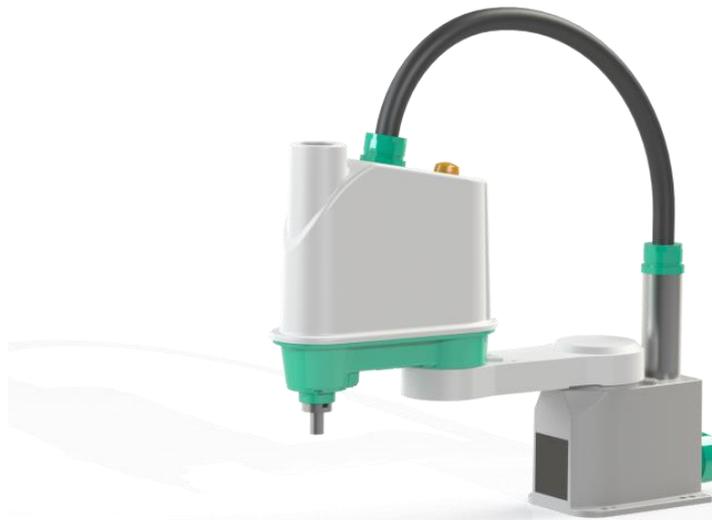


スカラロボット KSLシリーズ (追加機能)

取扱説明書

SM-A20058



- 製品をご使用になる前に、本取扱説明書を必ずお読みください。
- 特に安全に関する記述は、注意深くお読みください。
- 本取扱説明書は必要なときにすぐ取出して読めるように、大切に保管してください。

はじめに

このたびは、当社のスカラロボット「KHE・KHL シリーズ」をお買求めいただきまして、誠にありがとうございます。
本取扱説明書は本製品の性能を十分に発揮させるために、取付、使用方法などの基本的な事項を記載した
ものです。よくお読みいただき、正しくご使用ください。
なお、本取扱説明書は紛失しないように、大切に保管してください。

本取扱説明書に記載の仕様、外観は、将来予告なく変更することがあります。

注意：

- この取扱説明書は産業用ロボットを実際にご使用になられる方のお手元に必ず届くよう
お取りはからいください。
- 産業用ロボットをご使用前にこの取扱説明書を必ずご覧くださいますようお願いいたします。
- お読みになった後は必ず保管してくださいますようお願いいたします。

本書では、産業用ロボットコントローラKSL3000の追加機能について説明しています。

第 1 章

NCBOY付加軸機能について説明しています。

第 2 章

パルス出力機能について説明しています。

第 3 章

エリア出力機能について説明しています。

第 4 章

押付力[N]単位指定機能について説明しています。

第 5 章

軌跡精度向上機能について説明しています。

第 6 章

加速度自動調整機能について説明しています。

第 7 章

ビジョンキャリブレーション機能について説明しています。

なお、本書の内容は、予告なく変更される場合がありますので、予めご了承ください。
安全編、操作編、ロボット言語編、インターフェース編、据付・輸送編、保守編、通信編、
ユーザーパラメータ編、アラーム編の各種取扱説明書が用意されていますので、必要により
そちらの取扱説明書と併用してご利用くださいますようお願いいたします。

目 次

はじめに	i
第 1 章 NCBOY 付加軸機能	5
第 2 章 パルス出力機能	6
第 3 章 エリア出力機能	7
3.1 エリア出力機能の概要	7
3.2 パラメータ	7
3.3 使用例	10
3.3.1 複数テーブルで同一のDOU T 番号を設定した場合	12
3.4 応用例	14
3.4.1 エリア出力機能を使用した 2 台のロボットの干渉回避	14
3.4.2 設定方法	15
第 4 章 押付力[N]単位指定機能	18
4.1 押付力[N]単位指定機能の概要	18
4.2 言語仕様	18
FORCE3LIM	19
4.3 ティーチングペンダントのトルク表示の変更	20
第 5 章 軌跡精度向上機能（開発中）	21
5.1 軌跡精度向上機能の概要	21
5.2 言語仕様	21
LOCUS	22
第 6 章 加速度自動調整機能	23
6.1 加速度自動調整機能の概要	23
6.2 言語仕様	23
SPURT	24
6.3 使用上の注意事項	25
第 7 章 ビジョンキャリブレーション機能	26
7.1 ビジョンキャリブレーション機能の概要	26
7.2 キャリブレーションシート	27
7.3 パラメータ	29
7.4 言語仕様	35

VSCALBIN	36
VSCALBOUT	37
7.5 サンプルプログラム	38

第1章

NCBOY付加軸機能

本機能は現在、垂直多関節ロボットシステムでのみ使用可能
(スカラロボットシステムは今後拡張予定です)

第2章

パルス出力機能

KSL3000コントローラでは使用できません

第3章

エリア出力機能

本章では、エリア出力機能について説明します。

3.1 エリア出力機能の概要

エリア出力機能とは、ロボットが指定された領域に侵入したときに、指定したDOU T信号をONまたはOFFする機能です。指定する領域とDOU T信号の組み合わせを64通り設定することができます。

本機能のコントローラシステムの対応バージョンは以下になります。

メイン部：X8GCAS-13E以上(スカラ)

3.2 パラメータ

エリア出力機能用の設定パラメータはパラメータファイル「AREA.PAR」に記述します。パラメータのフォーマットは以下の通りです。

```
[A00] AREA OUTPUT SETTING PARAMETER
{Area output flag (0:Disable 1:Enable)}
{P1:+Direction position coordinate [mm]}
{P2:-Direction position coordinate [mm]}
{Dout number}
{Table No. 1}
= 0
= 9999.9 9999.9 9999.9 0.0 0.0 0.0 0.0 0.0
= -9999.9 -9999.9 -9999.9 0.0 0.0 0.0 0.0 0.0
= 1
{Table No. 2}
= 0
= 9999.9 9999.9 9999.9 0.0 0.0 0.0 0.0 0.0
= -9999.9 -9999.9 -9999.9 0.0 0.0 0.0 0.0 0.0
= 2
{Table No. 3}
= 0
= 9999.9 9999.9 9999.9 0.0 0.0 0.0 0.0 0.0
= -9999.9 -9999.9 -9999.9 0.0 0.0 0.0 0.0 0.0
= 3
.
.
.
{Table No. 64}
= 0
= 9999.9 9999.9 9999.9 0.0 0.0 0.0 0.0 0.0
= -9999.9 -9999.9 -9999.9 0.0 0.0 0.0 0.0 0.0
= 64
```

} ①エリア出力機能有効無効設定
} ②直方体領域 P1, P2
} ③出力 DOUT 番号及び出力論理

① エリア出力機能有効無効設定

エリア出力機能を有効にするか無効にするかを設定します。

0：機能無効

1：機能有効

データ形式 : 整数型

データ範囲 : 0または1

値の例 : 0

② 直方体領域P1、P2

信号を出力する領域を設定します。設定した直方体領域内にロボットが侵入した場合に、指定したDOU T信号をONまたはOFFします。

※(+位置)と(-位置)の値を入れ替えても動作に支障はありません。

設定 : = (+位置x座標), (+位置y座標), (+位置z座標), (予備), (予備), (予備),
(予備), (予備)

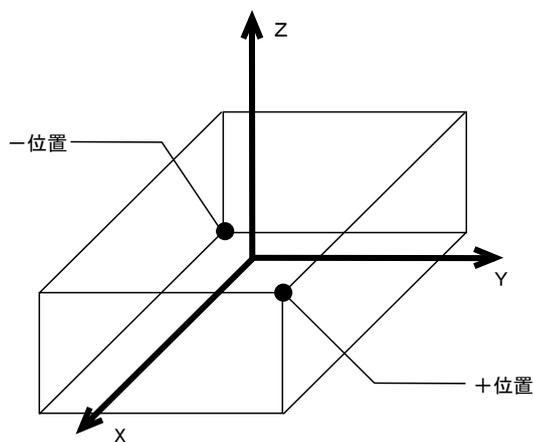
= (-位置x座標), (-位置y座標), (-位置z座標), (予備), (予備), (予備),
(予備), (予備)

データ形式 : 実数型

データ単位 : mm

データ範囲 : -9999.9～9999.9

値の例 : 1.0



③ 出力DOUT番号及び出力論理

信号を出力するDOUT番号を指定します。

DOUT番号を+に設定した場合、設定領域に侵入したとき指定DOUTがONになります。

また、DOUT番号を-に設定した場合、設定領域に侵入したとき指定DOUTがOFFになります。

データ形式 : 整数型

データ範囲 : 1～64、-1～-64、101～164、-101～-164、301～364、-301～-364、401～464、
-401～-464

値の例 : 1

注意

- ・複数テーブルで同一のDOUT番号を指定した場合、出力信号は各テーブルのDOUT出力の論理和になります。詳細は3.3.1 複数テーブルで同一のDOUT番号を設定した場合で説明します。
- ・本機能にて割り当てたDOUTは、SCOL言語やティーチングペンダントを用いた操作で信号のON/OFFはできませんので注意してください。
- ・本機能はロボットプログラムの実行の有無にかかわらず、有効となります。ロボットプログラムが停止していてもロボットが指定された領域に入っていれば、指定のDOUT信号をONまたはOFFします。
- ・ロボットコントローラの電源を切りますと、出力信号も切れますので注意してください。

3.3 使用例

例としてAREA. PARを以下のように設定した場合について説明します。

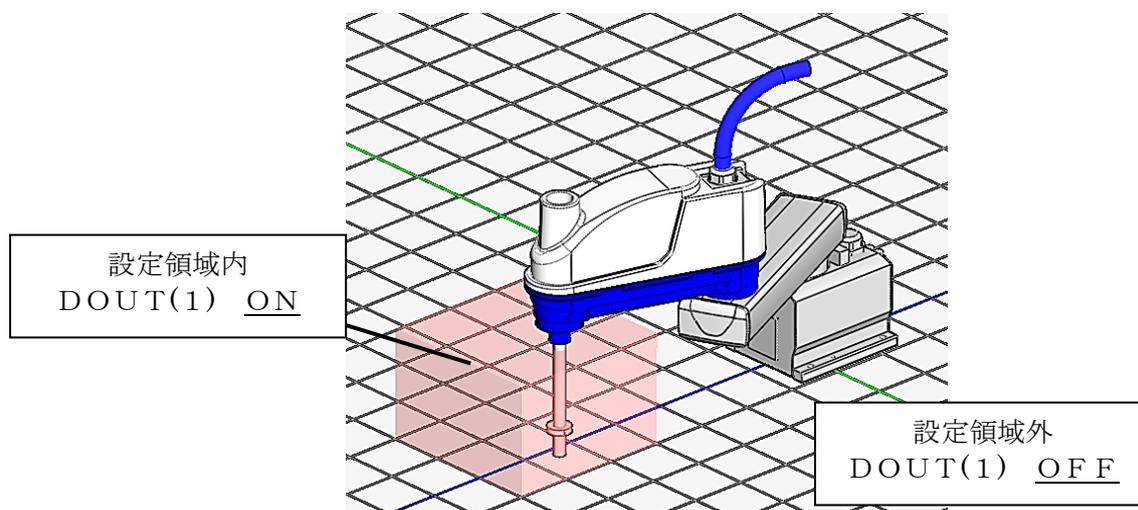
```

[A00] AREA OUTPUT SETTING PARAMETER
{Area output flag (0:Disable 1:Enable)}
{P1:+Direction position coordinate [mm]}
{P2:-Direction position coordinate [mm]}
{Dout number}
{Table No. 1}
= 1
= 700.0 100.0 200.0 0.0 0.0 0.0 0.0 0.0
= 400.0 -200.0 0.0 0.0 0.0 0.0 0.0 0.0
= 1

{Table No. 2}
= 1
= 650.0 250.0 200.0 0.0 0.0 0.0 0.0 0.0
= 350.0 -50.0 0.0 0.0 0.0 0.0 0.0 0.0
= -2
.
.
.

```

Table No. 1で設定した領域は下図のようになります。

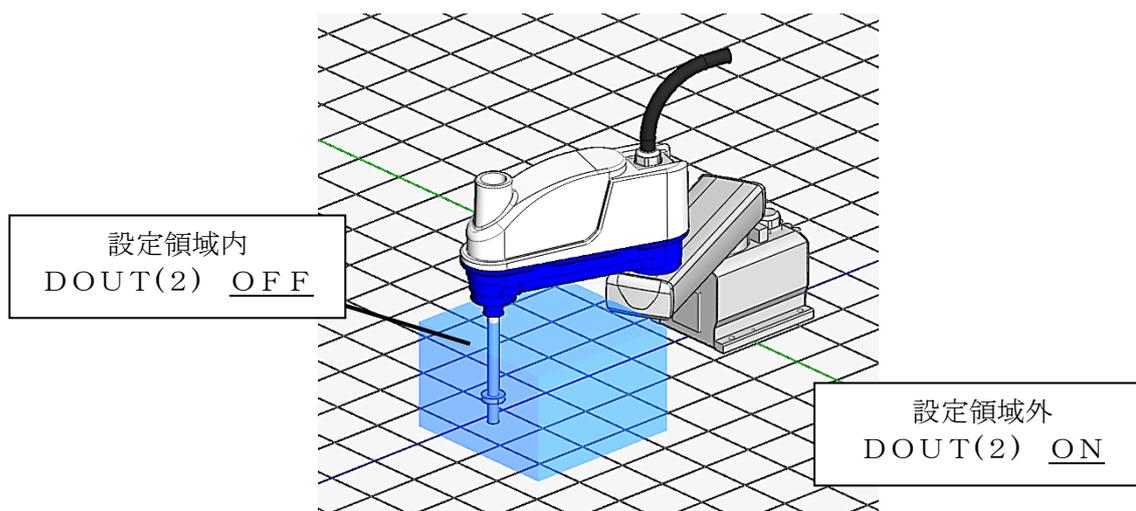


ロボット現在位置が設定した領域内にあるとき、DOUT (1) がON、領域外にあるときはDOUT (1) がOFFになります。

次に、Table No.2を設定した場合について示します。

```
[A00] AREA OUTPUT SETTING PARAMETER
{Area output flag (0:Disable 1:Enable)}
{P1:+Direction position coordinate [mm]}
{P2:-Direction position coordinate [mm]}
{Dout number}
{Table No.1}
= 0
= 700.0 100.0 200.0 0.0 0.0 0.0 0.0 0.0
= 400.0 -200.0 0.0 0.0 0.0 0.0 0.0 0.0
= 1

{Table No.2}
= 1
= 650.0 250.0 200.0 0.0 0.0 0.0 0.0 0.0
= 350.0 -50.0 0.0 0.0 0.0 0.0 0.0 0.0
= -2
.
.
.
```



ロボット現在位置が設定した領域内にあるとき、DOUT (2) がOFF、領域外にあるときはDOUT (2) がONになります。

3.3.1 複数テーブルで同一のDOU T番号を設定した場合

本エリア出力機能において、複数のテーブルで同一のDOU T番号を指定した場合、出力信号は各テーブルのDOU T出力の論理和になります。すなわち、出力ON命令とOFF命令が重なった場合、ON命令が優先されます。

例としてAREA. PARを以下のように設定した場合について説明します。

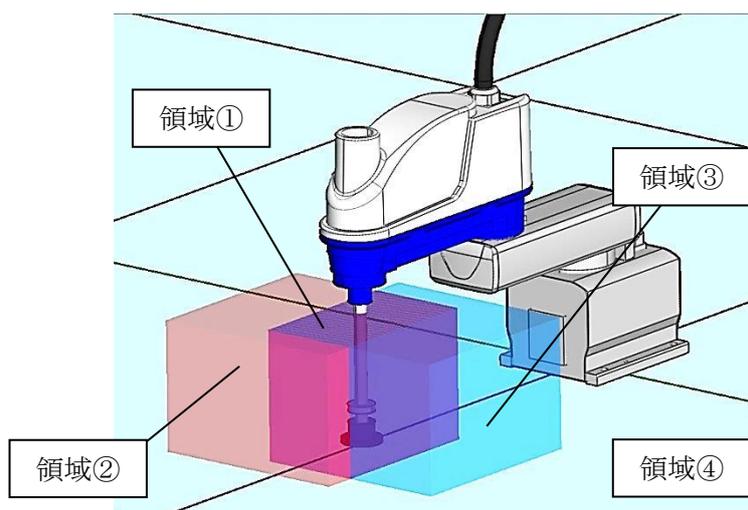
```

[A00] AREA OUTPUT SETTING PARAMETER
{Area output flag (0:Disable 1:Enable)}
{P1:+Direction position coordinate [mm]}
{P2:-Direction position coordinate [mm]}
{Dout number}
{Table No. 1}
= 1
= 700.0 100.0 200.0 0.0 0.0 0.0 0.0 0.0
= 400.0 -200.0 0.0 0.0 0.0 0.0 0.0 0.0
= 1

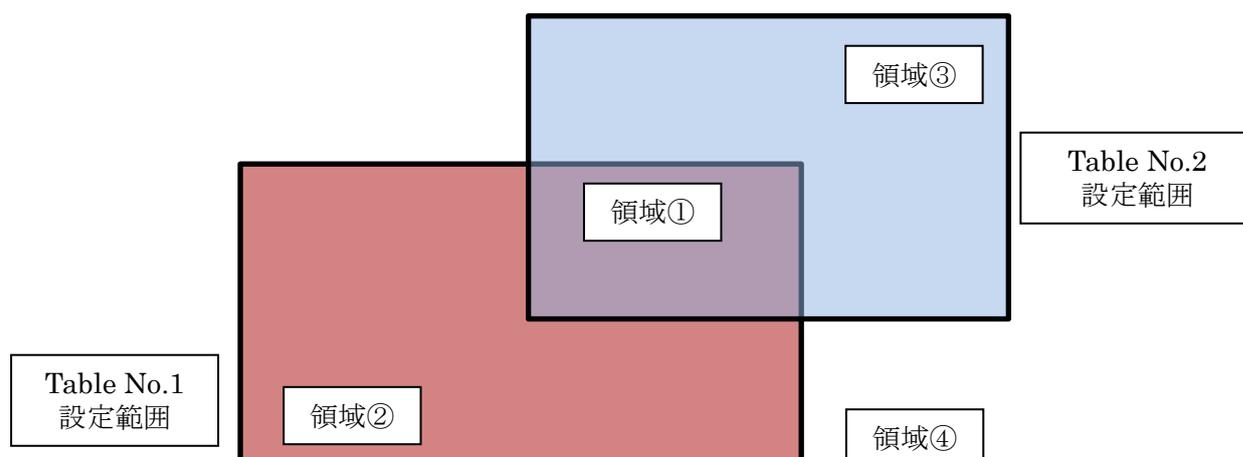
{Table No. 2}
= 1
= 650.0 250.0 200.0 0.0 0.0 0.0 0.0 0.0
= 350.0 -50.0 0.0 0.0 0.0 0.0 0.0 0.0
= -1
.
.
.

```

設定した領域は下図のようになります。



また、各テーブルの設定値の組み合わせによって、以下の4つの領域に分けられます。



各領域におけるDOUT (1) のON/OFFは以下のようになります。

		Table No.2	
		OFF	ON
Table No.1	ON	領域① DOUT (1) : <u>ON</u>	領域② DOUT (1) : <u>ON</u>
	OFF	領域③ DOUT (1) : <u>OFF</u>	領域④ DOUT (1) : <u>ON</u>

3.4 応用例

3.4.1 エリア出力機能を使用した2台のロボットの干渉回避

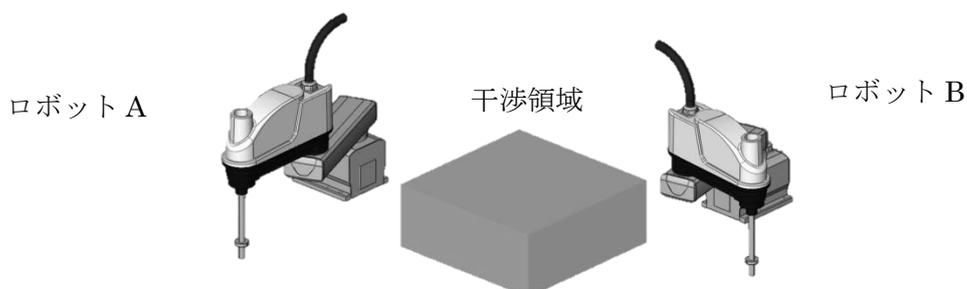
本節では、エリア出力機能の応用例として、外部信号によるFEEDHOLD機能と組み合わせた2台のロボットの干渉回避について説明します。

エリア出力機能と外部信号によるFEEDHOLD機能の両方に対応しているコントローラシステムのバージョンは以下になります。

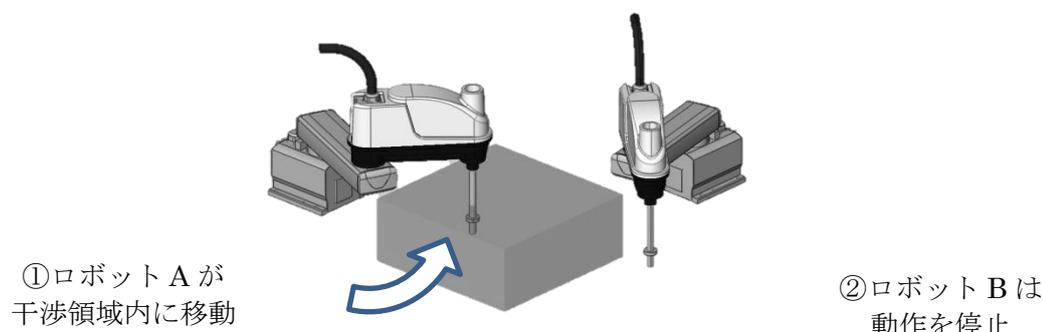
メイン部：X8GCAS-13E以上(スカラ)

本節では、以下のような2台のロボット(ロボットA、ロボットB)の干渉回避方法(下図参照)について説明します。

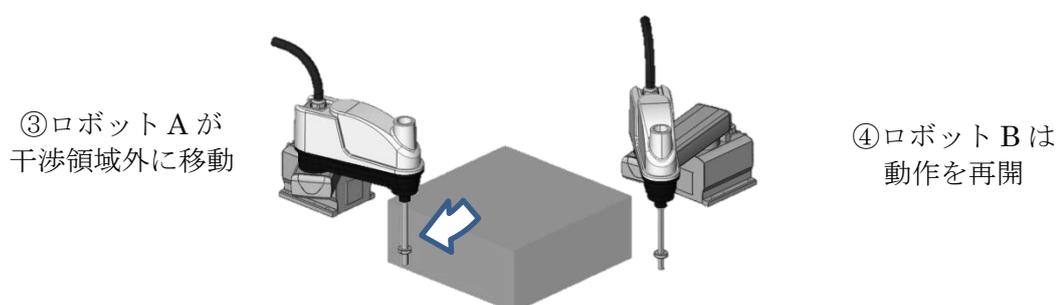
I 動作空間の一部(干渉領域)を共有するロボットA,Bが別々の動作を行っている



II ロボットAの先端が干渉領域内に移動した際にもうロボットBは動作を停止する



III ロボットAが領域内での作業を終了し、干渉領域外に移動したら、ロボットBは動作を再開する



3.4.2 設定方法

本節で説明する2台のロボットの干渉回避を行うためには以下の設定を行う必要があります。

- ・エリア出力機能の設定

ロボットAから見た干渉領域の座標をロボットAのAREA.PARに、

ロボットBから見た干渉領域の座標をロボットBのAREA.PARにそれぞれ設定してください。

今回の例ではどちらのロボットも干渉領域に入ったら、DOUT(1)がOFFするように設定した場合について説明します。

※ロボットごとにエリア出力機能で指定する領域の原点が異なりますので注意してください。

- ・「外部信号によるFEEDHOLD機能」の設定

2台のロボットそれぞれのUSER.PARの[U50]を以下のように設定し、外部信号によるFEED HOLD機能を有効にしてください。USER.PARの変更方法については、取扱説明書“ユーザーパラメータ編”を参照してください。なお、USER.PARの変更を反映させるにはコントローラの電源をOFF/ONする必要があります。

```
[U50] SIGNAL FEED HOLD FUNCTION
{0:Disabled 1:Enabled}
= 1
```

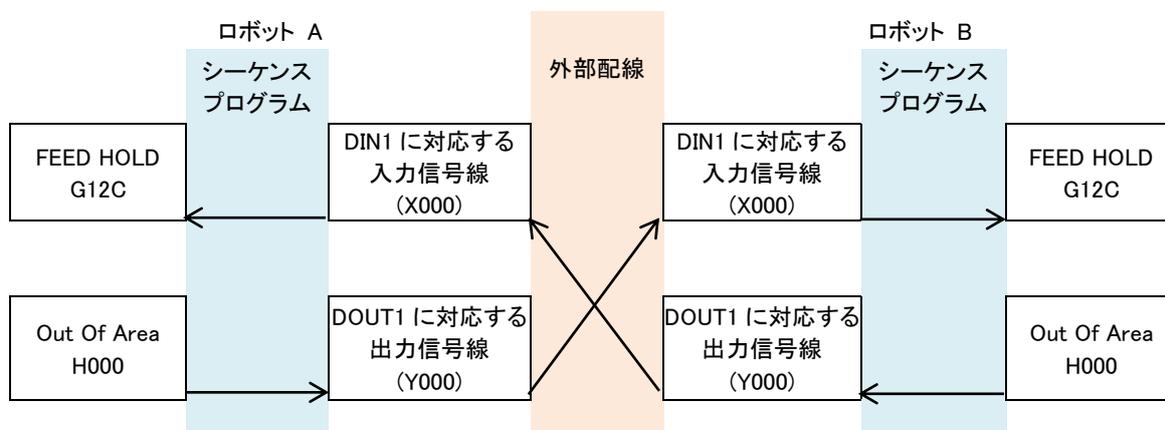
本機能を有効にした場合、コントローラに内蔵した簡易PLCのリレー“G12C”がOFFするとロボットは一時停止し、ONすると動作を再開します。

※ロボットは“G12C”がOFFしてから減速して停止するため、実際に停止するまである程度の距離を移動します。

- 2台のロボットコントローラの配線

2台のロボットそれぞれの入出力信号配線に以下の変更を加えて下さい。

配線のイメージ図を以下に示します。



- 外部配線

ロボットA側のDOUT1に対応する出力信号線をロボットB側のDIN1に対応する入力信号線に、ロボットB側のDOUT1に対応する出力信号線をロボットA側のDIN1に対応する入力信号線に接続してください。DIN1、DOUT1にそれぞれ対応する入力信号線、出力信号線のピン番号はコントローラの型式によって異なりますので、お使いのコントローラに合った取扱説明書“SM-A20054 インターフェース編”を参照してください。

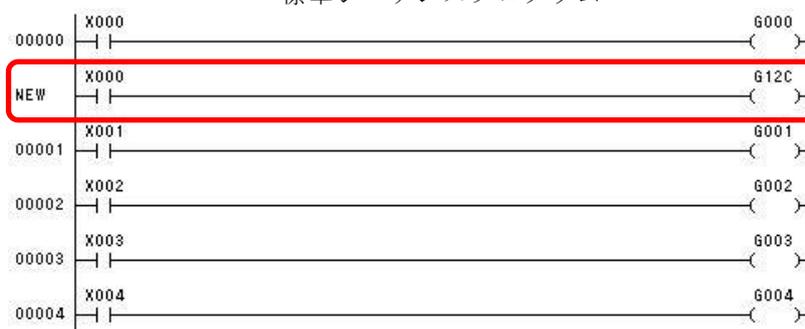
・シーケンスプログラム

ロボット A, B それぞれのシーケンスプログラムにおいて、
 干渉領域に入った時にOFFするDOUT(1)信号(H000)をOUTPUT-25pin (Y100) に、
 INPUT-23pin (X000)を自ロボットのFEEDHOLD指令(G12C)に接続してください。
 シーケンスプログラムの例を以下に示します。

シーケンスプログラムの変更方法については取扱説明書“SM-A20055 簡易PLC機能編”を参照してください。



標準シーケンスプログラム



変更後のシーケンスプログラム

<注意>

- ・動作確認の際には、安全のため2台のロボットが物理的に接触しない状態でロボットの動作を確認してください。また、ロボットの動作速度は低速から徐々に上げるようにしてください。
- ・本節で示した設定を行った状態で、2台のロボットが同時に干渉領域に入った場合、2台のロボットは共に停止します。ティーチングペンダントでの手動誘導等により、復帰は可能ですが、片方のロボットが干渉領域外に出たタイミングで、もう片方のロボットが動作を再開し、思わぬ動作をすることがあります。そのため、復帰操作の際には片方のロボットの非常停止スイッチを押すなど、安全に十分注意して復帰操作を行ってください。

第4章

押付力[N]単位指定機能

本章では、押付力[N]単位指定機能について説明します。

4.1 押付力[N]単位指定機能の概要

押付力[N]単位指定機能とは、スカラロボットでZ軸方向に押付ける作業をする際、通常[%]単位で押付力を指定するところを[N]単位で指定することができる機能になります。

本機能の対応機種とコントローラシステムの対応バージョンは以下になります。

機種：KHE-400 / KSL3000

KHL-600 / KSL3000

メイン部：X8GCAS-14B以上

4.2 言語仕様

押付力[N]単位指定機能では、FORCE3LIMという命令語を使用します。

言語の詳細は以下になります。

FORCE3LIM

機能

第3軸のトルク制限値をNmで指定します。

フォーマット

FORCE3LIM=<式>

文例

```
FORCE3LIM=50
MOVE A1 WITH FORCE3LIM=70
```

解説・注意

スカルロボットにおける第3軸のトルク制限値をNmで指定します。

<式>には、定数、変数、及び演算式を使用する事ができます。

トルクの制限値として、0以下を指定すると0と見なします。

トルクの制限値が低すぎると、ロボットが動作できずにエラーになる事があります
出力可能なトルク値より大きな値を指定すると、出力可能な最大値が設定されます。

たとえば、第3軸の出力トルクが50%=75Nmの時

TORQUE={300, 300, 50, 300, 300}と

FORCE3LIM=75は同じ効果となります。

内部では、FORCE3LIMとTORQUE命令の第3軸の設定箇所は同一のものです。

よってFORCE3LIM=75トルクを制限した後、

TORQUE={300, 300, 300, 300, 300}として制限値をTORQUE命令で変化させることも可能です。

サンプル
プログラム

```
PROGRAM TORQUESMPL
MOVE A1
FORCE3LIM=75
MOVE A2
OPEN1
DELAY 0.5
MOVE A1
FORCE3LIM=999
READY
END
```

A2への動作時にZ軸（この場合第3軸）
のトルクを、定格トルクの75Nmに
制限します。

トルク制限を最大値にします。
(トルク制限無効)

4.3 ティーチングペンダントのトルク表示の変更

ティーチングペンダントの「UTILITY」 → 「MOTOR」の3軸のトルク表示を[%]から[N]に変更することが可能です。

USER.PARの[U51]で3軸のトルク表示を従来通りパーセント[%]で表示か、ニュートン[N]で表示か設定します。

```
[U51] 3-AXIS TORQUE DISPLAY (SCARA)
{0:[%] 1:[N] (Add Payload) 2:[N] (Not Add Payload)}
= 0
```

デフォルトは0になります。

0：従来通りパーセント[%]で表示されます。

1：ニュートン[N]で表示されます。(PAYLOADの質量分を含む)

2：ニュートン[N]で表示されます。(PAYLOADの質量分を含まない)

第5章

軌跡精度向上機能（開発中）

本章では、軌跡精度向上機能について説明します。

5.1 軌跡精度向上機能の概要

軌跡精度向上機能とは、ロボットプログラムにてSCOL言語のLOCUS命令を実行すると、サーボ制御部にて速度フィードフォワード制御が有効になり、軌跡追従精度が向上する機能になります。

本機能の対応機種とコントローラシステムの対応バージョンは以下になります。

機種：

メイン部：X8GCAS-13C以上(スカラ)

サーボ部：X8YCB-11C以上(KSL3000)

※現在、機種ごとの設定パラメータを検証中です。使用できません。

5.2 言語仕様

軌跡精度向上機能では、LOCUSという命令語を使用します。

言語の詳細は以下になります。

LOCUS

機能	ロボットの軌跡精度を指定します。
フォーマット	LOCUS=<式>
文例	LOCUS=FINE LOCUS=0
解説・注意	<p>ロボットの軌跡追従精度を指定するシステム変数です。</p> <p>軌跡追従精度は0で粗、1で精となります。</p> <p>軌跡追従精度の精はMOVES、MOVEC動作時のみ有効になります。</p> <p>ロボットプログラムにてLOCUS命令で軌跡追従精度の精を実行するとサーボ制御部にて速度フィードフォワード制御が有効になり、軌跡追従精度が向上します。</p> <p>軌跡追従精度の指定には、システム定数のFINE、COARSEも使用できます。</p> <p>軌跡追従精度は、LOCUS=FINEで精に、LOCUS=COARSEで粗に設定できます。</p> <p><式>には、定数、変数、及び演算式を使用する事ができます。ただし、ベクトル型のデータは使用できません。</p> <p>本命令は式の中で使用します。</p> <p>軌跡追従精度として、0以下、1以上を指定すると、それぞれ0、1が指定されたものとします。</p> <p>軌跡追従精度の初期値は、粗になっています。</p>
サンプルプログラム	<pre>PROGRAM LOCUSMPL LOCUS=FINE MOVE A1 MOVES A2 MOVES A3 MOVE A4 END</pre> <p>軌跡追従精度の精を指定します。</p> <p>MOVES動作であるA2, A3への移動時の軌跡追従精度が向上します。</p>

第6章

加速度自動調整機能

本章では、加速度自動調整機能について説明します。

6.1 加速度自動調整機能の概要

加速度自動調整機能とは、スカラロボットにおいて動作に応じてアームの加速度を変更することで、従来よりもサイクルタイムを短縮することができる機能です。

ロボットプログラムにてSCOL言語のSPURT命令を実行すると、機能が有効になります。

本機能の対応機種とコントローラシステムの対応バージョンは以下になります。

機種：スカラロボット

KHLシリーズ : KHL-500, KHL-600, KHL-700

KHEシリーズ : KHE-400

メイン部 : X8GCAS-14C以上(スカラ)

6.2 言語仕様

加速度自動調整機能では、SPURTという命令語を使用します。

言語の詳細は以下になります。

SPURT

機能	ロボットの加速度自動調整機能の有効・無効を指定します。
フォーマット	SPURT=<式>
文例	SPURT=ON
解説・注意	<p>ロボットの加速度自動調整機能の有効・無効指定するシステム変数です。SPURT=ONで有効に、SPURT=OFFで無効に設定できます。初期状態では、無効になっています。</p> <p>加速度自動調整機能はPTP動作（MOVE、MOVEA、MOVEI、MOVEJ、MOVEPLT）において有効になります。CP動作（MOVES、MOVEC）に対しては無効です。</p>
サンプル プログラム	<pre>PROGRAM SPURTSMPL SPURT=ON MOVE A1 MOVES A2 SPURT=OFF MOVE A3 END</pre> <p>加速度自動調整機能の有効を指定します。 PTP動作であるA1への移動時の動作加速度が向上します。</p>

6.3 使用上の注意事項

- ・本機能では動作により生じるモータの負荷を予測して加速度を決定しています。そのためアームを折り畳んだ状態で行う動作のような負荷の小さい動作においてサイクルタイムの短縮に効果的です。一方で、負荷が大きい動作においては機能無効の場合と比較してサイクルタイムが変化しない場合や若干遅くなる場合がありますことをご了承ください。
- ・本機能を有効にした場合はモータの負荷が全体的に向上するため、標準状態でのご使用に比べて、製品寿命が短くなる可能性があります。また、位置決め時の手先のオーバーシュートや振動が標準状態よりも大きくなる場合があります。本機能使用時の繰り返し位置決め精度やオーバーシュート量は保証の対象外となりますことをご了承ください。
- ・多くの動作において本機能を有効にしますとモータの負荷率エラー（2-4-1-1 Axis1 Motor overload(15m)等）が発生する場合があります。その際には適用する動作を限定してモータの負荷率が規定値以上にならないようにしてください。

第7章

ビジョンキャリブレーション機能

本章では、ビジョンキャリブレーション機能について説明します。

7.1 ビジョンキャリブレーション機能の概要

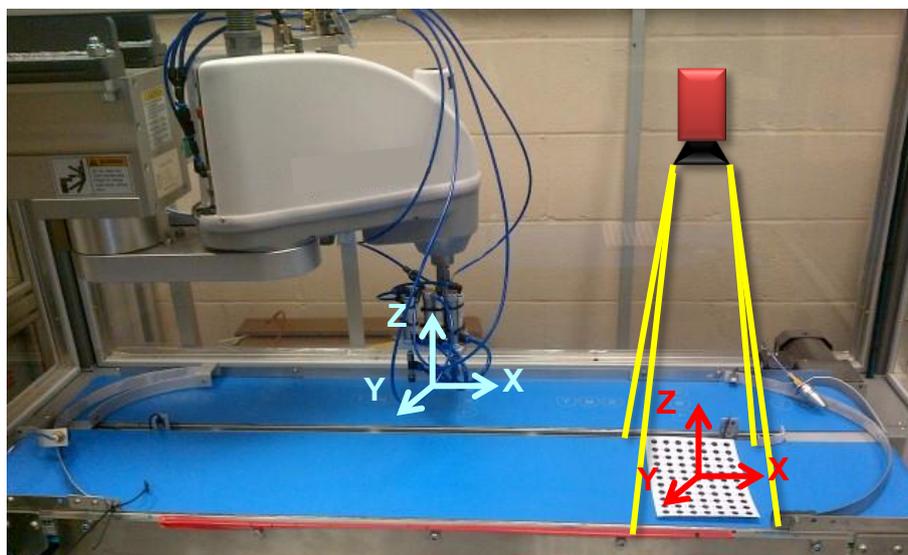
ロボットとビジョンセンサを組み合わせ、ワークのピッキング等の作業を行う場合、ロボットの座標系とビジョンセンサの座標系を合わせこむ必要があります。これをビジョンキャリブレーション作業と呼びます。

ビジョンキャリブレーション作業には、①ロボット側の位置データをビジョンセンサに登録する方法と、②ビジョンセンサ側の位置データをロボット内のパラメータに登録する方法があります。

①の場合は、ビジョンセンサ内でビジョンセンサの座標系からロボットの座標系への変換処理が行われ、ロボットの座標系に対応した位置データがビジョンセンサからロボットに送られてくるので、その位置データをそのまま使用することができます。設定方法などはビジョンセンサ側の取扱説明書に従ってください。

②はビジョンセンサ側でキャリブレーションを行うことができない場合など、ロボット側でキャリブレーションを行う場合に使用する方法になります。②の場合は、ビジョンセンサからビジョンセンサの座標系の位置データがロボットに送られてくるので、ロボット内でビジョンセンサの座標系からロボットの座標系への変換処理を行ない、位置データを使用します。

本章では、この②の方法について説明していきます。

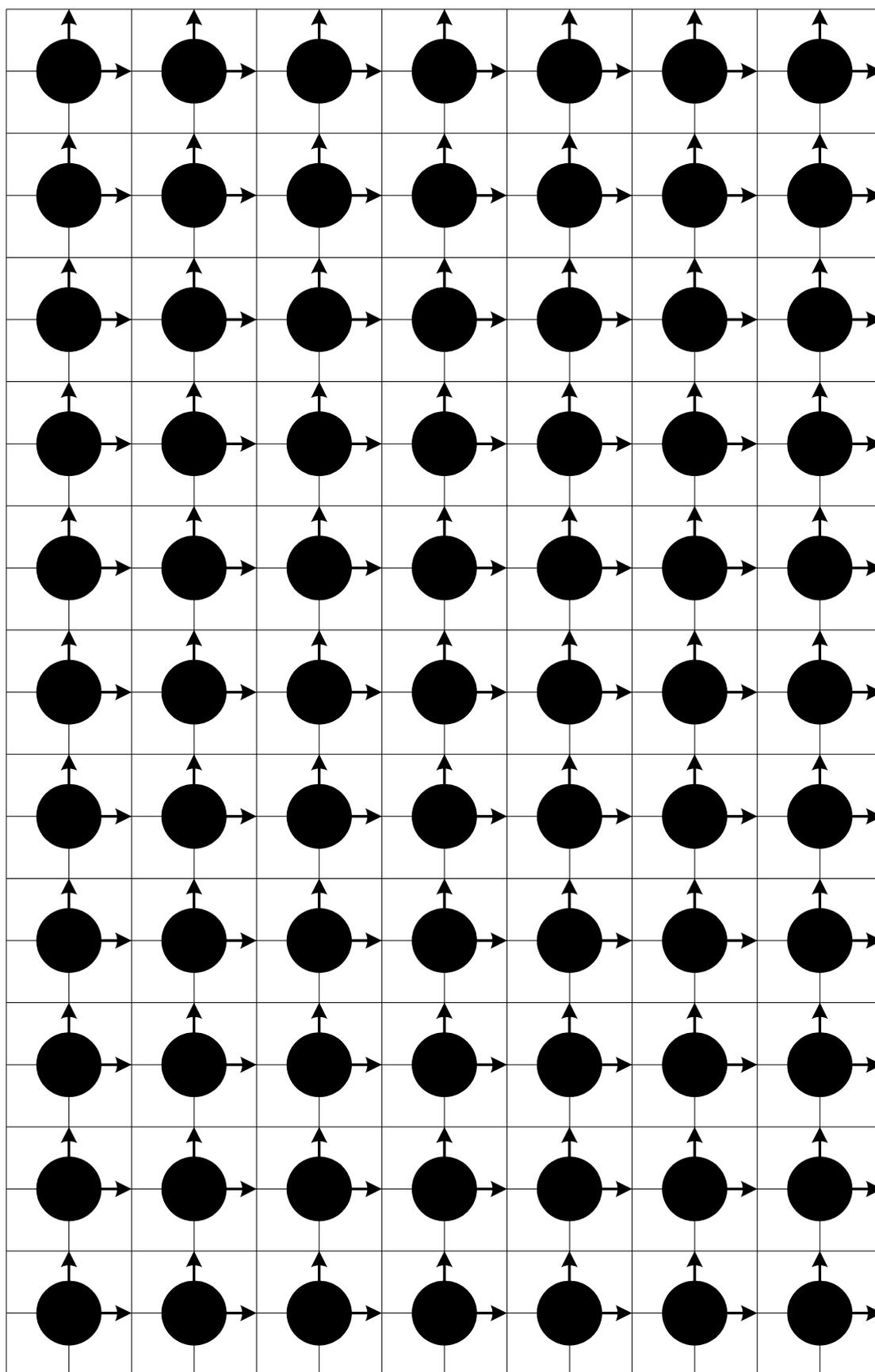


7.2 キャリブレーションシート

ビジョンキャリブレーションを行うにあたり、基準となるポイントを数点設定する作業が発生します。その際には、作業用ワークや下図のようなシートを使用します。

下図のシートをキャリブレーションシートと呼び、次ページに掲載してあります。任意のサイズに印刷してご使用ください。





キャリブレーションシート

7.3 パラメータ

ビジョンキャリブレーション機能を使用するには、「VISION.PAR」というパラメータを編集する必要があります。パラメータのフォーマットは以下の通りです。

```
[V00] COORDINATE TRANSFORMATION PARAMETER FOR 2D VISION CALIBRATION
{<V1.x>,<V1.y>,<V2.x>,<V2.y>,<V3.x>,<V3.y>:Vision's reference position (X,Y)}
{<R1.x>,<R1.y>,<R2.x>,<R2.y>,<R3.x>,<R3.y>:Robot's reference position (X,Y)}
{<Method>:Calibration method (0:3point)}
{<Coord>:Vision coordinate system (0:Righty/CCW, 1:Righty/CW, 2:Lefty/CCW, 3:Lefty/CW)}
{Table No. 1}
= 0.0 0.0 0.0 0.0 0.0 0.0 ... ①
= 0.0 0.0 0.0 0.0 0.0 0.0 ... ②
= 0.0 0.0 0.0 0.0 0.0 0.0 ... ③
= 0 ... ④
= 0 ... ⑤
.
.
.
{Table No. 20}
= 0.0 0.0 0.0 0.0 0.0 0.0
= 0.0 0.0 0.0 0.0 0.0 0.0
= 0
= 0
[V01] COORDINATE TRANSFORMATION MATRIX FOR MULTIPLE POINT CALIBRATION
{Table No. 21} ... ①
{Coordinate transformation matrix}
= 0.0 0.0 0.0 0.0
= 0.0 0.0 0.0 0.0
= 0.0 0.0 0.0 0.0 ... ②
= 0.0 0.0 0.0 0.0
{Vision coordinate system rotation angle (deg)}
= 0.0 ... ③
{Vision coordinate system rotation direction (0:CCW, 1:CW)}
= 0 ... ④
.
.
.
{Table No. 40}
{Coordinate transformation matrix}
= 0.0 0.0 0.0 0.0
= 0.0 0.0 0.0 0.0
= 0.0 0.0 0.0 0.0
= 0.0 0.0 0.0 0.0
{Vision coordinate system rotation angle (deg)}
= 0.0
{Vision coordinate system rotation direction (0:CCW, 1:CW)}
= 0
[END]
```

(1)

(2)

(1) [V00] ビジョンキャリブレーション用座標変換パラメータ

このパラメータは5行で1つのテーブルとなっており、{Table No.1}~{Table No.20}まで最大20組設定可能です。

本機能のコントローラシステムの対応バージョンは以下になります。

メイン部：X8GCAS-10E以上(スカラ)

① テーブル番号 {Table No.**}

1~20まで用意されており、ロボットプログラムでテーブルを切り替えながら使用します。

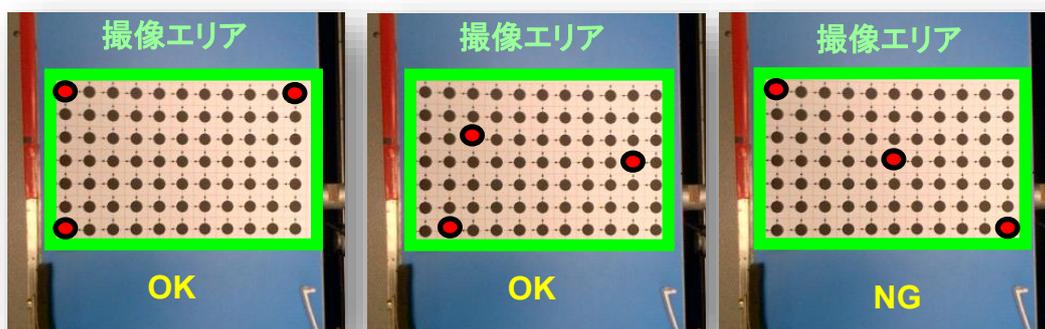
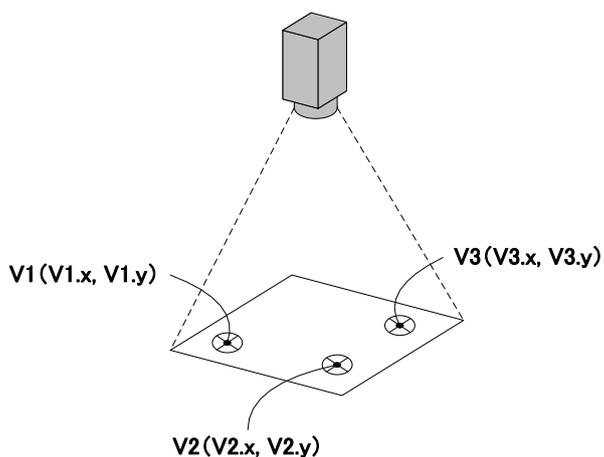
② ビジョンセンサの基準位置 (X,Y)

{<V1.x>,<V1.y>,<V2.x>,<V2.y>,<V3.x>,<V3.y>:Vision's reference position (X,Y)}

= 0.0 0.0 0.0 0.0 0.0 0.0

ビジョンセンサ座標系上で任意の3点を決定し、データの座標(x,y)を入力します。

3点が一直線上にならないように注意してください。



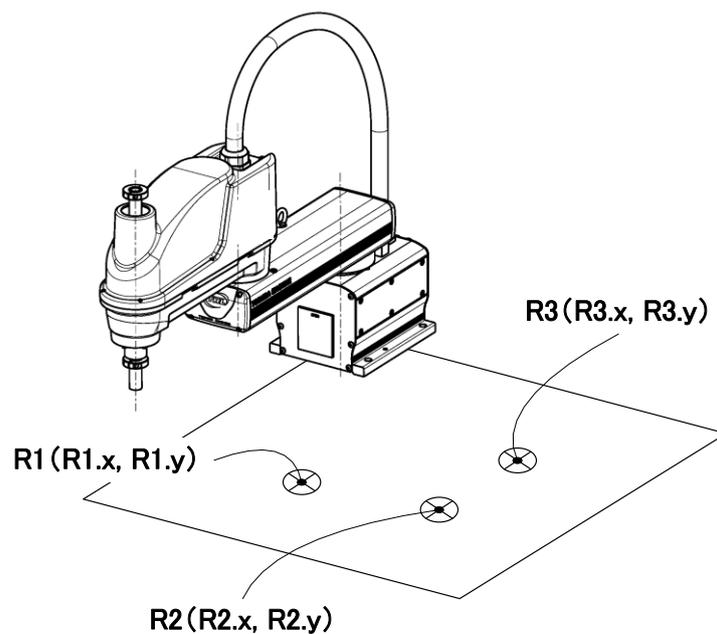
③ ロボットの基準位置(X,Y)

 $\{<R1.x>, <R1.y>, <R2.x>, <R2.y>, <R3.x>, <R3.y> : \text{Robot's reference position (X,Y)}\}$

= 0.0 0.0 0.0 0.0 0.0 0.0

ロボットの座標系上で任意の3点を決定し、データの座標(x,y)を入力します。

先ほど②で決定した3点をロボットの座標系で入力します。



④ キャリブレーション方法

 $\{<Method> : \text{Calibration method (0:3point)}\}$

= 0

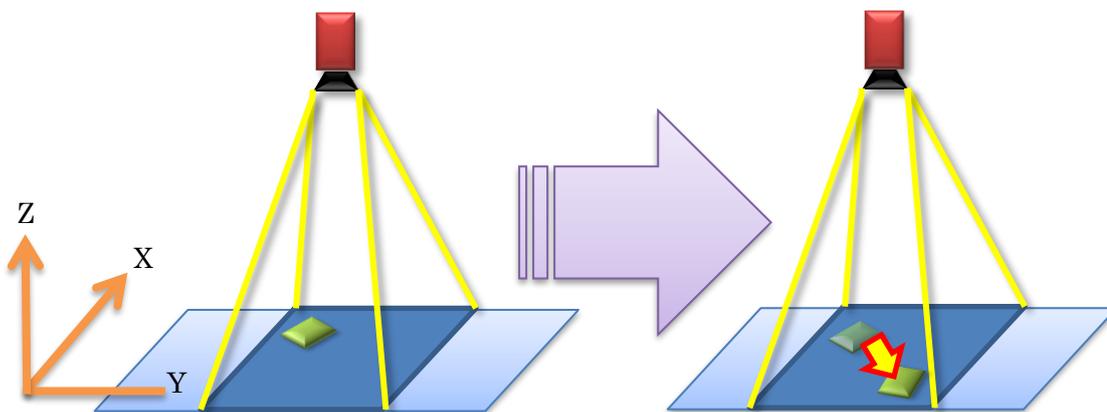
位置データの変換方式を設定します。「0」(3点教示方式)を設定してください。

⑤ ビジョンセンサの座標系タイプ

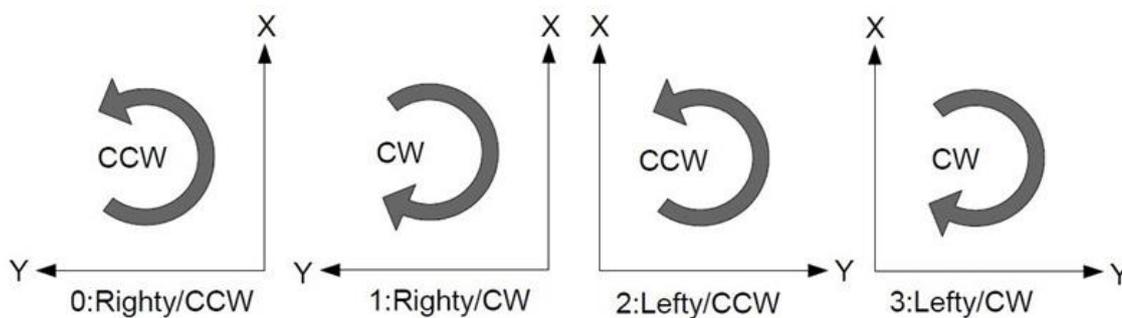
{<Coord>:Vision coordinate system(0:Righty/CW,1:Righty/CW,2:Lefty/CCW,3:Lefty/CW)}

= 0

ビジョンセンサの座標系タイプを設定します。座標系タイプは、下図のようにワークの位置をずらした際の座標値の変化により決定します。

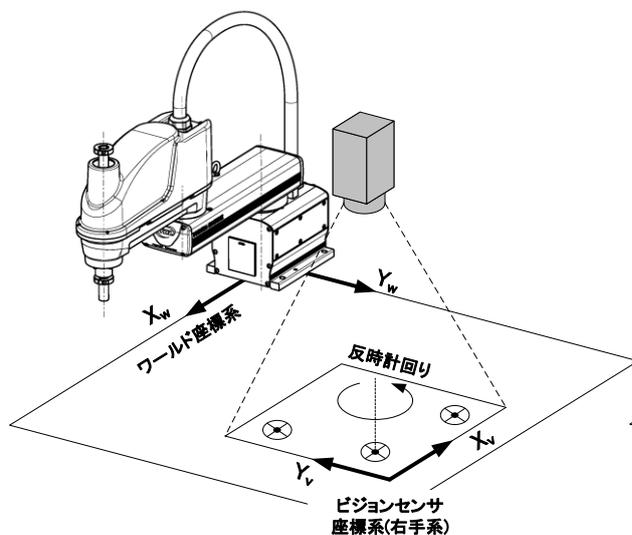


座標系タイプは、ワークの位置をずらした際の座標値が増加する方向によって、下記4種類のケースに分けられます。

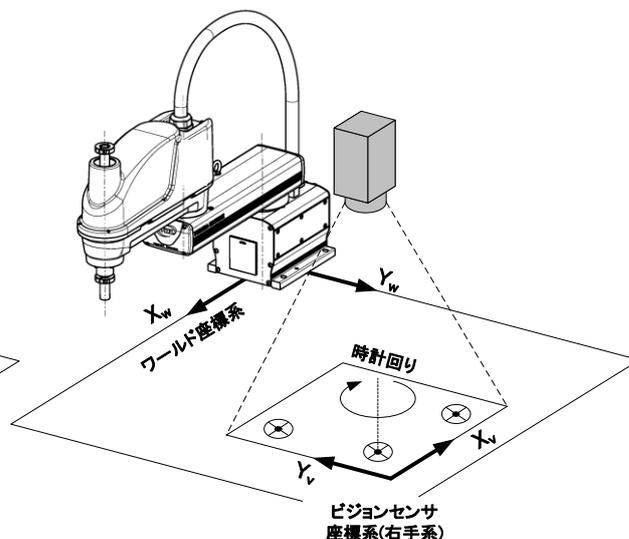


X,Yの増加する方向によって、右手系，左手系が決定し、回転させた際の増加方向で、回転方向(時計回り，反時計回り)が決定します。

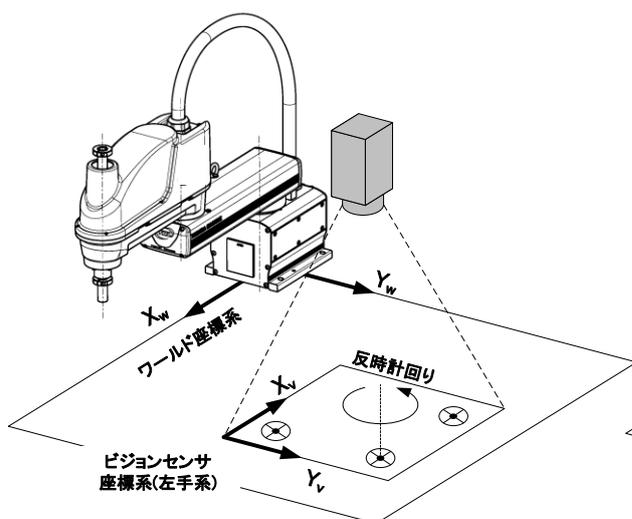
右手系，左手系と回転方向（時計回り，反時計回り）の組み合わせで、ビジョンセンサの座標系タイプを0～3の値で設定してください。



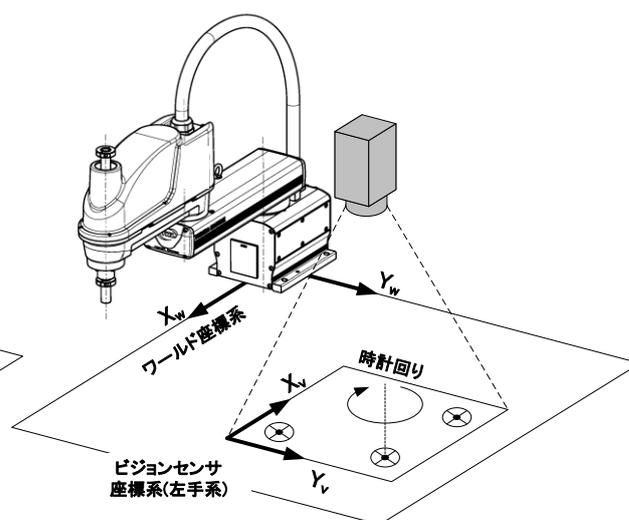
0：右手系（反時計回り）



1：右手系（時計回り）



2：左手系（反時計回り）

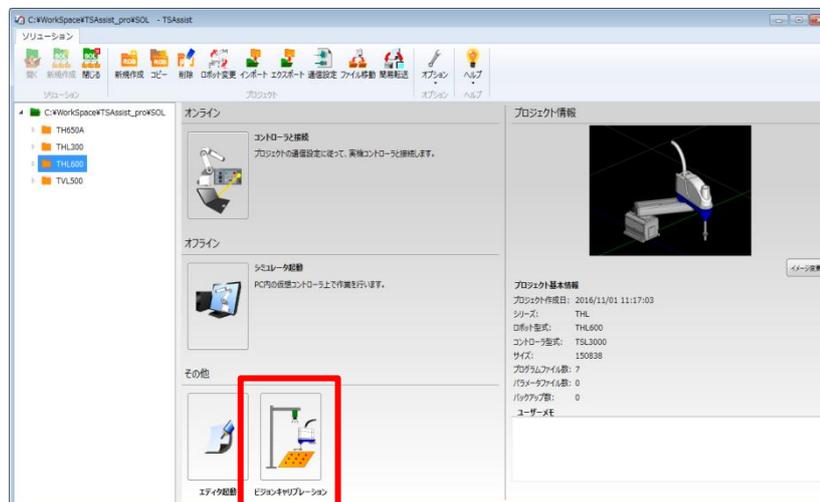


3：左手系（時計回り）

(2) [V01] 9点キャリブレーション用座標変換パラメータ

このパラメータは9点キャリブレーション用の座標変換パラメータで、TSAssist(ロボットプログラミング支援ツール)から設定することができます。詳細はTSAssistの手順に従ってください。

{Table No.21}~{Table No.40}まで最大20組設定可能です。



TSAssist画面

本機能の対応機種とコントローラシステムの対応バージョン等は以下になります。

対応機種

スカラロボット：KHLシリーズ

コントローラシステムメイン部ソフトウェア対応バージョン

スカラロボット：X8GCAS-13G以上

TSAssist対応バージョン

TSAssist ver2.1.0以上

※ver1.4.0で、下向き固定カメラのみ対応

ver2.1.0で、上向き固定カメラ、ハンドカメラ対応

その他のカメラ設置方法でのキャリブレーション機能は順次対応予定

ビジョンシステム

・KEYENCE製 CV-X100

・OMRON製 FH-1050 (OMRON製 旧F**シリーズは未対応)

※上記については動作確認済み

①テーブル番号 {Table No.**}

21～40まで用意されており、ロボットプログラムでテーブルを切り替えながら使用します。

②座標変換行列

{Coordinate transformation matrix}

= 0.0 0.0 0.0 0.0

= 0.0 0.0 0.0 0.0

= 0.0 0.0 0.0 0.0

= 0.0 0.0 0.0 0.0

③ビジョン座標系の回転角

{Vision coordinate system rotation angle (deg)}

= 0.0

④ビジョン座標系の回転方向

{Vision coordinate system rotation direction (0:CCW, 1:CW)}

= 0

7.4 言語仕様

ビジョンキャリブレーション機能では、VSCALBINとVSCALBOUTという2つの命令語を使用します。

VSCALBIN : ビジョンセンサの座標系からロボットの座標系(ワールド座標系)へ座標変換処理を行います。

VSCALBOUT : VSCALBINによってロボットの座標系へ座標変換された結果を出力します。

言語の詳細は以下になります。

VSCALBIN

機能

ビジョンセンサの座標系からロボットの座標系（ワールド座標系）へ座標変換処理を行います。

フォーマット

```
VSCALBIN (<X>, <Y>, <Z>, <C>, <T>,
          <CONFIG>, <テーブル番号>)
VSCALBIN (<X>, <Y>, <Z>, <A>, <B>, <C>,
          <U>, <V>, <CONFIG>, <テーブル番号>)
```

文例

```
VSCALBIN (100.0, 200.0, 20.0, 10.0, 0.0, 2, 1)
VSCALBIN (150.0, 100.0, 0.0, 0.0, 0.0, 20.0,
          0.0, 0.0, 12121, 3)
```

解説・注意

ビジョンセンサの座標データをビジョンセンサの座標系からロボットの座標系（ワールド座標系）へ変換します。変換した座標データはVSCALBOUTに出力されます。

<X> (mm), <Y> (mm), <C> (deg) はビジョンセンサの座標データを入力します。2次元の座標変換であるため、<Z> (mm), <T> (deg or mm), <CONFIG> (垂直多関節ロボットの場合、<Z>, <A>, , <U>, <V>, <CONFIG>) は座標変換の計算に使用しません。入力された値がVSCALBOUTにそのまま出力されます。

<テーブル番号>はVISION.PAR内のテーブル番号に対応します。1～20が設定されているときは、[V00] ビジョンキャリブレーション用座標変換パラメータ、21～40が設定されているときは、[V01] 9点キャリブレーション用座標変換パラメータに基づいて座標変換が行われます。

VSCALBOUT

機能	V S C A L B I Nによってロボットの座標系へ座標変換された結果を出力します。
フォーマット	V S C A L B O U T
文例	P 0 = V S C A L B O U T
解説・注意	V S C A L B I Nによってロボットの座標系（ワールド座標系）へ座標変換された位置データを任意の位置型データへ出力します。
	V S C A L B O U Tは、通常的位置型データと同様に使用可能ですが、数値の参照のみ可能で、数値を代入することはできません。
サンプルプログラム	<pre> PROGRAM VSCALBSMPL X = 0.0 Y = 0.0 C = 0.0 LP: MOVE HOME PRINT, IP1 "REQ", CR INPUT X, Y, C VSCALBIN(X, Y, 300.0, C, 0.0, 2, 1) P1 = VSCALBOUT MOVE P1 + POINT(, , 100.0) MOVE P1 MOVE P1 + POINT(, , 100.0) END </pre>
	ビジョンセンサにリクエストを送信
	ビジョンセンサからX, YおよびCの座標データを受信
	座標変換されたデータを位置型データへ代入

7.5 サンプルプログラム

スカラロボットを使用した場合のサンプルプログラムです。

(本サンプルプログラムでは、座標データの通信をCOM1ポートで実施するものとします。)

```
GLOBAL
  T1 = 0
  X  = 0.0
  Y  = 0.0
  C  = 0.0
END

PROGRAM CLBTEST
  MOVE HOME
LP:
  PRINT COM1, "BUFFRESET", CR           COM1ポートのバッファをクリア
  PRINT COM1, "T1", CR                 ビジョンセンサに文字列T1を送信 (トリガ)
  INPUT COM1, T1                       ビジョンセンサからトリガの返答を受信
  INPUT COM1, X, Y, C                   ビジョンセンサから座標データを受信
  IF (ABS(X) < 0.0001) AND (ABS(Y) < 0.0001) THEN GOTO LP
  VSCALBIN(X, Y, 50.0, C, 0.0, 2, 10)  受信した座標データを座標変換
  A1 = VSCALBOUT                        座標変換されたデータを位置型データへ代入
  RESTORE("A1")                        位置型データを更新
  MOVES A1
  DELAY 1.0
  MOVE HOME
  GOTO LP
END
```