IAI Corporation
IAI PCON provider

Version 1.0.2

User's guide

May 16, 2022

Remarks:

[Revision history]

Version	Date	Content
1.0.0	2017-12-6	First edition.
1.0.1	2018-11-5	Bug fixed at AddController.
1.0.2	2019-4-26	Bug fixed at memory leak when creating / deleting CaoRobot.
	2020-10-6	Addition of license addition items. Typographical correction.
	2022-5-16	Sample program fixed.

[Supported models]

Model	Version	Remarks
ERC2(SE) / ERC3(V0002 or later)/RCP6S series ゛	-	
PCON-C / CA / CB / CFA / CFB / CG / CGB / CGFB / CF / CY / SE	-	
ACON-C / CG / CA / CB / CY / SE	-	
SCON-C / CA / CAL / CB(includes servo press type) / CGB	-	
DCON-CA / CB	-	
ROBONET_RS485(When RTU mode and SIO through mode)	-	
	-	

【Attention】

Additional license for " IAI Positioner Type Controller Provider " is required to use this provider.

1. Introduction

This is a user’s guide of CAO provider that is designed for IAI position controller PCON series. Hereafter, this provider is called PCON provider.

This guide explains about the function of PCON provider and the implemented method.

PCON provider wraps Modbus.X provider because PCON provider uses Modbus/RTU protocol.

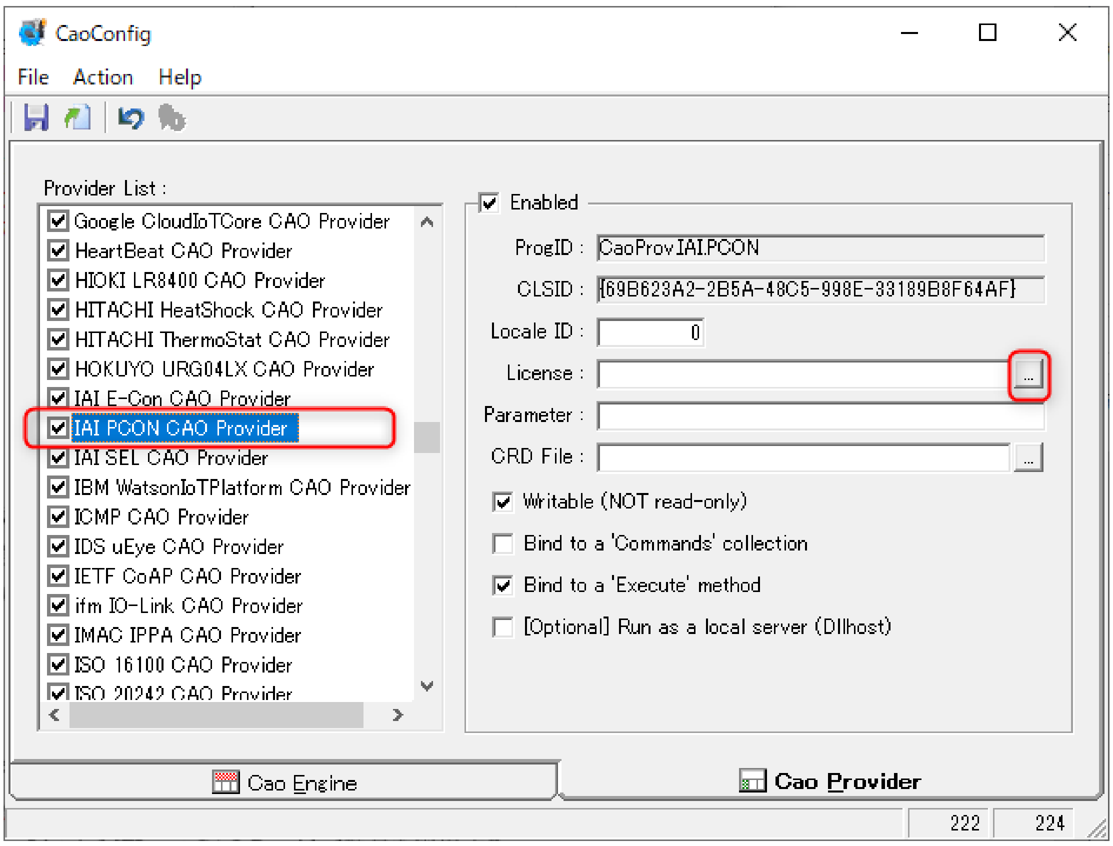
1.1. Installing license

To use OpenCV Provider, you need to install ORiN2 SDK, and also need to input "IAI Positioner Type Controller Provider" license information. If you would like to install it for evaluation, please use the following license.

IIEN-M1WA-JTYV-AFWF (valid for 3 months)

How to add the license is as follows.

1. Run the CaoConfig tool from the [Start] menu, and select the [Cao Provider] tab.
2. Select the [IAI PCON CAO Provider] item on the provider list.
3. Click the [...] button of the license input box.
4. Click the [Add] button in the "ORiN2 License Manager" window.
5. Input a license key, and click the [OK] button.
6. Click the [Close] button to exit.



Provider List :
- [✓] Google CloudIoTCore CAO Provider
- [✓] HeartBeat CAO Provider
- [✓] HIOKI LR8400 CAO Provider
- [✓] HITACHI HeatShock CAO Provider
- [✓] HITACHI ThermoStat CAO Provider
- [✓] HOKUYO URG04LX CAO Provider
- [✓] IAI E-Con CAO Provider
- [✓] IAI PCON CAO Provider
- [✓] IAI SEL CAO Provider
- [✓] IBM WatsonIoTPlatform CAO Provider
- [✓] ICMP CAO Provider
- [✓] IDS uEye CAO Provider
- [✓] IETF CoAP CAO Provider
- [✓] ifm IO-Link CAO Provider
- [✓] IMAC IPPA CAO Provider
- [✓] ISO 16100 CAO Provider
- [✓] ISO 20242 CAO Provider

[✓] Enabled

ProgID : CaoProv.IAI.PCON

CLSID : {69B623A2-2B5A-48C5-998E-33189B8F64AF}

Locale ID : 0

License : ...

Parameter :

CRD File : ...

[✓] Writable (NOT read-only)

[ ] Bind to a 'Commands' collection

[✓] Bind to a 'Execute' method

[ ] [Optional] Run as a local server (Dllhost)

Figure 1-1 Installing ' IAI Positioner Type Controller Provider ' license

2. Overview of this provider

2.1. Overview

PCON provider reads/writes data though COM communication or TCP/IP communication. The file format of PCON provider is DLL (Dynamic Link　Library). Table 2-1 shows the details.

Table 2-1　PCON provider

File name	CaoProvIAIPCON.dll
ProgID	CaoProv.IAI.PCON
Registration	regsvr32 CaoProvIAIPCON.dll
Unregistration	regsvr32 /u CaoProvIAIPCON.dll

Figure 2-1 shows the correspondence between items on PCON provider and PCON devices.

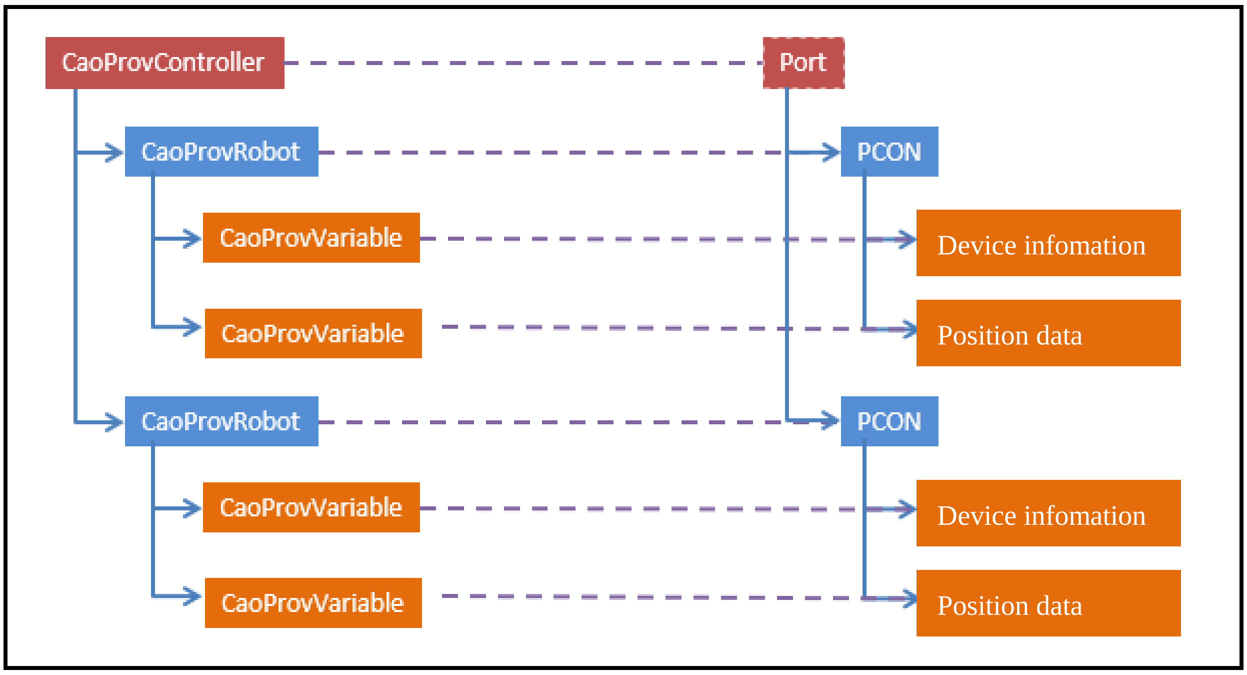


Figure 2-1 Corresponding flows of CaoProvController and PCON device

2.2. Method and Properties

2.2.1. CaoWorkspace::AddController method

This provider specifies required options to connect to a PCON device when a Controller object is created. The following shows the specification of AddController.

Syntax

```
AddController
(
"<Controller name>",        // controller name (any name)
"CaoProv.IAI.PCON",         // provider name (fixed)
"<computer name>",          // a computer name where this provider runs (unused).
"<Option>"            // Option character string
)
```

The following shows the list of option string items.

Table 2-2 Option character string of CaoWorkspace: : AddController

Option	Required	Description	Value range	Default value
CONN= COM:<Connection destination COM port number>[:<baud rate>[:<parity>:<data bit count>:<stop bit count>]]	Yes[1]	Specify the connection destination COM port number according to the operation environment.	-----	Baud rate: 38400 Parity: NONE Data bit count: 8 bits Stop bit count: 1 bit
CONN= ETH<connection destination IP>[:<connection destination port>]	Yes	Specify the connection destination IP address and port number according to the operation environment.	-----	Port: 502
PacketType=<packet parameter>[2]	-	Enter a data type of Modbus communication protocol. 0: RTU (default) 1: ASCII	0 to 1	0
Sync=True / False	-	Enable or disable the synchronous mode.	True / False	True

[1] Either COM or ETH must be entered.
[2] If "eth" is selected for the communication device in Conn option, the internal data type of TCP/IP is fixed to RTU. In this case, this option is ignored.

Option	Required	Description	Value range	Default value
		True: Synchronous mode (default) Start-Stop Synchronization System False: Asynchronous mode		
Timeout=<data sending/receive timeout>	-	Specify sending and receiving timeout period [ms].	1 to 100000	1000
Retry=<retry count>	-	Specify the number of communication retry at data sending/receiving.	0 to 10	0
RtsTransmitDelayTime=<sending/receiving switching delay>[3]	-	Specify the sending/receiving switching delay of RTS signal[1][ms]. 0: RTS signal[4] is always ON (default) 1 to 100000: RTS signal affects sending/receiving circuit.	0 to 100000	0
PollDelayTime=<Polling delay time>	-	Specify the polling delay time [ms].	0 to 100000	0

About RtsTransmitDelayTime option

・This option is mainly used in the following hardware configuration conditions;

- Transmission mode is half duplex, and,
- Change of sending/receiving requires software.

Specifying 1ms or larger allows to change sending/receiving by RTS signal.

・If this option is set to 1ms or longer, the communication is done in the following sequence.

[Turn ON the RTS signal just before the sending] > [Data sending start] > [Data sending complete] > [Turn OFF the RTS signal once the predetermined time elapses]

・Data sending completion in this provider is determined earlier than the sending completion on the actual transmission line. This is because the sending completion in this provider is determined based on the sending completion notification from the communication device driver. (The calculation of the delay time for the actual sending completion differs depending on the vendor if FIFO on the communication hardware is used. It is recommended not to use FIFO for delay time calculation.)

[3] If "eth" is selected for the communication device in Conn option, this option is ignored.
[4] RTS signal means Request-To-Send signal.

2.2.2. CaoController::AddRobot method

This provider specifies required options to connect to a PCON-connected Robot device when a Robot object is created.

The following shows the specification of AddRobot.

Syntax

```
AddRobot
(
"<Robot name>", // Robot name (any name)
  "<Option>"        // Option character string
)
```

The following shows a list of option string strings.

Table 2-3 CaoController::Option string of AddRobot

Option	Required	Description	Value range	Default value
UnitAddress=<device address>	-	Specify a server device address (for com) or an unit identifier (for eth) of communication destination. For com: Server device address[5] For eth: Unit identifier	0 to 255	For com: 1 For eth: 0
Type=<device type>	-	Specify the connection destination device 0: other than the following devices 1 : SCON-CA/CAL/CB 2 : PCON-CA/CFA , ACON-CA/CB	0 to 2	0

5 Specifying "0"broadcasts a command. In this case, an identical command will be sent to all devices being connected.

2.2.3. CaoRobot:: Execute method

Specifying a command with CaoRobot::Execute enables to read/write some data.

The following shows the specification of Execute method.

Syntax

```
Execute
(
"<method name>",            // method name
"<argument>"      // argument
)
```

Table 2-4 CaoRobot::Execute command list

Command Name	Description	Broadcast-available	Link
ReadMultipleRegisters	Reads a register value.	-	P.19
ReadAlarmInfo	Obtains the latest alarm information	-	P.19
ReadPositionData	Obtains position data.	-	P.19
ReadMoveCount	Obtains the total moving count.	-	P.20
ReadMoveDistance	Obtains the total moving distance.	-	P.20
ReadCurrentTime	Obtains the current time.	-	P.21
ReadFanDriveTime	Obtains the total fan driving time.	-	P.21
ReadCurrentPosition	Obtains the present position.	-	P.21
ReadCurrentAlarmCode	Obtains an alarm code currently occurred.	-	P.21
ReadIOPortInputSignal	Obtains the value of IO port input signal.	-	P.22
ReadIOPortOutputSignal	Obtains the value of IO port output signal.	-	P.22
ReadDSS1	Obtains the value of device status register 1.	-	P.22
ReadDSS2	Obtains the value of device status register 2.	-	P.22
ReadDSSE	Obtains the value of the expansion device status register.	-	P.23
ReadSTAT	Obtains the value of the system status register.	-	P.23
ReadCurrentSpeed	Obtains the monitor data of the present motor speed.	-	P.23
ReadCurrentValue	Obtains the motor current monitor data.	-	P.23
ReadDeviation	Obtains the deviation between the position command value and the feedback value (actual position) in every 1ms interval.	-	P.24

Command Name	Description	Broadcast-available	Link
ReadIntegrationTime	Obtains the total Power-ON time after the controller power-ON.	-	P.24
ReadSpecialInputPort	Obtains the value of the special input port monitor register.	-	P.24
ReadZoneStatus	Obtains the value of the zone status register.	-	P.24
ReadCompletePositionNo	Obtains the value of the position number status register.	-	P.25
ReadSSSE	Obtains the value of the expansion system status register.	-	P.25
ReadLoad	Obtains the value of the monitor data of the load cell measurement.	-	P.25
ReadLoadLevel	Obtains the load level currently charged to the motor in ratio.	-	P.25
ReadPressProgramAlarmCode	Obtains the value of the press program alarm code.	-	P.26
ReadAlarmPressProgramNo	Obtains the press program number that issues an alarm.	-	P.26
ReadPressProgramStatus	Obtains the value of the press program status register.	-	P.26
ReadPressProgramJudgeStatus	Obtains the value of the press program judgment status register.	-	P.26
WriteSingleDiscreteOutput	Switches ON/OFF the slave DO.	Yes	P.27
SwitchSafetySpeed	Enables/disables the safety speed.	Yes	P.27
SwitchServo	Switches ON/OFF the servo.	Yes	P.27
ResetAlarm	Resets an alarm.	Yes	P.28
ReleaseBreak	Forcefully releases a brake.	Yes	P.28
Pause	Pause.	Yes	P.28
Home	Returns to the home position.	Yes	P.28
StartPosition	Moves to the position specified by the position number.	Yes	P.29
SwitchJogInching	Switches between jogging and inching.	Yes	P.29
SwitchMode	Switches between the teaching mode and the normal operation mode.	Yes	P.30
GetCurrentPositionData	Obtains the position data.	Yes	P.30
JogPlus	Executes jogging or inching motion to the opposite to the home direction.	Yes	P.30
JogMinus	Executes jogging or inching to the home direction.	Yes	P.30
MoveToStartPosition	Moves to the start position.	Yes	P.31

Command Name	Description	Broadcast-available	Link
Calibration	Calibrates the dedicated load cell.	Yes	P.31
SwitchPIOModbus	Enables/disables the Modbus command.	Yes	P.31
Stop	Starts deceleration to a stop.	Yes	P.32
SwitchAxisMove	Allows/prohibits the axis operation.	Yes	P.32
ProgramHome	Moves to the program home position.	Yes	P.32
SearchStop	Switches whether to finish the press program or not after the complete of the search operation.	Yes	P.33
ForceProgramStop	Forcibly competes the press program.	Yes	P.33
ProgramStart	Executes the press program.	Yes	P.33
WriteSingleRegister	Writes data to a specified address register.	Yes	P.34
WriteMultipleRegisters	Writes data to the sequence address register starting from the specified address.	Yes	P.34
MovePTP	Moves to the target position by entering a numerical value.	Yes	P.35
WritePositionData	Writes position data.	Yes	P.35

Table 2-5 CaoRobot::Execute command – Modbus function code correspondence table

Command Name	IAI PCON (Modbus)	
	Function code	Symbol (address)
ReadMultipleRegisters	Data, status reading (FC:03)	-
ReadAlarmInfo	Data, status reading (FC:03)	ALA0,ALC0,ALT0 (0x0500 to 0x0505)
ReadPositionData	Data, status reading (FC:03)	PCMD,INP,VCMD,ZNMP,ZNLP,ACMD, DCMD,PPOW,LPOW,CTLF (Starting address=0x1000+(0x10×Position No.))
ReadMoveCount	Data, status reading (FC:03)	TLMC (0x8400 to 0x8401)
ReadMoveDistance	Data, status reading (FC:03)	ODOM (0x8402 to 0x8403)
ReadCurrentTime	Data, status reading (FC:03)	TIMN

Command Name	IAI PCON (Modbus)	
	Function code	Symbol (address)
		(For SCON-CA/CAL/CB; 0x841E to 0x841F) (For PCON-CA/CFA,ACON-CA/CB; 0x8420 to 0x8421)
ReadFanDriveTime	Data, status reading (FC:03)	TFAN (For SCON-CA/CAL/CB; 0x842A to 0x842B) (For PCON-CA/CFA,ACON-CA/CB; 0x842E to 0x842F)
ReadCurrentPosition	Data, status reading (FC:03)	PNOW (0x9000 to 0x9001)
ReadCurrentAlarmCode	Data, status reading (FC:03)	ALMC (0x9002)
ReadIOPortInputSignal	Data, status reading (FC:03)	DIPM (0x9003)
ReadIOPortOutputSignal	Data, status reading (FC:03)	DOPM (0x9004)
ReadDSS1	Data, status reading (FC:03)	DSS1 (0x9005)
ReadDSS2	Data, status reading (FC:03)	DSS2 (0x9006)
ReadDSSE	Data, status reading (FC:03)	DSSE (0x9007)
ReadSTAT	Data, status reading (FC:03)	STAT (0x9008)
ReadCurrentSpeed	Data, status reading (FC:03)	VNOW (0x900A to 0x900B)
ReadCurrentValue	Data, status reading (FC:03)	CNOW (0x900C to 0x900D)
ReadDeviation	Data, status reading (FC:03)	DEVI (0x900E to 0x900F)
ReadIntegrationTime	Data, status reading (FC:03)	STIM (0x9010 to 0x9011)
ReadSpecialInputPort	Data, status reading (FC:03)	SIPM

Command Name	IAI PCON (Modbus)	
	Function code	Symbol (address)
		(0x9012)
ReadZoneStatus	Data, status reading (FC:03)	ZONS (0x9013)
ReadCompletePositionNo	Data, status reading (FC:03)	POSS (0x9014)
ReadSSSE	Data, status reading (FC:03)	SSSE (0x9015)
ReadLoad	Data, status reading (FC:03)	FBFC (0x901E to 0x901F)
ReadLoadLevel	Data, status reading (FC:03)	OLLV (0x9020 to 0x9021)
ReadPressProgramAlarmCode	Data, status reading (FC:03)	ALMP (0x9022)
ReadAlarmPressProgramNo	Data, status reading (FC:03)	ALMP (0x9023)
ReadPressProgramStatus	Data, status reading (FC:03)	PPST (0x9024)
ReadPressProgramJudgeStatus	Data, status reading (FC:03)	PPJD (0x9025)
WriteSingleDiscreteOutput	Operation commands and data rewrite (FC:05)	-
SwitchSafetySpeed	Operation commands and data rewrite (FC:05)	SFTY (0x0401)
SwitchServo	Operation commands and data rewrite (FC:05)	SON (0x0403)
ResetAlarm	Operation commands and data rewrite (FC:05)	ALRS (0x0407)
ReleaseBreak	Operation commands and data rewrite (FC:05)	BKRL (0x0408)
Pause	Operation commands and data rewrite (FC:05)	STP (0x040A)
Home	Operation commands and data rewrite (FC:05)	HOME (0x040B)
StartPosition	Operation commands and	CSTR

Command Name	IAI PCON (Modbus)	
	Function code	Symbol (address)
	data rewrite (FC:05)	(0x040C)
SwitchJogInching	Operation commands and data rewrite (FC:05)	JISL (0x0411)
SwitchMode	Operation commands and data rewrite (FC:05)	MOD (0x0414)
GetCurrentPositionData	Operation commands and data rewrite (FC:05)	TEAC (0x0415)
JogPlus	Operation commands and data rewrite (FC:05)	JOG+ (0x0416)
JogMinus	Operation commands and data rewrite (FC:05)	JOG- (0x0417)
MoveToStartPosition	Operation commands and data rewrite (FC:05)	ST0 to ST7
Calibration	Operation commands and data rewrite (FC:05)	CLBR (0x0426)
SwitchPIOModbus	Operation commands and data rewrite (FC:05)	PMSL (0x0427)
Stop	Operation commands and data rewrite (FC:05)	STOP (0x042C)
SwitchAxisMove	Operation commands and data rewrite (FC:05)	ENMV (0x049B)
ProgramHome	Operation commands and data rewrite (FC:05)	PHOM (0x049C)
SearchStop	Operation commands and data rewrite (FC:05)	SSTP (0x049D)
ForceProgramStop	Operation commands and data rewrite (FC:05)	FPST (0x049E)
ProgramStart	Operation commands and data rewrite (FC:05)	PSTR (0x049F)
WriteSingleRegister	Direct entry of the control information (FC:06)	-
WriteMultipleRegisters	Direct entry of the positioning data (FC:10)	-
MovePTP	Direct entry of the	PCMD,INP,VCMD,ACMD,PPOW,CTLF

Command Name	IAI PCON (Modbus)	
	Function code	Symbol (address)
	positioning data (FC:10)	(0x9900 to 9908)
WritePositionData	Direct entry of the positioning data (FC:10)	PCMD,INP,VCMD,ZNMP,ZNLP,ACMD, DCMD,PPOW,LP0OW,CTLF (Starting address=0x1000+(0x10×Position No.))

2.2.3.1. CaoRobot::Execute("ReadMultipleRegisters") command

Reads values from the multiple registers specified.

Argument:

VT_ARRAY | VT_I4		
0	VT_I4	Specify the starting address of the reading registers.
1	VT_I4	Specify the reading register count.

Return value:

VT_ARRAY | VT_UI2		
n	VT_UI2	Values in the sequence reading register that starts from the starting address.

:Example

```
Dim vArray As Variant
vArray = caoRobot.Execute("ReadMultipleRegisters" ,Array(1284, 2))
//values in 0x0504(1284) and 0x0505(1285) are obtained and stored in vArray.
```

2.2.3.2. CaoRobot::Execute("ReadAlarmInfo") command

Obtains the latest alarm information.

Argument : none

Return value:

VT_ARRAY | VT_VARIANT		
0	VT_UI2	An alarm detail code.
1	VT_UI2	An alarm address
2	VT_UI4	An alarm code
3	VT_UI4	Alarm occurrence time (If RTC is not available, this shows the elapsed time from the controller power-ON.[ms])

Example:

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

2.2.3.3. CaoRobot::Execute("ReadPositionData") command

Obtains position data.

Argument :

VT_ARRAY | VT_UI2		
0	VT_UI2	Specify the reading start position number.
1	VT_UI2	Specify the reading target record count.

Return value:

VT_ARRAY | VT_VARIANT				
	n	VT_ARRAY | VT_VARIANT		
		0	VT_I4	Target position (unit: 0.01mm)
		1	VT_UI4	Positioning band (unit: 0.01mm)
		2	VT_UI4	Speed command (unit:0.01mm/s)
		3	VT_I4	Individual zone boundary positive side (unit: 0.01mm)
		4	VT_I4	Individual zone boundary negative side (unit: 0.01mm)
		5	VT_UI2	Acceleration command (unit: 0.01G)
		6	VT_UI2	Deceleration command (unit: 0.01G)
		7	VT_UI2	Push-current limiting value (100%=0xFF)
		8	VT_UI2	Load current threshold (100%=0xFF)
		9	VT_UI2	Control flag specification

Example:

```
Dim vArray As Variant
vArray = caoRobot.Execute("ReadPositionData" ,Array(10, 2))
// position data values of Position No.10 and 11 are obtained and then stored in vArray.
```

2.2.3.4. CaoRobot::Execute("ReadMoveCount") command

Obtains the total moving count.

Argument : none

Return value:

VT_UI4	Total moving count

Example:

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

2.2.3.5. CaoRobot::Execute("ReadMoveDistance") command

Obtains the total moving distance.

Argument : none

Return value:

VT_UI4	Total moving distance (unit: 1m)

Example:

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

2.2.3.6. CaoRobot::Execute("ReadCurrentTime") command

Obtains the current time.

Argument : none

Return value:

VT_UI4	Current time

Example:

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

2.2.3.7. CaoRobot::Execute("ReadFanDriveTime") command

Obtains the total fan driving time.

Argument: none

Return value:

VT_UI4	Total fan driving time (unit: 1s)

Example:

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

2.2.3.8. CaoRobot::Execute("ReadCurrentPosition") command

Obtains the current position.

Argument: none

Return value:

VT_I4	Current position (unit: 0.01mm)

Example:

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

2.2.3.9. CaoRobot::Execute("ReadCurrentAlarmCode") command

Obtains the alarm code currently issued.

Argument: none

Return value:

VT_UI2	An alarm code currently issued

Example:

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

2.2.3.10. CaoRobot::Execute("ReadIOPortInputSignal") command

Obtains the value of IO port input signal.

Argument: none

Return value:

VT_UI2	The value of IO port input signal

Example:

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

2.2.3.11. CaoRobot::Execute("ReadIOPortOutputSignal") command

Obtains the value of IO port output signal.

Argument: none

Return value:

VT_UI2	The value of IO port output signal

Example:

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

2.2.3.12. CaoRobot::Execute("ReadDSS1") command

Obtains the value of device status register 1.

Argument: none

Return value:

VT_UI2	The value of device status register 1

Example:

```
Dim lVal As Long
lVal = caoRobot.Execute("ReadDSS1")
```

2.2.3.13. CaoRobot::Execute("ReadDSS2") command

Obtains the value of device status register 2.

Argument: none

Return value:

VT_UI2	The value of device status register 2

Example:

```
Dim lVal As Long
lVal = caoRobot.Execute("ReadDSS2")
```

2.2.3.14. CaoRobot::Execute("ReadDSSE") command

Obtains the value of the expansion device status register.

Argument: none

Return value:

VT_UI2	The value of the expansion device status register

Example:

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

2.2.3.15. CaoRobot::Execute("ReadSTAT") command

Obtains the value of the system status register.

Argument: none

Return value:

VT_UI2	The value of the system status register

Example:

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

2.2.3.16. CaoRobot::Execute("ReadCurrentSpeed") command

Obtains the monitor data of the current motor speed.

Argument: none

Return value:

VT_I4	The monitor data of the actual motor speed (unit:0.01mm/s)

Example:

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

2.2.3.17. CaoRobot::Execute("ReadCurrentValue") command

Obtains the motor current monitor data in ampare unit.

Argument: none

Return value:

VT_UI4	The monitor data of the motor current value (unit: 1mA)

Example:

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

2.2.3.18. CaoRobot::Execute("ReadDeviation") command

Obtains the deviation between the position command value and the feedback value (actual position) in every 1ms interval.

Argument: none

Return value:

VT_UI4	Deviation between the position command value and the feedback value (actual position) in every 1ms interval (unit: Pulse)

Example:

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

2.2.3.19. CaoRobot::Execute("ReadIntegrationTime") command

Obtains the total Power-ON time after the controller power-ON.

Argument: none

Return value:

VT_UI4	Total power-ON time after the controller turn-ON is returned. (unit: ms)

Example:

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

2.2.3.20. CaoRobot::Execute("ReadSpecialInputPort") command

Obtains the value of the special input port monitor register.

Argument: none

Return value:

VT_UI2	The value of the special input port monitor register

Example:

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

2.2.3.21. CaoRobot::Execute("ReadZoneStatus") command

Obtains the value of the zone status register.

Argument: none

Return value:

VT_UI2	The value of the zone status register

Example:

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

2.2.3.22. CaoRobot::Execute("ReadCompletePositionNo") command

Obtains the value of the position number status register.

Argument: none

Return value:

VT_UI2	The value of the position number status register

Example:

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

2.2.3.23. CaoRobot::Execute("ReadSSSE") command

Obtains the value of the expansion system status register.

Argument: none

Return value:

VT_UI2	The value of the expansion system status register

Example:

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

2.2.3.24. CaoRobot::Execute("ReadLoad") command

Obtains the value of the monitor data of the load cell measurement.

Argument: none

Return value:

VT_UI4	Value of the monitoring data of the load cell measurement (unit: 0.01N)

Example:

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

2.2.3.25. CaoRobot::Execute("ReadLoadLevel") command

Obtains the load level currently charged to the motor in ratio.

Argument: none

Return value:

VT_UI4	Load level currently charged to the motor (unit:1%)

Example:

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

2.2.3.26. CaoRobot::Execute("ReadPressProgramAlarmCode") command

Obtains the value of the press program alarm code.

Argument: none

Return value:

VT_UI2	An alarm code currently issued in the press program

Example:

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

2.2.3.27. CaoRobot::Execute("ReadAlarmPressProgramNo") command

Obtains the press program number that issues an alarm.

Argument: none

Return value:

VT_UI2	Press program number that issues an alarm

Example:

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

2.2.3.28. CaoRobot::Execute("ReadPressProgramStatus") command

Obtains the value of the press program status register.

Argument: none

Return value:

VT_UI2	The value of the press program status register

Example:

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

2.2.3.29. CaoRobot::Execute("ReadPressProgramJudgeStatus") command

Obtains the value of the press program judgment status register.

Argument: none

Return value:

VT_UI2	The value of the press program judgment status register

Example:

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

2.2.3.30. CaoRobot::Execute("WriteSingleDiscreteOutput") command

Switch ON/OFF the slave DO.

Argument :

VT_ARRAY | VT_VARIANT			
	0	VT_I4	Specify the writing target address.
	1	VT_BOOL	Specify the value to write.

Return value : none

Example:

```
caoRobot.Execute("WriteSingleDiscreteOutput", Array(1027, True))
// Set “ON(true)” to the address 0x0403(1027)
```

2.2.3.31. CaoRobot::Execute("SwitchSafetySpeed") command

Enables/Disables the safety speed.

Argument :

VT_BOOL	Specify the safety speed status True: Enable the safety speed False: Disable the safety speed

Return value : none

Example:

```
caoRobot.Execute("SwitchSafetySpeed", True)
// Enable the safety speed
```

2.2.3.32. CaoRobot::Execute("SwitchServo") command

Turns ON/OFF the servo.

Argument :

VT_BOOL	Specify the servo-ON/OFF status True: Servo ON False: Servo OFF

Return value : none

Example:

```
caoRobot.Execute("SwitchServo", True)
// Set the servo-ON.
```

2.2.3.33. CaoRobot::Execute("ResetAlarm") command

Resets the alarm.

Argument :

VT_BOOL	Specify the status of the alarm reset. True : Reset the alarm False: Normal (To reset the alarm, set this command False and then switch to True)

Return value: none

Example:

caoRobot.Execute("ResetAlarm", False)

caoRobot.Execute("ResetAlarm", True)

// reset an alarm by switching the command False > True.

2.2.3.34. CaoRobot::Execute("ReleaseBreak") command

Releases the brake forcibly.

Argument :

VT_BOOL	Specify the status of the forcible brake release. True: Forcefully release the brake False: Normal

Return value : none

Example:

caoRobot.Execute("ReleaseBreak", True)

// Enable the forcible brake release

2.2.3.35. CaoRobot::Execute("Pause") command

Suspends the operation.

Argument :

VT_BOOL	Specify the status of pause. True: Pause False: Normal

Return value : none

Example:

caoRobot.Execute("Pause", True)

// Pause the operation.

2.2.3.36. CaoRobot::Execute("Home") command

Returns to the home position.

Argument :

VT_BOOL	Specify the Home-return status. True: Return to the home position False: Normal (To return to the home position, set the command False and then switch to True)

Return value : none

Example:

```
caoRobot.Execute("Home", False)
caoRobot.Execute("Home", True)
// set the command “False” to “True” to return to the home position.
```

2.2.3.37. CaoRobot::Execute("StartPosition") command

Moves to the predetermined start position which is specified by the position number.

Argument :

VT_BOOL	Specify whether or not to move to the start position. True: Position start command False: Normal (To move to the start position, set this command False and then switch to True)

Return value : none

Example:

```
caoRobot.Execute("StartPosition", False)
caoRobot.Execute("StartPosition", True)
// set the command “False” to “True” to move to the start position.
```

2.2.3.38. CaoRobot::Execute("SwitchJogInching") command

Switches between jogging and inching.

Argument :

VT_BOOL	Specify the jogging/inching operation status. True: Inching operation False: Jogging operation

Return value : none

Example:

```
caoRobot.Execute("SwitchJogInching", True)
// specify the inching operation
```

2.2.3.39. CaoRobot::Execute("SwitchMode") command

Switches between the teaching mode and the normal operation mode.

Argument :

VT_BOOL	Specify the normal mode/teaching mode status. True: teaching mode False: normal operation mode

Return value : none

Example:

caoRobot.Execute("SwitchMode", True)

// specify the teaching mode

2.2.3.40. CaoRobot::Execute("GetCurrentPositionData") command

Obtains the position data.

Argument :

VT_BOOL	Specify the position data reading status. True: Read the position data False: Normal

Return value : none

Example:

caoRobot.Execute("GetCurrentPositionData", True)

// Specify to read the position data

2.2.3.41. CaoRobot::Execute("JogPlus") command

Executes jogging or inching motion to the opposite to the home direction.

Argument :

VT_BOOL	Specify the status of the jogging or inching motion to the opposite to the home direction. True: Jog to the opposite to the home direction False: Normal

Return value : none

Example:

caoRobot.Execute("JogPlus", True)

// Execute the jogging operation to the opposite to the home direction (+)

2.2.3.42. CaoRobot::Execute("JogMinus") command

Executes jogging or inching to the home direction.

Argument :

VT_BOOL	Specifies the status of the jogging or inching to the home direction. True: Jog to the home direction False: Normal

Return value : none

Example:

caoRobot.Execute("JogMinus", True)

// Execute the jogging operation to the home direction (-)

2.2.3.43. CaoRobot::Execute("MoveToStartPosition") command

Moves to the start position.

Argument :

VT_ARRAY | VT_VARIANT		
0	VT_UI2	Specify the start position number.
1	VT_BOOL	Specify the status of operation command True: operation command ON False: operation command OFF

Return value : none

Example:

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

// Move to the position No.1

2.2.3.44. CaoRobot::Execute("Calibration") command

Calibrates the dedicated load cell.

Argument :

VT_BOOL	Specify the calibration execution status of the dedicated load cell. True: Execute calibration False: Normal operation

Return value : none

Example:

caoRobot.Execute("Calibration", True)

// Execute calibaration

2.2.3.45. CaoRobot::Execute("SwitchPIOModbus") command

Enables/disables the Modbus command.

Argument :

VT_BOOL	Enables/disables the Modbus command True: Enable the Modbus command False: Disable the Modbus command

Return value : none

Example:

caoRobot.Execute("SwitchPIOModbus", True)

// Enable the Modbus command

2.2.3.46. CaoRobot::Execute("Stop") command

Starts deceleration to a stop.

Argument :

VT_BOOL	Specify the status of the deceleration stop command. True: Deceleration stop. False: Normal

Return value : none

Example:

caoRobot.Execute("Stop", True)

// Decelerate to a stop

2.2.3.47. CaoRobot::Execute("SwitchAxisMove") command

Allows/prohibits the axis operation.

Argument:

VT_BOOL	Enable/disable the axis operation True: Axis operation permitted False: Axis operation prohibited

Return value : none

Example:

caoRobot.Execute("SwitchAxisMove", True)

// Permit the axis operation

2.2.3.48. CaoRobot::Execute("ProgramHome") command

Moves to the program home position.

Argument:

VT_BOOL	Specify the program home position return status. True: Return to the program home position False: Normal

	(To return to the program home position, set the command False and then switch to True)

Return value : none

Example:

caoRobot.Execute("ProgramHome", False)

caoRobot.Execute("ProgramHome", True)

// Return to the program home position by setting False > True

2.2.3.49. CaoRobot::Execute("SearchStop") command

Specifies whether to finish the press program or not after the complete of the search operation.

Argument:

VT_BOOL	Specify whether to finish the press program or not after the complete of the search operation. True: Stop the program after the complete of search operation False: Not stop the program after the complete of search operation

Return value: none

Example:

caoRobot.Execute("SearchStop", True)

// stop the program after the complete of search operation

2.2.3.50. CaoRobot::Execute("ForceProgramStop") command

Forcibly competes the press program.

Argument:

VT_BOOL	Specifies the press program forcibly stop True: Forcibly stop the press program False: Normal (To stop the press program forcibly, set the command False and then switch to True.)

Return value: none

Example:

caoRobot.Execute("ForceProgramStop", False)

caoRobot.Execute("ForceProgramStop", True)

// Forcibly stop the press program

2.2.3.51. CaoRobot::Execute("ProgramStart") command

Starts the press program.

Argument :

VT_BOOL	Start the press program. True: Program start False: Normal (To start the press program, set the command False and then switch to True.)

Return value : none

Example:

caoRobot.Execute("ProgramStart", False)

caoRobot.Execute("ProgramStart", True)

// Start the presss program by specifying False > True

2.2.3.52. CaoRobot::Execute("WriteSingleRegister") command

Writes data to a specified address register.

Argument :

VT_ARRAY | VT_VARIANT			
	0	VT_I4	Specify the writing start address.
	1	VT_UI2	Specify the data to write.

Return value : none

Example:

caoRobot.Execute("WriteSingleRegister", Array(3328, 4096))

// write 0x1000(4096) to the address of 0x0D00(3328) (servo-ON)

caoRobot.Execute("WriteSingleRegister", Array(3328, 4112))

// write 0x1010(4112) to the address of 0x0d00(3328) (return-to-home)

2.2.3.53. CaoRobot::Execute("WriteMultipleRegisters") command

Writes data to the sequence address register starting from the specified address.

Argument :

VT_ARRAY | VT_VARIANT				
	0	VT_I4		Specify the writing start address.
	1	VT_I4		Specify the writing data count (register count).
	2	VT_ARRAY | VT_UI2		
		n	VT_UI2	Specify the data to write.

Return value : none

Example:

caoRobot.Execute("WriteMultipleRegisters", Array(39168, 2, Array(0, 5000)))

// For each 0x9900(39168) and 0x9901(39169), write 0 and 5000. (Set the target position of position No.0 to 50.00mm)

2.2.3.54. CaoRobot::Execute("MovePTP") command

Moves to a target position by entering a numerical value.

Argument :

VT_ARRAY | VT_VARIANT		
0	VT_I4	Specify the target position. (unit: 0.01mm)
1	VT_UI4	Specify the positioning band. (unit: 0.01mm)
2	VT_UI4	Specify the speed. (unit:0.01mm/s)
3	VT_UI2	Specify the acceleration/deceleration. (unit: 0.01G)
4	VT_UI2	Specify the push-current limiting value. (unit: %)
5	VT_UI2	Specify the control flag.

Return value : none

Example:

```
caoRobot.Execute("MovePTP", Array(5000, 10, 10000, 30, 0, 8)
// Target position 50.00mm
// Positioning band 0.10mm
// Speed 100.00mm/s
// acceleration/deceleration 0.30G
// push-current limiting value 0%
```

2.2.3.55. CaoRobot::Execute("WritePositionData") command

Writes position data.

Argument :

VT_ARRAY | VT_VARIANT		
0	VT_UI2	Specify the writing target position number.
1	VT_I4	Specify the target position. (unit: 0.01mm)
2	VT_UI4	Specify the positioning band. (unit: 0.01mm)
3	VT_UI4	Specify the speed command. (unit:0.01mm/s)
4	VT_I4	Specify the individual zone boundary positive side (+). (unit: 0.01mm)
5	VT_I4	Specify the individual zone boundary negative side (-). (unit: 0.01mm)
6	VT_UI2	Specify the acceleration command. (unit: 0.01G)
7	VT_UI2	Specify the deceleration command. (unit: 0.01G)

	8	VT_UI2	Specify the push-current limiting value. (100%=0xFF)
	9	VT_UI2	Specify the load current threshold (100%=0xFF)
	10	VT_UI2	Specify the control flag specification.

Return value : none

Example:

```
caoRobot.Execute("WritePositionData", Array(12, 10000, 10, 20000, 6000, 4000, 10, 30, 0, 0, 0)
// Target position number     No.12
// Target position 100.00mm
// Positioning band 0.10mm
// Speed 100.00mm/s
// individual zone boundary positive side (+) 60.00mm
// individual zone boundary negative side (-) 40.00mm
// Acceleration 0.10G
// Deceleration 0.30G
```

2.2.4. CaoRobot::AddVariable method

When creating Variable object from CaoController, the data to be obtained from the connected PCON device can be determined, by specifying the variable name.

The following shows the specification of AddVariable.

Syntax

```
AddVariable
(
"<Variable name >",        // Variable name
"<Option>"         // Option character string
)
```

For about available variable names and options, and detail information, see Variable list.

2.2.5. CaoRobot::get_VariableNames method

Obtains a variable name list shown in Table 2-5.

2.2.6. CaoVariable::get_Value property

Reads data from a PCON device with a specified option.

2.2.7. CaoVariable::put_Value property

Writes data from a PCON device with a specified option.

2.3. Variable list

2.3.1. Controller class

Table 2-5 shows that a variable list that is available for AddVariable of Controller class.

Table 2-5 Robot class variable list

Variable name	Description	Value		Link
		get	put	
@MAKER_NAME	Obtains the manufacturer’s name.	✓	-	P.39
@VERSION	Obtains version information.	✓	-	P.39

※For <??>, enter any character strings.

2.3.1.1. @MAKER_NAME

Obtains the manufacturer’s name.

Option: none

Return value data structure of get_value

VT_BSTR	Fixed text :　IAI

2.3.1.2. @VERSION

Obtains the version information.

Option: none

Return value data structure of get_value

VT_BSTR	Character string with the format of “*.*.*” that shows the currently used DLL version.

2.3.2. Robot class

Table 2-5 shows a variable list that is available for AddVariable of Robot class.

Table 2-6 Robot class variable list

Variable name	Description	Value		Link
		get	put	
REGISTER<??>	Reads and sets the value of the specified register.	✓	✓	P.42
@ALARM_INFO	Obtains the alarm information that has occured the last.	✓	-	P.42
POSITION_DATA<??>	Reads and set the specified position data.	✓	✓	P.42
@MOVE_COUNT	Obtains the total moving count.	✓	-	P.43
@MOVE_DISTANCE	Obtains the total moving distance.	✓	-	P.43
@CURRENT_TIME	Obtains the current time.	✓	-	P.43
@FAN_DRIVE_TIME	Obtains the total fan driving time.	✓	-	P.43
@CURRENT_POSITION	Obtains the current position.	✓	-	P.43
@ALARM_CODE	Obtains an alarm code that is currently issued.	✓	-	P.44
@IO_INPUT	Obtains the value of the IO port input signal.	✓	-	P.44
@IO_OUTPUT	Obtains the value of the IO port output signal.	✓	-	P.44
@DSS1	Obtains the value of the device status register 1.	✓	-	P.44
@DSS2	Obtains the value of the device status register 2.	✓	-	P.44
@DSSE	Obtains the value of the expansion device status register.	✓	-	P.44
@STAT	Obtains the value of the system status register.	✓	-	P.45
@SPEED	Obtains the monitor data of the actual motor speed.	✓	-	P.45
@VALUE	Obtains the monitor data of the motor current.	✓	-	P.45
@DEVIATION	Obtains the deviation between the position command value and the feedback value (actual position) in every 1ms interval.	✓	-	P.45
@INTEGRATION_TIME	Obtains the total Power-ON time after the controller power-ON.	✓	-	P.45
@SPECIAL_INPUT	Obtains the value of the special input port monitor register.	✓	-	P.45
@ZONE	Obtains the value of the zone status register.	✓	-	P.46
@COMPLETE_POSITION_NO	Obtains the value of the position number status register.	✓	-	P.46
@SSSE	Obtains the value of the expansion system status register.	✓	-	P.46

Variable name	Description	Value		Link
		get	put	
@LOAD	Obtains the value of the monitor data of the load cell measurement.	✓	-	P.46
@LOAD_LEVEL	Obtains the load level currently charged to the motor in ratio.	✓	-	P.46
@PRESS_PROGRAM_ALARM_CODE	Obtains the value of the press program alarm code.	✓	-	P.46
@ALARM_PRESS_PROGRAM_NO	Obtains the press program number that issues an alarm.	✓	-	P.47
@PRESS_PROGRAM_STATUS	Obtains the value of the press program status register.	✓	-	P.47
@PRESS_PROGRAM_JUDGE	Obtains the value of the press program judgment status register.	✓	-	P.47

※For <??>, enter any character strings.

2.3.2.1. REGISTER<??>

Reads and sets the value of the specified register.

Option:

ADDRESS	Specify the starting address (0 or larger)
NUM	Specify the target register count to control.

Return value data structure of get_value

VT_ARRAY | VT_UI2		
n	VT_UI2	Values in the sequence reading register that starts from the starting address.

Argument data structure of put_value :

The same data structure of the return value of get_value

2.3.2.2. @ALARM_INFO

Obtains the alarm information that has occurred the last.

Option: none

Return value data structure of get_value

VT_ARRAY | VT_VARIANT		
0	VT_UI2	Alarm detail code
1	VT_UI2	Alarm address
2	VT_UI4	Alarm code
3	VT_UI4	Alarm occurrence time (If RTC is not available, this shows the elapsed time from the controller power-ON.[ms])

2.3.2.3. POSITION_DATA<??>

Read and set the specified position data.

Option:

POSITION_NO	Specify the position number (0 or larger).

Return value data structure of get_value

VT_ARRAY | VT_VARIANT		
0	VT_I4	Target position (unit: 0.01mm)
1	VT_UI4	Positioning band (unit: 0.01mm)
2	VT_UI4	Speed command (unit: 0.01mm/s)
3	VT_I4	individual zone boundary positive side (+) (unit: 0.01mm)
4	VT_I4	individual zone boundary negative side (-) (unit: 0.01mm)
5	VT_UI2	Acceleration command (unit: 0.01G)
6	VT_UI2	Deceleration command (unit: 0.01G)
7	VT_UI2	Push-current limiting value (100%=0xFF)

	8	VT_UI2	Load current threshold (100%=0xFF)
	9	VT_UI2	Control flag specification

Argument data structure of get_value

Same as the return value data structure of get_value

2.3.2.4. @MOVE_COUNT

Obtains the total moving count.

Option: none

Return value data structure of get_value

VT_UI4	Total moving count

2.3.2.5. @MOVE_DISTANCE

Obtains the total moving distance.

Option: none

Return value data structure of get_value

VT_UI4	Total moving distance

2.3.2.6. @CURRENT_TIME

Obtains the current time.

Option: none

Return value data structure of get_value

VT_UI4	Current time

2.3.2.7. @FAN_DRIVE_TIME

Obtains the total fan driving time.

Option: none

Return value data structure of get_value

VT_UI4	Total fan driving time (unit: 1s)

2.3.2.8. @CURRENT_POSITION

Obtains the current position.

Option: none

Return value data structure of get_value

VT_I4	Current position (unit: 0.01mm)

2.3.2.9. @ALARM_CODE

Obtains an alarm code that is currently issued.

Option: none

Return value data structure of get_value

VT_UI2	An alarm code that is currently issued

2.3.2.10. @IO_INPUT

Obtains the value of the IO port input signal.

Option: none

Return value data structure of get_value

VT_UI2	The value of the IO port input signal.

2.3.2.11. @IO_OUTPUT

Obtains the value of the IO port output signal.

Option: none

Return value data structure of get_value

VT_UI2	The value of the IO port output signal.

2.3.2.12. @DSS1

Obtains the value of the device status register 1.

Option: none

Return value data structure of get_value

VT_UI2	The value of the device status register 1.

2.3.2.13. @DSS2

Obtains the value of the device status register 2.

Option: none

Return value data structure of get_value

VT_UI2	The value of the device status register 2.

2.3.2.14. @DSSE

Obtains the value of the expansion device status register.

Option: none

Return value data structure of get_value

VT_UI2	The value of the expansion device status register.

2.3.2.15. @STAT

Obtains the value of the system status register.

Option: none

Return value data structure of get_value

VT_UI4	The value of the system status register.

2.3.2.16. @SPEED

Obtains the monitor data of the actual motor speed.

Option: none

Return value data structure of get_value

VT_I4	The monitor data of the actual motor speed (unit: 0.01mm/s)

2.3.2.17. @VALUE

Obtains the monitor data of the motor current.

Option: none

Return value data structure of get_value

VT_UI4	The monitor data of the motor current (unit: 1mA)

2.3.2.18. @DEVIATION

Obtains the deviation between the position command value and the feedback value (actual position) in every 1ms interval.

Option: none

Return value data structure of get_value

VT_UI4	The deviation between the position command value and the feedback value (actual position) in every 1ms interval (unit: Pulse)

2.3.2.19. @INTEGRATION_TIME

Obtains the total Power-ON time after the controller power-ON.

Option: none

Return value data structure of get_value

VT_UI4	Obtains the total power-ON time after the controller power-ON.

2.3.2.20. @SPECIAL_INPUT

Obtains the value of the special input port monitor register.

Option: none

Return value data structure of get_value

VT_UI2	The value of the special input port monitor register

2.3.2.21. @ZONE

Obtains the value of the zone status register.

Option: none

Return value data structure of get_value

VT_UI2	The value of the zone status register.

2.3.2.22. @COMPLETE_POSITION_NO

Obtains the value of the position number status register.

Option: none

Return value data structure of get_value

VT_UI2	The value of the position number status register.

2.3.2.23. @SSSE

Obtains the value of the expansion system status register.

Option: none

Return value data structure of get_value

VT_UI2	The value of the expansion system status register

2.3.2.24. @LOAD

Obtains the value of the monitor data of the load cell measurement.

Option: none

Return value data structure of get_value

VT_UI4	The value of the monitor data of the load cell measurement (unit: 0.01N)

2.3.2.25. @LOAD_LEVEL

Obtains the load level currently charged to the motor in ratio.

Option: none

Return value data structure of get_value

VT_UI4	The load level currently charged to the motor (unit: 1%)

2.3.2.26. @PRESS_PROGRAM_ALARM_CODE

Obtains the value of the press program alarm code.

Option: none

Return value data structure of get_value

VT_UI2	An alarm code currently issued in the press program

2.3.2.27. @ALARM_PRESS_PROGRAM_NO

Obtains the press program number that issues an alarm.

Option: none

Return value data structure of get_value

VT_UI2	Press program number that issues an alarm

2.3.2.28. @PRESS_PROGRAM_STATUS

Obtains the value of the press program status register.

Option: none

Return value data structure of get_value

VT_UI2	The value of press program status register

2.3.2.29. @PRESS_PROGRAM_JUDGE

Obtains the value of the press program judgment status register.

Option: none

Return value data structure of get_value

VT_UI2	The value of the press program judgment status register

2.4. Error code

This provider has original error codes as shown below. (Refer to Table 2-7　Original error code list)

For ORiN2 common errors, refer to the error code section of ORiN2 programming guide.

Table 2-7 Original error code list

Error name	Error number	Description
E_CAOP_ILLEGAL_ARGUMENT	0x80100F01	Argument error. Argument handed to a command is invalid or out of available range.
E_CAOP_ILLEGAL_STATE	0x80100F02	Status error. A function is invoked by abnormal state. If the protocol has not been opened normally, this return code will be returned by all functions.
E_CAOP_ILLEGAL_SLAVE_ADDRESS	0x80100F05	Invalid server device address. Address 0 is used by a function that does not support broadcasting.
E_CAOP_OPEN	0x80100F42	Port open error or socket open error. Failed to open TCP/IP socket or serial port. If the error is serial port open error, the specified serial port may not exist in the system.
E_CAOP_FTALK_PORT_ALREADY_OPEN	0x80100F43	Serial port has already opened. A serial port designated for open operation has already been taken by other application.
E_CAOP_FTALK_TCPIP_CONNECT	0x80100F44	TCP/IP connection error. Failed to establish TCP/IP connection. This error occurs when a host exists on the network or on IP address, or the name of host is incorrect. Remote host needs to listen appropriate port number.
E_CAOP_CONNECTION_WAS_CLOSED	0x80100F45	Remote pier closed the TCP/IP connection. This error notifies that TCP/IP

Error name	Error number	Description
		connection was closed or damaged by a remote pier.
E_CAOP_SOCKET_LIB	0x80100F46	Socket library error. Failed to load TCP/IP socket library (such as WINSOCK). DLL may not be found or may not be installed.
E_CAOP_PORT_ALREADY_BOUND	0x80100F47	TCP port has already been bound. This error notifies that the specified TCP port cannot be bound. A port may already been occupied by other application or may not been released by TCP/IP stack for reuse.
E_CAOP_LISTEN_FAILED	0x80100F48	Failed to listen. Failed to listen the specified TCP port.
E_CAOP_FILEDES_EXCEEDED	0x80100F49	File descriptor exceeds the available range. File descriptor exceeds the maximum limit.
E_CAOP_PORT_NO_ACCESS	0x80100F4A	There is no permission to access the serial port or TCP port. For a serial port error, change the access permission. If a TCP/IP port error occurrs, the specified TCP port number is out of the IPPORT_RESERVED range..
E_CAOP_PORT_NOT_AVAIL	0x80100F4B	TCP port is not available. The specified TCP port is not available in this operation environment.
E_CAOP_LINE_BUSY	0x80100F4C	Serial line is busy. Serial line receives any noise or other signals although it should not have any traffic.
E_CAOP_CHECKSUM	0x80100F81	Checksum error. Checksum of received frame is invalid.
E_CAOP_INVALID_FRAME	0x80100F82	Invalid frame error.

Error name	Error number	Description
		The received frame does not correspond to any structure or content in communication protocol or does not match with the frame of query that has been sent before.
E_CAOP_INVALID_REPLY	0x80100F83	Invalid reply error. The received reply frame does not correspond to the communication protocol.
E_CAOP_REPLY_TIMEOUT	0x80100F84	Timeout error. This error may occur when a server device does not respond within the predetermined time or does not respond completely. Incorrect server device address may cause this error.
E_CAOP_SEND_TIMEOUT	0x80100F85	Send timeout error. This error notifies that the data transmission has been time-out. This error may occur when the handshake line is not configured properly.
E_CAOP_INVALID_MBAP_ID	0x80100F86	Invalid identifier. Protocol or transaction identifier is invalid. TCP server device needs to return the identifier received from TCP client.
E_CAOP_MBUS_EXCEPTION_RESPONSE	0x80100FA0	This error notifies that a Modbus exception response message has been received.
E_CAOP_MBUS_ILLEGAL_FUNCTION_RESPONSE	0x80100FA1	This error notifies that an exception response (code 01) of Modbus invalid function has been received.
E_CAOP_MBUS_MBUS_ILLEGAL_ADDRESS_RESPONSE	0x80100FA2	This error notifies that an exception response (code 02) of Modbus invalid data address has been received.
E_CAOP_MBUS_ILLEGAL_VALUE_RESPONSE	0x80100FA3	This error notifies that an exception

Error name	Error number	Description
		response (code 03) of Modbus invalid value is received.
E_CAOP__MBUS_SLAVE_FAILURE_RESPONSE	0x80100FA4	This error notifies that an exception response (code 04) of Modbus slave failure has been received.
E_MACHINE_TYPE_UNSPECIFIED	0x80110001	Selected device model is not supported.

3. Sample program

The following shows the sample program that writes data in a PCON device.

Precondition:

- The target of this program is RC8 PAC script.
- Both of COM port and device address of PCON device are [1].

List 3-1 **Sample.pcs**

```
Sub Main

‘Object
Dim caoCtrl as Object
Dim caoRobot as Object

'Create a controller object
caoCtrl = cao.AddController("PCON", "CaoProv.IAI.PCON", "", "conn=com:1")

'Create a robot object
caoRobot = caoCtrl.AddRobot("Robot", "UnitAddress=1")

‘Servo ON
caoRobot.Execute "SwitchServo", true

‘Return to the home position (execute at the rising edge)
caoRobot.Execute "Home", 0
caoRobot.Execute "Home", 1
‘Wait until the return-to-home position completes
delay 10000

‘Change the position (execute at the rising edge)
caoCtrl.Execute "StartPosition", 0
caoCtrl.Execute "StartPosition", 1
‘Wait until the position change completes
delay 10000

End Sub
```