CV Provider
KEYENCE CV series

Version 1.0.2

User's guide

August 23, 2019

[Remarks]

[Revision history]

Version	Date	Content
1. 0. 0	2/11/2014	First edition.
1. 0. 1	1/13/2016	Expanded arguments of Trigger command. Corrected TriggerAndGetResult command. Added Asynchronous commands (ChangeModeAsync, ChangeInspectionSettingAsync, RefreshReferencePositionAsync) Added GetCommandResult command. Updated the error code list.
1.0.2	2019-03-18	Added Synchronous commands (RegistImage, GetRegistImageList, SaveAllImage, GetLibraryList,ChangePassword, InputPseudoConsole, SaveStatisticalData, ChangeShutterSpeed, ChangeCameraSensitivity, ChangeTriggerDelay, ChangeLightIntensityLevel, ChangePatternCounter)
1.0.2	2019-08-23	Added precautions for the use of asynchronous command

Content

1. Introduction

This document is a user's guide of CV provider that is CAO provider designed for the vision system "CV series" manufactured by KEYENCE.

CV provider executes non-procedure commands and also notifies transaction results to the CV series controller through Ethernet or RS232C

This document describes functions of CV provider and equipped methods.

1.1. Device setup

In CV provider, RS-232C and Ethernet are available for communication. Select a desired communication setting from the Global in the menu bar.

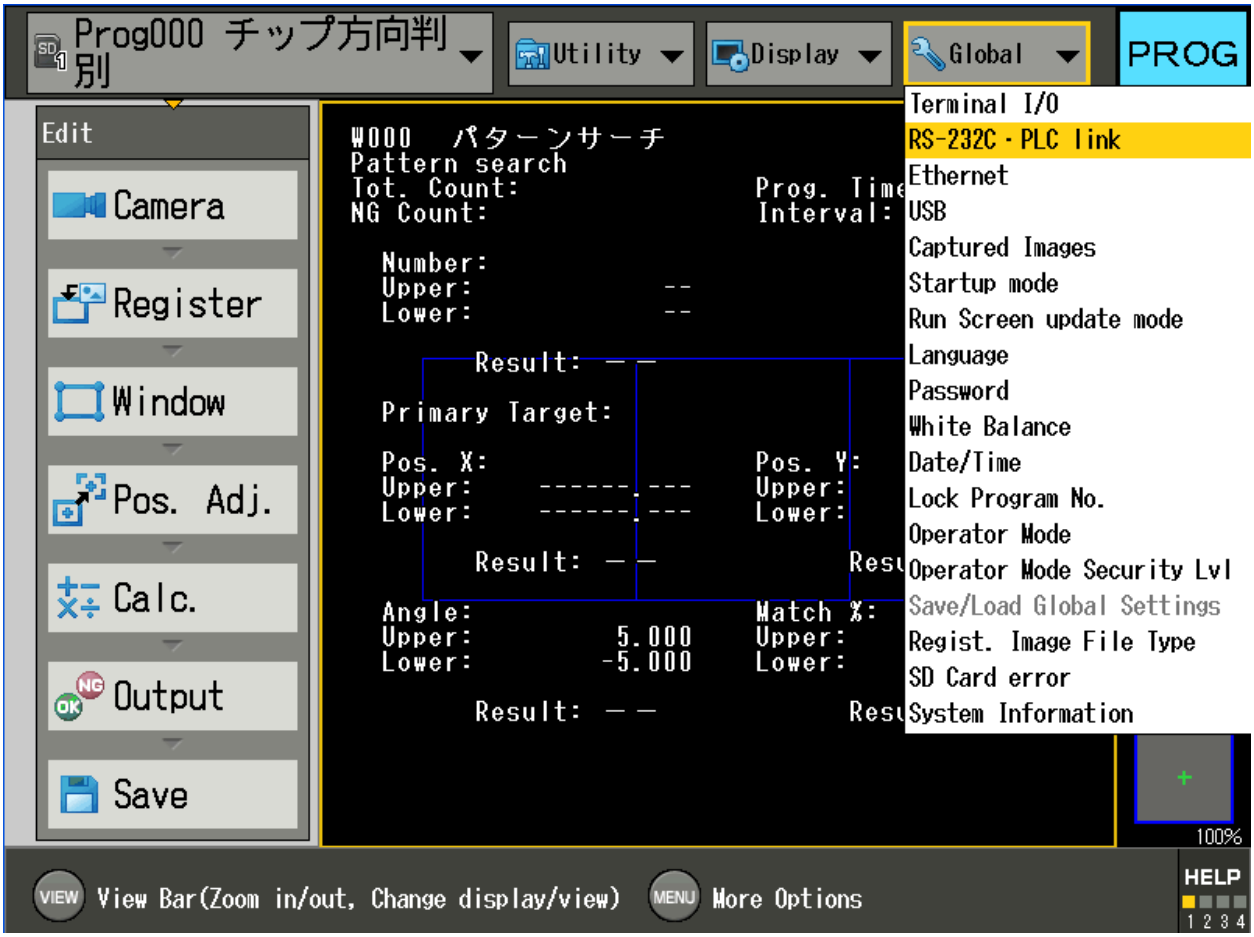


Figure 1−1 Global setting

1.1.1. Setup for RS232C connection

In RS232C communication setting, all item settings other than the Delimiter can be changed. If you change any setting from the default values, you need to change the connection parameters of AddController as well (see 2.2.1.1). Select CR for the Delimiter.

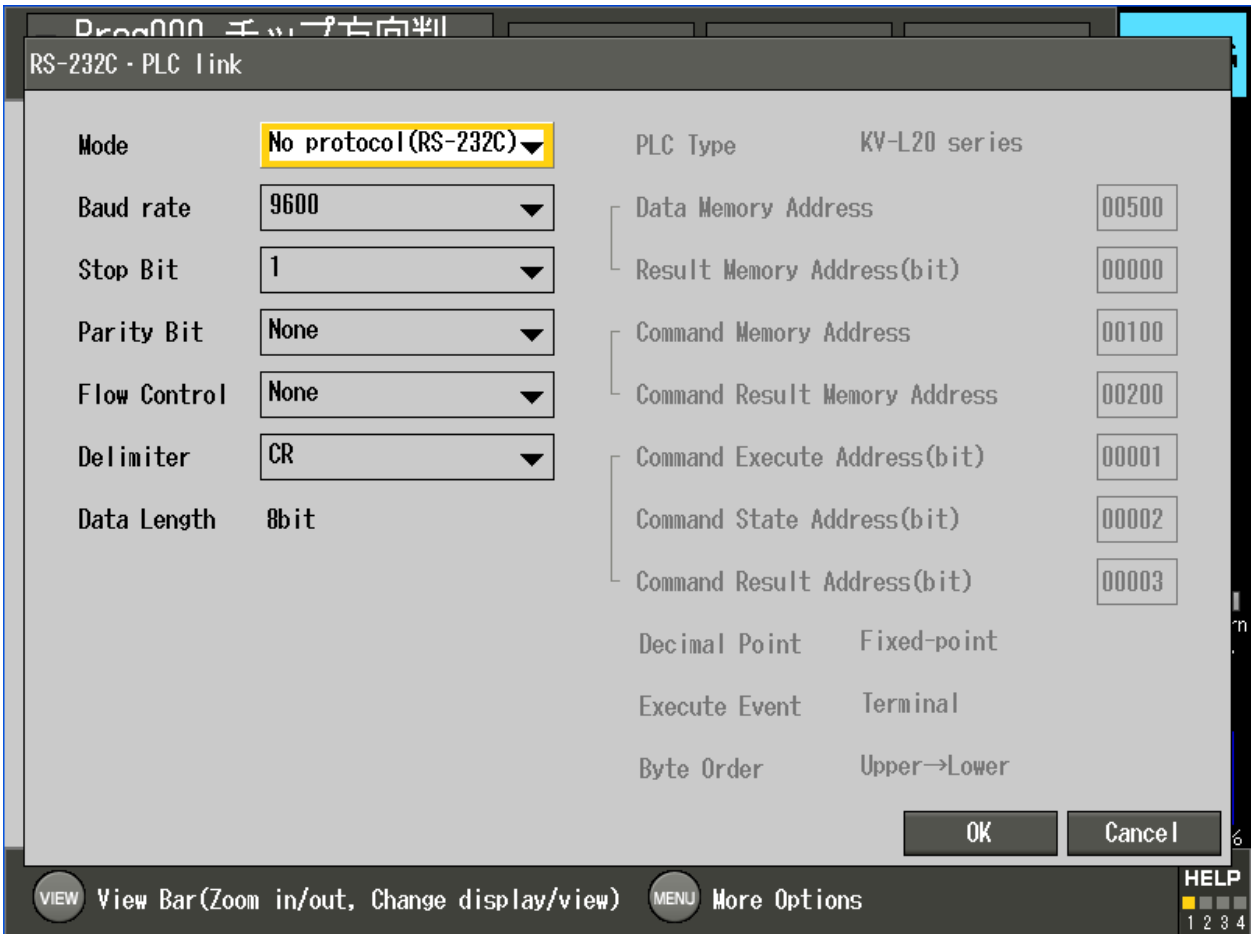
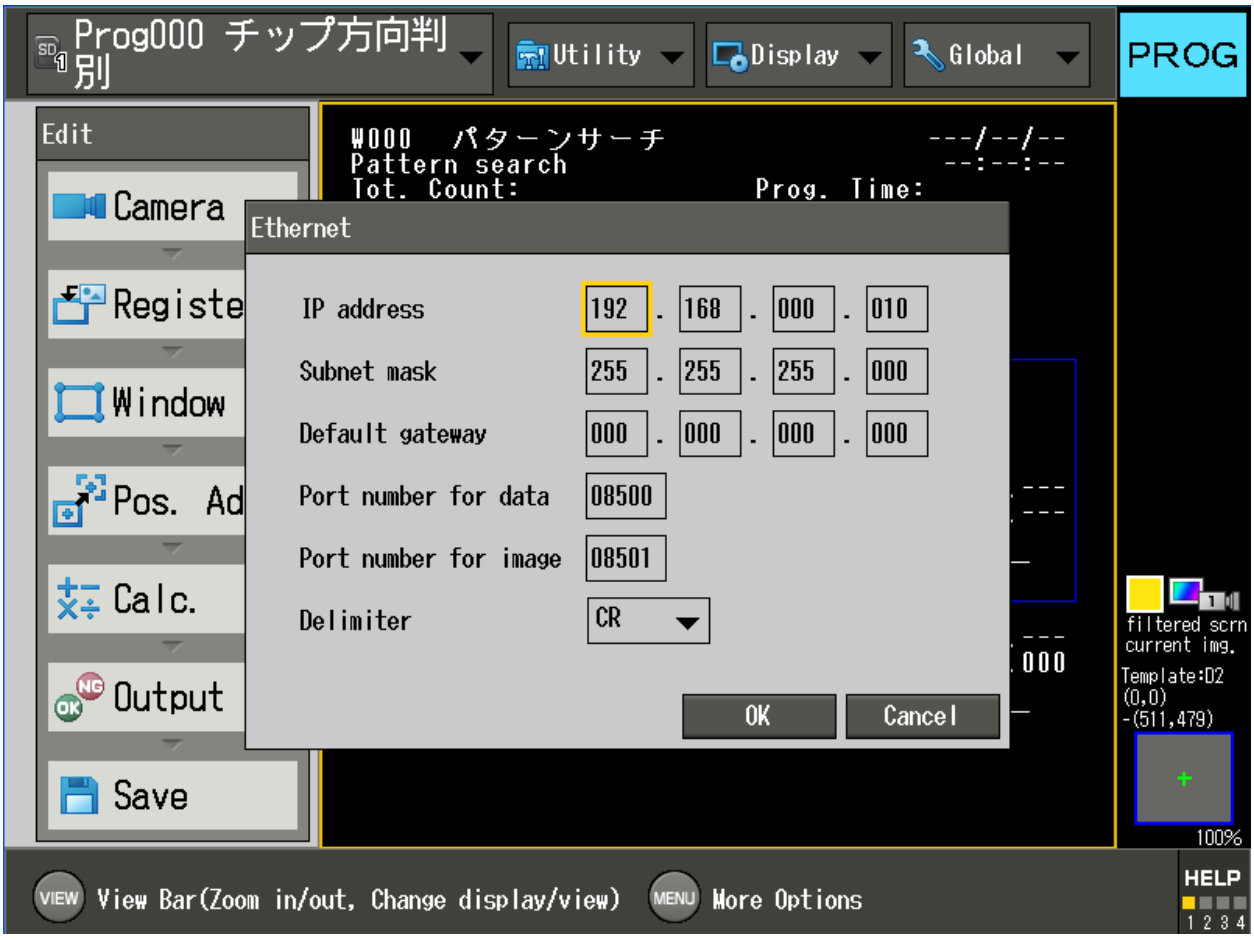


Figure 1−2 RS232C setting

1.1.2. Setup for Ethernet connection

In Ethernet connection setting, all items other than the Delimiter can be changed. If you change any settings from the default values, you need to change the connection parameters of AddController as well (see 2.2.1.1). Select CR for the Delimiter.


Prog000 チップ方向判別
Utility
Display
Global
PROG
Edit
Camera
Registe
Window
Pos. Ad
Calc.
Output
Save
W000 パターンサーチ
Pattern search
Tot. Count:
Prog. Time:
---/--/--
--:--:--
Ethernet
IP address
192 . 168 . 000 . 010
Subnet mask
255 . 255 . 255 . 000
Default gateway
000 . 000 . 000 . 000
Port number for data
08500
Port number for image
08501
Delimiter
CR
OK
Cancel
filtered scrn
current img.
Template:D2
(0,0)
-(511,479)
100%
VIEW View Bar(Zoom in/out, Change display/view)
MENU More Options
HELP
1 2 3 4

2. Overview of provider

2.1. Overview

CV provider provides CaoController::Execute methods for a command execution.

In CaoController::Execute, commands are sent or received through a serial interface.

Table 2−1 CV provider

File name	CaoProvCV. dll
ProgID	CaoProv. KEYENCE. CV
Registry registration[1]	regsvr32 CaoProvCV. dll
Remove registry registration	regsvr32 /u CaoProvCV. dll

2.2. Method and Property

2.2.1. CaoWorkspace::AddController method

CV provider establishes communication by referring to the connection parameters for communication when AddController is executed. At that time, communication method is specified in option.

Syntax AddController(<bstrCtrlName:VT_BSTR>, <bstrProvName:VT_BSTR>,
<bstrPcName:VT_BSTR > [, <bstrOption:VT_BSTR>])

bstrCtrlName : [in] A controller name (any name)

bstrProvName : [in] Provider name (Fixed to" CaoProv. KEYENCE. CV")

bstrPcName : [in] Provider execution machine name

bstrOption : [in] Option character string

Following is a list of option string items.

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

Option	Description
Conn =<Connection parameter>	This must be specified. Set communication configuration and its connection parameter. For details, see 2.2.1.1.
Timeout[=<Time>]	Specify a timeout period (millisecond) at the receiving and sending. (Default: 500)

[1] If the registry is installed by ORiN SDK, you do not need to register/delete it manually

2.2.1.1. Conn option

Following shows connection parameter strings for Conn option. Parameters surrounded by the square brackets (“[]”) can be omitted. Underlined part shows the default value when the option is not specified.

- Ethernet device

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

  <IP Address> : IP address of CV series being connected
  Example: ”192. 168. 0. 10”, “192. 168. 0. 11”

  <Port No> : Connection port number
  8500, 8501, . . . Arbitrary number can be specified.

Example

```
    Dim caoEng as CaoEngine
    Dim caoCtrl as CaoController

    Set caoEng  = New caoEngine
    Set caoCtrl = caoEng. Workspaces(0). AddController("CV", "caoProv. KEYENCE. CV", "",
  "conn=eth:192. 168. 0. 1, timeout=800")
```

- RS232C device

  "Conn=com:<COM Port>[:<BaudRate>[:<Parity>:<DataBits>:<StopBits>[:<Flow>]]]"

  <COM Port> : COM port number
  ‘1’-COM1, ’2’-COM2, …

  <BaudRate> : Baud rate
  4800, 9600, 19200, 38400, 57600, 115200.

  <Parity> : Parity
  ‘N’-NONE, ’E’-EVEN, ’O’-ODD.

  <DataBits> : Number of the data bit
  ‘7’-7bit, ’8’-8bit.

  <StopBits> : Number of the stop bit
  ‘1’-1bit, ’2’-2bit.

  <Flow> : Flow control
  ‘0’- Without flow control, ‘1’-Xon/Xoff, ’2’-Hardware control.

Example

```
Dim caoEng as CaoEngine
Dim caoCtrl as CaoController

Set caoEng  = New caoEngine
Set caoCtrl = caoEng. Workspaces(0). AddController("CV",  "caoProv. KEYENCE. CV",  "",
"conn=com:1")
```

2.2.2. CaoController::Execute method

CV provider sends and receives command with Non-procedure methods through Ethernet. Set a command name to the first argument, a command parameter to the second argument. For details about each command, see Command reference.

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

bstrCommandName: [in] Command name

vntParam : [in] Parameter

2.2.3. Error code

The processing result from the CV series at the method execution is returned as HRESULT. CV series-specific errors are masked with 0x80108000 and returned. For details about CV series-specific errors, see CV series user's guide.

When the process completed successfully (OK): S_OK (0)

When the process completed with errors (ER): 0x80108000 + Return value

Example : When ChangeMode is executed.

hr = 0x80108016: Either the number or content of the parameters is incorrect.

For details about other provider-common errors, refer to the ORiN2SDK Programmers User's Guide.

Table 2-1 Error code list

Error	Error number	Description
E_CVERROR_CVERR	0x80108000 | CV error	CV series original error (see2.2.3)
E_CVERROR_LENGTH	0x80100000	Packet length error
E_CVERROR_PACKET	0x80100001	Packet error
E_COMMAND_EXECUTING	0x80100002	Another command was executed during a command execution
E_GET_COMMAND_RESULT	0x80100003	GetCommandResult command was executed after a Synchronous command execution

3. Command reference

This chapter describes each command of CaoController::Execute method. For operation details of each command, refer to “Details of Communication Commands” in CV series user’s manual from KEYENCE.

Table 3−1 CaoController::Execute command list

CV series Non-procedure command	Command	Function	
Trigger			
T1, T2	Trigger	Inputs a trigger	P. 12
System control			
R0, S0	ChangeMode	Changes operation mode	P. 13
	ChangeModeAsync	Changes operation mode asynchronously (not recommended)	P. 13
RS	Reset	Reset	P. 14
SS	StoreSetting	Saves the program setting	P. 14
DS	ChangeDisplayPattern	Changes the display pattern	P. 15
UW	ChangeCurrentUnit	Changes the window number	P. 15
UR	ReadCurrentUnit	Reads the window number	P. 15
M0	ReoutputResult	Outputs the latest process result again	P. 16
BS	RegistImage	Registers an image.	P. 16
BL	GetRegistImageList	Obtains the list of registered images.	P. 17
BT	SaveAllImage	Reads all of the image data.	P. 17
DL	GetLibraryList	Obtains the list of libraries	P. 18
PS	ChangePassword	Changes a password.	P. 19
KY	InputPseudoConsole	Performs a console pseudo input.	P. 19
ST	SaveStatisticalData	Saves the statistical data.	P. 20
Change program settings			
PW	ChangeInspectSetting	Switches the program number	P. 21
	ChangeInspectSettingAsync	Switches the program number asynchronously (not recommended)	P. 21
PR	ReadInspectSetting	Reads the program number	P. 22
DR	ReadToolParameter	Reads a limit value	P. 22
DW	ChangeToolParameter	Writes the window or calculation limits	P. 23
JR	ReadBinaryData	Reads binary filter data	P. 24
JW	ChangeBinaryData	Writes binary filter data	P. 24
MI	InitCommandMemory	Sets the initial value for command memory	P. 24

MR	ReadCommandMemory	Reads the data from the command memory	P. 25
MW	ChangeCommandMemory	Writes the data to the command memory	P. 25
RR	RefreshReferencePosition	Recalculates the reference values	P. 26
	RefreshReferencePositionAsync	Recalculates the reference values asynchronously (not recommended)	P. 26
Input/Output control			
TE	EnableTrigger	Enables/Disables trigger input	P. 27
Camera control setting			
CW	ChangeShutterSpeed	Changes the shutter speed	P. 27
CW	ChangeCameraSensitivity	Changes the camera sensitivity.	P. 28
CW	ChangeTriggerDelay	Sets the amount of time to delay after the trigger input.	P. 28
CW	ChangeLightIntensityLevel	Changes the light intensity level value.	P. 2929
CW	ChangePatternCounter	Changes the pattern counter of multi pattern mode.	P. 29
Original extended command			
-	ExecuteCommand	Executes Non-procedure command	P. 29
-	ExecuteCommandAsync	Executes Non-procedure command asynchronously (not recommended)	P. 30
-	TriggerAndGetResult	Issues trigger and receive the output result	P. 30
-	RecievePacket	Receives packets	P. 31
-	ClearPacket	Clears the receiving packet in the buffer	P. 31
-	GetTimeout	Get the timeout period	P. 31
-	SetTimeout	Set the timeout period	P. 32
-	GetCommandResult	Gets the return value of asynchronous command	P. 32

3.1. Trigger

3.1.1. CaoController::Execute ("Trigger") command

Issues a trigger.

Syntax Trigger(< iTriggerNo >, [<iMode>]))

iTriggerNo : [in] Specify a trigger number (VT_I2)

1 : Trigger 1

2 : Trigger 2

iMode : [in] Specify the result flag. (VT_I4)

0 : Issues a trigger. Not receive the response result.

1 : Issues a trigger. Receive the response result. (default)

Return value : [out] Result strings (This effective only when “iMode = 1”)

To receive the execution result later, T1, which is the response packet of the trigger command, is added to the top. If an error occurs, response packet will be "ER, T1, NN"(NN is an CV original error code).

Do not execute any other command until the result has been received successfully.

The following sample shows how to issue Trigger number 1 and then output x =11 and y= 12

Example

```
Dim strResult as String

‘ To receive a result at the same time
‘ Issue a trigger & receive the response
strResult = caoCtrl. Execute("Trigger",  1)
‘ strResult: “+11, +12”

‘ To receive a result later
‘ Issue a trigger only.
Call caoCtrl. Execute("Trigger",  Array(1, 0))
‘ Any processing
‘ Receive the result
strResult = caoCtrl. Execute("RecievePacket")
‘ strResult: “T1, +11, +12”
```

3.2. System control

3.2.1. CaoController::Execute ("ChangeMode") command

Changes to Run mode or Program mode.

Syntax ChangeMode(< iMode >)

iMode : [in] Specify a desired mode

0 : Program mode

1 : Run mode

Return value : none

Executing the following example will switch the mode to Run mode.

Example

```
caoCtrl. Execute "ChangeMode",  1
```

3.2.2. CaoController::Execute ("ChangeModeAsync") command (not recommended)

Change to Run mode or Program mode asynchronously.

To obtain and check the return value of the command, use GetCommandResult command. For details about GetCommandResult, please refer to 3.6.8.CaoController::Execute ("GetCommandResult”) command.

When you use this command, be sure to disconnect the provider after the execution of GetCommandResult command.

Otherwise, an error may occur at asynchronous operation.

Syntax ChangeModeAsync(< iMode >)

iMode	:	[in] Specify a desired mode
		0 : Program mode
		1 : Run mode
Return value	:	[out] none

Executing the following example will switch the mode to Run mode.

Example

```
Dim vntResult as variant

caoCtrl.Execute "ChangeModeAsync", 1

' Obtain the return value of ChangeMoveAsync command
vntResult = caoCtrl.Execute("GetCommandResult")

' vntResult : Return value (Empty)
```

3.2.3. CaoController::Execute ("Reset") command

Resets items, such as Total Count, NG count, etc. For details, see CV series user's manual.

Syntax Reset

Argument	:	none
Return value	:	none

Example

```
caoCtrl. Execute "Reset"
```

3.2.4. CaoController::Execute ("StoreSetting") command

Saves the data of the currently selected program setting number.

Syntax StoreSetting

Argument	:	none
Return value	:	none

Example

```
caoCtrl. Execute "StoreSetting"
```

3.2.5. CaoController::Execute ("ChangeDisplayPattern") command

Changes the display pattern.

Syntax ChangeDisplayPattern(<bstrMode>, <bstrParam>)

bstrMode	:	[in] Select desired display pattern (VT_BSTR) PT : Display template RS : Result display PG : Page FC : Screen
bstrParam	:	[in] Subsidiary parameter according to the parameter selected above (VT_BSTR)
Return value	:	none

Executing the following example will display a previous page.

Example

```
caoCtrl. Execute "ChangeDisplayPattern", Array("PG", "P")
```

3.2.6. CaoController::Execute ("ChangeCurrentUnit") command

Switches to the specified window number.

Syntax ChangeCurrentUnit (<iWindowNo>)

iWindowNo	:	[in] Window No (0 to 127) (VT_I4)
Return value	:	[out] none

Executing the following example will change the window to the window number 2.

Example

```
Call caoCtrl. Execute("ChangeCurrentUnit", 2)
```

3.2.7. CaoController::Execute ("ReadCurrentUnit") command

Returns the current window number.

Syntax ReadCurrentUnit()

Argument : [in] none

Return value : [out] Window No (0 to 127) (VT_I4)

Executing the following example will return the current window number.

Example

```
Din lNum as Long
lNum = caoCtrl. Execute("ReadCurrentUnit")
```

3.2.8. CaoController::Execute ("ReoutputResult") command

Outputs the latest measurement result again.

Syntax ReoutputResult ()

Argument : [in] none

Return value : [out] Measurement result (VT_BSTR)

Example

```
Din bstrResult as String
bstrResult = caoCtrl. Execute("ReoutputResult")
```

3.2.9. CaoController::Execute ("RegistImage") command

Registers the latest captured image specified by camera No.

Syntax RegistImage (<lCameraNo>, <lRegistImageNo>)

lCameraNo : [in] Camera No. (VT_I4)

1 to 4

lRegistImageNo : [In] Registered image No. (VT_I4)

0 to 999

Return value : none

The following sample program shows how to register CameraNo1 image acquired at the latest into the Registered image No.0.

Example

```
Call caoCtrl.Execute("RegistImage", Array(1,0))
```

3.2.10. CaoController::Execute ("GetRegistImageList") command

Obtains the list of registered images that have been previously saved.

1. To obtain the list of registered images that are currently set.

Syntax GetRegistImageList (<lTarget>, <lCameraNo>)

lTarget	:	[in] Destination to obtain the list (VT_I4)
		0: Current setting
lCameraNo		[in] Camera No.(VT_I4)
		1 to 4
Return value	:	[Out] the list of registered image numbers (VT_I4 | VT_ARRAY)

2. To obtain the list of images saved in an SD card.

Syntax GetRegistImageList (<lTarget>, < lRegistImageNo>,<lCameraNo>)

lTarget	:	[in] Destination to obtain the list (VT_I4)
		1:SD card 1
		2:SD card 2
lRegistImageNo	:	[in] Program No. of the registered image to be obtained (VT_I4)
		0 to 999
lCameraNo		[in] Camera No.(VT_I4)
		1 to 4
Return value	:	[Out]the list of registered image numbers(VT_I4 | VT_ARRAY)

The following sample program shows how to obtain a list of image numbers that are currently set.

Example

```
Dim vntRet as variant
vntRet = caoCtrl.Execute("GetRegistImageList", Array(0,1))
```

3.2.11. CaoController::Execute ("SaveAllImage ") command

All the images that are stored in the image buffer are written to the SD card in bitmap format.

Syntax SaveAllImage (<lCameraNo>, < lCompressionRatio>, <bstrImageKind>,<bstrFolderName>)

lCameraNo	:	[in] Camera No. (VT_I4)
		1 to 4

lCompressionRatio	:	[in] Compression ratio (VT_I4) 0 : No compression 1 : 1/2 2 : 1/4 3 : 1/8
bstrImageKind	:	[in] Type of images to be saved (VT_BSTR) AL : All the image data stored in the image buffer NG : All the NG images stored in the image buffer OK : All the OK images stored in the image buffer
bstrFolderName	:	[in] Folder name (VT_BSTR) Enter any folder name.
Return value	:	none

The following sample program shows how to save the all of the image in the image buffer of Camera No.1 with 1/2 compression ratio.

Example

```
call caoCtrl.Execute("SaveAllImage", Array(1,1,"AL","ImageFolder"))
```

3.2.12. CaoController::Execute ("GetLibraryList") command

Obtains the list of library numbers that have set.

1. To obtain the list of library numbers used in the current program number.

Syntax GetLibraryList (<lTarget>)

lTarget	:	[in] Destination to obtain the list (VT_I4) 0: Current setting
Return value	:	[Out] Library number list (VT_I4 | VT_ARRAY)

2. To obtain a list of library numbers stored in an SD card.

Syntax GetLibraryList (<lTarget>, <lSettingNo>)

lTarget	:	[in] Destination to obtain the list (VT_I4) 1 : SD card 1 2 : SD card 2
lSettingNo	:	[in] Program No. of library number to obtain.(VT_I4) 0 to 999
Return value	:	[out] Library number list (VT_I4 | VT_ARRAY)

The following sample program shows how to obtain the list of library number used in the current program number.

Example

```
Dim vntRet as variant
vntRet = caoCtrl.Execute("GetLibraryList", 0))
```

3.2.13. CaoController::Execute ("ChangePassword") command

Changes the password.

Syntax ChangePassword (<lOldPassword>, < lNewPassword>)

lOldPassword	:	[in] old passwordVT_I4) 0000 to 9999
lNewPassword	:	[in] new password(VT_I4) 0000 to 9999
Return value	:	none

The following sample program shows how to change the password from “0000” to “1111”.

Example

```
call caoCtrl.Execute("ChangePassword", Array("0000","1111")
```

3.2.14. CaoController::Execute ("InputPseudoConsole ") command

This is a pseudo input command that mimics the functionality of the remote control console.

Syntax InputPseudoConsole (<bstrConsoleInputCode>)

bstrConsoleInputCode	:	[in] Remote control console input code (VT_BSTR) FN:FUNCTION button ES:ESCAPE button TG:TRG button SC:SCREEN button VI:VIEW button MN:MENU button EN:ENTER button UP:ENTER button Upward direction DN:ENTER button Downward direction

LT : ENTER button Leftward direction
RT : ENTER button Rightward direction
LU : ENTER button Upward and leftward direction
LD : ENTER button Downward and leftward direction
RU : ENTER button Upward and rightward direction
RD : ENTER button Downward and rightward direction
FU : FUNCTION＋ENTER button upward direction
FD : FUNCTION＋ENTER button downward direction
FL : FUNCTION＋ENTER Leftward direction
FR : FUNCTION＋ENTER Rightward direction
FLU : FUNCTION ＋ ENTER button Upward and leftward direction
FLD : FUNCTION ＋ ENTER button Downward and leftward direction
FRU : FUNCTION ＋ ENTER button Upward and rightward direction
FRD : FUNCTION＋ENTER button Downward and rightward direction
RS : Switching RUN/Program mode
FV : FUNCTION＋VIEW
FT : FUNCTION＋TRG
FM : FUNCTION＋MENU
FE : FUNCTION＋ENTER
FS : FUNCTION＋ESCAPE
SL : SCREEN＋ENTER Leftward direction
SR : SCREEN＋ENTER Rightward direction

Return value : none

The following sample program shows how to press a function button of a pseudo console.

Example

```
call caoCtrl.Execute("InputPseudoConsole", "FN")
```

3.2.15. CaoController::Execute ("SaveStatisticalData") command

Writes all the statistics data that have been saved in the Statistics menu to the SD card in the comma-delimited text format.

Syntax SaveStatisticalData (< bstrFolderDirectory>)

bstrFolderDirectory : [in] The destination directory of the SD card (VT_BSTR)

Return value : none

The following sample program shows how to write the statistic data in the [stat] folder with the text format.

Example

```
call caoCtrl.Execute("SaveStatisticalData", "/CV/stat"))
```

3.3. Change program settings

3.3.1. CaoController::Execute ("ChangeInspectSetting") command

Changes the setting to the inspection setting number of the specified SD card.

Syntax ChangeInspectSetting (< iDriveNo >, < iSettingNo >)

iDriveNo : [in] Specify an SD card number (VT_I4)

1 : SD1

2 : SD2

iSettingNo : [in] Specify an inspection setting number (0 to 999)(VT_I4)

Return value : none

Executing the following example will change the setting to the inspection setting number 1 of the SD card (SD1).

Example

```
call caoCtrl. Execute("ChangeInspectSetting", Array(1, 1))
```

3.3.2. CaoController::Execute ("ChangeInspectSettingAsync") command (not recommended)

Changes the setting to the inspection setting number of the specified SD card asynchronously.

To obtain and check the return value of the command, use GetCommandResult command. For details about GetCommandResult, please refer to 3.6.8.CaoController::Execute ("GetCommandResult") command.

When you use this command, be sure to disconnect the provider after the execution of GetCommandResult command.

Otherwise, an error may occur at asynchronous operation.

Syntax ChangeInspectSettingAsync (< iDriveNo >, < iSettingNo >)

iDriveNo	:	[in] Specify an SD card number. (VT_I4) 1 : SD1 2 : SD2
iSettingNo	:	[in] Specify an inspection setting number. (0 to 999)(VT_I4)
Return value	:	none

Executing the following example will change the setting to the inspection setting number 1 of the SD card (SD1).

Example

```
Dim vntResult as variant

call caoCtrl.Execute("ChangeInspectSettingAsync", Array(1, 1))

' Obtain the return value of ChangeInspectionSettingAsync command
vntResult = caoCtrl.Execute("GetCommandResult")

' vntResult : Return value (Empty)
```

3.3.3. CaoController::Execute ("ReadInspectSetting") command

Reads the inspection setting number.

Syntax ReadInspectSetting ()

Argument	:	[in] none
Return value	:	[out] < iDriveNo >, < iSettingNo > (VT_I4 | VT_ARRAY) iDriveNo : SD card number iSettingNo : Inspection setting number

Executing the following example will read the current inspection number.

Example

```
Dim vntRet as Variant
vntRet = caoCtrl. Execute("ReadInspectSetting")

' vntRet(0) : SD card number
' vntRet(1) : Inspection setting number
```

3.3.4. CaoController::Execute ("ReadToolParameter") command

Reads the upper and lower limits of the specified window

Syntax ReadToolParameter (<bstrWindow>, <bstrKind>, <bstrLimit>)

bstrWindow	:	[in] Specify a desired window (VT_BSTR)

Measurement window: W000 to W127
Calculation window : C000 to C127

bstrKind : [in] Limit type (VT_BSTR)
For details, refer to the User's manual of KEYENCE.

bstrLimit : [in] Specify upper/lower limit (VT_BSTR)
HL : Upper limit
LL : Lower limit

dblMargin : [out] Limit values (VT_R8)

Executing the following example will read the lower limit of X coordinates (256. 030) in the measurement window 005 (pattern search)

Example

```
Dim dblMargin as Double
dblMargin = caoCtrl. Execute("ReadToolParameter", Array("W005", "X", "LL"))

dblMargin : 256. 030
```

3.3.5. CaoController::Execute ("ChangeToolParameter") command

Sets the upper and lower limit of the specified window.

Syntax ChangeToolParameter (<bstrWindow>, <bstrKind>, <bstrLimit>, <bstrMargin>)

bstrWindow : [in] Specify a desired window (VT_BSTR)
Measurement window : W000 to W127
Calculation window : C000 to C127

bstrKind : [in] Limit type (VT_BSTR)
For details, refer to the User's manual of KEYENCE.

bstrLimit : [in] Specify upper/lower limit (VT_BSTR)
HL : Upper limit
LL : Lower limit

bstrMargin : [in] Limit values (VT_BSTR)

Return value : none

Executing the following example will change the lower limit of the calculation window 010 to -142. 214.

Example

```
caoCtrl. Execute "ChangeToolParameter", Array("C010", "MS", "LL", "-142. 214")
```

3.3.6. CaoController::Execute ("ReadBinaryData") command

You can specify the upper and lower limit on the binary filter of the specified window.

Syntax ReadBinaryData (<lWindowNo>)

lWindowNo	:	[in] Window number (VT_I4) 0 to 127
Return value	:	[out] Upper limit and lower limit(VT_I4 | VT_ARRAY) (max, min) max : Upper limit of the binary data (0 to 255) min : Lower limit of the binary data (0 to 255)

Executing the following example will read the upper and lower limit on the binary filter of the window 3.

Example

```
Dim vntParams as Variant
vntParams = caoCtrl. Execute("ReadBinaryData",  3)

vntParams(0) : 255
vntParams(1) : 125
```

3.3.7. CaoController::Execute ("ChangeBinaryData") command

You can specify the upper and lower limit on the binary filter of the specified window.

Syntax ChangeBinaryData (<lWindowNo>, <lMax>, <lMin>)

lWindowNo	:	[in] Window number (VT_I4) 0 to 127
lMax	:	[in] Upper limit of the binary data (0 to 255)
lMin	:	[in] Lower limit of the binary data (0 to 255)
Return value	:	[out] none

Executing the following example will set the upper limit to 200 and the lower limit to 100 on the binary filter of window 3.

Example

```
caoCtrl. Execute "ChangeBinaryData",  Array(3,  200, 100)
```

3.3.8. CaoController::Execute ("InitCommandMemory") command

Sets all of the current command memory values as the initial values for command memory.

Syntax InitCommandMemory ()

Argument : [in] none

Return value : [out] none

Example

```
caoCtrl. Execute "InitCommandMemory"
```

3.3.9. CaoController::Execute ("ReadCommandMemory") command

Reads data of the specified command memory

Syntax ReadCommandMemory (<lCommandMemNo>)

lCommandMemNo : [in] Command memory number. (VT_I4)
0 to 999

Return value : [out] Data to be read out (VT_I4)
-2147483648 to 2147483647

Example

```
Dim lParam as Long
lParam = caoCtrl. Execute("ReadCommandMemory", 4)
```

3.3.10. CaoController::Execute ("ChangeCommandMemory") command

Sets data into a maximum of 32 pieces of specified command memory

Syntax ChangeCommandMemory(<lCommandMemNo1>, <lParam1> [, lCommandMemNo2, lParam2 . . . , lCommandMemNon, lParamn])

lCommandMemNon : [in] Command memory number. (VT_I4)
0 to 999

lParamn : [in] Data to be written in (VT_I4)
-2147483648 to 2147483647

Return value : [out] none

Make sure to set both command memory number and writing data as pairs.

Executing the following example will set the value for the command memory 000 to1, and the value for the command memory 100 to -1000.

Example

```
Dim lParam as Long
lParam = caoCtrl. Execute("ChangeCommandMemory", Array(0, 1, 100, -1000))
```

3.3.11. CaoController::Execute ("RefreshReferencePosition") command

Recalculates the base reference values using the currently registered images.

Syntax RefreshReferencePosition()

Argument : [in] none

Return value : [out] none

Example

```
caoCtrl. Execute "RefreshReferencePosition"
```

3.3.12. CaoController::Execute ("RefreshReferencePositionAsync") command (not recommended)

Recalculates the base reference values using the currently registered images asynchronously.

To obtain and check the return value of the command, use GetCommandResult command. For details about GetCommandResult, please refer to 3.6.8.CaoController::Execute ("GetCommandResult") command.

When you use this command, be sure to disconnect the provider after the execution of GetCommandResult command.

Otherwise, an error may occur at asynchronous operation.

Syntax RefreshReferencePositionAsync()

Argument : [in] none

Return value : [out] none

The following shows an example.

Example

```
Dim vntResult as variant

caoCtrl.Execute "RefreshReferencePositionAsync"

' Obtain the return value of RefreshReferencePositionAsync command
vntResult = caoCtrl.Execute("GetCommandResult")

' vntResult : Return value (Empty)
```

3.4. Input/Output control

3.4.1. CaoController::Execute ("EnableTrigger") command

Enables/Disables trigger input

Syntax EnableTrigger(< iMode >)

iMode	:	[in] Enable/Disable a trigger (VT_I2)
		0 : Disable a trigger
		1 : Enable a trigger
Return value	:	none

Executing the following example will disable the trigger input.

Example

```
caoCtrl. Execute "EnableTrigger",  0
```

3.5. Camera control setting

3.5.1. CaoController::Execute ("ChangeShutterSpeed") command

Changes the shutter speed of the specified camera.

Syntax ChangeShutterSpeed (<lCameraNo>, < lShutterSpeed>)

lCameraNo	:	[in] Camera No..(VT_I4)
		1 to 4
lShutterSpeed		[in] Shutter speed (VT_I4)
		0:1/15
		1:1/30
		2:1/60
		3:1/120
		4:1/240
		5:1/500
		6:1/1000
		7:1/2000
		8:1/5000
		9:1/10000
		10:1/20000
Return value	:	none

The following sample program shows how to set the shutter speed of Camer No.1 to 1/30.

Example

```
call caoCtrl.Execute("ChangeShutterSpeed", Array(1,1))
```

3.5.2. CaoController::Execute ("ChangeCameraSensitivity") command

Changes the sensitivity of specified camera.

One tenth (1/10) of entered value is specified as a sensitivity of the camera.

Syntax ChangeCameraSensitivity (<lCameraNo>, <lSensitivity>)

lCameraNo	:	[in] Camera No..(VT_I4) 1 to 4
lSensitivity		[in] Sensitivity (VT_I4) 10 to 90
Return value	:	none

The following sample program shows how to set the sensitivity of camera number 1 to 1.0.

Example

```
call caoCtrl.Execute("ChangeCameraSensitivity", Array(1,10))
```

3.5.3. CaoController::Execute ("ChangeTriggerDelay") command

Sets the amount of time (ms) to delay after the trigger input has been receive to when the actual image acquisition begins.

Syntax ChangeTriggerDelay (<lCameraNo>, <lDelay>)

lCameraNo	:	[in] Camera No..(VT_I4) 1 to 4
lDelay		[in] delay time (VT_I4) 0 to 999 (ms)
Return value	:	none

The following sample program shows how to change the delay time of Camera number 1 trigger input to 100ms.

Example

```
call caoCtrl.Execute("ChangeTriggerDelay", Array(1,100))
```

3.5.4. CaoController::Execute ("ChangeLightIntensityLevel") command

Changes the intensity level of the specified light.

Syntax ChangeLightIntensityLevel (<lLightingNo>, <lLightingVolume>)

lLightingNo	:	[in] Light No.(VT_I4) 1 to 8
lLightingVolume		[in] Light intensity value (VT_I4) 0 to 255
Return value	:	none

The following sample program shows how to change the light intensity of Light number 1 to 50.

Example

```
call caoCtrl.Execute("ChangeLightIntensityLevel", Array(1,50))
```

3.5.5. CaoController::Execute ("ChangePatternCounter") command

Changes the pattern counter of the multi pattern mode

Syntax ChangePatternCounter (<lCounter>)

lCounter	:	[in] Counter value (VT_I4) 0 to 3
Return value	:	none

The following sample program shows how to change the pattern counter of multi pattern mode to 1.

Example

```
call caoCtrl.Execute("ChangePatternCounter", 1)
```

3.6. Original command

3.6.1. CaoController::Execute ("ExecuteCommand") command

Executes specified Non-procedure command. Command response will be obtained regardless of the command execution result.

For about supported non-procedure commands, refer to the CV series user's manual.

Syntax [<vntRet> =]ExecuteCommand(< bstrCommand >)

bstrCommand	:	[in] Return character string of command (VT_BSTR)
vntRet	:	[out] Return the command response (VT_BSTR)

Executing the following example will specify the Non-procedure command and switch CV to Run mode.

Example

```
Dim strRet as string
strRet = caoCtrl. Execute("ExecuteCommand","R0")
```

3.6.2. CaoController::Execute ("ExecuteCommandAsync") command (not recommended)

Executes specified non-procedure command asynchronously.

For about supported non-procedure commands, refer to the CV series user's manual

To obtain and check the return value of the command, use GetCommandResult command. For details about GetCommandResult, please refer to 3.6.8.CaoController::Execute ("GetCommandResult") command.

When you use this command, be sure to disconnect the provider after the execution of GetCommandResult command.

Otherwise, an error may occur at asynchronous operation.

Syntax ExecuteCommandAsync(< bstrCommand >)

bstrCommand : [in] Specify character strings of command (VT_BSTR)

Return value : [out] none

Executing the following example will specify the Non-procedure command and switch CV to Run mode.

Example

```
Dim vntResult as variant

Call caoCtrl.Execute("ExecuteCommandAsync", "R0")

' Obtain the return value of ExecuteCommandAsync command
vntResult = caoCtrl.Execute("GetCommandResult")

' vntResult : Return value of non-procedure command (BSTR)
```

3.6.3. CaoController::Execute ("TriggerAndGetResult") command

Receives a result after the trigger execution. If no output result is received, wait until the timeout-period passes. The output result includes the response packet of trigger.

Example)

If x=10, y=11 are output as the execution result of T1, the return values will be "T1, +10, +11".

Syntax [<strRet>] = TriggerAndGetResult(< iTriggerNo >)

iTriggerNo : [in] Trigger number 1 to 2 (VT_I2)

strRet : [out] Output result (VT_BSTR)

Example

```
Dim strRet as string
strRet = caoCtrl. Execute("TriggerAndGetResult",  1)
```

3.6.4. CaoController::Execute ("RecievePacket") command

Receives packets. If any packets have already been stored in the receiving buffer, packets in the receiving buffer will be obtained.

Syntax RecievePacket ()

Argument : [in] none

Return value : [out] Receiving packet (VT_BSTR)

The following is an example when this command is used together with a trigger command

Example

```
Dim strRet as String

Call caoCtrl. Execute("Trigger",  1)
strRet = caoCtrl. Execute("RecievePacket")
```

3.6.5. CaoController::Execute ("ClearPacket") command

Deletes packets stored in the receiving buffer

Syntax ClearPacket( )

Argument : none

Return value : [out] none

Example

```
Call caoCtrl. Execute("ClearPacket")
```

3.6.6. CaoController::Execute ("SetTimeout") command

Sets a timeout period

Syntax SetTimeout(<iTimeout>)

iTimeout : [in] Timeout period (msec) (VT_UI4)

Return value : [out] none

Example

```
Call caoCtrl. Execute("SetTimeout", 1000)
```

3.6.7. CaoController::Execute ("GetTimeout") command

Gets the currently assigned timeout period.

Syntax <uiTimeout> = GetTimeout()

Argument : [in] none

<uiTimeout> : [out] Timeout value (msec) (VT_UI4)

Example

```
Dim timeout as long
timeout = caoCtrl. Execute("GetTimeout")
```

3.6.8. CaoController::Execute ("GetCommandResult") command

Waits for the completion of the asynchronous command to get the return value of it.

If the executed asynchronous command which has not return value (e.g,; ChangeModeAsync) is executed, it returns nothing.

If any synchronous command is used before this command, "Get result error" (0x80100003) occurs and no value will be returned.

If an asynchronous command, which is the target of GetCommandResult command, ends with an error, the error will be ignored within the process of asynchronous command, and the error occurs at GetCommandResult command execution.

If there is no response within the specified timeout-period during the waiting time of the asynchronous command completion, a time-out error (0x80000900) will occur.

Note that if another command is executed after an asynchronous command, the execution result of the asynchronous command that you've just get will be deleted.

Syntax [<vntRet> =]GetCommandResult ()

Argument : none

vntRet : [out] Return value of asynchronous command (VT_VARIANT)

The following shows how to obtain the return value of asynchronous inspection.

Example

```
Dim vntResult as variant
```

```
            caoCtrl. Execute "ReStartAsync"
            vntResult = caoCtrl. Execute("GetCommandResult")
```