

株式会社アイエイアイ IAI PCON プロバイダ

Version 1.0.2

ユーザーズ ガイド

May 16, 2022

備考：

【改版履歴】

バージョン	日付	内容
1.0.0	2017-12-6	初版.
1.0.1	2018-11-5	AddController 失敗時のメモリリーク修正.
1.0.2	2019-04-26	CaoRobot の作成/削除を繰り返すとメモリリークが発生する問題を修正.
	2020-10-06	ライセンス追加事項追記
	2022-05-16	サンプルプログラム修正

【対応機器】

機種	バージョン	注意事項
ERC2 (SE) / ERC3 (V0002 以降) / RCP6S シリーズ	-	
PCON-C / CA / CB / CFA / CFB / CG / CGB / CGFB / CF / CY / SE	-	
ACON-C / CG / CA / CB / CY / SE	-	
SCON-C / CA / CAL / CB (サーボプラス仕様含む) / CGB	-	
DCON-CA / CB	-	
ROBONET_RS485 (RTU モードかつ SIO スルモード時)	-	
	-	

【ご注意】

本プロバイダを使用する場合は “IAI ポジショナータイプコントローラプロバイダ” ライセンスが必要です.

1. はじめに.....	6
1.1. ライセンスの追加	7
2. プロバイダの概要	8
2.1. 概要.....	8
2.2. メソッド・プロパティ	9
2.2.1. CaoWorkspace::AddController メソッド.....	9
2.2.2. CaoController::AddRobot メソッド	11
2.2.3. CaoRobot:: Execute メソッド.....	12
2.2.3.1. CaoRobot::Execute(“ReadMultipleRegisters”)コマンド	16
2.2.3.2. CaoRobot::Execute(“ReadAlarmInfo”)コマンド	16
2.2.3.3. CaoRobot::Execute(“ReadPositionData”)コマンド	16
2.2.3.4. CaoRobot::Execute(“ReadMoveCount”)コマンド	17
2.2.3.5. CaoRobot::Execute(“ReadMoveDistance”)コマンド	17
2.2.3.6. CaoRobot::Execute(“ReadCurrentTime”)コマンド	17
2.2.3.7. CaoRobot::Execute(“ReadFanDriveTime”)コマンド	18
2.2.3.8. CaoRobot::Execute(“ReadCurrentPosition”)コマンド.....	18
2.2.3.9. CaoRobot::Execute(“ReadCurrentAlarmCode”)コマンド.....	18
2.2.3.10. CaoRobot::Execute(“ReadIOPortInputSignal”)コマンド	18
2.2.3.11. CaoRobot::Execute(“ReadIOPortOutputSignal”)コマンド	19
2.2.3.12. CaoRobot::Execute(“ReadDSS1”)コマンド.....	19
2.2.3.13. CaoRobot::Execute(“ReadDSS2”)コマンド.....	19
2.2.3.14. CaoRobot::Execute(“ReadDSSE”)コマンド	19
2.2.3.15. CaoRobot::Execute(“ReadSTAT”)コマンド	20
2.2.3.16. CaoRobot::Execute(“ReadCurrentSpeed”)コマンド.....	20
2.2.3.17. CaoRobot::Execute(“ReadCurrentValue”)コマンド	20
2.2.3.18. CaoRobot::Execute(“ReadDeviation”)コマンド	20
2.2.3.19. CaoRobot::Execute(“ReadIntegrationTime”)コマンド.....	21
2.2.3.20. CaoRobot::Execute(“ReadSpecialInputPort”)コマンド	21
2.2.3.21. CaoRobot::Execute(“ReadZoneStatus”)コマンド.....	21
2.2.3.22. CaoRobot::Execute(“ReadCompletePositionNo”)コマンド	22
2.2.3.23. CaoRobot::Execute(“ReadSSSE”)コマンド.....	22
2.2.3.24. CaoRobot::Execute(“ReadLoad”)コマンド	22
2.2.3.25. CaoRobot::Execute(“ReadLoadLevel”)コマンド	22
2.2.3.26. CaoRobot::Execute(“ReadPressProgramAlarmCode”)コマンド	23
2.2.3.27. CaoRobot::Execute(“ReadAlarmPressProgramNo”)コマンド	23
2.2.3.28. CaoRobot::Execute(“ReadPressProgramStatus”)コマンド	23
2.2.3.29. CaoRobot::Execute(“ReadPressProgramJudgeStatus”)コマンド	23
2.2.3.30. CaoRobot::Execute(“WriteSingleDiscreteOutput”)コマンド	24
2.2.3.31. CaoRobot::Execute(“SwitchSafetySpeed”)コマンド.....	24
2.2.3.32. CaoRobot::Execute(“SwitchServo”)コマンド	24
2.2.3.33. CaoRobot::Execute(“ResetAlarm”)コマンド.....	24
2.2.3.34. CaoRobot::Execute(“ReleaseBreak”)コマンド.....	25

2.2.3.35. CaoRobot::Execute("Pause")コマンド	25
2.2.3.36. CaoRobot::Execute("Home")コマンド	26
2.2.3.37. CaoRobot::Execute("StartPosition")コマンド	26
2.2.3.38. CaoRobot::Execute("SwitchJogInching")コマンド	27
2.2.3.39. CaoRobot::Execute("SwitchMode")コマンド	27
2.2.3.40. CaoRobot::Execute("GetCurrentPositionData")コマンド	27
2.2.3.41. CaoRobot::Execute("JogPlus")コマンド	27
2.2.3.42. CaoRobot::Execute("JogMinus")コマンド	28
2.2.3.43. CaoRobot::Execute("MoveToStartPosition")コマンド	28
2.2.3.44. CaoRobot::Execute("Calibration")コマンド	28
2.2.3.45. CaoRobot::Execute("SwitchPIOModbus")コマンド	29
2.2.3.46. CaoRobot::Execute("Stop")コマンド	29
2.2.3.47. CaoRobot::Execute("SwitchAxisMove")コマンド	29
2.2.3.48. CaoRobot::Execute("ProgramHome")コマンド	30
2.2.3.49. CaoRobot::Execute("SearchStop")コマンド	30
2.2.3.50. CaoRobot::Execute("ForceProgramStop")コマンド	30
2.2.3.51. CaoRobot::Execute("ProgramStart")コマンド	31
2.2.3.52. CaoRobot::Execute("WriteSingleRegister")コマンド	31
2.2.3.53. CaoRobot::Execute("WriteMultipleRegisters")コマンド	32
2.2.3.54. CaoRobot::Execute("MovePTP")コマンド	32
2.2.3.55. CaoRobot::Execute("WritePositionData")コマンド	33
2.2.4. CaoRobot::AddVariable メソッド	34
2.2.5. CaoRobot::get_VariableNames メソッド	34
2.2.6. CaoVariable::get_Value プロパティ	34
2.2.7. CaoVariable::put_Value プロパティ	34
2.3. 変数一覧	35
2.3.1. コントローラクラス	35
2.3.1.1. @MAKER_NAME	36
2.3.1.2. @VERSION	36
2.3.2. ロボットクラス	37
2.3.2.1. REGISTER<??>	39
2.3.2.2. @ALARM_INFO	39
2.3.2.3. POSITION_DATA<??>	39
2.3.2.4. @MOVE_COUNT	40
2.3.2.5. @MOVE_DISTANCE	40
2.3.2.6. @CURRENT_TIME	40
2.3.2.7. @FAN_DRIVE_TIME	40
2.3.2.8. @CURRENT_POSITION	40
2.3.2.9. @ALARM_CODE	40
2.3.2.10. @IO_INPUT	41
2.3.2.11. @IO_OUTPUT	41
2.3.2.12. @DSS1	41
2.3.2.13. @DSS2	41
2.3.2.14. @DSSE	41
2.3.2.15. @STAT	41
2.3.2.16. @SPEED	42
2.3.2.17. @VALUE	42
2.3.2.18. @DEVIATION	42

2.3.2.19. @INTEGRATION_TIME	42
2.3.2.20. @SPECIAL_INPUT	42
2.3.2.21. @ZONE	42
2.3.2.22. @COMPLETE_POSITION_NO	43
2.3.2.23. @SSSE	43
2.3.2.24. @LOAD	43
2.3.2.25. @LOAD_LEVEL	43
2.3.2.26. @PRESS_PROGRAM_ALARM_CODE	43
2.3.2.27. @ALARM_PRESS_PROGRAM_NO	43
2.3.2.28. @PRESS_PROGRAM_STATUS	44
2.3.2.29. @PRESS_PROGRAM_JUDGE	44
2.4. エラーコード	45
3. サンプルプログラム	48

1. はじめに

本書は、アイエイアイ社製ポジションコントローラの PCON シリーズの CA0 プロバイダである、PCON プロバイダのユーザーズガイドです。

本書は、この PCON プロバイダの機能と実装されているメソッドについて説明します。

PCON プロバイダでは Modbus/RTU プロトコルを使用するため、Modbus.X プロバイダをラップして作成されています。

1.1. ライセンスの追加

本プロバイダを使用可能にするには ORiN2 SDK をインストール後、別途「IAI ポジショナータイプコントローラプロバイダ」ライセンスを入力する必要があります。評価用にインストールする場合は下記のライセンスキーをご使用ください。

IIEN-M1WA-JTYV-AFWF （評価用 3 ヶ月）

下記に「IAI ポジショナータイプコントローラプロバイダ」ライセンスの追加手順を示します。

1. CaoConfig を起動し、[Cao Provider] タブを選択する
2. Provider List から [IAI PCON CAO Provider] 項目を選択する
3. License 項目の [...] ボタンをクリックする
4. ORiN2 License Manager で [Add] ボタンをクリックする
5. 入手したライセンスキーを入力後、[OK] ボタンをクリックする
6. [Close] ボタンをクリックし、CaoConfig を終了する

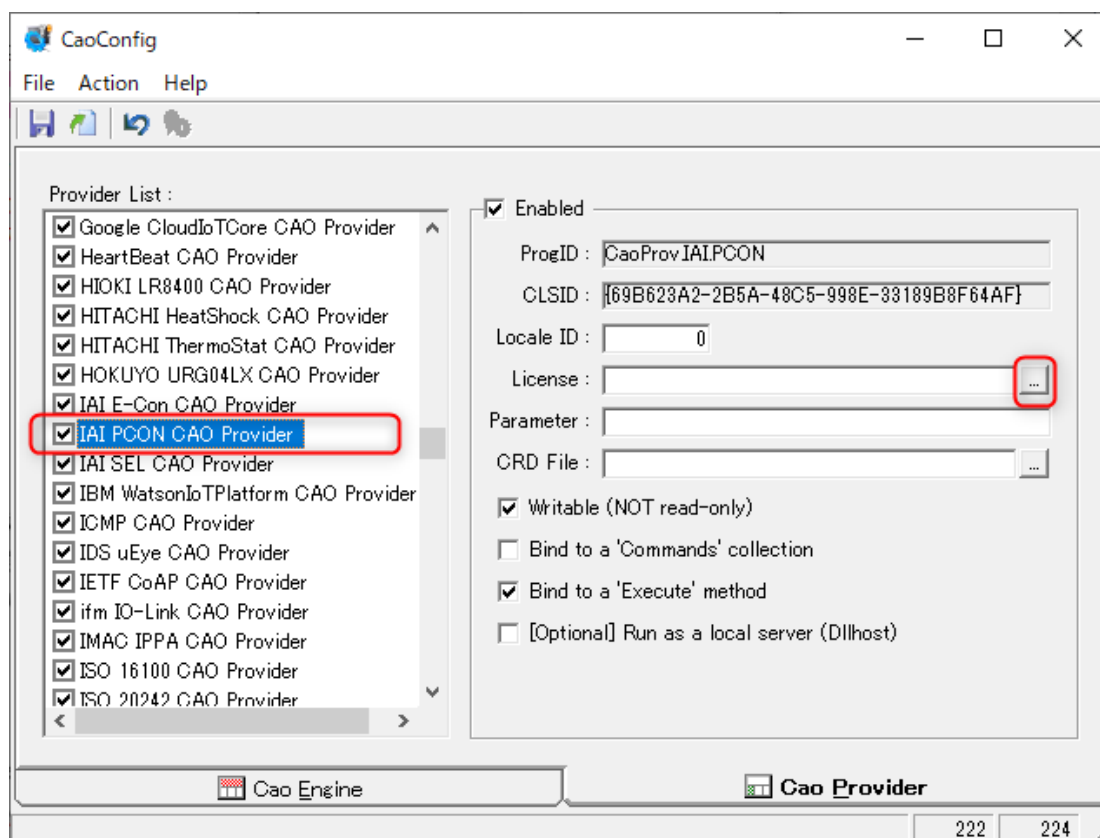


図 1-1 「IAI ポジショナータイプコントローラプロバイダ」ライセンス追加

2. プロバイダの概要

2.1. 概要

PCON プロバイダは、COM 通信 もしくは TCP/IP 通信によってデータを読み書きします。PCON 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 PCON プロバイダ

ファイル名	CaoProvIAIPCON.dll
ProgID	CaoProv.IAI.PCON
レジストリ登録	regsvr32 CaoProvIAIPCON.dll
レジストリ登録の抹消	regsvr32 /u CaoProvIAIPCON.dll

また、PCON プロバイダと PCON デバイスそれぞれの対応を表した図が下図 2-1 となります。

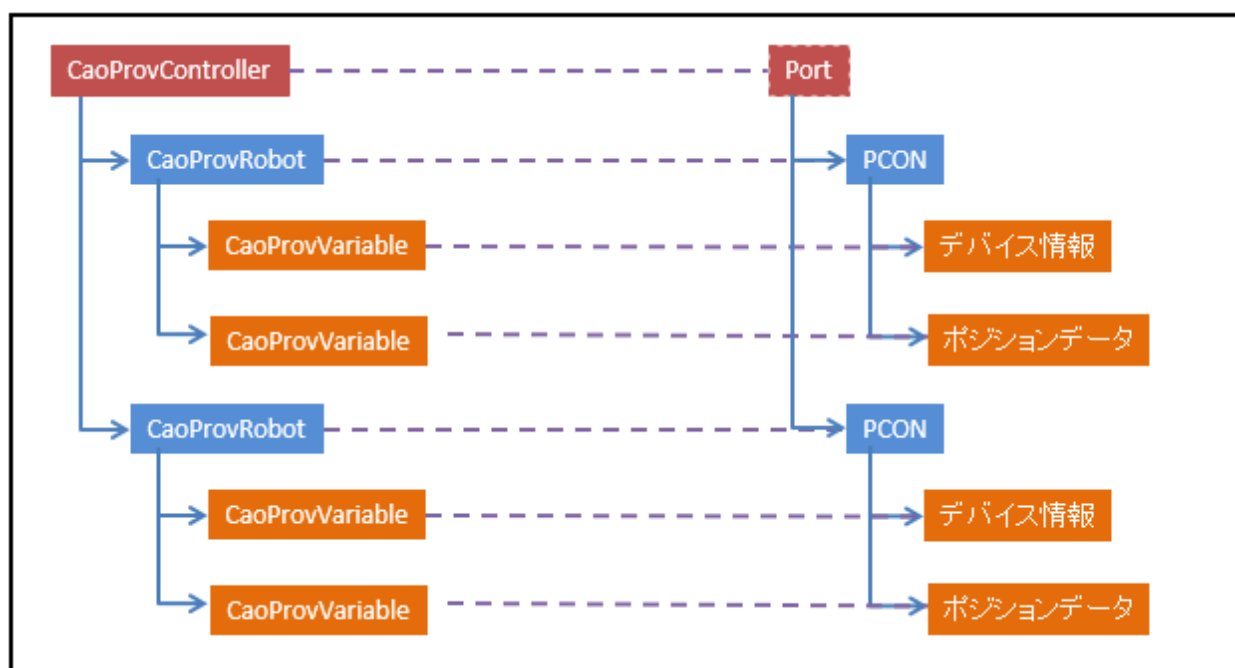


図 2-1 CaoProvController と PCON デバイスの対応図

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

Controller オブジェクトの生成時に PCON デバイスに接続するために必要なオプションを指定します。

以下に、AddController の仕様を示します。

書式

AddController

```
(
    "<コントローラ名>",           // コントローラ名(任意)
    "CaoProv. IAI. PCON",         // プロバイダ名(固定)
    "<マシン名>",                 // プロバイダ実行マシン名(未使用)
    "<オプション>"                // オプション文字列
)
```

以下にオプション文字列に指定する文字列を示します。

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

オプション	必須	説明	値範囲	デフォルト値
CONN= COM:<接続先 COM ポート番号>[:<ボーレート>[:<パリティ>:<データビット数>:<ストップビット数>]]	○ ¹	接続先の COM ポート番号を環境に合わせて指定します。	-----	ボーレート: 38400 パリティ: NONE データビット数: 8bit ストップビット数: 1bit
CONN= ETH:<接続先 IP>[:<接続先ポート>]	○ ¹	接続先の IP アドレスおよびポートを環境に合わせて指定します。	-----	ポート: 502

¹ COM か ETH はいずれかが必須

オプション	必須	説明	値範囲	デフォルト値
PacketType=<パケットパラメータ> ²	-	Modbus 通信プロトコルデータタイプの設定をします。 0 : RTU (デフォルト) 1 : ASCII	0 - 1	0
Sync=True / False	-	同期モードの設定を行います。 True : 同期モード (デフォルト) 調歩同期式 False : 非同期モード	True / False	True
Timeout=<送受信タイムアウト時間>	-	送受信タイムアウト時間 [ms] を設定します。	1 - 100000	1000
Retry=<リトライ回数>	-	送受信時の通信リトライ回数の設定をします。	0 - 10	0
RtsTransmitDelayTime=<送受信切替遅延時間> ³	-	RTS 信号 ⁴ による送受信切替遅延時間 [ms] の設定をします。 0 : RTS 信号は常に ON (デフォルト) 1~100000 : RTS 信号による送受信回路制御あり	0 - 100000	0
PollDelayTime=<ポーリング遅延時間>	-	ポーリング遅延時間 [ms] の設定をします。	0 - 100000	0

RtsTransmitDelayTime オプションについて

・主に伝送モードが半二重の場合で、ソフトウェアによる送受信回路の切り替えが必要とされるハードウェア構成の場合に利用され、“1”ms 以上に設定することにより、RTS 信号により送受信の切り替えを行います。

・オプションの値を“1”ms 以上に設定されている場合、以下のようなシーケンスで通信を行います。
送信直前に RTS ON → データ送信開始 → 送信完了 → 設定遅延時間経過後に RTS OFF

本プロバイダ内部における上記の送信完了の判定は、通信デバイスドライバからの送信完了通知を起点にしているため、実際の伝送路上の送信完了時期よりも早期になります。(実際の送信完了までの遅延時間の算出は、通信ハードウェア側の FIFO を使用した場合、ベンダ依存となるため、FIFO 未使用で遅延時間を算出されることを推奨します)

² Conn オプションで指定した通信デバイスが“eth”時、TCP/IP の内部データタイプは RTU 固定であることから、このオプションは無視されます。

³ Conn オプションで指定した通信デバイスが“eth”時、このオプションは無視されます。

⁴ RTS 信号は、送信要求 (Request To Send) 信号のことです。

2.2.2. CaoController::AddRobot メソッド

Robot オブジェクトの生成時に PCON に接続された Robot デバイスに接続するために必要なオプションを指定します。

以下に、AddRobot の仕様を示します。

書式

AddRobot

```
(
    "<ロボット名>",          // ロボット名(任意)
    "<オプション>"          // オプション文字列
)
```

以下にオプション文字列に指定する文字列を示します。

表 2-3 CaoController::AddRobot のオプション文字列

オプション	必須	説明	値範囲	デフォルト値
UnitAddress=<機器アドレス>	－	通信先の機器アドレス (com 時) 又は ユニット識別子 (eth 時) を設定しま す。 com 時: サーバ機器アドレス ⁵ eth 時: ユニット識別子	0 - 255	com 時:1 eth 時:0
Type=<機種種別>	－	通信先の機器の機種を設定します。 0 : 下記以外 1 : SCON-CA/CAL/CB 2 : PCON-CA/CFA, ACON-CA/CB	0 - 2	0

⁵ 0 指定時はブロードキャスト通信を行います。その場合、接続されている全ての機器に同一内容のコマンドを実行します。

2.2.3. CaoRobot::Execute メソッド

CaoRobot にて Execute を使用することによってデータの読み書きができるものがあります。

以下に、Execute の仕様を示します。

書式

Execute

```
(
    "<メソッド名>", // メソッド名
    "<引数>" // 引数
)
```

表 2-4 CaoRobot::Execute コマンド一覧

コマンド名	説明	記号	ブロードキャスト対応	リンク
データ、ステータス読み取り (FC:03)				
ReadMultipleRegisters	レジスタの値の取得を実行します。	—	—	P. 16
ReadAlarmInfo	最後に発生したアラーム情報の取得を実行します。	ALAO ALCO ALTO	—	P. 16
ReadPositionData	ポジションデータの取得を実行します。	PCMD, INP, VCMD, ZNMP, ZNLP, ACMD, DCMD, PPOW , LPOW, CTLF	—	P. 16
ReadMoveCount	通算移動回数の取得を実行します。	TLMC	—	P. 17
ReadMoveDistance	通算走行距離の取得を実行します。	ODOM	—	P. 17
ReadCurrentTime	現在時刻の取得を実行します。	TIMN	—	P. 17
ReadFanDriveTime	ファンの通算駆動時間の取得を実行します。	TFAN	—	P. 18
ReadCurrentPosition	現在位置の取得を実行します。	PNOW	—	P. 18
ReadCurrentAlarmCode	現在発生アラームコードの取得を実行します。	ALMC	—	P. 18
ReadIOPortInputSignal	I/O ポート入力信号の値の取得を実行します。	DIPM	—	P. 18
ReadIOPortOutputSignal	I/O ポート出力信号の値の取得を実行します。	DOPM	—	P. 19

コマンド名	説明	記号	ブロードキャスト対応	リンク
ReadDSS1	デバイスステータスレジスタ 1 の値の取得を実行します。	DSS1	—	P. 19
ReadDSS2	デバイスステータスレジスタ 2 の値の取得を実行します。	DSS2	—	P. 19
ReadDSSE	拡張デバイスステータスレジスタの値の取得を実行します。	DSSE	—	P. 19
ReadSTAT	システムステータスレジスタの値の取得を実行します。	STAT	—	P. 20
ReadCurrentSpeed	モータ実速度のモニタデータの取得を実行します。	VNOW	—	P. 20
ReadCurrentValue	モータ電流のモニタデータの取得を実行します。	CNOW	—	P. 20
ReadDeviation	1ms 周期毎の位置指令値とフィードバック値(実位置)の偏差量の取得を実行します。	DEVI	—	P. 20
ReadIntegrationTime	電源投入後の積算時間の取得を実行します。	STIM	—	P. 21
ReadSpecialInputPort	特殊入力ポートモニタレジスタの値の取得を実行します。	SIPM	—	P. 21
ReadZoneStatus	ゾーンステータスレジスタの値の取得を実行します。	ZONS	—	P. 21
ReadCompletePositionNo	ポジション番号ステータスレジスタの値の取得を実行します。	POSS	—	P. 22
ReadSSSE	拡張システムステータスレジスタの値の取得を実行します。	SSSE	—	P. 22
ReadLoad	ロードセル測定値のモニタデータの値の取得を実行します。	FBFC	—	P. 22
ReadLoadLevel	現在のモータへの負荷レベルの比率の取得を実行します。	OLLV	—	P. 22
ReadPressProgramAlarmCode	プレスプログラムアラームコードの値の取得を実行します。	ALMP	—	P. 23
ReadAlarmPressProgramNo	アラームが発生しているプレスプログラム番号の取得を実行します。	ALMP	—	P. 23
ReadPressProgramStatus	プレスプログラムステータスレジスタの	PPST	—	P. 23

コマンド名	説明	記号	ブロードキャスト対応	リンク
	値の取得を実行します。			
ReadPressProgramJudgeStatus	プレスプログラム判定ステータスレジスタの値の取得を実行します。	PPJD	-	P. 23
動作指令 及び、データ書き換え (FC:05)				
WriteSingleDiscreteOutput	スレーブ D0 の ON/OFF を実行します。	-	○	P. 24
SwitchSafetySpeed	セーフティ速度の有効/無効の切り替えを実行します。	SFTY	○	P. 24
SwitchServo	サーボの ON/OFF を実行します。	SON	○	P. 24
ResetAlarm	アラームリセットを実行します。	ALRS	○	P. 24
ReleaseBreak	ブレーキの強制解除を実行します。	BKRL	○	P. 25
Pause	一時停止を実行します。	STP	○	P. 25
Home	原点復帰を実行します。	HOME	○	P. 26
StartPosition	ポジション番号の指定位置への移動を実行します。	CSTR	○	P. 26
SwitchJogInching	ジョグ/インチングの切り替えを実行します。	JISL	○	P. 27
SwitchMode	教示モード/通常運転モードの切り替えを実行します。	MOD	○	P. 27
GetCurrentPositionData	ポジションデータの取り込みを実行します。	TEAC	○	P. 27
JogPlus	反原点方向へのジョグ/インチング動作を実行します。	JOG+	○	P. 27
JogMinus	原点方向へのジョグ/インチング動作を実行します。	JOG-	○	P. 28
MoveToStartPosition	スタートポジションへの移動を実行します。	ST0~ST7	○	P. 28
Calibration	専用ロードセルのキャリブレーションを実行します。	CLBR	○	P. 28
SwitchPIOModbus	Modbus 指令の有効/無効の切り替えを実行します。	PMSL	○	P. 29
Stop	減速停止を実行します。	STOP	○	P. 29
SwitchAxisMove	軸動作の有効/無効の切り替えを実行します。	ENMV	○	P. 29

コマンド名	説明	記号	ブロードキャスト対応	リンク
ProgramHome	プログラム原点への移動を実行します。	PHOM	○	P. 30
SearchStop	探し動作完了後、プレスプログラムを終了する/しないの切り替えを実行します。	SSTP	○	P. 30
ForceProgramStop	プレスプログラムの強制終了を実行します。	FPST	○	P. 30
ProgramStart	プレスプログラムを実行します。	PSTR	○	P. 31
動作指令 及び、データ書き換え (FC:05)				
WriteSingleRegister	指定アドレスレジスタへの書き込みを実行します。	-	○	P. 31
位置決めデータの直接書き込み (FC:10)				
WriteMultipleRegisters	指定アドレスから連続したレジスタへの書き込みを実行します。	-	○	P. 32
MovePTP	直値移動を実行します。	PCMD, INP, VCMD, ACMD, PPOW, CTLF	○	P. 32
WritePositionData	ポジションデータの書き込みを実行します。	PCMD, INP, VCMD, ZNMP, ZNLP, ACMD, DCMD, PPOW, LPOOW, CTLF	○	P. 33

2.2.3.1. CaoRobot::Execute("ReadMultipleRegisters") コマンド

レジスタの値の取得を実行します。

引数：

VT_ARRAY VT_I4		
0	VT_I4	読み込みレジスタの開始アドレスを指定します。
1	VT_I4	読み込みレジスタ数を指定します。

戻り値：

VT_ARRAY VT_UI2		
n	VT_UI2	開始アドレスから連続した読み込みレジスタの値が返却されます。

使用例：

```
Dim vArray As Variant
vArray = caoRobot.Execute("ReadMultipleRegisters", Array(1284, 2))
// vArray にアドレス 0x0504 (1284), 0x0505 (1285) の値を取得
```

2.2.3.2. CaoRobot::Execute("ReadAlarmInfo") コマンド

最後に発生したアラーム情報の取得を実行します。

引数：なし

戻り値：

VT_ARRAY VT_VARIANT		
0	VT_UI2	アラーム詳細コードが返却されます。
1	VT_UI2	アラームアドレスが返却されます。
2	VT_UI4	アラームコードが返却されます。
3	VT_UI4	アラーム発生時刻が返却されます。(RTC 無効の場合は電源投入後経過時間[ms])

使用例：

```
Dim vArray As Variant
vArray = caoRobot.Execute("ReadAlarmInfo")
```

2.2.3.3. CaoRobot::Execute("ReadPositionData") コマンド

ポジションデータの取得を実行します。

引数：

VT_ARRAY VT_UI2		
0	VT_UI2	読み込み開始ポジション No. を指定します。
1	VT_UI2	読み込み対象レコード数を指定します。

戻り値：

VT_ARRAY VT_VARIANT		
-----------------------	--	--

n	VT_ARRAY VT_VARIANT		
	0	VT_I4	目標位置が返却されます。(単位 : 0.01mm)
	1	VT_UI4	位置決め幅が返却されます。(単位 : 0.01mm)
	2	VT_UI4	速度指令が返却されます。(単位 : 0.01mm/s)
	3	VT_I4	個別ゾーン境界+側が返却されます。(単位 : 0.01mm)
	4	VT_I4	個別ゾーン境界-側が返却されます。(単位 : 0.01mm)
	5	VT_UI2	加速度指令が返却されます。(単位 : 0.01G)
	6	VT_UI2	減速度指令が返却されます。(単位 : 0.01G)
	7	VT_UI2	押付け時電流制限値が返却されます。(100%=0xFF)
	8	VT_UI2	負荷電流閾値が返却されます。(100%=0xFF)
	9	VT_UI2	制御フラグ指定が返却されます。

使用例 :

```
Dim vArray As Variant
vArray = caoRobot.Execute("ReadPositionData", Array(10, 2))
// vArray にポジション No. 10, 11 のポジションデータの値を取得
```

2.2.3.4. CaoRobot::Execute("ReadMoveCount") コマンド

通算移動回数の取得を実行します。

引数 : なし

戻り値 :

VT_UI4	通算移動回数が返却されます。
--------	----------------

使用例 :

```
Dim lVal As Long
lVal = caoRobot.Execute("ReadMoveCount")
```

2.2.3.5. CaoRobot::Execute("ReadMoveDistance") コマンド

通算走行距離の取得を実行します。

引数 : なし

戻り値 :

VT_UI4	通算走行距離が返却されます。(単位 : 1m)
--------	-------------------------

使用例 :

```
Dim lVal As Long
lVal = caoRobot.Execute("ReadMoveDistance")
```

2.2.3.6. CaoRobot::Execute("ReadCurrentTime") コマンド

現在時刻の取得を実行します。

引数：なし

戻り値：

VT_UI4	現在時刻が返却されます。
--------	--------------

使用例：

Dim IVal As Long

IVal = caoRobot.Execute("ReadCurrentTime")

2.2.3.7. CaoRobot::Execute("ReadFanDriveTime") コマンド

ファンの通算駆動時間の取得を実行します。

引数：なし

戻り値：

VT_UI4	ファンの通算駆動時間が返却されます。(単位：1s)
--------	---------------------------

使用例：

Dim IVal As Long

IVal = caoRobot.Execute("ReadFanDriveTime")

2.2.3.8. CaoRobot::Execute("ReadCurrentPosition") コマンド

現在位置の取得を実行します。

引数：なし

戻り値：

VT_I4	現在位置が返却されます。(単位：0.01mm)
-------	-------------------------

使用例：

Dim IVal As Long

IVal = caoRobot.Execute("ReadCurrentPosition")

2.2.3.9. CaoRobot::Execute("ReadCurrentAlarmCode") コマンド

現在発生アラームコードの取得を実行します。

引数：なし

戻り値：

VT_UI2	現在発生アラームコードが返却されます。
--------	---------------------

使用例：

Dim IVal As Long

IVal = caoRobot.Execute("ReadCurrentAlarmCode")

2.2.3.10. CaoRobot::Execute("ReadIOPortInputSignal") コマンド

I/O ポート入力信号の値の取得を実行します。

引数：なし

戻り値：

VT_UI2	I/O ポート入力信号の値が返却されます。
--------	-----------------------

使用例：

```
Dim IVal As Long
```

```
IVal = caoRobot.Execute("ReadIOPortInputSignal")
```

2.2.3.11. CaoRobot::Execute("ReadIOPortOutputSignal") コマンド

I/O ポート出力信号の値の取得を実行します。

引数：なし

戻り値：

VT_UI2	I/O ポート出力信号の値が返却されます。
--------	-----------------------

使用例：

```
Dim IVal As Long
```

```
IVal = caoRobot.Execute("ReadIOPortOutputSignal")
```

2.2.3.12. CaoRobot::Execute("ReadDSS1") コマンド

デバイスステータスレジスタ 1 の値の取得を実行します。

引数：なし

戻り値：

VT_UI2	デバイスステータスレジスタ 1 の値が返却されます。
--------	----------------------------

使用例：

```
Dim IVal As Long
```

```
IVal = caoRobot.Execute("ReadDSS1")
```

2.2.3.13. CaoRobot::Execute("ReadDSS2") コマンド

デバイスステータスレジスタ 2 の値の取得を実行します。

引数：なし

戻り値：

VT_UI2	デバイスステータスレジスタ 2 の値が返却されます。
--------	----------------------------

使用例：

```
Dim IVal As Long
```

```
IVal = caoRobot.Execute("ReadDSS2")
```

2.2.3.14. CaoRobot::Execute("ReadDSSE") コマンド

拡張デバイスステータスレジスタの値の取得を実行します。

引数：なし

戻り値：

VT_UI2	拡張デバイスステータスレジスタの値が返却されます。
--------	---------------------------

使用例：

Dim IVal As Long

IVal = caoRobot.Execute("ReadDSSE")

2. 2. 3. 15. CaoRobot::Execute("ReadSTAT") コマンド

システムステータスレジスタの値の取得を実行します。

引数：なし

戻り値：

VT_UI2	システムステータスレジスタの値が返却されます。
--------	-------------------------

使用例：

Dim IVal As Long

IVal = caoRobot.Execute("ReadSTAT")

2. 2. 3. 16. CaoRobot::Execute("ReadCurrentSpeed") コマンド

モータ実速度のモニタデータの取得を実行します。

引数：なし

戻り値：

VT_I4	モータ実速度のモニタデータが返却されます。(単位：0.01mm/s)
-------	------------------------------------

使用例：

Dim IVal As Long

IVal = caoRobot.Execute("ReadCurrentSpeed")

2. 2. 3. 17. CaoRobot::Execute("ReadCurrentValue") コマンド

モータ電流のモニタデータの取得を実行します。

引数：なし

戻り値：

VT_UI4	モータ電流のモニタデータが返却されます。(単位：1mA)
--------	------------------------------

使用例：

Dim IVal As Long

IVal = caoRobot.Execute("ReadCurrentValue")

2. 2. 3. 18. CaoRobot::Execute("ReadDeviation") コマンド

1ms 周期毎の位置指令値とフィードバック値(実位置)の偏差量の取得を実行します。

引数 : なし

戻り値 :

VT_UI4	1ms 周期毎の位置指令値とフィードバック値(実位置)の偏差量が返却されます。(単位 : Pulse)
--------	---

使用例 :

```
Dim IVal As Long
```

```
IVal = caoRobot.Execute("ReadDeviation")
```

2.2.3.19. CaoRobot::Execute("ReadIntegrationTime") コマンド

電源投入後の積算時間の取得を実行します。

引数 : なし

戻り値 :

VT_UI4	電源投入時からの積算時間が返却されます。(単位 : ms)
--------	-------------------------------

使用例 :

```
Dim IVal As Long
```

```
IVal = caoRobot.Execute("ReadIntegrationTime")
```

2.2.3.20. CaoRobot::Execute("ReadSpecialInputPort") コマンド

特殊入力ポートモニタレジスタの値の取得を実行します。

引数 : なし

戻り値 :

VT_UI2	特殊入力ポートモニタレジスタの値が返却されます。
--------	--------------------------

使用例 :

```
Dim IVal As Long
```

```
IVal = caoRobot.Execute("ReadSpecialInputPort")
```

2.2.3.21. CaoRobot::Execute("ReadZoneStatus") コマンド

ゾーンステータスレジスタの値の取得を実行します。

引数 : なし

戻り値 :

VT_UI2	ゾーンステータスレジスタの値が返却されます。
--------	------------------------

使用例 :

```
Dim IVal As Long
```

```
IVal = caoRobot.Execute("ReadZoneStatus")
```

2. 2. 3. 22. CaoRobot::Execute("ReadCompletePositionNo") コマンド

ポジション番号ステータスレジスタの値の取得を実行します。

引数：なし

戻り値：

VT_UI2	ポジション番号ステータスレジスタの値が返却されます。
--------	----------------------------

使用例：

```
Dim IVal As Long
```

```
IVal = caoRobot.Execute("ReadCompletePositionNo")
```

2. 2. 3. 23. CaoRobot::Execute("ReadSSSE") コマンド

拡張システムステータスレジスタの値の取得を実行します。

引数：なし

戻り値：

VT_UI2	拡張システムステータスレジスタの値が返却されます。
--------	---------------------------

使用例：

```
Dim IVal As Long
```

```
IVal = caoRobot.Execute("ReadSSSE")
```

2. 2. 3. 24. CaoRobot::Execute("ReadLoad") コマンド

ロードセル測定値のモニタデータの値の取得を実行します。

引数：なし

戻り値：

VT_UI4	ロードセル測定値のモニタデータの値が返却されます。（単位：0.01N）
--------	-------------------------------------

使用例：

```
Dim IVal As Long
```

```
IVal = caoRobot.Execute("ReadLoad")
```

2. 2. 3. 25. CaoRobot::Execute("ReadLoadLevel") コマンド

現在のモータへの負荷レベルの比率の取得を実行します。

引数：なし

戻り値：

VT_UI4	現在のモータへの負荷レベルの比率が返却されます。（単位：1%）
--------	---------------------------------

使用例：

```
Dim IVal As Long
```

```
IVal = caoRobot.Execute("ReadLoadLevel")
```

2. 2. 3. 26. CaoRobot::Execute("ReadPressProgramAlarmCode") コマンド

プレスプログラムアラームコードの値の取得を実行します。

引数：なし

戻り値：

VT_UI2	プレスプログラムの現在発生アラームコードが返却されます。
--------	------------------------------

使用例：

```
Dim IVal As Long
```

```
IVal = caoRobot.Execute("ReadPressProgramAlarmCode")
```

2. 2. 3. 27. CaoRobot::Execute("ReadAlarmPressProgramNo") コマンド

アラームが発生しているプレスプログラム番号の取得を実行します。

引数：なし

戻り値：

VT_UI2	アラームが発生しているプレスプログラム番号が返却されます。
--------	-------------------------------

使用例：

```
Dim IVal As Long
```

```
IVal = caoRobot.Execute("ReadAlarmPressProgramNo")
```

2. 2. 3. 28. CaoRobot::Execute("ReadPressProgramStatus") コマンド

プレスプログラムステータスレジスタの値の取得を実行します。

引数：なし

戻り値：

VT_UI2	プレスプログラムステータスレジスタの値が返却されます。
--------	-----------------------------

使用例：

```
Dim IVal As Long
```

```
IVal = caoRobot.Execute("ReadPressProgramStatus")
```

2. 2. 3. 29. CaoRobot::Execute("ReadPressProgramJudgeStatus") コマンド

プレスプログラム判定ステータスレジスタの値の取得を実行します。

引数：なし

戻り値：

VT_UI2	プレスプログラム判定ステータスレジスタの値が返却されます。
--------	-------------------------------

使用例：

```
Dim IVal As Long
```

```
IVal = caoRobot.Execute("ReadPressProgramJudgeStatus")
```

2. 2. 3. 30. CaoRobot::Execute("WriteSingleDiscreteOutput") コマンド

スレーブ D0 の ON/OFF を実行します。

引数：

VT_ARRAY VT_VARIANT		
0	VT_I4	書き込み対象アドレスを指定します。
1	VT_BOOL	書き込む値を指定します。

戻り値：なし

使用例：

```
caoRobot.Execute("WriteSingleDiscreteOutput", Array(1027, True))
```

// アドレス 0x0403 (1027) に ON (True) を書き込み

2. 2. 3. 31. CaoRobot::Execute("SwitchSafetySpeed") コマンド

セーフティ速度の有効/無効の切り替えを実行します。

引数：

VT_BOOL	セーフティ速度の有効/無効を指定します。 True：セーフティ速度有効 False：セーフティ速度無効
---------	---

戻り値：なし

使用例：

```
caoRobot.Execute("SwitchSafetySpeed", True)
```

// セーフティ速度有効を指定

2. 2. 3. 32. CaoRobot::Execute("SwitchServo") コマンド

サーボの ON/OFF を実行します。

引数：

VT_BOOL	サーボの ON/OFF を指定します。 True：サーボ ON False：サーボ OFF
---------	---

戻り値：なし

使用例：

```
caoRobot.Execute("SwitchServo", True)
```

// サーボ ON を指定

2. 2. 3. 33. CaoRobot::Execute("ResetAlarm") コマンド

アラームリセットを実行します。

引数：

VT_BOOL	アラームリセットの状態を指定します。 True：アラームリセット実行 False：通常 (通常→アラームリセット実行で実行されます。)
---------	--

戻り値：なし

使用例：

```
caoRobot.Execute("ResetAlarm", False)
```

```
delay 100
```

```
// ブロードキャストの場合、コマンド間に無通信時間を設ける
```

```
caoRobot.Execute("ResetAlarm", True)
```

```
// 通常→アラームリセット実行を指定
```

2.2.3.34. CaoRobot::Execute("ReleaseBreak") コマンド

ブレーキの強制解除を実行します。

引数：

VT_BOOL	ブレーキの強制解除の状態を指定します。 True：ブレーキ強制解除 False：通常
---------	--

戻り値：なし

使用例：

```
caoRobot.Execute("ReleaseBreak", True)
```

```
// ブレーキ強制解除を指定
```

2.2.3.35. CaoRobot::Execute("Pause") コマンド

一時停止を実行します。

引数：

VT_BOOL	一時停止の状態を指定します。 True：一時停止指令 False：通常
---------	---

戻り値：なし

使用例：

```
caoRobot.Execute("Pause", True)
```

```
// 一時停止指令を指定
```

2. 2. 3. 36. CaoRobot::Execute("Home") コマンド

原点復帰を実行します。

引数：

VT_BOOL	原点復帰の状態を指定します。 True：原点復帰実行 False：通常 (通常→原点復帰実行で実行されます。)
---------	--

戻り値：なし

使用例：

```
caoRobot.Execute("Home", False)
```

```
delay 100
```

```
// ブロードキャストの場合、コマンド間に無通信時間を設ける
```

```
caoRobot.Execute("Home", True)
```

```
// 通常→原点復帰実行を指定
```

2. 2. 3. 37. CaoRobot::Execute("StartPosition") コマンド

ポジション番号指定レジスタ内のポジション番号の指定位置への移動を実行します。

引数：

VT_BOOL	ポジション移動の状態を指定します。 True：ポジションスタート指令 False：通常 (通常→ポジションスタート指令で実行されます。)
---------	---

戻り値：なし

使用例：

```
caoRobot.Execute("WriteSingleRegister", Array(3328, 1))
```

```
// アドレス 0x0D03 (3331) に 0x0001 (1) を書き込み (ポジション番号指定)
```

```
caoRobot.Execute("StartPosition", False)
```

```
delay 100
```

```
// ブロードキャストの場合、コマンド間に無通信時間を設ける
```

```
caoRobot.Execute("StartPosition", True)
```

```
// 通常→ポジションスタート指令を指定
```

```
delay 100
```

```
// ブロードキャストの場合、コマンド間に無通信時間を設ける
```

```
caoRobot.Execute("StartPosition", False)
```

```
// ON のままでは位置決め幅内に入っても完了ポジションが出力されないため、OFF
```

2. 2. 3. 38. CaoRobot::Execute("SwitchJogInching") コマンド

ジョグ/インチングの切り替えを実行します。

引数：

VT_BOOL	ジョグ/インチング動作状態を指定します。 True：インチング動作状態 False：ジョグ動作状態
---------	---

戻り値：なし

使用例：

```
caoRobot.Execute("SwitchJogInching", True)
// インチング動作状態を指定
```

2. 2. 3. 39. CaoRobot::Execute("SwitchMode") コマンド

教示モード/通常運転モードの切り替えを実行します。

引数：

VT_BOOL	通常運転/教示モードを指定します。 True：教示モード False：通常運転モード
---------	--

戻り値：なし

使用例：

```
caoRobot.Execute("SwitchMode", True)
// 教示モードを指定
```

2. 2. 3. 40. CaoRobot::Execute("GetCurrentPositionData") コマンド

ポジションデータの取り込みを実行します。

引数：

VT_BOOL	ポジションデータ取り込みの状態を指定します。 True：ポジションデータ取り込み指令 False：通常
---------	---

戻り値：なし

使用例：

```
caoRobot.Execute("GetCurrentPositionData", True)
// ポジションデータ取り込み指令を指定
```

2. 2. 3. 41. CaoRobot::Execute("JogPlus") コマンド

反原点方向へのジョグ/インチング動作を実行します。

引数：

VT_BOOL	反原点方向へのジョグ/イン칭動作の状態を指定します。 True : ジョグ+指令 False : 通常
---------	---

戻り値 : なし

使用例 :

```
caoRobot.Execute("JogPlus", True)
```

```
// ジョグ+指令を指定
```

2. 2. 3. 42. CaoRobot::Execute("JogMinus") コマンド

原点方向へのジョグ/イン칭動作を実行します。

引数 :

VT_BOOL	原点方向へのジョグ/イン칭動作の状態を指定します。 True : ジョグ-指令 False : 通常
---------	--

戻り値 : なし

使用例 :

```
caoRobot.Execute("JogMinus", True)
```

```
// ジョグ-指令を指定
```

2. 2. 3. 43. CaoRobot::Execute("MoveToStartPosition") コマンド

指定されたスタートポジションへの移動を実行します。

引数 :

VT_ARRAY VT_VARIANT		
0	VT_UI2	スタートポジション No. を指定します。
1	VT_BOOL	動作指令の状態を指定します。 True : 動作指令 ON False : 動作指令 OFF

戻り値 : なし

使用例 :

```
caoRobot.Execute("MoveToStartPosition", Array(1, True))
```

```
// ポジション No. 1 への動作指令 ON を指定
```

2. 2. 3. 44. CaoRobot::Execute("Calibration") コマンド

専用ロードセルのキャリブレーションを実行します。

引数 :

VT_BOOL	専用ロードセルのキャリブレーションの実行状態を指定します。
---------	-------------------------------

	True : キャリブレーション実行指令 False : 通常運転時
--	---------------------------------------

戻り値 : なし

使用例 :

```
caoRobot.Execute("Calibration", True)
```

```
// キャリブレーション実行指令を指定
```

2. 2. 3. 45. CaoRobot::Execute("SwitchPIOModbus") コマンド

Modbus 指令の有効/無効の切り替えを実行します。

引数 :

VT_BOOL	Modbus 指令の有効/無効モードを指定します。 True : Modbus 指令有効 False : Modbus 指令無効
---------	--

戻り値 : なし

使用例 :

```
caoRobot.Execute("SwitchPIOModbus", True)
```

```
// Modbus 指令有効を指定
```

2. 2. 3. 46. CaoRobot::Execute("Stop") コマンド

減速停止を実行します。

引数 :

VT_BOOL	減速停止指令の状態を指定します。 True : 減速停止指令 False : 通常
---------	---

戻り値 : なし

使用例 :

```
caoRobot.Execute("Stop", True)
```

```
// 減速停止指令を指定
```

2. 2. 3. 47. CaoRobot::Execute("SwitchAxisMove") コマンド

軸動作の有効/無効の切り替えを実行します。

引数 :

VT_BOOL	軸動作の有効/無効を指定します。 True : 軸動作有効 False : 軸動作無効
---------	---

戻り値 : なし

使用例：

```
caoRobot.Execute("SwitchAxisMove", True)
// 軸動作有効を指定
```

2.2.3.48. CaoRobot::Execute("ProgramHome") コマンド

プログラム原点への移動を実行します。

引数：

VT_BOOL	プログラム原点復帰の状態を指定します。 True：プログラム原点移動実行 False：通常 (通常→プログラム原点移動実行で実行されます。)
---------	--

戻り値：なし

使用例：

```
caoRobot.Execute("ProgramHome", False)
delay 100
// ブロードキャストの場合、コマンド間に無通信時間を設ける
caoRobot.Execute("ProgramHome", True)
// 通常→プログラム原点移動実行を指定
```

2.2.3.49. CaoRobot::Execute("SearchStop") コマンド

プログラム原点への移動を実行します。

引数：

VT_BOOL	探し動作完了後、プレスプログラムを終了する/しないの切り替えを指定します。 True：探し動作後停止 False：探し動作後停止しない
---------	--

戻り値：なし

使用例：

```
caoRobot.Execute("SearchStop", True)
// 探し動作後停止を指定
```

2.2.3.50. CaoRobot::Execute("ForceProgramStop") コマンド

プレスプログラムの強制終了を実行します。

引数：

VT_BOOL	プレスプログラム強制終了の状態を指定します。 True：プログラム強制終了
---------	---

	False : 通常 (通常→プログラム強制終了で実行されます。)
--	--------------------------------------

戻り値 : なし

使用例 :

```
caoRobot.Execute("ForceProgramStop", False)
delay 100
// ブロードキャストの場合, コマンド間に無通信時間を設ける
caoRobot.Execute("ForceProgramStop", True)
// 通常→プログラム強制終了を指定
```

2.2.3.51. CaoRobot::Execute("ProgramStart") コマンド

プレスプログラムを実行します。

引数 :

VT_BOOL	プレスプログラムスタートの状態を指定します。 True : プログラムスタート False : 通常 (通常→プログラムスタートで実行されます。)
---------	--

戻り値 : なし

使用例 :

```
caoRobot.Execute("ProgramStart", False)
delay 100
// ブロードキャストの場合, コマンド間に無通信時間を設ける
caoRobot.Execute("ProgramStart", True)
// 通常→プログラムスタートを指定
```

2.2.3.52. CaoRobot::Execute("WriteSingleRegister") コマンド

指定アドレスレジスタへの書き込みを実行します。

引数 :

VT_ARRAY VT_VARIANT		
0	VT_I4	書き込み開始アドレスを指定します。
1	VT_UI2	書き込みデータを指定します。

戻り値 : なし

使用例 :

```
caoRobot.Execute("WriteSingleRegister", Array(3328, 4096))
// アドレス 0x0D00 (3328) に 0x1000 (4096) を書き込み (サーボ ON)
delay 100
```

```
// ブロードキャストの場合、コマンド間に無通信時間を設ける
caoRobot.Execute("WriteSingleRegister", Array(3328, 4112))
// アドレス 0x0D00(3328)に 0x1010(4112)を書き込み(原点復帰)
```

2.2.3.53. CaoRobot::Execute("WriteMultipleRegisters") コマンド

指定アドレスから連続したレジスタへの書き込みを実行します。

引数：

VT_ARRAY VT_VARIANT		
0	VT_I4	書き込み開始アドレスを指定します。
1	VT_I4	書き込みデータ数(レジスタ数)を指定します。
2	VT_ARRAY VT_UI2	
	n VT_UI2	書き込みデータを指定します。

戻り値：なし

使用例：

```
caoRobot.Execute("WriteMultipleRegisters", Array(39168, 2, Array(0, 5000)))
// アドレス 0x9900(39168), 0x9901(39169)にそれぞれ, 0, 5000 を書き込み(ポジション No.0 の目標位置を 50.00mm に設定)
```

2.2.3.54. CaoRobot::Execute("MovePTP") コマンド

直値移動を実行します。

引数：

VT_ARRAY VT_VARIANT		
0	VT_I4	目標位置を指定します。(単位：0.01mm)
1	VT_UI4	位置決め幅を指定します。(単位：0.01mm)
2	VT_UI4	速度を指定します。(単位：0.01mm/s)
3	VT_UI2	加減速度を指定します。(単位：0.01G)
4	VT_UI2	押し付け時電流制限を指定します。(単位：%)
5	VT_UI2	制御フラグを指定します。

戻り値：なし

使用例：

```
caoRobot.Execute("MovePTP", Array(5000, 10, 10000, 30, 0, 8))
// 目標位置 50.00mm
// 位置決め幅 0.10mm
// 速度 100.00mm/s
// 加減速度 0.30G
// 押し付け時電流制限 0%
```

2.2.3.55. CaoRobot::Execute("WritePositionData") コマンド

ポジションデータの書き込みを実行します。

引数：

VT_ARRAY VT_VARIANT		
0	VT_UI2	書き込み対象ポジション No. を指定します。
1	VT_I4	目標位置を指定します。(単位：0.01mm)
2	VT_UI4	位置決め幅を指定します。(単位：0.01mm)
3	VT_UI4	速度指令を指定します。(単位：0.01mm/s)
4	VT_I4	個別ゾーン境界+側を指定します。(単位：0.01mm)
5	VT_I4	個別ゾーン境界-側を指定します。(単位：0.01mm)
6	VT_UI2	加速度指令を指定します。(単位：0.01G)
7	VT_UI2	減速度指令を指定します。(単位：0.01G)
8	VT_UI2	押付け時電流制限値を指定します。(100%=0xFF)
9	VT_UI2	負荷電流閾値を指定します。(100%=0xFF)
10	VT_UI2	制御フラグ指定を指定します。

戻り値：なし

使用例：

```
caoRobot.Execute("WritePositionData", Array(12, 10000, 10, 20000, 6000, 4000, 10, 30, 0, 0, 0))
```

```
// 対象ポジション No. 12
```

```
// 目標位置 100.00mm
```

```
// 位置決め幅 0.10mm
```

```
// 速度 100.00mm/s
```

```
// 個別ゾーン境界+側 60.00mm
```

```
// 個別ゾーン境界-側 40.00mm
```

```
// 加速度 0.10G
```

```
// 減速度 0.30G
```

2.2.4. CaoRobot::AddVariable メソッド

CaoRobot から Variable オブジェクトを生成する際に、変数名を指定することによって、接続した PCON デバイスから読み書きするデータを決定します。

以下に、AddVariable の仕様を示します。

書式

AddVariable

```
(  
    "<変数名>",           // 変数名  
    "<オプション>"       // オプション文字列  
)
```

使用できる変数名、オプション、詳細に付いては変数一覧を参照して下さい。

2.2.5. CaoRobot::get_VariableNames メソッド

表 2-5 の変数名リストを取得します。

2.2.6. CaoVariable::get_Value プロパティ

指定したオプションによって、PCON デバイスからデータを取得します。

2.2.7. CaoVariable::put_Value プロパティ

指定したオプションによって、PCON デバイスへデータを書き込みます。

2.3. 変数一覧

2.3.1. コントローラクラス

以下の表 2-5 にコントローラクラスの AddVariable にて使用できる変数一覧を記述します。

表 2-5 ロボットクラス変数一覧

変数名	説明	Value		リンク
		get	put	
@MAKER_NAME	メーカー名を取得します。	○	－	P. 36
@VERSION	バージョンを取得します。	○	－	P. 36

※<??>は任意の文字列

2.3.1.1. @MAKER_NAME

メーカー名を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_BSTR	固定値：“IAI”
---------	-----------

2.3.1.2. @VERSION

バージョンを取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_BSTR	現在の DLL のバージョンを表す“*. *.*”文字列
---------	------------------------------

2.3.2. ロボットクラス

以下の表 2-5 にロボットクラスの AddVariable にて使用できる変数一覧を記述します。

表 2-6 ロボットクラス変数一覧

変数名	説明	Value		リンク
		get	put	
REGISTER<??>	指定のレジスタの値を取得、設定します。	○	○	P. 39
@ALARM_INFO	最後に発生したアラーム情報を取得します。	○	–	P. 39
POSITION_DATA<??>	指定のポジションデータを取得、設定します。	○	○	P. 39
@MOVE_COUNT	通算移動回数を取得します。	○	–	P. 40
@MOVE_DISTANCE	通算走行距離を取得します。	○	–	P. 40
@CURRENT_TIME	現在時刻を取得します。	○	–	P. 40
@FAN_DRIVE_TIME	ファンの通算駆動時間を取得します。	○	–	P. 40
@CURRENT_POSITION	現在位置を取得します。	○	–	P. 40
@ALARM_CODE	現在発生アラームコードを取得します。	○	–	P. 40
@IO_INPUT	I/O ポート入力信号の値を取得します。	○	–	P. 41
@IO_OUTPUT	I/O ポート出力信号の値を取得します。	○	–	P. 41
@DSS1	デバイスステータスレジスタ 1 の値を取得します。	○	–	P. 41
@DSS2	デバイスステータスレジスタ 2 の値を取得します。	○	–	P. 41
@DSSE	拡張デバイスステータスレジスタの値を取得します。	○	–	P. 41
@STAT	システムステータスレジスタの値を取得します。	○	–	P. 41
@SPEED	モータ実速度のモニタデータを取得します。	○	–	P. 42
@VALUE	モータ電流のモニタデータを取得します。	○	–	P. 42
@DEVIATION	1ms 周期毎の位置指令値とフィードバック値(実位置)の偏差量を取得します。	○	–	P. 42
@INTEGRATION_TIME	電源投入後の積算時間を取得します。	○	–	P. 42
@SPECIAL_INPUT	特殊入力ポートモニタレジスタの値を取得します。	○	–	P. 42
@ZONE	ゾーンステータスレジスタの値を取得します。	○	–	P. 42
@COMPLETE_POSITION_NO	ポジション番号ステータスレジスタの値を取得します。	○	–	P. 43
@SSSE	拡張システムステータスレジスタの値を取得します。	○	–	P. 43
@LOAD	ロードセル測定値のモニタデータの値を取得します。	○	–	P. 43
@LOAD_LEVEL	現在のモータへの負荷レベルの比率を取得します。	○	–	P. 43
@PRESS_PROGRAM_ALARM_CODE	プレスプログラムアラームコードの値を取得します。	○	–	P. 43

変数名	説明	Value		リンク
		get	put	
@ALARM_PRESS_PROGRAM_NO	アラームが発生しているプレスプログラム番号を取得します.	○	-	P. 43
@PRESS_PROGRAM_STATUS	プレスプログラムステータスレジスタの値を取得します.	○	-	P. 44
@PRESS_PROGRAM_JUDGE	プレスプログラム判定ステータスレジスタの値を取得します.	○	-	P. 44

※<??>は任意の文字列

2.3.2.1. REGISTER<??>

指定のレジスタの値を取得，設定します。

オプション：

ADDRESS	開始アドレス (0～) を指定します。
NUM	操作対象レジスタ数 (1～) を指定します。

get_value の戻り値のデータ構成：

VT_ARRAY VT_UI2		
n	VT_UI2	開始アドレスから連続した読み込みレジスタの値

put_value 時の引数のデータ構成：

get_value の戻り値のデータ構成と同じ

2.3.2.2. @ALARM_INFO

最後に発生したアラーム情報を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_ARRAY VT_VARIANT		
0	VT_UI2	アラーム詳細コード
1	VT_UI2	アラームアドレス
2	VT_UI4	アラームコード
3	VT_UI4	アラーム発生時刻 (RTC 無効の場合は電源投入後経過時間[ms])

2.3.2.3. POSITION_DATA<??>

指定のポジションデータを取得，設定します。

オプション：

POSITION_NO	ポジション No. (0～) を指定します。
-------------	------------------------

get_value の戻り値のデータ構成：

VT_ARRAY VT_VARIANT		
0	VT_I4	目標位置 (単位：0.01mm)
1	VT_UI4	位置決め幅 (単位：0.01mm)
2	VT_UI4	速度指令 (単位：0.01mm/s)
3	VT_I4	個別ゾーン境界+側 (単位：0.01mm)
4	VT_I4	個別ゾーン境界-側 (単位：0.01mm)
5	VT_UI2	加速度指令 (単位：0.01G)
6	VT_UI2	減速度指令 (単位：0.01G)
7	VT_UI2	押付け時電流制限値 (100%=0xFF)
8	VT_UI2	負荷電流閾値 (100%=0xFF)

9	VT_UI2	制御フラグ指定
---	--------	---------

put_value 時の引数のデータ構成 :

get_value の戻り値のデータ構成と同じ

2.3.2.4. @MOVE_COUNT

通算移動回数を取得します.

オプション : なし

get_value 時の戻り値のデータ構成 :

VT_UI4	通算移動回数
--------	--------

2.3.2.5. @MOVE_DISTANCE

通算走行距離を取得します.

オプション : なし

get_value 時の戻り値のデータ構成 :

VT_UI4	通算走行距離
--------	--------

2.3.2.6. @CURRENT_TIME

現在時刻を取得します.

オプション : なし

get_value 時の戻り値のデータ構成 :

VT_UI4	現在時刻
--------	------

2.3.2.7. @FAN_DRIVE_TIME

ファンの通算駆動時間を取得します.

オプション : なし

get_value 時の戻り値のデータ構成 :

VT_UI4	ファンの通算駆動時間 (単位 : 1s)
--------	----------------------

2.3.2.8. @CURRENT_POSITION

現在位置を取得します.

オプション : なし

get_value 時の戻り値のデータ構成 :

VT_I4	現在位置 (単位 : 0.01mm)
-------	--------------------

2.3.2.9. @ALARM_CODE

現在発生アラームコードを取得します.

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	現在発生アラームコード
--------	-------------

2.3.2.10. @IO_INPUT

I0 ポート入力信号の値を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	I0 ポート入力信号の値
--------	--------------

2.3.2.11. @IO_OUTPUT

I0 ポート出力信号の値を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	I0 ポート出力信号の値
--------	--------------

2.3.2.12. @DSS1

デバイスステータスレジスタ 1 の値を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	デバイスステータスレジスタ 1 の値
--------	--------------------

2.3.2.13. @DSS2

デバイスステータスレジスタ 2 の値を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	デバイスステータスレジスタ 2 の値
--------	--------------------

2.3.2.14. @DSSE

拡張デバイスステータスレジスタの値を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	拡張デバイスステータスレジスタの値
--------	-------------------

2.3.2.15. @STAT

システムステータスレジスタの値を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI4	システムステータスレジスタの値
--------	-----------------

2.3.2.16. @SPEED

モータ実速度のモニタデータを取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_I4	モータ実速度のモニタデータ（単位：0.01mm/s）
-------	----------------------------

2.3.2.17. @VALUE

モータ電流のモニタデータを取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI4	モータ電流のモニタデータ（単位：1mA）
--------	----------------------

2.3.2.18. @DEVIATION

1ms 周期毎の位置指令値とフィードバック値(実位置)の偏差量を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI4	1ms 周期毎の位置指令値とフィードバック値(実位置)の偏差量（単位：Pulse）
--------	---

2.3.2.19. @INTEGRATION_TIME

電源投入後の積算時間を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI4	電源投入時からの積算時間（単位：ms）
--------	---------------------

2.3.2.20. @SPECIAL_INPUT

特殊入力ポートモニタレジスタの値を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	特殊入力ポートモニタレジスタの値
--------	------------------

2.3.2.21. @ZONE

ゾーンステータスレジスタの値を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	ゾーンステータスレジスタの値
--------	----------------

2.3.2.22. @COMPLETE_POSITION_NO

ポジション番号ステータスレジスタの値を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	ポジション番号ステータスレジスタの値
--------	--------------------

2.3.2.23. @SSSE

拡張システムステータスレジスタの値を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	拡張システムステータスレジスタの値
--------	-------------------

2.3.2.24. @LOAD

ロードセル測定値のモニタデータの値を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI4	ロードセル測定値のモニタデータの値（単位：0.01N）
--------	-----------------------------

2.3.2.25. @LOAD_LEVEL

現在のモータへの負荷レベルの比率を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI4	現在のモータへの負荷レベルの比率（単位：1%）
--------	-------------------------

2.3.2.26. @PRESS_PROGRAM_ALARM_CODE

プレスプログラムアラームコードの値を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	プレスプログラムの現在発生アラームコード
--------	----------------------

2.3.2.27. @ALARM_PRESS_PROGRAM_NO

アラームが発生しているプレスプログラム番号を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	アラームが発生しているプレスプログラム番号
--------	-----------------------

2.3.2.28. @PRESS_PROGRAM_STATUS

プレスプログラムステータスレジスタの値を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	プレスプログラムステータスレジスタの値
--------	---------------------

2.3.2.29. @PRESS_PROGRAM_JUDGE

プレスプログラム判定ステータスレジスタの値を取得します。

オプション：なし

get_value 時の戻り値のデータ構成：

VT_UI2	プレスプログラム判定ステータスレジスタの値
--------	-----------------------

2.4. エラーコード

本プロバイダには、以下の独自エラーコードが存在します。（表 2-7 参照）

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 2-7 独自エラーコード表

エラー名	エラー番号	説明
E_CAOP_ILLEGAL_ARGUMENT	0x80100F01	引数エラー。 このエラーコードを返すコマンドに渡されたパラメータが無効か、範囲外です。
E_CAOP_ILLEGAL_STATE	0x80100F02	状態エラー。 関数が間違った状態で呼ばれています。プロトコルがまだ正常にオープンされていない場合、この戻りコードはすべての関数によって返されます。
E_CAOP_ILLEGAL_SLAVE_ADDRESS	0x80100F05	不正サーバ機器アドレス。 ブロードキャストに対応していない機能でアドレス 0 が使用されました。
E_CAOP_OPEN	0x80100F42	ポートまたはソケットオープンエラー。 TCP/IP ソケットまたはシリアルポートを開けませんでした。シリアルポートの場合には、シリアルポートがシステムに存在していない可能性があります。
E_CAOP_FTALK_PORT_ALREADY_OPEN	0x80100F43	シリアル・ポートは既に関いています。 オープン操作用に定義されたシリアルポートが既に他のアプリケーションで開かれます。
E_CAOP_FTALK_TCPIP_CONNECT	0x80100F44	TCP/IP 接続エラー。 TCP/IP 接続を確立できませんでした。ホストがネットワークまたは IP アドレス上に存在するか、名前が間違っているホストの場合、通常このエラーが発生します。リモートホストは、適切な Port 番号を Listen する必要があります。
E_CAOP_CONNECTION_WAS_CLOSED	0x80100F45	リモートピアは、TCP/IP 接続を閉じました。 TCP / IP 接続がリモートピアによって閉じたり壊れていたことを通知します。
E_CAOP_SOCKET_LIB	0x80100F46	ソケットライブラリエラー。 TCP/IP ソケットライブラリー（例えば WINSOCK）が

エラー名	エラー番号	説明
		ロードできませんでした。DLL が見つからないか、インストールされていない可能性があります。
E_CAOP_PORT_ALREADY_BOUND	0x80100F47	TCP ポートは既にバインドしています。 指定された TCP ポートをバインドすることができないことを示します。Port が既に別のアプリケーションによって取られていたり、再使用のための TCP / IP スタックによってまだリリースされていない可能性があります。
E_CAOP_LISTEN_FAILED	0x80100F48	Listen に失敗しました。 指定された TCP ポート Listen に失敗しました。
E_CAOP_FILEDES_EXCEEDED	0x80100F49	ファイル記述子を超えました。 使用可能なファイル記述子の最大数を超えました。
E_CAOP_PORT_NO_ACCESS	0x80100F4A	シリアルポートまたは TCP ポートにアクセスする権限がありません。 シリアルポートの場合、アクセス権を変更します。 TCP / IP の場合、TCP ポート番号が IPPORT_RESERVED 範囲外です。
E_CAOP_PORT_NOT_AVAIL	0x80100F4B	TCP ポートは使用できません。 指定された TCP ポートは、この動作環境では利用できません。
E_CAOP_LINE_BUSY	0x80100F4C	シリアルラインがビジーです。 シリアル回線は、トラフィックがあってはならない状態であるにもかかわらず、ノイズ等を受信しています。
E_CAOP_CHECKSUM	0x80100F81	チェックサムエラー。 受信したフレームのチェックサムが無効です。
E_CAOP_INVALID_FRAME	0x80100F82	無効なフレームエラー。 受信したフレームが通信プロトコルのいずれかの構造または内容によって対応していないか、以前に送信されたクエリのフレームと一致しないことを通知します。
E_CAOP_INVALID_REPLY	0x80100F83	無効な応答エラー。 受信した応答フレームが通信プロトコルに対応していないことを通知します。
E_CAOP_REPLY_TIMEOUT	0x80100F84	タイムアウトエラー。

エラー名	エラー番号	説明
		サーバ機器が時間内に応答しない、またはまったく応答しない場合に発生することがあります。間違ったサーバ機器アドレスは、このエラーを誘発します。
E_CAOP_SEND_TIMEOUT	0x80100F85	送信タイムアウトエラー。 データ送信がタイムアウトしたことを通知します。 ハンドシェーク・ラインが正しく設定されていない場合に発生する可能性があります。
E_CAOP_INVALID_MBAP_ID	0x80100F86	無効な識別子。 プロトコルまたはトランザクション識別子ですが間違っています。TCP サーバデバイスは、TCP クライアントから受信した識別子を返す必要があります。
E_CAOP_MBUS_EXCEPTION_RESPONSE	0x80100FA0	Modbus 例外応答メッセージを受信したことを通知します。
E_CAOP_MBUS_ILLEGAL_FUNCTION_RESPONSE	0x80100FA1	Modbus で無効な関数例外応答（コード 01）を受信したことを通知します。
E_CAOP_MBUS_MBUS_ILLEGAL_ADDRESS_RESPONSE	0x80100FA2	Modbus で不正データアドレス例外応答（コード 02）を受信したことを通知します。
E_CAOP_MBUS_ILLEGAL_VALUE_RESPONSE	0x80100FA3	Modbus で不正な値例外応答（コード 03）を受信したことを通知します。
E_CAOP_MBUS_SLAVE_FAILURE_RESPONSE	0x80100FA4	Modbus でスレーブ失敗例外応答（コード 04）を受信したことを通知します。
E_MACHINE_TYPE_UNSPECIFIED	0x80110001	非対応の機種を選択しています。

3. サンプルプログラム

以下に PCON デバイスへ書き込みをする簡単なサンプルを示します。

前提条件：

- ・ RC8 の PAC スクリプトを対象としたものとする。
- ・ PCON デバイスの COM ポートは「1」、機器アドレスは「1」とする。

List 3-1

Sample.pcs

Sub Main

’ オブジェクト

Dim caoCtrl as Object

Dim caoRobot as Object

’ コントローラーオブジェクト作成

caoCtrl = cao.AddController("PCON", "CaoProv. IAI.PCON", "", "conn=com:1")

’ ロボットオブジェクト作成

caoRobot = caoCtrl.AddRobot("Robot", "UnitAddress=1")

’ サーボON

caoRobot.Execute "SwitchServo", true

’ 原点復帰(立ち上がりで動作)

caoRobot.Execute "Home", 0

caoRobot.Execute "Home", 1

’ 原点復帰が完了するまで待ち

delay 10000

’ ポジション移動(立ち上がりで動作)

caoCtrl.Execute "StartPosition", 0

caoCtrl.Execute "StartPosition", 1

’ ポジション移動が完了するまで待ち

delay 10000

End Sub