

RV プロバイダ Canon RV シリーズ用プロバイダ

Version 1.0.1

ユーザーズ ガイド

September 21, 2017

【備考】

【改版履歴】

バージョン	日付	内容
1.0.0	2016-07-18	初版.
1.0.1	2017-09-21	<p>RV Version2.0.0 用コマンド対応</p> <p>【修正】</p> <ul style="list-style-type: none"> •SaveImage -受信文字列にファイル名称が2つ存在する場合, 受信戻り値が追加されるよう修正 •InputPin -コマンドの説明文の誤りを修正 •2.3.エラーコード -新規エラーコードを一覧に追加 •2.2.1 CaoWorkSpace::AddController メソッド -オプション文字列 Position の対応表を修正 <p>【新規追加】</p> <ul style="list-style-type: none"> •2.1.1. 制限事項 作成 -RV プロバイダでのコマンドを実行する上での制限事項として, データ分割機能が無効にすること旨を追加 -位置姿勢返却モードに 0 を設定できない旨を追加 •以下の新規コマンド作成 -GetScannerInfo -QueryGraspID -SyncCatchPoint -PoseWork3 -PoseWork3Async -PoseNext3 -PoseMultiWork3 -PoseMultiWork3Async -GetTime -CaptureImage

目次

1. はじめに.....	6
2. プロバイダの概要.....	7
2.1. 概要.....	7
2.1.1. 制限事項.....	7
2.2. メソッド・プロパティ.....	7
2.2.1. CaoWorkspace::AddController メソッド.....	7
2.2.1.1. Conn オプション.....	8
2.2.2. CaoController::Execute メソッド.....	9
2.3. エラーコード.....	9
2.4. コマンドリファレンス.....	12
2.4.1. 通信コマンド.....	15
2.4.1.1. CaoController::Execute("LoadTask") コマンド.....	15
2.4.1.2. CaoController::Execute("LoadTaskAsync") コマンド.....	15
2.4.1.3. CaoController::Execute("FreeTask") コマンド.....	15
2.4.1.4. CaoController::Execute("FreeTaskAsync") コマンド.....	16
2.4.1.5. CaoController::Execute("PosePallet") コマンド.....	16
2.4.1.6. CaoController::Execute("PosePalletAsync") コマンド.....	17
2.4.1.7. CaoController::Execute("PoseWork") コマンド.....	18
2.4.1.8. CaoController::Execute("PoseWorkAsync") コマンド.....	19
2.4.1.9. CaoController::Execute("PoseWork2") コマンド.....	20
2.4.1.10. CaoController::Execute("PoseWork2Async") コマンド.....	22
2.4.1.11. CaoController::Execute("PoseMultiWork") コマンド.....	24
2.4.1.12. CaoController::Execute("PoseMultiWorkAsync") コマンド.....	26
2.4.1.13. CaoController::Execute("PoseNext") コマンド.....	27
2.4.1.14. CaoController::Execute("PoseNext2") コマンド.....	28
2.4.1.15. CaoController::Execute("QueryStatus") コマンド.....	30
2.4.1.16. CaoController::Execute("ReadyCalib") コマンド.....	30
2.4.1.17. CaoController::Execute("CalibMarker") コマンド.....	31
2.4.1.18. CaoController::Execute("CatchPoint") コマンド.....	31
2.4.1.19. CaoController::Execute("VerVision") コマンド.....	33
2.4.1.20. CaoController::Execute("QueryMode") コマンド.....	33
2.4.1.21. CaoController::Execute("ReadyRim") コマンド.....	34
2.4.1.22. CaoController::Execute("InputRim") コマンド.....	35

2.4.1.23. CaoController::Execute ("ScannerPos") コマンド	35
2.4.1.24. CaoController::Execute ("FreeAll") コマンド	36
2.4.1.25. CaoController::Execute ("FreeAllAsync") コマンド	36
2.4.1.26. CaoController::Execute ("ReadyCorrect") コマンド	36
2.4.1.27. CaoController::Execute ("PoseCorrect") コマンド	37
2.4.1.28. CaoController::Execute ("SaveImage") コマンド	38
2.4.1.29. CaoController::Execute ("InputPin") コマンド	39
2.4.1.30. CaoController::Execute ("GetScannerInfo") コマンド	39
2.4.1.31. CaoController::Execute ("QueryGraspID") コマンド	40
2.4.1.32. CaoController::Execute ("SyncCatchPoint") コマンド	41
2.4.1.33. CaoController::Execute ("PoseWork3") コマンド	42
2.4.1.34. CaoController::Execute ("PoseWork3Async") コマンド	45
2.4.1.35. CaoController::Execute ("PoseNext3") コマンド	46
2.4.1.36. CaoController::Execute ("PoseMultiWork3") コマンド	48
2.4.1.37. CaoController::Execute ("PoseMultiWork3Async") コマンド	52
2.4.1.38. CaoController::Execute ("GetTime") コマンド	53
2.4.1.39. CaoController::Execute ("CaptureImage") コマンド	54
2.4.2. 独自コマンド	55
2.4.2.1. CaoController::Execute ("ExecuteCommand") コマンド	55
2.4.2.2. CaoController::Execute ("ExecuteCommandAsync") コマンド	55
2.4.2.3. CaoController::Execute ("ClearPacket") コマンド	55
2.4.2.4. CaoController::Execute ("SetTimeout") コマンド	56
2.4.2.5. CaoController::Execute ("GetTimeout") コマンド	56
2.4.2.6. CaoController::Execute ("GetCommandResult") コマンド	57

1. はじめに

本書は Canon 社製 3D マシンビジョンシステム RV シリーズ用の CAO プロバイダである, RV プロバイダのユーザーズガイドです.

RV プロバイダは Ethernet 接続された RV シリーズ用の PC で起動している RC I/F モジュールに TCP/IP で接続し, コマンド文字列の送受信を行います.

2. プロバイダの概要

2.1. 概要

RV プロバイダは、コマンドの実行方法として `CaoController::Execute` による方法を提供しています。

`CaoController::Execute` は TCP/IP 経由で、マシンビジョンシステムの RC I/F モジュールと文字列によるコマンドの応答を行います。

2.1.1. 制限事項

RV プロバイダは、RC I/F モジュールの環境設定の「送受信するデータを分割する」機能に対応していません。「送受信するデータを分割する」の値を「false」に設定してください。

また、位置姿勢情報の対象を RC I/F モジュール準拠に設定する機能に対応していません。送信時の位置姿勢返却モードは 0 に設定しないでください。

2.2. メソッド・プロパティ

2.2.1. `CaoWorkspace::AddController` メソッド

In-Sight プロバイダでは `AddController` 時に、通信用の接続パラメータを設定し、通信の接続を行います。ここで角括弧("[]")内は省略可能を示します。また、各パラメータの解説中の下線部はオプションを指定しなかった時のデフォルト値を示します。

書式 `AddController(<bstrCtrlName:VT_BSTR>,<bstrProvName:VT_BSTR>,
<bstrPcName:VT_BSTR > , <bstrOption:VT_BSTR>)`

`bstrCtrlName` : [in] コントローラ名 任意
`bstrProvName` : [in] プロバイダ名 固定値 =" CaoProv.Canon.RV"
`bstrPcName` : [in] プロバイダの実行マシン名
`bstrOption` : [in] オプション文字列

以下にオプション文字列に指定するリストを示します。

表 2-1 `CaoWorkspace::AddController` のオプション文字列

オプション	意味
<code>Conn =<接続パラメータ></code>	必須。通信形態とその接続パラメータを設定します。 詳細は 2.2.1.1 を参照してください。
<code>Timeout[=<タイムアウト時間>]</code>	送受信時のタイムアウト時間(ミリ秒)を指定します。 (デフォルト値: <u>500</u>)
<code>Delimiter[=<コマンドのデリミタ番号>]</code>	コマンド API のデリミタ文字を指定します。 <u>0</u> : カンマ

	1: 半角スペース 2: 水平 TAB 3: CR
Terminator[=<コマンド結果末尾の改行コード番号>]	コマンド実行結果末尾に付加する改行コードの種別を指定します。 0: 改行コードなし 1: CR 2: <u>LF</u> 3: CRLF
Position[=<返却する位置姿勢情報の対象>]	返却する位置姿勢情報の対象を指定します。 RC I/F の設定に合わせて設定してください。 0: <u>ワーク(重心)</u> 1: ワーク(CAD 原点) 2: ハンド(P4) 3: ハンド(P4), ハンド(P3) 4: ワーク(重心), ハンド(P4) 5: ワーク(CAD 原点), ハンド(P4) 6: ワーク(重心), ハンド(P4), ハンド(P3) 7: ワーク(CAD 原点), ハンド(P4), ハンド(P3) 8: ワーク(重心), ワーク(CAD 原点), ハンド(P4), ハンド(P3)
Timestamp[=<撮影時タイムスタンプ出力>]	撮影開始時刻と撮影終了時刻の出力の有無を指定します。 True: 出力する <u>False: 出力しない</u>

2.2.1.1. Conn オプション

以下に Conn オプションの接続パラメータ文字列を示します。

・ Ethernet デバイス

"eth:<IP Address>[:<Port No>]"

<IP Address> : : 必須. IP アドレスを指定します.

例:"127.0.0.1"

<Port No> : :RC I/F モジュールのポート番号を指定します.

(デフォルト:8889)

例:"127.0.0.1:8889"

使用例

```
Dim caoEng as CaoEngine
Dim caoCtrl as CaoController

Set caoEng = New caoEngine
Set caoCtrl = caoEng.Workspaces(0).AddController("RV", "caoProv. Canon. RV", "",
"conn=eth:192.168.0.1, timeout=800")
```

2.2.2. CaoController::Execute メソッド

RC I/F モジュールとコマンドの送受信を行います。第 1 引数にコマンド名、第 2 引数にコマンドのパラメータを指定します。各コマンドの詳細は 2.4 章コマンドリファレンスを参照してください。

書式 Execute (<bstrCommandName:VT_BSTR>,[<vntParam : VT_VARIANT>])

bstrCommandName: [in] コマンド名

vntParam : [in] パラメータ

2.3. エラーコード

コマンドを実行した際の RC I/F モジュールからのビジョンステータスは HRESULT として返されます。

S_OK(0) : 正常, エラーなし

0x80100000 + 3D ビジョンエラーコード : 3D ビジョンエラー発生

0x8011xxxx : 独自規定エラー

例: LoadTask を実行したとき。

hr = 0x80103203 : タスク数上限エラー (10 進数: 12803)

3D ビジョンのエラーは表 2-2, 表 2-3, 表 2-4 を参照してください。内容については Canon 社の 3D マシンビジョンシステムユーザーズガイドを参照してください。

独自規定エラーは表 2-5 を参照してください。

その他プロバイダ共通のエラー内容については ORIN2SDK プログラマーズユーザーズガイドを参照してください。

表 2-2 3D ビジョン 致命的エラー

番号 (10 進数表記)	説明
63489	LED 昇温異常
63490	LED 回路異常
63492	Ethernet I/F 異常

63493	USB I/F 異常
63494	Serial I/F 異常
63495	RAM Read エラー
63496	RAM Write エラー
63497	ROM Read エラー
63498	ROM Write エラー
63499	投影用デバイス異常
63500	カメラ異常
63501	LED 昇温異常解除
63502	割り込み I/F 異常
63503	システム起動エラー
63504	EEPROM 異常
63505	RTC 異常
63712	デバイス切断
63728	3D スキャナー通信タイムアウト
62465	メモリ確保に失敗
61959	ファイル破損エラー(致命的)
61972	ライセンス認証エラー

表 2-3 3D ビジョン 警告的エラー

番号 (10 進数表記)	説明
29698	HDD 残容量不足
29190	“測距ずれ量の計測”機能における測距ずれ許容量超過 マーカー1 点の場合:測距ずれ量 マーカー2 点以上で「補正を実施する」にチェック無しの場合:測距ずれ量 マーカー2 点以上で「補正を実施する」にチェック有りの場合:補正後の残差
29189	モード番号エラー

表 2-4 3D ビジョン 一般的エラー

番号 (10 進数表記)	説明
12801	未定義コマンド
12802	シーケンス外コマンド(モードとコマンドの不一致)
12803	タスク数上限エラー
12804	キャリブレーションシーン番号エラー
12806	不完全タスクエラー
12807	ファイル破損エラー(一時的)
12808	タスク未ロードエラー
12809	ワーク未計測
12810	シーン数未設定
12811	キャリブレーションシーン数範囲外
12812	アプリケーションビジー状態
12813	マーカ検出失敗
12814	パレット計測失敗
12815	把持情報 ID エラー
12816	パレット未計測
12817	マーカ位置不正
12818	不正な 3D スキャナー位置 ID が指定された
12819	RV キャリブレーション未実施
12820	ワーク計測タイムアウト
12821	撮影未実施(タスク未ロード,ワーク計測未実施の場合)
12822	ファイル保存失敗
12823	ワークの最大個数設定エラー
12824	探索する ROI 数設定エラー
12825	POSE_NEXT のシーケンスエラー(事前に複数の認識ワークを通知している状態)
12826	ROI(部品検出領域)サイズ倍率設定エラー(ROI サイズ倍率の設定が範囲外)
12827	GPU リソースエラー(指定された ROI(部品検出領域)サイズ倍率だとリソースが不足する場合)
12828	ROI(部品検出領域)の中心座標エラー
12829	認識スコア閾値設定エラー
12830	概略検出ワーク上限数設定エラー
12831	モデルフィッティングワーク上限数設定エラー
12832	把持候補出力上限数設定エラー

12833	露光時間設定エラー
12834	位置姿勢返却モードエラー
12835	CAD 原点未登録
12836	スキャナー未接続
12964	把持情報不一致

表 2-5 独自規定エラーコード一覧

エラー名	エラー番号	説明
E_STATUS_ERROR	0x80100000 RVエラー	3D マシンビジョンシステムエラー
E_INVALID_COMMAND	0x80110001	コマンド文字列が不正です
E_NOT_EQUAL_ID	0x80110002	送受信したコマンド文字列の ID が不一致です
E_RECIEVED_PACKET	0x80110003	受信したコマンド文字列が異常です
E_GET_STATUS	0x80110004	受信したコマンド文字列解析時にエラーが発生しました
E_NO_DATA	0x80110005	コマンド文字列が受信できません
E_LENGTH_ORVER	0x80110006	コマンド文字列が長すぎます
E_COMMAND_EXCUTING	0x80110007	非同期コマンドを実行中です
E_COMMAND_NOTEXECUTE	0x80110008	非同期コマンドを実行せずに GetCommandResult を実行しました

2.4. コマンドリファレンス

本章では CaoController::Execute メソッドの各コマンドについて解説します。各コマンドの詳細動作については Canon 社の 3D マシンビジョンシステムユーザーズガイドの通信リファレンスを参照してください。

表 2-6 CaoController::Execute コマンド一覧

RV コマンド	コマンド	機能	
通信コマンド			
LOAD_TASK	LoadTask	タスク情報を情報管理ファイルから読み込みます。	P15

	LoadTaskAsync	非同期で LoadTask を実行します。	P15
FREE_TASK	FreeTask	タスク情報を開放します。	P15
	FreeTaskAsync	非同期で FreeTask を実行します。	P16
POSE_PALLET	PosePallet	パレットの位置・姿勢を計測します。	P16
	PosePalletAsync	非同期で PosePallet を実行します。	P17
POSE_WORK	PoseWork	把持可能なワークの位置・姿勢を計測します。	P18
	PoseWorkAsync	非同期で PoseWork を実行します。	P19
POSE_WORK2	PoseWork2	PoseWork を実行します。 ワーク計測に関するパラメータの変更も行えます。 ハンド P3 の位置・姿勢や撮影開始・終了時刻も取得できます。	P20
	PoseWork2Async	非同期で PoseWork2 を実行します。	P22
POSE_MULTI_WORK	PoseMultiWork	把持可能なワークの位置・姿勢を計測し、全ての候補を返却します。 ワーク計測に関するパラメータの変更も行えます。	P24
	PoseMultiWorkAsync	非同期で PoseMultiWorkAsync を実行します。	P26
POSE_NEXT	PoseNext	新たな計測はせず、直近の計測で算出した次点の把持可能ワークの位置・姿勢を取得します。	P27
POSE_NEXT2	PoseNext2	PoseNext を実行します。ハンド P3 の位置姿勢も取得できます。	P28
QUERY_STATUS	QueryStatus	3D マシンビジョンシステムの状態を取得します。	P30
READY_CALIB	ReadyCalib	RV キャリブレーションでマーカー計測を行うシーンを設定します。	P30
CALIB_MARKER	CalibMarker	RV キャリブレーションマーカーの中心座標を計測します。	P31
CATCH_POINT	CatchPoint	ワークの把持位置情報を設定します。	P31
VER_VISION	VerVision	3D マシンビジョンシステムのバージョン情報を取得します。	P33
QUERY_MODE	QueryMode	3D マシンビジョンシステムの実行モード状態を取得します。	P33
READY_RIM	ReadyRim	パレット平面推定に使用する計測シー数を設定します。	P34
INPUT_RIM	InputRim	パレット平面推定に使用するロボット位置を設定します。	P35
SCANNER_POS	ScannerPos	3D スキャナーの位置を設定します。	P35
FREE_ALL	FreeAll	読み込まれているすべてのタスク情報を開放します。	P36
	FreeAllAsync	非同期で FreeAll を実行します。	P36
READY_CORRECT	ReadyCorrect	座標変換補正パラメータを算出する為のマーカー計測シー数を設定します。	P36
POSE_CORRECT	PoseCorrect	マーカーの中心位置座標計測を行います。	P37

SAVE_IMAGE	SaveImage	直近の撮影画像を保存します。	P38
INPUT_PIN	InputPin	ロボットピン先端の中心座標の入力を行います。	P39
GET_SCANNER_INFO	GetScannerInfo	接続されている 3D スキャナーの情報を取得します。	P39
QUERY_GRASP_ID	QueryGraspID	ロボットコントローラの同期状態の情報を取得します。	P40
SYNC_CATCH_POINT	SyncCatchPoint	ワーク・ハンドの位置・姿勢の情報を取得します。	P41
POSE_WORK3	PoseWork3	PoseWork を実行します。 ワーク計測に関するパラメータの変更も行えます。 撮影開始・終了時刻も取得できます。 位置・姿勢の情報を取得する対象を変更できます。	P42
	PoseWork3Async	非同期で PoseWork3 を実行します。	P45
POSE_NEXT3	PoseNext3	PoseNext を実行します。 撮影開始・終了時刻も取得できます。 位置・姿勢の情報を取得する対象を変更できます。	P46
POSE_MULTI_WORK3	PoseMultiWork3	PoseMultiWork を実行します。 ワーク計測に関するパラメータの変更も行えます。 撮影開始・終了時刻も取得できます。 位置・姿勢の情報を取得する対象を変更できます。	P48
	PoseMultiWork3Async	非同期で PoseMultiWork3 を実行します。	P52
GET_TIME	GetTime	撮影時間とタクトタイムを取得します。	P53
CAPTURE_IMAGE	CaptureImage	画像を撮影し、ビットマップ画像形式で取得します。	P54
独自コマンド			
-	ExecuteCommand	RC I/F モジュールにコマンド文字列を送信して、命令を実行します。	P55
	ExecuteCommandAsync	非同期で ExecuteCommand を実行します。	P55
-	ClearPacket	RC I/F モジュールからの応答結果を削除します。	P55
-	SetTimeout	応答結果を受信する際の Timeout 時間を設定します。	P56
-	GetTimeout	応答結果を受信する際の Timeout 時間を取得します。	P56
-	GetCommandResult	非同期で実行したコマンドの結果を取得します。	P57

2.4.1. 通信コマンド

2.4.1.1. GaoController::Execute("LoadTask") コマンド

計測対象ワークに関連するタスク情報をファイルから読み込みます。

書式

LoadTask (<TaskID>)

引数 : [in]<TaskID>: ロードするタスク ID [VT_I4]

戻り値 : なし

タスク(ワーク)を動的に切り替え、同時に複数のタスクをロードすることが可能です(最大 8 個まで)。

使用例

```
Dim iTaskID as Integer = 1
m_caoCtrl.Execute "LoadTask", iTaskID
```

2.4.1.2. GaoController::Execute("LoadTaskAsync") コマンド

非同期で LoadTask コマンドを実行します。

書式

LoadTaskAsync (<TaskID>)

引数 : [in]<TaskID>: ロードするタスク ID [VT_I4]

戻り値 : なし

実行後"GetCommandResult"コマンドで実行結果を確認してください。"GetCommandResult"を実行する前に他のコマンドを使用すると、E_COMMAND_EXECUTING のエラーが発生します。

E_TIMEUOT(0x80000900)が発生した場合は、再度"GetCommandResult"を使用してコマンド実行結果を受信できます。結果が受信できない時は"ClearPacket"コマンドで実行状態を解除してください。

使用例

```
Dim iTaskID as Integer = 1
m_caoCtrl.Execute "LoadTask", iTaskID

m_caoCtrl.Execute "GetCommandResult"
```

2.4.1.3. GaoController::Execute("FreeTask") コマンド

LoadTask でロードされているタスク情報を開放します。

書式

FreeTask (<TaskID>)

引数 : [in]<TaskID>: 解放するタスク ID [VT_I4]

戻り値 : なし

ロードされているすべてのタスク情報を解放する場合は, FreeAll コマンドを使用してください。

使用例

```
Dim iTaskID as Integer = 1
m_caoCtrl.Execute "FreeTask", iTaskID
```

2.4.1.4. GaoController::Execute ("FreeTaskAsync") コマンド

非同期で FreeTask を実行します。

書式

FreeTaskAsync (<TaskID>)

引数 : [in]<TaskID>: 解放するタスク ID [VT_I4]

戻り値 : なし

実行後"GetCommandResult"コマンドで実行結果を確認してください。"GetCommandResult"を実行する前に他のコマンドを使用すると, E_COMMAND_EXCUTING のエラーが発生します。

E_TIMEUOT(0x80000900)が発生した場合は, 再度"GetCommandResult"を使用してコマンド実行結果を受信できます。結果が受信できない時は"ClearPacket"コマンドで実行状態を解除してください。

ロードされているすべてのタスク情報を非同期で解放する場合は, FreeAllAsync コマンドを使用してください。

使用例

```
Dim iTaskID as Integer = 1
m_caoCtrl.Execute "FreeTaskAsync", iTaskID

m_caoCtrl.Execute "GetCommandResult"
```

2.4.1.5. GaoController::Execute ("PosePallet") コマンド

パレットの位置・姿勢を計測します。

書式

PosePallet (<TaskID>)

引数 : [in]<TaskID>: タスク ID [VT_I4]

戻り値 : [out] パレット位置情報 (<Tx>, <Ty>, <Tz>, <Rx>, <Ry>, <Rz>)
[VT_R8 | VT_ARRAY]

<Tx>: 並進ベクトル x

<Ty>: 並進ベクトル y

<Tz>: 並進ベクトル z

<Rx>: 回転行列 x

<Ry>: 回転行列 y

<Rz>: 回転行列 z

計測対象のパレットは、タスク ID で指定されたタスクに属するパレットです。

使用例

```
Dim vntPose As Object
vntPose = m_caoCtrl.Execute("PosePallet", 1)
```

2.4.1.6. GaoController::Execute ("PosePalletAsync") コマンド

非同期で PosePalletAsync を実行します。

書式 PosePalletAsync (<TaskID>)

引数 : [in]<TaskID>: タスク ID [VT_I4]

戻り値 : なし

実行後"GetCommandResult"コマンドで実行結果を確認してください。"GetCommandResult"を実行する前に他のコマンドを使用すると、E_COMMAND_EXCUTING のエラーが発生します。

E_TIMEUOT(0x80000900)が発生した場合は、再度"GetCommandResult"を使用してコマンド実行結果を受信できます。結果が受信できない時は"ClearPacket"コマンドで実行状態を解除してください。

使用例

```
Dim vntPose As Object
call m_caoCtrl.Execute("PosePalletAsync", 1)

vntPose = m_caoCtrl.Execute("GetCommandResult")
```

2.4.1.7. GaoController::Execute (“PoseWork”) コマンド

把持可能なワークの位置・姿勢を計測します。



PoseWork (<TaskID>)

引数 : [in]<TaskID>: 計測したいワークのタスク ID (VT_I4)

戻り値 : [out] ワーク情報 (<Pallet>, <Work>, <Value>, <HandID>, <WorkPos>, <Shield>, <Num>, <Pos1>, <Pos2>, <Pos3>, <Pos4>, <Pos5>)
[VT_VARIANT|VT_ARRAY]

<Pallet> : パレット空情報 [VT_I4]

0 : 空である, 1 : 空でない, 65535 : 空判定ができない

<Work> : 把持候補の有無 [VT_I4]

0 : 候補なし, 1 : 候補あり

<Value> : ワーク評価値 [VT_I4]

<HandID> : 把持情報ID [VT_I4]

<WorkPos> : ワークの位置・姿勢(Tx, Ty, Tz, Rx, Ry, Rz)

[VT_R8|VT_ARRAY]

<Shield> : 遮蔽率 [VT_R8]

<Num> : 3次元計測点が局地的に高い位置の個数 [VT_I4]

<Posn(n=1~5)> : 局地的に高い位置 (X, Y, Z)

[VT_R8|VT_ARRAY]

計測対象のワークは、タスク ID で指定されたタスクに属するワークです。

注意)<Shield>は実数の値を返します。例)3.8% ⇒ 3.8 99.9% ⇒ 99.9



```
Dim vntRet As Object
Dim iTaskID As Integer = 1
vntRet = m_caoCtrl.Execute("PoseWork", iTaskID)
```

```
Dim iPallet As Integer
Dim iWork As Integer
Dim iValue As Integer
Dim iHandID As Integer
Dim vntWorkPos As Object
Dim dblShield As Double
Dim iNum As Integer
Dim vntPos(4) As Object
Dim iCnt As Integer
```

```
iPallet = vntRet(0)
```


Next

2.4.1.9. CaoController::Execute (“PoseWork2”) コマンド

把持可能なワークの位置・姿勢を計測します。

ワーク計測に関するパラメータについて、本コマンドで指定することで設定値を変更することができます。

パラメータ省略時には変更されません。



PoseWork2(<TaskID> [, <ROISize>, <ROI X>, <ROI Y>, <Score>, <WorkMax>, <FitMax>, <CatchMax>, <Exposure>])

引数 : [in] パラメータ

<TaskID>: 計測したいワークのタスク ID [VT_I4]

<ROISize>: ROI のサイズ, 1~4(GUP メモリ依存) [VT_I4]

<ROI X>: ROI の中心位置 X, 1~2048 [VT_I4]

<ROI Y>: ROI の中心位置 Y, 1~2048 [VT_I4]

<Score>: 認識スコアの閾値, 1.0~100.0 [VT_R8]

<WorkMax>: 概略検出ワーク上限数, 1~100 [VT_I4]

<FitMax>: モデルフィッティングワーク上限数, 1~20 [VT_I4]

<CatchMax>: 把持候補出力上限数, 1~20 [VT_I4]

<Exposure>: 露光時間 [VT_R8]

0.05~54.0 の範囲で設定可能 (RV1100)

0.05~47.0 の範囲で設定可能 (RV500, RV300)

戻り値 : [out] ワーク情報 (<Pallet>, <Work>, <Value>, <HandID>, <WorkPos>[, <HandP3>], <Shield>, <Num>, <Pos1>, <Pos2>, <Pos3>, <Pos4>, <Pos5> [, <StartTime>, <EndTime>]) [VT_VARIANT|VT_ARRAY]

<Pallet>: パレット空情報 [VT_I4]

0: 空である, 1: 空でない, 65535: 空判定ができない

<Work>: 把持候補の有無 [VT_I4]

0: 候補なし, 1: 候補あり

<Value>: ワーク評価値 [VT_I4]

<HandID>: 把持情報ID [VT_I4]

<WorkPos>: ワークの位置・姿勢(X, Y, Z, Rx, Ry, Rz)

[VT_R8|VT_ARRAY]

<HandP3>: ワーク又はハンドP3 の位置・姿勢(*2)

(X, Y, Z, Rx, Ry, Rz) [VT_R8|VT_ARRAY]
 <Shield>: 遮蔽率(*1) [VT_R8]
 <Num>: 3次元計測点が局地的に高い位置の個数 [VT_I4]
 <Posn(n=1~5)>: 局地的に高い位置 (X, Y, Z)
 [VT_R8|VT_ARRAY]
 <StartTime>: 撮影開始時刻(*2) [VT_BSTR]
 <EndTime>: 撮影終了時刻(*2) [VT_BSTR]

計測対象のワークは、タスク ID で指定されたタスクに属するワークです。

注意)

*1 <Shield>は実数の値を返します。例)3.8% ⇒ 3.8 99.9% ⇒ 99.9

*2 RC I/F の設定により戻り値の形式が異なります。

<HandP3>は AddController のオプション Position=3 を指定した時に出力します。

<StartTime>,<EndTime>は AddController のオプション Timestamp=True を指定した時に出力します。

使用例

```

Dim vntRet As Object
Dim iCnt As Integer

Dim iTaskID As Integer = 1

Dim iPallet As Integer
Dim iWork As Integer
Dim iValue As Integer
Dim iHandID As Integer
Dim vntWorkPos As Object
Dim vntHandP3 as Object
Dim dblShield As Double
Dim iNum As Integer
Dim vntPos(4) As Object
Dim strStartTime as String
Dim strEndTime as String

' ----- パラメータ省略時 -----
m_caoCtrl = m_caoEng.Workspaces(0).AddController("RV", "caoProv. Canon. RV", "",
"conn=eth:192.168.0.1")
vntRet = m_caoCtrl.Execute("PoseWork2", iTaskID)

iPallet = vntRet(0)
iWork = vntRet(1)
iValue = vntRet(2)
iHandID = vntRet(3)
vntWorkPos = vntRet(4)
dblShield = vntRet(5)
iNum = vntRet(6)
For iCnt = 0 To 4
    vntPos(iCnt) = vntRet(7 + iCnt)
Next

' ----- パラメータ設定時 -----
Dim vntParam(8) As Object
vntParam(0) = iTaskID
  
```

```

vntParam(1) = 2 ' Size
vntParam(2) = 512 ' ROIX
vntParam(3) = 512 ' ROIY
vntParam(4) = 88.8 ' Score
vntParam(5) = 50 ' WorkMax
vntParam(6) = 10 ' FitMax
vntParam(7) = 8 ' CatchMax
vntParam(8) = 40.4 ' Exposure

m_caoCtrl = caoEng.Workspaces(0).AddController("RV", "caoProv. Canon. RV", "",
"conn=eth:192.168.0.1")

vntRet = m_caoCtrl.Execute("PoseWork2", vntParam)

iPallet = vntRet(0)
iWork = vntRet(1)
iValue = vntRet(2)
iHandID = vntRet(3)
vntWorkPos = vntRet(4)
dblShield = vntRet(5)
iNum = vntRet(6)
For iCnt = 0 To 4
    vntPos(iCnt) = vntRet(7 + iCnt)
Next

' ----- RC I/F で P3 と撮影時刻を出力している時 -----
m_caoCtrl = m_caoEng.Workspaces(0).AddController("RV", "caoProv. Canon. RV", "",
"conn=eth:192.168.0.1, Position=3, Timestamp=true")
vntRet = m_caoCtrl.Execute("PoseWork2", iTaskID)

iPallet = vntRet(0)
iWork = vntRet(1)
iValue = vntRet(2)
iHandID = vntRet(3)
vntWorkPos = vntRet(4)
vntHandP3 = vntRet(5)
dblShield = vntRet(6)
iNum = vntRet(7)
For iCnt = 0 To 4
    vntPos(iCnt) = vntRet(8 + iCnt)
Next
strStartTime = vntRet(13)
strEndTime = vntRet(14)

```

2.4.1.10. CaoController::Execute ("PoseWork2Async") コマンド

非同期で PoseWork2 を実行します。

書式 PoseWork2Async(<TaskID> [, <ROISize>, <ROIX>, <ROIY>, <Score>, <WorkMax>, <FitMax>, <CatchMax>, <Exposure>])

引数 : [in] パラメータ

<TaskID>: 計測したいワークのタスク ID [VT_I4]

<ROISize>: ROI のサイズ, 1~4(GUP メモリ依存) [VT_I4]

<ROIX>: ROI の中心位置 X, 1~2048 [VT_I4]

<ROIY>: ROI の中心位置 Y, 1~2048 [VT_I4]
 <Score>: 認識スコアの閾値, 1.0~100.0 [VT_R8]
 <WorkMax>: 概略検出ワーク上限数, 1~100 [VT_I4]
 <FitMax>: モデルフィッティングワーク上限数, 1~20 [VT_I4]
 <CatchMax>: 把持候補出力上限数, 1~20 [VT_I4]
 <Exposure>: 露光時間 [VT_R8]
 0.05~54.0 の範囲で設定可能 (RV1100)
 0.05~47.0 の範囲で設定可能 (RV500, RV300)

戻り値 : なし

実行後"GetCommandResult"コマンドで実行結果を確認してください。"GetCommandResult"を実行する前に他のコマンドを使用すると、E_COMMAND_EXECUTING のエラーが発生します。

E_TIMEUOT(0x80000900)が発生した場合は、再度"GetCommandResult"を使用してコマンド実行結果を受信できます。結果が受信できない時は"ClearPacket"コマンドで実行状態を解除してください。

使用例

```
Dim vntRet As Object
Dim iCnt As Integer

Dim iTaskID As Integer = 1

Dim vntParam(8) As Object
vntParam(0) = iTaskID
vntParam(1) = 2 ' Size
vntParam(2) = 512 ' ROIx
vntParam(3) = 512 ' ROIy
vntParam(4) = 88.8 ' Score
vntParam(5) = 50 ' WorkMax
vntParam(6) = 10 ' FitMax
vntParam(7) = 8 ' CatchMax
vntParam(8) = 40.4 ' Exposure

m_caoCtrl = caoEng.Workspaces(0).AddController("RV", "caoProv.Canon.RV", "",
"conn=eth:192.168.0.1")

m_caoCtrl.Execute("PoseWork2Async", vntParam)

vntRet = caoCtrl.Execute("GetCommandResult")

iPallet = vntRet(0)
iWork = vntRet(1)
iValue = vntRet(2)
iHandID = vntRet(3)
vntWorkPos = vntRet(4)
dblShield = vntRet(5)
iNum = vntRet(6)
For iCnt = 0 To 4
  vntPos(iCnt) = vntRet(7 + iCnt)
```

Next

2.4.1.11. GaoController::Execute (“PoseMultiWork”) コマンド

把持可能なワークの位置・姿勢を計測し、全ての候補を返却します。

計測対象のワークは、タスク ID で指定されたタスクに属するワークです。

ワークの最大個数に達するまでは探索 ROI 数で設定された ROI を探索し続けます。

ワークの最大個数に達しない場合でも探索する ROI 数の探索が完了すると、応答を返します。

ワーク計測に関するパラメータについて、本コマンドで指定することで設定値を変更することができます

書式 PoseMultiWork (<TaskID>, <WorkMaxCnt>, <ROICnt> [,<ROISize>, <ROIY1>, <ROIY1>, <ROIYn>, <ROIYn>, <Score>, <WorkMax>, <FitMax>, <CatchMax>, <Exposure>)

引数 : [in]パラメータ (VT_VARIANT | VT_ARRAY)

<TaskID>: 計測するワークが属するタスク ID [VT_I4]

<WorkMaxCnt>: ワークの最大個数 [VT_I4]

<ROICnt>: 探索する ROI 数 [VT_I4]

<ROISize>: ROI のサイズ, 1~4 [VT_I4]

<ROIYn>: ROI の中心位置 X, 1~2048 (*1) [VT_I4]

<ROIYn>: ROI の中心位置 Y, 1~2048 (*1) [VT_I4]

<Score>: 認識スコア閾値, 1.0~100.0 [VT_R8]

<WorkMax>: 概略検出ワーク上限数, 1~100 [VT_I4]

<FitMax>: モデルフィッティングワーク上限数, 1~20 [VT_I4]

<CatchMax>: 把持候補出力上限数, 1~20 [VT_I4]

<Exposure>: 露光時間 [VT_R8]

0.05~54.0 (RV1100)

0.05~47.0 (RV500, RV300)

戻り値 : [out] ワーク情報 (<Pallet>, <WorkCnt>, <Value>, <HandID>, <WorkPos1> [,<HandP31>, ,<WorkPosn> [, <HandP3n>], <Shield>, <Num>, <Pos1>,<Pos2>,<Pos3>,<Pos4>,<Pos5> [, <StartTime>, [EndTime]]) [VT_VARIANT|VT_ARRAY]

<Pallet> : パレット空情報 [VT_I4]

0 : 空である, 1 : 空でない, 65535 : 空判定ができない

<WorkCnt> : 把持候補数 [VT_I4]

<Value> : ワーク評価値 [VT_I4]
 <HandID> : 把持情報ID [VT_I4]
 <WorkPosn> : ワークの位置・姿勢(*3) (Tx, Ty, Tz, Rx, Ry, Rz)
 [VT_R8|VT_ARRAY]
 <HandP3n>: ハンドP3の位置・姿勢(*2,3) (X, Y, Z, Rx, Ry, Rz)
 [VT_R8 | VT_ARRAY]
 <Shield> : 遮蔽率 [VT_R8]
 <Num> : 3次元計測点が局地的に高い位置の個数 [VT_I4]
 <Posn(n=1~5)> : 局地的に高い位置 (X, Y, Z)
 [VT_R8|VT_ARRAY]
 <StartTime>: 撮影開始時刻(*2) [VT_BSTR]
 <EndTime>: 撮影終了時刻(*2) [VT_BSTR]

注意)

*1 ROI の中心位置 X, Y については探索する ROI 数と同じ数だけ設定してください。

*2 RC I/F の設定により戻り値の形式が異なります。

<HandP3>は AddController のオプション Position=3 を指定した時に出力します。

<StartTime>,<EndTime>は AddController のオプション Timestamp=True を指定した時に出力します。

*3 ワークの位置とハンド P3 の位置姿勢は把持候補数の数分返ります。

使用例

```

Dim vntRet As Object
Dim iTaskID As Integer = 1
vntRet = m_caoCtrl.Execute("PoseNext", iTaskID)

Dim iPallet As Integer
Dim iWork As Integer
Dim iValue As Integer
Dim iHandID As Integer
Dim vntWorkPos As Object
Dim dblShield As Double
Dim iNum As Integer
Dim vntPos(4) As Object
Dim iCnt As Integer

iPallet = vntRet(0)
iWork = vntRet(1)
iValue = vntRet(2)
iHandID = vntRet(3)
vntWorkPos = vntRet(4)
dblShield = vntRet(5)
iNum = vntRet(6)
For iCnt = 0 To 4
    vntPos(iCnt) = vntRet(7 + iCnt)
Next
  
```

2.4.1.12. GaoController::Execute ("PoseMultiWorkAsync") コマンド

非同期で PoseMultiWork を実行します。

書式 PoseMultiWorkAsync (<TaskID>, <WorkMaxCnt>, <ROICnt> [,<ROISize>, <ROIx1>, <ROIy1>, ... <ROIxn>, <ROIyn>, <Score>, <WorkMax>, <FitMax>, <CatchMax>, <Exposure>)

引数 : [in]パラメータ (VT_VARIANT | VT_ARRAY)

- <TaskID>: 計測するワークが属するタスク ID [VT_I4]
- <WorkMaxCnt>: ワークの最大個数 [VT_I4]
- <ROICnt>: 探索する ROI 数 [VT_I4]
- <ROISize>: ROI のサイズ, 1~4 [VT_I4]
- <ROIxn>: ROI の中心位置 X, 1~2048 (*1) [VT_I4]
- <ROIyn>: ROI の中心位置 Y, 1~2048 (*1) [VT_I4]
- <Score>: 認識スコア閾値, 1.0~100.0 [VT_R8]
- <WorkMax>: 概略検出ワーク上限数, 1~100 [VT_I4]
- <FitMax>: モデルフィッティングワーク上限数, 1~20 [VT_I4]
- <CatchMax>: 把持候補出力上限数, 1~20 [VT_I4]
- <Exposure>: 露光時間 [VT_R8]

0.05~54.0 (RV1100)
0.05~47.0 (RV500, RV300)

戻り値 : なし

実行後"GetCommandResult"コマンドで実行結果を確認してください。"GetCommandResult"を実行する前に他のコマンドを使用すると, E_COMMAND_EXCUTING のエラーが発生します。

E_TIMEUOT(0x80000900)が発生した場合は, 再度"GetCommandResult"を使用してコマンド実行結果を受信できます。結果が受信できない時は"ClearPacket"コマンドで実行状態を解除してください。

使用例

```
Dim vntRet As Object
Dim iTaskID As Integer = 1
vntRet = m_caoCtrl.Execute("PoseNextAsync", iTaskID)

Dim iPallet As Integer
Dim iWork As Integer
Dim iValue As Integer
Dim iHandID As Integer
```

```

Dim vntWorkPos As Object
Dim dblShield As Double
Dim iNum As Integer
Dim vntPos(4) As Object
Dim iCnt As Integer

iPallet = vntRet(0)
iWork = vntRet(1)
iValue = vntRet(2)
iHandID = vntRet(3)
vntWorkPos = vntRet(4)
dblShield = vntRet(5)
iNum = vntRet(6)
For iCnt = 0 To 4
    vntPos(iCnt) = vntRet(7 + iCnt)
Next

```

2.4.1.13. CaoController::Execute ("PoseNext") コマンド

新たな計測はせず、直近の計測で算出した次点の把持可能ワークの位置・姿勢を返します。



PoseNext (<TaskID>)

<TaskID> : [in] 計測したいワークのタスク ID (VT_I4)

戻り値 : [out] ワーク情報 (<Pallet>, <Work>, <Value>, <HandID>, <WorkPos>, <Shield>, <Num>, <Pos1>, <Pos2>, <Pos3>, <Pos4>, <Pos5>)
[VT_VARIANT|VT_ARRAY]

<Pallet> : パレット空情報 [VT_I4]

0 : 空である, 1 : 空でない, 65535 : 空判定ができない

<Work> : 把持候補の有無 [VT_I4]

0 : 候補なし, 1 : 候補あり

<Value> : ワーク評価値 [VT_I4]

<HandID> : 把持情報ID [VT_I4]

<WorkPos> : ワークの位置・姿勢(Tx, Ty, Tz, Rx, Ry, Rz)

[VT_R8|VT_ARRAY]

<Shield> : 遮蔽率 [VT_R8]

<Num> : 3次元計測点が局地的に高い位置の個数 [VT_I4]

<Posn(n=1~5)> : 局地的に高い位置 (X, Y, Z)

[VT_R8|VT_ARRAY]

先に PoseWork 候補ワークをロボット側の理由で把持不可能と判断した場合に、ワーク山は崩れていないと判断し、新たに計測せずに直近の計測での次点(把持優先順の次点)の把持可能ワークの位置・姿勢を返します。

使用例

```
Dim vntRet As Object
Dim iTaskID As Integer = 1
vntRet = m_caoCtrl.Execute("PoseNext", iTaskID)

Dim iPallet As Integer
Dim iWork As Integer
Dim iValue As Integer
Dim iHandID As Integer
Dim vntWorkPos As Object
Dim dblShield As Double
Dim iNum As Integer
Dim vntPos(4) As Object
Dim iCnt As Integer

iPallet = vntRet(0)
iWork = vntRet(1)
iValue = vntRet(2)
iHandID = vntRet(3)
vntWorkPos = vntRet(4)
dblShield = vntRet(5)
iNum = vntRet(6)
For iCnt = 0 To 4
    vntPos(iCnt) = vntRet(7 + iCnt)
Next
```

2.4.1.14. CaoController::Execute ("PoseNext2") コマンド

新たな計測はせず、直近の計測で算出した次点の把持可能ワークの位置・姿勢、もしくはハンドの把持位置・姿勢について返します。

先に POSE_WORK もしくは POSE_WORK2 で受け取った把持候補ワークをロボット側の理由で把持不可能と判断した場合に、ワーク山は崩れていないと判断し、新たに計測せずに直近の計測での次点(把持優先順の次点)の把持可能ワークの位置・姿勢、もしくはハンドの把持位置・姿勢を返します。

書式

PoseNext2 (<TaskID>)

<TaskID> : [in] 計測したいワークのタスク ID (VT_I4)

戻り値 : [out] ワーク情報 (<Pallet>, <Work>, <Value>, <HandID>, <WorkPos> [, HandP3], <Shield>, <Num>, <Pos1>, <Pos2>, <Pos3>, <Pos4>, <Pos5>)
[VT_VARIANT|VT_ARRAY]

<Pallet> : パレット空情報 [VT_I4]

0: 空である, 1: 空でない, 65535: 空判定ができない

<Work>: 把持候補の有無 [VT_I4]

0: 候補なし, 1: 候補あり

<Value>: ワーク評価値 [VT_I4]

<HandID>: 把持情報ID [VT_I4]

<WorkPos>: ワークの位置・姿勢(Tx, Ty, Tz, Rx, Ry, Rz)

[VT_R8|VT_ARRAY]

<HandP3>: ハンドP3の位置・姿勢(*1) (X,Y,Z, Rx, Ry, Rz)

[VT_R8 | VT_ARRAY]

<Shield>: 遮蔽率 [VT_R8]

<Num>: 3次元計測点が局地的に高い位置の個数 [VT_I4]

<Posn(n=1~5)>: 局地的に高い位置 (X, Y, Z)

[VT_R8|VT_ARRAY]

注意)

*1 RC I/F の設定により戻り値の形式が異なります。

<HandP3>は AddController のオプション Position=3 を指定した時に出力します。

使用例

```

Dim vntRet As Object
Dim iTaskID As Integer = 1

m_caoCtrl = cao.AddController("RV", "CaoProv. Canon. RV", "", "conn=eth:192.168.0.1,
Position=3")
vntRet = m_caoCtrl.Execute("PoseNext2", iTaskID)

Dim iPallet As Integer
Dim iWork As Integer
Dim iValue As Integer
Dim iHandID As Integer
Dim vntWorkPos As Object
Dim vntHandP3 as Object
Dim dblShield As Double
Dim iNum As Integer
Dim vntPos(4) As Object
Dim iCnt As Integer

iPallet = vntRet(0)
iWork = vntRet(1)
iValue = vntRet(2)
iHandID = vntRet(3)
vntWorkPos = vntRet(4)
vntHandP3 = vntRet(5)
dblShield = vntRet(6)
iNum = vntRet(7)
For iCnt = 0 To 4
    vntPos(iCnt) = vntRet(8 + iCnt)
Next

```

2.4.1.15. GaoController::Execute (“QueryStatus”) コマンド

RC I/F モジュールから 3D マシンビジョンシステムの状態を問い合わせます。

書式

QueryStatus

引数 : なし

戻り値 : なし

システムが異常の場合のみ HRESULT でエラーコードを返します。エラー内容については 2.3 章を参照してください。

使用例

```
m_caoCtrl.Execute “QueryStatus”
```

2.4.1.16. GaoController::Execute (“ReadyCalib”) コマンド

RV キャリブレーションでマーカークの位置計測を行う予定のシーン数を設定します。

書式

ReadyCalib(<Num>)

引数 : [in]<Num>: RV キャリブレーションでマーカーク計測するシーン数 (VT_I4)

戻り値 : なし

ロボットと 3D マシンビジョンの相対位置・姿勢のキャリブレーションでは、複数位置のロボットに把持させたマーカークの中心座標を画像処理により算出し、それらから座標変換パラメータを算出します。そのため、事前にマーカークの位置計測を行う予定のシーン数を設定し、本コマンドで 3D マシンビジョン認識ソフトウェアに通知します。また、3D マシンビジョン認識ソフトウェアはロボットと 3D マシンビジョンのキャリブレーションに必要な情報を事前に読み込み計測準備を行います。

注意)ロボットを利用して設置後 3D スキャナークャリブレーションを行う場合も本コマンドを使用します。

注意)シーン数は 4 シーン以上設定してください。

使用例

```
Dim iNum as Integer = 5  
m_caoCtrl.Execute “ReadyCalib”, iNum
```

2.4.1.17. GaoController:Execute (“CalibMarker”) コマンド

RV キャリブレーションマーカールの中心座標位置を計測します。



CalibMarker (<Scene>, <Pos>)

: [in]パラメータ [VT_VARAINT | VT_ARRAY]
 <Scene>: シーン番号 [VT_I4]
 <Pos>: マーカールのロボット座標 (Tx, Ty, Tz)
 [VT_R8 | VT_ARRAY]

戻り値 : なし

マーカール計測結果は、

- ・ シーン番号
- ・ ロボット座標 (x,y,z)
- ・ ビジョンが位置計測したマーカールのビジョン座標

をセットで保管します。

既に計測済みのシーン番号に再度計測指定がされた場合、既存データを上書きします。(ユーザーが当該シーンのマーカール計測をやり直したと推定します)

最大シーン番号は ReadyCalib で指定したシーン数です。最大シーン数を超えるシーン番号が指定された場合、エラーとなります。

※ロボットを利用して設置後 3D スキャナーキャリブレーションを行う場合も本コマンドを使用します。



```
Dim vntParam(1) As Object
Dim iNum As Integer = 1
Dim dblPos(2) As Double
dblPos(0) = 10
dblPos(1) = 20
dblPos(2) = 50
vntParam(0) = iNum
vntParam(1) = dblPos
m_caoCtrl.Execute("CalibMarker", vntParam)
```

2.4.1.18. GaoController:Execute (“CatchPoint”) コマンド

ワークの把持位置情報を設定します。



CatchPoint (<TaskID>, <HandID>, <ToolPos>, <AploachPos>, <PickPos>)

引数 : [in] パラメータ

<TaskID>: 計測したいワークのタスク ID [VT_I4]
 <HandID>: 把持情報ID [VT_I4]
 <ToolPos>: ツール座標 (Tx, Ty, Tz, Rx, Ry, Rz)
 [VT_R8 | VT_ARRAY]
 <AploachPos>: アプローチ位置 (Tx, Ty, Tz, Rx, Ry, Rz)
 [VT_R8 | VT_ARRAY]
 <PickPos>: 把持位置 (Tx, Ty, Tz, Rx, Ry, Rz)
 [VT_R8 | VT_ARRAY]

戻り値 : なし

ロボットコントローラよりワーク把持教示の教示結果として、下記データが通知されます。

- ・ アプローチの位置・姿勢
- ・ 把持の位置・姿勢
- ・ エンドエフェクタ(フランジ)から見たハンド先端の位置・姿勢
- ・ この把持位置(アプローチ⇒把持)で把持する把持情報の識別子

これらは、タスク ID に紐付けて管理保持されます。

アプローチ、把持、エンドエフェクタ(フランジ)から見たハンド先端の位置・姿勢は、干渉判定で把持位置をロボットが把持可能であるかを判断するために使用されます。

アプローチ位置と把持位置は、ロボット座標系でのハンド先端位置・姿勢です。

エンドエフェクタ(フランジ)からハンド先端の座標(ツール座標)は、ロボット側の校正ツールで事前に校正済みであるものとします。

使用例

```
Dim vntParams(4) As Object
Dim iTaskID As Integer = 1
Dim iHandID As Integer = 5
Dim dblTool(5) As Double
Dim dblApproach(5) As Double
Dim dblPickPos(5) As Double
```

```
dblTool(0) = 10
dblTool(1) = 0
dblTool(2) = 20
dblTool(3) = 0
dblTool(4) = 0
dblTool(5) = 0
```

```
dblApproach(0) = 300
dblApproach(1) = 0
dblApproach(2) = 350
dblApproach(3) = 180
dblApproach(4) = 0
dblApproach(5) = 180
```

```
dblPickPos(0) = 300
dblPickPos(1) = 0
```

```
dblPickPos(2) = 250
dblPickPos(3) = 180
dblPickPos(4) = 0
dblPickPos(5) = 180

vntParams(0) = iTaskID
vntParams(1) = iHandID
vntParams(2) = dblTool
vntParams(3) = dblApproach
vntParams(4) = dblPickPos

m_caoCtrl.Execute("CatchPoint", vntParams)
```

2.4.1.19. GaoController::Execute ("VerVision") コマンド

マシンビジョンシステムのバージョン番号を問い合わせます。



VerVision

引数 : なし

戻り値 : [out] バージョン情報 (<Vision>, <Scanner>)
[VT_BSTR | VT_ARRAY]

<Vision> : マシンビジョンシステムのバージョン番号

AAA.BBB.CCC

AAA: メジャー BBB:マイナー CCC:リビジョン

<Scanner> : 3D スキャナーのバージョン番号

XXXX



```
Dim vntRet As Object
Dim strVision As String
Dim strScanner As String

vntRet = m_caoCtrl.Execute("VerVision")
strVision = vntRet(0)
strScanner = vntRet(1)
```

2.4.1.20. GaoController::Execute ("QueryMode") コマンド

マシンビジョンシステムの実行モードの状態を問い合わせます。



QueryMode

引数	: なし
戻り値	: [out] 実行モード番号 [VT_I4] 0:ランタイム 1:事前セットアップ(RVキャリブレーション) 2:事前セットアップ(RVキャリブレーション確認) 3:事前セットアップ(把持教示) 4:事前セットアップ(その他) 5:事前セットアップ(パレット平面推定)

起動時のモードは 0 あるいは 4 となっています。1, 2, 3, 5 の各モードへは、モード 4 を経て遷移します。(セットアッププログラムとランタイムプログラムは別プロセスで起動しているため。)

使用例

```
Dim iMode As Integer
iMode = m_caoCtrl.Execute("QueryMode")
```

2.4.1.21. CaoController::Execute ("ReadyRim") コマンド

パレット平面推定に使用する計測シーン数を設定します。

書式 ReadyRim(<SceneNum>)

引数 : [in] <SceneNum> ロボットピン指で計測するシーン数 [VT_I4]

戻り値 : なし

パレット平面推定時には、複数位置へのロボットピン指しで得た座標を使用して、パレット上枠の平面パラメータを算出します。そのため、ピン指しの座標の入力数を事前に設定する必要があります。

本コマンドを使用し、3D マシンビジョン認識ソフトウェアに入力する数を通知します。

また、3D マシンビジョン認識ソフトウェアはパレット平面推定に必要な情報を事前に読み込み計測準備を行います。

使用例

```
Dim iSceneNum As Integer = 5
iMode = m_caoCtrl.Execute("SceneNum", iSceneNum)
```

2.4.1.22. GaoController::Execute (“InputRim”) コマンド

パレット平面推定に使用するロボットピン先端の中心座標を入力します。

書式

InputRim(<SceneID>, <RobVec>)

引数 : [in] パラメータ [VT_VARIANT | VT_ARRAY]
 <SceneID>: ロボットピン指で計測するシーン番号 [VT_I4]
 <RobVec>: ピン先端のロボット座標 (Tx, Ty, Tz)
 [VT_R8 | VT_ARRAY]

戻り値 : なし

3D マシンビジョン認識ソフトウェアは、

- ・ シーン番号
- ・ ロボット座標 (x,y,z)

をセットで保管します。

既に入力済みのシーン番号に再度入力指示がされた場合、既存データを上書きします。(ユーザーが当該シーンのピン指し入力をやり直したと推定しています。)

最大シーン番号は ReadyRim で指定したシーン数です。最大シーン数を超えるシーン番号が指定された場合、エラーとなります。

使用例

```
Dim iSceneNum As Integer = 5
iMode = m_caoCtrl.Execute("SceneNum", iSceneNum)
```

2.4.1.23. GaoController::Execute (“ScannerPos”) コマンド

3D スキャナーの位置を設定します。

書式

InputRim(<ScannerID>)

引数 : [in]<ScannerID>: スキャナー位置 ID [VT_I4]

戻り値 : なし

複数の 3D スキャナー(3D マシンビジョンヘッド)位置がある場合に、マシンビジョンシステムに 3D スキャナー位置を指定します。

※本コマンドは、スライダ搭載対応オプション利用時のみ使用可能です。

使用例

```
Dim iScannerID As Integer = 1
m_caoCtrl.Execute("ScannerPos", iScannerID)
```

2.4.1.24. CaoController::Execute ("FreeAll") コマンド

LoadTask でロードされているすべてのタスクを解放します。

書式 FreeAll

引数 : なし

戻り値 : なし

使用例

```
m_caoCtrl.Execute("FreeAll")
```

2.4.1.25. CaoController::Execute ("FreeAllAsync") コマンド

非同期で FreeAll を実行します。

書式 FreeAllAsync

引数 : なし

戻り値 : なし

実行後 "GetCommandResult" コマンドで実行結果を確認してください。 "GetCommandResult" を実行する前に他のコマンドを使用すると、E_COMMAND_EXCUTING のエラーが発生します。

E_TIMEUOT(0x80000900)が発生した場合は、再度 "GetCommandResult" を使用してコマンド実行結果を受信できます。結果が受信できない時は "ClearPacket" コマンドで実行状態を解除してください。

使用例

```
m_caoCtrl.Execute("FreeAllAsync")
m_caoCtrl.Execute("GetCommandResult")
```

2.4.1.26. CaoController::Execute ("ReadyCorrect") コマンド

測距ずれ量から座標変換パラメータの補正値を算出するためのマーカー計測を実施する事前準備として、

マーカークの計測シーン数を通知します。

書式 ReadyCorrect(<SceneNum>)

引数 : [in] <SceneNum>: シーン数 [VT_I4]

戻り値 : なし

使用例

```
m_caoCtrl.Execute("ReadyCorrect", 5)
```

2.4.1.27. CaoController::Execute ("PoseCorrect") コマンド

マーカークの中心位置座標計測を行います。

書式 PoseCorrect(<Scene>)

引数 : [in] <Scene>: シーン番号 [VT_I4]

戻り値 : [out]計測結果 (<MarkerResult>, <CorrectEnable>, <ResidualMax>, <Residual>, <DistanceMax>, <Distance>)
[VT_VARIANT | VT_ARRAY]

< MarkerResult >: 全マーカークの計測完/未完 [VT_I4]

全マーカークが計測完のとき 1

全マーカークの計測が未完またはマーカーク登録時 0

<CorrectEnable>: 補正可能/不可 [VT_I4]

計測完かつ補正可能のとき 1

補正不可または計測未完またはマーカーク登録時 0

<ResidualMax>: 補正残差 長さ最大値 [VT_R8]

<Residual>: 補正残差 (X, Y, Z 方向成分最大値)

[VT_R8 | VT_ARRAY]

<DistanceMax>: 測距ずれ量 長さ最大値 [VT_R8]

<DistanceX>: 測距ずれ量 (X, Y, Z 方向成分最大値)

[VT_R8 | VT_ARRAY]

使用例

```

Dim vntRet as Object

vntRet = m_caoCtrl.Execute("PoseCorrect", 2)

Dim iMarkerResult as Integer
Dim iCorrectEnable as Integer
Dim dResidualMax as Double
Dim vntResidual as Object
Dim dDistance as Double
Dim vntDistance as Object

iMarkerResult = vntRet(0)
iCorrectEnable = vntRet(1)
dResidualMax = vntRet(2)
vntResidual = vntRet(3)
dDistance = vntRet(4)
vntDistance = vntRet(5)

```

2.4.1.28. CaoController:Execute ("SaveImage") コマンド

直近の撮影画像を保存します。

このとき、撮影画像は指定されたフォルダに.cdm3 という拡張子のファイル形式で保存されます。



SaveImage

引数 : なし

戻り値 : ファイル名称が一つの場合
[out] <Name> ファイル名称 [VT_BSTR]

ファイル名称が二つの場合

[out] ファイル名称リスト (<Name1>,<Name2>) [VT_BSTR | VT_ARRAY]

<Name1>: ファイル名称 1 [VT_BSTR]

<Name2>: ファイル名称 2 [VT_BSTR]

撮影未実施時、もしくは撮影は完了しているものの撮像ファイル保存に失敗した場合には、ファイル名称には 0 が入ります。



```

Dim vntName as Object
vntName = m_caoCtrl.Execute("SaveImage")

```

```

Dim strFileName1 As String
Dim strFileName2 As String

'-----ファイル名称が一つの場合-----
strFileName1 = vntName

'-----ファイル名称が二つの場合-----
strFileName1 = vntName(0)
strFileName2 = vntName(1)

```

2.4.1.29. CaoController::Execute (“InputPin”) コマンド

システム上向きを表すベクトルを教示するためのロボットピン先端の中心座標の入力を行います。

書式 InputPin (<Scene>, <RobPos>)

引数 : [in] パラメータ [VT_VARIANT | VT_ARRAY]
 <SceneNum>: シーン番号[VT_I4]
 <RobPos>: ピン先端のロボット座標 (X, Y, Z)
 [VT_R8 | VT_ARRAY]

戻り値 : なし

使用例

```

Dim vntParams(2) as Object
Dim iScene as integer
Dim dPos(3) as Double

iScene = 5
dPos(0) = 10.0
dPos(1) = 20.0
dPos(2) = 30.0

vntParams(0) = iScene
vntParams(1) = dPos

m_caoCtrl.Execute("InputPin", vntParams)

```

2.4.1.30. CaoController::Execute (“GetScannerInfo”) コマンド

接続されている 3D スキャナーの情報(型式, シリアル ID, 通信部のファームウェアバージョン, FPGA のファームウェアバージョン, 投影用デバイスのファームウェアバージョン)を取得します。

書式 GetScannerInfo

引数 : なし

戻り値 : [out] 3D スキャナーの情報
 (<TypeName>,<SerialID>,<CommFirmware>,
 <FpgaFirmware>,<DeviceFirmware>)[VT_BSTR | VT_ARRAY]

<TypeName >:型式 [VT_BSTR]
 <SerialID >: シリアル ID[VT_BSTR]
 <CommFirmware>: 通信部のファームウェアバージョン[VT_BSTR]
 <FpgaFirmware >: FPGA のファームウェアバージョン [VT_BSTR]
 <DeviceFirmware>: 投影用デバイスのファームウェアバージョン
 [VT_BSTR]

使用例

```
Dim vntRet as Object
vntRet = m_caoCtrl.Execute("GetScannerInfo")

Dim bstrTypeName as String
Dim bstrSerialID as String
Dim bstrCommFirmware as String
Dim bstrFpgaFirmware as String
Dim bstrDeviceFirmware as String

bstrTypeName = vntRet(0)
bstrSerialID = vntRet(1)
bstrCommFirmware = vntRet(2)
bstrFpgaFirmware = vntRet(3)
bstrDeviceFirmware = vntRet(4)
```

2.4.1.31. CaoController::Execute("QueryGraspID")コマンド

タスクに登録されている把持 ID 数と、各把持 ID のロボットコントローラとの同期状態の情報を取得します。

書式

QueryGraspID(<TaskID>)

引数 : [in]<TaskID>: タスク ID [VT_I4]
 戻り値 : [out] 把持情報 (<HandInfoNum> ,[<HandIDN >,<SynsStateN>))
 [VT_I4 | VT_ARRAY]

<HandInfoNum >:タスク内の全把持情報の数 [VT_I4]

<HandIDN (N= 1~HandInfoNum)>把持 IDN の ID [VT_I4]

<SynsStateN (N= 1~HandInfoNum)>把持 IDN の同期状態
[VT_I4]

使用例

```
Dim vntRet as Object
Dim iTaskID As Integer = 1
vntRet = m_caoCtrl.Execute("QueryGraspID" , iTaskID)

'-----タスク内の全把持情報の数が 2 個の場合
Dim iHandInfoNum As Integer
Dim iHandID(1) As Integer
Dim iSynsState(1) As Integer

iHandInfoNum = vntRet(0)
vntHandInfo(0) = vntRet(1)
vntSynsState(0) = vntRet(2)
vntHandInfo(1) = vntRet(3)
vntSynsState(1) = vntRet(4)
```

2.4.1.32. GaoController::Execute("SyncCatchPoint")コマンド

ロボット座標系のワークの位置・姿勢と、アプローチ状態でのハンド先端の位置・姿勢、把持状態でのハンド先端の位置・姿勢を取得します..

書式

SyncCatchPoint(<TaskID>, <HandInfoID>, <ToolPos>, <UpdateFlag>)

引数 : [in] パラメータ [VT_VARIANT | VT_ARRAY]

<TaskID>: タスク ID[VT_I4]

<HandInfoID>: 把持 ID[VT_I4]

<ToolPos>: フランジからツール先端の座標・回転角度

(Tx, Ty, Tz, Rx, Ry, Rz)

[VT_R8 | VT_ARRAY]

<UpdateFlag>: 更新フラグ[VT_I4]

戻り値 : [out] 位置・姿勢情報 (<WorkPos>, <ApproachPos>, <PickPos>)
[VT_VARIANT | VT_ARRAY]

<WorkPos>: ロボット座標系におけるワークの座標 (Tx, Ty, Tz, Rx, Ry, Rz)

[VT_R8 | VT_ARRAY]

<ApproachPos>: アプローチ状態でのロボット座標系におけるツール
位置・回転角度 (Tx, Ty, Tz, Rx, Ry, Rz)

[VT_R8 | VT_ARRAY]

<PickPos>: 把持状態でのロボット座標系におけるツールの座標・
回転角度 (Tx, Ty, Tz, Rx, Ry, Rz)

[VT_R8 | VT_ARRAY]

使用例

```
Dim vntRet As Object
Dim iTaskID As Integer = 1
Dim iHandInfoID As Integer = 5
Dim dblToolPos(5) As Double
dblToolPos(0) = 0
dblToolPos(1) = 0
dblToolPos(2) = 112
dblToolPos(3) = 0
dblToolPos(4) = 0
dblToolPos(5) = 0

Dim iUpdateFlag As Integer = 0

Dim vntTool As Object
Dim vntApproach As Object
Dim vntPickPos As Object
vntRet = m_caoCtrl.Execute("SyncCatchPoint", iTaskID, iHandInfoID, dblToolPos, iUpdateFlag)

vntTool = vntRet(0)
vntApproach = vntRet(1)
vntPickPos = vntRet(2)
```

2.4.1.33. CaoController:Execute("PoseWork3")コマンド

把持可能なワークの位置・姿勢を計測します。

ワーク計測に関するパラメータについて、本コマンドで指定することで設定値を変更することができます。
パラメータ省略時には変更されません。

取得する位置姿勢情報の対象について、本コマンドで指定することで対象を変更することができます。

書式 PoseWork3(<TaskID>, <Mode> [, <ROISize>, <ROIx>, <ROIy>, <Score>, <WorkMax>,
<FitMax>, <CatchMax>, <Exposure>])

引数 : [in] パラメータ

<TaskID>: 計測したいワークのタスク ID [VT_I4]

<Mode>: 位置姿勢返却モード(*4) [VT_I4]

<ROISize>: ROI(部品検出領域)サイズ倍率 [VT_I4]

<ROIX>: ROI の中心位置 X, 1~2048 [VT_I4]
 <ROIY>: ROI の中心位置 Y, 1~2048 [VT_I4]
 <Score>: 認識スコアの閾値, 1.0~100.0 [VT_R8]
 <WorkMax>: 概略検出ワーク上限数, 1~100 [VT_I4]
 <FitMax>: モデルフィッティングワーク上限数, 1~20 [VT_I4]
 <CatchMax>: 把持候補出力上限数, 1~20 [VT_I4]
 <Exposure>: 露光時間 [VT_R8]
 0.05~54.0 の範囲で設定可能 (RV1100)
 0.05~47.0 の範囲で設定可能 (RV500, RV300)

戻り値 : [out] ワーク情報 (<Pallet>, <Work>, <Value>, <HandID>, <WorkPos>, <Shield>, <Num>[, <Posn>] [, <StartTime>, <EndTime>]) [VT_VARIANT|VT_ARRAY]

<Pallet>: パレット空情報 [VT_I4]
 0: 空である, 1: 空でない, 65535: 空判定ができない
 <Work>: 把持候補の有無 [VT_I4]
 0: 候補なし, 1: 候補あり
 <Value>: ワーク評価値 [VT_I4]
 <HandID>: 把持情報ID [VT_I4]
 <WorkPosN(N = 1 ~ 4)>: ワーク又はハンドの位置・姿勢(X, Y, Z, Rx, Ry, Rz) (*4) [VT_R8|VT_ARRAY]
 <Shield>: 遮蔽率(*1) [VT_R8]
 <Num>: 3次元計測点が局地的に高い位置の個数 [VT_I4]
 <Posn(n=1~5)>: 局地的に高い位置 (X, Y, Z)(*2) [VT_R8|VT_ARRAY]
 <StartTime>: 撮影開始時刻(*3) [VT_BSTR]
 <EndTime>: 撮影終了時刻(*3) [VT_BSTR]

計測対象のワークは、タスク ID で指定されたタスクに属するワークです。

注意)

*1 <Shield>は実数の値を返します。例)3.8% ⇒ 3.8 99.9% ⇒ 99.9

*2 <Posn>は結果取得時の<Num>の値により出力の個数が変化します。

*3 RC I/F の設定により戻り値の形式が異なります。

<StartTime>,<EndTime>は AddController のオプション Timestamp=True を指定した時に出力します。

*4 <WrokPosN(N = 1~4)>の個数 N は送信時の<Mode>の値によって増減します。

<Mode> = 0 :使用不可

<Mode> = 1,2,3 : N = 1
 <Mode> = 4,5,6 : N = 2
 <Mode> = 7,8 : N = 3
 <Mode> = 9 : N = 4

使用例

```
Dim vntRet As Object

Dim iTaskID As Integer = 101
Dim iMode As Integer = 1

Dim iPallet As Integer
Dim iWork As Integer
Dim iValue As Integer
Dim iHandID As Integer
Dim vntWorkHand As Object
Dim dblShield As Double
Dim iNum As Integer
Dim vntPos As Object
Dim strStartTime as String
Dim strEndTime as String

' ----- パラメータ省略時 -----
m_caoCtrl = cao.AddController ("RV", "caoProv. Canon. RV", "", "conn=eth:192.168.0.1")
vntRet = m_caoCtrl.Execute("PoseWork3", iTaskID, iMode)

iPallet = vntRet(0)
iWork = vntRet(1)
iValue = vntRet(2)
iHandID = vntRet(3)
vntWorkPos = vntRet(4)
dblShield = vntRet(5)
iNum = vntRet(6)
vntPos = vntRet(7)

' ----- パラメータ設定時 -----
Dim vntParam(9) As Object
vntParam(0) = iTaskID
vntParam(1) = iMode
vntParam(2) = 2 ' Size
vntParam(3) = 512 ' ROI_X
vntParam(4) = 512 ' ROI_Y
vntParam(5) = 88.8 ' Score
vntParam(6) = 50 ' WorkMax
vntParam(7) = 10 ' FitMax
vntParam(8) = 8 ' CatchMax
vntParam(9) = 40.4 ' Exposure

m_caoCtrl = cao.AddController ("RV", "caoProv. Canon. RV", "", "conn=eth:192.168.0.1")

vntRet = m_caoCtrl.Execute("PoseWork3", vntParam)

iPallet = vntRet(0)
iWork = vntRet(1)
iValue = vntRet(2)
iHandID = vntRet(3)
vntWorkPos = vntRet(4)
dblShield = vntRet(5)
```

```

iNum = vntRet (6)
vntPos = vntRet (7)

' ----- RC I/F で撮影時刻を出力している時 -----
m_caoCtrl = cao.AddController ("RV", "caoProv.Canon.RV", "", "conn=eth:192.168.0.1,
Timestamp=true")
vntRet = m_caoCtrl.Execute ("PoseWork3", iTaskID, iMode)

iPallet = vntRet (0)
iWork = vntRet (1)
iValue = vntRet (2)
iHandID = vntRet (3)
vntWorkPos = vntRet (4)
dblShield = vntRet (5)
iNum = vntRet (6)
vntPos = vntRet (7)
strStartTime = vntRet (8)
strEndTime = vntRet (9)

```

2.4.1.34. CaoController::Execute("PoseWork3Async")コマンド

非同期で PoseWork3 を実行します。

書式 PoseWork3Async(<TaskID>, <Mode> [, <ROISize>, <ROI X>, <ROI Y>, <Score>, <WorkMax>, <FitMax>, <CatchMax>, <Exposure>])

引数 : [in] パラメータ

- <TaskID>: 計測したいワークのタスク ID [VT_I4]
- <Mode>: 位置姿勢返却モード [VT_I4]
- <ROISize>: ROI (部品検出領域) サイズ倍率 [VT_I4]
- <ROI X>: ROI の中心位置 X, 1~2048 [VT_I4]
- <ROI Y>: ROI の中心位置 Y, 1~2048 [VT_I4]
- <Score>: 認識スコアの閾値, 1.0~100.0 [VT_R8]
- <WorkMax>: 概略検出ワーク上限数, 1~100 [VT_I4]
- <FitMax>: モデルフィッティングワーク上限数, 1~20 [VT_I4]
- <CatchMax>: 把持候補出力上限数, 1~20 [VT_I4]
- <Exposure>: 露光時間 [VT_R8]
0.05~54.0 の範囲で設定可能 (RV1100)
0.05~47.0 の範囲で設定可能 (RV500, RV300)

戻り値 : なし

実行後 "GetCommandResult" コマンドで実行結果を確認してください。 "GetCommandResult" を実行する前

に他のコマンドを使用すると、E_COMMAND_EXECUTING のエラーが発生します。

E_TIMEUOT(0x80000900)が発生した場合は、再度"GetCommandResult"を使用してコマンド実行結果を受信できます。結果が受信できない時は"ClearPacket"コマンドで実行状態を解除してください。

使用例

```
Dim vntRet As Object

Dim iTaskID As Integer = 101
Dim iMode As Integer = 1

Dim vntParam(9) As Object
vntParam(0) = iTaskID
vntParam(1) = iMode
vntParam(2) = 2 ' Size
vntParam(3) = 512 ' ROI_X
vntParam(4) = 512 ' ROI_Y
vntParam(5) = 88.8 ' Score
vntParam(6) = 50 ' WorkMax
vntParam(7) = 10 ' FitMax
vntParam(8) = 8 ' CatchMax
vntParam(9) = 40.4 ' Exposure

m_caoCtrl = cao.AddController ("RV", "caoProv. Canon. RV", "", "conn=eth:192.168.0.1")

m_caoCtrl.Execute("PoseWork3Async", vntParam)

vntRet = caoCtrl.Execute("GetCommandResult")

iPallet = vntRet(0)
iWork = vntRet(1)
iValue = vntRet(2)
iHandID = vntRet(3)
vntWorkPos = vntRet(4)
dblShield = vntRet(5)
iNum = vntRet(6)
vntPos = vntRet(7)
```

2.4.1.35. CaoController::Execute("PoseNext3")コマンド

新たな計測はせず、直近の計測で算出した次点の把持可能ワークの位置・姿勢、もしくはハンドの把持位置・姿勢について返します。

取得する位置姿勢情報の対象について、本コマンドで指定することで対象を変更することができます。

先に POSE_WORK3 で受け取った把持候補ワークをロボット側の理由で把持不可能と判断した場合に、ワーク山は崩れていないと判断し、新たに計測せずに直近の計測での次点(把持優先順の次点)の把持可能ワークの位置・姿勢、もしくはハンドの把持位置・姿勢を返します。



PoseNext3 (<TaskID>, <Mode>)

<TaskID> : [in] パラメータ

<TaskID>: 計測したいワークのタスク ID [VT_I4]

<Mode>: 位置姿勢返却モード(*3) [VT_I4]

戻り値 : [out] ワーク情報 (<Pallet>, <Work>, <Value>, <HandID>, <WorkPos>, <Shield>, <Num>[, <Posn>])
[VT_VARIANT|VT_ARRAY]

<Pallet> : パレット空情報 [VT_I4]
0 : 空である, 1 : 空でない, 65535 : 空判定ができない

<Work> : 把持候補の有無 [VT_I4]
0 : 候補なし, 1 : 候補あり

<Value> : ワーク評価値 [VT_I4]

<HandID> : 把持情報ID [VT_I4]

<WorkPosN(N = 1~4)> : ワーク又はハンドの位置・姿勢(X, Y, Z, Rx, Ry, Rz) (*3) [VT_R8|VT_ARRAY]

<Shield> : 遮蔽率(*1) [VT_R8]

<Num> : 3次元計測点が局地的に高い位置の個数 [VT_I4]

<Posn(n=1~5)> : 局地的に高い位置 (X, Y, Z) (*2)
[VT_R8|VT_ARRAY]

注意)

*1 <Shield>は実数の値を返します。例)3.8% ⇒ 3.8 99.9% ⇒ 99.9

*2 <Posn>は結果取得時の<Num>の値により出力の個数が変化します。

*3 <WrokPosN(N = 1~4)>の個数 N は送信時の<Mode>の値によって増減します。

<Mode> = 0 : 使用不可
<Mode> = 1,2,3 : N = 1
<Mode> = 4,5,6 : N = 2
<Mode> = 7,8 : N = 3
<Mode> = 9 : N = 4

使用例

```
Dim vntRet As Object
Dim iTaskID As Integer = 101
Dim iMode As Integer = 1

m_caoCtrl = cao.AddController("RV", "CaoProv. Canon. RV", "", "conn=eth:192.168.0.1,
Position=3")
vntRet = m_caoCtrl.Execute("PoseNext3", iTaskID, iMode)

Dim iPallet As Integer
Dim iWork As Integer
```

```
Dim iValue As Integer
Dim iHandID As Integer
Dim vntWorkPos As Object
Dim dblShield As Double
Dim iNum As Integer
Dim vntPos As Object
```

```
iPallet = vntRet(0)
iWork = vntRet(1)
iValue = vntRet(2)
iHandID = vntRet(3)
vntWorkPos = vntRet(4)
dblShield = vntRet(5)
iNum = vntRet(6)
vntPos = vntRet(7)
```

2.4.1.36. GaoController:Execute(“PoseMultiWork3”)コマンド

把持可能なワークの位置・姿勢を計測し、全ての候補を返却します。

計測対象のワークは、タスク ID で指定されたタスクに属するワークです。探索する ROI は、中心位置と共に複数個指定することができます。

ワークの最大個数に達しない場合でも探索する ROI 数の探索が完了すると、応答を返します。

取得する位置姿勢情報の対象について、本コマンドで指定することで対象を変更することができます。

ワーク計測に関するパラメータについて、本コマンドで指定することで設定値を変更することができます。

書式 PoseMultiWork3(<TaskID>, <WorkMaxCnt>, <ROICnt>, <Mode>, [, <ROISize>, <ROI X>, <ROI Y>, <Score>, <WorkMax>, <FitMax>, <CatchMax>, <Exposure>])

引数 : [in] パラメータ

<TaskID>: 計測したいワークのタスク ID [VT_I4]

<WorkMaxCnt>: ワークの最大個数 [VT_I4]

<ROICnt>: 探索する ROI 数 [VT_I4]

<Mode>: 位置姿勢返却モード(*5) [VT_I4]

<ROISize>: ROI(部品検出領域)サイズ倍率 [VT_I4]

<ROI X>: ROI の中心位置 X, 1~2048 [VT_I4]

<ROI Y>: ROI の中心位置 Y, 1~2048 [VT_I4]

<Score>: 認識スコアの閾値, 1.0~100.0 [VT_R8]

<WorkMax>: 概略検出ワーク上限数, 1~100 [VT_I4]

<FitMax>: モデルフィッティングワーク上限数, 1~20 [VT_I4]

<CatchMax>: 把持候補出力上限数, 1~20 [VT_I4]

<Exposure>: 露光時間 [VT_R8]

0.05~54.0 の範囲で設定可能 (RV1100)

0.05~47.0 の範囲で設定可能 (RV500, RV300)

戻り値 : [out] ワーク情報 (<Pallet>, <WorkNum>[, <Value>, <HandID>, <WorkPos>, <Shield>], <Num>[, <Posn>] [, <StartTime>, <EndTime>]) [VT_VARIANT|VT_ARRAY]

<Pallet> : パレット空情報 [VT_I4]
 0 : 空である, 1 : 空でない, 65535 : 空判定ができない

<WorkNum> : 把持候補の数 [VT_I4]

<Value> : ワーク評価値 [VT_I4>(*2)

<HandID> : 把持情報ID [VT_I4] (*2)

<WorkPosN(N = 1~4)> : ワーク又はハンドの位置・姿勢(X, Y, Z, Rx, Ry, Rz) (*5) [VT_R8|VT_ARRAY] (*2)

<Shield> : 遮蔽率(*1) [VT_R8] (*2)

<Num> : 3次元計測点が局地的に高い位置の個数 [VT_I4]

<Posn(n=1~5)> : 局地的に高い位置 (X, Y, Z)(*3) [VT_R8|VT_ARRAY]

<StartTime>: 撮影開始時刻(*4) [VT_BSTR]

<EndTime>: 撮影終了時刻(*4) [VT_BSTR]

計測対象のワークは、タスク ID で指定されたタスクに属するワークです。

注意)

*1 <Shield>は実数の値を返します。例)3.8% ⇒ 3.8 99.9% ⇒ 99.9

*2<Value>,<HandID>,<WorkPos>,<Shield>は、結果取得時の<WorkNum>の値により出力の個数が変化します。

<WorkNum>の値が 0 の場合は戻り値として取得できません。

*3<Posn>は結果取得時の<Num>の値により出力の個数が変化します。

*4 RC I/F の設定により戻り値の形式が異なります。

<StartTime>,<EndTime>は AddController のオプション Timestamp=True を指定した時に出力します。

*5 <WrokPosN(N = 1~4)>の個数 N は送信時の<Mode>の値によって増減します。

<Mode> = 0	:使用不可
<Mode> = 1,2,3	: N = 1
<Mode> = 4,5,6	: N = 2
<Mode> = 7,8	: N = 3
<Mode> = 9	: N = 4

使用例

```
Dim vntRet As Object
Dim iCnt As Integer

Dim iTaskID As Integer = 101
Dim iWorkNum As Integer = 1
Dim iROICnt As Integer = 1
Dim iMode As Integer = 1

Dim iPallet As Integer
Dim iWorkNum As Integer
Dim iNum As Integer
Dim vntPos As Object
Dim strStartTime as String
Dim strEndTime as String

' ----- パラメータ省略時 -----
m_caoCtrl = cao.AddController "RV", "caoProv. Canon. RV", "", "conn=eth:192.168.0.1")
vntRet = m_caoCtrl.Execute("PoseMultiWork3", iTaskID, iMode)

iPallet = vntRet(0)
iWorkNum = vntRet(1)

If iWorkNum = 0
    vntPos = vntRet(2)
    iNum = vntRet(3)
    If iNum <> 0
        vntPos = vntRet(4)
    Endif
Else
    Dim vntValue(iWorkNum) As Object
    Dim vntHandID(iWorkNum) As Object
    Dim vntWorkPos(iWorkNum) As Object
    Dim vntShield(iWorkNum) As Object

    For iCnt 0 to iWorkNum
        vntValue(iCnt) = vntRet(2 + (iCnt * 4))
        vntHandID(iCnt) = vntRet(3 + (iCnt * 4))
        vntWorkPos(iCnt) = vntRet(4 + (iCnt * 4))
        vntShield(iCnt) = vntRet(5 + (iCnt * 4))
    Next
    iNum = vntRet(6 + (iWorkNum * 4))
    vntPos = vntRet(7 + (iWorkNum * 4))
Endif

' ----- パラメータ設定時 -----
Dim vntParam(11) As Object
vntParam(0) = iTaskID
vntParam(1) = iWorkMaxCnt
vntParam(2) = iROICnt
vntParam(3) = iMode
vntParam(4) = 2 ' Size
vntParam(5) = 512 ' ROIx
vntParam(6) = 512 ' ROIy
vntParam(7) = 88.8 ' Score
vntParam(8) = 50 ' WorkMax
vntParam(9) = 10 ' FitMax
vntParam(10) = 8 ' CatchMax
vntParam(11) = 40.4 ' Exposure

m_caoCtrl = cao.AddController ("RV", "caoProv. Canon. RV", "", "conn=eth:192.168.0.1")
vntRet = m_caoCtrl.Execute("PoseMultiWork3", vntParam)
```

```
If iWorkNum = 0
    vntPos = vntRet(2)
    iNum = vntRet(3)
    If iNum <> 0
        vntPos = vntRet(4)
    Endif
Else
    Dim vntValue(iWorkNum) As Object
    Dim vntHandID(iWorkNum) As Object
    Dim vntWorkPos(iWorkNum) As Object
    Dim vntShield(iWorkNum) As Object

    For iCnt 0 to iWorkNum
        vntValue(iCnt) = vntRet(2 + (iCnt * 4))
        vntHandID(iCnt) = vntRet(3 + (iCnt * 4))
        vntWorkPos(iCnt) = vntRet(4 + (iCnt * 4))
        vntShield(iCnt) = vntRet(5 + (iCnt * 4))
    Next
    iNum = vntRet(6 + (iWorkNum * 4))
    vntPos = vntRet(7 + (iWorkNum * 4))
Endif

' ----- RC I/F で撮影時刻を出力している時 -----
m_caoCtrl = cao.AddController("RV", "caoProv.Canon.RV", "", "conn=eth:192.168.0.1,
Timestamp=true")
vntRet = m_caoCtrl.Execute("PoseMultiWork3", iTaskID, iMode)

If iWorkNum = 0
    vntPos = vntRet(2)
    iNum = vntRet(3)
    If iNum = 0
        strStartTime = vntRet(4)
        strEndTime = vntRet(5)
    Else
        vntPos = vntRet(4)
        strStartTime = vntRet(5)
        strEndTime = vntRet(6)
    Endif
Else
    Dim vntValue(iWorkNum) As Object
    Dim vntHandID(iWorkNum) As Object
    Dim vntWorkHand(iWorkNum) As Object
    Dim vntShield(iWorkNum) As Object

    For iCnt 0 to iWorkNum
        vntValue(iCnt) = vntRet(2 + (iCnt * 4))
        vntHandID(iCnt) = vntRet(3 + (iCnt * 4))
        vntWorkHandPos(iCnt) = vntRet(4 + (iCnt * 4))
        vntShield(iCnt) = vntRet(5 + (iCnt * 4))
    Next
    iNum = vntRet(6 + (iWorkNum * 4))
    vntPos = vntRet(7 + (iWorkNum * 4))
    strStartTime = vntRet(8 + (iWorkNum * 4))
    strEndTime = vntRet(9 + (iWorkNum * 4))
Endif
```



```

Dim vntParam(11) As Object

vntParam(0) = iTASKID
vntParam(1) = iWorkMaxCnt
vntParam(2) = iROICnt
vntParam(3) = iMode
vntParam(4) = 2 ' Size
vntParam(5) = 512 ' ROI X
vntParam(6) = 512 ' ROI Y
vntParam(7) = 88.8 ' Score
vntParam(8) = 50 ' WorkMax
vntParam(9) = 10 ' FitMax
vntParam(10) = 8 ' CatchMax
vntParam(11) = 40.4 ' Exposure

m_caoCtrl = cao.AddController ("RV", "caoProv. Canon. RV", "", "conn=eth:192.168.0.1")
m_caoCtrl.Execute("PoseMultiWork3Async", vntParam)

vntRet = caoCtrl.Execute("GetCommandResult")

If iWorkNum = 0
    vntPos = vntRet(2)
    iNum = vntRet(3)
    If iNum <> 0
        vntPos = vntRet(4)
    Endif
Else
    Dim vntValue(iWorkNum) As Object
    Dim vntHandID(iWorkNum) As Object
    Dim vntWorkHand(iWorkNum) As Object
    Dim vntShield(iWorkNum) As Object

    For iCnt 0 to iWorkNum
        vntValue(iCnt) = vntRet(2 + (iCnt * 4))
        vntHandID(iCnt) = vntRet(3 + (iCnt * 4))
        vntWorkHandPos(iCnt) = vntRet(4 + (iCnt * 4))
        vntShield(iCnt) = vntRet(5 + (iCnt * 4))
    Next
    iNum = vntRet(6 + (iWorkNum * 4))
    vntPos = vntRet(7 + (iWorkNum * 4))
Endif

```

2.4.1.38. CaoController::Execute("GetTime")コマンド

撮影時間とタクトタイムを取得します。



GetTime(<TaskID>)

引数 : [in]<TaskID>: タスク ID [VT_I4]
 戻り値 : [out] 時間情報 (<PhotoTime> ,<TactTime>)
 [VT_R8 | VT_ARRAY]

<PhotoTime>: 撮影時間[VT_R8]

<TactTime>: タクトタイム[VT_R8]

使用例

```

Dim vntRet As Object
Dim iTaskID As Integer = 1
vntRet = m_caoCtrl.Execute("GetTime", iTaskID)

Dim dblPhotoTime As Double
Dim dblTactTime As Double

dblPhotoTime = vntRet(0)
dblTactTime = vntRet(1)

```

2.4.1.39. CaoController::Execute("CaptureImage")コマンド

画像を撮影し、ビットマップ画像形式で取得します。

書式

CaptureImage(<Mode>, <Param>)

引数 : [in] パラメータ [VT_VARIANT | VT_ARRAY]
 <Mode>: パラメーター指定モード [VT_I4]
 <Param>: 指定パラメーター(*1)
 Mode = 0: <TaskID>: タスク ID [VT_I4]
 Mode = 1: <Exposure>: 露光時間 [msec] [VT_R8]

戻り値 : [out] ビットマップ画像情報 [VT_UI1 | VT_ARRAY]

*1 第一引数<Mode>の値によって第二引数<Param>の型と意味が変更されます。

使用例

```

Dim vntRet As Object
Dim iMode As Integer = 1
Dim iTaskID As Integer = 10
Dim dblExposure As Double = 30.0

'-----タスク指定モードの場合-----
vntRet = m_caoCtrl.Execute("CaptureImage", iMode, iTaskID)

'-----露光時間指定モードの場合-----
vntRet = m_caoCtrl.Execute("CaptureImage", iMode, dblExposure)

```

2.4.2. 独自コマンド

2.4.2.1. GaoController::Execute (“ExecuteCommand”) コマンド

RC I/F モジュールにコマンド文字列を送信して、命令を実行します。

書式 ExecuteCommand(<Command>)

引数 : [in]<Command>: コマンド文字列 [VT_BSTR]

戻り値 : [out] コマンド応答文字列 [VT_BSTR]

使用例

```
Dim strRet as String
strRet = m_caoCtrl.Execute("ExecuteCommand", "LOAD_TASK,1")
```

2.4.2.2. GaoController::Execute (“ExecuteCommandAsync”) コマンド

非同期で ExecuteCommand を実行します。

書式 ExecuteCommandAsync(<Command>)

引数 : [in]<Command>: コマンド文字列 [VT_BSTR]

戻り値 : [out] コマンド応答文字列 [VT_BSTR]

実行後"GetCommandResult"コマンドで実行結果を確認してください。"GetCommandResult"を実行する前に他のコマンドを使用すると、E_COMMAND_EXCUTING のエラーが発生します。

E_TIMEUOT(0x80000900)が発生した場合は、再度"GetCommandResult"を使用してコマンド実行結果を受信できます。結果が受信できない時は"ClearPacket"コマンドで実行状態を解除してください。

使用例

```
Dim strRet as String
m_caoCtrl.Execute("ExecuteCommand", "LOAD_TASK,1")
strRet = m_caoCtrl.Execute("GetCommandResult")
```

2.4.2.3. GaoController::Execute (“ClearPacket”) コマンド

受信したパケットをクリアします。

非同期コマンドを実行していた場合、実行状態を解除します。

書式 ClearPacket

引数 : なし

戻り値 : なし

使用例

```
m_caoCtrl.Execute("ClearPacket")
```

2.4.2.4. CaoController::Execute ("SetTimeout") コマンド

Timeout 時間を設定します。

書式

SetTimeout (<Timeout>)

引数 : [in]<Timeout>: タイムアウト時間 msec [VT_I4]

戻り値 : なし

使用例

```
m_caoCtrl.Execute("SetTimeout", 1000)
```

2.4.2.5. CaoController::Execute ("GetTimeout") コマンド

設定されている Timeout 時間を取得します。

書式

GetTimeout

引数 : なし

戻り値 : [out] タイムアウト時間 msec [VT_I4]

使用例

```
Dim iTimeout as integer  
iTimeout = m_caoCtrl.Execute("GetTimeout")
```

2.4.2.6. GaoController::Execute (“GetCommandResult”) コマンド

非同期で実行したコマンドの実行結果を取得します。

書式

GetTimeout

引数 : なし

戻り値 : [out] 非同期で実行したコマンドの戻り値

戻り値は直前に実行した非同期コマンドに依存します。

使用例

```
Dim vntPoseData as Object  
  
m_caoCtrl.Execute (“PoseWorkAsync”, 1)  
vntPoseData = m_caoCtrl.Execute (“GetCommandResult”)
```
