ResetFrcLimit	'力制限割合を初期化します。

ST_SetCompRate (ステートメント) [Ver.1.9 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	ST_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>ST_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>
関連事項	ST_ResetCompRate, ST_SetFrcCoord, ST_SetCompControl
注意事項	<p>(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。</p> <p>(2) 力制限中は、実行できません。力制限中に本ステートメントを実行すると、エラー「60fa 力制限中です」が発生します。</p> <p>(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。</p>
用例	<pre>ST_SetFrcCoord(1) ' 力制限座標系をツール座標に設定する。 ST_SetCompRate 100, 0, 100, 100, 100, 100 ' 柔らかさ割合を設定する。 ST_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>

ST_ResetCompRate (ステートメント) [Ver.1.9 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	ST_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。						
関連事項	ST_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。力制限中に本ステートメントを実行すると、エラー「60fa 力制限中です」が発生します。						
用例	<table border="0" style="width: 100%;"> <tr> <td style="width: 40%;">ST_ResetCompControl</td> <td style="width: 10%;">’</td> <td>力機能を無効にする。</td> </tr> <tr> <td>ST_ResetCompRate</td> <td>’</td> <td>柔らかさ割合を初期化する。</td> </tr> </table>	ST_ResetCompControl	’	力機能を無効にする。	ST_ResetCompRate	’	柔らかさ割合を初期化する。
ST_ResetCompControl	’	力機能を無効にする。					
ST_ResetCompRate	’	柔らかさ割合を初期化する。					

ST_SetFrcAssist (ステートメント) [Ver.1.9 以降]

機能	力制限時のオフセット力を設定します。(力制限特殊機能ステートメント)(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。
関連事項	ST_ResetFrcAssist, ST_SetFrcCoord, ST_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

ST_SetFrcCoord 1	'力制限座標系をツール座標に設定する。
ST_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。
ST_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。

ST_ResetFrcAssist (ステートメント) [Ver.1.9 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

ST_ResetFrcAssist

説明

ST_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。

関連事項

ST_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

ST_ResetCompControl	'力制限を無効にする。
ST_ResetFrcAssist	'オフセット力、モーメントを 0 に設定する。

ST_SetCompJLimit (ステートメント) [Ver.1.9 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

ST_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。ST_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。

関連事項

ST_ResetCompJLimit, ST_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) 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) ST_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 ST_SetCompRate, ST_SetDumpRate にて調整してください。

用例

```

ST_SetFrcCoord 1          ' 力制限座標系をツール座標に設定する。
ST_SetCompJLimit 30, 0, 0, 0, 0
                          ' J1 の電流制限値を 30%に設定する。
ST_SetFrcLimit 100, 0, 100, 100, 100, 100
                          ' 力制限割合を設定する。
ST_SetCompControl        ' 力機能を有効にする。
    
```

ST_ResetCompJLimit (ステートメント) [Ver.1.9 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

ST_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。

関連事項

ST_SetCompJLimit, ST_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

```

ST_ResetCompControl      ' 力機能を無効にする。
ST_ResetCompJLimit       ' 電流制限値を初期化する。
    
```

ST_SetCompVMode (ステートメント) [Ver.1.9 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	ST_SetCompVMode
説明	ST_SetCompControl 実行時の力制限速度制御モードを有効にします。
関連事項	ST_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

ST_SetCompVMode	' 力制限速度制御モードを有効にする。
ST_SetCompControl	' 力制限を有効にする。

ST_ResetCompVMode (ステートメント) [Ver.1.9 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	ST_ResetCompVMode
説明	ST_SetCompControl 実行時の力制限速度制御モードを無効にします。
関連事項	ST_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

ST_ResetCompVMode	' 力制限速度制御モードを無効にする。
ST_SetCompControl	' 力制限を有効にする。

ST_ResetCompEralw (ステートメント) [Ver.1.9 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	ST_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。
関連事項	ST_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	ST_ResetCompEralw

ST_SetDampRate (ステートメント) [Ver.1.9 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	ST_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	ST_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。
関連事項	ST_ResetDampRate, ST_SetFrcCoord, ST_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。力制限中に本ステートメントを実行すると、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) ST_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) ST_SetDampRate 実行後、ST_SetCompRate を設定すると、粘性割合は、ST_SetCompRate の設定値となります。

用例

ST_SetFrcCoord 1 '力制限座標系をツール座標に設定する。
ST_SetCompRate 100, 0, 100, 100, 100, 100 '柔らかさ割合を設定する。
ST_SetDampRate 100, 20, 100, 100, 100, 100 '粘性割合を設定する。
ST_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。

ST_ResetDampRate (ステートメント) [Ver.1.9 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 ST_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。

関連事項 ST_SetDampRate

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。力制限中に本ステートメントを実行すると、エラー「60fa 力制限中です」が発生します。

用例

ST_ResetDampRate '粘性割合を初期化する。

ST_SetZBalance (ステートメント) [Ver. 1.9 以降]

機能	Z 軸、T 軸の重力補償値を設定します。(4 軸専用)
書式	ST_SetZBalance
説明	エアバランスのない 4 軸ロボットの Z 軸・T 軸は、重力により、下向きの静荷重（重力トルク）を受けます。電流制限設定時、制限値を重力トルク以下に設定すると Z 軸が落下、T 軸が回転します。(4 軸専用) 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、重力補償値を設定します。本機能を使用することにより、電流制限値を小さく設定した場合の Z 軸の落下を、T 軸の回転を防止します。
関連事項	ST_ResetZBalance, ST_SetCurLmt
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) 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) 重力補償値設定を実行した後、ワーク重量が変化した場合は、設定値にずれが生じます。その場合は再度本命令を実行してください。

用例

ST_SetZBalance

ST_ResetZBalance (ステートメント) [Ver. 1.9 以降]

機能	重力補償値の補正を無効にします。(4 軸専用)
書式	ST_ResetZBalance
説明	4 軸ロボットの Z 軸、T 軸の重力補償値の設定を無効にします。
関連事項	ST_SetZBalance
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

ST_ResetZBalance

13. エラーコード表の追加・修正

[関連頁：エラーコード表]

Ver. 1.9から追加・修正されたエラーコードを下記に示します。

コード	メッセージ	Level	説明	復帰処置
120A	スレーブ排他フラグ異常	4	DeviceNetスレーブまたは、CC-Linkリモートデバイス通信処理部の排他フラグが正常に動作していません。	DeviceNetスレーブボードまたは、CC-Linkリモートデバイスボードの接続状態を確認してください。復帰しない場合、ボードが故障している可能性があります。
120C	ネットワークエラーが解除されました	3	ネットワーク (DeviceNet、PROFIBUS、CC-Link) のエラーが解除されました。	ティーチングペンダント、オペレーティングペンダント、ミニペンダント、または外部よりエラークリア操作を行なってください。
1230	ロボット側DPRAMリトライ異常(スレーブ)	4	ロボット側からDeviceNetスレーブモジュール内または、CC-Linkリモートデバイスボード内のDPRAMへアクセスできない状態です。	コントローラのパワースイッチを一度切ってから再操作を行なって下さい。
1238	通信側DPRAMリトライ異常(スレーブ)	4	DeviceNetスレーブ通信部ソフトまたは、CC-Linkリモートデバイス通信部ソフトがDPRAMへアクセスできなくなりました。	コントローラのパワースイッチを一度切ってから再操作を行なって下さい。
128A	CC-Link電源立ち上げ時のハード異常	4	電源立ち上げ時にCC-Linkボードのハード異常を検出しました。	コントローラのパワースイッチを一度切ってから再操作を行なって下さい。
128B	CC-Link ← コントローラ ハンドシェイク異常	4	コントローラの内部状態情報をCC-Linkボードとやり取りする上でエラーが発生しました。	コントローラのパワースイッチを一度切ってから再操作を行なって下さい。
128C	CC-LinkDPRAMデータ異常	4	CC-LinkボードのDPRAMのデータに異常が発生しました。	コントローラのパワースイッチを一度切ってから再操作を行なって下さい。

ソフトウェア更新 (Ver. 1.9)

コード	メッセージ	Level	説明	復帰処置
128D	CC-Link通信エラー	4	CC-Link通信できない状態になりました。	1. マスタの設定とコントローラのCC-Linkの設定内容とで食い違いが無い点検してください。 2. CC-Link通信ケーブルの断線または、CC-Linkコネクタが外れていないか点検してください。 3. ロボット本体及びコントローラの近くにノイズ発生源となる設備(溶接機等)がない事を点検して下さい。 上記内容と、CC-Linkボード正面の各LEDの状態もあわせて参照してください。
128E	CC-Link局番設定異常	2	局番+占有局数-1が1～63の範囲を超えています。	局番+占有局数-1を1～63の範囲で設定してください。
128F	CC-Link通信速度設定異常	2	通信速度設定スイッチの設定値が0～4の範囲を超えています。	通信速度設定スイッチの設定値を0～4の範囲で設定してください。
129A	CC-Link占有局数設定異常	4	占有局数が2～4局の範囲を超えています。	占有局数を2～4局の範囲内で設定してください。
21B9	ローカル変数編集中	2	ローカル変数編集中に出来ない操作を実行しました。	ローカル変数編集後、再度実行してください。
2498	禁止エリア0侵入 (DIRECTモード中)	2	ダイレクトモード中にツール先端が禁止領域に侵入しました。	禁止エリア0外に移動させてください。
2499	禁止エリア1侵入 (DIRECTモード中)	2	↑	禁止エリア1外に移動させてください。
249A	禁止エリア2侵入 (DIRECTモード中)	2	↑	禁止エリア2外に移動させてください。
249B	禁止エリア3侵入 (DIRECTモード中)	2	↑	禁止エリア3外に移動させてください。
249C	禁止エリア4侵入 (DIRECTモード中)	2	↑	禁止エリア4外に移動させてください。
249D	禁止エリア5侵入 (DIRECTモード中)	2	↑	禁止エリア5外に移動させてください。
249E	禁止エリア6侵入 (DIRECTモード中)	2	↑	禁止エリア6外に移動させてください。
249F	禁止エリア7侵入 (DIRECTモード中)	2	↑	禁止エリア7外に移動させてください。
27BC	引数が設定されていません	2	引数設定画面が開かない状態で引数付きプログラムを起動しようとした。	再度起動操作をし、引数設定画面が現れるのを確認後、起動してください。
27BD	コントローラ起動時の初期化異常です	5	コントローラ立ち上げの初期化に失敗しました	コントローラのパワースwitchを1度切って、再立ち上げして下さい
27BE	現在位置では動作できません	2	簡単教示動作時において、現在のロボットの状態では動作不可能なコマンドを実行しようとした。	簡単教示において円弧動作及びDEPART動作を行なう場合は、直前の動作命令の目標位置へ移動後、行なってください。

ソフトウェア更新 (Ver. 1.9)

コード	メッセージ	Level	説明	復帰処置
2AF4	冷却ファン異常	2	コントローラ内の冷却ファンの停止を検出しました。	エラークリア後、ロボットを安全な位置へ移動させて下さい。冷却ファンの交換が必要です。
60DA	指定したZ軸位置まで動作しません	1	現在のアーチ開始、完了距離の設定では指定したZ軸位置に到達しません。	アーチ開始、完了距離の設定を変更して下さい。
64FA	ベルト切断を検出しました	3	U軸アームのベルト切断を検出しました。	カバーをはずし、ベルト切れの有無を確認して下さい。
6819	不正コマンド実行異常(ホスト1)	5	コントローラ内部異常 (ソフトウェア異常)が発生しました。	コントローラのパワースイッチを一度切ってから再操作を行なって下さい。
681A	不正コマンド実行異常(ホスト2)	5	↑	↑
681B	不正コマンド実行異常(ホスト3)	5	↑	↑
681C	不正コマンド実行異常(サーボ1)	5	↑	↑
681D	不正コマンド実行異常(サーボ2)	5	↑	↑
681E	不正コマンド実行異常(サーボ3)	5	↑	↑
6829	高軌跡制御処理遅延	4	高軌跡制御処理は間に合いませんでした	高軌跡制御時に、RS232C、イーサネットによる通信頻度が高い場合やコントローラをキーボード操作した場合、本エラーが発生する場合があります。通信頻度を下げてください。
682b	高軌跡制御有効、無効操作異常です	4	高軌跡制御有効、無効設定中にプログラムリセット、ステップバック操作をしました。エラー発生によりプログラムリセットされました。	高軌跡制御切り替え操作中の瞬時停止時、プログラムリセット、ステップバック操作する場合は、一度モータOFFしてください。
736F	引数に配列のあるプログラムは起動できません	2	引数に配列のあるプログラムの単独起動はサポートしていません。	引数に配列のあるプログラムの単独起動はできません。CALLでの呼び出しのみサポートしています。CALLで呼び出すようにして下さい。
75B0	クライアントポートオープン失敗	3	指定クライアントポートが使用中か設定に誤りがあります。	指定ポートを変更するかクライアントポートの設定を変更してください。

取扱説明書 追補版

Ver.1.9 の新機能

初 版 2002 年 1 月
第 2 版 2002 年 3 月

株式会社デンソーウェーブ FA 事業部

3D100C

- 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。

