espec Ltd.
Espec TempHumi provider

Version 1.0.0

User's guide

October 23, 2019

Remarks:

This document is translated into English using machine translation.

【 revision history 】

Version	Date	Content
1.0.0	2019-10-23	First edition.

【 ..confirming the operation.. model 】

Model	Version	Notes
STH-120	-	-
SH-642	-	-
SU-642	-	-

Contents

1. Introduction

This book small, environmental examination machine series (SU-***,SH-***), mini subzero series (MC-***), and light specs constant temperature (equilibrium humidity) is container series (LH-***,LHL-***,LHU-***,LU-***), and air forced ovens (LC-***,LG-***) of espec Ltd.. The user's guide of the provider that acquires and sets data for the device of vacuum dryness machine (LCV-***), small high temperature chamber series (ST-***,STH-***), stability examination machine (CSH), stability examination room (CWH), and low temperature constant temperature [**] ..dampness.. [utsuwa] (CRH). About the corresponding model, "..it is in the appendix... Table6-3" and "Table6-4Refer to". The corresponding model of espec Ltd. is named TempHumi device at the following. Moreover, standard equipment names the device of Ethernet and it is named TempHumi RS-485 device that it is TempHumi Ethernet device, and standard equipment is a device of RS-485. Figure1-1However, it becomes a whole of device block diagram with this provider. This provider is named espec TempHumi provider later.

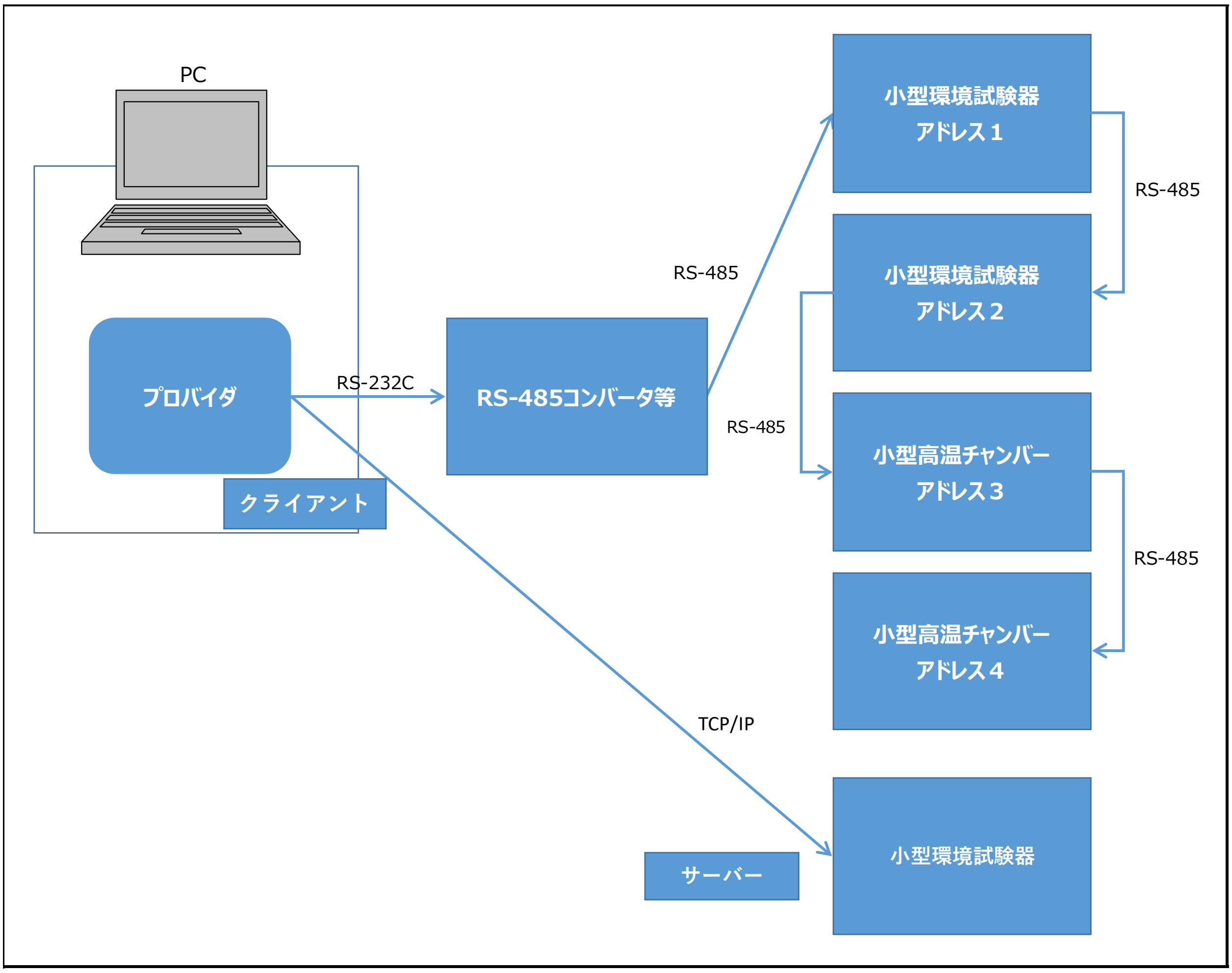


Figure1-1Block diagram

Moreover, please refer to figure 1-2 for this provider and correspondence each device.

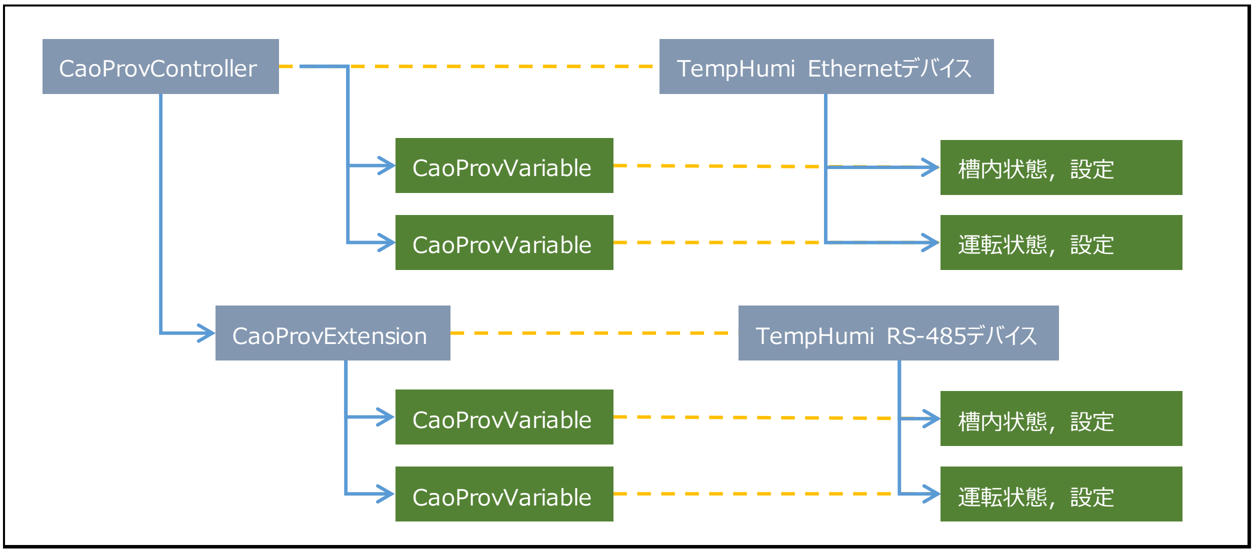


Figure1-2Espec TempHumi provider and correspondence chart with TempHumi device

1.1. Source that becomes reference

The espec TempHumi provider is developing referring to the user's manual offered by espec Ltd.. This manual is named the TempHumi RS-485 manual and TempHumi Ethernet manual later.

The future reference

- User's manual communication function RS-485 (material number 4009204000631)
- User's manual communication function Ethernet (material number 4000104005150)

2. Environmental setup for application development

2.1. Connection of TempHumi device and client PC

2.1.1. Connect it by serial communications.

TempHumi RS-485 device and client PC are connected by serial communications through RS-232C. Moreover, a standard mode and the delimiter correspond to the data transfer with CR+LF. Set it as follows in the connected TempHumi RS-485 device referring to "Chapter 2 environment setting" of the TempHumi RS-485 manual.

Set item	Set value
RS-485 forwarding mode setting	Standard mode
RS-485 delimiter setting	CR+LF

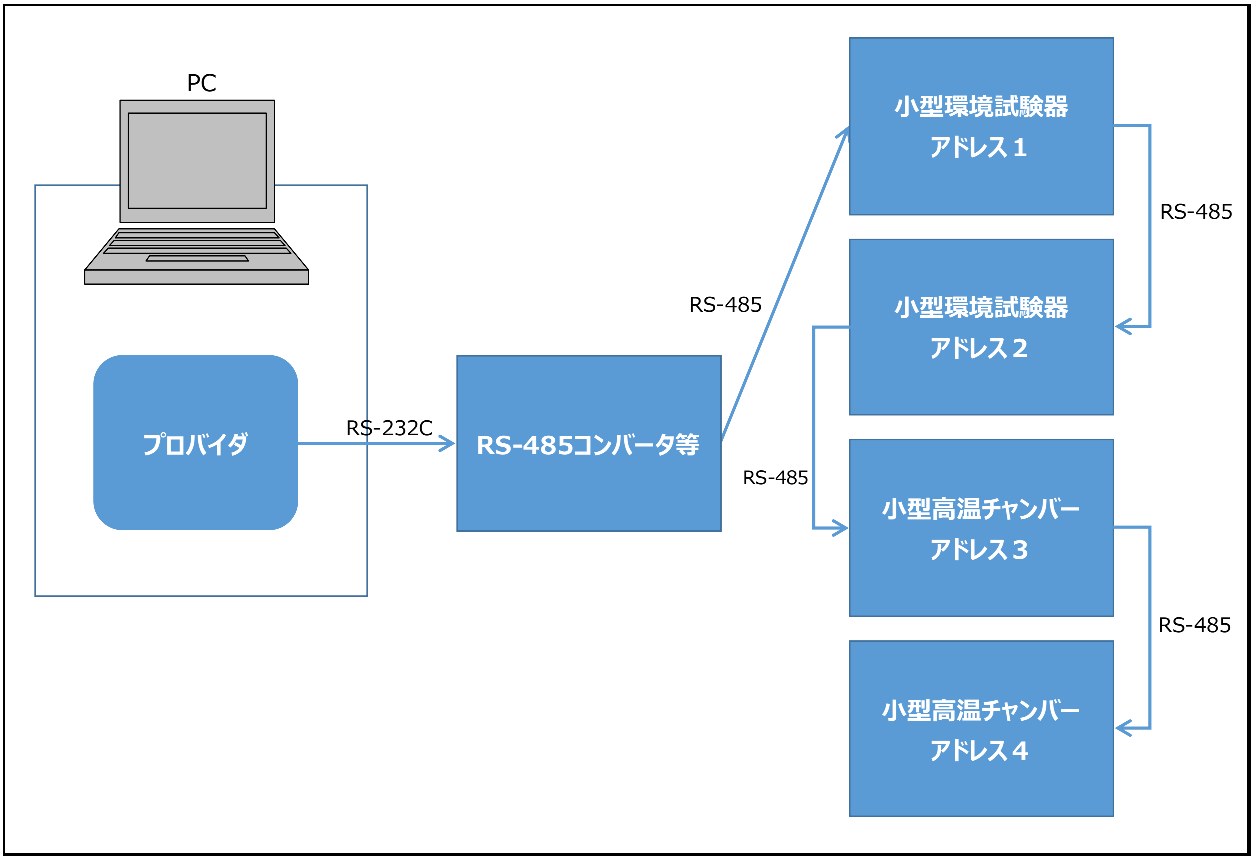


Figure2-1Block diagram at serial communications

2.1.2. Connect it by the Ethernet communication.

TempHumi Ethernet device and client PC are connected by the TCP/IP communication by Ethernet. It is not necessary to set it in the environmental setting because the delimiter is CR+LF fixation in the connected TempHumi Ethernet device.

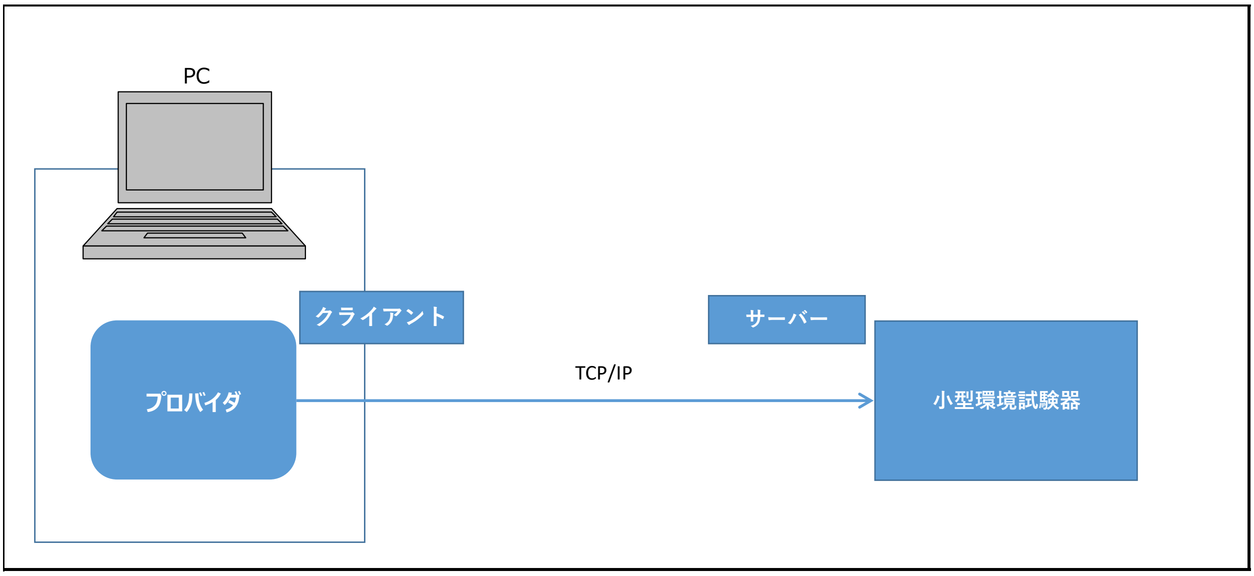


Figure2-2Block diagram when Ethernet is communicated

2.2. Setup of client PC development setting

2.2.1. Manual installation of espec TempHumi provider

It is necessary to register the following registry when installing it to use the espec TempHumi provider by hand power. Start the command prompt by the manager authority, and execute the regsvr32 command when you register the registry.

Table2-1Espec TempHumi provider

File name	CaoProvespecTempHumi.dll
ProgID	CaoProv.espec.TempHumi
Registry registration	regsvr32 CaoProvespecTempHumi.dll
Blotting out of registry registration	regsvr32 /u CaoProvespecTempHumi.dll

3. Command reference

3.1. Method/property list

Table3-1Method/property list

Category	[1]Method/property		Function	page
CaoWorkspace				
	AddController	M	Connect it with the controller.	P. 9
CaoController				
	VariableNames	P	Acquisition of variable identifier list that can be connected	P. 11
	AddExtension	M	Addition of enhancing board object	P. 11
	AddVariable	M	Addition of variable object	P. 12
	Execute	M	Execution of enhancing command	P. 13
CaoExtenstion				
	ID	P	Acquisition/setting of ID	P. 13
	VariableNames	P	Acquisition of variable identifier list that can be connected	P. 13
	AddVariable	M	Addition of variable object	P. 14
	Execute	M	Execution of enhancing command	P. 14
CaoVariable				
	Value	P	Acquisition/setting of value	P. 14

3.2. Method property

3.2.1. CaoWorkspace class

3.2.1.1. AddController method

Connect it with the corresponding TempHumi device referring to the parameter passed when the AddController method is executed. The specification of the AddController method is shown as follows.

Format

AddController

```
(
        ″< controller name >″          // Controller name (arbitrariness)
        ″CaoProv.espec.TempHumi″,      // Provider name (fixation)
        ″< machine name >″                     // Provider execution machine name
```

[1] Indicates M: method, P: property, E: event.

```
        ″< option >″                    // Optional character string
)
```

Option

The option specified for an optional character string is shown as follows. An optional character string becomes a character string to which each option shown in the following ties by comma (,).

Option	Indispensability	Explanation	Range of value	Default value
Conn	✓	Specify a cereal or optional Ethernet connection to connect it with the TempHumi device. Refer to 3.2.1.1.1.	--	--
Timeout	--	Specify connection/sending and receiving time-out by the unit of ms.	1 - 4294967295	500

Usage example

```
    Dim caoEng As CaoEngine      ' Engine object
    Dim caoWor As CaoWorkspace   ' Workspace object
    Dim caoCon As CaoController  ' Controller object

    Set caoEng = New CaoEngine
    Set caoWor = caoEng.Workspaces.Item(0)
    Set caoCon = caoWor.AddController(″Controller″, _
                                      ″CaoProv.espec.TempHumi″, _
                                      ″″, _
                                      ″Conn=com:1″)
```

3.2.1.1.1. Conn is optional.

Connected parameter character string of optional Conn is shown as follows. It is shown to omit it here in the brace) and the underlined part under the explanation of each parameter shows the default value when the option is not specified respectively.

When you connect the TempHumi RS-485 device (serial communications)

(when two or more TempHumi RS-485 devices are connectedFigure2-1Refer to [wo]) Do the RS-485 transmission rate setting, the RS-485 stop bit setting, the RS-485 data bit setting, and the RS-485 parity bit setting of each device thoroughly. Refer to ″Chapter

2 environment setting" of the TempHumi RS-485 manual and "Chapter 5 specification" for the setting of each device.

"Conn=Com:<COM Port>[:<BaudRate>[:<Parity>:<DataBits>:<StopBits>]]"
<COM Port> : COM port number.
<BaudRate> : Baud rate. 4800,9600,19200
<Parity> : Parity. 'N'-NONE, 'E'-EVEN, 'O'-ODD
<DataBits> : Data bit. '7'-7bit, '8'-8bit
<StopBits> : Stop bit. '1'-1bit, '2'-2bit

When you connect the TempHumi Ethernet device (TCP/IP communication)

The TempHumi Ethernet device uses the port number of 57732.

"Conn=Eth:<IP>[:<Port>]"
<IP> : Connection destination Internet Protocol address.
<Port> : Connection destination port. 57732

3.2.2. CaoController class

This class responds to the RS-485 converter when connecting it with the TempHumi RS-485 device by serial communications. Moreover, it corresponds to each TempHumi Ethernet device when connecting it by the TCP/IP communication.

3.2.2.1. VariableNames property

Acquire the variable identifier list. Describe the variable identifier acquired in this property later. It is possible to use it for one argument drinking.

Usage example

```
    ' connection
    Call Connect
    ' Variable identifier list  acquisition
    Dim variables as Variant
    variables = caoCon.VariableNames
    '  cutting
    Call Disconnect
```

3.2.2.2. AddExtension method

When the TempHumi RS-485 device is connected, it is necessary to add the CaoExtension object corresponding to each device to CaoController. Moreover, specify the address number set with the device by an optional character string. The specification of the AddExtension method is shown as follows.

Format

```
AddExtension
(
        ″< enhancing board name >″       // Enhancing board name (arbitrariness)
        ″< option >″                     // Optional character string
)
```

Option

The option specified for an optional character string is shown as follows. An optional character string becomes a character string to which each option shown in the following ties by comma (,).

Option	Indispensability	Explanation	Range of value	Default value
Addr	✓	Specify the address of the connected device. The address is necessary to identify two or more connected devices. Set a unique address to each device. (Refer to the address in Figure 2-1.)	1 – 16	--

Usage example

```
    ’  connection
    Call Connect
    ’  addition of extension board
    Dim caoExt As CaoExtension
    Set caoExt = caoCon.AddExtension(″Extension″, ″Addr = 1″)
    Call caoCon.Extensions.Remove(caoExt.Index)
    Set caoExt = Nothing
    ’  cutting
    Call Disconnect
```

3.2.2.3. AddVariable method

Add the variable object to CaoController. Only the variables shown in 3.4.1 can be used. The specification of AddVariable is shown below.

Format

```
AddVariable
(
        "< variable identifier >"               // Variable identifier
        "< option >"            // Optional character string (It is possible to omit it).
)
```

3.2.2.4. Execute method

Execute the enhancing command of CaoController. For the extended commands that can be specified with Execute, only the commands shown in 3.3.1 can be executed. The specification of Execute is shown below.

Format

```
Execute
(
        "< enhancing command name >"            // Enhancing command name
        "< optional character string >"         // Optional character string (It is possible
to omit it).
)
```

3.2.3. CaoExtension class

This provider is a class that accesses the connected each TempHumi RS-485 device from the CaoController class with which the cereal is connected.

3.2.3.1. ID property

Acquire the address of the extension board.

Data type

Type explanation	
VT_I4	The address is returned.

Usage example

```
    ' addition of extension board
    Dim caoExt As CaoExtension
    Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
    ' ID (address)  acquisition
    Dim address Long
    address = caoExt.ID
```

3.2.3.2. VariableNames property

Acquire the variable identifier list. The obtained variable name can be used as the first argument of AddVariable method.

Usage example

```
    '  addition of extension board
    Dim caoExt As CaoExtension
    Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
    ' Variable identifier list  acquisition
    Dim variables as Variant
    variables = caoExt.VariableNames
```

3.2.3.3. AddVariable method

Add the variable object to CaoExtension. T Only variable names shown in 3.4.2 can be specified as variable names.

The specification of AddVariable is shown as follows.

Format

AddVariable

```
(
        "< variable identifier >"               // Variable identifier
        "< option >"            // Optional character string (It is possible to omit it).
 )
```

3.2.3.4. Execute method

Execute the enhancing command of CaoExtension. Only the commands shown in 3.3.2 can be specified. The specification of Execute is shown as follows.

Format

Execute

```
(
        "< enhancing command name >"            // Enhancing command name
        "< optional character string >"         // Optional character string (It is possible
to omit it).
)
```

3.2.4. CaoVariable class

3.2.4.1. Value property

Get / set data from the connected device. Please refer to chapter 3.4 for details.

3.3. Enhancing command list

Define the enhancing command list that can be used in each class.

3.3.1. Enhancing command of CaoController class

Only when Ethernet is connected, the enhancing command of the CaoController class is effective. Each command is detailed and the usage example is described.

Table3-2List of enhancing command of CaoController class

Command	Explanation	Reference
Command related to set value of equipment		
GetROM	Acquire ROM version.	P. 17
GetROMDISP	Acquire ROM version of the indicator.	P. 17
GetMASK	Acquire the interrupt mask.	P. 17
SetMASK	Set the interrupt mask.	P. 18
GetTIMERMON	Acquire the started timer.	P. 19
GetTIMERSET	Acquire the set timer.	P. 20
GetTIMER0	Acquire the setting of a quick timer.	P. 21
SetTIMER0	Set the setting of a quick timer.	P. 22
GetTIMER1	Acquire the setting of driving beginning timer.	P. 22
SetTIMER1	Set the setting of driving beginning timer.	P. 23
GetTIMER2	Acquire the setting of the driving stop timer.	P. 23
SetTIMER2	Set the setting of the driving stop timer.	P. 24
GetKEYPROTECT	Acquire the protecting setting.	P. 24
SetKEYPROTECT	Set the protecting setting.	P. 24
GetTYPE	Acquire the chamber type.	P. 25
GetTEMP	Acquire the temperature parameter.	P. 26
SetTEMP	Set the temperature parameter.	P. 26
SetSTEMP	Set a set value of the temperature parameter.	P. 27
SetHTEMP	Set the upper bound value of the temperature parameter.	P. 27
SetLTEMP	Set the lower bound value of the temperature parameter.	P. 27
GetHUMI	Acquire the humidity parameter.	P. 28
SetHUMI	Set the humidity parameter.	P. 28
SetSHUMI	Set a set value of the humidity parameter.	P. 28
SetHHUMI	Set the upper bound value of the humidity parameter.	P. 29

Command	Explanation	Reference
SetLHUMI	Set the lower bound value of the humidity parameter.	P. 29
GetSET	Acquire the control value of the freezer.	P. 29
SetSET	Set the control value of the freezer.	P. 30
GetCONSTANTSETTEMP	Acquire the temperature setting value of the fixed command setting.	P. 30
GetCONSTANTSETHUMI	Acquire the humidity setting value of the fixed command setting.	P. 31
GetCONSTANTSETREF	Acquire the freezer setting value of the fixed command setting.	P. 31
GetSYSTEMSETPTS	Acquire the function of the monitor of the temperature of the sample.	P. 32
GetSYSTEMSETPTC	Acquire the function of the control of the temperature of the sample.	P. 32
GetSYSTEMSETPTCOPT	Acquire an optional function of the control of the temperature of the sample.	P. 32
SetTIMERMERASE	Delete the setting of the specified timer.	P. 33
SetTIMERON	Start the specified timer.	P. 33
SetTIMEROFF	Stop the specified timer.	P. 34
Command related to state of equipment		
GetDATE	Acquire the date of an internal calendar.	P. 34
SetDATE	Set the date of an internal calendar.	P. 35
GetTIME	Acquire the time of an internal calendar now.	P. 35
SetTIME	Set the time of an internal calendar now.	P. 35
GetSRQ	Acquire interrupt.	P. 36
SetSRQ	Reset interrupt.	P. 36
GetALARM	Acquire the alarm state.	P. 37
GetMODE	Acquire the mode of operation.	P. 37
SetMODE	Set the mode of operation.	P. 38
GetMODEDETAIL	Acquire a detailed mode of operation.	P. 38
GetMONDETAIL	Acquire the state in a detailed tank.	P. 39
GetHEATER	Acquire the state of the heater.	P. 40
Command related to program		
GetRUNPRGDATA	Acquire the remote program data.	P. 41

Command		Explanation	Reference
	SetRUNPRGDATA	Set the remote program data.	P. 43

3.3.1.1. Command related to set value of equipment

It is a command that acquisition/sets the set value set to the equipment.

3.3.1.1.1. GetROM command

Acquire ROM version.

Item	Type explanation	
Return value	VT_BSTR	ROM type ROM version

Usage example

```
    GetROM command ' execution
   Dim val As String
   val = caoCon.Execute("GetROM")
```

3.3.1.1.2. GetROMDISP command

Acquire ROM version of the indicator.

Item	Type explanation	
Return value	VT_BSTR	ROM type ROM version

Usage example

```
    GetROMDISP ' command execution
   Dim val As String
   val = caoCon.Execute("GetROMDISP")
```

3.3.1.1.3. GetMASK command

Acquire the interrupt mask.

Item	Type explanation

Item	Type explanation	
Return value	VT_UI1	Interrupt mask It is addition as follows crack of bit (order of most significant bit) The first bit :Unused The second bit :When alarm is generated by device The third bit :When driving one step ends end in remote program driving The fourth bit :When changing into power-off/power supply on The fifth bit :Unused The sixth bit :Unused The seventh bit :Reserve it in the GPIB telecommunication facility. The eighth bit :Unused If the bit confronted is one when the event allocated to each bit occurs, interrupt is generated.

Usage example

```
 GetMASK '  command execution
Dim val As Byte
val = caoCon.Execute("GetMASK")
```

3.3.1.1.4. SetMASK command

Set the interrupt mask.

Item	Type explanation

Item	Type explanation	
Argument	VT_UI1	Specify the interrupt mask. It is addition as follows crack of bit (order of most significant bit) The first bit :Unused The second bit :When alarm is generated by device The third bit :When driving one step ends end in remote program driving The fourth bit :When changing into power-off/power supply on The fifth bit :Unused The sixth bit :Unused The seventh bit :Reserve it in the GPIB telecommunication facility. The eighth bit :Unused If the bit confronted is one when the event allocated to each bit occurs, interrupt is generated.
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
    SetMASK ’  command execution
  Dim result As String
  result = caoCon.Execute(”SetMASK”, 64)
```

3.3.1.1.5. GetTIMERMON command

Acquire the started timer.

Item	Type explanation	
Return value	VT_EMPTY	There is no started timer.
	Or,	
	VT_ARRAY | VT_I4	Timer numbers started up to three

<table>
<tr><th>Item</th><th colspan="3">Type explanation</th></tr>
<tr><td></td><td>i</td><td>VT_I4</td><td>Timer number
The timer number is as follows.
·0 :Quick timer
·1 :Beginning timer
·2 :Stop timer

<b>– Subscript i: Starting a few minutes of timer (0 – 2)</b></td></tr>
</table>

Usage example

```
    GetTIMERMON command '  execution
Dim values As Variant
values = caoCon.Execute("GetTIMERMON")

'  acquire only the number of started timers.
If Not IsEmpty(values) Then
    Dim i As Long
    For i = 0 To UBound(values)
        Dim val As Long
        val = values(i)
    Next i
End If
```

3.3.1.1.6. GetTIMERSET command

Acquire the set timer.

<table>
<tr><th>Item</th><th colspan="3">Type explanation</th></tr>
<tr><td rowspan="4">Return value</td><td colspan="2">VT_EMPTY</td><td>There is no set timer.</td></tr>
<tr><td colspan="3">Or,</td></tr>
<tr><td colspan="2">VT_ARRAY | VT_I4</td><td>Timer numbers set up to three</td></tr>
<tr><td>i</td><td>VT_I4</td><td>Timer number
The timer number is as follows.
·0 :Quick timer
·1 :Beginning timer
·2 :Stop timer

<b>– Subscript i : Set a few minutes of timer (0 – 2)</b></td></tr>
</table>

Usage example

```
    GetTIMERSET '  command execution
Dim values As Variant
values = caoCon.Execute("GetTIMERSET")

'  acquire only the number of set timers.
If Not IsEmpty(values) Then
    Dim i As Long
    For i = 0 To UBound(values)
        Dim val As Long
        val = values(i)
    Next i
End If
```

3.3.1.1.7. GetTIMERO command

Acquire the setting of a quick timer.

Item	Type explanation	
Return value	VT_BSTR	Mode of operation, set time or stop mode and set time The mode of operationTable3-3Refer to [wo]. The stop modeTable3-4Refer to [wo].

Table3-3Details of mode of operation

Set content	Response data display	Response example
Program driving	"RUN and RAM: Program number and STEPxx"	"RUN,RAM:1,STEP1"
Fixed value driving	"CONSTANT"	"CONSTANT"

The range of the program number becomes 1-8.

Table3-4Details of stop mode

Set content	Response data display	Response example
Halt condition	"STANDBY"	"STANDBY"
Panel power supply OFF	"OFF"	"OFF"

Usage example

```
    GetTIMERO command '  execution
Dim val As String
val = caoCon.Execute("GetTIMERO")
```

3.3.1.1.8. SetTIMER0 command

Set the setting of a quick timer.

Item	Type explanation	
Argument	VT_BSTR	Specify the stop mode at the mode of operation or uptime at uptime. The mode of operationTable3-3Refer to [wo]. The stop modeTable3-4Refer to [wo].
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
 SetTIMER0 command '  execution
Dim result As String
result = caoCon.Execute("SetTIMER0", "10:00,CONSTANT")
```

3.3.1.1.9. GetTIMER1 command

Acquire the setting of driving beginning timer.

Item	Type explanation	
Return value	VT_BSTR	Timer number, start mode, and mode of operation The start modeTable3-5Refer to [wo]. The mode of operationTable3-3Refer to [wo].

Table3-5Details of start mode

Set content	Response data display	Response example
Execution mode once	"MODE1, start day, and uptime"	"MODE1,12.03/04,10:00"
Execution mode every week	"MODE2, start a day of the week, and uptime" A start day of the weekTable3-6Refer to [wo].	"MODE2,SAT,23:00"
Execution mode of every day	"MODE3 and uptime"	"MODE3,0:00"

Table3-6Details on a start day of the week

A day of the week	Response data display
Monday	"MON"
Tuesday	"TUE"

Wednesday	″WED″
Thursday	″THU″
Friday	″FRI″
Saturday	″SAT″
Sunday	″SUN″

Two or more start day of the week can be specified.

Use″ ″/when you specify plurals.

Example:″MODE2,MON/SAT,10:00″

Usage example

```
 GetTIMER1 command '  execution
Dim val As String
val = caoCon.Execute(″GetTIMER1″)
```

3.3.1.1.10. SetTIMER1 command

Set the setting of driving beginning timer.

Item	Type explanation	
Argument	VT_BSTR	Specify the start mode and the mode of operation. The start modeTable3-5Refer to [wo]. The mode of operationTable3-3Refer to [wo].
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
 SetTIMER1 command '  execution
Dim result As String
result = caoCon.Execute(″SetTIMER1″, ″MODE1,12.03/04,10:00,CONSTANT″)
```

3.3.1.1.11. GetTIMER2 command

Acquire the setting of the driving stop timer.

Item	Type explanation	
Return value	VT_BSTR	Timer number, start mode, and stop mode The start modeTable3-5Refer to [wo]. The stop modeTable3-4Refer to [wo].

Usage example

```
 GetTIMER1 command '  execution
```

```
    Dim val As String
    val = caoCon.Execute("GetTIMER2")
```

3.3.1.1.12. SetTIMER2 command

Set the setting of the driving stop timer.

Item	Type explanation	
Argument	VT_BSTR	Specify the start mode and the stop mode. The start modeTable3-5Refer to [wo]. The stop modeTable3-4Refer to [wo].
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
     SetTIMER2 command '  execution
    Dim result As String
    result = caoCon.Execute("SetTIMER2", "MODE2,SAT,10:00,OFF")
```

3.3.1.1.13. GetKEYPROTECT command

Acquire the protecting setting.

Item	Type explanation	
Return value	VT_BSTR	State of key protecting Either the following is acquired. ・ ON: Key protecting ON ・ OFF: Key protecting OFF Key protecting ON is ..either the setting change protecting or the driving operation protecting.. in the state of turning on.

Usage example

```
     GetKEYPROTECT '  command execution
    Dim val As String
    val = caoCon.Execute("GetKEYPROTECT")
```

3.3.1.1.14. SetKEYPROTECT command

Set the protecting setting. The setting change protecting and the driving operation protecting cannot be separately set.

Item	Type explanation

Item	Type explanation	
Argument	VT_BSTR	Specify it in either of following of states of the key protecting. ・ ON: Key protecting ON ・ OFF: Key protecting OFF
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
  SetKEYPROTECT ’  command execution
Dim result As String
result = caoCon.Execute(″SetKEYPROTECT″, ″OFF″)
```

3.3.1.1.15. GetTYPE command

Acquire the chamber type.

Item	Type explanation		
Return value	VT_ARRAY | VT_VARIANT		
	0	VT_BSTR	[Kawakyuu] sensor type
	1	VT_BSTR	[Kyuu] sensor ..damp.. type - Because the acquisition data is omitted, only the temperature becomes VT_EMPTY for the device of the type.
	2	VT_BSTR	Attemperation machine type
	3	VT_R8	Upper temperature limit

Usage example

```
  GetTYPE ’  command execution
Dim values As Variant
values = caoCon.Execute(″GetTYPE″)

[Kawakyuu] sensor ’  type
Dim drySensor As String
drySensor = values(0)

[Kyuu] sensor ’  ..damp.. type
If Not IsEmpty(values(1)) Then
    Dim wetSensor As String
    wetSensor = values(1)
End If
```

```
    Attemperation machine ’  type
    Dim tempCon As String
    tempCon = values(2)

    ’  upper temperature limit
    Dim tempUplimit As Double
    tempUplimit = values(3)
```

3.3.1.1.16. GetTEMP command

Acquire the temperature parameter.

Item	Type explanation		
Return value	VT_ARRAY ｜ VT_R8		
	0	VT_R8	Measurements
	1	VT_R8	Set value
	2	VT_R8	Upper bound value
	3	VT_R8	Lower bound value

Usage example

```
     GetTEMP ’  command execution
    Dim values() As Double
    values = caoCon.Execute("GetTEMP")
```

3.3.1.1.17. SetTEMP command

Set the temperature parameter.

Item	Type explanation		
Argument	VT_ARRAY ｜ VT_R8		
	0	VT_R8	Specify a set value.
	1	VT_R8	Specify the upper bound value.
	2	VT_R8	Specify the lower bound value.
Return value	VT_BSTR		When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
     SetTEMP ’  command execution
    Dim result As String
    result = caoCon.Execute("SetTEMP", Array(23.0, 100.0, -40.0))
```

3.3.1.1.18. SetSTEMP command

Set a set value of the temperature parameter.

Item	Type explanation	
Argument	VT_R8	Specify a set value.
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
    SetSTEMP ’  command execution
  Dim result As String
  result = caoCon.Execute(”SetSTEMP”, 23.0)
```

3.3.1.1.19. SetHTEMP command

Set the upper bound value of the temperature parameter.

Item	Type explanation	
Argument	VT_R8	Specify the upper bound value.
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
    SetHTEMP ’  command execution
  Dim result As String
  result = caoCon.Execute(”SetHTEMP”, 100.0)
```

3.3.1.1.20. SetLTEMP command

Set the lower bound value of the temperature parameter.

Item	Type explanation	
Argument	VT_R8	Specify the lower bound value.
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
    SetLTEMP ’  command execution
  Dim result As String
  result = caoCon.Execute(”SetLTEMP”, -40.0)
```

3.3.1.1.21. GetHUMI command

Acquire the humidity parameter.

Item	Type explanation		
Return value	VT_ARRAY | VT_I4		
	0	VT_I4	Measurements
	1	VT_I4	Set value
	2	VT_I4	Upper bound value
	3	VT_I4	Lower bound value

Usage example

```
 GetHUMI ' command execution
Dim values() As Long
values = caoCon.Execute("GetHUMI")
```

3.3.1.1.22. SetHUMI command

Set the humidity parameter.

Item	Type explanation		
Argument	VT_ARRAY | VT_I4		
	0	VT_I4	Specify a set value.
	1	VT_I4	Specify the upper bound value.
	2	VT_I4	Specify the lower bound value.
Return value	VT_BSTR		When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
 SetHUMI ' command execution
Dim result As String
result = caoCon.Execute("SetHUMI", Array(23, 100, 0))
```

3.3.1.1.23. SetSHUMI command

Set a set value of the humidity parameter.

Item	Type explanation	
Argument	VT_I4	Specify a set value.
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
    SetSHUMI '  command execution
  Dim result As String
  result = caoCon.Execute("SetSHUMI", 85)
```

3.3.1.1.24. SetHHUMI command

Set the upper bound value of the humidity parameter.

Item	Type explanation	
Argument	VT_I4	Specify the upper bound value.
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
    SetHHUMI '  command execution
  Dim result As String
  result = caoCon.Execute("SetHHUMI", 100)
```

3.3.1.1.25. SetLHUMI command

Set the lower bound value of the humidity parameter.

Item	Type explanation	
Argument	VT_I4	Specify the lower bound value.
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
    SetLHUMI '  command execution
  Dim result As String
  result = caoCon.Execute("SetLHUMI", 0)
```

3.3.1.1.26. GetSET command

Acquire the control value of the freezer.

Item	Type explanation	
Return value	VT_BSTR	Control value Either the following is acquired. ・ REF0 ～ REF8 : Manual setting ・ REF9 : Automatic setting

Usage example

```
    GetSET '  command execution
   Dim val As String
   val = caoCon.Execute("GetSET")
```

3.3.1.1.27. SetSET command

Set the control value of the freezer.

Item	Type explanation	
Argument	VT_BSTR	Specify it in either of following of the control values. ・ REF0 : Manual operation OFF ・ REF1 : Manual operation ON ・ REF2 ～ REF9 : Automatic operation
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
    SetSET '  command execution
   Dim result As String
   result = caoCon.Execute("SetSET", "REF9")
```

3.3.1.1.28. GetCONSTANTSETTEMP command

Acquire the temperature setting value of the fixed command setting.

Item	Type explanation		
Return value	VT_ARRAY ｜ VT_VARIANT		
	0	VT_R8	Temperature setting value
	1	VT_BSTR	Control permission Either the following is acquired. ・ ON : The temperature limiting is effective. ・ OFF : Temperature limiting invalidity

Usage example

```
    GetCONSTANTSETTEMP '  command execution
   Dim values As Variant
   values = caoCon.Execute("GetCONSTANTSETTEMP")

   Temperature setting '  value
   Dim temp As Double
```

```
    temp = values(0)

    Control '  permission
    Dim permission As String
    permission = values(1)
```

3.3.1.1.29. GetCONSTANTSETHUMI command

Acquire the humidity setting value of the fixed command setting.

<table>
<tr><th>Item</th><th colspan="3">Type explanation</th></tr>
<tr><td rowspan="3">Return value</td><td colspan="2">VT_ARRAY | VT_VARIANT</td><td></td></tr>
<tr><td>0</td><td>VT_I4</td><td>Humidity setting value</td></tr>
<tr><td>1</td><td>VT_BSTR</td><td>Control permission
Either the following is acquired.
・ ON : The moisture control is effective.
・ OFF : Moisture control invalidity</td></tr>
</table>

Usage example

```
     GetCONSTANTSETHUMI '  command execution
    Dim values As Variant
    values = caoCon.Execute("GetCONSTANTSETHUMI")

    Humidity setting '  value
    Dim humi As Long
    humi = values(0)

    Control '  permission
    Dim permission As String
    permission = values(1)
```

3.3.1.1.30. GetCONSTANTSETREF command

Acquire the freezer setting value of the fixed command setting.

Item	Type explanation	
Return value	VT_BSTR	Set value Either the following is acquired. ・ AUTO : Automatic setting ・ OFF : Manual setting OFF ・ 100 : Manual setting ON

Usage example

```
     GetCONSTANTSETREF '  command execution
    Dim val As String
    val = caoCon.Execute(″GetCONSTANTSETREF″)
```

3.3.1.1.31. GetSYSTEMSETPTS command

Acquire the function of the monitor of the temperature of the sample.

Item	Type explanation	
Return value	VT_BSTR	Installing information Either the following is acquired. ・ MON ：Equipped with function of monitor of temperature of sample ・ OFF ：Equipped with monitor of temperature of sample

Usage example

```
     GetSYSTEMSETPTS '  command execution
    Dim val As String
    val = caoCon.Execute(″GetSYSTEMSETPTS″)
```

3.3.1.1.32. GetSYSTEMSETPTC command

Acquire the function of the control of the temperature of the sample.

Item	Type explanation	
Return value	VT_BSTR	Installing information Either the following is acquired. ・ ON ：Equipped with function of control of temperature of sample ・ OFF ：Equipped with control of temperature of sample

Usage example

```
     GetSYSTEMSETPTC '  command execution
    Dim val As String
    val = caoCon.Execute(″GetSYSTEMSETPTC″)
```

3.3.1.1.33. GetSYSTEMSETPTCOPT command

Acquire an optional function of the control of the temperature of the sample.

Item	Type explanation

Item	Type explanation	
Return value	VT_BSTR	Installing information Either the following is acquired. ・ OFF : Uninstalling of temperature of sample ・ M : Equipped with specimen temperature (Only the monitor :). ・ C : Equipped with specimen temperature (Only the control of the temperature of the sample :). ・ MC : Equipped with specimen temperature (control of temperature of monitor & sample)

Usage example

```
    GetSYSTEMSETPTCOPT ’  command execution
   Dim val As String
   val = caoCon.Execute(″GetSYSTEMSETPTCOPT″)
```

3.3.1.1.34. SetTIMERMERASE command

Delete the setting of the specified timer.

Item	Type explanation	
Argument	VT_I4	Specify the timer number to be deleted. The timer number is as follows. ・0 :Quick timer ・1 :Beginning timer ・2 :Stop timer
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
    SetTIMERMERASE ’  command execution
   Dim result As String
   result = caoCon.Execute(″SetTIMERMERASE″, 1)
```

3.3.1.1.35. SetTIMERON command

Start the specified timer.

Item	Type explanation

Item	Type explanation	
Argument	VT_I4	Specify the timer number that starts. The timer number is as follows. ·0 :Quick timer ·1 :Beginning timer ·2 :Stop timer
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
 SetTIMERON command '  execution
Dim result As String
result = caoCon.Execute("SetTIMERON", 1)
```

3.3.1.1.36. SetTIMEROFF command

Stop the specified timer.

Item	Type explanation	
Argument	VT_I4	Specify the timer number that stops. The timer number is as follows. ·0 :Quick timer ·1 :Beginning timer ·2 :Stop timer
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
 SetTIMEROFF command '  execution
Dim result As String
result = caoCon.Execute("SetTIMEROFF", 2)
```

3.3.1.2. Command related to state of equipment

It is a command that acquisition/sets the state of the equipment.

3.3.1.2.1. GetDATE command

Acquire the date of an internal calendar.

Item	Type explanation

Item	Type explanation	
Return value	VT_BSTR	Date YY.MM/DD

Usage example

```
 GetDATE ' command execution
Dim val As String
val = caoCon.Execute("GetDATE")
```

3.3.1.2.2. SetDATE command

Set the date of an internal calendar.

Item	Type explanation	
Argument	VT_BSTR	It appoints a date. YY.MM/DD
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
 SetDATE ' command execution
Dim result As String
result = caoCon.Execute("SetDATE", "12.03/04")
```

3.3.1.2.3. GetTIME command

Acquire the time of an internal calendar now.

Item	Type explanation	
Return value	VT_BSTR	Time now hh:mm:ss

Usage example

```
 GetTIME ' command execution
Dim val As String
val = caoCon.Execute("GetTIME")
```

3.3.1.2.4. SetTIME command

Set the time of an internal calendar now.

Item	Type explanation

Item	Type explanation	
Argument	VT_BSTR	Specify time now. hh:mm:ss
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
    SetTIME '  command execution
    Dim result As String
    result = caoCon.Execute("SetTIME", "18:00:00")
```

3.3.1.2.5. GetSRQ command

Acquire interrupt.

Item	Type explanation	
Return value	VT_UI1	State of interrupt It is addition as follows crack of bit (order of most significant bit) The first bit :Unused The second bit :When alarm is generated by device The third bit :When driving one step ends end in remote program driving The fourth bit :When changing into power-off/power supply on The fifth bit :Unused The sixth bit :Unused The seventh bit :Reserve it in the GPIB telecommunication facility. The eighth bit :Unused If the bit confronted is one when the event allocated to each bit occurs, interrupt is generated.

Usage example

```
    GetSRQ '  command execution
    Dim val As Byte
    val = caoCon.Execute("GetSRQ")
```

3.3.1.2.6. SetSRQ command

Reset interrupt.

<table>
<tr><th>Item</th><th colspan="2">Type explanation</th></tr>
<tr><td>Return value</td><td>VT_BSTR</td><td>When the command is normally executed, the response character string from the equipment is stored as it is.</td></tr>
</table>

Usage example

```
  SetSRQ '  command execution
Dim result As String
result = caoCon.Execute("SetSRQ")
```

3.3.1.2.7. GetALARM command

Acquire the alarm state.

<table>
<tr><th>Item</th><th colspan="3">Type explanation</th></tr>
<tr><td rowspan="4">Return value</td><td colspan="2">VT_EMPTY</td><td>There is no generated alarm.</td></tr>
<tr><td colspan="3">Or,</td></tr>
<tr><td colspan="2">VT_ARRAY | VT_I4</td><td>Alarm information up to 16</td></tr>
<tr><td>i</td><td>VT_I4</td><td>Alarm number
– <b>Subscript i ： Generated a few minutes of alarm (0 – 15)</b></td></tr>
</table>

Usage example

```
  GetALARM '  command execution
Dim values As Variant
values = caoCon.Execute("GetALARM")

'  acquire only the number of generated alarms.
If Not IsEmpty(values) Then
    Dim i As Long
    For i = 0 To UBound(values)
        Dim val As Long
        val = values(i)
    Next i
End If
```

3.3.1.2.8. GetMODE command

Acquire the mode of operation.

Item	Type explanation

Item	Type explanation	
Return value	VT_BSTR	Mode of operation Either the following is acquired. ・ OFF ：State of panel power supply OFF ・ STANDBY ：Halt condition ・ CONSTANT ：State of fixed value driving ・ RUN ：Program/state of remote driving

Usage example

```
 GetMODE ’  command execution
Dim val As String
val = caoCon.Execute("GetMODE")
```

3.3.1.2.9. SetMODE command

Set the mode of operation.

Item	Type explanation	
Argument	VT_BSTR	Specify it in either of following of the mode of operations. ・ OFF ：Panel power supply OFF ・ STANDBY ：It is stopping. ・ CONSTANT ：Fixed value driving (No1) ・ RUN ：Program driving
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
 SetMODE ’  command execution
Dim result As String
result = caoCon.Execute("SetMODE", "OFF")
```

3.3.1.2.10. GetMODEDETAIL command

Acquire a detailed mode of operation.

Item	Type explanation

Item	Type explanation	
Return value	VT_BSTR	Mode of operation Either the following is acquired. ・ OFF : State of panel power supply OFF ・ STANDBY : Halt condition ・ CONSTANT : State of fixed value driving ・ RUN : The program is being driven. ・ RUN PAUSE : It is stopping at one o'clock of the program. ・ RUN END HOLD : Program finality maintenance ・ RMT RUN : It is driving remotely. ・ RMT RUN PAUSE : It is stopping at one remote o'clock. ・ RMT RUN END HOLD : Remote, final maintenance

Usage example

```
 GetMODEDETAIL ' command execution
Dim val As String
val = caoCon.Execute("GetMODEDETAIL")
```

3.3.1.2.11. GetMONDETAIL command

Acquire the state in a detailed tank.

Item	Type explanation		
Return value	VT_ARRAY | VT_VARIANT		
	0	VT_R8	Measurement temperature
	1	VT_I4	Measurement humidity - Because the acquisition data is omitted, only the temperature becomes VT_EMPTY for the device of the type.
	2	VT_BSTR	Mode of operation Similar to the return value of the GetMODEDETAIL command.
	3	VT_I4	Number of alarms under generation

Usage example

```
 GetMONDETAIL ' command execution
Dim values As Variant
```

```
    values = caoCon.Execute("GetMONDETAIL")

    ’  measurement temperature
    Dim temp As Double
    temp = values(0)

    Measurement ’  humidity
    If Not IsEmpty(values(1)) Then
        Dim humi As Long
        humi = values(1)
    End If

    ’  mode of operation
    Dim mode As String
    mode = values(2)

    ’  number of alarms under generation
    Dim numAlarm As Long
    numAlarm = values(3)
```

3.3.1.2.12. GetHEATER command

Acquire the state of the heater.

Item	Type explanation		
Return value	VT_ARRAY | VT_VARIANT		
	0	VT_R8	Heating heater output
	1	VT_R8	Humidifying heater output - Because the acquisition data is omitted, only the temperature becomes VT_EMPTY for the device of the type.

Usage example

```
     GetHEATER ’  command execution
    Dim values As Variant
    values = caoCon.Execute("GetHEATER")

    Heating heater ’  output
    Dim heating As Double
    heating = values(0)

    Humidifying heater ’  output
    If Not IsEmpty(values(1)) Then
        Dim humidification As Double
        humidification = values(1)
    End If
```

3.3.1.3. Command related to program

It is a command that acquisition/sets a state and a set value of the program.

3.3.1.3.1. GetRUNPRGDATA command

Acquire the remote program data.

Item	Type explanation			
Return value	VT_ARRAY | VT_VARIANT			
	0		VT_R8	Onset temperature
	1		VT_R8	Attainment temperature
	2		VT_I4	Beginning humidity - Because the acquisition data is omitted, only the temperature becomes VT_EMPTY for the device of the type.
	3		VT_I4	Attainment humidity - Because the acquisition data is omitted, only the temperature becomes VT_EMPTY for the device of the type.
	4		VT_BSTR	Setting at time ″ (variable-length)
	5		VT_BSTR	Control value of freezer Refer to the return value of the GetSET command.
	6		VT_ARRAY | VT_VARIANT	Signal setting of time - It becomes VT_EMPTY because the acquisition data is omitted when there is no time turned on signal.
		6.0	VT_BSTR	Time state of signal Either the following is acquired. ・ ON : Time turned on signal ・ OFF : Time turned off signal
		6.i	VT_I4	Signal number of time **- Subscript i : Signal a few minutes at time (1 ? 11)**

Usage example

```
 GetRUNPRGDATA '  command execution
Dim values As Variant
values = caoCon.Execute("GetRUNPRGDATA")

'  onset temperature
Dim temp As Double
temp = values(0)

Attainment '  temperature
Dim goTemp As Double
goTemp = values(1)

Beginning '  humidity
If Not IsEmpty(values(2)) Then
    Dim humi As Long
    humi = values(2)
End If

Attainment '  humidity
If Not IsEmpty(values(3)) Then
    Dim goHumi As Long
    goHumi = values(3)
End If

'  setting at time
Dim time As String
time = values(4)

Control '  value of freezer
Dim ref As String
ref = values(5)

Signal '  setting of time
Dim relays As Variant
relays= values(6)

If Not IsEmpty(relays) Then
    Time '  state of signal
    Dim relayStatus As String
    relayStatus = relays(0)

    Signal '  number of time

    Dim i As Long
    For i = 1 To UBound(relays)
        Dim relayNo As Long
        relayNo = relays(i)
    Next i
End If
```

3.3.1.3.2. SetRUNPRGDATA command

Set the remote program data.

Item	Type explanation			
Argument	VT_ARRAY | VT_VARIANT			
	0		VT_R8	Specify the onset temperature.
	1		VT_R8	Specify the attainment temperature. - The unsetting is possible when making it to VT_EMPTY.
	2		VT_I4	Specify the beginning humidity. - The unsetting is possible when making it to VT_EMPTY.
	3		VT_I4	Specify the attainment humidity. - The unsetting is possible when making it to VT_EMPTY.
	4		VT_BSTR	Specify the time setting as follows. ″ (variable-length)
	5		VT_BSTR	Specify the control value of the freezer. Please refer to the argument of SetSET command. - The unsetting is possible when making it to VT_EMPTY.
	6		VT_ARRAY | VT_VARIANT	Specify the signal setting of time. - The unsetting is possible when making it to VT_EMPTY.
		6.0	VT_BSTR	Specify it in either of following of the time states of the signal. ･ON　: Time turned on signal ･OFF　: Time turned off signal
		6.i	VT_I4	Specify the signal number of time. **- Subscript i　: Signal a few minutes at time (1 ? 11)**

Item	Type explanation	
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
 SetRUNPRGDATA ’  command execution
Dim result As String

result = caoCon.Execute("SetRUNPRGDATA", Array(10.0, 23.0, 85, 100, "1:00", _
                                                 Empty, Empty))
```

3.3.2. Enhancing command of CaoExtension class

The enhancing command of the CaoExtension class is effective only when connected by serial communications. Each command is detailed and the usage example is described.

Table3-7List of enhancing command of CaoExtension class

Command	Explanation	Reference
Command related to set value of equipment		
GetROM	Acquire ROM version.	P. 45
GetMASK	Acquire the interrupt mask.	P. 45
SetMASK	Set the interrupt mask.	P. 45
GetKEYLOCK	Acquire the state of the key lock.	P. 45
SetKEYLOCK	Set the state of the key lock.	P. 46
GetTYPE	Acquire the chamber type.	P. 46
GetTEMP	Acquire the temperature parameter.	P. 46
SetTEMP	Set the temperature parameter.	P. 47
SetSTEMP	Set a set value of the temperature parameter.	P. 47
SetHTEMP	Set the upper bound value of the temperature parameter.	P. 47
SetLTEMP	Set the lower bound value of the temperature parameter.	P. 47
GetHUMI	Acquire the humidity parameter.	P. 47
SetHUMI	Set the humidity parameter.	P. 47
SetSHUMI	Set a set value of the humidity parameter.	P. 47
SetHHUMI	Set the upper bound value of the humidity	P. 47

Command	Explanation	Reference
	parameter.	
SetLHUMI	Set the lower bound value of the humidity parameter.	P. 47
GetSET	Acquire the control value of the freezer.	P. 48
SetSET	Set the control value of the freezer.	P. 48
Command related to state of equipment		
GetSRQ	Acquire interrupt.	P. 49
SetSRQ	Reset interrupt.	P. 49
GetALARM	Acquire the alarm state.	P. 49
GetMODE	Acquire the mode of operation.	P. 49
SetMODE	Set the mode of operation.	P. 49
GetMON	Acquire the state in the tank.	P. 50
GetHEATER	Acquire the state of the heater.	P. 51
SetPOWER	Set the state of the power supply.	P. 51
Command related to program		
GetRUNPRGDATA	Acquire the remote program data.	P. 52
SetRUNPRGDATA	Set the remote program data.	P. 52

3.3.2.1. Command related to set value of equipment

It is a command that acquisition/sets the set value set to the equipment.

3.3.2.1.1. GetROM command

Acquire ROM version. Please refer to "GetROM" command of CaoController class for details.

3.3.2.1.2. GetMASK command

Acquire the interrupt mask. Please refer to "GetMASK" command of CaoController class for details.

3.3.2.1.3. SetMASK command

Set the interrupt mask. Please refer to "SetMASK" command of CaoController class for details.

3.3.2.1.4. GetKEYLOCK command

Acquire the state of the key lock.

Item	Type explanation

Item	Type explanation	
Return value	VT_BSTR	State of key lock Either the following is acquired. ・ ON : State of key lock ON ・ OFF : State of key lock OFF

Usage example

```
    '  addition of extension board
    Dim caoExt As CaoExtension
    Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
    ' GetKEYLOCK command execution
    Dim val As String
    val = caoExt.Execute("GetKEYLOCK")
```

3.3.2.1.5. SetKEYLOCK command

Set the state of the key lock.

Item	Type explanation	
Argument	VT_BSTR	Specify it in either of following of states of the key lock. ・ ON : Key lock ON ・ OFF : Key lock OFF
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
    '  addition of extension board
    Dim caoExt As CaoExtension
    Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
     SetKEYLOCK '  command execution
    Dim result As String
    result = caoExt.Execute("SetKEYLOCK", "OFF")
```

3.3.2.1.6. GetTYPE command

Acquire the chamber type. Details are "of the CaoController class. Please refer to "GetTYPE" command of CaoController class for details.

3.3.2.1.7. GetTEMP command

Acquire the temperature parameter. Details are "of the CaoController class. Please refer to "GetTEMP" command of CaoController class for details.

3.3.2.1.8. SetTEMP command

Set the temperature parameter. Details are ″of the CaoController class. Please refer to ″SetTEMP″ command of CaoController class for details.

3.3.2.1.9. SetSTEMP command

Set a set value of the temperature parameter. Please refer to ″SetSTEMP″ command of CaoController class for details.

3.3.2.1.10. SetHTEMP command

Set the upper bound value of the temperature parameter. Please refer to ″SetHTEMP″ command of CaoController class for details.

3.3.2.1.11. SetLTEMP command

Set the lower bound value of the temperature parameter. Please refer to ″SetLTEMP″ command of CaoController class for details.

3.3.2.1.12. GetHUMI command

Acquire the humidity parameter. Please refer to ″GetHUMI″ command of CaoController class for details.

3.3.2.1.13. SetHUMI command

Set the humidity parameter. Please refer to ″SetHUMI″ command of CaoController class for details.

3.3.2.1.14. SetSHUMI command

Set a set value of the humidity parameter. Please refer to ″SetSHUMI″ command of CaoController class for details.

3.3.2.1.15. SetHHUMI command

Set the upper bound value of the humidity parameter. Please refer to ″SetHHUMI″ command of CaoController class for details.

3.3.2.1.16. SetLHUMI command

Set the lower bound value of the humidity parameter. Please refer to ″SetLHUMI″ command of CaoController class for details.

3.3.2.1.17. GetSET command

Acquire the control value of the freezer.

Item	Type explanation	
Return value	VT_BSTR	Control value Either the following is acquired. ・ REF0 : Manual setting (stop) ・ REF1 : Manual setting (driving) ・ REF9 : Automatic setting

Usage example

```
    '  addition of extension board
    Dim caoExt As CaoExtension
    Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
     GetSET '  command execution
    Dim val As String
    val = caoExt.Execute("GetSET")
```

3.3.2.1.18. SetSET command

Set the control value of the freezer.

Item	Type explanation	
Argument	VT_BSTR	Specify it in either of following of the control values. ・ REF0 : Manual operation (deactivate request) ・ REF1 : Manual operation (driving demand) ・ REF9 : Automatic operation
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
    '  addition of extension board
    Dim caoExt As CaoExtension
    Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
     SetSET '  command execution
    Dim result As String
    result = caoExt.Execute("SetSET", "REF9")
```

3.3.2.2. Command related to state of equipment

It is a command that acquisition/sets the state of the equipment.

3.3.2.2.1. GetSRQ command

Acquire interrupt. Details are "of the CaoController class. Please refer to "GetSRQ" command of CaoController class for details.

3.3.2.2.2. SetSRQ command

Reset interrupt. Details are "of the CaoController class. Please refer to "SetSRQ" command of CaoController class for details.

3.3.2.2.3. GetALARM command

Acquire the alarm state. Details are "of the CaoController class. Please refer to "GetALARM" command of CaoController class for details.

3.3.2.2.4. GetMODE command

Acquire the mode of operation.

Item	Type explanation	
Return value	VT_BSTR	Mode of operation Either the following is acquired. ・ OFF : State of panel power supply OFF ・ STANDBY : State of standby ・ CONSTANT : State of fixed value driving ・ RUN : Program/state of remote driving

Usage example

```
    '  addition of extension board
    Dim caoExt As CaoExtension
    Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
     GetMODE '  command execution
    Dim val As String
    val = caoExt.Execute("GetMODE")
```

3.3.2.2.5. SetMODE command

Set the mode of operation.

Item	Type explanation

Item	Type explanation	
Argument	VT_BSTR	Specify it in either of following of the mode of operations. ・ OFF : Make it to panel power supply OFF. ・ STANDBY : Put it into the state of the standby. ・ CONSTANT : Drive a fixed value. ・ RUN program number : Do the program driving.
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
'  addition of extension board
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
 SetMODE '  command execution
Dim result As String
result = caoExt.Execute("SetMODE", "STANDBY")
```

3.3.2.2.6. GetMON command

Acquire the state in the tank.

Item	Type explanation		
Return value	VT_ARRAY | VT_VARIANT		
	0	VT_R8	Measurement temperature
	1	VT_I4	Measurement humidity - Because the acquisition data is omitted, only the temperature becomes VT_EMPTY for the device of the type.
	2	VT_BSTR	Mode of operation Please refer to GetMODE command.
	3	VT_I4	Number of alarms under generation

Usage example

```
'  addition of extension board
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
 GetMON command '  execution
Dim values As Variant
values = caoExt.Execute("GetMON")
```

```
    ’  measurement temperature
    Dim temp As Double
    temp = values(0)

    Measurement ’  humidity
    If Not IsEmpty(values(1)) Then
        Dim humi As Long
        humi = values(1)
    End If

    ’  mode of operation
    Dim mode As String
    mode = values(2)

    ’  number of alarms under generation
    Dim numAlarm As Long
    numAlarm = values(3)
```

3.3.2.2.7. GetHEATER command

Acquire the state of the heater. Please refer to ″GetHEATER″ command of CaoController class for details.

3.3.2.2.8. SetPOWER command

Set the state of the power supply.

Item	Type explanation	
Argument	VT_BSTR	Specify it in either of following of states of the power supply. ・ ON ：Turn on the panel power supply, and make it to the fixed value driving. ・ OFF ：Turn off the panel power supply.
Return value	VT_BSTR	When the command is normally executed, the response character string from the equipment is stored as it is.

Usage example

```
    ’  addition of extension board
    Dim caoExt As CaoExtension
    Set caoExt = caoCon.AddExtension(″Extension″, ″Addr = 1″)
     SetPOWER ’  command execution
    Dim result As String
    result = caoExt.Execute(″SetPOWER″, ″OFF″)
```

3.3.2.3. Command related to program

It is a command that acquisition/sets a state and a set value of the program.

3.3.2.3.1. GetRUNPRGDATA command

Acquire the remote program data. Please refer to "GetRUNPRGDATA" command of CaoController class for details.

3.3.2.3.2. SetRUNPRGDATA command

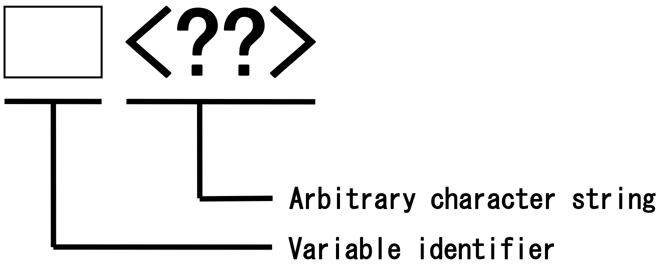
Set the remote program data. Please refer to "SetRUNPRGDATA" command of CaoController class for details.

3.4. Variable list

Define the variable list that can be used in each class. The variable indicates the object of the CaoVariable class. To register two or more variables (usefulness to change only the option), an arbitrary character string can be given.

The format to give the variable identifier an arbitrary character string is shown below.

Two or more variable commonness and specification formats



3.4.1. Variable of CaoController class

Only when Ethernet is connected, variables except "@MAKER_NAME" and "@VERSION" of the CaoController class are effective variables.

Table3-8List of variable of CaoController class

Variable identifier	Explanation	Value		Reference
		get	put	
Variable related to provider information				
@MAKER_NAME	Acquire the manufacturer name.	－	－	P.55

Variable identifier		Explanation	Value		Reference
			get	put	
	@VERSION	Acquire the provider version.	−	−	P.55
	@LASTDEVERR	Acquire the last device error.	−	-	P.55
Variable related to set value of equipment					
	@ROM	Acquire ROM version of [nurutotonoutsuwa].	−	-	P.56
	@ROMDISP	Acquire ROM version of the indicator.	−	-	P.56
	@MASK	Acquisition/set the interrupt mask.	−	−	P.57
	@TIMERMON	Acquire the started timer.	−	−	P.57
	@TIMERSET	Acquire the set timer.	−	−	P.57
	@TIMER0	Acquisition/set the setting of a quick timer.	−	−	P.58
	@TIMER1	Acquisition/set the setting of driving beginning timer.	−	−	P.58
	@TIMER2	Acquisition/set the setting of the driving stop timer.	−	−	P.59
	@TIMERERASE	Delete the setting of the specified timer.	−	−	P.59
	@TIMERON	Start the specified timer.	−	−	P.60
	@TIMEROFF	Stop the specified timer.	−	−	P.60
	@KEYPROTECT	Acquisition/set the protecting setting.	−	−	P.60
	@TYPE	Acquire the chamber type.	−	−	P.61
	@TEMPMON	Acquire the temperature parameter.	−	−	P.61
	@TEMP	Set the temperature parameter.	−	−	P.61
	@STEMP	Set the temperature setting value.	−	−	P.62
	@HTEMP	Set the temperature upper bound value.	−	−	P.62
	@LTEMP	Set the temperature lower bound value.	−	−	P.62
	@HUMIMON	Acquire the humidity parameter.	−	−	P.63
	@HUMI	Set the humidity parameter.	−	−	P.63
	@SHUMI	Set the humidity setting value.	−	−	P.63
	@HHUMI	Set the humidity upper bound value.	−	−	P.64
	@LHUMI	Set the humidity lower bound value.	−	−	P.64

Variable identifier	Explanation	Value		Reference
		get	put	
@SET	Acquisition/set the control value of the freezer.	–	–	P.64
@CONSTANTSETTEMP	Acquire the temperature setting value of the fixed command setting.	–	–	P.65
@CONSTANTSETHUMI	Acquire the humidity setting value of the fixed command setting.	–	–	P.65
@CONSTANTSETREF	Acquire the freezer setting value of the fixed command setting.	–	–	P.65
@SYSTEMSETPTS	Acquire the function of the monitor of the temperature of the sample.	–	–	P.66
@SYSTEMSETPTC	Acquire the function of the control of the temperature of the sample.	–	–	P.66
@SYSTEMSETPTCOPT	Acquire an optional function of the control of the temperature of the sample.	–	–	P.66
Variable related to state of equipment				
@DATE	Acquisition/set the date of an internal calendar.	–	–	P.67
@TIME	Acquisition/set the time of an internal calendar now.	–	–	P.67
@SRQ	Acquire interrupt.	–	–	P.67
@ALARM	Acquire the alarm state.	–	–	P.68
@MODE	Acquisition/set the mode of operation.	–	–	P.68
@MODEDETAIL	Acquire a detailed mode of operation.	–	–	P.69
@MONDETAIL	Acquire the state in a detailed tank.	–	–	P.69
@HEATER	Acquire the state of the heater.	–	–	P.69
Variable related to program				
@RUNPRGDATA	Acquisition/set the remote program data.	–	–	P.70

3.4.1.1. Variable related to provider information

It is a variable to acquire information that this provider is maintaining.

3.4.1.1.1. @MAKER_NAME

Acquire the manufacturer name.

Data type

Item	Type explanation	
Acquisition	VT_BSTR	Manufacturer name

Usage example

```
Variable ’ addition
Dim var As CaoVariable
Set var = caoCon.AddVariable(”@MAKER_NAME”)
Value ’ acquisition
Dim val As String
val = var.value
```

3.4.1.1.2. @VERSION

Acquire the provider version.

Data type

Item	Type explanation	
Acquisition	VT_BSTR	Provider version *.*.*

Usage example

```
Variable ’ addition
Dim var As CaoVariable
Set var = caoCon.AddVariable(”@VERSION”)
Value ’ acquisition
Dim val As String
val = var.value
```

3.4.1.1.3. @LASTDEVERR

Acquire the device error that occurred at the end.

Data type

Item	Type explanation		
Acquisition	VT_ARRAY | VT_VARIANT		
	0	VT_BSTR	Error message
	1	VT_I4	Error code

Usage example

```
    Variable '  addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@LASTDEVERR")
    Value '  acquisition
    Dim values As Variant
    values = var.value

    '  error message
    Dim errMsg As String
    errMsg = values(0)

    '  error code
    Dim errCode As Long
    errCode = values(1)
```

3.4.1.2. Variable related to set value of equipment

It is a variable that acquisition/sets the set value set to the equipment.

3.4.1.2.1. @ROM

Acquire ROM version.

Data type

Please refer to "GetROM" command of CaoController class for details.

Usage example

```
    Variable '  addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@ROM")
    Value '  acquisition
    Dim val As String
    val = var.value
```

3.4.1.2.2. @ROMDISP

Acquire ROM version of the indicator.

Data type

Please refer to "GetROMDISP" command of CaoController class for details.

Usage example

```
    Variable '  addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@ROMDISP")
    Value '  acquisition
    Dim val As String
    val = var.value
```

3.4.1.2.3. @MASK

Acquisition/set the interrupt mask. Please refer to "SetMASK" and "GEtMASK" command of CaoController class for details.

Data type

Item	Type explanation	
Acquisition/setting	VT_UI1	Interrupt mask

Usage example

```
Variable ' addition
Dim var As CaoVariable
Set var = caoCon.AddVariable("@MASK")
Value ' acquisition
Dim val As Byte
val = var.value
Value ' setting
var.value = 64
```

3.4.1.2.4. @TIMERMON

Acquire the started timer.

Data type

Please refer to "GetTIMERMON" command of CaoController class for details.

Usage example

```
Variable ' addition
Dim var As CaoVariable
Set var = caoCon.AddVariable("@TIMERMON")
Value ' acquisition
Dim values As Variant
values = var.value

' acquisition for started a few minutes of timer
If Not IsEmpty(values) Then
    Dim i As Long
    For i = 0 To UBound(values)
            Dim timerNo As Long
            timerNo = values(i)
    Next i
End If
```

3.4.1.2.5. @TIMERSET

Acquire the set timer.

Data type

Please refer to "GetTIMERSET" command of CaoController class for details.

Usage example

```
    Variable '  addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@TIMERSET")
    Value '  acquisition
    Dim values As Variant
    values = var.value

    '  acquisition for set a few minutes of timer
    If Not IsEmpty(values) Then
        Dim i As Long
        For i = 0 To UBound(values)
                Dim timerNo As Long
                timerNo = values(i)
        Next i
    End If
```

3.4.1.2.6. @TIMER0

Acquisition/set the setting of a quick timer. Please refer to "GetTIMER0" and "SetTIMER0" command of CaoController class for details.

Data type

Item	Type explanation	
Acquisition	VT_BSTR	Mode of operation, set time or stop mode and set time
Setting	VT_BSTR	Stop mode at mode of operation or uptime at uptime

Usage example

```
    Variable '  addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@TIMER0")
    Value '  acquisition
    Dim val As String
    val = var.value
    Value '  setting
    var.value = "10:00,CONSTANT"
```

3.4.1.2.7. @TIMER1

Acquisition/set the setting of driving beginning timer. Please refer to "GetTIMER1" command of CaoController class for details.

Data type

Item	Type explanation	
Acquisition	VT_BSTR	Timer number, start mode, and mode of operation
Setting	VT_BSTR	Start mode and mode of operation

Usage example

```
    Variable ’  addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable(″@TIMER1″)
    Value ’  acquisition
    Dim val As String
    val = var.value
    Value ’  setting
    var.value = ″ MODE1,12.03/04,10:00,CONSTANT″
```

3.4.1.2.8. @TIMER2

Acquisition/set the setting of the driving stop timer. Please refer to ″GetTIMER2″ and ″SetTIMER2″ command of CaoController class for details.

Data type

Item	Type explanation	
Acquisition	VT_BSTR	Timer number, start mode, and stop mode
Setting	VT_BSTR	Start mode and stop mode

Usage example

```
    Variable ’  addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable(″@TIMER2″)
    Value ’  acquisition
    Dim val As String
    val = var.value
    Value ’  setting
    var.value = ″MODE2,SAT,10:00,OFF″
```

3.4.1.2.9. @TIMERERASE

Delete the setting of the specified timer.

Data type

Please refer to ″SetTIMERMERASE″ command of CaoController class for details.

Usage example

```
    Variable '  addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@TIMERERASE")
    Value '  setting
    var.value = 1
```

3.4.1.2.10. @TIMERON

Start the specified timer.

Data type

Please refer to "SetTIMERON" command of CaoController class for details.

Usage example

```
    Variable '  addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@TIMERON")
    Value '  setting
    var.value = 1
```

3.4.1.2.11. @TIMEROFF

Stop the specified timer.

Data type

Please refer to "SetTIMEROFF" command of CaoController class for details.

Usage example

```
    Variable '  addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@TIMEROFF")
    Value '  setting
    var.value = 2
```

3.4.1.2.12. @KEYPROTECT

Acquisition/set the state of the key protecting. Please refer to "GetKEYPROTECT" and "SetKEYPROTECT" command of CaoController class for details.

Data type

Item	Type explanation	
Acquisition/setting	VT_BSTR	State of key protecting

Usage example

```
    Variable ' addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@KEYPROTECT")
    Value ' acquisition
    Dim val As String
    val = var.value
    Value ' setting
    var.value = "OFF"
```

3.4.1.2.13. @TYPE

Acquire the chamber type.

Data type

"of the CaoController class of the acquired data. Please refer to "GetTYPE" command of CaoController class for details.

Usage example

```
    Variable ' addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@TYPE")
    Value ' acquisition
    Dim values As Variant
    values = var.value
```

3.4.1.2.14. @TEMPMON

Acquire the temperature parameter.

Data type

Please refer to "GetTEMP" command of CaoController class for details.

Usage example

```
    Variable ' addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@TEMPMON")
    Value ' acquisition
    Dim values() As Double
    values = var.value
```

3.4.1.2.15. @TEMP

Set the temperature parameter.

Data type

Please refer to "SetTEMP" command of CaoController class for details.

Usage example

```
    Variable ’  addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable(”@TEMP”)
    Value ’  setting
    var.value = Array(23.0, 100.0, -40.0)
```

3.4.1.2.16. @STEMP

Set the temperature setting value.

Data type

Please refer to ”SetSTEMP” command of CaoController class for details.

Usage example

```
    Variable ’  addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable(”@STEMP”)
    Value ’  setting
    var.value = 23.0
```

3.4.1.2.17. @HTEMP

Set the temperature upper bound value.

Data type

Please refer to ”SetHTEMP” command of CaoController class for details.

Usage example

```
    Variable ’  addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable(”@HTEMP”)
    Value ’  setting
    var.value = 100.0
```

3.4.1.2.18. @LTEMP

Set the temperature lower bound value.

Data type

Please refer to ”SetLTEMP” command of CaoController class for details.

Usage example

```
    Variable ’  addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable(”@LTEMP”)
    Value ’  setting
    var.value = -40.0
```

3.4.1.2.19. @HUMIMON

Acquire the humidity parameter.

Data type

Please refer to "GetHUMI" command of CaoController class for details.

Usage example

```
    Variable ' addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@HUMIMON")
    Value ' acquisition
    Dim values() As Long
    values = var.value
```

3.4.1.2.20. @HUMI

Set the humidity parameter.

Data type

Please refer to "SetHUMI" command of CaoController class for details.

Usage example

```
    Variable ' addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@HUMI")
    Value ' setting
    var.value = Array(23, 100, 0)
```

3.4.1.2.21. @SHUMI

Set the humidity setting value.

Data type

Please refer to "SetSHUMI" command of CaoController class for details.

Usage example

```
    Variable ' addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@SHUMI")
    Value ' setting
    var.value = 85
```

3.4.1.2.22. @HHUMI

Set the humidity upper bound value.

Data type

Please refer to "SetHHUMI" command of CaoController class for details.

Usage example

```
    Variable ' addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@HHUMI")
    Value ' setting
    var.value = 100
```

3.4.1.2.23. @LHUMI

Set the humidity lower bound value.

Data type

Please refer to "SetLHUMI" command of CaoController class for details.

Usage example

```
    Variable ' addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@LHUMI")
    Value ' setting
    var.value = 0
```

3.4.1.2.24. @SET

Acquisition/set the control value of the freezer. Please refer to "GetSET" and "SetSET" command of CaoController class for details.

Data type

Item	Type explanation	
Acquisition/setting	VT_BSTR	Control value

Usage example

```
    Variable ' addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@SET")
    Value ' acquisition
    Dim val As String
    val = var.value
    Value ' setting
    var.value = "REF9"
```

3.4.1.2.25. @CONSTANTSETTEMP

Acquire the temperature setting value of the fixed command setting.

Data type

Please refer to ″GetCONSTANTSETTEMP″ command of CaoController class for details.

Usage example

```
    Variable ’  addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable(″@CONSTANTSETTEMP″)
    Value ’  acquisition
    Dim values As Variant
    values = var.value
```

3.4.1.2.26. @CONSTANTSETHUMI

Acquire the humidity setting value of the fixed command setting.

Data type

Please refer to ″GetCONSTANTSETHUMI″ command of CaoController class for details.

Usage example

```
    Variable ’  addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable(″@CONSTANTSETHUMI″)
    Value ’  acquisition
    Dim values As Variant
    values = var.value
```

3.4.1.2.27. @CONSTANTSETREF

Acquire the freezer setting value of the fixed command setting.

Data type

Please refer to ″GetCONSTANTSETREF″ command of CaoController class for details.

Usage example

```
    Variable ’  addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable(″@CONSTANTSETREF″)
    Value ’  acquisition
    Dim val As String
    val = var.value
```

3.4.1.2.28. @SYSTEMSETPTS

Acquire the function of the monitor of the temperature of the sample.

Data type

Please refer to "GetSYSTEMSETPTS" command of CaoController class for details.

Usage example

```
    ' Variable addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@SYSTEMSETPTS")
    ' Value acquisition
    Dim val As String
    val = var.value
```

3.4.1.2.29. @SYSTEMSETPTC

Acquire the function of the control of the temperature of the sample.

Data type

Please refer to "GetSYSTEMSETPTC" command of CaoController class for details.

Usage example

```
    ' Variable addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@SYSTEMSETPTC")
    ' Value acquisition
    Dim val As String
    val = var.value
```

3.4.1.2.30. @SYSTEMSETPTCOPT

Acquire an optional function of the control of the temperature of the sample.

Data type

Please refer to "GetSYSTEMSETPTCOPT" command of CaoController class for details.

Usage example

```
    ' Variable addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@SYSTEMSETPTCOPT")
    ' Value acquisition
    Dim val As String
    val = var.value
```

3.4.1.3. Variable related to state of equipment

It is a variable that acquisition/sets the state of the equipment.

3.4.1.3.1. @DATE

Acquisition/set the date of an internal calendar.

Data type

Item	Type explanation	
Acquisition/setting	VT_BSTR	Date YY.MM/DD

Usage example

```
    Variable ' addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@DATE")
    Value ' acquisition
    Dim val As String
    val = var.value
    Value ' setting
    var.value = "19.10/03"
```

3.4.1.3.2. @TIME

Acquisition/set the time of an internal calendar now.

Data type

Item	Type explanation	
Acquisition/setting	VT_BSTR	Time now hh:mm:ss

Usage example

```
    Variable ' addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@TIME")
    Value ' acquisition
    Dim val As String
    val = var.value
    Value ' setting
    var.value = "18:00:00
```

3.4.1.3.3. @SRQ

Acquire interrupt.

Data type

Please refer to "GetSRQ" command of CaoController class for details.

Usage example

```
    Variable ' addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@SRQ")
    Value ' acquisition
    Dim val As Byte
    val = var.value
```

3.4.1.3.4. @ALARM

Acquire the alarm state.

Data type

Please refer to "GetALARM" command of CaoController class for details.

Usage example

```
    Variable ' addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable("@ALARM")
    Value ' acquisition
    Dim values As Variant
    values = var.value

    If Not IsEmpty(values) Then
        Dim i As Long
        For i = 0 To UBound(values)
            Dim val As Long
            val = values(i)
        Next i
    End If
```

3.4.1.3.5. @MODE

Acquisition/set the mode of operation. Please refer to "GetMODE" and "SetMODE" command of CaoController class for details.

Data type

Item	Type explanation	
Acquisition/setting	VT_BSTR	Mode of operation

Usage example

```
    Variable ' addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable(″@MODE″)
    Value ’  acquisition
    Dim val As String
    val = var.value
    Value ’  setting
    var.value = ″OFF″
```

3.4.1.3.6. @MODEDETAIL

Acquire a detailed mode of operation.

Data type

Please refer to ″GetMODEDETAIL″ command of CaoController class for details.

Usage example

```
    Variable ’  addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable(″@MODEDETAIL″)
    Value ’  acquisition
    Dim val As String
    val = var.value
```

3.4.1.3.7. @MONDETAIL

Acquire the state in a detailed tank.

Data type

Please refer to ″GetMONDETAIL″ command of CaoController class for details.

Usage example

```
    Variable ’  addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable(″@MONDETAIL″)
    Value ’  acquisition
    Dim values As Variant
    values = var.value
```

3.4.1.3.8. @HEATER

Acquire the state of the heater.

Data type

Please refer to ″GetHEATER″ command of CaoController class for details.

Usage example

```
    Variable ’  addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable(″@HEATER″)
    Value ’  acquisition
    Dim values As Variant
    values = var.value
```

3.4.1.4. Variable related to program

It is a variable that acquisition/sets a state and a set value of the program.

3.4.1.4.1. @RUNPRGDATA

Acquisition/set the remote program data. Please refer to ″GetRUNPRGDATA″ and ″SetRUNPRGDATA″ command of CaoController class for details.

Data type

Item	Type explanation			
Acquisition/setting	VT_ARRAY | VT_VARIANT			
	0	VT_R8		Onset temperature
	1	VT_R8		Attainment temperature
	2	VT_I4		Beginning humidity
	3	VT_I4		Attainment humidity
	4	VT_BSTR		Setting at time
	5	VT_BSTR		Control value of freezer
	6	VT_ARRAY | VT_VARIANT		Signal setting of time
		6.0	VT_BSTR	Time state of signal
		6.i	VT_I4	Signal number of time

Usage example

```
    The controller’s ’  addition
    Dim caoCon As CaoController
    Set caoCon = caoWor.AddController(″Controller″, ″CaoProv.espec.TempHumi″, ″″, _
                                     conn=eth:192.168.0.2″)
    Variable ’  addition
    Dim var As CaoVariable
    Set var = caoCon.AddVariable(″@RUNPRGDATA″)
    Value ’  acquisition
    Dim values As Variant
    values = var.value
    Value ’  setting
    var.value = Array(10.0, 23.0, 85, 100, ″1:00″)
```

3.4.2. Variable of CaoExtension class

The variable of the CaoExtension class is effective only when connected by serial communications.

Table3-9List of variable of CaoExtension class

Variable identifier	Explanation	Value		Reference
		get	put	
Variable related to provider information				
@LASTDEVERR	Acquire the last device error.	－	-	P.72
Variable related to set value of equipment				
@ROM	Acquire ROM version.	－	－	P.72
@MASK	Acquisition/set the interrupt mask.	－	－	P.72
@KEYLOCK	Acquisition/set the state of the key lock.	－	－	P.72
@TYPE	Acquire the chamber type.	－	－	P.73
@TEMPMON	Acquire the temperature parameter.	－	－	P.73
@TEMP	Set the temperature parameter.	－	－	P.73
@STEMP	Set the temperature setting value.	－	－	P.73
@HTEMP	Set the temperature upper bound value.	－	－	P.73
@LTEMP	Set the temperature lower bound value.	－	－	P.73
@HUMIMON	Acquire the humidity parameter.	－	－	P.73
@HUMI	Set the humidity parameter.	－	－	P.73
@SHUMI	Set the humidity setting value.	－	－	P.73
@HHUMI	Set the humidity upper bound value.	－	－	P.73
@LHUMI	Set the humidity lower bound value.	－	－	P.74
@SET	Acquisition/set the control value of the freezer.	－	－	P.74
Variable related to state of equipment				
@SRQ	Acquire interrupt.	－	－	P.74
@ALARM	Acquire the alarm state.	－	－	P.74
@MODE	Acquisition/set the mode of operation.	－	－	P.74
@MON	Acquire the state in the tank.	－	－	P.75
@HEATER	Acquire the state of the heater.	－	－	P.75
@POWER	Set the state of the power supply.	－	－	P.75

Variable identifier	Explanation	Value		Reference
		get	put	
Variable related to program				
@RUNPRGDATA	Acquisition/set the remote program data.	－	－	P.76

3.4.2.1. Variable related to provider information

It is a variable to acquire information that this provider is maintaining.

3.4.2.1.1. @LASTDEVERR

Acquire the device error that occurred at the end. Details are variable ″of the CaoController class. @LASTDEVERRRefer to″.

3.4.2.2. Variable related to set value of equipment

It is a variable that acquisition/sets the set value set to the equipment.

3.4.2.2.1. @ROM

Acquire ROM version.

3.4.2.2.2. @MASK

Acquisition/set the interrupt mask. Please refer to ″@MASK″ variable of CaoController class for details.

3.4.2.2.3. @KEYLOCK

Acquisition/set the state of the key lock. Please refer to ″GetKEYLOCK″ and ″SetKEYLOCK″ command of CaoExtension class for details.

Data type

Item	Type explanation	
Acquisition/setting	VT_BSTR	State of key lock

Usage example

```
' addition of extension board
Dim caoExt As CaoExtension
Set caoExt = caoCon.AddExtension(″Extension″, ″Addr = 1″)
Variable ' addition
Dim var As CaoVariable
Set var = caoExt.AddVariable(″@KEYLOCK″)
Value ' acquisition
Dim val As String
```

```
    val = var.value
    Value ’  setting
    var.value = ″ON″
```

3.4.2.2.4. @TYPE

Please refer to ″@TYPE″ variable of CaoController class for details.

3.4.2.2.5. @TEMPMON

Please refer to ″@TEMPMON″ variable of CaoController class for details.

3.4.2.2.6. @TEMP

Please refer to ″@TEMP″ variable of CaoController class for details.

3.4.2.2.7. @STEMP

Please refer to ″@STEMP″ variable of CaoController class for details.

3.4.2.2.8. @HTEMP

Please refer to ″@HTEMP″ variable of CaoController class for details.

3.4.2.2.9. @LTEMP

Please refer to ″@LTEMP″ variable of CaoController class for details.

3.4.2.2.10. @HUMIMON

Please refer to ″@HUMIMON″ variable of CaoController class for details.

3.4.2.2.11. @HUMI

Please refer to ″@HUMI″ variable of CaoController class for details.

3.4.2.2.12. @SHUMI

Please refer to ″@SHUMI″ variable of CaoController class for details.

3.4.2.2.13. @HHUMI

Please refer to ″@HHUMI″ variable of CaoController class for details.

3.4.2.2.14. @LHUMI

Please refer to "@LHUMI" variable of CaoController class for details.

3.4.2.2.15. @SET

Acquisition/set the control value of the freezer. Please refer to "GetSET" and "SetSET" command of CaoExtension class for details.

Data type

Item	Type explanation	
Acquisition/setting	VT_BSTR	Control value

Usage example

```
    '  addition of extension board
    Dim caoExt As CaoExtension
    Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
    Variable '  addition
    Dim var As CaoVariable
    Set var = caoExt.AddVariable("@SET")
    Value '  acquisition
    Dim val As String
    val = var.value
    Value '  setting
    var.value = "REF9"
```

3.4.2.3. Variable related to state of equipment

It is a variable that acquisition/sets the state of the equipment.

3.4.2.3.1. @SRQ

Acquire interrupt. Please refer to "@SRQ" variable of CaoController class for details.

3.4.2.3.2. @ALARM

Acquire the alarm state. Please refer to "@ALARM" variable of CaoController class for details.

3.4.2.3.3. @MODE

Acquisition/set the mode of operation. Please refer to "@TEMPMON" variable of CaoController class for details.

Data type

Item	Type explanation

Item	Type explanation	
Acquisition/setting	VT_BSTR	Mode of operation

Usage example

```
    '  addition of extension board
    Dim caoExt As CaoExtension
    Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
    Variable '  addition
    Dim var As CaoVariable
    Set var = caoExt.AddVariable("@MODE")
    Value '  acquisition
    Dim val As String
    val = var.value
    Value '  setting
    var.value = "STANDBY"
```

3.4.2.3.4. @MON

Acquire the state in the tank.

Data type

Please refer to "GetMON" command of CaoExtension class for details.

Usage example

```
    '  addition of extension board
    Dim caoExt As CaoExtension
    Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
    Variable '  addition
    Dim var As CaoVariable
    Set var = caoExt.AddVariable("@MON")
    Value '  acquisition
    Dim values As Variant
    values = var.value
```

3.4.2.3.5. @HEATER

Acquire the state of the heater. Please refer to "@HEATER" variable of CaoController class for details.

3.4.2.3.6. @POWER

Set the state of the power supply.

Data type

Please refer to "SetPOWER" command of CaoExtension class for details.

Usage example

```
    '  addition of extension board
    Dim caoExt As CaoExtension
    Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
    Variable '  addition
    Dim var As CaoVariable
    Set var = caoExt.AddVariable("@POWER")
    Value '  setting
    var.value = "OFF"
```

3.4.2.4. Variable related to program

It is a variable that acquisition/sets a state and a set value of the program.

3.4.2.4.1. @RUNPRGDATA

Acquisition/set the remote program data. Please refer to "@RUNPRGDATA" variable of CaoController class for details.

4. Programming by espec TempHumi provider

In the espec TempHumi provider, client PC and the TempHumi RS-485 device or the TempHumi Ethernet device can be connected according to the following procedures.

Client PC and the TempHumi RS-485 device can be connected by serial communications according to the following procedures in the beginning.

- Making of CaoEngine
- Making of CaoWorkspace
- Making of CaoController
- Making of CaoExtension

After it connects it with the TempHumi RS-485 device, it can access information on the TempHumi RS-485 device by using the Execute method of CaoExtension or generating the CaoVariable object.

Next, client PC and the TempHumi Ethernet device can be connected by the Ethernet communication according to the following procedures.

- Making of CaoEngine
- Making of CaoWorkspace
- Making of CaoController

After it connects it with the TempHumi Ethernet device, it can access information on the TempHumi Ethernet device by using the Execute method of CaoController or generating the CaoVariable object.

4.1. Sample programming to which mode of operation of TempHumi RS-485 device is changed

The sample program that changes the mode of operation as an example is shown here. Table4-1The requirement for [ni] sample programFigure4-1The flow of [ni] sample program is described respectively.

Table4-1Requirement for sample program

Requirement	Explanation
Connection destination	Connect it with the cereal.
	Connection destination COM port is COM1.
Content of processing	Acquire a present mode of operation.
	Change the mode of operation in the state of the standby.

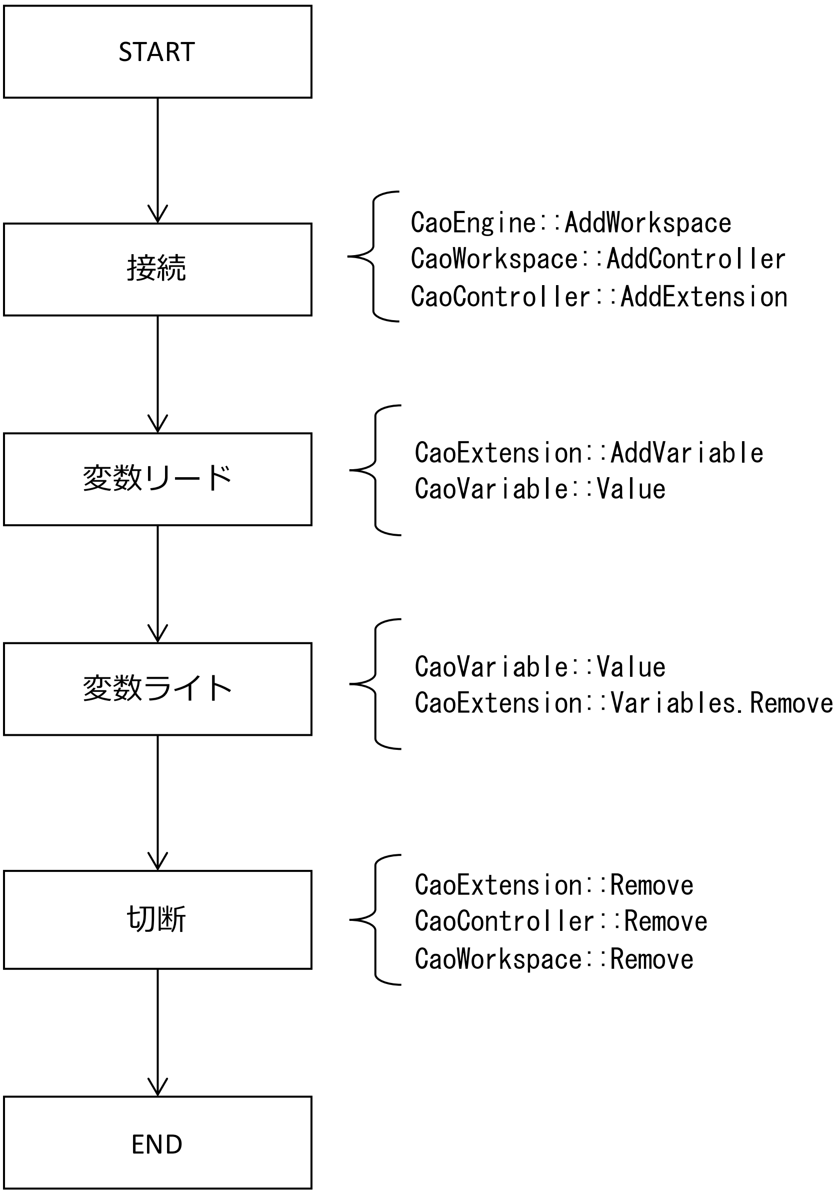


Figure4-1Flow of driving mode change

A concrete code is shown from the following paragraphs.

4.1.1. Sample program

The whole image of the sample program is shown as follows.

Sample	ChangeOperationMode_STH-120.vb

```
'  object
Dim caoEng As CaoEngine
Dim caoWor As CaoWorkspace
Dim caoCon As CaoController
Dim caoExt As CaoExtension
```

```
Private Sub Main
    '  connection
    Call Connect

     '  generation of CaoVariable object
    Dim varMODE As CaoVariable
    Set varMODE = caoExt.AddVariable("@MODE")

    '  acquire the mode of operation.
    Dim value As String
    value = varMODE.value

    '  set the mode of operation.
    varMODE.value = "STANDBY"

     '  delete CaoVariable from CaoExtension.
    Call caoExt.Variables.Remove(varMODE.Index)
     '  deletion of CaoVariable
    Set varMODE = Nothing

    '  cutting
    Call Disconnect
End Sub

Connection '  method
Private Sub Connect()
     '  generation of CaoEngine object
    Set caoEng = New CaoEngine
     '  generation of CaoWorkspace object
    Set caoWor = caoEng.AddWorkspace("Workspace", "")
     '  generation of CaoController object
    Set caoCon = caoWor.AddController("Controller", _
                                      "CaoProv.espec.TempHumi", _
                                      "", _
                                      "conn=com:1")
     '  generation of CaoExtension object
    Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
End Sub

Cutting '  method
Private Sub Disconnect()
     '  delete CaoExtension from CaoController.
    Call caoCon.Extensions.Remove(caoExt.Index)
     '  deletion of CaoExtension
    Set caoExt = Nothing

     '  delete CaoController from CaoWorkspace.
    Call caoWor.Controllers.Remove(caoCon.Index)
     '  deletion of CaoController
    Set caoCon = Nothing

     '  delete CaoWorkspace from CaoEngine.
    Call caoEng.Workspaces.Remove(caoWor.Index)
     '  deletion of CaoWorkspace
    Set caoWor = Nothing

     '  deletion of CaoEngine
    Set caoEng = Nothing
End Sub
```

4.1.1.1. TempHumi RS-485 device connection

Take the following procedures to connect it with the TempHumi RS-485 device.

(1) Prepare the variable to maintain the object. The object necessary for the device connection is CaoEngine object, CaoWorkspace object, CaoController object, and CaoExtension object. When the CaoController object is acquired from CaoWorkspaces, the CaoWorkpace object need not prepare the variable. Moreover, the CaoVariable object to access the variable is needed. The example of the code in VB6 is shown as follows.

```
    Dim caoEng As CaoEngine                 '  variable for CaoEngine object
    Dim caoWor As CaoWorkspace              '  variable for CaoWorkspace object
    Dim caoCon As CaoController             '  variable for CaoController object
    Dim caoExt As CaoExtension              '  variable for CaoExtension object
    Dim varMODE As CaoVariable              '  variable for CaoVariable object
```

(2) Generate the CaoEngine object. The CaoEngine object uses and generates the New key word.

```
     '  generation of CaoEngine object
    Set caoEng = New CaoEngine
```

(3) Acquire the CaoWorkspace object or generate it. When the CaoEngine object is generated, the CaoWorkspaces object and the CaoWorkspace object are generated with default one by one. The example of the code of newly generating the CaoWorkspace object and CaoWorkspace of default are shown as follows.

```
     '  generation of CaoWorkspace object
    Set caoWor = caoEng.AddWorkspace("Workspace", "")
```

(4) Generate the CaoController object. Set the CaoController object and set the provider name used and the parameter to use it to generate it. In the espec TempHumi provider, specify the COM port of the connection destination in the option. The example of the code is shown

as follows.

```
 '  generation of CaoController object
Set caoCon = caoWor.AddController("Controller", _
                                    "CaoProv.espec.TempHumi", _
                                    "", _
                                    "conn=com:1")
```

(5) Generate the CaoExtension object. Set the CaoExtension object and set the parameter to use it to generate it. In the espec TempHumi provider, specify the address of the connection destination in the option. The example of the code is shown as follows.

```
 '  generation of CaoExtension object
Set caoExt = caoCon.AddExtension("Extension", "Addr = 1")
```

4.1.1.2. Acquisition/setting of variable

Generate the CaoVariable object of the variable to be accessed acquisition/to set the value of the variable, and reference/set the Value property of the CaoVariable object. When the Value property reference/is set, it is necessary to prepare the type variable matched to the Value property. The example of the code is shown as follows.

```
 '  generation of CaoVariable object
Dim varMODE As CaoVariable
Set varMODE = caoExt.AddVariable("@MODE")

'  acquire the mode of operation.
Dim value As String
value = varMODE.value

'  set the mode of operation.
varMODE.value = "STANDBY"

 '  delete CaoVariable from CaoExtension.
Call caoExt.Variables.Remove(varMODE.Index)
 '  deletion of CaoVariable
Set varMODE = Nothing
```

4.1.1.3. TempHumi RS-485 device cutting

Delete the generated object, and delete the object deleted from the collection class that manages the object when the TempHumi RS-485 device and cutting it. The example of the code is shown as follows.

```
 '  delete CaoExtension from CaoController.
Call caoCon.Extensions.Remove(caoExt.Index)
 '  deletion of CaoExtension
Set caoExt = Nothing
 '  delete CaoController from CaoWorkspace.
```

```
    Call caoWor.Controllers.Remove(caoCon.Index)
     '  deletion of CaoController
    Set caoCon = Nothing
     '  delete CaoWorkspace from CaoEngine.
    Call caoEng.Workspaces.Remove(caoWor.Index)
     '  deletion of CaoWorkspace
    Set caoWor = Nothing
     '  deletion of CaoEngine
    Set caoEng = Nothing
```

4.2. Sample programming to which mode of operation of TempHumi Ethernet device is changed

The sample program that changes the mode of operation as an example is shown here. Table4-2The requirement for [ni] sample programFigure4-2The flow of [ni] sample program is described respectively.

Table4-2Requirement for sample program

Requirement	Explanation
Connection destination	Connect it by Ethernet (TCP/IP).
	Connection destination Internet Protocol address is 192.168.0.2.
Content of processing	Acquire a present mode of operation.
	Change the mode of operation in the state of the standby.

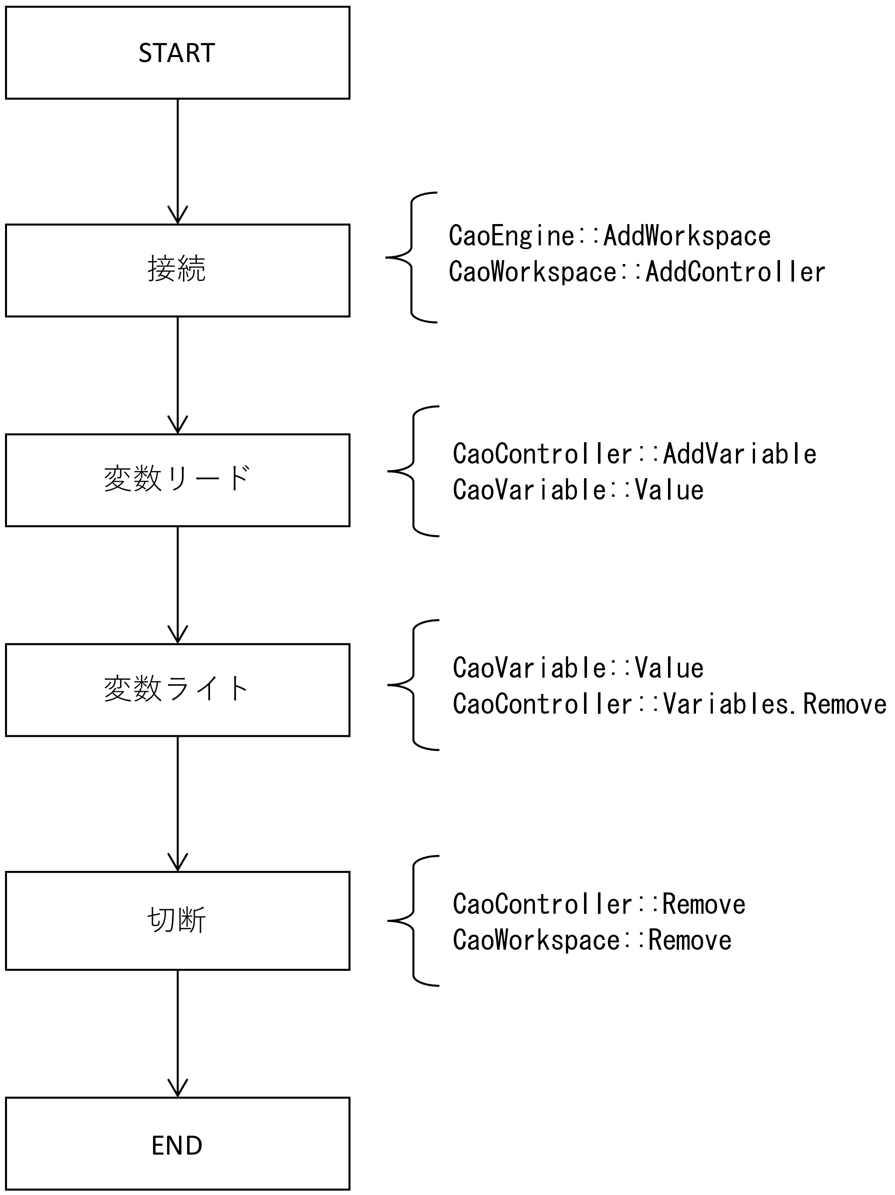


Figure4-2Flow of driving mode change

A concrete code is shown from the following paragraphs.

4.2.1. Sample program

The whole image of the sample program is shown as follows.

Sample	ChangeOperationMode_SH-642.vb

```
'  object
Dim caoEng As CaoEngine
Dim caoWor As CaoWorkspace
Dim caoCon As CaoController
```

```
Private Sub Main
    ’ connection
    Call Connect

     ’ generation of CaoVariable object
    Dim varMODE As CaoVariable
    Set varMODE = caoCon.AddVariable("@MODE")

    ’ acquire the mode of operation.
    Dim value As String
    value = varMODE.value

    ’ set the mode of operation.
    varMODE.value = "STANDBY"

     ’ delete CaoVariable from CaoExtension.
    Call caoCon.Variables.Remove(varMODE.Index)
     ’ deletion of CaoVariable
    Set varMODE = Nothing

    ’ cutting
    Call Disconnect
End Sub

Connection ’ method
Private Sub Connect()
     ’ generation of CaoEngine object
    Set caoEng = New CaoEngine
     ’ generation of CaoWorkspace object
    Set caoWor = caoEng.AddWorkspace("Workspace", "")
     ’ generation of CaoController object
    Set caoCon = caoWor.AddController("Controller", _
                                       "CaoProv.espec.TempHumi", _
                                       "", _
                                       "conn=eth:192.168.0.2")
End Sub

Cutting ’ method
Private Sub Disconnect()
     ’ delete CaoController from CaoWorkspace.
    Call caoWor.Controllers.Remove(caoCon.Index)
     ’ deletion of CaoController
    Set caoCon = Nothing

     ’ delete CaoWorkspace from CaoEngine.
    Call caoEng.Workspaces.Remove(caoWor.Index)
     ’ deletion of CaoWorkspace
    Set caoWor = Nothing
    '  deletion of CaoEngine
    Set caoEng = Nothing
End Sub
```

4.2.1.1. TempHumi Ethernet device connection

Take the following procedures to connect it with the TempHumi RS-485 device.

(1) Prepare the variable to maintain the object. The object necessary for the device connection is CaoEngine object, CaoWorkspace object, CaoController object, and CaoExtension object. When the CaoController object is acquired from CaoWorkspaces, the CaoWorkpace object need not prepare the variable. Moreover, the CaoVariable object to access the variable is needed. The example of the code in VB6 is shown as follows.

```
    Dim caoEng As CaoEngine              '  variable for CaoEngine object
    Dim caoWor As CaoWorkspace           '  variable for CaoWorkspace object
    Dim caoCon As CaoController          '  variable for CaoController object
    Dim varMODE As CaoVariable           '  variable for CaoVariable object
```

(2) Generate the CaoEngine object. The CaoEngine object uses and generates the New key word.

```
    '  generation of CaoEngine object
    Set caoEng = New CaoEngine
```

(3) Acquire the CaoWorkspace object or generate it. When the CaoEngine object is generated, the CaoWorkspaces object and the CaoWorkspace object are generated with default one by one. The example of the code of newly generating the CaoWorkspace object and CaoWorkspace of default are shown as follows.

```
    '  generation of CaoWorkspace object
    Set caoWor = caoEng.AddWorkspace("Workspace", "")
```

(4) Generate the CaoController object. Set the CaoController object and set the provider name used and the parameter to use it to generate it. In the espec TempHumi provider, specify the COM port of the connection destination in the option. The example of the code is shown as follows.

```
    '  generation of CaoController object
    Set caoCon = caoWor.AddController("Controller", _
                                      "CaoProv.espec.TempHumi", _
                                      "", _
                                      "conn=eth:192.168.0.2")
```

4.2.1.2. Acquisition/setting of variable

Generate the CaoVariable object of the variable to be accessed acquisition/to set the value of the variable, and reference/set the Value property of the CaoVariable object. When the Value property reference/is set, it is necessary to prepare the type variable matched to the Value property. The example of the code is shown as follows.

```
     '  generation of CaoVariable object
    Dim varMODE As CaoVariable
    Set varMODE = caoCon.AddVariable("@MODE")

    '  acquire the mode of operation.
    Dim value As String
    value = varMODE.value

    '  set the mode of operation.
    varMODE.value = "STANDBY"

     '  delete CaoVariable from CaoController.
    Call caoCon.Variables.Remove(varMODE.Index)
     '  deletion of CaoVariable
    Set varMODE = Nothing
```

4.2.1.3. TempHumi Ethernet device cutting

Delete the generated object, and delete the object deleted from the collection class that manages the object when the TempHumi RS-485 device and cutting it. The example of the code is shown as follows.

```
     '  delete CaoController from CaoWorkspace.
    Call caoWor.Controllers.Remove(caoCon.Index)
     '  deletion of CaoController
    Set caoCon = Nothing
     '  delete CaoWorkspace from CaoEngine.
    Call caoEng.Workspaces.Remove(caoWor.Index)
     '  deletion of CaoWorkspace
    Set caoWor = Nothing
     '  deletion of CaoEngine
    Set caoEng = Nothing
```

5. Espec TempHumi provider error code

In this provider, the following original error codes in which the mask is done with 0x8011**** exist. (Table5-1Reference)

About a common error of ORiN2, "ORiN2 プログラミングガイド Refer to the chapter of the error code of".

Table5-1Original error code table

Error number	Explanation
0x80110001	There is no optional address.
0x80110002	The address has already been registered.
0x80110003	The object cannot be added.

Moreover, this provider does the mask by "0x8010****" and returns the error code of the device. It is the following error codes based on the error message that has been sent from the device. (Table5-2Reference)

Table5-2Error code table of device

Error number	Error message	Explanation
0x80100001	COMMAND ERR	It is not possible to recognize it as command data.
0x80100002	CONTROLLER NOT READY-1	The command data concerning humidity was sent to the device that did not support the moisture control.
0x80100003	CONTROLLER NOT READY-2	The command data concerning the program was sent while driving the program.
0x80100004	CONTROLLER NOT READY-3	In the state of the power-off, the key lock setting was done.
0x80100005	CONTROLLER NOT READY-4	It tried to set the signal of the time of a setting change not acceptable.
0x80100006	CONTROLLER NOT READY-5	The command related to the freezer was issued to the chamber of the freezer uninstalling.
0x80100007	CONTROLLER NOT READY-6	The command related to the dump truck was issued to the chamber of the dump truck uninstalling.
0x80100008	DATA NOT READY	It tried to execute the program pattern where effective data was not set.
0x80100009	PARAMETER ERR	The parameter that is the parameter necessary for a set command or added cannot be recognized.
0x8010000A	DATA OUT OF RANGE	Set) degree setting value ..exceeded the setting range.. ..tepid.. ..(.. ..dampness...

Error number	Error message	Explanation
0x8010000B	PROTECT ON	Data was key lock set changed when it was effective.
0x8010000C	PRGM WRITE ERR-1	Data was written without specifying edit/replace mode.
0x8010000D	PRGM WRITE ERR-2	The command concerning the edit was issued while not editing it.
0x8010000E	PRGM WRITE ERR-3	The superscription operated while the edit was operating.
0x8010000F	PRGM WRITE ERR-4	The edit operation was done while overwriting.
0x80100010	PRGM WRITE ERR-5	When it was not an overwrite, the command concerning the superscription was sent.
0x80100011	PRGM WRITE ERR-6	A program pattern different while writing it was specified.
0x80100012	PRGM WRITE ERR-7	Step No is not continuous.
0x80100013	PRGM WRITE ERR-8	The mistake is repeatedly found in the setting.
0x80100014	PRGM WRITE ERR-9	It tried to edit the program while driving.
0x80100015	PRGM WRITE ERR-10	The repetition setting and end condition though effective data was not set were set.
0x80100016	PRGM WRITE ERR-11	It tried to set invalid data.
0x80100017	PRGM WRITE ERR-12	The processing setting of the exposing time was set at inclination setting ON.
0x80100018	PRGM WRITE ERR-13	When humidity control OFF was set, the humidity inclination was set.
0x80100019	CMD_ERR	The mistake is found in the main command.
0x8010001A	INVALID REQ	The function that the device was not able to correspond was specified.
0x8010001B	CHB NOT READY	When the device was not accepted, the command was specified.
0x8010001C	PARA ERR	The mistake is found in an optional parameter.
0x8010001D	The error message that had not been described to "Table 1.1 error message" of "Data treated by 1.4 communication function" of the TempHumi RS-485 manual was received.	

6. Appendix

付録A. Correspondence of command of TempHumi [puropaida] and command of device

The correspondence of the command of the CaoController class of this provider, the command of the CaoExtension class, and the command of the device is as follows.

Table6-1For command of provider and command of device

Command of [puropaida]		Command of device
CaoController class	CaoExtension class	
GetROM		ROM?
GetROMDISP	-	ROM?,DISP
GetMASK		MASK?
SetMASK		MASK
GetTIMERMON	-	TIMER ON?
GetTIMERSET	-	TIMER USE?
GetTIMER0	-	TIMER LIST?,0
SetTIMER0	-	TIMER WRITE,NO0
GetTIMER1	-	TIMER LIST?,1
SetTIMER1	-	TIMER WRITE,NO1
GetTIMER2	-	TIMER LIST?,2
SetTIMER2	-	TIMER WRITE,NO2
GetKEYPROTECT	-	KEYPROTECT?
SetKEYPROTECT	-	KEYPROTECT
-	GetKEYLOCK	KEY PROTECT?
-	SetKEYLOCK	KEY PROTECT
GetTYPE		TYPE?
GetTEMP		TEMP?
SetTEMP		TEMP
SetSTEMP		TEMP
SetHTEMP		TEMP
SetLTEMP		TEMP
GetHUMI		HUMI?
SetHUMI		HUMI
SetSHUMI		HUMI
SetHHUMI		HUMI
SetLHUMI		HUMI
GetSET		SET?

Command of [puropaida]		Command of device
CaoController class	CaoExtension class	
SetSET		SET
GetCONSTANTSETTEMP	－	CONSTANT SET?TEMP
GetCONSTANTSETHUMI	－	CONSTANT SET?HUMI
GetCONSTANTSETREF	－	CONSTANT SET?REF
GetSYSTEMSETPTS	－	SYSTEM SET?PTS
GetSYSTEMSETPTC	－	SYSTEM SET?PTC
GetSYSTEMSETPTCOPT	－	SYSTEM SET?PTCOPT
SetTIMERMERASE	－	TIMER ERASE and NO timer - Number
SetTIMERON	－	TIMER, turning on, and timer - Number
SetTIMEROFF	－	TIMER, turning off, and timer - Number
GetDATE	－	DATE?
SetDATE	－	DATE
GetTIME	－	TIME?
SetTIME	－	TIME
GetSRQ		SRQ?
SetSRQ		SRQ
GetALARM		ALARM?
GetMODE		MODE?
SetMODE		MODE
GetMODEDETAIL	－	MODE?,DETAIL
GetMONDETAIL	－	MON?,DETAIL
GetHEATER		%?
－	SetPOWER	POWER
GetRUNPRGDATA		RUN PRGM?
SetRUNPRGDATA		RUN PRGM

付録B. Variable of TempHumi [puropaida] and correspondence of command of device

The CaoController class variable, the CaoExtension class variable of this provider, and the correspondence of the command of the device are as follows.

Table6-2For command of device variable of provider

Variable of [puropaida]		Command of device	
CaoController class	CaoExtension class	get_Value	put_Value
@ROM		ROM?	－

Variable of [puropaida]		Command of device	
CaoController class	CaoExtension class	get_Value	put_Value
@ROMDISP	－	ROM?,DISP	－
@MASK		MASK?	MASK
@TIMERMON	－	TIMER ON?	－
@TIMERSET	－	TIMER USE?	－
@TIMER0	－	TIMER LIST?,0	TIMER WRITE,NO0
@TIMER1	－	TIMER LIST?,1	TIMER WRITE,NO1
@TIMER2	－	TIMER LIST?,2	TIMER WRITE,NO2
@TIMERERASE	－	－	TIMER ERASE and NO timer number
@TIMERON	－	－	TIMER, turning on, and timer number
@TIMEROFF	－	－	TIMER, turning off, and timer number
@KEYPROTECT	－	KEYPROTECT?	KEYPROTECT
－	@KEYLOCK	KEY PROTECT?	KEY PROTECT
@TYPE		TYPE?	－
@TEMPMON		TEMP?	－
@TEMP		－	TEMP
@STEMP		－	TEMP
@HTEMP		－	TEMP
@LTEMP		－	TEMP
@HUMIMON		HUMI?	－
@HUMI		－	HUMI
@SHUMI		－	HUMI
@HHUMI		－	HUMI
@LHUMI		－	HUMI
@SET		SET?	SET
@CONSTANTSETTEMP	－	CONSTANT SET?TEMP	－
@CONSTANTSETHUMI	－	CONSTANT SET?HUMI	－

Variable of [puropaida]		Command of device	
CaoController class	CaoExtension class	get_Value	put_Value
@CONSTANTSETREF	－	CONSTANT SET?REF	－
@SYSTEMSETPTS	－	SYSTEM SET?PTS	－
@SYSTEMSETPTC	－	SYSTEM SET?PTC	－
@SYSTEMSETPTCOPT	－	SYSTEM SET?PTCOPT	－
@DATE	－	DATE?	DATE
@TIME	－	TIME?	TIME
@SRQ		SRQ?	－
@ALARM		ALARM?	－
@MODE		MODE?	MODE
@MODEDETAIL	－	MODE?,DETAIL	－
－	@MON	MON?	－
@MONDETAIL	－	MON?,DETAIL	－
@HEATER		%?	－
－	@POWER	－	POWER
@RUNPRGDATA		RUN PRGM?	RUN PRGM

付録C. Corresponding model list

The corresponding model list described in the TempHumi RS-485 manual and the TempHumi Ethernet manual is described.

Table6-3RS-485 corresponding model list

Model
SU-221
SU-241
SU-261
SU-641
SU-661
SH-221
SH-241
SH-261
SH-641
SH-661

MC-711T
MC-811T
LH-113
LHL-113
LHU-113
LU-113
LC-113
LC-123
LC-223
LG-113
LG-123
LCV-233
LCV-233P
LCV-243
LCV-243P
ST-110
ST-120
STH-120

Table6-4Ethernet corresponding model list

Model
Small, environmental examination machine （SU/SH）
Stability examination machine （CSH）
Stability examination room （CWH）
Light specs constant temperature （equilibrium humidity） container （LU/LH）
Small ultra low temperature constant temperature machine mini-subzero （MC）
Low temperature constant temperature [**] [shimeutsuwa] （CRH）