

KEYENCE
LK-G3000LkIF providers
User's Guide
Version 1.0.0

June 10, 2020

Note: Do not change the settings in other applications while the provider is connected to the device.

© 2020 DENSO WAVE INCORPORATED

The copyright of this manual belongs to DENSO WAVE INCORPORATED.

The company name or the product name that has been described is a trademark or a registered trademark of each company.

The content on this user's manual may be changed without notice.

[Revision History]

Version	Date	Description
1.0.0	2020-06-10	First edition

[Operation Check Model]

Model	Version	Notes
LK-G3000	-	

[Compatible models]

Model
LK-G3000
LK-G3000P
LK-G3000V
LK-G3000PV

No part of this user's manual may be reproduced in any form without permission.

- The content of this user's manual are subject to be changed without notice.
- The contents of this manual have been prepared in a thorough manner. However, please contact us if you notice any questions, mistakes, or omissions.
- Note that we cannot be held responsible for the effects of the operation regardless of the above sections.

Contents

1. Introduction	7
1.1. Informative sources	8
2. Setting Up Your Environment for Application Development	9
2.1. Connecting LK-G3000 to Client-PC	9
2.2. Setting up a PC development environment	9
2.2.1. Installing LK-G3000LkIF Providers Manually	9
3. Command Reference	10
3.1. Method/Property List	10
3.2. Method properties	10
3.2.1. CaoWorkspace classes	10
3.2.1.1. AddController method	10
3.2.2. CaoController classes	11
3.2.2.1. VariableNames method	11
3.2.2.2. Variables Properties	11
3.2.2.3. AddVariable method	12
3.2.2.4. Execute method	12
3.2.3. CaoVariable classes	12
3.2.3.1. Value Properties	12
3.3. Extended command list	12
3.3.1. Mode change command	15
3.3.1.1. SetMode Commands	15
3.3.2. Measurement and Control-Related Commands	16
3.3.2.1. GetCalcData Commands	16
3.3.2.2. SetTiming Commands	17
3.3.2.3. SetZero Commands	17
3.3.2.4. SetReset Commands	18
3.3.2.5. SetPanelLock Commands	18
3.3.2.6. SetProgramNo Commands	18
3.3.2.7. GetProgramNo Commands	19
3.3.2.8. GetFigureData Commands	19
3.3.2.9. ClearFigureData Commands	22
3.3.2.10. StartDataStorage Commands	22
3.3.2.11. StopDataStorage Commands	22
3.3.2.12. ClearDataStorage Commands	23

3.3.2.13. GetDataStorageData Commands.....	23
3.3.2.14. GetDataStorageStatus Commands.....	24
3.3.2.15. GetLight Commands	24
3.3.3. Configuration-Related Commands	26
3.3.3.1. Panel Display-Related Commands	26
3.3.3.2. Commands Related to Tolerance Setting.....	26
3.3.3.3. Head setting-related commands	28
3.3.3.4. OUT setting-related commands.....	35
3.3.3.5. Common setting related commands	51
3.4. Variable list.....	56
3.4.1. CaoController class-variable	57
3.4.1.1. @MAKER_NAME	57
3.4.1.2. @VERSION	57
3.4.1.3. @CALCDATA	58
3.4.1.4. RECEIVED_WAVEFROM<??>.....	59
4. Programming by LK-G3000LkIF providers	61
4.1. Sample-programming to obtain OUT1 and OUT2 measurements	61
4.1.1. Sample program	61
4.1.1.1. Pre-processing	63
4.1.1.2. Obtain OUT1 and OUT2 readings	64
4.1.1.3. Post-processing.....	64
4.2. Sample programming for obtaining stored data in a data storage	65
4.2.1. Sample program	65
4.2.1.1. Pre-processing	67
4.2.1.2. Acquisition of accumulated data in data storage	67
4.2.1.3. Post-processing.....	68
5. LK-G3000LkIF Provider Error Codes	69
6. Appendix	70

1. Introduction

This manual is a user's guide for KEYENCE's laser displacement gauge LK-G3000 series. It uses the communication library "LkIF.dll" from KEYENCE to exchange data via USB communication. Fig. 1-1 shows the overall configuration of this provider and the device. The providers are referred to as LK-G3000LkIF providers.

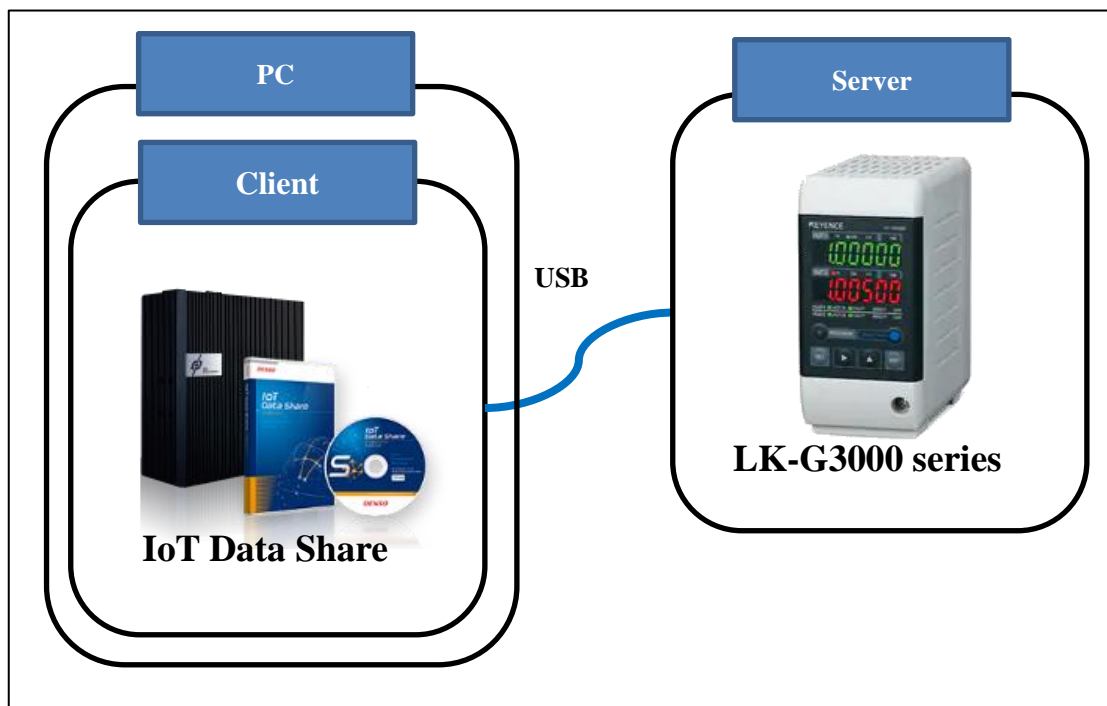


Fig. 1-1 Configuration Diagram

Fig. 1-2 shows the correspondence between this provider and each device.

(※An example. It does not represent everything.)

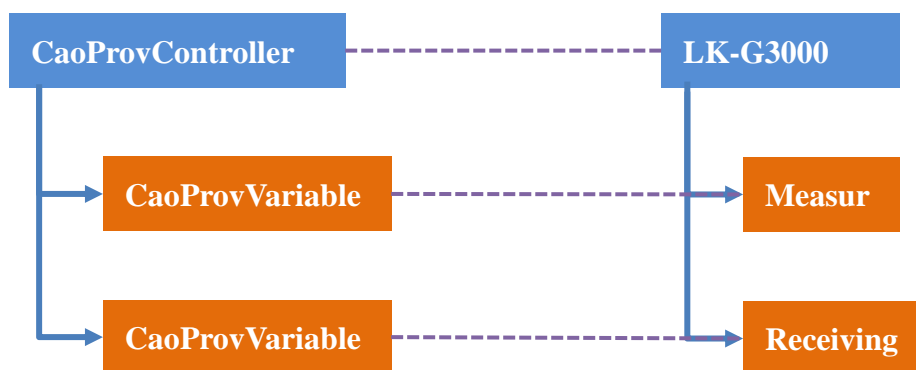


Fig. 1-2 Provider configuration and device information

1.1. Informative sources

LK-G3000LkIF providers refer to KEYENCE's "User's Manual for High-Speed, High-Precision CCD Laser Displacement Meter LK-G Series 96M12274" and "User's Manual for Setup and Support Software for LK-G Series 130074". This manual is referred to as LK-G3000 manual and LK-Navigator manual.

2. Setting Up Your Environment for Application Development

2.1. Connecting LK-G3000 to Client-PC

LK-G3000LkIF providers use KEYENCE communication libraries to communicate with LK-G3000 series. For more information on how to connect USB, see LK-Navigator documentation. In addition, KEYENCE communication library is included in the same Bin folder as the provider's dll. The following is a detailed description of KEYENCE Communication Library, which is the dependent modules that you want to use.

Table 2-1 Dependent Modules

DLL	Description
LkIF.dll	This is the DLL main unit.
KeyUsbDrv.dll	Required for DLL operation.

2.2. Setting up a PC development environment

When connecting LK-G3000 series to the client PC for the first time, install KEYENCE "LK-Navigator" USB driver. For more information about how to install, see LK-Navigator documentation.

2.2.1. Installing LK-G3000LkIF Providers Manually

If you install LK-G3000LkIF providers manually, you must register the registry as shown below. To register the registry, start the command prompt with administrator privileges and execute regsvr32 command. When executing the command, either move to the path where the file is located or specify the file path.

Table 2-2 Tabular LK-G3000LkIF providers

File name	CaoProvKEYENCELK-G3000LkIF.dll
ProgID	CaoProv.KEYENCE.LK-G3000LkIF
Registry registration	Regsvr32 CaoProvKEYENCELK-G3000LkIF.dll
Deletion of Registry Registration	Regsvr32 /u CaoProvKEYENCELK-G3000LkIF.dll

3. Command Reference

3.1. Method/Property List

Table 3-1 List of methods and properties

Category	Methods/Properties ¹	Function	See Also
CaoWorkspace			
	AddController	M Connected to controller	P.10
CaoController			
	VariableNames	M Get a list of variable names that can be connected	P.11
	Variables	P Retrieving Variable Collections Holded by the Controller	P.11
	AddVariable	M Adding Variable Objects	P.12
	Execute	M Execute Extended Commands	P.12
CaoVariable			
	Value	P Get/set value	P.12

3.2. Method properties

3.2.1. CaoWorkspace classes

3.2.1.1. AddController method

Add controller objects to CaoWorkspace. The following are the specifics of AddController method:

SYNOPSIS

AddController

```
(
    "<controller name>",           // Controller name (optional)
    "CaoProv.KEYENCE.LK-G3000LkIF", // Provider name (fixed)
    "<machine name>",             // Provider execution machine name (unused)
    "<Option>"                    // Option character string (optional)
)
```

Option

The following is an optional specification for Option character string: Option character string is a comma (,) string consisting of the options listed below.

¹ M: Indicates methods, P: properties, and E: events, respectively.

Option	Required	Description	Value Range	Default Value
Timeout	--	Response wait time	0 - 2147483647	500

Usage example

```

Dim caoEng As CaoEngine ' Engine
Dim caoWs As CaoWorkspace ' Workspace
Dim caoCtrl As CaoController ' Controller

' Generate CaoEngine
Set caoEng = new CaoEngine
' Generate CaoWorkspace
Set caoWs = caoEng.Workspaces.Item(0)
' Generate CaoController
Set caoCtrl = caoWs.AddController("LKG3000LkIF", _
                                "CaoProv.KEYENCE.LK-G3000LkIF", _
                                "", _
                                "Timeout=1000")

```

3.2.1.1.1. Notes on Connection

Even if more than one controller is added to LK-G3000LkIF provider, only one communication destination can be connected to the device with the lowest port number even if the device is connected to more than one LK-G3000 by USB. Communication is not performed when a controller is added. Timeout option is the response wait time from KEYENCE's communication library.

3.2.2. CaoController classes

3.2.2.1. VariableNames method

Gets a list of variable names that can be connected. The variable name obtained by this method can be used as the first argument of AddVariable method described later. AddVariable method

Usage example

```

' Get variable name list
Dim variableNames() As String
VariableNames = caoCtrl.variableNames

```

3.2.2.2. Variables Properties

Gets a collection of variables that the controller holds.

Usage example

```

' Variable Collection Retrieval
Dim variables As CaoVariables
Set variables = caoCtrl.variables
' Variable acquisition
Dim variable As CaoVariable
Set variable = variables.Item(0)

```

3.2.2.3. AddVariable method

Adds a variable object to CaoController. Only the variable names shown in 3.4.1 can be used.3.4.1

AddVariable is specified as follows.

SYNOPSIS

AddVariable

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

3.2.2.4. Execute method

Execute CaoController extension. In addition, LK-G3000LkIF providers communicate with devices when extended commands are executed. Extended commands that can be specified with Execute can only be used as shown in 3.3. List of Extended Commands. Execute is specified as follows.3.3

SYNOPSIS

Execute

```
(
    "<extension command name>",           // Extended command name
    "<Option string>"                   // Option character string (optional)
)
```

3.2.3. CaoVariable classes

3.2.3.1. Value Properties

Gets/sets the data from LK-G3000. In addition, LK-G3000LkIF providers communicate with the devices when they are acquired or set. The behavior depends on the variable name. For details, refer to section 3.4, Variable List.

3.3. Extended command list

Defines the list of extended commands that can be used. The usage examples are described in detail for each command.

Table 3-2 List of extended commands

Command	Description	See Also
Mode change command		

Command	Description	See Also
SetMode	Sets the operation mode of the main unit.	P.15
Measurement and Control-Related Commands		
GetCalcData	Gets the measured value.	P.16
SetTiming	Sets ON/OFF of the timing.	P.17
SetZero	Sets the autozero ON/OFF.	P.17
SetReset	Set the reset.	P.18
SetPanelLock	Sets the panel lock.	P.18
SetProgramNo	Switches the program number.	P.18
GetProgramNo	Gets the program number.	P.19
GetFigureData	Get statistics.	P.19
ClearFigureData	Clears the statistics.	P.22
StartDataStorage	Start data storage.	P.22
StopDataStorage	Stop the data storage.	P.22
ClearDataStorage	Clears the accumulated data in the data storage.	P.23
GetDataStorageData	Retrieves accumulated data for data storage.	P.23
GetDataStorageStatus	Acquires the accumulated state of data storage.	P.24
GetLight	Obtains the received light waveform.	P.24
Configuration-Related Commands		
Panel Display-Related Commands		
SetPanel	Switches the panel display.	P.26
GetPanel	Gets the panel display.	P.26
Commands Related to Tolerance Setting		
SetTolerance	Sets the tolerance.	P.26
GetTolerance	Gets the tolerance.	P.27
Head setting-related commands		
SetAbleMode	Sets ABLE tuning mode.	P.28
GetAbleMode	Gets ABLE tuning mode.	P.29
SetAbleMinMax	Sets ABLE control area.	P.29
GetAbleMinMax	Gets ABLE control scope.	P.30

Command	Description	See Also
SetMeasureMode	Set the measurement mode.	P.30
GetMeasureMode	Used to acquire the measurement mode.	P.31
SetNumAlarm	Set the number of alarm processing times.	P.32
GetNumAlarm	Acquires the number of alarm processing operations.	P.32
SetAlarmLevel	Sets the alarm level.	P.32
GetAlarmLevel	Gets the alarm level.	P.33
StartABLE	Starts ABLE tuning.	P.33
StopABLE	Finish ABLE tuning.	P.34
CancelABLE	Cancels ABLE tuning.	P.34
SetReflectionMode	Sets the installation mode.	P.34
GetReflectionMode	Gets the installation mode.	P.35
OUT setting-related commands		
SetCalcMethod	Sets the calculation method.	P.35
GetCalcMethod	Gets the calculation method.	P.39
SetScaling	Set the scaling.	P.40
GetScaling	Get the scaling.	P.41
SetFilterMode	Sets the filter mode.	P.41
GetFilterMode	Gets the filter mode.	P.42
SetAverage	Sets the average count.	P.42
GetAverage	Gets the average count.	P.43
SetCutOffFrequency	Sets the cutoff frequency when the filter mode is set to low-pass filter or high-pass filter.	P.43
GetCutOffFrequency	Used to acquire the cutoff frequency when the filter mode is set to low-pass filter or high-pass filter.	P.44
SetTriggerMode	Sets or queries the trigger mode.	P.44
GetTriggerMode	Gets the trigger mode.	P.45
SetOffset	Set the offset.	P.46
GetOffset	Gets the offset.	P.46
SetAnalogScaling	Sets the analog output scaling.	P.47
GetAnalogScaling	Obtains the analog output scaling.	P.47
SetCalcMode	Sets the measurement mode.	P.48

Command	Description	See Also
GetCalcMode	Gets the measurement mode.	P.49
SetDisplayUnit	Set the minimum display unit.	P.49
GetDisplayUnit	Gets the smallest display unit.	P.50
SetAnalogThrough	Sets the analog slew.	P.50
GetAnalogThrough	Acquires an analog through.	P.51
Common setting related commands		
SetDataStorage	Set the target OUT, number of accumulated points, and accumulation period for data storage.	P.51
GetDataStorage	Retrieves the target OUT, number of accumulated points, and accumulation period for data storage.	P.52
SetSamplingCycle	Sets the sampling period.	P.53
GetSamplingCycle	Acquires the sampling period.	P.53
SetMutualInterPrev	Set mutual interference prevention.	P.53
GetMutualInterPrev	Acquires mutual interference prevention.	P.54
SetTimingSync	Sets the timing synchronization.	P.54
GetTimingSync	Gets the timing synchronization.	P.55
SetTolCompOutputFormat	Sets the judgment output format.	P.55
GetTolCompOutputFormat	Acquires the judgment output format.	P.55
SetStrobeTime	Sets the strobe time.	P.56
GetStrobeTime	Gets the strobe time.	P.56

3.3.1. Mode change command

3.3.1.1. SetMode Commands

Changes the operation mode of the main unit. However, in the communication library, you do not need to explicitly execute this command to change the operation mode automatically. The following arguments are provided.

Item	Type	Description
Argument	VT_I4	Specifies the operation mode. Specify one of the following values. <ul style="list-style-type: none"> • 0-Operation mode • 1-Setting mode

Usage example

```
// Execute SetMode
```

```
Call caoCtrl.Execute("SetMode", 0)
```

3.3.2. Measurement and Control-Related Commands

3.3.2.1. GetCalcData Commands

Gets the measured value. The return values are shown below.

Item	Type Description		
Return Value	VT_ARRAY VT_VARIANT		
	0	VT_ARRAY VT_VARIANT	OUT1 Measured value
	0	VT_I4	Whether the data is valid or not. One of the following is obtained: <ul style="list-style-type: none"> • 0-Valid data • 1-+ RANGE OVER • 2--Range over • 3-Waiting for judgment
	1	VT_R4	The measured value. This value is invalid if it is not valid data.
	1	VT_ARRAY VT_VARIANT	OUT2 measured value
	0	VT_I4	Whether the data is valid or not. Retrieves values similar to the measurements in OUT1 above.
	1	VT_R4	The measured value. This value is invalid if it is not valid data.

Usage example

```
' Execute GetCalcData
```

```
Dim values As Variant
```

```
Values = caoCtrl.Execute("GetcalcData")
```

```
If Not IsEmpty(values) Then
```

```
    ' OUT1 readings
```

```
    Dim value1 As Variant
```

```
    Value1 = values(0)
```

```
    ' Valid data or not
```

```
    Dim validData1 As Long
```

```
    ValidData1 = value1(0)
```

```
    ' Measured value
```

```
    Dim fValue1 As Single
```

```
    fValue1 = value1(1)
```

```
    ' OUT2 readings
```

```
    Dim value2 As Variant
```

```
    Value2 = values(1)
```

```
'Whether the data is valid or not
```

```
Dim validData2 As Long
```

```
ValidData2 = value2(0)
```

```
' Measured value
```

```
Dim fValue2 As Single
```

```
fValue2 = value2(1)
```

```
End If
```

3.3.2.2. SetTiming Commands

Set the timing signal input. When the timing input is set, the measured value at the time of setting is retained. The function at the time of timing input differs according to the measurement mode. For more information on timing input, refer to the timing chart in "Chapter 3 FUNCTION SETTING-SETTING UP MEASUREMENT-USING THE HOLD FUNCTION (MEASUREMENT MODE)" in LK-G3000 manual. The following arguments are provided.

Item	Type Description		
Argument	VT_ARRAY VT_VARIANT		
	0	VT_I4	Specify the OUT number to be set. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2 • 2 - OUT1 + OUT2
	1	VT_BOOL	Specify timing input. Specify ON if TRUE, OFF if FALSE.

Usage example

```
'Execute SetTiming
```

```
Dim param As Variant
```

```
Param = Array(0, True)
```

```
Call caoCtrl.Execute("SetTiming", param)
```

3.3.2.3. SetZero Commands

Sets the auto zero. When Auto Zero is set to ON, the measured value during measurement is set to zero. The following arguments are provided.

Item	Type Description		
Argument	VT_ARRAY VT_VARIANT		
	0	VT_I4	Specify the OUT number to be set. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2 • 2 - OUT1 + OUT2
	1	VT_BOOL	Specify auto zero. Specify ON if TRUE, OFF if FALSE.

Usage example

```
' Execute SetZero
```

```
Dim param As Variant
```

```
Param = Array(0, True)
```

```
Call caoCtrl.Execute("SetZero", param)
```

3.3.2.4. SetReset Commands

Sets the reset input and resets the measured value of the specified OUT number. The following arguments are provided.

Item	Type Description	
Argument	VT_I4	Specify the OUT number to be set. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2 • 2 - OUT1 + OUT2

Usage example

```
' Execute SetReset
```

```
Call caoCtrl.Execute("SetReset", 0)
```

3.3.2.5. SetPanelLock Commands

Locks the keystrokes on the display panel. By locking the display panel, you can prevent accidental operation even if you accidentally touch the operation key. The following arguments are provided.

Item	Type Description	
Argument	VT_BOOL	Specify panel lock. Specify ON if TRUE, OFF if FALSE.

Usage example

```
' Execute SetPanelLock
```

```
Call caoCtrl.Execute("SetPanelLock", true)
```

3.3.2.6. SetProgramNo Commands

Switches the program number. The following arguments are provided.

Item	Type Description	
Argument	VT_I4	Specify the program number to be switched. You can specify a value from 0 to 7.

Usage example

```
' Execute SetProgramNo
```

```
Call caoCtrl.Execute("SetProgramNo", 0)
```

3.3.2.7. GetProgramNo Commands

Gets the current program number. The following arguments are provided.

Item	Type Description	
Return Value	VT_I4	Gets the program number.

Usage example

```
'Execute GetProgramNo
```

```
Dim value As Integer
```

```
Value = caoCtrl.Execute("GetProgramNo")
```

3.3.2.8. GetFigureData Commands

Acquires the statistical results of the measured values that have been statistically processed. The data subject to statistical processing is the data held in each measurement mode. The following are the arguments and return values:

Item	Type Description		
Argument	VT_I4	Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2 	
Return Value	VT_ARRAY VT_VARIANT		
	0	VT_ARRAY VT_VARIANT	Tolerance upper limit
	0	VT_I4	Whether the data is valid or not. One of the following is obtained: <ul style="list-style-type: none"> • 0-Valid data • 1-+ RANGE OVER • 2--Range over • 3-Waiting for judgment
	1	VT_R4	Numeric value. This value is invalid if it is not valid data.
	1	VT_ARRAY VT_VARIANT	Lower tolerance limit
	0	VT_I4	Whether the data is valid or not. Obtain the same value as the above tolerance upper limit.
	1	VT_R4	Numeric value. This value is invalid if it is not valid data.
	2	VT_ARRAY VT_VARIANT	Mean

Item	Type Description		
	0	VT_I4	Whether the data is valid or not. Obtain the same value as the above tolerance upper limit.
	1	VT_R4	Numeric value. This value is invalid if it is not valid data.
	3	VT_ARRAY VT_VARIANT	Maximum value
	0	VT_I4	Whether the data is valid or not. Obtain the same value as the above tolerance upper limit.
	1	VT_R4	Numeric value. This value is invalid if it is not valid data.
	4	VT_ARRAY VT_VARIANT	Minimum value
	0	VT_I4	Whether the data is valid or not. One of the following is obtained: <ul style="list-style-type: none"> • 0-Valid data • 1-+ RANGE OVER • 2--Range over • 3-Waiting for judgment
	1	VT_R4	Numeric value. This value is invalid if it is not valid data.
	5	VT_ARRAY VT_VARIANT	Maximum-Minimum
	0	VT_I4	Whether the data is valid or not. Obtain the same value as the above tolerance upper limit.
	1	VT_R4	Numeric value. This value is invalid if it is not valid data.
	6	VT_ARRAY VT_VARIANT	Standard deviation
	0	VT_I4	Whether the data is valid or not. Obtain the same value as the above tolerance upper limit.
	1	VT_R4	Numeric value. This value is invalid if it is not valid data.
	7	VT_I4	Total number of data
	8	VT_I4	High judgement data-count
9	VT_I4	Number of Go judgment data	
10	VT_I4	Number of low judgement data	

Usage example

```
'Execute GetFigureData
Dim values As Variant
```

```
Values = caoCtrl.Execute("GetFigureData", 0)
```

```
If Not IsEmpty(values) Then
```

```
    ' Tolerance upper limit
```

```
    Dim maxLimit As Variant
```

```
    MaxLimit = values(0)
```

```
    ' Valid data or not
```

```
    Dim validData1 As Long
```

```
    ValidData1 = maxLimit(0)
```

```
    ' Measured value
```

```
    Dim fValue1 As Single
```

```
    fValue1 = maxLimit(1)
```

```
    ' Lower tolerance limit
```

```
    Dim minLimit As Variant
```

```
    MinLimit = values(1)
```

```
    ' Valid data or not
```

```
    Dim validData2 As Long
```

```
    ValidData2 = minLimit(0)
```

```
    ' Measured value
```

```
    Dim fValue2 As Single
```

```
    fValue2 = minLimit(1)
```

```
    ' Mean
```

```
    Dim average As Variant
```

```
    Average = values(2)
```

```
    ' Valid data or not
```

```
    Dim validData3 As Long
```

```
    ValidData3 = average(0)
```

```
    ' Measured value
```

```
    Dim fValue3 As Single
```

```
    fValue3 = average(1)
```

```
    ' Maximum value
```

```
    Dim maxValue As Variant
```

```
    MaxValue = values(3)
```

```
    ' Valid data or not
```

```
    Dim validData4 As Long
```

```
    ValidData4 = maxValue(0)
```

```
    ' Measured value
```

```
    Dim fValue4 As Single
```

```
    fValue4 = maxValue(1)
```

```
    ' Minimum value
```

```
    Dim minValue As Variant
```

```
    MinValue = values(4)
```

```
    ' Valid data or not
```

```
    Dim validData5 As Long
```

```
    ValidData5 = minValue(0)
```

```
    ' Measured value
```

```
    Dim fValue5 As Single
```

```
    fValue5 = minValue(1)
```

```
    ' Maximum-Minimum
```

```
    Dim difValue As Variant
```

```

DifValue = values(5)
' Valid data or not
Dim validData6 As Long
ValidData6 = difValue(0)
' Measured value
Dim fValue6 As Single
fValue6 = difValue(1)

' Standard deviation
Dim stndDev As Variant
StndDev = values(6)
' Valid data or not
Dim validData7 As Long
ValidData7 = stndDev(0)
' Measured value
Dim fValue7 As Single
fValue7 = stndDev(1)

' Total number of data
Dim totalDataCnt As Integer
TotalDataCnt = values(7)
' High judgement data count
Dim highDataCnt As Integer
HighDataCnt = values(8)
' Number of Go judgment data
Dim goDataCnt As Integer
GoDataCnt = values(9)
' Number of low judgement data
Dim lowDataCnt As Integer
LowDataCnt = values(10)

```

End If

3.3.2.9. ClearFigureData Commands

Clears the statistics.

Usage example

```

' Execute ClearFigureData
Call caoCtrl.Execute("ClearFigureData")

```

3.3.2.10. StartDataStorage Commands

Starts accumulating data in data storage.

Usage example

```

' Execute StartDataStorage
Call caoCtrl.Execute("StartDataStorage")

```

3.3.2.11. StopDataStorage Commands

Stops accumulating data in data storage.

Usage example

```

' Execute StopDataStorage

```

Call caoCtrl.Execute("StopDataStorage")

3.3.2.12. ClearDataStorage Commands

Clears the accumulated data in the data storage.

Usage example

' Execute ClearDataStorage

Call caoCtrl.Execute("ClearDataStorage")

3.3.2.13. GetDataStorageData Commands

Retrieves accumulated data for data storage. The following are the arguments and return values:

Item	Type Description		
Argument	VT_ARRAY VT_I4		
	0	VT_I4	Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2
	1	VT_I4	Specify the number of data to be acquired. You can specify a value from 1 to 65536.
Return Value	VT_ARRAY VT_VARIANT		
	0	VT_I4	Number of accumulated data items acquired
	I	VT_ARRAY VT_VARIANT	I-th stored data
	0	VT_I4	Whether the data is valid or not. One of the following is obtained: <ul style="list-style-type: none"> • 0-Valid data • 1-+ RANGE OVER • 2--Range over • 3-Waiting for judgment
	1	VT_R4	Numeric value. This value is invalid if it is not valid data.

※ i-number of accumulated data items acquired

Usage example

' Execute GetDataStorageData

Dim param As Variant

Param = Array(0, 3)

Dim value As Variant

Value = caoCtrl.Execute("GetDataStorageData", param)

If Not IsEmpty(value) Then

' Number of accumulated data items acquired

Dim readDataCnt As Integer

```

ReadDataCnt = value(LBound (value))
Dim i As Integer
For i = LBound(value) + 1 To (UBound(value))
    ' Stored data
    Dim accumulationData As Variant
    AccumulationData = value(i)
    If Not IsEmpty(accumulationData) Then
        ' Valid data or not
        Dim result As Integer
        Result = accumulationData(0)
        ' Measured value
        Dim fValue As Single
        fValue = accumulationData(1)
    End If
Next i
End If

```

3.3.2.14. GetDataStorageStatus Commands

Acquires the accumulated state of data storage. The following are the arguments and return values:

Item	Type Description	
Argument	VT_I4	Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2
Return Value	VT_ARRAY VT_VARIANT	
	0	VT_BOOL Acquires whether or not accumulation is in progress. <ul style="list-style-type: none"> • TRUE-accumulating • FALSE-stopped
	1	VT_I4 Number of stored data items

Usage example

```

' Execute GetDataStorageStatus
Dim value As Variant
Value = caoCtrl.Execute("GetDataStorageStatus", 0)
If Not IsEmpty(value) Then
    ' Indicates whether or not data is being accumulated.
    Dim isAccumulation As Boolean
    IsAccumulation = value(LBound(value))
    ' Number of stored data items
    Dim dataCnt As Integer
    DataCnt = value(UBound(value))
End If

```

3.3.2.15. GetLight Commands

Obtains the received light waveform. The following are the arguments and return values:

Item	Type Description
------	------------------

Item	Type Description			
Argument	VT_ARRAY VT_I4			
	0	VT_I4	Specify the head number. Specify one of the following: <ul style="list-style-type: none"> • 0 - HEAD-A • 1 - HEAD-B 	
	1	VT_I4	Specify the number of data to be acquired. You can specify a value from 1 to 65536.	
Return Value	VT_ARRAY VT_VARIANT			
	0	VT_ARRAY VT_I4		Measuring position
		0	VT_I4	Measurement position of peak number 0
		1	VT_I4	Peak number 1 measurement position
		2	VT_I4	Measurement position of peak number 2
		3	VT_I4	Measurement position of peak number 3
	1	VT_I4	Number of valid data	
	2	VT_ARRAY VT_UI1		Read data
		0	VT_I4	First data
	
1023		VT_I4	1024st data	

```
' Execute GetLight
```

```
Dim param As Variant
```

```
Param = Array(0, 0)
```

```
Dim value As Variant
```

```
Value = caoCtrl.Execute("GetLight", param)
```

```
If Not IsEmpty(value) Then
```

```
    ' Measuring position
```

```
    Dim measurePosition As Variant
```

```
    MeasurePosition = value(0)
```

```
    If Not IsEmpty(value) Then
```

```
        ' Measurement position of peak number 0
```

```
        Dim peekPosition0 As Integer
```

```
        PeekPosition0 = measurePosition(0)
```

```
        ' Peak number 1 measurement position
```

```
        Dim peekPosition1 As Integer
```

```
        PeekPosition1 = measurePosition(1)
```

```
        ' Measurement position of peak number 2
```

```
        Dim peekPosition2 As Integer
```

```
        PeekPosition2 = measurePosition(2)
```

```
        ' Measurement position of peak number 3
```

```
        Dim peekPosition3 As Integer
```

```
        PeekPosition3 = measurePosition(3)
```

```
    End If
```

```
    ' Number of valid data
```

```

Dim effDataCnt As Integer
EffDataCnt = value(1)

' Read data
Dim allReadData As Variant
AllReadData = value(2)
If Not IsEmpty(value) Then
    Dim i As Integer
    For i = 0 To 1023
        Dim readData As Byte
        ReadData = allReadData(i)
    Next i
End If
End If

```

3.3.3. Configuration-Related Commands

3.3.3.1. Panel Display-Related Commands

3.3.3.1.1. SetPanel Commands

Switches the panel display. The following arguments are provided.

Item	Type Description	
Argument	VT_I4	Specify the OUT number to be displayed. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2 • 2 - OUT1+OUT2

Usage example

```

' Execute SetPanel
Call caoCtrl.Execute("SetPanel", 0)

```

3.3.3.1.2. GetPanel Commands

Gets the currently displayed panel. The return values are shown below.

Item	Type Description	
Return Value	VT_I4	Gets the displayed OUT number. Refer to 3.3.3.1.1 for details of the value.

Usage example

```

' Execute GetPanel
Dim outNo As Integer
OutNo = caoCtrl.Execute("GetPanel")

```

3.3.3.2. Commands Related to Tolerance Setting

3.3.3.2.1. SetTolerance Commands

Sets the judgment value (tolerance judgment value) of the tolerance range. When the upper limit value is

exceeded (HI), when the lower limit value is exceeded (LO), and when the upper limit value is exceeded (GO), the judgment is made in three levels, and the display and output can be performed. If the measured value is raised or lowered near the tolerance judgment value, the judgment output may repeat ON/OFF. When hysteresis is set, the width of the detected value and the restored value of tolerance judgment can be set, so that such a situation can be prevented. For details on the displayed information and hysteresis, refer to "Chapter 2-Setting Tolerance Determination Values" in LK-G3000 manuals. The following arguments are provided.

Item	Type Description	
Argument	VT_ARRAY VT_I4	
	0	VT_I4 Specify the OUT number. Specify one of the following: • 0 - OUT1 • 1 - OUT2
	1	VT_I4 Specifies the upper tolerance value. Specify a value from -999999 to 999999.
	2	VT_I4 Specifies the lower tolerance value. Specify a value from -999999 to 999999.
3	VT_I4 Specifies the tolerance hysteresis. Specify a value from 0 to 999999.	

※ If you do not specify "Tolerance high limit-(Tolerance low limit) > Tolerance hysteresis", an API error is returned. An example is shown below.

Good example-> Tolerance high limit = 1000, Tolerance low limit = 100, Tolerance hysteresis = 0

Poor Example-> Tolerance Upper Limit = -1000, Tolerance Lower Limit = 100, Tolerance Hysteresis = 0

Usage example

'Execute SetTolerance

Dim param As Variant

Param = Array(0, 1000, 100, 0)

Call caoCtrl.Execute("SetTolerance", param)

3.3.3.2.2. GetTolerance Commands

Acquires the judgment value (tolerance judgment value) of the tolerance range. The following are the arguments and return values:

Item	Type Description	
Argument	VT_I4 Specify the OUT number to be displayed. Specify one of the following: • 0 - OUT1 • 1 - OUT2	
	VT_ARRAY VT_I4	
Return Value	VT_ARRAY VT_I4	
	0	VT_I4 Gets the upper tolerance value.

Item	Type Description		
Argument	VT_I4	Specify the OUT number to be displayed. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2 	
	1	VT_I4	Gets the lower tolerance value.
	2	VT_I4	Obtains the tolerance hysteresis.

Usage example

```
' Execute GetTolerance
Dim tolerance As Variant
Tolerance = caoCtrl.Execute("GetTolerance", 0)
If Not IsEmpty(tolerance) Then
    ' Tolerance upper limit
    Dim maxLimit As Integer
    MaxLimit = tolerance(0)
    ' Lower tolerance limit
    Dim minLimit As Integer
    MinLimit = tolerance(1)
    ' Tolerance hysteresis
    Dim hysteresis As Integer
    Hysteresis = tolerance(2)
End If
```

3.3.3.3. Head setting-related commands

This command sets/obtains the function related to sensing for stable detection.

3.3.3.3.1. SetAbleMode Commands

Set ABLE tuning mode. ABLE function automatically adjusts the appropriate amount of light and sensitivity to the surface condition (color, gloss, material) of the object. The following arguments are provided.

Item	Type Description		
	VT_ARRAY VT_I4		
Argument	0	VT_I4	Specify the head number. Specify one of the following: <ul style="list-style-type: none"> • 0 - HEAD-A • 1 - HEAD-B
	1	VT_I4	Specify ABLE tuning mode. For ABLE tuning mode, See Table 3-3. Please specify one of the following. <ul style="list-style-type: none"> • 0-Automatic • 1-Documentation

Table 3-3 ABLE Tuning Mode Detailed

Mode	Function
------	----------

Automatic	Automatically adjust the light to an appropriate amount. Normally, select here.
Documentation	Adjust the light intensity and sensitivity by limiting the adjustment range to any range from 1 to 99. Select this if the reflectance of the object changes significantly at a quick cycle, or if you want to detect only the object.

Usage example

```
'Execute SetAbleMode
```

```
Dim param As Variant
```

```
Param = Array(0, 0)
```

```
Call caoCtrl.Execute("SetAbleMode", param)
```

3.3.3.3.2. GetAbleMode Commands

Gets ABLE tuning mode. The following are the arguments and return values:

Item	Type Description	
Argument	VT_I4	Specify the head number. Specify one of the following: <ul style="list-style-type: none"> • 0 - HEAD-A • 1 - HEAD-B
Return Value	VT_I4	Gets ABLE tuning mode. Refer to 3.3.3.3.1 for details of the value.

Usage example

```
'Execute GetAbleMode
```

```
Dim mode As Integer
```

```
Mode = caoCtrl.Execute("GetAbleMode", 0)
```

3.3.3.3.3. SetAbleMinMax Commands

Sets ABLE control area. The following arguments are provided.

Item	Type Description	
Argument	VT_ARRAY VT_I4	
	0	VT_I4 Specify the head number. Specify one of the following: <ul style="list-style-type: none"> • 0 - HEAD-A • 1 - HEAD-B
	1	VT_I4 Specifies ABLE minimum. Specify a value from 1 to 99.
	2	VT_I4 Specifies ABLE max. Specify a value from 1 to 99.

※ An API-error is returned except "ABLE max.-ABLE min. >= 0".

Usage example

```
' Execute SetAbleMinMax
```

```
Dim param As Variant
```

```
Param = Array(0, 8, 10)
```

```
Call caoCtrl.Execute("SetAbleMinMax", param)
```

3.3.3.3.4. GetAbleMinMax Commands

Gets ABLE control scope. The following are the arguments and return values:

Item	Type Description		
Argument	VT_I4	Specify the head number. Specify one of the following: <ul style="list-style-type: none"> • 0 - HEAD-A • 1 - HEAD-B 	
Return Value	VT_ARRAY VT_I4		
	0	VT_I4	Get ABLE min.
	1	VT_I4	Get ABLE max.

Usage example

```
' Execute GetAbleMinMax
```

```
Dim value As Variant
```

```
Value = caoCtrl.Execute("GetAbleMinMax", 0)
```

```
If Not IsEmpty(value) Then
```

```
    ' ABLE min.
```

```
    Dim minAble As Integer
```

```
    MinAble = value(0)
```

```
    ' ABLE max
```

```
    Dim maxAble As Integer
```

```
    MaxAble = value(1)
```

```
End If
```

3.3.3.3.5. SetMeasureMode Commands

Set the measurement mode according to the object to be measured. By specifying what the object to be measured is, stable detection is performed. The following arguments are provided.

Item	Type Description	
Argument	VT_ARRAY VT_I4	
	0	VT_I4

Item	Type Description	
	1	VT_I4 Specifies the measurement mode. For details of each measurement mode, see Table 3-4. Specify one of the following: <ul style="list-style-type: none"> • 0-Standard • 1-Translucent body • 2-transparency • 3-Transparency 2 • 4-Multiple reflector

Table 3-4 Measurement Mode Detail

Mode	Function
Standard	Normally, this setting is used.
Translucent body	Corresponds to objects such as semi-transparent plastic that can be used to soak light.
Transparency	Used to measure transparency variations and thickness. This is used when the reflectivity of multiple planes of the transparency is the same party.
Transparency 2	This is used when reflectivity of multiple surfaces (up to four surfaces) such as the front and back surfaces of a transparent body is different, etc.
Multiple reflector	Used for bending measurement of terminals of ICs and connectors, etc.

Usage example

```
'Execute SetMeasureMode
```

```
Dim param As Variant
```

```
Param = Array(0, 0)
```

```
Call caoCtrl.Execute("SetMeasureMode", param)
```

3.3.3.3.6. GetMeasureMode Commands

Gets the measurement mode. The following are the arguments and return values:

Item	Type Description	
Argument	VT_I4	Specify the head number. Specify one of the following: <ul style="list-style-type: none"> • 0 - HEAD-A • 1 - HEAD-B
Return Value	VT_I4	Gets the measurement mode. Refer to 3.3.3.3.5 for details of the value.

Usage example

```
'Execute GetMeasureMode
```

```
Dim mode As Integer
```

```
Mode = caoCtrl.Execute("GetMeasureMode", 0)
```

3.3.3.3.7. SetNumAlarm Commands

Set the number of alarm processing operations. The following arguments are provided.

Item	Type Description	
Argument	VT_ARRAY VT_I4	
	0	VT_I4 Specify the head number. Specify one of the following: <ul style="list-style-type: none"> • 0 - HEAD-A • 1 - HEAD-B
	1	VT_I4 Specify the number of times the alarm is processed. Specify a value from 0 to 999.

Usage example

```
'Execute SetNumAlarm
Dim param As Variant
Param = Array(0, 5)
Call caoCtrl.Execute("SetNumAlarm", param)
```

3.3.3.3.8. GetNumAlarm Commands

Acquires the number of alarm processing operations. The following are the arguments and return values:

Item	Type Description	
Argument	VT_I4	Specify the head number. Specify one of the following: <ul style="list-style-type: none"> • 0 - HEAD-A • 1 - HEAD-B
Return Value	VT_I4	Acquires the number of alarm processing operations.

Usage example

```
'Execute GetNumAlarm
Dim numAlram As Integer
NumAlram = caoCtrl.Execute("GetNumAlarm", 0)
```

3.3.3.3.9. SetAlarmLevel Commands

Sets the alarm level. The following arguments are provided.

Item	Type Description	
Argument	VT_ARRAY VT_I4	
	0	VT_I4 Specify the head number. Specify one of the following: <ul style="list-style-type: none"> • 0 - HEAD-A • 1 - HEAD-B

Item	Type Description	
	1	VT_I4 Specify the alarm level. Specify a value from 0 to 9. The larger the value, the more likely it is to be an alarm.

Usage example

```
'Execute SetAlarmLevel
```

```
Dim param As Variant
```

```
Param = Array(0, 3)
```

```
Call caoCtrl.Execute("SetAlarmLevel", param)
```

3.3.3.3.10. GetAlarmLevel Commands

Gets the alarm level. The following are the arguments and return values:

Item	Type Description	
Argument	VT_I4	Specify the head number. Specify one of the following: <ul style="list-style-type: none"> • 0 - HEAD-A • 1 - HEAD-B
Return Value	VT_I4	Gets the alarm level.

Usage example

```
'Execute GetAlarmLevel
```

```
Dim level As Integer
```

```
Level = caoCtrl.Execute("GetAlarmLevel", 0)
```

3.3.3.3.11. StartABLE Commands

Starts ABLE tuning. ABLE tuning function optimizes ABLE tuning by allowing the target head to actually measure the object. Perform ABLE tuning referring to Fig. 3-1. When ABLE tuning is executed, ABLE tuning mode is set manually, and ABLE upper limit and ABLE lower limit are set to the adjusted values. This command terminates the unit while the unit is in the communication mode. When this command is executed, be sure to execute StopABLE command or CancelABLE command to set the main unit to normal mode. The following arguments are provided.

Item	Type Description	
Argument	VT_I4	Specify the head number. Specify one of the following: <ul style="list-style-type: none"> • 0 - HEAD-A • 1 - HEAD-B

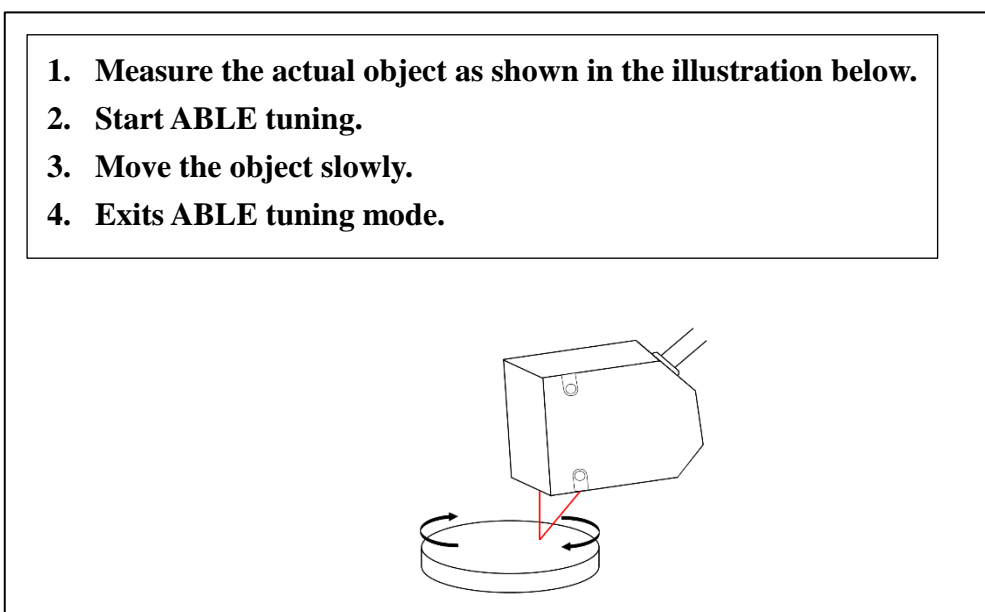


Fig. 3-1 ABLE Tuning Flow

Usage example

'Execute StartABLE

Call caoCtrl.Execute("StartABLE", 0)

3.3.3.3.12. StopABLE Commands

Finish ABLE tuning.

Usage example

'Execute StopABLE

Call caoCtrl.Execute("StopABLE", "")

3.3.3.3.13. CancelABLE Commands

Cancels ABLE tuning.

Usage example

'Execute CancelABLE

Call caoCtrl.Execute("CancelABLE", "")

3.3.3.3.14. SetReflectionMode Commands

Sets the installation mode. The following arguments are provided.

Item	Type Description		
Argument	VT_ARRAY VT_I4		
	0	VT_I4	Specify the head number. Specify one of the following: <ul style="list-style-type: none"> • 0 - HEAD-A • 1 - HEAD-B

Item	Type Description	
	1	VT_I4 Specify the installation mode. Refer to Table 3-5 for details on the installation mode. Specify one of the following: <ul style="list-style-type: none"> • 0-Diffuse reflection • 1-Regular reflection

Table 3-5 Installation Mode Details

Mode	Function
Diffuse reflection	Sets the diffuse reflection. Typically, you select this setting.
Regular reflection	Sets the regular reflection. Select this when the object to be measured is a mirror or glass.

Usage example

```
' Execute SetReflectionMode
Dim param As Variant
Param = Array(0, 0)
Call caoCtrl.Execute("SetReflectionMode", param)
```

3.3.3.3.15. GetReflectionMode Commands

Gets the installation mode. The following are the arguments and return values:

Item	Type Description	
Argument	VT_I4	Specify the head number. Specify one of the following: <ul style="list-style-type: none"> • 0 - HEAD-A • 1 - HEAD-B
Return Value	VT_I4	Gets the installation mode. Refer to 3.3.3.3.14 for details of the value.

Usage example

```
' Execute GetReflectionMode
Dim mode As Integer
Mode = caoCtrl.Execute("GetReflectionMode", 0)
```

3.3.3.4. OUT setting-related commands

This command sets/gets functions related to data processing.

3.3.3.4.1. SetCalcMethod Commands

Sets the calculation method between heads. Surface displacement, thickness, and level difference can be measured by calculating the data obtained by head setting of Head A or Head B according to the object to be measured. The following arguments are provided.

Item	Type Description	
Argument	VT_ARRAY VT_I4	
	0	VT_I4 Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2
	1	VT_I4 Specifies the operation method. See Table 3-6 for details on the calculation method. Specify one of the following: <ul style="list-style-type: none"> • 0-Head A • 1-Head B • 2-Head A + Head B • 3-Head A to Head B • 4-Head A transparency • 5-Head B transparency
2	VT_I4 Specify the measurement target. Refer to Table 3-7 for details on the measurement targets. Specify one of the following: <ul style="list-style-type: none"> • 0-Peak 1 • 1-Peak 2 • 2-Peak 3 • 3-Peak 4 • 4-Peak 1-Peak 2 • 5-Peak 1-Peak 3 • 6-Peak 1-Peak 4 • 7-Peak 2-Peak 3 • 8-Peak 2-Peak 4 • 9-Peak 3-Peak 4 	

Table 3-6 Operation method details

Operation	Function
Head A	Surface displacement measurement of head A or head B
Head B	
Head A + Head B	Thickness determination using head A and head B

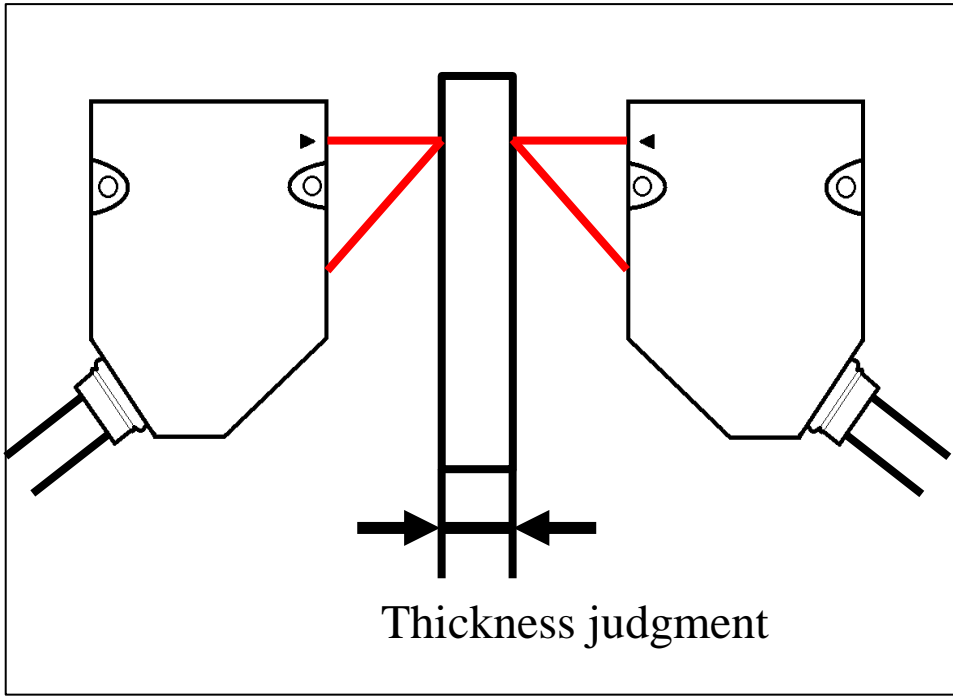
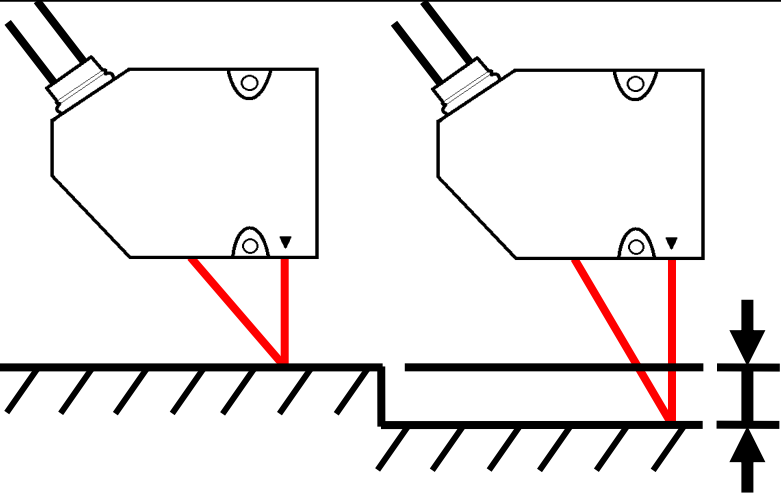
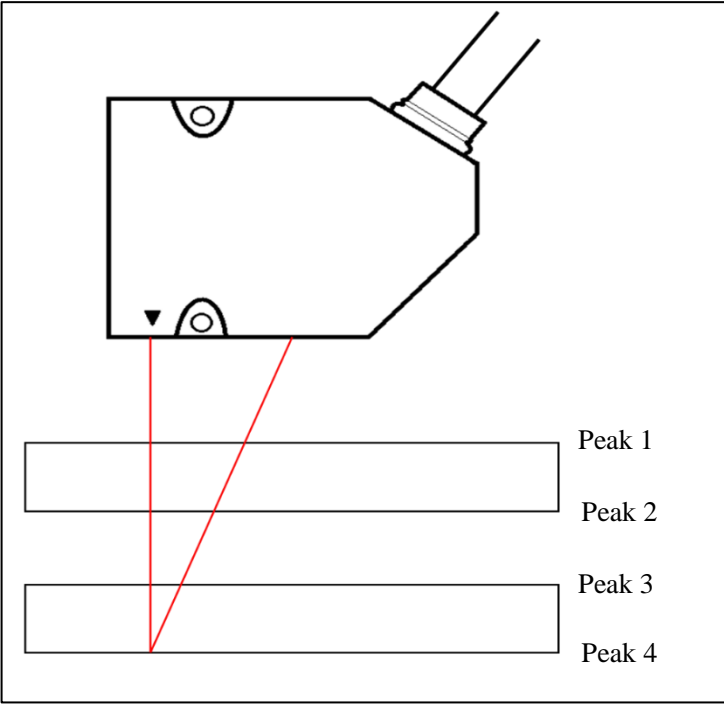
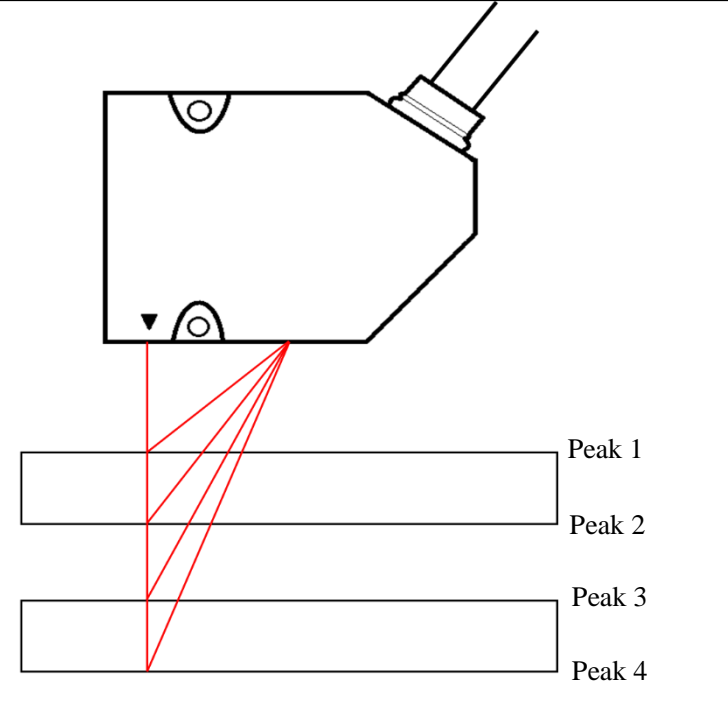
Operation	Function
	 <p style="text-align: center;">Thickness judgment</p>
<p>Head A to Head B</p>	 <p style="text-align: right;">Level difference Judgment</p>
<p>Head A transparency</p>	<p>Measure the displacement and thickness of the transparent body. The selection of the target plane is specified by the measurement target.</p>
<p>Head B transparency</p>	

Table 3-7 Measurement target details

Measuring surface	Function
Peak 1	<p>1 Displacement measurement of reflecting surface</p>  <p>The diagram shows a cross-section of a lens with a probe tip on its top surface. Four measurement points are indicated: Peak 1 is at the top surface of the lens; Peak 2 is at the top surface of the first glass layer; Peak 3 is at the top surface of the second glass layer; and Peak 4 is at the bottom surface of the second glass layer. Red lines connect the probe tip to each of these four peaks.</p>
Peak 2	
Peak 3	
Peak 4	
Peak 1-2	<p>2 Measure the reflective surface and select Peak 1-2 to measure the thickness of the first glass. You can also measure the spacing between the first and second glass by selecting Peak 2-3.</p>
Peak 1-3	
Peak 1-4	
Peak 2-3	
Peak 2-4	

Measuring surface	Function
Peak 3-4	

Usage example

```
'Execute SetCalcMethod
Dim param As Variant
Param = Array(0, 0, 0)
Call caoCtrl.Execute("SetCalcMethod", param)
```

3.3.3.4.2. GetCalcMethod Commands

Gets the calculation method. The following are the arguments and return values:

Item	Type Description	
Argument	VT_I4	Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2
Return Value	VT_ARRAY VT_I4	
	0	VT_I4 Gets the calculation method. Refer to 3.3.3.4.1 for details of the value.
	1	VT_I4 Gets the measurement object. Refer to 3.3.3.4.1 for details of the value.

Usage example

```
'Execute GetCalcMethod
Dim value As Variant
```

```

Value = caoCtrl.Execute("GetCalcMethod", 0)
If Not IsEmpty(value) Then
    ' Calculation method
    Dim method As Integer
    Method = value(0)
    ' Measurement object
    Dim target As Integer
    Target = value(1)
End If

```

3.3.3.4.3. SetScaling Commands

Set the scaling. By setting the scaling, the displayed value relative to the measured value can be calibrated arbitrarily. Calibration sets the values to be displayed for any two points. For each OUT1,OUT2, head A and head B can be calibrated. For more information about scaling, see LK-G3000 documentation. The following arguments are provided.

Item	Type Description		
Argument	VT_ARRAY VT_I4		
	0	VT_I4	Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2
	1	VT_I4	Specify the head number. Specify one of the following: <ul style="list-style-type: none"> • 0 - HEAD-A • 1 - HEAD-B
	2	VT_I4	Specify an input value of 1. Specify a value from-999999 to 999999.
	3	VT_I4	Specify the display value of 1. Specify a value from-999999 to 999999.
	4	VT_I4	Specify the input value 2. Specify a value from-999999 to 999999.
5	VT_I4	Specify the display value of 2. Specify a value from-999999 to 999999.	

※ If the following conditions are not met, the configuration fails and an API error is returned:

- (1) Input value 1-Input value 2 \neq 0

Usage example

```

' Execute SetScaling
Dim param As Variant
Param = Array(0, 0, 500, 10, 1000, 1000)
Call caoCtrl.Execute("SetScaling", param)

```


3.3.3.4.4. GetScaling Commands

Get the scaling. The following are the arguments and return values:

Item	Type Description		
Argument	VT_ARRAY VT_I4		
	0	VT_I4	Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2
	1	VT_I4	Specify the head number. Specify one of the following: <ul style="list-style-type: none"> • 0 - HEAD-A • 1 - HEAD-B
Return Value	VT_ARRAY VT_I4		
	0	VT_I4	Get input value 1.
	1	VT_I4	Gets the display value 1.
	2	VT_I4	Get input value 2.
	3	VT_I4	Gets the display value 2.

Usage example

```
' Execute GetScaling
Dim param As Variant
Param = Array(0, 0)
Dim value As Variant
Value = caoCtrl.Execute("GetScaling", param)
If Not IsEmpty(value) Then
    ' Input value 1
    Dim inputValue1 As Integer
    InputValue1 = value(0)
    ' Displayed value 1
    Dim outputValue1 As Integer
    OutputValue1 = value(1)
    ' Input value 2
    Dim inputValue2 As Integer
    InputValue2 = value(2)
    ' Displayed value 2
    Dim outputValue2 As Integer
    OutputValue2 = value(3)
End If
```

3.3.3.4.5. SetFilterMode Commands

Sets the filter mode. By applying filters, you can make stable measurements. For more information on filtering, refer to Chapter 3-Setting Measurement Output Conditions-Filtering for Stable Measurement in LK-G3000 Manual. The following arguments are provided.

Item	Type Description
------	------------------

Item	Type Description	
Argument	VT_ARRAY VT_I4	
	0	VT_I4 Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2
	1	VT_I4 Specifies the filter mode. Specify one of the following: <ul style="list-style-type: none"> • 0-Moving average • 1-Low-pass filter • 2-High-pass filter

Usage example

```
' Execute SetFilterMode
```

```
Dim param As Variant
```

```
Param = Array(0, 0)
```

```
Call caoCtrl.Execute("SetFilterMode", param)
```

3.3.3.4.6. GetFilterMode Commands

Gets the filter mode. The following are the arguments and return values:

Item	Type Description	
Argument	VT_I4	Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2
Return Value	VT_I4	Gets the filter mode. See SetFilterMode Commands for more information on the values.SetFilterMode Commands

Usage example

```
' Execute GetFilterMode
```

```
Dim mode As Integer
```

```
Mode = caoCtrl.Execute("GetFilterMode", 0)
```

3.3.3.4.7. SetAverage Commands

Sets the average count when the filter mode is moving average. The following arguments are provided.

Item	Type Description	
Argument	VT_ARRAY VT_I4	
	0	VT_I4 Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2

Item	Type Description	
	1	VT_I4 Specify the number of averages. Specify one of the following: <ul style="list-style-type: none"> • 0-Once • 1-4 times • 2-16 times • 3-64 times • 4-256 times • 5-1024 times • 6-4096 times • 7-16384 times • 8-65536 times • 9-262144 times

Usage example

```
'Execute SetAverage
```

```
Dim param As Variant
```

```
Param = Array(0, 0)
```

```
Call caoCtrl.Execute("SetAverage", param)
```

3.3.3.4.8. GetAverage Commands

The filter mode gets the average number of moving averages. The following are the arguments and return values:

Item	Type Description	
Argument	VT_I4	Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2
Return Value	VT_I4	Gets the average count. See 3.3.3.4.7 for more information on the values.

Usage example

```
'Execute GetAverage
```

```
Dim average As Integer
```

```
Average = caoCtrl.Execute("GetAverage", 0)
```

3.3.3.4.9. SetCutOffFrequency Commands

Sets the cutoff frequency when the filter mode is set to low-pass filter or high-pass filter. The following arguments are provided.

Item	Type Description
Argument	VT_ARRAY VT_I4

Item	Type Description	
	0	VT_I4 Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2
	1	VT_I4 Specifies the cutoff frequency. Specify one of the following: <ul style="list-style-type: none"> • 0 - 1000Hz • 1 - 300Hz • 2 - 100Hz • 3 - 30Hz • 4 - 10Hz • 5 - 3Hz • 6 - 1Hz • 7 - 0.3Hz • 8 - 0.1Hz

Usage example**'Execute SetCutOffFrequency**

Dim param As Variant

Param = Array(0, 0)

Call caoCtrl.Execute("SetCutOffFrequency", param)

3.3.3.4.10. GetCutOffFrequency Commands

Used to acquire the cutoff frequency when the filter mode is set to low-pass filter or high-pass filter. The following are the arguments and return values:

Item	Type Description	
Argument	VT_I4	Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2
Return Value	VT_I4	Gets the cutoff frequency. See 3.3.3.4.9 for more information on the values.

Usage example**'Execute GetCutOffFrequency**

Dim frequency As Integer

Frequency = caoCtrl.Execute("GetCutOffFrequency", 0)

3.3.3.4.11. SetTriggerMode Commands

Sets or queries the trigger mode. The following arguments are provided.

Item	Type Description
------	------------------

Item	Type Description		
Argument	VT_ARRAY VT_I4		
	0	VT_I4	Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2
	1	VT_I4	Specify the trigger mode. For details on the trigger mode, see Table 3-8. Specify one of the following: <ul style="list-style-type: none"> • 0-External trigger 1 • 1-External trigger 2

Table 3-8 Trigger mode details

Mode	Function		
	Standard	Peak Hold/Bottom Hold/Peak to Peak Hold/Average Hold	Sample Hold
Trigger 1	Holds the internal measurement value while the timing input is rising (ON).	The sampling period is defined as the timing input from the rising edge to the next rising edge.	Holds the internal measurement value when the timing input is started.
Trigger 2		The sampling period is defined as the period from the falling (OFF) edge of the timing input to the next rising edge.	When the timing input starts, the data of the set average number of times is sampled from that time, and the fixed internal measurement value is held.

Usage example

```
'Execute SetTriggerMode
```

```
Dim param As Variant
```

```
Param = Array(0, 0)
```

```
Call caoCtrl.Execute("SetTriggerMode", param)
```

3.3.3.4.12. GetTriggerMode Commands

Gets the trigger mode. The following are the arguments and return values:

Item	Type Description
------	------------------

Item	Type Description	
Argument	VT_I4	Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2
Return Value	VT_I4	Gets the trigger mode. Refer to 3.3.3.4.11 for details of the value.

Usage example

```
'Execute GetTriggerMode
```

```
Dim mode As Integer
```

```
Mode = caoCtrl.Execute("GetTriggerMode", 0)
```

3.3.3.4.13. SetOffset Commands

Set the offset. By setting the offset, you can add or subtract any value from the displayed value. If an offset is set, the offset value can be displayed when auto zero is executed. The offset value is set for the measured value after measurement mode processing and auto zero processing. The following arguments are provided.

Item	Type Description	
Argument	VT_ARRAY VT_I4	
	0	VT_I4 Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2
	1	VT_I4 Specify the offset. Specify a value from -999999 to 999999.

Usage example

```
'Execute SetOffset
```

```
Dim param As Variant
```

```
Param = Array(0, 0)
```

```
Call caoCtrl.Execute("SetOffset", param)
```

3.3.3.4.14. GetOffset Commands

Gets the offset. The following are the arguments and return values:

Item	Type Description	
Argument	VT_I4	Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2
Return Value	VT_I4	Gets the offset.

Usage example

'Execute GetOffset

Dim offset As Integer

Offset = caoCtrl.Execute("GetOffset", 0)

3.3.3.4.15. SetAnalogScaling Commands

Sets the analog output scaling. For more information on analog output scaling, refer to "Chapter 3-Setting the Output Conditions of Measured Values-Scaling the Analog Output" in LK-G3000 manuals. The following arguments are provided.

Item	Type Description	
Argument	VT_ARRAY VT_I4	
	0	VT_I4 Specify the OUT number. Specify one of the following: • 0 - OUT1 • 1 - OUT2
	1	VT_I4 Specify an input value of 1. Specify a value from-999999 to 999999.
	2	VT_I4 Specifies the output voltage value of 1. Specify a value from-10500 to 10500.
	3	VT_I4 Specify the input value 2. Specify a value from-999999 to 999999.
4	VT_I4 Specifies the output voltage value of 2. Specify a value from-10500 to 10500.	

※ If the following conditions are not met, the configuration fails and an API error is returned:

- (1) Input value 1-Input value 2 ≠ 0

Usage example**'Execute SetAnalogScaling**

Dim param As Variant

Param = Array(0, 500, 10, 1000, 1000)

Call caoCtrl.Execute("SetAnalogScaling", param)

3.3.3.4.16. GetAnalogScaling Commands

Obtains the analog output scaling. The following are the arguments and return values:

Item	Type Description	
Argument	VT_I4 Specify the OUT number. Specify one of the following: • 0 - OUT1 • 1 - OUT2	
	VT_ARRAY VT_I4	
Return Value	0	VT_I4 Get input value 1.
	1	VT_I4 Gets the output voltage value 1.

Item	Type Description		
	2	VT_I4	Get input value 2.
	3	VT_I4	Gets the output voltage value 2.

Usage example

```
' Execute GetAnalogScaling
Dim value As Variant
Value = caoCtrl.Execute("GetAnalogScaling", 0)
If Not IsEmpty(value) Then
    ' Input value 1
    Dim inputValue1 As Integer
    inputValue1 = value(0)
    ' Output voltages 1
    Dim outputValue1 As Integer
    outputValue1 = value(1)
    ' Input value 2
    Dim inputValue2 As Integer
    inputValue2 = value(2)
    ' Output voltages 2
    Dim outputValue2 As Integer
    outputValue2 = value(3)
End If
```

3.3.3.4.17. SetCalcMode Commands

Sets the measurement mode. For more information on measurement modes, refer to "Chapter 3-Setting Measurement Output Conditions-Using the Hold Function (Measurement Mode)" in LK-G3000 Manual. The following arguments are provided.

Item	Type Description		
	VT_ARRAY VT_I4		
Argument	0	VT_I4	Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2
	1	VT_I4	Specifies the measurement mode. For details on the measurement modes that can be specified, see Table 3-9. Specify one of the following: <ul style="list-style-type: none"> • 0-Standard • 1-Peak Hold • 2-Bottom Hold • 3-Peak to Peak Hold • 4-Sample Hold • 5-Average Hold

Table 3-9 Details of measurement modes

Mode	Function
Standard	Measured results can be displayed and output at any time.
Peak Hold	The maximum value within the sampling period can be measured.
Bottom Hold	The minimum value within the sampling period can be measured.
Peak to Peak Hold	You can measure the "maximum value-minimum value" within the sampling period.
Sample Hold	The value at the moment the timing input is turned ON can be measured.
Average Hold	You can measure the average value within the sampling period.

Usage example

```
' Execute SetCalcMode
```

```
Dim param As Variant
```

```
Param = Array(0, 0)
```

```
Call caoCtrl.Execute("SetCalcMode", param)
```

3.3.3.4.18. GetCalcMode Commands

Gets the measurement mode. The following are the arguments and return values:

Item	Type	Description
Argument	VT_I4	Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2
Return Value	VT_I4	Gets the measurement mode. Refer to 3.3.3.4.17 for details of the value.

Usage example

```
' Execute GetCalcMode
```

```
Dim mode As Integer
```

```
Mode = caoCtrl.Execute("GetCalcMode", 0)
```

3.3.3.4.19. SetDisplayUnit Commands

Set the minimum display unit to be displayed on the panel. The following arguments are provided.

Item	Type	Description
	VT_ARRAY VT_I4	
Argument	0 VT_I4	Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2

Item	Type Description	
	1	VT_I4 Specifies the smallest display unit. Specify one of the following: <ul style="list-style-type: none"> • 0 - 0.01mm • 1 - 0.001mm • 2 - 0.0001mm • 3 - 0.00001mm • 4 - 0.01μm • 5 - 0.001μm

Usage example

```
'Execute SetDisplayUnit
```

```
Dim param As Variant
```

```
Param = Array(0, 0)
```

```
Call caoCtrl.Execute("SetDisplayUnit", param)
```

3.3.3.4.20. GetDisplayUnit Commands

Gets the smallest display unit. The following are the arguments and return values:

Item	Type Description	
Argument	VT_I4	Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2
Return Value	VT_I4	Gets the smallest display unit. Refer to 3.3.3.4.19 for details of the value.

Usage example

```
'Execute GetDisplayUnit
```

```
Dim unit As Integer
```

```
Unit = caoCtrl.Execute("GetDisplayUnit", 0)
```

3.3.3.4.21. SetAnalogThrough Commands

Sets the analog through. If the analog slew is set to ON while the measurement value is being held in measurement mode, the internal measurement value before holding is output in analog. The following arguments are provided.

Item	Type Description	
Argument	VT_ARRAY VT_VARIANT	
	0	VT_I4 Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2

Item	Type Description	
	1	VT_BOOL Specifies the analog through. <ul style="list-style-type: none"> • TRUE - ON • FALSE - OFF

Usage example

```
'Execute SetAnalogThrough
```

```
Dim param As Variant
```

```
Param = Array(0, false)
```

```
Call caoCtrl.Execute("SettAnalogThrough", param)
```

3.3.3.4.22. GetAnalogThrough Commands

Acquires an analog through. The following are the arguments and return values:

Item	Type Description	
Argument	VT_I4	Specify the OUT number. Specify one of the following: <ul style="list-style-type: none"> • 0 - OUT1 • 1 - OUT2
Return Value	VT_BOOL	Acquires an analog through.

Usage example

```
'Execute GetAnalogThrough
```

```
Dim value As Boolean
```

```
Value = caoCtrl.Execute("GetAnalogThrough", 0)
```

3.3.3.5. Common setting related commands

This command sets/obtains the common functions related to the head setting and OUT setting.

3.3.3.5.1. SetDataStorage Commands

Set the target OUT, number of accumulated points, and accumulation period to accumulate data storage. The following arguments are provided.

Item	Type Description	
	VT_ARRAY VT_I4	
Argument	0	VT_I4 Specify the target OUT. Specify one of the following: <ul style="list-style-type: none"> • 0-No target OUT • 1 - OUT1 • 2 - OUT2 • 3-OUT1 and OUT2
	1	VT_I4 Specify the number of points to be accumulated. Specify a value from 1 to 65536.

Item	Type Description		
	2	VT_I4	Specify the accumulation period. Specify one of the following: <ul style="list-style-type: none"> • 0-Sampling cycle × 1 • 1-Sampling period x 2 • 2-Sampling cycle × 5 • 3-Sampling period x 10 • 4-Sampling period x 20 • 5-Sampling cycle × 50 • 6-Sampling period × 100 • 7-Sampling period × 200 • 8-Sampling period × 500 • 9-Sampling period × 1000

Usage example

```
' Execute SetDataStorage
Dim param As Variant
Param = Array(1, 100, 0)
Call caoCtrl.Execute("SetDataStorage", param)
```

3.3.3.5.2. GetDataStorage Commands

Retrieves the target OUT, number of accumulated points, and accumulation period for data storage. The following are the arguments and return values:

Item	Type Description		
Return Value	VT_ARRAY VT_I4		
	0	VT_I4	Gets the target OUT. See SetDataStorage Commands for more information on the values.SetDataStorage Commands
	1	VT_I4	Retrieves the accumulated score.
	2	VT_I4	Acquires the accumulation period. See SetDataStorage Commands for more information on the values.SetDataStorage Commands

Usage example

```
' Execute GetDataStorage
Dim value As Variant
Value = caoCtrl.Execute("GetDataStorage")
If Not IsEmpty(value) Then
    ' Target OUT
    Dim out As Integer
    Out = value(0)
    ' Accumulated score
    Dim dataCnt As Integer
```

```

DataCnt = value(1)
' Storage cycle
Dim cycle As Integer
Cycle = value(2)
End If

```

3.3.3.5.3. SetSamplingCycle Commands

Sets the sampling period. The following arguments are provided.

Item	Type Description	
Argument	VT_I4	Specify the sampling period. Specify one of the following: <ul style="list-style-type: none"> • 0 - 20µs • 1 - 50µs • 2 - 100µs • 3 - 200µs • 4 - 500µs • 5 - 1000µs

Usage example

```

' Execute SetSamplingCycle
Call caoCtrl.Execute("SetSamplingCycle", 0)

```

3.3.3.5.4. GetSamplingCycle Commands

Acquires the sampling period. The return values are shown below.

Item	Type Description	
Return Value	VT_I4	Acquires the sampling period. Refer to 3.3.3.5.3 for details of the value.

Usage example

```

' Execute GetSamplingCycle
Dim value As Integer
Value = caoCtrl.Execute("GetSamplingCycle")

```

3.3.3.5.5. SetMutualInterPrev Commands

Configure mutual interference prevention. When mutual interference prevention is set to ON, two heads are issued alternately to prevent interference from the other head. The following arguments are provided.

Item	Type Description	
Argument	VT_BOOL	Specify mutual interference prevention. <ul style="list-style-type: none"> • TRUE - ON • FALSE - OFF

Usage example

```
'Execute SetMutualInterPrev
```

```
Call caoCtrl.Execute("SetMutualInterPrev", true)
```

3.3.3.5.6. GetMutualInterPrev Commands

Acquires mutual interference prevention. The return values are shown below.

Item	Type Description	
Return Value	VT_BOOL	Acquires mutual interference prevention.

Usage example

```
'Execute GetMutualInterPrev
```

```
Dim value As Boolean
```

```
Value = caoCtrl.Execute("GetMutualInterPrev")
```

3.3.3.5.7. SetTimingSync Commands

Sets how OUT1 and OUT2 timings are controlled. The following arguments are provided.

Item	Type Description	
Argument	VT_I4	Specify the timing synchronization. For details on timing synchronization, see Table 3-10. Specify one of the following: <ul style="list-style-type: none"> • 0-Asynchronous • 1-Synchronization

Table 3-10 Details of timing synchronization

Timing Synchronization	Function
Asynchronous	Controls OUT1 and OUT2 asynchronously. Assign independent inputpins to each OUT1,OUT2. <ul style="list-style-type: none"> • OUT1 : 8 of the 12-pin terminal block • OUT2 : Expansion connector number 8
Synchronization	Synchronize and control OUT1 and OUT2. No. 8 of the 12-pole terminal block corresponds, and No. 6 of the expansion connector is disabled.

※ For input terminals, refer to "Chapter 4 Input Terminals" in LK-G3000 manuals.

Usage example

```
'Execute SetTimingSync
```

```
Call caoCtrl.Execute("SetTimingSync", 0)
```

3.3.3.5.8. GetTimingSync Commands

Gets the timing synchronization. The return values are shown below.

Item	Type Description	
Return Value	VT_I4	Gets the timing synchronization. See 3.3.3.5.7 for more information on the values.

Usage example

```
'Execute GetTimingSync
```

```
Dim value As Integer
```

```
Value = caoCtrl.Execute("GetTimingSync")
```

3.3.3.5.9. SetTolCompOutputFormat Commands

Set the output form of tolerance judgment output. The following arguments are provided.

Item	Type Description	
Argument	VT_I4	Specifies the judgment output format. Refer to Table 3-11 for details on the judgment output format. Specify one of the following: <ul style="list-style-type: none"> • 0-Normal • 1-Hold • 2-off-delay

Table 3-11 Details of the judgment output mode

Timing Synchronization	Function
Normal	Output according to tolerance judgment.
Hold	Holds the output that is turned ON. Releases the hold by resetting the measured value.
Off delay	An off delay of 60ms is applied to the normal output. Releases the hold by resetting the measured value.

Usage example

```
'Execute SetTolCompOutputFormat
```

```
Call caoCtrl.Execute("SetTolCompOutputFormat", 0)
```

3.3.3.5.10. GetTolCompOutputFormat Commands

Acquires the output format of tolerance judgment output. The return values are shown below.

Item	Type Description
------	------------------

Item	Type Description	
Return Value	VT_I4	Acquires the judgment output format. See 3.3.3.5.9 for more information on the values.

Usage example

```
' Execute GetTolCompOutputFormat
Dim value As Integer
Value = caoCtrl.Execute("GetTolCompOutputFormat")
```

3.3.3.5.11. SetStorobeTime Commands

Sets the time (one-shot output time) at which the strobe output turns ON. The following arguments are provided.

Item	Type Description	
Argument	VT_I4	Specifies the strobe time. Specify one of the following: <ul style="list-style-type: none"> • 0 - 2ms • 1 - 5ms • 2 - 10ms • 3 - 20ms

Usage example

```
' Execute SetStorobeTime
Call caoCtrl.Execute("SetStorobeTime", 0)
```

3.3.3.5.12. GetStorobeTime Commands

Used to acquire the time (one-shot output time) during which the strobe output turns ON. The return values are shown below.

Item	Type Description	
Return Value	VT_I4	Gets the strobe time. Refer to 3.3.3.5.11 for details of the value.

Usage example

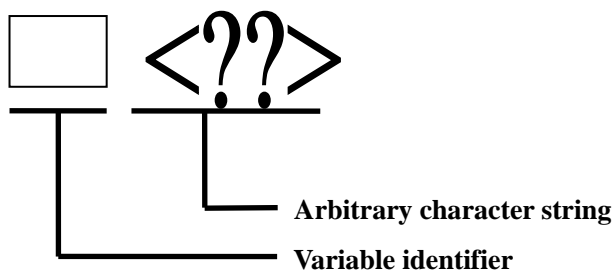
```
' Execute GetStorobeTime
Dim value As Integer
Value = caoCtrl.Execute("GetStorobeTime")
```

3.4. Variable list

Defines a list of variables that can be used in each class. Variables refer to objects of CaoVariable classes. An arbitrary string can be added to register multiple variables (useful when changing only options, etc.).

The following format gives an arbitrary string to a variable name:

Multi-variable common specification format



3.4.1. CaoController class-variable

Variable name	Description	Value		See Also
		Get	Put	
@MAKER_NAME	Obtain the manufacturer's name.	✓	-	P.57
@VERSION	Get the DLL version.	✓	-	P.57
@CALCDATA	Gets the measured value.	✓	-	P.58
RECEIVED_WAVEFROM<??>	Obtains the received light waveform.	✓	-	P.59

3.4.1.1. @MAKER_NAME

Obtain the manufacturer's name.

Data Type

Type	Description
VT_BSTR	Obtain the manufacturer's name.

Usage example

```
' Add Variable
Dim var As CaoVariable
Var = caoCtrl.AddVariable("@MAKER_NAME")
' Acquisition of Values
Dim name As String
Name = var.value
```

3.4.1.2. @VERSION

Gets the DLL version.

Data Type

Type	Description
------	-------------

VT_BSTR	Get the DLL version. *. *.*
---------	--------------------------------

Usage example

```
'Add Variable
Dim var As CaoVariable
Var = caoCtrl.AddVariable("@VERSION ")
'Acquisition of Values
Dim version As String
Version = var.value
```

3.4.1.3. @CALCDATA

Gets the measured value.

Data Type

Type Description		
VT_ARRAY VT_VARIANT		
0	VT_ARRAY	OUT1 Measured value
	VT_VARIANT	
	0 VT_I4	Whether the data is valid or not. One of the following is obtained: <ul style="list-style-type: none"> • 0-Valid data • 1+ RANGE OVER • 2--Range over • 3-Waiting for judgment
1	VT_R4	The measured value. This value is invalid if it is not valid data.
1	VT_ARRAY	OUT2 measured value
	VT_VARIANT	
	0 VT_I4	Whether the data is valid or not. Retrieves values similar to the measurements in OUT1 above.
1	VT_R4	The measured value. This value is invalid if it is not valid data.

Usage example

```
'Add Variable
Dim var As CaoVariable
Set var = caoCtrl.AddVariable("@CALCDATA")
'Retrieving Values
Dim values As Variant
Values = var.Value

If Not IsEmpty(values) Then
    'OUT1 readings
    Dim value1 As Variant
```

```

Value1 = values(0)
' Valid data or not
Dim validData1 As Long
ValidData1 = value1(0)
'Measured value
Dim fValue1 As Single
fValue1 = value1(1)

' OUT2 readings
Dim value2 As Variant
Value2 = values(1)
'Whether the data is valid or not
Dim validData2 As Long
ValidData2 = value2(0)
'Measured value
Dim fValue2 As Single
fValue2 = value2(1)

```

End If

3.4.1.4. RECEIVED_WAVEFROM<??>

Obtain the received waveform. Enter an arbitrary character string after RECEIVED_WAVEFROM to specify the variable name.

Option

Option	Required	Description	Value Range	Default Value
HeadNo	--	Specify the head number to be acquired. 0 (HEAD-A), Specify 1 (HEAD-B).	0 - 1	0
PeekNo	--	Specifies the peak position at which the waveform is to be acquired when the measurement mode is "Transparent body 2".	0 - 3	0

Data Type

Type Description		
VT_ARRAY VT_VARIANT		
0	VT_ARRAY VT_I4	Measuring position
0	VT_I4	Measurement position of peak number 0
1	VT_I4	Peak number 1 measurement position
2	VT_I4	Measurement position of peak number 2
3	VT_I4	Measurement position of peak number 3
1	VT_I4	Number of valid data
2	VT_ARRAY VT_UI1	Read data
0	VT_I4	First data

Type Description		
...
1023	VT_I4	1024st data

Usage example

' Add Variable

```
Dim var As CaoVariable
```

```
Set var = caoCtrl.AddVariable("RECEIVED_WAVEFROM1", "HeadNo = 0, PeekNo = 0")
```

' Retrieving Values

```
Dim value As Variant
```

```
Value = var.Value
```

```
If Not IsEmpty(value) Then
```

' Measuring position

```
Dim measurePosition As Variant
```

```
MeasurePosition = value(0)
```

```
If Not IsEmpty(value) Then
```

' Measurement position of peak number 0

```
Dim peekPosition0 As Integer
```

```
PeekPosition0 = measurePosition(0)
```

' Peak number 1 measurement position

```
Dim peekPosition1 As Integer
```

```
PeekPosition1 = measurePosition(1)
```

' Measurement position of peak number 2

```
Dim peekPosition2 As Integer
```

```
PeekPosition2 = measurePosition(2)
```

' Measurement position of peak number 3

```
Dim peekPosition3 As Integer
```

```
PeekPosition3 = measurePosition(3)
```

```
End If
```

' Number of valid data

```
Dim effDataCnt As Integer
```

```
EffDataCnt = value(1)
```

' Read data

```
Dim allReadData As Variant
```

```
AllReadData = value(2)
```

```
If Not IsEmpty(value) Then
```

```
Dim i As Integer
```

```
For i = 0 To 1023
```

```
Dim readData As Byte
```

```
ReadData = allReadData(i)
```

```
Next i
```

```
End If
```

```
End If
```

4. Programming by LK-G3000LkIF providers

LK-G3000LkIF providers prepare to communicate with the equipment as follows:

- Creating a CaoEngine
- Creating a CaoWorkspace
- Creating a CaoController

After you complete the steps, you can communicate with LK-G3000 series and access the device information by using Execute method of CaoController or by obtaining Value properties of CaoVariable object.

4.1. Sample-programming to obtain OUT1 and OUT2 measurements

This example shows a sample program that reads OUT1 and OUT2 readings. Table 4-1 describes the requirements of the sample program, and Fig. 4-1 describes the flow of the sample program.

Table 4-1 Sample program requirements

Requirements	Description
Process Description	Read OUT1 and OUT2 readings from LK-G3000.

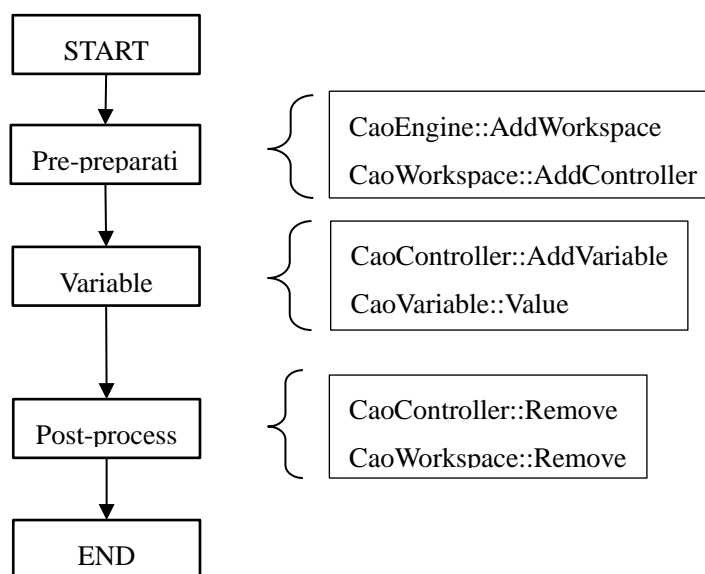


Fig. 4-1 OUT1 and OUT2 Reading Flow

Specific codes are given in the following sections.

4.1.1. Sample program

The following is an overview of the sample program.

Sample	GetCalcData.vb
'Object	
Dim caoEng As CaoEngine	
Dim caoWs As CaoWorkspace	
Dim caoCtrl As CaoController	

Dim caoVrl As CaoVariable

Private Sub Main()

' Preparation

Call Preprocessing

' Retrieving Values

Dim values As Variant

Values = caoVrl.value

If Not IsEmpty(values) Then

' OUT1 readings

Dim value1 As Variant

Value1 = values(0)

' Valid data or not

Dim validData1 As Long

ValidData1 = value1(0)

' Measured value

Dim fValue1 As Single

fValue1 = value1(1)

' OUT2 readings

Dim value2 As Variant

Value2 = values(1)

' Valid data or not

Dim validData2 As Long

ValidData2 = value2(0)

' Measured value

Dim fValue2 As Single

fValue2 = value2(1)

End If

' Post-processing

Call Postprocessing

End Sub

' Preparation method

Private Sub Preprocessing()

' Generate CaoEngine

Set caoEng = New CaoEngine

' Generate CaoWorkspace

Set caoWs = caoEng.AddWorkspace("Workspace", "")

' Generate CaoController

Set caoCtrl = caoWs.AddController("LKG3000LkIF", _
 "CaoProv.KEYENCE.LK-G3000LkIF", _
 "" , _
 "Timeout=1000")

' Generate CaoVariable

Set caoVrl = caoCtrl.AddVariable("@CALCDATA", "")

End Sub

' Post-processing method

Private Sub Postprocessing()

' Remove CaoVariable from CaoController

Call caoCtrl.variables.Remove(caoVrl.Index)

```
' Clear CaoVariable
Set caoVrl = Nothing
' Remove CaoController from CaoWorkspace
Call caoWs.Controllers.Remove(caoCtrl.Index)
' Clear CaoController
Set caoCtrl = Nothing
' Remove CaoWorkspace from CaoEngine
Call caoEng.Workspaces.Remove(caoWs.Index)
' Clear CaoWorkspace
Set caoWs = Nothing
' Clear CaoEngine
Set caoEng = Nothing
```

End Sub

4.1.1.1. Pre-processing

To communicate with the instrument, proceed as follows:

- (1) Prepare a variable to hold the object. The objects required to connect to the controller are CaoEngine object, CaoWorkspace object, and CaoController object. CaoWorkspace object does not need to have a variable to obtain CaoController object from CaoWorkspaces. You will also need a CaoVariable for accessing the variable. The following is a sample code for VB6.

```
' Variables for CaoEngine Objects
Dim caoEng As CaoEngine
' Variables for CaoWorkspace Objects
Dim caoWs As CaoWorkspace
' Variables for CaoController Objects
Dim caoCtrl As CaoController
' Variables for CaoVariable Objects
Dim caoVrl As CaoVariable
```

- (2) Creates a CaoEngine object. CaoEngine object is generated using the New keyword.

```
' Generate CaoEngine
Set caoEng = New CaoEngine
```

- (3) Gets or generates a CaoWorkspace object. When you create a CaoEngine object, it defaults to one CaoWorkspaces object and one object. The following is a sample code/default CaoWorkspace for creating a new CaoWorkspace.

```
' Generate CaoWorkspace
Set caoWs = caoEng.AddWorkspace("Workspace", "")
```

- (4) Create a CaoController object. To generate a CaoController object, set the provider name to use and the parameters to use. For LK-G3000LkIF providers, optionally specify the response time from the API. The following is a code example:

```
' Generate CaoController
Set caoCtrl = caoWs.AddController("LKG3000LkIF", _
```

```
"CaoProv.KEYENCE.LK-G3000LkIF", _
"" , _
"Timeout=1000")
```

- (5) Creates a CaoVariable. Create a CaoVariable for the variable you want to retrieve. The following are examples of codes that generate variable objects that access OUT1 and OUT2 measurements.

```
' Generate CaoVariable
Set caoVrl = caoCtrl.AddVariable("@CALCDATA", "")
```

4.1.1.2. Obtain OUT1 and OUT2 readings

To get OUT1 and OUT2 metrics, see Value Properties for CaoVariable Objects. Variables must be provided for each measurement. The following is a code example:

```
' Retrieving Values
Dim values As Variant
Values = caoVrl.value

If Not IsEmpty(values) Then
    ' OUT1 readings
    Dim value1 As Variant
    Value1 = values(0)
    ' Valid data or not
    Dim validData1 As Long
    ValidData1 = value1(0)
    ' Measured value
    Dim fValue1 As Single
    fValue1 = value1(1)

    ' OUT2 readings
    Dim value2 As Variant
    Value2 = values(1)
    ' Valid data or not
    Dim validData2 As Long
    ValidData2 = value2(0)
    ' Measured value
    Dim fValue2 As Single
    fValue2 = value2(1)
End If
```

4.1.1.3. Post-processing

To perform post-processing, you can erase the generated objects and delete the objects that you want to erase from the collection class that manages the objects. The following is a code example:

```
' Remove CaoVariable from CaoController
Call caoCtrl.variables.Remove(caoVrl.Index)
' Clear CaoVariable
Set caoVrl = Nothing
' Remove CaoController from CaoWorkspace
Call caoWs.Controllers.Remove(caoCtrl.Index)
' Clear CaoController
```

```

Set caoCtrl = Nothing
' Remove CaoWorkspace from CaoEngine
Call caoEng.Workspaces.Remove(caoWs.Index)
' Clear CaoWorkspace
Set caoWs = Nothing
' Clear CaoEngine
Set caoEng = Nothing
    
```

4.2. Sample programming for obtaining stored data in a data storage

This example shows a sample program that reads data stored in data storage. Table 4-2 describes the requirements of the sample program, and Fig. 4-2 describes the flow of the sample program.

Table 4-2 Sample program requirements

Requirements	Description
Process Description	Read data stored in the data storage from LK-G3000.

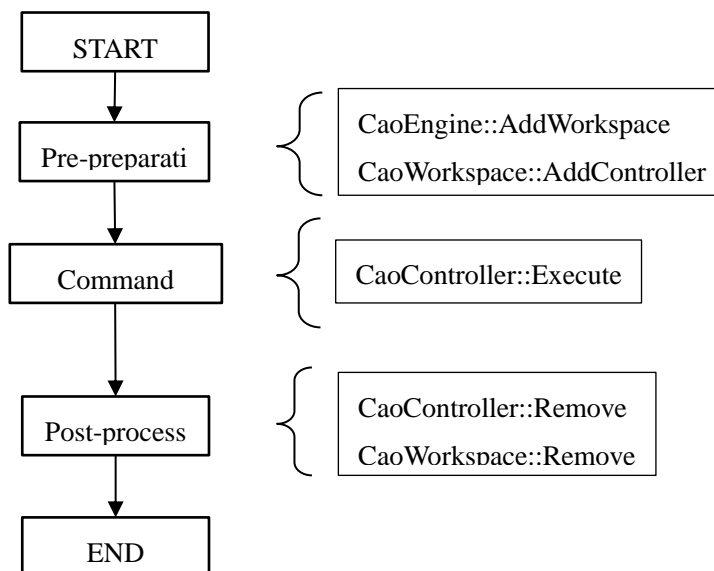


Fig. 4-2 OUT1 and OUT2 Reading Flow

Specific codes are given in the following sections.

4.2.1. Sample program

The following is an overview of the sample program.

Sample	GetDataStorageData.vb
---------------	------------------------------

```

' Object
Dim caoEng As CaoEngine
Dim caoWs As CaoWorkspace
Dim caoCtrl As CaoController

Private Sub Main()
    ' Preparation
    Call Preprocessing
    
```

```

' Specifying Arguments
Dim param As Variant
' Specify the OUT number for element 1 and the number of accumulated data items to be acquired for
element 2.
Param = Array(0, 3)
' Retrieving Values
Dim value As Variant
Value = caoCtrl.Execute("GetDataStorageData", param)

If Not IsEmpty(value) Then
    Dim i As Integer
    ' Repeat for the number of accumulated data items acquired.
    For i = LBound(value) To (UBound(value) - 1)
        ' Stored data
        Dim accumulationData As Variant
        AccumulationData = value(i)
        If Not IsEmpty(accumulationData) Then
            ' Valid data or not
            Dim result As Integer
            Result = accumulationData(LBound(accumulationData))
            ' Measured value
            Dim fValue As Single
            fValue = accumulationData(UBound(accumulationData))
        End If
    Next i
    ' Number of accumulated data items acquired
    Dim readDataCnt As Integer
    ReadDataCnt = value(UBound(value))
End If

' Post-processing
Call Postprocessing
End Sub

' Preparation method
Private Sub Preprocessing()
    ' Generate CaoEngine
    Set caoEng = New CaoEngine
    ' Generate CaoWorkspace
    Set caoWs = caoEng.AddWorkspace("Workspace", "")
    ' Generate CaoController
    Set caoCtrl = caoWs.AddController("LKG3000LkIF", _
        "CaoProv.KEYENCE.LK-G3000LkIF", _
        "", _
        "Timeout=1000")
End Sub

' Post-processing method
Private Sub Postprocessing()
    ' Remove CaoController from CaoWorkspace
    Call caoWs.Controllers.Remove(caoCtrl.Index)
    ' Clear CaoController
    Set caoCtrl = Nothing
    ' Remove CaoWorkspace from CaoEngine
    Call caoEng.Workspaces.Remove(caoWs.Index)

```

```
' Clear CaoWorkspace
Set caoWs = Nothing
' Clear CaoEngine
Set caoEng = Nothing
End Sub
```

4.2.1.1. Pre-processing

To communicate with the instrument, proceed as follows:

- (1) Prepare a variable to hold the object. The objects required to connect to the controller are CaoEngine object, CaoWorkspace object, and CaoController object. CaoWorkspace object does not need to have a variable to obtain CaoController object from CaoWorkspaces. The following is a sample code for VB6.

```
' Variables for CaoEngine Objects
Dim caoEng As CaoEngine
' Variables for CaoWorkspace Objects
Dim caoWs As CaoWorkspace
' Variables for CaoController Objects
Dim caoCtrl As CaoController
```

- (2) Creates a CaoEngine object. CaoEngine object is generated using the New keyword.

```
' Generate CaoEngine
Set caoEng = New CaoEngine
```

- (3) Gets or generates a CaoWorkspace object. When you create a CaoEngine object, it defaults to one CaoWorkspaces object and one object. The following is a sample code/default CaoWorkspace for creating a new CaoWorkspace.

```
' Generate CaoWorkspace
Set caoWs = caoEng.AddWorkspace("Workspace", "")
```

- (4) Create a CaoController object. To generate a CaoController object, set the provider name to use and the parameters to use. For LK-G3000LkIF providers, optionally specify the response time from the API. The following is a code example:

```
' Generate CaoController
Set caoCtrl = caoWs.AddController("LKG3000LkIF", _
    "CaoProv.KEYENCE.LK-G3000LkIF", _
    "", _
    "Timeout=1000")
```

4.2.1.2. Acquisition of accumulated data in data storage

To retrieve the accumulated data in the data storage, execute Execute method of CaoController object. The following is a code example:

```
' Specifying Arguments
Dim param As Variant
' Specify the OUT number for element 1 and the number of accumulated data items to be acquired for element 2.
Param = Array(0, 3)
```

' Retrieving Values

```
Dim value As Variant
```

```
Value = caoCtrl.Execute("GetDataStorageData", param)
```

```
If Not IsEmpty(value) Then
```

```
    Dim i As Integer
```

```
    ' Repeat for the number of accumulated data items acquired.
```

```
    For i = LBound(value) To (UBound(value) - 1)
```

```
        ' Stored data
```

```
        Dim accumulationData As Variant
```

```
        AccumulationData = value(i)
```

```
        If Not IsEmpty(accumulationData) Then
```

```
            ' Valid data or not
```

```
            Dim result As Integer
```

```
            Result = accumulationData(LBound(accumulationData))
```

```
            ' Measured value
```

```
            Dim fValue As Single
```

```
            fValue = accumulationData(UBound(accumulationData))
```

```
        End If
```

```
    Next i
```

```
    ' Number of accumulated data items acquired
```

```
    Dim readDataCnt As Integer
```

```
    ReadDataCnt = value(UBound(value))
```

```
End If
```

4.2.1.3. Post-processing

To perform post-processing, you can erase the generated objects and delete the objects that you want to erase from the collection class that manages the objects. The following is a code example:

```
' Remove CaoController from CaoWorkspace
```

```
Call caoWs.Controllers.Remove(caoCtrl.Index)
```

```
' Clear CaoController
```

```
Set caoCtrl = Nothing
```

```
' Remove CaoWorkspace from CaoEngine
```

```
Call caoEng.Workspaces.Remove(caoWs.Index)
```

```
' Clear CaoWorkspace
```

```
Set caoWs = Nothing
```

```
' Clear CaoEngine
```

```
Set caoEng = Nothing
```

5. LK-G3000LkIF Provider Error Codes

This provider does not have its own error code, but there is an error code when the API terminates abnormally. (Refer to Table 5-1 Error Code Table.)

For information about common ORiN2 errors, see the Error Codes section of ORiN2 Programming Guide ([Link](#)).C:\¥ORiN2¥CAO¥Doc¥ORiN2_ProgrammersGuide_en.pdf

Table 5-1 Error Code Table

Error Number	Description
0x80100001	When the API terminates abnormally

6. Appendix

Appendix A. API correspondence table

CaoController::Execute

Command name	API function name
SetMode	LKIF_SetMode
GetCalcData	LKIF_GetCalcData
SetTiming	LKIF_SetTiming
SetZero	LKIF_SetZero
SetReset	LKIF_SetReset
SetPanelLock	LKIF_SetPanelLock
SetProgramNo	LKIF_SetProgramNo
GetProgramNo	LKIF_GetProgramNo
GetFigureData	LKIF_GetFigureData
ClearFigureData	LKIF_ClearFigureData
StartDataStorage	LKIF_DataStorageStart
StopDataStorage	LKIF_DataStorageStop
ClearDataStorage	LKIF_DataStorageInit
GetDataStorageData	LKIF_DataStorageGetData
GetDataStorageStatus	LKIF_DataStorageGetStatus
GetLight	LKIF_GetLight
SetPanel	LKIF_SetPanel
GetPanel	LKIF_GetPanel
SetTolerance	LKIF_SetTolerance
GetTolerance	LKIF_GetTolerance
SetAbleMode	LKIF_SetAbleMode
GetAbleMode	LKIF_GetAbleMode
SetAbleMinMax	LKIF_SetAbleMinMax
GetAbleMinMax	LKIF_GetAbleMinMax
SetMeasureMode	LKIF_SetMeasureMode
GetMeasureMode	LKIF_GetMeasureMode
SetNumAlarm	LKIF_SetNumAlarm
GetNumAlarm	LKIF_GetNumAlarm
SetAlarmLevel	LKIF_SetAlarmLevel
GetAlarmLevel	LKIF_GetAlarmLevel

StartABLE	LKIF_AbleStart
StopABLE	LKIF_AbleStop
CancelABLE	LKIF_AbleCancel
SetReflectionMode	LKIF_SetReflectionMode
GetReflectionMode	LKIF_GetReflectionMode
SetCalcMethod	LKIF_SetCalcMethod
GetCalcMethod	LKIF_GetCalcMethod
SetScaling	LKIF_SetScaling
GetScaling	LKIF_GetScaling
SetFilterMode	LKIF_SetFilterMode
GetFilterMode	LKIF_GetFilterMode
SetAverage	LKIF_SetAverage
GetAverage	LKIF_GetAverage
SetCutOffFrequency	LKIF_SetCutOffFrequency
GetCutOffFrequency	LKIF_GetCutOffFrequency
SetTriggerMode	LKIF_SetTriggerMode
GetTriggerMode	LKIF_GetTriggerMode
SetOffset	LKIF_SetOffset
GetOffset	LKIF_GetOffset
SetAnalogScaling	LKIF_SetAnalogScaling
GetAnalogScaling	LKIF_GetAnalogScaling
SetCalcMode	LKIF_SetCalcMode
GetCalcMode	LKIF_GetCalcMode
SetDisplayUnit	LKIF_SetDisplayUnit
GetDisplayUnit	LKIF_GetDisplayUnit
SetAnalogThrough	LKIF_SetAnalogThrough
GetAnalogThrough	LKIF_GetAnalogThrough
SetDataStorage	LKIF_SetDataStorage
GetDataStorage	LKIF_GetDataStorage
SetSamplingCycle	LKIF_SetSamplingCycle
GetSamplingCycle	LKIF_GetSamplingCycle
SetMutualInterPrev	LKIF_SetMutualInterferencePrevention
GetMutualInterPrev	LKIF_GetMutualInterferencePrevention
SetTimingSync	LKIF_SetTimingSynchronization
GetTimingSync	LKIF_GetTimingSynchronization
SetTolCompOutputFormat	LKIF_SetToleranceComparatorOutputFormat

GetToleCompOutputFormat	LKIF_GetToleranceComparatorOutputFormat
SetStorobeTime	LKIF_SetStorobeTime
GetStorobeTime	LKIF_GetStorobeTime

CaoVariable

Variable name	Get_Value	Set_Value
@CALCDATA	LKIF_GetCalcData	---
RECEIVED_WAVEFROM<??>	LKIF_GetLight	---