

IV プロバイダ

SHARP IV シリーズ用プロバイダ

ユーザーズ ガイド

Version 1.0.2

March 12, 2019

【備考】

【改版履歴】

日付	版数	内容
2013-02-01	1.0.0	初版.
2016-12-16	1.0.1	IV-S300X,IV-S310X 対応 GetRegData, PutRegData, PutDateString, GetModuleString, GetBlockString, PutBlockString, Put20BlockString, Put10BlockString, SysSettingSave, SelfCheck, ClearFigures 追加 PutKind, PutViewMode, PutDispMode, RegStdImage 修正
2017-11-28	1.0.2	・ IV-S301M 対応 SAlignmentTrigger, SAlignmentCalibration, RobotCalibration 追加 ・ 記載方針変更に伴う修正 ChangeTimeout, TriggerAndWait, Trigger, GetData, SerialEnable, RemoteEnable, ViewLockEnable, PutKind, PutModule, GetMeasureData, PutMeasureData, PutPassword, GetVal, PutVal, RegStdImage, GetShutterSp, PutShutterSp, GetThreshold, PutThreshold, GetGain, PutGain, PutDate, PutRegData, GetRegData, PutDateString, GetModuleString, GetBlockString, PutBlockString, Put20BlockString, Put10BlockString, GetBrightness, GetParallel 修正
2018-08-24	1.0.2	誤植修正 E_RESPONSE_AREA_FAILED : 0x80100004 → 0x80100005
2019-03-12	1.0.2	・ IV-S402M,IV-S412M 対応

【対応機器】

以下の機種と制御用通信コマンドが同じ機種については、本プロバイダから制御できます。
各対応機種がサポートするコマンド一覧は、「3 コマンドリファレンス」を参照ください。

機種	バージョン	注意事項
----	-------	------

IV-S150X		
IV-S200X		
IV-S210X		
IV-C250X		
IV-S300X		
IV-S310X		
IV-S301M		
IV-S402M		
IV-S412M		

目次

1. はじめに	6
2. プロバイダの概要	7
2.1. 概要	7
2.2. メソッド・プロパティ	8
2.2.1. CaoWorkspace::AddController メソッド	8
2.2.2. CaoController::Execute メソッド	9
2.3. エラーコード	10
3. コマンドリファレンス	11
3.1. Execute コマンド	15
3.1.1. CaoController::Execute("Raw") コマンド	15
3.1.2. CaoController::Execute("ChangeTimeout") コマンド	15
3.1.3. CaoController::Execute("TriggerAndWait") コマンド	15
3.1.4. CaoController::Execute("Trigger") コマンド	15
3.1.5. CaoController::Execute("GetData") コマンド	15
3.1.6. CaoController::Execute("RobotCalibration") コマンド	16
3.1.7. CaoController::Execute("SerialEnable") コマンド	16
3.1.8. CaoController::Execute("SAlignmentTrigger") コマンド	16
3.1.9. CaoController::Execute("SAlignmentCalibration") コマンド	16
3.1.10. CaoController::Execute("RemoteEnable") コマンド	17
3.1.11. CaoController::Execute("ViewLockEnable") コマンド	17
3.1.12. CaoController::Execute("GetKind") コマンド	17
3.1.13. CaoController::Execute("PutKind") コマンド	17
3.1.14. CaoController::Execute("GetModule") コマンド	18
3.1.15. CaoController::Execute("PutModule") コマンド	18
3.1.16. CaoController::Execute("GetViewMode") コマンド	18
3.1.17. CaoController::Execute("PutViewMode") コマンド	18
3.1.18. CaoController::Execute("GetDispMode") コマンド	19
3.1.19. CaoController::Execute("PutDispMode") コマンド	20
3.1.20. CaoController::Execute("GetMeasureData") コマンド	20
3.1.21. CaoController::Execute("ClearFigures") コマンド	20
3.1.22. CaoController::Execute("PutMeasureData") コマンド	21
3.1.23. CaoController::Execute("PutRemoteKey") コマンド	21

3.1.24. CaoController::Execute("PutPassword") コマンド	21
3.1.25. CaoController::Execute("GetVal") コマンド	21
3.1.26. CaoController::Execute("PutVal") コマンド	22
3.1.27. CaoController::Execute("RegStdImage") コマンド	23
3.1.28. CaoController::Execute("GetShutterSp") コマンド	24
3.1.29. CaoController::Execute("PutShutterSp") コマンド	24
3.1.30. CaoController::Execute("GetThreshold") コマンド	24
3.1.31. CaoController::Execute("PutThreshold") コマンド	24
3.1.32. CaoController::Execute("GetGain") コマンド	25
3.1.33. CaoController::Execute("PutGain") コマンド	25
3.1.34. CaoController::Execute("GetDate") コマンド	25
3.1.35. CaoController::Execute("PutDate") コマンド	26
3.1.36. CaoController::Execute("GetRegData") コマンド	26
3.1.37. CaoController::Execute("PutRegData") コマンド	26
3.1.38. CaoController::Execute("PutDateString") コマンド	27
3.1.39. CaoController::Execute("GetModuleString") コマンド	28
3.1.40. CaoController::Execute("GetBlockString") コマンド	28
3.1.41. CaoController::Execute("PutBlockString") コマンド	28
3.1.42. CaoController::Execute("Put20BlockString") コマンド	29
3.1.43. CaoController::Execute("Put10BlockString") コマンド	30
3.1.44. CaoController::Execute("SnapShot") コマンド	30
3.1.45. CaoController::Execute("ImageClear") コマンド	30
3.1.46. CaoController::Execute("GetKindState") コマンド	31
3.1.47. CaoController::Execute("GetStdImageState") コマンド	31
3.1.48. CaoController::Execute("GetVersion") コマンド	31
3.1.49. CaoController::Execute("AllReset") コマンド	31
3.1.50. CaoController::Execute("SettingSave") コマンド	31
3.1.51. CaoController::Execute("Reset") コマンド	32
3.1.52. CaoController::Execute("SysSettingSave") コマンド	32
3.1.53. CaoController::Execute("GetBrightness") コマンド	32
3.1.54. CaoController::Execute("GetParallel") コマンド	32
3.1.55. CaoController::Execute("SelfCheck") コマンド	33
4. サンプルプログラム	34
4.1. VisualBasic Ver6.0	34
4.2. PacScript	35

1. はじめに

本書はシャープマニファクチャリングシステム製の画像センサカメラ用の CAO プロバイダである、IV プロバイダのユーザーズガイドです。

IV プロバイダは、IV に対してコマンドの送信とレスポンスの受信を行います。

2. プロバイダの概要

2.1. 概要

IV 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` メソッドは、実行時にコマンドを自動生成、送信を行います。またレスポンスの解析も行い、データを取得します。

IV プロバイダのファイル形式は DLL(Dynamic Link Library)となっており、その詳細は表 2-1 のようになっています。

表 2-1 IV プロバイダ

ファイル名	CaoProv.SHARP.IV.dll
ProgID	CaoProv.SHARP.IV
レジストリ登録 ¹	regsvr32 CaoProv.SHARP.IV.dll
レジストリ登録の抹消	regsvr32 /u CaoProv.SHARP.IV.dll

¹ ORiN SDK でインストールした場合は手動で登録/抹消する必要はありません

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

IV プロバイダでは AddController 時に、通信用の接続パラメータを参照し、通信の接続を行います。このときオプションで通信形態、タイムアウト、機種タイプ、局番、チェックサム有無を指定します。

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.SHARP.IV”
 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 2-2 CaoWorkspace::AddController のオプション文字列

オプション	意味
Conn =<接続パラメータ>	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 に示します。
Timeout[=<タイムアウト時間>]	送受信時のタイムアウト時間を指定します。 (デフォルト : 500)
Type[=<機種タイプ>]	機種タイプを指定します。 0 : IV-S150X (デフォルト) 1 : IV-S200X,IV-S210X,IV-C250X 2 : IV-S300X,IV-S310X,IV-S301M,IV-S402M,IV-S412M
AreaNo[=<局番>]	通信時の局番を指定します。(0~255) (デフォルト : 0)
Chksum[=<チェックサム>]	通信時のチェックサム有無を指定します。 0 : 無し (デフォルト) 1 : 有り

2.3. エラーコード

IV プロバイダでは、以下の固有エラーコードが定義されています。ORiN2 共通エラーについては、「ORiN2 プログラミングガイド」のエラーコードの章を参照してください。

表 2-4 独自エラーコード一覧

エラー名	エラー番号	説明
E_RESPONSE_FAILED	0x80100001	無効なレスポンスを受信しました
E_RESPONSE_CHECKSUM_FAILED	0x80100002	レスポンスデータチェックサムエラー
E_RESPONSE_COMMAND_FAILED	0x80100003	レスポンスデータコマンドエラー
E_RESPONSE_LENGTH_FAILED	0x80100004	レスポンスデータ長エラー
E_RESPONSE_AREA_FAILED	0x80100005	レスポンス局番エラー
E_DEVICE_ERROR	0x80100100～	デバイスエラー

※E_DEVICE_ERROR について

デバイスからのエラーレスポンスは、"0x80100100"でマスクした値が出力されます。

例：コマンド実行不可：40(H) → CAO API のエラー：0x80100140

エラー内容についてはシャープマニファクチャリングシステム社の画像センサユーザーズマニュアルを参照してください。

3. 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。

表 3-1 CaoController::Execute コマンド一覧

IV コマ ンド	コマンド	機能	対応機種								
			IV-S150		IV-S2** IV-C250		IV-S3**		IV-S4*2		
			com	eth	com	eth	com	eth	com	eth	
-	Raw	生データ送受信.	○	○	○	○	○	○	○	○	○
-	ChangeTimeout	通信タイムアウト時間変更	-								
T00	TriggerAndWait	トリガを入力し結果を取得します.	○	○	○	○	○	○	○	○	○
T01	Trigger	トリガを入力します.	○	○	○	○	○	○	○	○	○
T02	GetData	画像処理の結果を取得します.	○	○	○	○	○	○	○	○	○
T10	RobotCalibration	ロボットのキャリブレーションを実行します.					○	○	○	○	
A00, A01	SerialEnable	シリアル通信の許可/禁止を設定します.			○						
A00	SAlignmentTrigger	各軸の現在値を設定します.					○	○	○	○	
A01	SAlignmentCalibration	キャリブレーション実行時の各種設定を行います.					○	○	○	○	
A10, A11	RemoteEnable	リモート設定キー入力の許可/禁止を設定します.			○						
A20, A21	ViewLockEnable	運転画面ロックの許可/禁止を設定します.			○						

C00	GetKind	品種番号を読み出します。	○	○	○	○	○	○	○	○
C01	PutKind	品番番号を書き込みます。	○	○	○	○	○	○	○	○
C10	GetModule	モジュール番号を読み込みます。			○	○				
C11	PutModule	モジュール番号を書き込みます。			○	○				
C20	GetViewMode	画像モードを読み出します。	○	○	○		○	○	○	○
C21	PutViewMode	画像モードを書き込みます。	○	○	○		○	○	○	○
C30	GetDispMode	表示モードを読み出します。	○	○	○		○	○	○	○
C31	PutDispMode	表示モードを書き込みます。	○	○	○		○	○	○	○
C40	GetMeasureData	手動計測座標を読み出します。			○					
C40	ClearFigures	計測回数をリセットします。 (全トリガ分)					○	○	○	○
C41	PutMeasureData	手動計測座標を書き込みます。			○					
C50	PutRemoteKey	リモート設定キー入力を書き込みます。 (未実装)			○					
C60	PutPassword	運転画面ロックパスワードを書き込みます。			○					
C80	GetVal	変数値を読み込みます。			○	○	○	○	○	○
C81	PutVal	変数値を書き込みます。			○	○	○	○	○	○
R00	RegStdImage	基準画像を上書き登録します。			○		○	○	○	○
R10	GetShutterSp	シャッター速度を読み出します。			○					
R11	PutShutterSp	シャッター速度を書き込みます。			○					
R30	GetThreshold	しきい値設定を読み出します。			○					

R31	PutThreshold	しきい値設定を書き込みます。			○					
R40	GetGain	ゲイン・オフセット設定を読み出します。			○					
R41	PutGain	ゲイン・オフセット設定を書き込みます。			○					
R50	GetDate	日時設定を読み出します。	○	○	○		○	○	○	○
R51	PutDate	日時設定を書き込みます。	○	○	○		○	○	○	○
R80	GetRegData	コードリーダ登録データを読み出します。					○	○	○	○
R81	PutRegData	コードリーダ登録データを書き込みます。					○	○	○	○
R89	PutDateString	設定文字列を書き込みます。 (日付ブロック一括8個, カメラ指定あり)					○	○	○	○
R90	GetModuleString	設定文字列を読み出します。 (モジュール指定)					○	○	○	○
R92	GetBlockString	設定文字列を読み出します。 (ブロック指定)					○	○	○	○
R93, R94	PutBlockString	設定文字列を書き込みます。 (ブロック指定)					○	○	○	○
R96	Put20BlockString	設定文字列を書き込みます。 (可変ブロック一括20個, 末尾の空白除去あり)					○	○	○	○
R98	Put10BlockString	設定文字列を書き込みます。 (可変ブロック一括10個, カメラ指定あり, 末尾の空白除去あり)					○	○	○	○

I01	SnapShot	スナップショット画像を外部メモリに保存します。	○	○	○	○	○	○	○	○
I20	ImageClear	画像メモリを消去します。			○	○				
P10	GetKindState	品種設定番号情報を読み出します。				○				
P20	GetStdImageState	基準画像番号情報を読み出します。				○				
D00	GetVersion	バージョン情報を読み出します。			○	○				
D10	AllReset	システム設定, 品種設定を初期化します。			○	○				
D11	SettingSave	システム設定, 品種設定を保存します。	○	○	○	○	○	○	○	○
D12	Reset	本機をリセットします。			○					
D14	SysSettingSave	設定を保存します。(システム, 品種)					○	○	○	○
D20	GetBrightness	平均濃度を読み出します。	○	○	○		○	○	○	○
D21	GetParallel	パラレルの入出力情報を読み出します	○	○	○		○	○	○	○
D40	SelfCheck	自己診断を行います。					○	○	○	○

3.1. Execute コマンド

3.1.1. CaoController::Execute(“Raw”) コマンド

生データ送受信を行います。

書式 Raw (<bstrSendData>)

<bstrSendData> : [in] 送信文字列 (VT_BSTR)

戻り値 : [out] 受信文字列 (VT_BSTR)

引数で指定した文字列を送信します。受信したレスポンスを文字列で返します。

このとき送受信でコマンド及びレスポンスの内容は一切加工しません。

3.1.2. CaoController::Execute(“ChangeTimeout”) コマンド

通信タイムアウト時間を設定します。

書式 ChangeTimeout (<lTimeout>)

<lTimeout> : [in] タイムアウト時間 ms (VT_I4)

戻り値 : 無し

タイムアウト時間に-1を指定すると、AddController時のタイムアウト時間を設定します。

3.1.3. CaoController::Execute(“TriggerAndWait”) コマンド

トリガを入力し、結果を取得します。

書式 TriggerAndWait (<lTrgNo>)

<lTrgNo> : [in] トリガ番号 (VT_I4)

戻り値 : [out] 出力データ (VT_R4 | VT_ARRAY)

3.1.4. CaoController::Execute(“Trigger”) コマンド

トリガを入力します。

書式 Trigger (<lTrgNo>)

<lTrgNo> : [in] トリガ番号 (VT_I4)

戻り値 : なし

3.1.5. CaoController::Execute(“GetData”) コマンド

結果を取得します。

書式 GetData (<lTrgNo>)

<lTrgNo> : [in] トリガ番号 (VT_I4)

戻り値 : [out] 出力データ (VT_R4 | VT_ARRAY)

3.1.6. CaoController::Execute(“RobotCalibration”) コマンド

トリガを入力して画像処理を実行します。

ロボットのキャリブレーションを行う場合に実行します。

各軸の絶対座標は小数点第4位で四捨五入され、小数部は3桁になります。

書式 RobotCalibration (<lTriggerNo >,<fltXPos>,<fltYPos>,<fltRPos>)

<lTriggerNo > : [in] トリガ番号 (VT_I4)
 <fltXPos > : [in] 現在のロボットの X 座標絶対座標 (mm) (VT_R4)
 <fltYPos > : [in] 現在のロボットの Y 座標絶対座標 (mm) (VT_R4)
 <fltRPos > : [in] 現在のロボットの R 座標絶対座標 (deg) (VT_R4)
 戻り値 : [out] 完了フラグ (VT_BOOL)

TRUE	終了
FALSE	継続

: [out] 次のロボットの X 座標絶対座標(mm) (VT_R4)
 : [out] 次のロボットの Y 座標絶対座標 (mm) (VT_R4)
 : [out] 次のロボットの R 座標絶対座標 (deg) (VT_R4)

3.1.7. CaoController::Execute(“SerialEnable”) コマンド

シリアル通信の許可/禁止を設定します。

書式 SerialEnable (<bState >)

<bState > : [in] 許可 : True / 禁止 : False (VT_BOOL)
 戻り値 : なし

3.1.8. CaoController::Execute(“SAlignmentTrigger”) コマンド

アライメント実行前に、各軸の現在値を設定します。

書式 SAlignmentTrigger (<lTriggerNo >,<lXPos>,<lYPos>,<lTheta>)

<lTriggerNo > : [in] トリガ番号 (VT_I4)
 <lXPos > : [in] X 軸現在値 (VT_I4)
 <lYPos > : [in] Y(Y1)軸現在値 (VT_I4)
 <lTheta > : [in] θ (Y2)軸現在値 (VT_I4)
 戻り値 : [out] 出力データ (VT_VARIANT | VT_ARRAY)

3.1.9. CaoController::Execute(“SAlignmentCalibration”) コマンド

キャリブレーション実行時の各種設定を行います。

書式 SAlignmentCalibration (<lTriggerNo >,<lXPos>,<lYPos>,<lTheta>,<bFlag>)

<lTriggerNo > : [in] トリガ番号 (VT_I4)

<IXPos> : [in] X 軸現在値 (VT_I4)
 <IYPos> : [in] Y(Y1)軸現在値 (VT_I4)
 <ITheta> : [in] θ (Y2)軸現在値 (VT_I4)
 <bFlag> : [in] 開始フラグ (VT_BOOL)

TRUE	キャリブレーション開始時, またはキャリブレーション再スタート時
FALSE	上記以外

戻り値 : [out] X 軸移動量 (VT_I4)
 : [out] Y(Y1)軸移動量 (VT_I4)
 : [out] θ (Y2)軸移動量 (VT_I4)
 : [out] 終了フラグ(VT_BOOL)

TRUE	キャリブレーション完了
FALSE	キャリブレーション未完了

3.1.10. GaoController::Execute(“RemoteEnable”) コマンド

リモート設定キー入力の許可/禁止を設定します。

書式 RemoteEnable (<bState >)

<bState > : [in] 許可 : True / 禁止 : False (VT_BOOL)
 戻り値 : なし

3.1.11. GaoController::Execute(“ViewLockEnable”) コマンド

運転画面ロックの許可/禁止を設定します。

書式 ViewLockEnable (<bState >)

<bState > : [in] 許可 : True / 禁止 : False (VT_BOOL)
 戻り値 : なし

3.1.12. GaoController::Execute(“GetKind”) コマンド

品種番号を読み出します。

書式 GetKind ()

引数 : なし
 戻り値 : [out] 品種番号 (VT_I4 | VT_ARRAY)

結果はトリガ番号 0 品種番号, トリガ番号 1 品種番号の順番に格納されています。

3.1.13. GaoController::Execute(“PutKind”) コマンド

品種番号を書き込みます。

書式 PutKind (<lKindNo >)

<lKindNo > : [in] 品種番号 (VT_I4)
 戻り値 : なし

品種の内容によっては、品種切替え時に時間が掛かる場合があります。

3.1.14. CaoController::Execute(“GetModule”) コマンド

モジュール番号を読み出します。

書式 GetModule ()

引数 : 無し
 戻り値 : [out] モジュール番号 (VT_I4)

3.1.15. CaoController::Execute(“PutModule”) コマンド

モジュール番号を書き込みます。

書式 PutModule (<lModuleNo>)

<lModuleNo > : [in] モジュール番号 (VT_I4)
 戻り値 : 無し

3.1.16. CaoController::Execute(“GetViewMode”) コマンド

画面更新モードを読み出します。

書式 GetViewMode ()

引数 : なし
 戻り値 : [out] 画面更新モード (VT_BSTR)

IV-S150	LV	動画
IV-S3*0	SC	カメラ画像
IV-S4*2	RC	処理画像
	LV	動画
IV-S2*0	ST	静止画(トリガ毎)
IV-C250	SN	静止画(NG 毎)
	SO	静止画(OK 毎)

3.1.17. CaoController::Execute(“PutViewMode”) コマンド

画面更新モードを書き込みます。

書式 PutViewMode (<bstrMode >)

<bstrMode> : [in] 画面更新モード (VT_BSTR)

IV-S150	LV	動画
IV-S3*0	SC	カメラ画像
IV-S4*2	RC	処理画像

IV-S2*0 IV-C250	LV	動画
	ST	静止画(トリガ毎)
	SN	静止画(NG 毎)
	SO	静止画(OK 毎)

戻り値 : なし

3.1.18. GaoController::Execute(“GetDispMode”) コマンド

表示モードを読み出します。

書式 GetDispMode ()

引数 : なし

戻り値 : [out] 表示モード (VT_BSTR)

IV-S150	C1	カメラ 1 表示
	C2	カメラ 2 表示
	DV	分割表示
IV-S2*0 IV-C250	MI	モジュール指定カメラ
	MO	モジュール出力画像
	C1	カメラ 1 表示
	C2	カメラ 2 表示
	C3	カメラ 3 表示
	C4	カメラ 4 表示
	DV	分割表示
IV-S3*0 IV-S4*2	C1	カメラ 1 表示
	C2	カメラ 2 表示
	C3	カメラ 3 表示
	C4	カメラ 4 表示
	DV	カメラ 1+2 表示
	DW	カメラ 3+4 表示
	DX	カメラ 1+2+3+4 表示

3.1.19. CaoController::Execute(“PutDispMode”) コマンド

表示モードを書き込みます。

書式 PutDispMode (< bstrMode >)

< bstrMode > : [in] 表示モード (VT_BSTR)

IV-S150	C1	カメラ 1 表示
	C2	カメラ 2 表示
	DV	分割表示
IV-S2*0 IV-C250	MI	モジュール指定カメラ
	MO	モジュール出力画像
IV-S3*0 IV-S4*2	C1	カメラ 1 表示
	C2	カメラ 2 表示
	C3	カメラ 3 表示
	C4	カメラ 4 表示
	DV	分割表示
	DW	カメラ 1+2 表示
	DX	カメラ 3+4 表示
	DX	カメラ 1+2+3+4 表示

戻り値 : なし

3.1.20. CaoController::Execute(“GetMeasureData”) コマンド

手動計測座標を読み出します。

書式 GetMeasureData (<lCamNo>)

<lCamNo> : [in] カメラ番号 (VT_I4)

戻り値 : [out] 座標値 (VT_I4 | VT_ARRAY)

座標値は第 1 点の X 座標値, 第 1 点の Y 座標値, 第 2 点の X 座標値, 第 2 点の Y 座標値の順番に格納されています

3.1.21. CaoController::Execute(“ClearFigures”) コマンド

計測回数を全トリガ分リセットします。

書式 ClearFigures ()

引数 : なし

戻り値 : なし

3.1.22. CaoController::Execute(“PutMeasureData”) コマンド

手動計測座標を書き込みます。

書式 PutMeasureData (<ICamNo> , <IX1> , <IY1> , <IX2> , <IY2>)

<ICamNo> : [in] カメラ番号 (VT_I4)
 <IX1> : [in] 第1点の X 座標値 (VT_I4)
 <IY1> : [in] 第1点の Y 座標値 (VT_I4)
 <IX2> : [in] 第2点の X 座標値 (VT_I4)
 <IY2> : [in] 第2点の Y 座標値 (VT_I4)
 戻り値 : なし

3.1.23. CaoController::Execute(“PutRemoteKey”) コマンド

リモート設定キー入力を書き込みます。(未実装)

3.1.24. CaoController::Execute(“PutPassword”) コマンド

運転画面ロックパスワードを書き込みます。

書式 PutPassword (<IPassword>)

<IPassword> : [in] パスワード (VT_I4)
 戻り値 : なし

3.1.25. CaoController::Execute(“GetVal”) コマンド

変数値を読み出します。

書式 GetVal (<IValType> , <IValNo>)

<IValType> : [in] IV-S2*0 の場合
 変数種類 (VT_I4)

0	システム変数
1	モジュール変数(トリガ 1)
2	モジュール変数(トリガ 2)

IV-S3*0,IV-S4*2 の場合

トリガ番号(VT_I4)

<IValNo> [in] 変数番号 (VT_I4)

戻り値 : [out] 変数値 (VT_R4)

変数値の有効桁数は、画像センサ本体設定による指定となります。

3.1.26. CaoController::Execute(“PutVal”) コマンド

変数値を書き込みます。

書式 PutVal (<lValType> , <lValNo> , <fltValData>)

<lValType> : [in] IV-S2*0 の場合

変数種類 (VT_I4)

0	システム変数
1	モジュール変数(トリガ 1)
2	モジュール変数(トリガ 2)

IV-S3*0,IV-S4*2 の場合

トリガ番号(VT_I4)

<lValNo> [in] 変数番号 (VT_I4)

<fltValData> [in] 変数値 (VT_R4)

戻り値 : 無し

変数値の有効桁数は、画像センサ本体設定による指定となります。

3.1.27. GaoController::Execute(“RegStdImage”) コマンド

基準画像を上書き登録します。

書式

RegStdImage (<ICamNo> , <IStdImageNo>)

<ICamNo> : [in] IV-S2*0 の場合
カメラ番号 (VT_I4)
IV-S3*0,IV-S4*2 の場合
トリガ番号(VT_I4)

<IStdImageNo> : [in] IV-S2*0 の場合
基準画像番号 (VT_I4)
IV-S3*0,IV-S4*2 の場合
カメラ組み合わせ番号(VT_I4)

カメラ組み合わせ	カメラ組み合わせ No
なし	0
カメラ 1	1
カメラ 2	2
カメラ 3	4
カメラ 4	8
カメラ 1+2	3
カメラ 1+3	5
カメラ 1+4	9
カメラ 2+3	6
カメラ 2+4	10
カメラ 3+4	12
カメラ 1+2+3	7
カメラ 1+2+4	11
カメラ 1+3+4	13
カメラ 2+3+4	14
カメラ 1+2+3+4	15

戻り値 : 無し

3.1.28. CaoController::Execute(“GetShutterSp”) コマンド

シャッター速度を読み出します。

書式	GetShutterSp (<IKindNo > , <IModuleNo > , <ICamNo >)
<IKindNo >	: [in] 品種番号 (VT_I4)
<IModuleNo>	: [in] モジュール番号 (VT_I4)
<ICamNo >	: [in] カメラ番号 (VT_I4)
戻り値	: [out] シャッター速度 (VT_I4)

3.1.29. CaoController::Execute(“PutShutterSp”) コマンド

シャッター速度を書き込みます。

書式	PutShutterSp (<IKindNo > , <IModuleNo > , <ICamNo > , <IShutterSp>)
<IKindNo >	: [in] 品種番号 (VT_I4)
<IModuleNo>	: [in] モジュール番号 (VT_I4)
<ICamNo >	: [in] カメラ番号 (VT_I4)
<IShutterSp>	: [in] シャッター速度 (VT_I4)
戻り値	: なし

3.1.30. CaoController::Execute(“GetThreshold”) コマンド

しきい値設定を読み出します。

書式	GetThreshold (<IKindNo > , <IModuleNo >)
<IKindNo >	: [in] 品種番号 (VT_I4)
<IModuleNo>	: [in] モジュール番号 (VT_I4)
戻り値	: [out] しきい値設定 (VT_I4 VT_ARRAY)

しきい値設定は上限値, 下限値の順番に格納されています

3.1.31. CaoController::Execute(“PutThreshold”) コマンド

しきい値設定を書き込みます。

書式	PutThreshold (<IKindNo > , <IModuleNo > , <IThresholdU > , <IThresholdL>)
<IKindNo >	: [in] 品種番号 (VT_I4)
<IModuleNo>	: [in] モジュール番号 (VT_I4)
<IThresholdU>	: [in] しきい値上限値 (VT_I4)
<IThresholdL>	: [in] しきい値下限値 (VT_I4)
戻り値	: なし

3.1.32. CaoController::Execute(“GetGain”) コマンド

ゲイン・オフセット設定を読み出します。

書式

GetGain (<lKindNo > , <lModuleNo > , <lCamNo >)

<lKindNo > : [in] 品種番号 (VT_I4)
<lModuleNo> : [in] モジュール番号 (VT_I4)
<lCamNo > : [in] カメラ番号 (VT_I4)
戻り値 : [out] ゲイン値, オフセット値 (VT_I4 | VT_ARRAY)

結果はゲイン値, オフセット値の順番に格納されています

3.1.33. CaoController::Execute(“PutGain”) コマンド

ゲイン・オフセット設定を書き込みます。

書式

PutGain (<lKindNo > , <lModuleNo > , <lCamNo > , <lGain> , <lOffset>)

<lKindNo > : [in] 品種番号 (VT_I4)
<lModuleNo> : [in] モジュール番号 (VT_I4)
<lCamNo > : [in] カメラ番号 (VT_I4)
<lGain > : [in] ゲイン値 (VT_I4)
<lOffset > : [in] オフセット値 (VT_I4)
戻り値 : なし

3.1.34. CaoController::Execute(“GetDate”) コマンド

日時設定を読み出します。

書式

GetDate ()

引数 : なし
戻り値 : [out] 日時設定 (VT_I4 | VT_ARRAY)

日時設定は年, 月, 日, 時, 分, 秒の順番に格納されています

3.1.35. CaoController::Execute(“PutDate”) コマンド

日時設定を書き込みます。

書式 PutDate (<lYear > , <lMonth > , <lDay > , <lHour > , <lMinute> , <lSec>)

<lYear >	:	[in] 年 (VT_I4)
<lMonth >	:	[in] 月 (VT_I4)
<lDay >	:	[in] 日 (VT_I4)
<lHour >	:	[in] 時 (VT_I4)
<lMinute >	:	[in] 分 (VT_I4)
<lSec >	:	[in] 秒 (VT_I4)
戻り値	:	なし

3.1.36. CaoController::Execute(“GetRegData”) コマンド

コードリーダーモジュールの登録データを取得します。

書式 GetRegData (<lTrigNo > , <lModuleNo >)

<lTrigNo >	:	[in] トリガ番号 (VT_I4)
<lModuleNo >	:	[in] モジュール番号 (VT_I4)
戻り値	:	[out] 登録データ(VT_BSTR)

COM 通信を使用する場合は、機器側とプロバイダ側でデータビット数を 8bit に設定してください。データビット数の設定がデフォルトで 7bit となっているため、使用する文字列によっては、8bit 目が欠落し、意図した動作となりません。

3.1.37. CaoController::Execute(“PutRegData”) コマンド

コードリーダーモジュールの登録データを書き込みます。

書式 PutRegData (<lTrigNo > , <lModuleNo > , <bstrRegistData >)

<lTrigNo >	:	[in] トリガ番号 (VT_I4)
<lModuleNo >	:	[in] モジュール番号 (VT_I4)
<bstrRegistData >	:	[in] 登録データ(VT_BSTR)
戻り値	:	なし

COM 通信を使用する場合は、機器側とプロバイダ側でデータビット数を 8bit に設定してください。データビット数の設定がデフォルトで 7bit となっているため、使用する文字列によっては、8bit 目が欠落し、意図した動作となりません。

3.1.39. CaoController::Execute(“GetModuleString”) コマンド

文字検査モジュールの設定文字列を読み出します。

書式 GetModuleString (<ITrigNo > , <IModuleNo > ,)

<ITrigNo > : [in] トリガ番号 (VT_I4)
 <IModuleNo > : [in] モジュール番号 (VT_I4)
 戻り値 : [out] 登録データ (VT_BSTR)

COM 通信を使用する場合は、機器側とプロバイダ側でデータビット数を 8bit に設定してください。データビット数の設定がデフォルトで 7bit となっているため、使用する文字列によっては、8bit 目が欠落し、意図した動作となりません。

3.1.40. CaoController::Execute(“GetBlockString”) コマンド

文字検査モジュールの設定文字列を読み出します。

書式 GetBlockString (<ITrigNo > , <IModuleNo > , <IBlockNo>)

<ITrigNo > : [in] トリガ番号 (VT_I4)
 <IModuleNo > : [in] モジュール番号 (VT_I4)
 <IBlockNo> : [in] ブロック番号 (VT_I4)
 戻り値 : [out] 登録データ (VT_BSTR)

COM 通信を使用する場合は、機器側とプロバイダ側でデータビット数を 8bit に設定してください。データビット数の設定がデフォルトで 7bit となっているため、使用する文字列によっては、8bit 目が欠落し、意図した動作となりません。

3.1.41. CaoController::Execute(“PutBlockString”) コマンド

文字検査モジュールの設定文字列を書き込みます。

書式 PutBlockString (<ITrigNo > , <IModuleNo > , <IBlockNo> , <bstrChar> , <bDelSpace>)

<ITrigNo > : [in] トリガ番号 (VT_I4)
 <IModuleNo> : [in] モジュール番号 (VT_I4)
 <IBlockNo> : [in] ブロック番号 (VT_I4)
 <bstrChar> : [in] 文字列(VT_BSTR)
 <bDelSpace> : [in] 空白除去あり/なし(VT_BOOL)

TRUE	空白除去あり
FALSE	空白除去なし

戻り値 : なし

空白除去は、文字列末尾の空白の除去を行います。

COM 通信を使用する場合は、機器側とプロバイダ側でデータビット数を 8bit に設定してください。データビット数の設定がデフォルトで 7bit となっているため、使用する文字列によっては、8bit 目が欠落し、意図した動作となりません。

3.1.42. GaoController::Execute(“Put20BlockString”) コマンド

文字検査モジュールの設定文字列を書き込みます。

書式

```
Put20BlockString (<ITrigNo> , <bstrChar1 > [, <bstrChar2> [, <bstrChar3>
                    [, <bstrChar4> [, <bstrChar5> [, <bstrChar6> [, <bstrChar7>
                    [, <bstrChar8> [, <bstrChar9> [, <bstrChar10 > [, <bstrChar11>
                    [, <bstrChar12 > [, <bstrChar13> [, <bstrChar14 > [, <bstrChar15 >
                    [, <bstrChar16 > [, <bstrChar17 > [, <bstrChar18 > [, <bstrChar19 >
                    [, <bstrChar20 > ]]]]]]]]]]]]]]]]]]]]]))
```

```
<ITrigNo >      : [in] トリガ番号 (VT_I4)
<bstrChar1 >    : [in] 文字列 1(VT_BSTR)
<bstrChar2>    : [in] 文字列 2(VT_BSTR)
<bstrChar3>    : [in] 文字列 3(VT_BSTR)
<bstrChar4 >   : [in] 文字列 4(VT_BSTR)
<bstrChar5 >   : [in] 文字列 5(VT_BSTR)
<bstrChar6 >   : [in] 文字列 6(VT_BSTR)
<bstrChar7 >   : [in] 文字列 7(VT_BSTR)
<bstrChar8 >   : [in] 文字列 8(VT_BSTR)
<bstrChar9 >   : [in] 文字列 9(VT_BSTR)
<bstrChar10 >  : [in] 文字列 10(VT_BSTR)
<bstrChar11 > : [in] 文字列 11(VT_BSTR)
<bstrChar12 > : [in] 文字列 12(VT_BSTR)
<bstrChar13 > : [in] 文字列 13(VT_BSTR)
<bstrChar14 > : [in] 文字列 14(VT_BSTR)
<bstrChar15 > : [in] 文字列 15(VT_BSTR)
<bstrChar16 > : [in] 文字列 16(VT_BSTR)
<bstrChar17 > : [in] 文字列 17(VT_BSTR)
<bstrChar18 > : [in] 文字列 18(VT_BSTR)
<bstrChar19 > : [in] 文字列 19(VT_BSTR)
<bstrChar20 > : [in] 文字列 20(VT_BSTR)
戻り値          : なし
```

COM 通信を使用する場合は、機器側とプロバイダ側でデータビット数を 8bit に設定してください。データビット数の設定がデフォルトで 7bit となっているため、使用する文字列によっては、8bit 目が欠落し、意図した動作となりません。

3.1.43. CaoController::Execute(“Put10BlockString”) コマンド

文字検査モジュールの設定文字列を書き込みます。

書式 Put10BlockString (<ITrigNo> , <ICamNo>, <bstrChar1 > [, <bstrChar2> [, <bstrChar3>
 [, <bstrChar4> [, <bstrChar5> [, <bstrChar6> [, <bstrChar7>
 [, <bstrChar8> [, <bstrChar9> [, <bstrChar10 >]]]]]]]]]))

<ITrigNo > : [in] トリガ番号 (VT_I4)
 <ICamNo> : [in] カメラ番号 (VT_I4)
 <bstrChar1 > : [in] 文字列 1(VT_BSTR)
 <bstrChar2 > : [in] 文字列 2(VT_BSTR)
 <bstrChar3 > : [in] 文字列 3(VT_BSTR)
 <bstrChar4 > : [in] 文字列 4(VT_BSTR)
 <bstrChar5 > : [in] 文字列 5(VT_BSTR)
 <bstrChar6 > : [in] 文字列 6(VT_BSTR)
 <bstrChar7 > : [in] 文字列 7(VT_BSTR)
 <bstrChar8 > : [in] 文字列 8(VT_BSTR)
 <bstrChar9 > : [in] 文字列 9(VT_BSTR)
 <bstrChar10 > : [in] 文字列 10(VT_BSTR)
 戻り値 : なし

COM 通信を使用する場合は、機器側とプロバイダ側でデータビット数を 8bit に設定してください。データビット数の設定がデフォルトで 7bit となっているため、使用する文字列によっては、8bit 目が欠落し、意図した動作となりません。

3.1.44. CaoController::Execute(“SnapShot”) コマンド

スナップショット画像を外部メモリに保存します。

書式 SnapShot ()

引数 : なし
 戻り値 : なし

3.1.45. CaoController::Execute(“ImageClear”) コマンド

本体内蔵の画像メモリを消去します。

書式 ImageClear ()

引数 : なし
 戻り値 : なし

3.1.46. CaoController::Execute(“GetKindState”) コマンド

品種設定番号情報を読み出します。

書式 GetKindState ()

引数 : なし

戻り値 : [out] 品種設定番号情報 (VT_I4 | VT_ARRAY)

品種設定番号情報は品種設定データ数, 品種番号 1, 品種番号 2, . . . , 品種番号 n の順番に格納されています

3.1.47. CaoController::Execute(“GetStdImageState”) コマンド

基準画像番号情報を読み出します。

書式 GetStdImageState ()

引数 : なし

戻り値 : [out] 基準画像番号情報 (VT_I4 | VT_ARRAY)

基準画像番号情報は基準画像数, 基準画像番号 1, 基準画像番号 2, . . . , 基準画像番号 n の順番に格納されています

3.1.48. CaoController::Execute(“GetVersion”) コマンド

バージョン情報を読み出します。

書式 GetVersion ()

引数 : なし

戻り値 : [out] バージョン情報 (VT_BSTR | VT_ARRAY)

バージョン情報は機種コード, バージョン情報の順番に格納されています

3.1.49. CaoController::Execute(“AllReset”) コマンド

システム設定, 品種設定を初期化します。

書式 AllReset ()

引数 : なし

戻り値 : なし

3.1.50. CaoController::Execute(“SettingSave”) コマンド

システム設定, 品種設定を保存します。

書式 SettingSave ()

引数 : なし

戻り値 : なし

3.1.51. CaoController::Execute(“Reset”) コマンド

本機をリセットします。

書式 Reset ()

引数 : なし
戻り値 : なし

3.1.52. CaoController::Execute(“SysSettingSave”) コマンド

システム設定と品種設定を保存します。

書式 SysSettingSave ()

引数 : なし
戻り値 : なし

運転モードで最後に利用していた品種が、システム起動時の品種(起動品種)となります。

3.1.53. CaoController::Execute(“GetBrightness”) コマンド

平均濃度を読み出します。

書式 GetBrightness (<lCamNo > , <lPosX1 > , <lPosY1 > , <lPosX2> , <lPosY2>)

<lCamNo > : [in] カメラ番号 (VT_I4)
<lPosX1 > : [in] 左上 X 座標値 (VT_I4)
<lPosY1 > : [in] 左上 Y 座標値 (VT_I4)
<lPosX2> : [in] 右下 X 座標値 (VT_I4)
<lPosY2> : [in] 右下 Y 座標値 (VT_I4)
戻り値 : [out] 平均濃度 (VT_I4)

3.1.54. CaoController::Execute(“GetParallel”) コマンド

パラレルの入出力情報を読み出します。

書式 GetParallel (<lDataSelect >)

<lDataSelect > : [in] 入出力指定 (VT_I4)

0	入出力
1	入力のみ
2	出力のみ

戻り値 : [out] 入出力状態 (VT_I4 | VT_ARRAY)

結果は入力 1 バイト目～4 バイト目、出力 1 バイト目～5 バイト目の順番で 1 バイト毎に格納されています。

(入力のみ、出力のみの場合は、指定した状態のみ読み込みます。)

3.1.55. CaoController::Execute(“SelfCheck”) コマンド

コントローラの自己診断(5種類のテスト)を実行します。

書式 SelfCheck ()

引数 : なし
 戻り値 : [out]システムメモリーテスト結果(VT_I4)※1
 : [out] RAM テスト結果(VT_I4)※1
 : [out] FPGA アクセステスト結果(VT_I4)※1
 : [out] カメラ 1 接続テスト結果(VT_I4)※2
 : [out] カメラ 2 接続テスト結果(VT_I4)※2
 : [out] カメラ 3 接続テスト結果(VT_I4)※2
 : [out] カメラ 4 接続テスト結果(VT_I4)※2

※1 テスト結果の値は次表のとおりです。

値	結果
0	正常
1	異常

※2 カメラテスト結果の値は次表のとおりです。

値	結果
0	正常
1	カメラ接続テスト失敗
2	カメラ種別テスト失敗
3	カメラ視野テスト失敗
4	カメラ取込テスト失敗
5	カメラ取込ラインテスト失敗

4. サンプルプログラム

品種の切替え, 及びトリガ入力, 結果取得するコードを示します.

4.1. VisualBasic Ver6.0

List 4-1 Sample.frm

```
Dim eng As CaoEngine
Dim ctrl As CaoController

Private Sub Form_Load()

    ' CAO エンジンの生成
    Set eng = New CaoEngine

    ' 画像センサ(IV-S2**)への接続 (局番:1)
    Set ctrl = eng.Workspaces(0).AddController("Sample", "CaoProv.SHARP.IV", "",
"Conn=eth:192.168.0.2, Type=1, AreaNo=1")

End Sub

Private Sub Command1_Click()

    Dim Res() As Single
    Dim xData As Single
    Dim yData As Single
    Dim AngleData As Single

    ' トリガの入力
    Res = Ctrl.TriggerAndWait(1)
    If Res(0) = 1 Then
        xData = Res(1)
        yData = Res(2)
        AngleData = Res(3)
    Else
        MsgBox "画像処理失敗"
    End If

End Sub

Private Sub Command2_Click()

    ' 品種切替え
    Ctrl.ChangeTimeout 5000          'タイムアウト時間を5秒に変更
    Ctrl.PutKind 10                  '品種10に切替え
    Ctrl.ChangeTimeout -1           'タイムアウト時間を元に戻す

End Sub
```

4.2. PacScript

List 4-2**Sample.pcs**

```
Dim ctrl As Object

Sub Main()

    Dim Res As Variant
    Dim xData As Single
    Dim yData As Single
    Dim AngleData As Single

    ' 画像センサ (IV-S2**) への接続 (局番:1)
    Ctrl = cao.AddController("Sample", "CaoProv. SHARP. IV", "",
        "Conn=eth:192.168.0.2, Type=1, AreaNo=1")

    ' 品種切替え
    Ctrl.ChangeTimeout 5000          ' タイムアウト時間を 5 秒に変更
    Ctrl.PutKind 10                  ' 品種 10 に切替え
    Ctrl.ChangeTimeout -1           ' タイムアウト時間を元に戻す

    ' トリガの入力
    Res = Ctrl.TriggerAndWait(1)
    If Res(0) = 1 Then
        xData = Res(1)
        yData = Res(2)
        AngleData = Res(3)
    End If

End Sub
```