DENSO

DENSO Robotics

THIRD PARTY PRODUCTS

PROVIDER MANUAL

Maker

KEYENCE

Products／Series

Machine Vision System

MODEL:CV Series

Vision

Introduction

This document is a user's manual for the provider to use "KEYENCE Machine Vision System CV Series" connected to the DENSO robot controller RC8 series.

Caution: (1) Note that the functions and performance cannot be guaranteed if this product is used without observing instructions in this manual.
(2) All products and company names mentioned are trademarks or registered trademarks of their respective holders.

This document targets the following models in CV series. (as of June, 2014)

KEYENCE CV-3000 Series / CV-5000 Series

In this document, the above models are called CV series.

Important

To ensure proper and safe operation, be sure to read "Safety Precautions Manual" before using the provider.

Notice to Customers

1. Risks associated with using this product

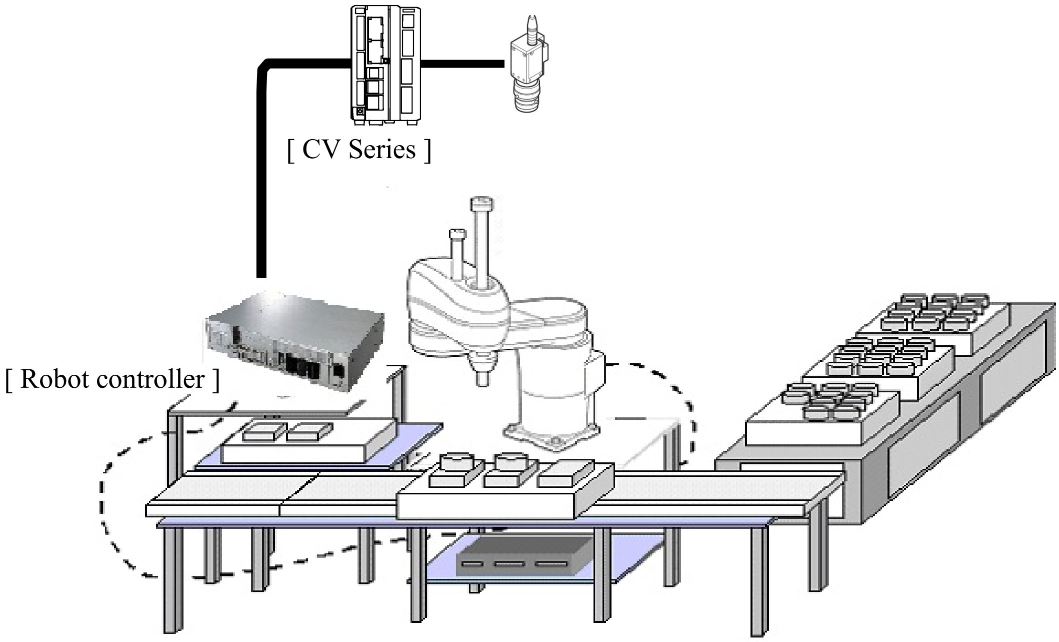
The user of this product shall be responsible for embedding and using the product (software) on a system and any result from using it.

Contents

1. Outline of This Product (Provider)

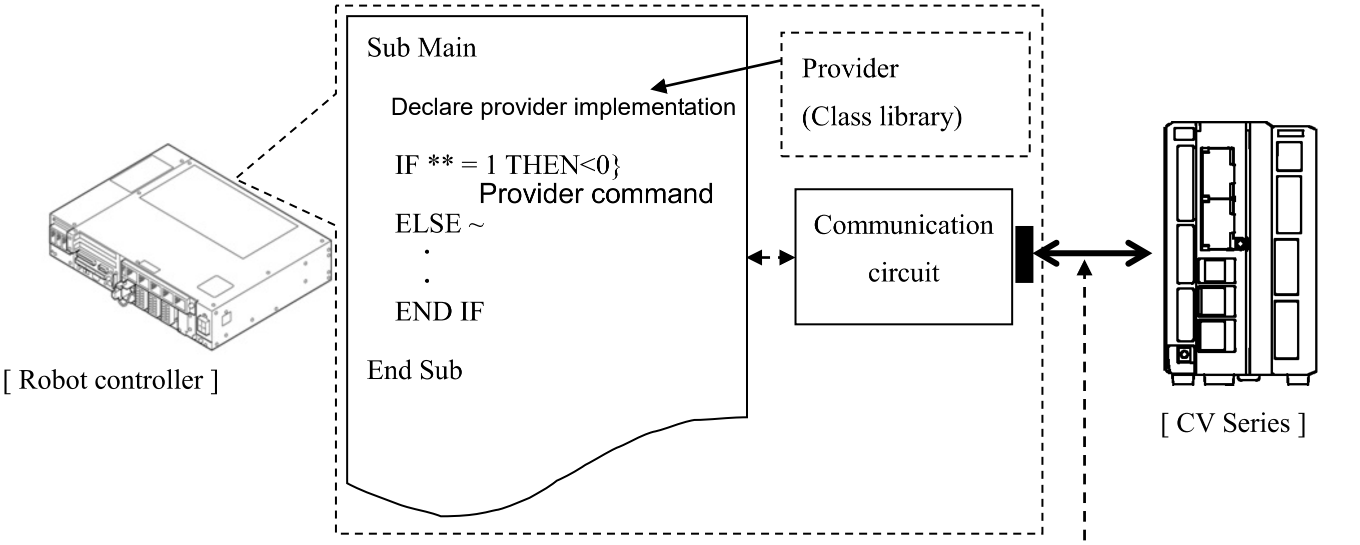
1.1. Target device of provider

This provider can be used only when a DENSO robot controller (RC8 series) is connected to the CV Series.



1.2. Features of provider

This provider is provided to use the CV Series native commands required to access CV Series in the robot program. Use of this provider allows customers to establish communication with a robot easily without creating a communication program for CV Series. The following shows a diagram of provider embedding.



```
Sub Main
  Declare provider implementation
  IF ** = 1 THEN<0}
        Provider command
  ELSE ~
    ・
    ・
  END IF
End Sub
```

Using the dedicated communication format for CV Series, data transmission is performed.

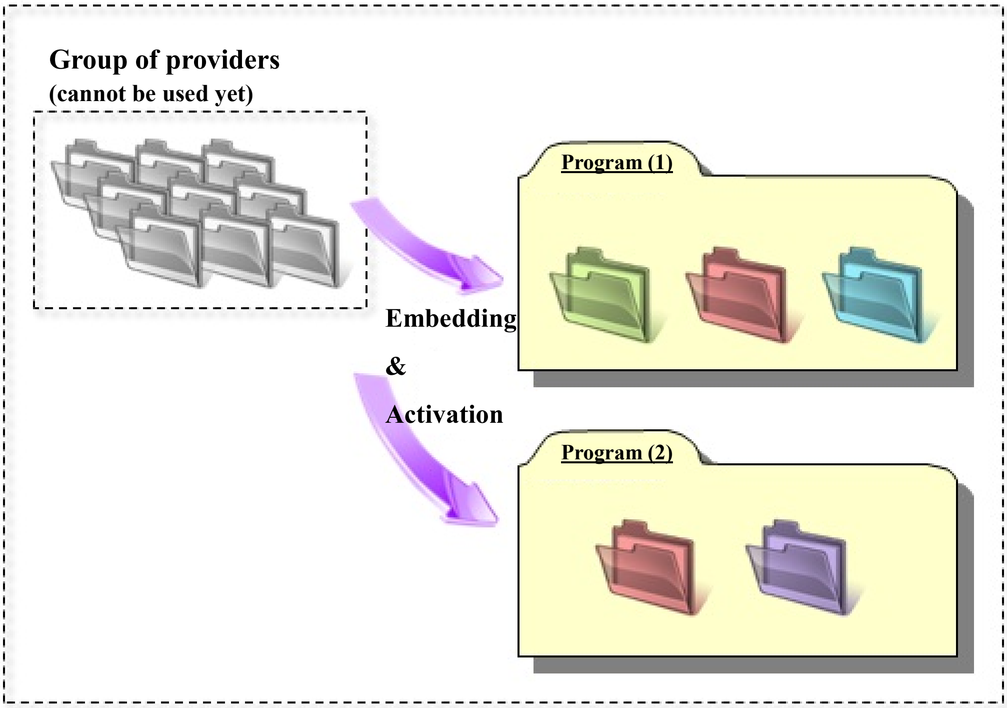
1.3. Mechanism of provider

This provider offers various programs required to control the target device as a single provider. Just activate the license to use the provider. Once provider implementation is declared on a desired program file, the functions prepared by the provider can be used as commands in the user program. Since the provider is included in the controller, there is no need of installation. Also, it is possible to implement multiple providers of different type. Note that a program (procedure) cannot contain the providers of the same type.

If provider is compared to a folder,

the folder contains various commands.

Controller internal image



📁 **Provider prepared in the system. This cannot be used yet.**

🟩🟥🟦🟪 **Provider after embedding. This can be used in a provider-embedded program. Different colors are used to indicate the provider type.**

Note: When the same provider exists in different programs like 🟥 in the above figure, exclusion process is required between the programs (tasks).

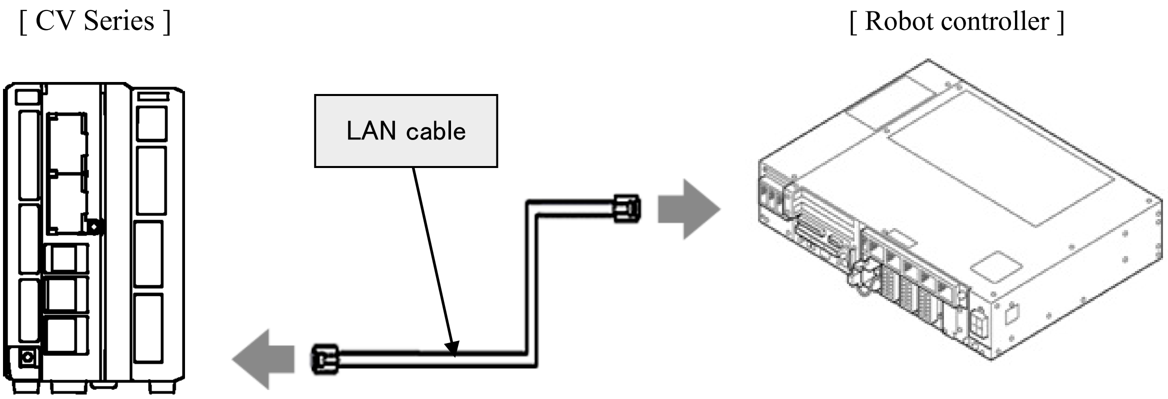
* The provider is provided as a dynamic link library (abbreviated as DLL) which can be used from PacScript.

2. How to Connect

You can use either Ethernet or RS232C for connection between a robot controller and CV series. When establishing a connection, use a cable compatible with the communication specification you use. For detailed information about each communication cable, refer to the CV Series User's Manual of KEYENCE.

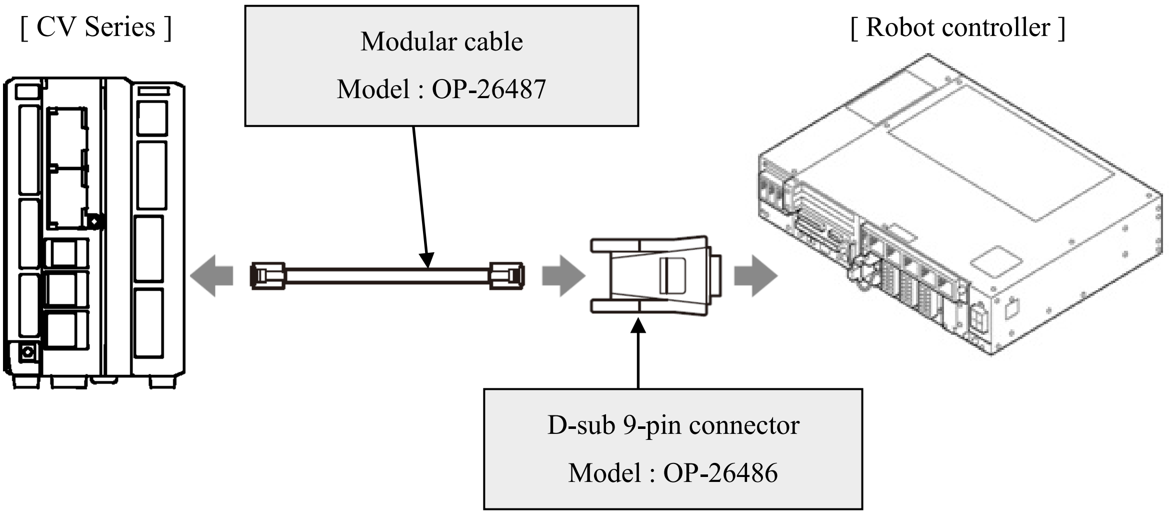
2.1. Ethernet Connection Example

To connect to the robot controller via Ethernet, use a crossover LAN cable. Also, when a switching hub/router is used, use the cable suitable for the switching hub/router specifications.



2.2. RS232C Connection Example

When you establish a connection with RS232C, use a Modular cable and D-sub 9-pin connector. Both of them are sold by KEYENCE as optional parts. There are two types of connectors though, use a D-sub 9-pin connector since the RS232C connector mounted in the robot controller is D-sub 9-pin.



3. Communication settings

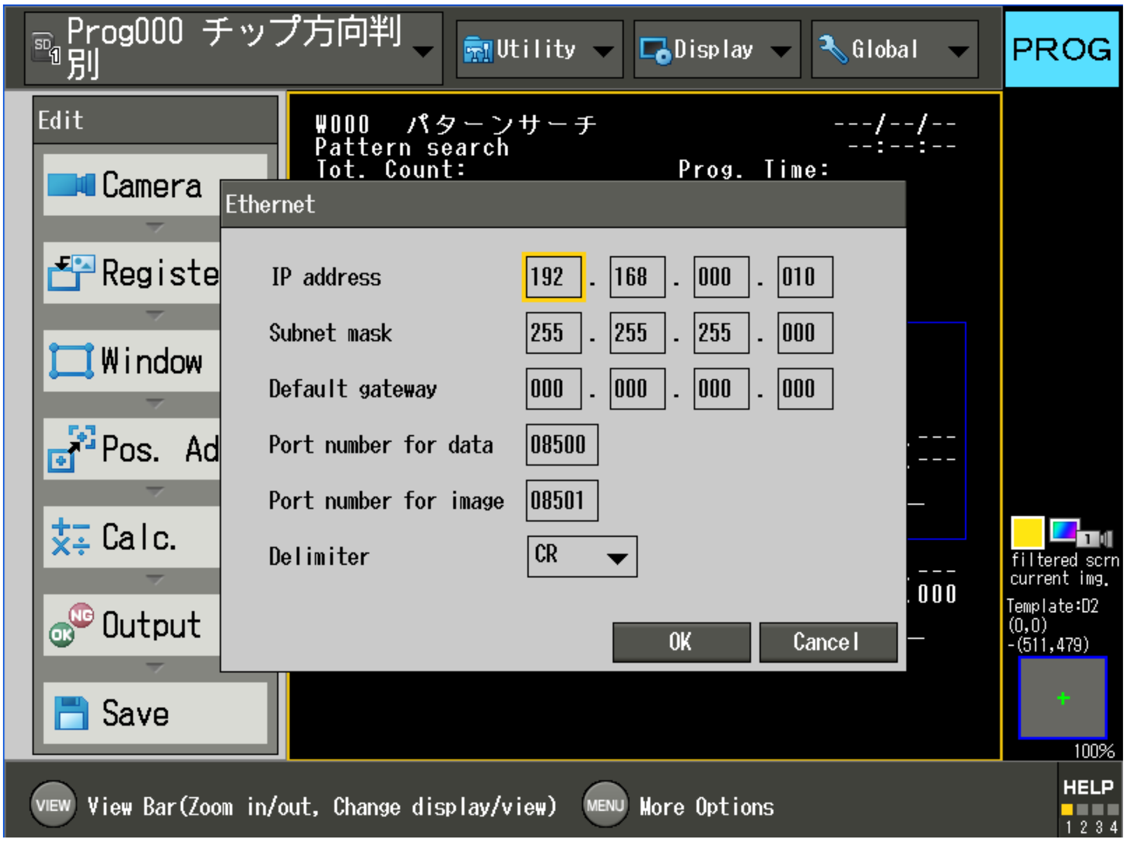
3.1. Setup for Ethernet connection

3.1.1. Communication setting for CV Series

Communication settings for CV series are carried out by manipulating a setting window displayed in the monitor (sold separately) plugged in the CV series main unit, by means of a console that comes with CV series. For details, refer to the CV Series User's Manual of KEYENCE.
Delimiter must be set to "CR" always.

This setting example shows when CV-3000 series is used.
From the CV-3000 setting window, click [Global]-[Ethernet] to display the following Ethernet window.

- When you set IP address and subnet mask, make sure that these of the robot controller and CV series are in the same subnet mask. In this example, IP address and the subnet mask are 192.168.0.10 and 255.255.255.0., respectively.
- Set a gateway, if necessary. In this example, 0.0.0.0 is set.
- Set desired port numbers to the data port and the image output port. The port number specified here will be the port number that is specified at the robot controller's Cao.AddController command execution as an option. In this example, the data port number and the image output port number are set to 08500 and 08501, respectively.
- Delimiter must be set to "CR" always.



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

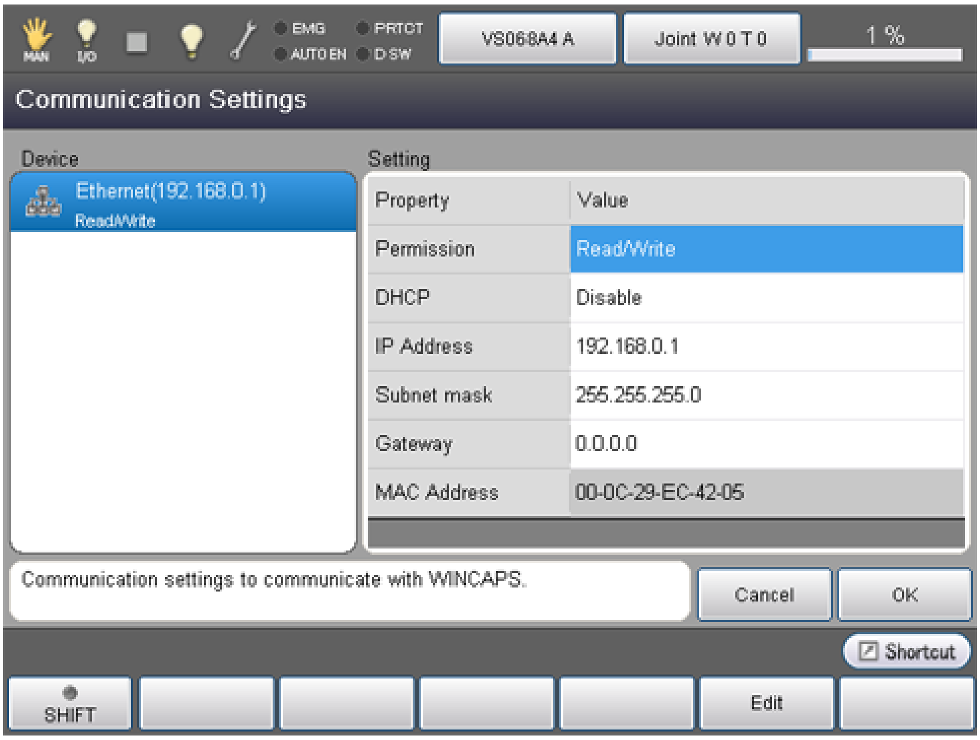
3.1.2. Communication setting for Robot controller

To set Ethernet communication setting for a robot controller, you can use a teach pendant or a mini-pendant. For detailed information about setting, refer to the followings on the DENSO ROBOT USER MANUALS.

Device	Referenced
Teach pendant	"Displaying and Changing Communication Settings Screen" of the TEACH PENDANT OPERATION GUIDE
Mini-pendant	"Setting DHCP" and "Setting IP Address" of the MINI PENDANT OPERATION GUIDE

This example shows the way of communication setting with a teach pendant.
From the top screen of a teach pendant, press [F6 Setting] - [F5 Communication and Token] - [F2 Network and Permission] to display the [Communication Settings] window.

- [Permission] has no relation to CV series communication.
- Once DHCP is enabled, IP address will automatically set. (Note that DHCP server may connect to the same network.) This example select "Disable".
- If you set DHCP to "Disable", make sure that IP addresses and subnet masks of the robot controller and CV series are the same. In this example, IP address and the subnet mask are 192.168.0.1 and 255.255.255.0, respectively.
- Set a gateway, if necessary. In this example, 0.0.0.0 is set.



Communication Settings

Device

Ethernet(192.168.0.1)
Read/Write

Setting

Property	Value
Permission	Read/Write
DHCP	Disable
IP Address	192.168.0.1
Subnet mask	255.255.255.0
Gateway	0.0.0.0
MAC Address	00-0C-29-EC-42-05

Communication settings to communicate with WINCAPS.

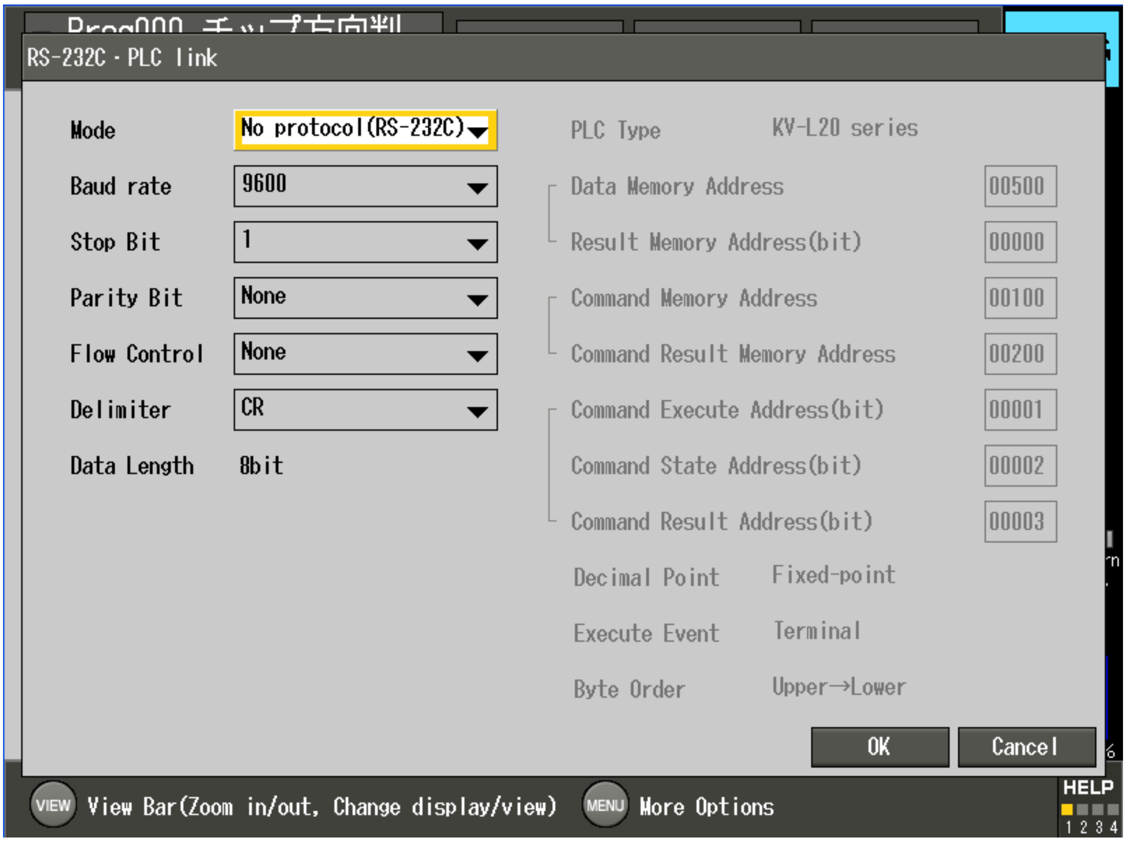
3.2. Setup for RS232C connection

3.2.1. Communication setting for CV Series

RS232C communication setting for CV series is carried out by manipulating a setting window displayed in the monitor (sold separately) plugged in the CV series main unit by means of a console that comes with CV series. For details, refer to the CV Series User's Manual of KEYENCE.
Note that the following items must be the same settings always.

Item	Setting
Mode	No protocol(RS-232C)
Flow Control	None
Delimiter	CR

This setting example shows when CV-3000 series is used.
From the CV-3000 setting window, click [Global]-[RS-232C･PLC link] to display the following window.
You can set arbitrary values, except for items on the table above.



RS-232C・PLC link

Mode: No protocol(RS-232C)
Baud rate: 9600
Stop Bit: 1
Parity Bit: None
Flow Control: None
Delimiter: CR
Data Length: 8bit

PLC Type: KV-L20 series
Data Memory Address: 00500
Result Memory Address(bit): 00000
Command Memory Address: 00100
Command Result Memory Address: 00200
Command Execute Address(bit): 00001
Command State Address(bit): 00002
Command Result Address(bit): 00003
Decimal Point: Fixed-point
Execute Event: Terminal
Byte Order: Upper→Lower

3.2.2. Communication setting for Robot controller

RS232C communication setting for the robot controller is carried out at the Cao.AddController command execution, by specifying an option parameter. Set an appropriate option according to the communication setting of RS232C on the CV series.
You can carry out the RS232C communication setup with a teach pendant or a mini-pendant; however, these are for Comm.Open command execution, so not applicable to this provider.

4. Provider Execution Procedure

The basic process of the provider is implementation (declaration) -> execution. This provider takes a connection process at the time of implementation. The operation can be repeated as many times as needed. A program example is shown below.

```
Sub Main

    On Error Goto ErrorProc      (1)                    'Declare error process routine
    Dim caoCtrl as Object        (2)                    'Declare provider variable
    Dim vntResult as Variant     (3)                    'Declare result acquisition variable

    caoCtrl = Cao.AddController("CV", "CaoProv.KEYENCE.CV", "", "conn=eth:192.168.0.10")    (4)

    "State from trigger to data receiving process"   (5)

EndProc:
    'End process
    Exit Sub

ErrorProc:
    'Error process

End Sub
```

(1) Declare the provider error processing routine as needed. (Connection error detection at declaration)

(2) Declare the provider implementation variable as Object type. The variable name can be specified arbitrarily.

(3) Declare the result acquisition variable. The data type depends on the command.

(4) Execute implementation with the provider declaration command cao.AddController. The parameters required for settings vary by provider. From this point the provider commands are available using the implementation variable caoCtrl.

(5) Now the program can be stated using the provider commands.

5. Command Description

This page contains a description of commands.

Commands are classified the following three types.

- Connection commands
- CV series-supported command
- Proprietary extension commands

A CV series-supported command is the command that is paired with a CV series command. The correspondence between the CV series commands and the CV series-supported commands is shown in the command list on the next page.

For the detailed operation of CV Series commands, refer to the CV Series User's manual of KEYENCE.

Table 5-1 Command list

CV series provider command	CV series command	Description	Page
Connection commands			
Cao.AddController	—	Implements the provider to a variable and makes a connection to CV series.	14
CV series-supported command			
Trigger	T1、T2	Input trigger. The response result can be obtained after the trigger input.	17
ChangeMode	R0、S0	Changes the operation mode to run or stop modes.	18
ChangeModeAsync		Changes the operation mode to run or stop modes asynchronously.	18
Reset	RS	Reset an item.	19
StoreSetting	SS	Save the data of the currently selected program setting number.	19
ChangeDisplayPattern	DS	Change the display pattern.	20
ChangeCurrentUnit	UW	Switch to the specified window number.	20
ReadCurrentUnit	UR	Obtain a currently activated window number.	21
ReoutputResult	M0	Obtain the latest measurement result.	21
RegistImage	BS	Registers an image.	22
GetRegistImageList	BL	Obtains the list of registered images.	23
SaveAllImage	BT	Reads all of the image data.	24
GetLibraryList	DL	Obtains the list of libraries.	25
ChangePassword	PS	Changes a password.	26
InputPseudoConsole	KY	Performs a console pseudo input.	27
SaveStatisticalData	ST	Saves the statistics data.	28
ChangeInspectSetting	PW	Change the setting to the inspection setting number of the specified SD card.	29
ChangeInspectSettingAsync		Change the setting to the inspection setting number of the specified SD card asynchronously.	30
ReadInspectSetting	PR	Obtain currently selected inspection setting number and its SD card number.	31
ReadToolParameter	DR	Obtain an upper or lower limit of the specified window.	32
ChangeToolParameter	DW	Set an upper or lower limit of the specified window.	33
ReadBinaryData	JR	Obtain the upper and lower limits on the binarization filter of the specified window.	34
ChangeBinaryData	JW	Set the upper and lower limits on the binarization filter of the specified window.	35
InitCommandMemory	MI	Set all of the current command memory values as the initial values for command memory.	35
ReadCommandMemory	MR	Obtain data of the specified command memory.	36
ChangeCommandMemory	MW	Set data into a maximum of 32 pieces of command memory.	36
RefreshReferencePosition	RR	Recalculate the base reference values using the currently registered images.	37
RefreshReferencePositionAsync		Recalculate the base reference values using the currently registered images asynchronously.	37
EnableTrigger	TE	Enable or disable trigger input.	38

ChangeShutterSpeed	CW	Changes the shutter speed.	39
ChangeCameraSensitivity	CW	Changes the camera sensitivity.	40
ChangeTriggerDelay	CW	Sets the amount of time to delay after the trigger input.	40
ChangeLightIntensityLevel	CW	Changes the light intensity level value.	41
ChangePatternCounter	CW	Changes the pattern counter of multi pattern mode.	41
Proprietary extension commands			
ExecuteCommand	—	Execute a CV series command with a syntax of CV series command.	42
ExecuteCommandAsync	—	Execute a CV series command with a syntax of CV series command asynchronously.	43
TriggerAndGetResult	—	Obtain a result after trigger execution.	44
RecievePacket	—	Obtain the result of trigger input.	45
ClearPacket	—	Delete result data stored in a robot controller.	46
SetTimeout	—	Set a time-out period.	46
GetTimeout	—	Obtain a currently assigned time-out period.	47
GetCommandResult	—	Wait for the completion of the asynchronous command to get the return value of it.	48

Cao.AddController

Usage Implements the provider to a variable and makes a connection to CV series.

Syntax Cao.AddController(<Controller name>,<Provider name>,
< Provider running machine name>,<Option>)

Argument <Controller name>

Assign a name (The name is used for control) (character string).

<Provider name>

Specify "CaoProv.KEYENCE.CV" with character string type data.

< Provider running machine name>

Specify "" with character string type data.

<Option>

Specify following items with character string type data.

Syntax "Conn=<Connection parameter>,Timeout=<Time>"

Argument <Connection parameter>

This differ from communication methods. Refer to "Description for parameters of each connection".

<Time>

Set an allowable waiting time given to the response from CV series at this provider's command execution by millisecond-unit. This is optional. This should be 500 milliseconds if it is omitted.

Description for parameters of each connection

For Ethernet

Syntax "eth:<IP address>:<Port number>"

Argument <IP address>

Specify IP address of CV series to connect.

<Port number>

Specify port number of CV series to connect. This is optional. This should be 8500 if it is omitted.

For RS232C

Syntax com:<COM Port>:<BaudRate>:<Parity>
:<DataBits>:<StopBits>:<Flow>

Argument <COM Port>

Specify a COM port number of a robot controller plugged in the CV series. Entered number will be the COM port number. For example, if you enter 1, it indicates COM1 is specified. If you use a serial communication connector on the front side of the controller while expansion RS232C communication module is not used, enter 2 in this parameter.

<BaudRate>

According to the communication speed of CV series to connect, select suitable baud rate from 4800, 9600, 19200, 38400, 57600, 115200 (bps).

This is optional. This should be "9600" if it is omitted.

<Parity>

According to the CV series to connect, select suitable parity from the followings.

N : None
E : Even parity
O : Odd parity

This is optional. This should be "N" if it is omitted.

<DataBits>

According to the data bit count of CV series to connect, select suitable number from the followings.

7 : 7 bits
8 : 8 bits

This is optional. This should be "8" if it is omitted.

<StopBits>

According to the stop bit count of CV series to connect, select suitable number from the followings.

1 : 1 bit
2 : 2 bits

This is optional. This should be "1" if it is omitted.

<Flow>

The flow control selection is prepared as shown below. However, to communicate with CV series, set this parameter to "0: Without flow control".

0 : Without flow control
1 : Xon / Xoff
2 : Hardware control

This is optional. This should be "0 " if it is omitted.

Return value

Implemented objects are returned (Object).

Description

The provider becomes effective when implemented to a variable. From this point the implemented Object type variable is used to access the provider. (The implemented variable is called "Implementation Variable".)

Example

```
Dim caoCtrl as Object

'=================   For Ethernet   ===================
'To specify a time-out period, but not specify COM port
caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV", _
                                        "", "conn=eth:192.168.0.10, timeout=1000")

"To specify a time-out period and COM port
caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV", _
                                        "", "conn=eth:192.168.0.10:8503")

'=================   For RS232C   ====================
'To omit a baud rate and the following.
caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV", _
                                        "", "conn= com:2")

'To specify a baud rate and the followings.
caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV", _
                                        "", "conn= com:2:115200:E:8:1:0, timeout=1000")

'To specify a baud rate and the following.( Specify parity but omit others )
caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV", _
                                        "", "conn= com:2::E:::")
```

<ImplVar>.Trigger

Usage Input trigger. The response result can be obtained after the trigger input.

Syntax <ImplVar>.Trigger (<Trigger No.> , [<Mode>])

Argument <Trigger No.>

Specify a trigger number (integer).

1	:	Trigger 1
2	:	Trigger 2

< Mode.>

Specify the result flag (integer). This is optional. This should be "1" if it is omitted.

0	:	Issue a trigger. Not receive the response result.
1	:	Issue a trigger. Receive the response result.

Return value Result strings (This effective only when "Mode = 1")

Description Input trigger. The response result can be obtained after the trigger input.
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 caoCtrl as Object
Dim strRet as String

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'To receive a result at the same time.
'Issue a trigger & receive the response
strRet = caoCtrl. Trigger(1)
'strRet: "+11, +12"

'To receive a result later
'Issue a trigger only.
caoCtrl. Trigger 1, 0

'Receive the result
strRet = caoCtrl.RecievePacket
'strRet: "T1, +11, +12"
```

<ImplVar>.ChangeMode

Usage Changes the operation mode to run or stop modes.

Syntax <ImplVar>.ChangeMode <Mode>

Argument <Mode>

Specify a desired mode (integer).

0 ： Stop mode

1 ： Run mode

Return value None

Description Changes the operation mode to run or stop modes.

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                    "conn=eth:192.168.0.10")

'Switch the mode to Run mode.
caoCtrl.ChangeMode 1
```

<ImplVar>.ChangeModeAsync

Usage Changes the operation mode to run or stop modes asynchronously.

Syntax <ImplVar>.ChangeModeAsync <Mode>

Argument <Mode>

Specify a desired mode (integer).

0 ： Stop mode

1 ： Run mode

Return value None

Description Changes the operation mode to run or stop modes asynchronously.
To obtain and check the return value of the command, use GetCommandResult command.

Example

```
Dim caoCtrl as Object
Dim vntResult as variant

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                    "conn=eth:192.168.0.10")

'Switch the mode to Run mode.
caoCtrl.ChangeModeAsync 1

'Obtain the return value of ChangeMoveAsync command
vntResult = caoCtrl.GetCommandResult
```

<ImplVar>.Reset

Usage Reset an item.

Syntax <ImplVar>.Reset

Argument None

Return value None

Description Reset an item.

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                          "conn=eth:192.168.0.10")

'Reset.
caoCtrl.Reset
```

<ImplVar>.StoreSetting

Usage Save the data of the currently selected program setting number.

Syntax <ImplVar>.StoreSetting

Argument None

Return value None

Description Save the data of the currently selected program setting number.

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                          "conn=eth:192.168.0.10")

'Save the data of the currently selected program setting number.
caoCtrl.StoreSetting
```

<ImplVar>.ChangeDisplayPattern

Usage Change the display pattern.

Syntax <ImplVar>.ChangeDisplayPattern <Desired display pattern>,<Parameter>

Argument <Desired display pattern>

Specify the desired display pattern with character string type data.

PT	:	Display template
RS	:	Result display
PG	:	Page
FC	:	Screen

<Parameter>

Subsidiary parameter according to the Desired display pattern selected above with character string type data. For details, refer to the DS command of the KEYENCE CV Series User's Manual.

Return value None

Description Change the display pattern.

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'Display a previous page.
caoCtrl.ChangeDisplayPattern "PG","P"
```

<ImplVar>.ChangeCurrentUnit

Usage Switch to the specified window number.

Syntax <ImplVar>.ChangeCurrentUnit <Window No.>

Argument <Window No.>

Specify a window number with an integer ranging from 0 to 127.

Return value None

Description Switch to the specified window number.

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'Change the window to the window number 2.
caoCtrl.ChangeCurrentUnit 2
```

<ImplVar>.ReadCurrentUnit

Usage Obtain a currently activated window number.

Syntax <ImplVar>.ReadCurrentUnit

Argument None

Return value <Window No.>

Currently activated window number is returned with integer type data ranging from 0 to 127.

Description Obtain a currently activated window number.

Example

```
Dim caoCtrl as Object
Dim iNum as Integer

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'Obtain a currently activated window number.
iNum = caoCtrl.ReadCurrentUnit
```

<ImplVar>.ReoutputResult

Usage Obtain the latest measurement result.

Syntax <ImplVar>.ReoutputResult

Argument None

Return value <Measurement result >

The latest measurement result is returned with character string type data.

Description Obtain the latest measurement result.

Example

```
Dim caoCtrl as Object
Dim bstrResult as String

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'Obtain the latest measurement result
bstrResult = caoCtrl.ReoutputResult
```

<ImplVar>.RegistImage

Usage Registers the latest captured image specified by camera No.

Syntax <ImplVar>.RegistImage <Camera No.>,<Regist image No.>

Argument <Camera No.>

Specify a camera number with integer type data ranging from 1 to 4.

< Regist image No.>

Specify a regist image number with integer type data ranging from 0 to 999.

Return value None

Description Registers the latest captured image specified by camera No.

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                                    "conn=eth:192.168.0.10")

'Register Camera No.1 image acquired at the latest into the Registered image No.0.
caoCtrl.RegistImage 1,0
```

<ImplVar>.GetRegistImageList

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

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

Syntax <ImplVar>.GetRegistImageList <Target>,<Camera No.>

Argument <Target>

0 ：Current setting

<Camera No.>

Specify a camera number with integer type data ranging from 1 to 4.

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

Syntax <ImplVar>.GetRegistImageList < Target >,<Regist image No.>, <Camera No.>

Argument < Target >

Specify a SD card number with integer type data.

1 ：SD1

2 ：SD2

< Regist image No.>

Specify a regist image number with integer type data ranging from 0 to 999.

<Camera No.>

Specify a camera number with integer type data ranging from 1 to 4.

Return value < The list of registered image numbers >

The list of registered image numbers are stored in an array of integer.

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

Example

```
Dim caoCtrl as Object
Dim vntRet as variant

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'Obtain a list of image numbers that are currently set.
vntRet = caoCtrl. GetRegistImageList 0,1
```

<ImplVar>.SaveAllImage

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

Syntax <ImplVar>.SaveAllImage <Camera No.>,< Compression ratio >,< Image kind >,< Folder name >

Argument <Camera No.>

Specify a camera number with integer type data ranging from 1 to 4.

<Compression ratio>

Specify a compression ratio with integer type data ranging from 0 to 3.

0	: No compression
1	: 1/2
2	: 1/4
3	: 1/8

<Image kind>

Specify an image kind with character string type data.

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.

<Folder name>

Enter any folder name with character string type data.

Return value None

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

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                "conn=eth:192.168.0.10")

'Save the all of the image in the image buffer of Camera No.1 with 1/2 compression ratio.
caoCtrl.SaveAllImage 1,1,"AL","ImageFolder"
```

<ImplVar>.GetLibraryList

Usage Obtains the list of library numbers that have set.

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

Syntax <ImplVar>.GetLibrabyList <Target>

Argument <Target>

0 ：Current setting

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

Syntax <ImplVar>.GetLibrabyList<Target>, < Program No. of library number to obtain.>

Argument <Target>

Specify a SD card number with integer type data.

1 ：SD1

2 ：SD2

< Program No. of library number to obtain.>

Specify a program No. of library number to obtain with integer type data ranging from 0 to 999.

Return value < Library number list >

The list of library numbers are stored in an array of integer.

Description Obtains the list of library numbers that have set.

Example

```
Dim caoCtrl as Object
Dim vntRet as variant

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                  "conn=eth:192.168.0.10")

'Obtain the list of library number used in the current program number.
vntRet = caoCtrl. GetLibraryList 0
```

<ImplVar>.ChangePassword

Usage Changes the password.

Syntax <ImplVar>.ChangePassword < Old password >,< New password >

Argument <Old password>

Enter the old password with integer type data.

<New password>

Specify a new password with integer type data ranging from 0000 to 9999.

Return value None

Description Changes the password.

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'Change the password from “0000” to “1111”.
caoCtrl.ChangePassword 0000,1111
```

<ImplVar>.InputPseudoConsole

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

Syntax <ImplVar>.InputPseudoConsole < Remote control console input code >

Argument < Remote control console input code >

Specify a remote control console input code with character string type data.

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 Upward direction
FD	: FUNCTION ＋ ENTER 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

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

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                             "conn=eth:192.168.0.10")

'Press a function button of a pseudo console.
caoCtrl.InputPseudoConsole "FN"
```

<ImplVar>.SaveStatisticalData

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

Syntax <ImplVar>.SaveStatisticalData < The destination directory of the SD card >

Argument < The destination directory of the SD card >

Specify the destination directory of the SD card with character string type data.

Return value None

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

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                             "conn=eth:192.168.0.10")

'Write the statistics data in the [stat] folder with the text format.
caoCtrl.SaveStatisticalData "/CV/stat"
```

<ImplVar>.ChangeInspectSetting

Usage Change the setting to the inspection setting number of the specified SD card.

Syntax <ImplVar>.ChangeInspectSetting <SD card number>,<Inspection setting number>

Argument <SD card number>

Specify an SD card number with integer type data.

1 ： SD1

2 ： SD2

<Inspection setting number>

Specify an inspection setting number with integer type data ranging from 0 to 999.

Return value None

Description Change the setting to the inspection setting number of the specified SD card.

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'Change the setting to the inspection setting number 1 of the SD1.
caoCtrl.ChangeInspectSetting 1,1
```

<ImplVar>.ChangeInspectSettingAsync

Usage Change the setting to the inspection setting number of the specified SD card asynchronously.

Syntax <ImplVar>.ChangeInspectSettingAsync <SD card number>,<Inspection setting number>

Argument <SD card number>

Specify an SD card number with integer type data.

1 ： SD1

2 ： SD2

<Inspection setting number>

Specify an inspection setting number with integer type data ranging from 0 to 999.

Return value None

Description Change 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.

Example

```
Dim caoCtrl as Object
Dim vntResult as variant

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'Change the setting to the inspection setting number 1 of the SD1.
caoCtrl.ChangeInspectSettingAsync 1,1

'Obtain the return value of ChangeInspectionSettingAsync command
vntResult = caoCtrl.GetCommandResult
```

<ImplVar>.ReadInspectSetting

Usage Obtain currently selected inspection setting number and its SD card number.

Syntax <ImplVar>.ReadInspectSetting

Argument None

Return value The following two items are stored in an array of integer.

<SD card number>

Currently selected SD card number

1 ： SD1

2 ： SD2

<Inspection setting number>

Currently selected inspection setting number.

Description Obtain currently selected inspection setting number and its SD card number.

Example

```
Dim caoCtrl as Object
Dim vntRet as Variant
Dim iaryData(1) as Integer

caoCtrl=Cao.AddController("CV","CaoProv.KEYENCE.CV","", _
                                          "conn=eth:192.168.0.10")

'Obtain currently selected inspection setting number and
'its SD card number.
'iaryData(0) stores an SD card number.
'iaryData(1) stores an inspection setting number.
vntRet = caoCtrl.ReadInspectSetting
iaryData(0) = vntRet(0)
iaryData(1) = vntRet(1)
```

<ImplVar>.ReadToolParameter

Usage Obtain an upper or lower limit of the specified window.

Syntax <ImplVar>.ReadToolParameter (<Window>,<Limit type>,<Upper / lower limit>)

Argument <Window>

Specify a desired window with character string type data.

Wnnn : Specify a measurement window with character string type data. The "nnn" part contains a number ranging from 000 to 127. Specify desired character string from W000 to W127.

Cnnn : Specify a calculation window with character string type data. The "nnn" part contains a number ranging from 000 to 127. Specify desired character string from C000 to C127.

<Limit type>

Specify a limit type with character string type data. For details, refer to the DR command of the KEYENCE CV Series User's Manual.

<Upper / lower limit>

Specify upper / lower limit with character string type data.

HL : Upper limit

LL : Lower limit

Return value <Limit values>

Limit values (setting values) specified by an argument will be returned with a double precision real number .

Description Obtain an upper or lower limit of the specified window.

Example

```
Dim caoCtrl as Object
Dim dblMargin as Double

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'Obtain a lower limit of X-coordinates in the measurement window 005.
'( The measurement window 005 has been set in the pattern search ).
dblMargin = caoCtrl.ReadToolParameter("W005","X","LL")
```

<ImplVar>.ChangeToolParameter

Usage Set an upper or lower limit of the specified window.

Syntax <ImplVar>.ChangeToolParameter <Window>,<Limit type>,
<Upper / lower limit>,<Limit values>

Argument <Window>

Specify a desired window (character string) .

Wnnn : Specify a measurement window with character string type data. The "nnn" part contains a number ranging from 000 to 127. Specify desired character string from W000 to W127.

Cnnn : Specify a calculation window with character string type data. The "nnn" part contains a number ranging from 000 to 127. Specify desired character string from C000 to C127.

<Limit type>

Specify a limit type with character string type data. For details, refer to the DW command of the KEYENCE CV Series User's Manual.

<Upper / lower limit>

Specify upper / lower limit (character string) .

HL : Upper limit

LL : Lower limit

<Limit values>

Specify limit values with character string type data.

Return value None

Description Set an upper or lower limit of the specified window.

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                    "conn=eth:192.168.0.10")

'Set a lower limit of the calculation window 010 to -142.214.
caoCtrl.ChangeToolParameter "C010","MS","LL","-142.214"
```

<ImplVar>.ReadBinaryData

Usage Obtain the upper and lower limits on the binarization filter of the specified window.

Syntax <ImplVar>.ReadBinaryData (<Window No.>)

Argument <Window No.>

Specify a desired window with integer type data ranging from 0 to 127.

Return value The following two items are stored in an array of integer.

<Upper limit of the binary data>

The upper limit of the binarization filter.

<Lower limit of the binary data>

The lower limit of the binarization filter.

Description Obtain the upper and lower limits on the binarization filter of the specified window.

Example

```
Dim caoCtrl as Object
Dim vntRet as Variant
Dim iaryParams(1) as Integer

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'Obtain the upper and lower limits on the binarization filter of the window 3.
'iaryParams(0) stores the upper limit.
'iaryParams(1) stores the lower limit.
vntRet = caoCtrl.ReadBinaryData(3)
iaryParams(0) = vntRet(0)
iaryParams(1) = vntRet(1)
```

<ImplVar>.ChangeBinaryData

Usage Set the upper and lower limits on the binarization filter of the specified window.

Syntax <ImplVar>.ChangeBinaryData <Window No.>,<Upper limit of the binary data>,
<Lower limit of the binary data>

Argument <Window No.>

Specify a desired window with integer type data ranging from 0 to 127.

<Upper limit of the binary data>

Specify a upper limit of the binary data with integer type data ranging from 0 to 255.

<Lower limit of the binary data>

Specify a lower limit of the binary data with integer type data ranging from 0 to 255.

Return value None

Description Set the upper and lower limits on the binarization filter of the specified window.

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

''Set the upper limit to 200 and the lower limit to 100
'on the binarization filter of window 3.
caoCtrl.ChangeBinaryData 3,200,100
```

<ImplVar>.InitCommandMemory

Usage Set all of the current command memory values as the initial values for command memory.

Syntax <ImplVar>.InitCommandMemory

Argument None

Return value None

Description Set all of the current command memory values as the initial values for command memory.

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'Set all of the current command memory values as the initial values
'for command memory.
caoCtrl.InitCommandMemory
```

<ImplVar>.ReadCommandMemory

Usage Obtain data of the specified command memory.

Syntax <ImplVar>.ReadCommandMemory (<Command memory No.>)

Argument <Command memory No.>

Specify a command memory number with integer type data ranging from 0 to 999.

Return value <Command memory data>

Command memory data is returned with integer type data.

Description Obtain data of the specified command memory.

Example

```
Dim caoCtrl as Object
Dim iParam as Integer

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'Obtain data of the command memory 4.
iParam = caoCtrl.ReadCommandMemory(4)
```

<ImplVar>.ChangeCommandMemory

Usage Set data into a maximum of 32 pieces of command memory.

Syntax <ImplVar>.ChangeCommandMemory <Command memory No.>,<Data>,
<Command memory No.>,<Data>,......

Argument Specify the following two arguments as a pair. Up to 32 pairs can be set.

<Command memory No.>

Specify a command memory number with integer type data ranging from 0 to 999.

<Data>

Specify a data with integer type data ranging from -2147483648 to 2147483647.

Return value None

Description Set data into a maximum of 32 pieces of specified command memory.

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'Set the value for the command memory 000 to 1,
'and the value for the command memory 100 to -1000.
caoCtrl.ChangeCommandMemory 0,1,100,-1000
```

<ImplVar>.RefreshReferencePosition

Usage Recalculate the base reference values using the currently registered images.

Syntax <ImplVar>.RefreshReferencePosition

Argument None

Return value None

Description Recalculate the base reference values using the currently registered images.

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                    "conn=eth:192.168.0.10")

'Recalculate the base reference values using the currently registered images.
caoCtrl.RefreshReferencePosition
```

<ImplVar>.RefreshReferencePositionAsync

Usage Recalculate the base reference values using the currently registered images asynchronously.

Syntax <ImplVar>.RefreshReferencePositionAsync

Argument None

Return value None

Description Recalculate the base reference values using the currently registered images asynchronously.
To obtain and check the return value of the command, use GetCommandResult command.

Example

```
Dim caoCtrl as Object
Dim vntResult as variant

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                    "conn=eth:192.168.0.10")

'Recalculate the base reference values using the currently registered images.
caoCtrl.RefreshReferencePositionAsync

'Obtain the return value of RefreshReferencePositionAsync command
vntResult = caoCtrl.GetCommandResult
```

<ImplVar>.EnableTrigger

Usage Enable or disable trigger input.

Syntax <ImplVar>.EnableTrigger <Enable / Disable>

Argument <Enable / Disable>

Set enable or disable trigger input with integer type data.

0 ： Disable trigger

1 ： Enable trigger

Return value None

Description Enable or disable trigger input.

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'Disable the trigger input.
caoCtrl.EnableTrigger 0
```

<ImplVar>.ChangeShutterSpeed

Usage Changes the shutter speed of the specified camera.

Syntax <ImplVar>.ChangeShutterSpeed <Camera No.>, < Shutter speed >

Argument <Camera No.>

Specify a camera number with integer type data ranging from 1 to 4.

< Shutter speed >

Specify a shutter speed with integer type data ranging from 0 to 10.

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

Description Changes the shutter speed of the specified camera.

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                              "conn=eth:192.168.0.10")

'Set the shutter speed of Camera No.1 to 1/30.
caoCtrl.ChangeShutterSpeed 1,1
```

<ImplVar>.ChangeCameraSensitivity

Usage Changes the sensitivity of specified camera.

Syntax <ImplVar>.ChangeCameraSensitivity <Camera No.>,< Sensitivity >

Argument <Camera No.>

Specify a camera number with integer type data ranging from 1 to 4.

< Sensitivity >

Specify the sensitivity of camera with integer type data ranging from 10 to 90.

Return value None

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

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'Set the sensitivity of camera number 1 to 1.0.
caoCtrl.ChangeCameraSensitivity 1,10
```

<ImplVar>.ChangeTriggerDelay

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

Syntax <ImplVar>.ChangeTriggerDelay <Camera No.>,< Delay time>

Argument <Camera No.>

Specify a camera number with integer type data ranging from 1 to 4.

<Delay time>

Specify the delay time with integer type data ranging from 0 to 999 [ms].

Return value None

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

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'Change the delay time of Camera number 1 trigger input to 100 [ms].
caoCtrl.ChangeTriggerDelay 1,100
```

<ImplVar>.ChangeLightIntensityLevel

Usage Changes the intensity level of the specified light.

Syntax <ImplVar>.ChangeLightIntensityLevel < Light number >,< Light number value >

Argument <Light number>

Specify a light number with integer type data ranging from 1 to 8.

<Light number value>

Specify the light number value with integer type data ranging from 0 to 255.

Return value None

Description Changes the intensity level of the specified light.

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                          "conn=eth:192.168.0.10")

'Change the light intensity of Light number 1 to 50.
caoCtrl.ChangeLightIntensityLevel 1,50
```

<ImplVar>.ChangePatternCounter

Usage Changes the pattern counter of the multi pattern mode.

Syntax <ImplVar>.ChangePatternCounter <Counter value>

Argument <Counter value>

Specify the counter value with integer type data ranging from 0 to 3.

Return value None

Description Changes the pattern counter of the multi pattern mode.

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                          "conn=eth:192.168.0.10")

'Change the pattern counter of multi pattern mode to 1.
caoCtrl.ChangePatternCounter 1
```

<ImplVar>.ExecuteCommand

Usage Execute a CV series command with a syntax of CV series command.

Syntax <ImplVar>.ExecuteCommand (<CV series command syntax>)

Argument <CV series command syntax>

Specify CV series command syntax with character string type data.

Return value <Execution result data of CV series command>

The return value is the execution result data of CV series command. The data is returned with character string type data.

Description Execute a CV series command with a syntax of CV series command. For detailed operation of CV Series commands, refer to the CV Series User's manual of KEYENCE.

Example

```
Dim caoCtrl as Object
Dim strRet as String

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'Change the display pattern to the Raw screen.
'If the command successfully finishes, strRet stores "DS".
'If the command fails, strRet stores "ER,DS,nn".
'( "nn" contains an error code. )
strRet = caoCtrl.ExecuteCommand("DS,PT,0")
```

<ImplVar>ExecuteCommandAsync

Usage Execute a CV series command with a syntax of CV series command asynchronously.

Syntax <ImplVar>.ExecuteCommandAsync <CV series command syntax>

Argument <CV series command syntax>

Specify CV series command syntax with character string type data.

Return value None

Description Execute a CV series command with a syntax of CV series command asynchronously. To get the command execution result and return value, use GetCommandResult command. To obtain and check the return value of the command, use GetCommandResult command.
For detailed operation of CV Series commands, refer to the CV Series User's manual of KEYENCE.

Example

```
Dim caoCtrl as Object
Dim vntResult as variant

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'Change the display pattern to the Raw screen.
'Obtain the return value of ExecuteCommandAsync command
caoCtrl.ExecuteCommandAsync "DS,PT,0"

'Obtain the return value of ExecuteCommandAsync command
vntResult = caoCtrl.GetCommandResult
```

<ImplVar>.TriggerAndGetResult

Usage Obtain a result after trigger execution.

Syntax <ImplVar>.TriggerAndGetResult (<Trigger No.>)

Argument <Trigger No.>

Specify a trigger number with integer type data.

1 ： Trigger 1

2 ： Trigger 2

Return value <Result data>

Result of a trigger execution is returned with character string type data.

Description Obtain the result after trigger execution.
Receive 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".

Example

```
Dim caoCtrl as Object
Dim strRet as String

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'Input trigger in Trigger 1 and then obtain the result.
strRet = caoCtrl.TriggerAndGetResult(1)
```

<ImplVar>.RecievePacket

Usage Obtain the result of trigger input.

Syntax <ImplVar>.RecievePacket

Argument None

Return value <Result data>

Result data generated by trigger input is received with character string type data.

Description Obtain result data generated by trigger input.

If the CV series is set so as to generate no result output against trigger input, no result data returns from CV series. As a result, an error is issued when a time-out period passes. (Time-out period is set by Cao.AddController command option, or SetTimeout command).

Also, after trigger input, if you input trigger one more time without executing ReceivePacket command, the result data for two of trigger inputs are stored in a robot controller. Under the condition if you execute the ReceivePacket command, the first trigger's result data will be returned.

Therefore, if the situation where the number of trigger input does not match with the number of ReceivePacket command execution occurs, delete the result data stored in the robot controller by executing ClearPacket command first. Then, input trigger again, and then execute ReceivePacket command to obtain result data.

Example

```
Dim caoCtrl as Object
Dim strRet as String

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                    "conn=eth:192.168.0.10")

'Input trigger in Trigger 1.
caoCtrl.Trigger 1

'Obtain the result data.
strRet = caoCtrl.RecievePacket
```

<ImplVar>.ClearPacket

Usage Delete result data stored in a robot controller.

Syntax <ImplVar>.ClearPacket

Argument None

Return value None

Description Delete result data stored in a robot controller

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                    "conn=eth:192.168.0.10")

'Delete result data
caoCtrl.ClearPacket
```

<ImplVar>.SetTimeout

Usage Set a time-out period.

Syntax <ImplVar>.SetTimeout <Time>

Argument <Time>

Set a time-out period with integer type data. Unit is millisecond.

Return value None

Description Basically, a time-out period is set at the Cao.AddController command execution. Use this command if you want to set a time-out period after Cao.AddController command execution.

Example

```
Dim caoCtrl as Object

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                    "conn=eth:192.168.0.10")

'Set a time-out period to 1000 milliseconds.
caoCtrl.SetTimeout 1000
```

<ImplVar>.GetTimeout

Usage Obtain a currently assigned time-out period.

Syntax <ImplVar>.GetTimeout

Argument None

Return value <Time>

Currently assigned time-out period is returned with integer type data. Unit is millisecond.

Description Obtain a currently assigned time-out period.

Example

```
Dim caoCtrl as Object
Dim iTimeout as Integer

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'Obtain a time-out period.
iTimeout = caoCtrl.GetTimeout
```

<ImplVar>.GetCommandResult

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

Syntax <ImplVar>.GetCommandResult

Argument None

Return value <Return value of asynchronous command>

The return value of asynchronous command is stored.

Description Wait 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.

Example

```
Dim caoCtrl as Object
Dim vntResult as variant

caoCtrl=Cao.AddController("CV"," CaoProv.KEYENCE.CV ", "", _
                                        "conn=eth:192.168.0.10")

'Change the display pattern to the Raw screen.
caoCtrl.ExecuteCommandAsync "DS,PT,0"

To obtain the return value of command, use GetCommandResult.
caoCtrl.ExecuteCommandAsync "DS,PT,0
vntResult = caoCtrl.GetCommandResult
```

6. Error Code

As for how to check the provider errors, refer to Provider Errors in PROVIDER GUIDE on the DENSO ROBOT USER MANUALS.
In provider errors, an error issued by CV series will have original number ranging from 80108000 (hexadecimal) to 80108063 (hexadecimal), which lower two digits represents an error code sending from CV series. For example, when ChangeCurrentUnit command execution, if you enter a value larger than 127 in the Window number, the original number of the robot controller's error will be 80108016 (hexadecimal). The lower two digits "16" (hexadecimal) is equal to "22" in decimal number. According to the explanation of UW command written in the CV Series User's Manual, the error code 22 stands for "Either the number, number of digits, or range of parameters is incorrect."

Error	Error number	Description
E_CVERROR_CVERR	0x80108000 to 0x80108063	CV series original error
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

7. Sample Program

```
Sub Main
        Dim caoCtrl As Object
        Dim strRet As String

        'CV series provider implementation
        caoCtrl = Cao.AddController("CV", "CaoProv.KEYENCE.CV", "", _
                                        "conn=eth:192.168.0.3, timeout=1000")

        'Input trigger in Trigger 1 and then obtain the result data.
        strRet = caoCtrl.TriggerAndGetResult(1)

        'Output the result data to the message output window on the teach pendant.
        PrintDbg strRet

        ' Disconnect CV series provider and delete it.
        cao.Cotrollers.Remove caoCtrl.Index
        caoCtrl = Nothing
End Sub
```

Revision History

DENSO Robot
Provider
User's Manual

KEYENCE Machine Vision System CV Series

Version	Supported RC8	Content
Ver.1.0.0	Ver.1.8.6	First version
Ver.1.0.1	Ver.1.13.0	Expanded arguments of Trigger command. Corrected TriggerAndGetResult command. Added Asynchronous commands (ChangeModeAsync, ChangeInspectionSettingAsync, RefreshReferencePositionAsync) Added GetCommandResult command. Updated the error code list.
Ver.1.0.2	Ver.1.13.0	Modified the sample program.
Ver.1.0.3	Ver.2.3.*	Modified version.
Ver.1.0.4	Ver.2.8.*	Added Synchronous commands (RegistImage, GetRegistImageList, SaveAllImage, GetLibraryList,ChangePassword, InputPseudoConsole, SaveStatisticalData, ChangeShutterSpeed, ChangeCameraSensitivity, ChangeTriggerDelay, ChangeLightIntensityLevel, ChangePatternCounter)

DENSO WAVE INCORPORATED

- No part of this manual may be duplicated or reproduced without permission.
- The contents of this manual are subject to change without notice.
- Every effort has been made to ensure that the information in this manual is accurate. However, should any unclear point, error or omission be found, please contact us.
- Please note that we will not be responsible for any effects resulted from the use of this manual regardless of the above clauses.

DENSO Robotics

THIRD PARTY PRODUCTS

DENSO WAVE INCORPORATED