

KEYENCE

LK-G5000 provider

Version 1.0.0

User's guide

June 26, 2019

Remarks:

Do not do the setting change by other applications etc. while it connects it with the equipment in the provider.

This document is translated into English using machine translation.

Contents

1. Introduction.....	7
2. Outline of provider	8
2.1. Outline.....	8
2.2. Method property	9
2.2.1. CaoWorkspace::AddController method	9
2.2.2. CaoController::Execute method	9
2.3. Error code	9
3. Command reference.....	11
3.1. System control	11
3.1.1. CaoController::Execute("GetCalcDataSingle") command	11
3.1.2. CaoController::Execute("GetCalcDataMulti ") command.....	11
3.1.3. CaoController::Execute("GetCalcDataALL") command	13
3.1.4. CaoController::Execute("SetTimingSingle") command	13
3.1.5. CaoController::Execute("SetTimingMulti") command	14
3.1.6. CaoController::Execute("SetTimingSync") command	15
3.1.7. CaoController::Execute("SetZeroSingle") command	15
3.1.8. CaoController::Execute("SetZeroMulti") command.....	16
3.1.9. CaoController::Execute("SetZeroSync") command	17
3.1.10. CaoController::Execute("SetResetMulti") command.....	17
3.1.11. CaoController::Execute("SetResetSync") command	18
3.1.12. CaoController::Execute("SetPanelLock") command.....	18
3.1.13. CaoController::Execute("SetProgramNo") command	19
3.1.14. CaoController::Execute("GetProgramNo") command.....	19
3.1.15. CaoController::Execute("DataStorageStart") command	19
3.1.16. CaoController::Execute("DataStorageStop") command	20
3.1.17. CaoController::Execute("DataStorageGetData") command	20
3.1.18. CaoController::Execute("DataStorageInit") command	21
3.1.19. CaoController::Execute("DataStorageGetStatus") command.....	21
3.1.20. CaoController::Execute("GetLight") command	22
3.2. Measurement change	23
3.2.1. CaoController::Execute("SetPanel") command	23
3.2.2. CaoController::Execute("SetTolerance") command	23
3.2.3. CaoController::Execute("SetAbleMode") command	24
3.2.4. CaoController::Execute("SetAbleMinMax") command.....	24
3.2.5. CaoController::Execute("SetMeasureMode") command.....	25
3.2.6. CaoController::Execute("SetBasicPoint") command.....	26

3.2.7. CaoController::Execute("SetNumAlarm") command.....	26
3.2.8. CaoController::Execute("SetNumRecovery") command.....	27
3.2.9. CaoController::Execute("SetAlarmLevel ") command.....	27
3.2.10. CaoController::Execute("AbleStart") command	27
3.2.11. CaoController::Execute("AbleStop") command.....	28
3.2.12. CaoController::Execute("AbleCancel") command.....	28
3.2.13. CaoController::Execute("SetReflectionMode") command.....	29
3.2.14. CaoController::Execute("SetMask") command	30
3.2.15. CaoController::Execute("SetMedian") command.....	30
3.2.16. CaoController::Execute("SetLaserCtrlGroup") command	31
3.2.17. CaoController::Execute("SetRange") command	31
3.2.18. CaoController::Execute("SetMutualInterferencePreventionGroup") command.....	32
3.2.19. CaoController::Execute("SetCalcMethod") command	32
3.2.20. CaoController::Execute("SetCalcTarget") command	33
3.2.21. CaoController::Execute("SetOutAddSub") command	34
3.2.22. CaoController::Execute("SetOutOperation") command	34
3.2.23. CaoController::Execute("SetScaling") command.....	35
3.2.24. CaoController::Execute("SetFilter") command.....	36
3.2.25. CaoController::Execute("SetTriggerMode") command	37
3.2.26. CaoController::Execute("SetOffset") command	38
3.2.27. CaoController::Execute("SetCalcMode") command	38
3.2.28. CaoController::Execute("SetAnalogScaling") command.....	39
3.2.29. CaoController::Execute("SetDisplayUnit") command	39
3.2.30. CaoController::Execute("SetMeasureType") command.....	41
3.2.31. CaoController::Execute("SetSynchronization") command	41
3.2.32. CaoController::Execute("SetStorageTarget") command	43
3.2.33. CaoController::Execute("SetSamplingCycle") command.....	43
3.2.34. CaoController::Execute("SetMutualInterferencePrevention") command.....	44
3.2.35. CaoController::Execute("SetToleranceComparatorOutputFormat") command	44
3.2.36. CaoController::Execute("SetStrobeTime") command	45
3.2.37. CaoController::Execute("SetDataStorage") command	45
3.2.38. CaoController::Execute("SetAnalogChannel") command	46
3.2.39. CaoController::Execute("SetAlarmOutputForm") command	47
3.2.40. CaoController::Execute("SetNumOfUsedHeads") command	47
3.2.41. CaoController::Execute("SetNumOfUsedOut") command	47
3.2.42. CaoController::Execute("GetPanel") command	48
3.2.43. CaoController::Execute("GetTolerance") command.....	48

3.2.44. CaoController::Execute("GetAbleMode") command	49
3.2.45. CaoController::Execute("GetAbleMinMax") command	49
3.2.46. CaoController::Execute("GetMeasureMode") command	50
3.2.47. CaoController::Execute("GetBasicPoint") command	50
3.2.48. CaoController::Execute("GetNumAlarm") command	51
3.2.49. CaoController::Execute("GetNumRecovery") command	51
3.2.50. CaoController::Execute("GetAlarmLevel") command	52
3.2.51. CaoController::Execute("GetReflectionMode") command	52
3.2.52. CaoController::Execute("GetMask") command.....	53
3.2.53. CaoController::Execute("GetMedian") command	53
3.2.54. CaoController::Execute("GetLaserCtrlGroup") command.....	54
3.2.55. CaoController::Execute("GetRange") command.....	55
3.2.56. CaoController::Execute("GetMutualInterferencePreventionGroup") command	55
3.2.57. CaoController::Execute("GetCalcTarget") command	56
3.2.58. CaoController::Execute("GetOutAddSub") command.....	56
3.2.59. CaoController::Execute("GetOutOperation") command.....	57
3.2.60. CaoController::Execute("GetScaling") command	58
3.2.61. CaoController::Execute("GetFilter") command	58
3.2.62. CaoController::Execute("GetTriggerMode") command	60
3.2.63. CaoController::Execute("GetOffset") command.....	60
3.2.64. CaoController::Execute("GetCalcMode") command	61
3.2.65. CaoController::Execute("GetAnalogScaling") command	61
3.2.66. CaoController::Execute("GetDisplayUnit") command	62
3.2.67. CaoController::Execute("GetMeasureType") command	63
3.2.68. CaoController::Execute("GetSynchronization") command.....	64
3.2.69. CaoController::Execute("GetStorageTarget") command.....	64
3.2.70. CaoController::Execute("GetSamplingCycle") command	65
3.2.71. CaoController::Execute("GetMutualInterferencePrevention") command	65
3.2.72. CaoController::Execute("GetToleranceComparatorOutputFormat") command.....	66
3.2.73. CaoController::Execute("GetDataStorage") command	66
3.2.74. CaoController::Execute("GetAnalogChannel") command.....	67
3.2.75. CaoController::Execute("GetAlarmOutputForm") command.....	67
3.2.76. CaoController::Execute("GetNumOfUsedHeads") command	68
3.2.77. CaoController::Execute("GetNumOfUsedOut") command.....	68
3.2.78. CaoController::Execute("GetNumOfUsedAnalogCh") command.....	69
3.3. Mode change command.....	69
3.3.1. CaoController::Execute("StartMeasure") command	69

3.3.2. CaoController::Execute("StopMeasure") command.....70

1. Introduction

This book is an user's guide of the LK-G5000 provider that is the CAO provider of made by the KEYENCE company super-high speed and highly accurate laser displacement sensor LK-G5000 series.

The LK-G5000 provider communicates with the LK-G5000 series controller with whom Ethernet was connected, and acquires the control of the measurement and the result of a measurement.

In this book, it explains the function of this LK-G5000 provider and the mounting method.

2. Outline of provider

2.1. Outline

The LK-G5000 provider is offering the telecommunication facility by executing the command of the communication made by KEYENCE company library by CaoController::Execute.

Table2-1LK-G5000 provider

File name	CaoProvKEYENCELKG5000.dll
ProgID	CaoProv.KEYENCE.LK-G5000
Registry registration	regsvr32 CaoProvKEYENCELKG5000.dll
Blotting out of registry registration	regsvr32 /u CaoProvKEYENCELKG5000.dll

Table2-2Dependence module

No.	File name	Explanation
1	LkIF2.DLL	It is a main body of DLL.
2	CmnLib.dll	It is necessary for operating DLL.
3	KeyUsbDrv.dll	It is necessary for operating DLL.

2.2. Method property

2.2.1. CaoWorkspace::AddController method

Execute initialization of the library of KEYENCE and connected processing with the product.

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

bstrCtrlName : [in] CAO controller name
 bstrProvName : [in] Provider name. fixed value ="CaoProv.KEYENCE.LK-G5000"
 bstrPcName : [in] Execution machine name of provider
 bstrOption : [in] optional character string

The list specified for an optional character string is shown as follows.

Table2-3Optional character string of CaoWorkspace::AddController

Option	Explanation
IPAddress=< IP address >	Indispensability. Specify Internet Protocol address of the controller of the displacement sensor to be connected. Example:IPAddress=192.168.0.2

Example

```
// Example program
Dim caoCtrl As Object
set caoCtrl = caoWs.AddController ("LKG", "CaoProv.KEYENCE.LK-G5000", "",
"IPAddress=192.168.0.2")
```

2.2.2. CaoController::Execute method

Send and receive the command. Specify the command name in the first argument and specify the parameter of the command for the second argument. Please refer to chapter 3.Command reference for details.

Format Execute(< bstrCommandName:BSTR >[, < vntParam:VT_VARIANT>])

bstrCommandName : [in] Command name
 vntParam : [in] Argument of command

2.3. Error code

In the LK-G5000 provider, the following and peculiar the error code is defined.

About the ORiN2 commonness error, refer to the chapter of the error code of "ORiN2 Programming guide".

Table2-4Peculiar error code

Error number	Explanation
0x80101001	Command error
0x80101002	Command length error
0x80101003	Time-out
0x80101004	Checksum error
0x80101005	State error
0x80101006	Other error
0x80101007	Parameter error
0x80101008	Number of use heads and OUT numerical exaggerated
0x80101009	Number of use heads and OUT numerical exaggerated
0x8010100A	Specify OUT that cannot be specified for the operation object.
0x8010100B	There is no OUT specified for the operation object.
0x8010100C	Sampling period that cannot be used
0x8010100D	The main unit is abnormal.
0x8010100E	A set value is abnormal.
0x80102000	It fails in the opening of the device.
0x80102001	The device has not been opened.
0x80102002	Command transmission error
0x80102003	Response reception error
0x80102004	Time-out
0x80102005	Data none
0x80102006	Empty memory none
0x80102007	There is a possibility of the disconnection.
0x80102008	Undefined error

3. Command reference

This chapter explains each command of the CaoController::Execute method. Refer to "LK-G5000 series communication library reference manual" of the KEYENCE Co. for detailed operation of each command.

3.1. System control

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

Acquire measurements of one specified OUT number.

Format GetCalcDataSingle (<ulOutNo>)

ulOutNo : OUT number in which it wants to acquire in (VT_UI4)
0 - 11

Return value : Out result data (VT_VARIANT | VT_ARRAY)

Index	Content
0	The 0th OUT number (VT_I4)
1	The 0th Flag (VT_I4) 0 :The valid data 1 :+ cooking stove exaggerated 2 :- Cooking stove exaggerated 3 :Judgment standby 4 :Alarm 5 :Invalidity (error etc.)
2	The 0th measurements (VT_R4)

Example

```
// Example program
Dim retVal As Variant
retVal = caoCtrl.Execute("GetCalcDataSingle", 5)
```

3.1.2. CaoController::Execute("GetCalcDataMulti") command

Acquire two or more OUT numbers and acquire the value.

Format GetCalcDataMulti (<ulOutNo>)

ulOutNo : Logical add of OUT number in which it wants to acquire in (VT_UI4)

0x0001, // OUT1

0x0002, // OUT2

0x0004, // OUT3

0x0008, // OUT4

0x0010, // OUT5

0x0020, // OUT6

0x0040, // OUT7

0x0080, // OUT8

0x0100, // OUT9

0x0200, // OUT10

0x0400, // OUT11

0x0800, // OUT12

Return value : Out result data (VT_VARIANT | VT_ARRAY)

Index	Content
0	The 0th OUT number (VT_I4)
1	The 0th Flag (VT_I4) 0 :The valid data 1 :+ cooking stove exaggerated 2 :- Cooking stove exaggerated 3 :Judgment standby 4 :Alarm 5 :Invalidity (error etc.)
2	The 0th measurements (VT_R4)
...	
0 + n*3	Nth OUT number (VT_I4)
1 + n*3	Nth Flag (VT_I4)
2 + n*3	Nth measurements (VT_R4)

Example

```
// Example program
Dim retVal As Variant
retVal = caoCtrl.Execute("GetCalcDataMulti", 5) // OUT1 and OUT3
```

3.1.3. CaoController::Execute("GetCalcDataALL") command

Acquire all the OUT numbers and acquire the value.

Format GetCalcDataALL()

Argument : In none

Return value : GetCalcDataMulti

Index	Content
0	The 0th OUT number (VT_I4)
1	The 0th Flag (VT_I4) 0 :The valid data 1 :+ cooking stove exaggerated 2 :- Cooking stove exaggerated 3 :Judgment standby 4 :Alarm 5 :Invalidity (error etc.)
2	The 0th measurements (VT_R4)
...	
33	The 11th OUT number (VT_I4)
34	The 12th Flag (VT_I4)
35	The 13th measurements (VT_R4)

Example

```
// Example program
Dim retVal As Variant
retVal = caoCtrl.Execute("GetCalcDataALL")
```

3.1.4. CaoController::Execute("SetTimingSingle") command

Turn timing on and off for one specified OUT number.

Format SetTimingSingle (<ulOutNo>, <ucTiming>)

ulOutNo	:	OUT number in which it wants to acquire in (VT_UI4) 0 - 11
ucTiming	:	In timing (VT_UI1) 0: OFF 1: ON
Return value	:	Out none

Example

```
// Example program
call caoCtrl.Execute("SetTimingSingle", Array(5, 0))
```

3.1.5. CaoController::Execute("SetTimingMulti") command

Turn timing on and off for two or more specified OUT numbers.

Format SetTimingMulti (<ulOutNo>, <ucTiming>)

ulOutNo	:	Logical add of OUT number in which it wants to acquire in (VT_UI4) 0x0001, // OUT1 0x0002, // OUT2 0x0004, // OUT3 0x0008, // OUT4 0x0010, // OUT5 0x0020, // OUT6 0x0040, // OUT7 0x0080, // OUT8 0x0100, // OUT9 0x0200, // OUT10 0x0400, // OUT11 0x0800, // OUT12
ucTiming	:	In timing (VT_UI1) 0: OFF 1: ON

Example

```
// Example program
call caoCtrl.Execute("SetTimingMulti", Array(5, 0))
```

3.1.6. CaoController::Execute("SetTimingSync") command

Turn the timing of OUT synchronously set on and off.

Format SetTimingSync (<ucTiming>)

ucTiming : In timing (VT_UI1)
 0: OFF
 1: ON

Return value : Out none

Example

```
// Example program
call ctrl.Execute("SetTimingSync")
```

3.1.7. CaoController::Execute("SetZeroSingle") command

Turn auto 0 on and off for one specified OUT number.

Format SetTimingSingle (<ulOutNo>, <ucFlag>)

ulOutNo : OUT number in which it wants to acquire in (VT_UI4)
 0 - 11

ucFlag : ON/OFF (VT_UI1) of in auto 0 it
 0 : OFF
 1 : ON

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetTimingSingle", Array(5, 1))
```

3.1.8. CaoController::Execute("SetZeroMulti") command

Turn auto 0 of two or more specified OUT numbers on and off.

Format SetZeroMulti(<ulOutNo>)

ulOutNo : Logical add of OUT number in which it wants to acquire in (VT_UI4)

- 0x0001, // OUT1
- 0x0002, // OUT2
- 0x0004, // OUT3
- 0x0008, // OUT4
- 0x0010, // OUT5
- 0x0020, // OUT6
- 0x0040, // OUT7
- 0x0080, // OUT8
- 0x0100, // OUT9
- 0x0200, // OUT10
- 0x0400, // OUT11
- 0x0800, // OUT12

ucFlag : ON/OFF (VT_UI1) of in auto 0 it

- 0 : OFF
- 1 : ON

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetZeroMulti", Array(5, 1))
```

3.1.9. CaoController::Execute("SetZeroSync") command

Turn auto 0 of OUT synchronously set on and off.

Format SetZeroSync (<ucFlag>)

ucFlag : ON(1)/OFF(0) of in auto 0 it (VT_UI1)
 0 : OFF
 1 : ON
 Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetZeroSync", 0)
```

3.1.10. CaoController::Execute("SetResetMulti") command

Input reset to two or more specified OUT numbers.

Format SetResetMulti (<ulOutNo>,)

ulOutNo : Logical add of OUT number in which it wants to reset in (VT_UI4)
 0x0001, // OUT1
 0x0002, // OUT2
 0x0004, // OUT3
 0x0008, // OUT4
 0x0010, // OUT5
 0x0020, // OUT6
 0x0040, // OUT7
 0x0080, // OUT8
 0x0100, // OUT9
 0x0200, // OUT10
 0x0400, // OUT11
 0x0800, // OUT12

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetResetMulti", 5)
```

3.1.11. CaoController::Execute("SetResetSync") command

Input reset to OUT synchronously set.

Format SetResetSync ()

Argument : In none

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetResetSync")
```

3.1.12. CaoController::Execute("SetPanelLock") command

Set ON/OFF of the panel lock.

Format SetPanelLock (< ucFlag >)

ucFlag : ON(1)/OFF(0) of in panel lock (VT_UI1)
 0 : OFF
 1 : ON

Return value : Out none

Example

```
// Example program
```

```
call caoCtrl.Execute("SetPanelLock", 0)
```

3.1.13. CaoController::Execute("SetProgramNo") command

Switch to the number in which the program number is specified.

Format SetProgramNo (< ucFlag >)

ucFlag	:	In program number (VT_UI4)
		0 - 7
Return value	:	Out none

Example

```
// Example program
call caoCtrl.Execute("SetProgramNo", 0)
```

3.1.14. CaoController::Execute("GetProgramNo") command

Acquire a present program number.

Format GetProgramNo ()

Argument	:	In none
Return value	:	Out program number (VT_UI4)
		0 - 7

Example

```
// Example program
Dim PrgNo as Variant
PrgNo = caoCtrl.Execute("GetProgramNo")
```

3.1.15. CaoController::Execute("DataStorageStart") command

Begin the data storage.

Format DataStorageStart ()

Argument : In none

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("DataStorageStart")
```

3.1.16. GaoController::Execute("DataStorageStop") command

Stop the data storage.

Format DataStorageStop ()

Argument : In none

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("DataStorageStop", 0)
```

3.1.17. GaoController::Execute("DataStorageGetData") command

Acquire the accumulation data of the data storage.

Format DataStorageGetData (< ulOutNo >, < ulNum >)

ulOutNo : OUT number of data to acquire in (VT_UI4)
0 - 11

ulNum : Number of data to acquire in
1 - 1200000

Return value : Data to acquire out (VT_VARIANT | VT_ARRAY)

Index	Content
0	The 0th Flag (VT_I4)

	0 :The valid data 1 :+ cooking stove exaggerated 2 :- Cooking stove exaggerated 3 :Judgment standby 4 :Alarm 5 :Invalidity (error etc.)
1	The 0th measurements (VT_R4)
...	
0 + n*2	Nth OUT number (VT_I4)
1 + n*2	Nth Flag (VT_I4)

Example

```
// Example program
Dim vntRet as Variant
vntRet = caoCtrl.Execute("DataStorageGetData", 0)
```

3.1.18. CaoController::Execute("DataStorageInit") command

Clear the accumulation data of the data storage.

Format DataStorageInit (< ucFlag >)

Argument : In none
Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("DataStorageInit")
```

3.1.19. CaoController::Execute("DataStorageGetStatus") command

Acquire the accumulation of the data storage.

Format DataStorageGetStatus ()

Argument : In none

Return value : State of accumulation of out data storage (VT_VARIANT | VT_ARRAY)

Index	Content
0	State of accumulation (VT_I4) 0 :It is stopping. 1 :It is accumulating.
1	Number of data that has been accumulated (VT_I4)

Example

```
// Example program
Dim vntRet as Variant
vntRet = caoCtrl.Execute("DataStorageGetStatus")
```

3.1.20. CaoController::Execute("GetLight") command

Acquire the receiving light wave type data of the specified head number.

Format GetLight (<ulHeadNo>, <ulPeakNo>)

ulHeadNo : Head number of data to acquire in (VT_UL4)
0 - 11

ulPeakNo : Peak number at in transparent body 2 (VT_UL4)
0 - 3

Return value : Out receiving light wave type data (VT_VARIANT | VT_ARRAY)

Index	Content
0	Location of measurement of peak number 0 (VT_I4)
...	
3	Location of measurement of peak number 3 (VT_I4)
4	Waveform data 0 (VT_UI1)
...	
2051	Waveform data 2047 (VT_UI1)

Example

```
// Example program
Dim vntRet as Variant
vntRet = caoCtrl.Execute("GetLight", Array(5, 0))
```

3.2. Measurement change

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

Set the OUT number displayed in the upper row and the the lower of the display panel.

Format SetPanel (<IUpperOUTNo>, <ILowerOUTNo >)

IUpperOUTNo	:	OUT number displayed in in upper row (VT_I4) -1 :Non-display 0 - 11: OUT number
ILowerOUTNo	:	OUT number displayed in in the lower (VT_I4) -1 :Non-display 0 - 11: OUT number
Return value	:	Out none

Example

```
// Example program
call caoCtrl.Execute("SetPanel", Array(5, 7))
```

3.2.2. CaoController::Execute("SetTolerance") command

It establishes tolerance.

Format SetTolerance (<IOUTNo>, <IUpperLimit>, <ILowerLimit>, <IToleranceHysteresis>)

IOUTNo	:	In OUT number (VT_I4) 0 - 11: OUT number
IUpperLimit	:	In allowance upper bound value (VT_I4) -999999 - 999999

lLowerLimit	In allowance lower bound value (VT_I4)
	-999999 - 999999
lToleranceHysteresis	In allowance hysteresis (VT_I4)
	0 - 999999
Return value	: Out none

Example

```
// Example program
call caoCtrl.Execute("SetTolerance", Array(5, -10, 100, 50))
```

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

Set the ABLE mode.

Format SetAbleMode (<ulHeadNo>, < ulABLEMode >)

ulHeadNo	: In head number (VT_UI4)
	0 - 11
ulABLEMode	In ABLE mode (VT_UI4)
	0 :Automatic operation
	1 :Manual
Return value	: Out none

Example

```
// Example program
call caoCtrl.Execute("SetAbleMode", Array(0, 0))
```

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

Set the ABLE sphere of control.

Format SetAbleMinMax (< ulHeadNo >, < ulMinABLE >, <ulMaxABLE>)

ulHeadNo	:	In head number (VT_UI4) 0 - 11
ulMinABLE	:	In Minimum value of ABLE (VT_UI4) 1-99
ulMaxABLE	:	In ABLE maximum value (VT_UI4) 1-99
Return value	:	Out none

Example

```
// Example program
call caoCtrl.Execute("SetAbleMinMax", Array(0, 1, 50))
```

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

Set the measurement mode.

Format SetMeasureMode (< ulHeadNo >, < ulMode >)

ulHeadNo	:	In head number (VT_UI4) 0 - 11
ulMode	:	In measurement mode (VT_UI4) 0 :Standard 1 :Semitransparent body 2 :Transparent medium 3 :Transparent body 2 4 :Multiple reflection body Luster resin
Return value	:	Out none

Example

```
// Example program
call caoCtrl.Execute("SetMeasureMode", Array(0, 0))
```

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

Set the base point.

Format SetBasicPoint (< ulHeadNo >, < ulPoint >)

ulHeadNo	:	In head number (VT_UI4)
		0 - 11
ulPoint	:	In base point (VT_UI4)
		0 : NEAR
		1 : FAR
Return value	:	Out none

Example

```
// Example program  
call caoCtrl.Execute("SetBasicPoint", Array(0, 0))
```

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

Set the alarm processing frequency.

Format SetNumAlarm (< ulHeadNo >, < ulCount >)

ulHeadNo	:	In head number (VT_UI4)
		0 - 11
ulCount	:	In alarm processing frequency (VT_UI4)
		0 - 9999
Return value	:	Out none

Example

```
// Example program  
call caoCtrl.Execute("SetNumAlarm", Array(0, 0))
```

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

Set the alarm processing frequency.

Format SetNumRecovery (< ulHeadNo >, < ulCount >)

ulHeadNo	:	In head number (VT_UI4)
		0 - 11
ulCount	:	In alarm return frequency (VT_UI4)
		0 - 9999
Return value	:	Out none

Example

```
// Example program
call caoCtrl.Execute("SetNumRecovery ", Array(0,0))
```

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

Set the alarm processing frequency.

Format SetAlarmLevel (< ulHeadNo >, < ulLevel >)

ulHeadNo	:	In head number (VT_UI4)
		0 - 11
ulLevel	:	In alarm level (VT_UI4)
		0 - 9
Return value	:	Out none

Example

```
// Example program
call caoCtrl.Execute("SetAlarmLevel", Array(0,0))
```

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

Begin the ABLE tuning.

Format AbleStart (< ulHeadNo >, < ulCount >)

ulHeadNo : In head number (VT_UI4)
0 - 11

Return value : Out none

Example

```
// Example program  
call caoCtrl.Execute("AbleStart", 0)
```

3.2.11. GaoController::Execute("AbleStop") command

ABLE tuning end.

Format AbleStop ()

Argument : In none

Return value : Out none

Example

```
// Example program  
call caoCtrl.Execute("AbleStop")
```

3.2.12. GaoController::Execute("AbleCancel") command

ABLE tuning discontinuance.

Format AbleCancel ()

Argument : In none

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("AbleCancel")
```

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

Set the installation mode.

Format SetReflectionMode (< ulHeadNo >, < ulMode >)

ulHeadNo : In head number (VT_UI4)
0 - 11

ulMode : In installation mode (VT_UI4)
0 :Diffuse reflection
1 :Regular reflection

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetReflectionMode", Array(0,0))
```

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

Set the mask.

Format SetMask (< ulHeadNo >, <bFlag>, <lMask1>, <lMask2>)

ulHeadNo	:	In head number (VT_UI4) 0 - 11
bFlag	:	ON/OFF (VT_BOOL) of in mask FALSE : OFF TRUE : ON
lMask1	:	In mask boundary value 1 (VT_I4) -999999 - 999999
lMask2	:	In mask boundary value 2 (VT_I4) -999999 - 999999
Return value	:	Out none

Example

```
// Example program
call caoCtrl.Execute("SetMask", Array(0, 0, 0, 0, 0))
```

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

Set the median.

Format SetMedian (< ulHeadNo >, < ulMedian >)

ulHeadNo	:	In head number (VT_UI4) 0 - 11
ulMedian	:	In median (VT_UI4) 0 : OFF 1 : Seven points 2 : 15 points 3 : 31 points

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetMedian", Array(0, 0))
```

3.2.16. CaoController::Execute("SetLaserCtrlGroup") command

Set the LASER CTRL group.

Format SetLaserCtrlGroup (< ulHeadNo >, < ulGroup >)

ulHeadNo : In head number (VT_UI4)
0 - 11

ulGroup : In LASER CTRL group (VT_UI4)
0 : LASER CTRL 1
1 : LASER CTRL 2

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetLaserCtrlGroup", Array(0, 0))
```

3.2.17. CaoController::Execute("SetRange") command

Set the cooking stove.

Format SetRange (< ulHeadNo >, < ulRange >)

ulHeadNo : In head number (VT_UI4)
0 - 11

ulRange : In median (VT_UI4)
0 : CENTER

1 : FAR
 Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetRange", Array(0,0))
```

3.2.18. CaoController::Execute("SetMutualInterferencePreventionGroup") command

Set the group of the mutual interference prevention.

Format SetMutualInterferencePreventionGroup(< ulHeadNo >, < ulGroup >)

ulHeadNo : In head number (VT_UI4)
 0 - 11

ulGroup : Group of in mutual interference prevention (VT_UI4)
 0 : Group A
 1 : Group B
 2 : Group C

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetMutualInterferencePreventionGroup", Array(0,0))
```

3.2.19. CaoController::Execute("SetCalcMethod") command

Set the median.

Format SetCalcMethod (< ulHeadNo >, < ulMethod >, <ulNo>)

ulHeadNo : In head number (VT_UI4)
 0 - 11

ulMethod	In computing type (VT_UI4)
	0 :Head
	1 : OUT
	2 :Addition
	3 :Subtraction
	4 : AVE
	5 : P-P
	6 : MAX
	7 : MIN
ulNo	Number when in computing type is HEAD, and OUT (VT_UI4)
Return value	: Out none

Example

```
// Example program
call caoCtrl.Execute("SetCalcMethod", Array(0, 0, 0))
```

3.2.20. CaoController::Execute("SetCalcTarget") command

Set the measuring plane.

Format SetCalcTarget (< ulHeadNo >, < ulTarget >)

ulHeadNo	: In head number (VT_UI4)
	0 - 11
ulTarget	In computing type (VT_UI4)
	0 :Peak 1
	1 :Peak 2
	2 :Peak 3
	3 :Peak 4
	4 :Peak ..one..-peak 2
	5 :Peak ..one..-peak 3
	6 :Peak ..one..-peak 4
	7 :Peak ..two..-peak 3
	8 :Peak ..two..-peak 4

9 :Peak ..three..-peak 4
 Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetCalcTarget", Array(0, 0))
```

3.2.21. CaoController::Execute("SetOutAddSub") command

Set the target number of OUT between OUT for the operation addition and subtraction.

Format SetOutAddSub(< ulHeadNo >, < ulOUT1 >, < ulOUT2 >)

ulHeadNo : In head number (VT_UI4)
 0 - 11

ulOUT1 : Side OUT number to which in is added (It is subtracted) (VT_UI4).
 0 - 11

ulOUT2 : In addition (subtraction) side OUT number (VT_UI4)
 0 - 11

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetOutAddSub", Array(0, 0, 1))
```

3.2.22. CaoController::Execute("SetOutOperation") command

Set the target number of OUT between OUT for operation multi paragraph calculation (AVE/P-P/MAX/MIN).

Format SetOutOperation (< ulOUTNo >, < ulOUTNo2>)

ulOUTNo : In OUT number (VT_UI4)
 0 - 11

ulOUTNo2 Targeted OUT number of in (VT_UI4)
 Two or more specification takes LKIF_OUTNO member's OR.

0x0001, //OUT1
 0x0002, //OUT2
 0x0004, //OUT3
 0x0008, //OUT4
 0x0010, //OUT5
 0x0020, //OUT6
 0x0040, //OUT7
 0x0080, //OUT8
 0x0100, //OUT9
 0x0200, //OUT10
 0x0400, //OUT11
 0x0800, //OUT12
 0x0FFF and // all OUT

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetOutOperation", Array(0, 1))
```

3.2.23. CaoController::Execute("SetScaling") command

Set scaling.

Format SetScaling (< IOUtno >, < IInput1>, < IOutput1>, < IInput2>, < IOutput2>)

IOUtno : OUT number in which in is set (VT_UI4)
 0 - 11

IInput1 In input value 1 (VT_I4)
 -999999 - 999999

IOutput1 In indicated value 1 (VT_I4)
 -999999 - 999999

IInput2 In input value 2 (VT_I4)

-999999 - 999999

lOutput2 In indicated value 2 (VT_I4)

-999999 - 999999

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetScaling", Array(0, 0, 0, 0, 0))
```

3.2.24. CaoController::Execute("SetFilter") command

Set the filter mode and the filter parameter.

Format SetFilter (< ulOUTNo >, < ulMode >, < ulParam >)

ulOUTNo : In OUT number (VT_UI4)

0 - 11

ulMode Specify the in filter mode (VT_UI4).

0 :Moving average

1 :Low-pass filter

2 :High-pass filter

ulParam In filter parameter (VT_UI4)

(If it is an average frequency, and LPF·HPF in case of the moving average, it is cutoff frequency.)

For the moving average

0 : Once

1 : 4

2 : 16

3 : 64

4 : 256

5 : 1024

6 : 4096

7 : 16384

8 : 65536

9 : 262144

For LPF·HPF

0 : 3000 Hz

1 : 1000

2 : 300

3 : 100

4 : 30

5 : 10

6 : 3

7 : 1

8 : 0.3

9 : 0.1

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetFilter", Array(0, 0, 0))
```

3.2.25. CaoController::Execute("SetTriggerMode") command

Set the trigger mode.

Format SetTriggerMode (< ulOUTNo >, < ulMode >)

ulOUTNo : In OUT number (VT_UI4)

0 - 11

ulMode : In trigger mode (VT_UI4)

0 : External trigger 1

1 : External trigger 2

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetTriggerMode", Array(0, 0))
```

3.2.26. CaoController::Execute("SetOffset") command

Set the offset.

Format SetOffset (< IOUTNo >, < IOffset >)

IOUTNo	:	In OUT number (VT_I4)
		0 - 11
IOffset	:	In offset (VT_I4)
		-999999 - 999999
Return value	:	Out none

Example

```
// Example program  
call caoCtrl.Execute("SetOffset", Array(0, 0))
```

3.2.27. CaoController::Execute("SetCalcMode") command

Set the measurement mode.

Format SetCalcMode (< ulOUTNo >, < ulMode >)

ulOUTNo	:	In OUT number (VT_UI4)
		0 - 11
ulMode	:	In trigger mode (VT_UI4)
		0 :Standard
		1 :Peak holding
		2 :Bottom holding
		3 :Peak to peak holding
		4 :Sample holding
		5 :Average holding
Return value	:	Out none

Example

```
// Example program
call caoCtrl.Execute("SetCalcMode", Array(0, 0))
```

3.2.28. CaoController::Execute("SetAnalogScaling") command

Set the analog output scaling.

Format SetAnalogScaling (< IOUTNo >, < IInput1 >, < IOutput1 >, < IInput2 >, < IOutput2 >)

IOUTNo	:	In OUT number (VT_I4) 0 - 11
IInput1	:	In input value 1 (VT_I4) -999999 - 999999
IOutput1	:	1 in in output voltage (VT_I4) -10500(mV) - 10500(vm)
IInput2	:	In input value 2 (VT_I4) -999999 - 999999
IOutput2	:	2 in in output voltage (VT_I4) -10500(mV) - 10500(vm)
Return value	:	Out none

Example

```
// Example program
call caoCtrl.Execute("SetAnalogScaling", Array(0, 0, 0, 100, 100))
```

3.2.29. CaoController::Execute("SetDisplayUnit") command

Set the unit of a minimum display.

Format SetDisplayUnit (< uIOUTNo >, < uIMode >)

ulOUTNo	:	In OUT number (VT_UI4)
		0 - 11
ulMode	:	Unit of in display (VT_UI4)
		0 : 0.01mm
		1 : 0.001mm
		2 : 0.0001mm
		3 : 0.00001mm
		4 : 0.1μm
		5 : 0.01μm
		6 : 0.001μm
		0 : 0.1m/s
		1 : 0.01m/s
		2 : 0.001m/s
		3 : 0.1mm/s
		4 : 0.01mm/s
		5 : 0.001mm/s
		6 : 0.0001mm/s
		0 : 0.1km/s ²
		1 : 0.01km/s ²
		2 : 0.001km/s ²
		3 : 0.1m/s ²
		4 : 0.01m/s ²
		5 : 0.001m/s ²
		6 : 0.0001m/s ²
Return value	:	Out none

Example

```
// Example program  
call caoCtrl.Execute("SetDisplayUnit", Array(0,0))
```

3.2.30. GaoController::Execute("SetMeasureType") command

Set the measurement type.

Format SetMeasureType (< ulOUTNo >, < ulMode >)

ulOUTNo	:	In OUT number (VT_UI4)
		0 - 11
ulMode	:	In measurement type (VT_UI4)
		0 :Displacement
		1 :Speed
		1 :Acceleration
Return value	:	Out none

Example

```
// Example program
call caoCtrl.Execute("SetMeasureType", Array(0,0))
```

3.2.31. GaoController::Execute("SetSynchronization") command

Set the timing synchronization.

Format SetSynchronization (< ulOUTNo >, < ulMode >)

ulOUTNo	:	In OUT number (VT_UI4)
		0 - 11
ulMode	:	Whether make to the in accumulation object (VT_UI4) or not-
		0 :Off the subject
		1 :Object
Return value	:	Out none

Example

```
// Example program
call caoCtrl.Execute("SetSynchronization", Array(0,0))
```

3.2.32. GaoController::Execute("SetStorageTarget") command

Set the accumulation object of storage.

Format SetStorageTarget (< ulOUTNo >, < ulMode >)

ulOUTNo	:	In OUT number (VT_UI4)
		0 - 11
ulMode		Whether make to the in accumulation object (VT_UI4) or not-
		0 :Off the subject
		1 :Object
Return value	:	Out none

Example

```
// Example program
call caoCtrl.Execute("SetStorageTarget", Array(0, 0))
```

3.2.33. GaoController::Execute("SetSamplingCycle") command

Set the sampling period.

Format SetSamplingCycle (< ulCycle >)

ulCycle		In sampling period (VT_UI4)
		0 : 2.55us
		1 : 5us
		2 : 10us
		3 : 20us
		4 : 50us
		5 : 100us
		6 : 200us
		7 : 500us
		8 : 1ms

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetSamplingCycle", 3)
```

3.2.34. CaoController::Execute("SetMutualInterferencePrevention") command

Set the mutual interference prevention.

Format SetMutualInterferencePrevention (< ulMode >)

ulMode Specification of in mutual interference prevention (VT_UI4)

0 : OFF

1 : AB-ON

2 : ABC-ON

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetMutualInterferencePrevention", 1)
```

3.2.35. CaoController::Execute("SetToleranceComparatorOutputFormat") command

Set the sampling period.

Format SetToleranceComparatorOutputFormat (< ulMode >)

ulMode In judgment output form mode (VT_UI4)

0 : Normality

1 : Holding

2 : Off DeLay

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetToleranceComparatorOutputFormat", 1)
```

3.2.36. CaoController::Execute("SetStrobeTime") command

Set the time of the strobe.

Format SetStrobeTime (< ulCycle >)

ulCycle	Time of in strobe (VT_UI4)
	0 : 2ms
	1 : 5ms
	2 : 10ms
	3 : 20ms
Return value	: Out none

Example

```
// Example program
call caoCtrl.Execute("SetStrobeTime", 3)
```

3.2.37. CaoController::Execute("SetDataStorage") command

Set the data storage (number of accumulation and accumulation cycle).

Format SetDataStorage (<ulPoint>, < ulCycle >)

ulPoint	: In accumulation point (VT_UI4)
	The maximum number changes in the number of accumulation objects.
	1 - 1200000
ulCycle	: In accumulation cycle (VT_UI4)
	0 :Sampling period ×1

1 :Sampling period ×2
 2 :Sampling period ×5
 3 :Sampling period ×10
 4 :Sampling period ×20
 5 :Sampling period ×50
 6 :Sampling period ×100
 7 :Sampling period ×200
 8 :Sampling period ×500
 9 :Sampling period ×1000
 10 :Timing synchronization

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetDataStorage", Array(1000, 10))
```

3.2.38. CaoController::Execute("SetAnalogChannel") command

Set analog output Ch.

Format SetAnalogChannel (<ulChNo>, <ulOUTNo >)

ulChNo : In analogue output Ch number (VT_UI4)
 0 - (number -1 of use analogue outputs Ch)

ulOUTNo : OUT number in which in is output (VT_UI4)
 0 - 11

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetAnalogChannel", Array(0, 0))
```

3.2.39. GaoController::Execute("SetAlarmOutputForm") command

Set the alarm output form.

Format SetAlarmOutputForm (< ulMode >)

ulMode : In alarm output form (VT_UI4)
 0 :Only the system alarm
 1 :Only the measurements alarm
 2 :System alarm and measurements alarm

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetAlarmOutputForm", 2)
```

3.2.40. GaoController::Execute("SetNumOfUsedHeads") command

Set the number of use heads.

Format SetNumOfUsedHeads (< ulHeadNum >)

ulHeadNum : Number of in use heads (VT_UI4)
 2 - 12

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetNumOfUsedHeads", 2)
```

3.2.41. GaoController::Execute("SetNumOfUsedOut") command

Set the number of use OUT.

Format SetNumOfUsedOut (< ulOUTNum >)

ulOUTNum : Number of in use OUT (VT_UI4)
2 - 12

Return value : Out none

Example

```
// Example program
call caoCtrl.Execute("SetNumOfUsedOut", 2)
```

3.2.42. GaoController::Execute("GetPanel") command

Acquire the OUT number displayed in the upper row and the the lower of the display panel.

Format GetPanel ()

Argument : In none

Return value : Out setting number (VT_VARIANT | VT_ARRAY)

Index	Content
0	OUT number displayed in upper row (VT_I4)
1	OUT number displayed in the lower (VT_I4)

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetPanel")
```

3.2.43. GaoController::Execute("GetTolerance") command

Acquire the allowance.

Format GetTolerance (<ulOUTNo>)

ulOUTNo : In OUT number (VT_UI4)
0 - 11

Return value : Out setting number (VT_VARIANT | VT_ARRAY)

Index	Content
0	Allowance upper bound value (VT_I4)
1	Allowance lower bound value (VT_I4)
2	Value of allowance hysteresis (VT_I4)

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetTolerance", 0)
```

3.2.44. GaoController::Execute("GetAbleMode") command

Acquire the ABLE mode.

Format GetAbleMode (<ulOUTNo>)

ulOUTNo : In OUT number (VT_UI4)
0 - 11

Return value : Out ABLE mode (VT_I4)
0 :Automatic operation
1 :Manual

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetAbleMode", 1)
```

3.2.45. GaoController::Execute("GetAbleMinMax") command

Acquire the ABLE sphere of control.

Format GetAbleMinMax (<ulOUTNo>)

ulOUTNo : In OUT number (VT_UI4)
0 - 11

Return value : Out ABLE sphere of control (VT_VARIANT | VT_ARRAY)

Index	Content
0	Minimum value of ABLE (VT_I4)
1	ABLE maximum value (VT_I4)

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetAbleMinMax", 1)
```

3.2.46. CaoController::Execute("GetMeasureMode") command

Acquire the measurement mode.

Format GetMeasureMode (<ulHeadNo>)

ulHeadNo : In head number (VT_UI4)
0 - 11

Return value : Out measurement mode (VT_I4)

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetMeasureMode", 1)
```

3.2.47. CaoController::Execute("GetBasicPoint") command

Acquire the base point.

Format GetBasicPoint (<ulHeadNo>)

ulHeadNo : In head number (VT_UI4)

0 - 11

Return value : Out base point (VT_I4)

0 : NEAR

1 : FAR

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetBasicPoint", 1)
```

3.2.48. CaoController::Execute("GetNumAlarm") command

Acquire the alarm processing frequency.

Format GetNumAlarm (<ulHeadNo>)

ulHeadNo : In head number (VT_UI4)

0 - 11

Return value : Out alarm processing frequency (VT_I4)

0 - 9999

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetNumAlarm", 1)
```

3.2.49. CaoController::Execute("GetNumRecovery") command

Acquire the alarm return frequency.

Format GetNumRecovery (<ulHeadNo>)

ulHeadNo : In head number (VT_UI4)

0 - 11

Return value : Out alarm return frequency (VT_I4)
0 - 9999

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetNumRecovery", 1)
```

3.2.50. CaoController::Execute("GetAlarmLevel") command

Acquire the alarm level.

Format GetAlarmLevel (<ulHeadNo>)

ulHeadNo : In head number (VT_UI4)
0 - 11

Return value : Out alarm level (VT_I4)
0 - 9

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetAlarmLevel", 1)
```

3.2.51. CaoController::Execute("GetReflectionMode") command

Acquire the installation mode.

Format GetReflectionMode (<ulHeadNo>)

ulHeadNo : In head number (VT_UI4)
0 - 11

Return value : Out installation mode (VT_I4)
0 :Diffuse reflection

1 :Regular reflection

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetReflectionMode", 1)
```

3.2.52. CaoController::Execute("GetMask") command

Acquire the mask.

Format GetMask (<ulHeadNo>)

ulHeadNo : In head number (VT_UI4)

0 - 11

Return value : Out mask setting (VT_VARIANT | VT_ARRAY)

Index	Content
0	ON/OFF (VT_I4) of mask 0 : OFF 1 : ON
1	Mask boundary value 1 (VT_I4) -999999 - 999999
2	Mask boundary value 2 (VT_I4) -999999 - 999999

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetMask", 1)
```

3.2.53. CaoController::Execute("GetMedian") command

Acquire the median.

Format GetMedian (<ulHeadNo>)

ulHeadNo : In head number (VT_UI4)
0 - 11

Return value : Out median (VT_I4)
0 : OFF
1 : Seven points
2 : 15 points
3 : 31 points

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetMedian", 1)
```

3.2.54. CaoController::Execute("GetLaserCtrlGroup") command

Acquire the LASER CTRL group.

Format GetLaserCtrlGroup (<ulHeadNo>)

ulHeadNo : In head number (VT_UI4)
0 - 11

Return value : Out LASER CTRL group (VT_I4)
0 : LASER CTRL 1
1 : LASER CTRL 2

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetLaserCtrlGroup", 1)
```

3.2.55. GaoController::Execute("GetRange") command

Acquire the median.

Format GetRange (<ulHeadNo>)

ulHeadNo : In head number (VT_UI4)
0 - 11

Return value : Out cooking stove (VT_I4)
0 : CENTER
1 : FAR

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetRange", 1)
```

3.2.56. GaoController::Execute("GetMutualInterferencePreventionGroup") command

Acquire the group of the mutual interference prevention.

Format GetMutualInterferencePreventionGroup (<ulHeadNo>)

ulHeadNo : In head number (VT_UI4)
0 - 11

Return value : Group of out mutual interference prevention (VT_I4)
0 : Group A
1 : Group B
2 : Group C

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetMutualInterferencePreventionGroup", 1)
```

3.2.57. CaoController::Execute("GetCalcTarget") command

Acquire the measuring plane.

Format GetCalcTarget (<ulHeadNo>)

ulHeadNo : In head number (VT_UI4)
0 - 11

Return value : Out computing type (VT_I4)
0 :Peak 1
1 :Peak 2
2 :Peak 3
3 :Peak 4
4 :Peak ..one..-peak 2
5 :Peak ..one..-peak 3
6 :Peak ..one..-peak 4
7 :Peak ..two..-peak 3
8 :Peak ..two..-peak 4
9 :Peak ..three..-peak 4

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetCalcTarget", 1)
```

3.2.58. CaoController::Execute("GetOutAddSub") command

Acquire the target number of OUT between OUT for the operation addition and subtraction.

Format GetOutAddSub (<ulOUTNo>)

ulOUTNo : In OUT number (VT_UI4)
0 - 11

Return value : Out operation object (addition and subtraction) (VT_VARIANT | VT_ARRAY)

Index	Content
-------	---------

0	Side OUT number added (It is subtracted) (VT_I4).
1	Addition (subtraction) side OUT number (VT_I4)

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetOutAddSub", 1)
```

3.2.59. CaoController::Execute("GetOutOperation") command

Acquire the target number of OUT between OUT for operation multi paragraph calculation (AVE/P-P/MAX/MIN).

Format GetOutOperation (<ulOUTNo>)

ulOUTNo : In OUT number (VT_UI4)
0 - 11

Return value : Out operation object (addition and subtraction) (VT_I4)
The targeted OUT number is stored.
Two or more specification is LKIF_OUTNO member's OR.

- 0x0001 : OUT1
- 0x0002 : OUT2
- 0x0004 : OUT3
- 0x0008 : OUT4
- 0x0010 : OUT5
- 0x0020 : OUT6
- 0x0040 : OUT7
- 0x0080 : OUT8
- 0x0100 : OUT9
- 0x0200 : OUT10
- 0x0400 : OUT11
- 0x0800 : OUT12
- 0x0FFF: All OUT

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetOutOperation", 1)
```

3.2.60. CaoController::Execute("GetScaling") command

Acquire scaling.

Format GetScaling (<ulOUTNo>)

ulOUTNo : In OUT number (VT_UI4)

0 - 11

Return value : Out scaling (VT_VARIANT | VT_ARRAY)

Index	Content
0	Input value 1 (VT_I4)
1	Indicated value 1 (VT_I4)
2	Input value 2 (VT_I4)
3	Indicated value 2 (VT_I4)

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetScaling", 1)
```

3.2.61. CaoController::Execute("GetFilter") command

Acquire the filter mode and the filter parameter.

Format GetFilter (<ulOUTNo>)

ulOUTNo : In OUT number (VT_UI4)

0 - 11

Return value : Out filter mode and parameter (VT_VARIANT | VT_ARRAY) of

filter

Index	Content
0	Filter mode (VT_I4) 0 :Moving average 1 :Low-pass filter 2 :High-pass filter
1	Filter parameter (VT_I4) (If it is an average frequency, and LPF·HPF in case of the moving average, it is cutoff frequency.) For the moving average 0 : Once 1 : 4 2 : 16 3 : 64 4 : 256 5 : 1024 6 : 4096 7 : 16384 8 : 65536 9 : 262144 For LPF·HPF 0 : 3000 Hz 1 : 1000 2 : 300 3 : 100 4 : 30 5 : 10 6 : 3 7 : 1 8 : 0.3 9 : 0.1

Example

```
// Example program
```

```
Dim vntRet
vntRet = caoCtrl.Execute("GetFilter", 1)
```

3.2.62. CaoController::Execute("GetTriggerMode") command

Acquire the trigger mode.

Format GetTriggerMode (<ulOUTNo>)

ulOUTNo	:	In OUT number (VT_UI4)
		0 - 11
Return value	:	Out trigger mode (VT_I4)
		0 :External trigger 1
		1 :External trigger 2

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetTriggerMode", 1)
```

3.2.63. CaoController::Execute("GetOffset") command

Acquire the offset.

Format GetOffset (<ulOUTNo>)

ulOUTNo	:	In OUT number (VT_UI4)
		0 - 11
Return value	:	Out offset (VT_I4)
		-999999 - 999999

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetOffset", 1)
```

3.2.64. CaoController::Execute("GetCalcMode") command

Acquire the measurement mode.

Format GetCalcMode (<ulOUTNo>)

ulOUTNo : In OUT number (VT_UI4)
0 - 11

Return value : Out measurement mode (VT_I4)
0 :Standard
1 :Peak holding
2 :Bottom holding
3 :Peak to peak holding
4 :Sample holding
5 :Average holding

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetCalcMode", 1)
```

3.2.65. CaoController::Execute("GetAnalogScaling") command

Acquire the analog output scaling.

Format GetAnalogScaling (<ulOUTNo>)

ulOUTNo : In OUT number (VT_UI4)
0 - 11

Return value : Out analogue output scaling (VT_VARIANT | VT_ARRAY)

Index	Content
0	Input value 1 (VT_I4) -999999 - 999999

1	1 in output voltage (VT_I4) -10500(mV) - 10500(mV)
2	Input value 2 (VT_I4) -999999 - 999999
3	2 in output voltage (VT_I4) -10500(mV) - 10500(mV)

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetAnalogScaling", 1)
```

3.2.66. CaoController::Execute("GetDisplayUnit") command

Acquire the trigger mode.

Format GetDisplayUnit (<ulOUTNo>)

ulOUTNo : In OUT number (VT_UI4)
 : 0 - 11

Return value : Unit of out minimum display (VT_I4)
 : 0 : 0.01mm
 : 1 : 0.001mm
 : 2 : 0.0001mm
 : 3 : 0.00001mm
 : 4 : 0.1μm
 : 5 : 0.01μm
 : 6 : 0.001μm

 : 0 : 0.1m/s
 : 1 : 0.01m/s
 : 2 : 0.001m/s
 : 3 : 0.1mm/s
 : 4 : 0.01mm/s

5 : 0.001mm/s
 6 : 0.0001mm/s

0 : 0.1km/s²
 1 : 0.01km/s²
 2 : 0.001km/s²
 3 : 0.1m/s²
 4 : 0.01m/s²
 5 : 0.001m/s²
 6 : 0.0001m/s²

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetDisplayUnit", 1)
```

3.2.67. CaoController::Execute("GetMeasureType") command

Acquire the measurement type.

Format GetMeasureType (<ulOUTNo>)

ulOUTNo	:	In OUT number (VT_UI4)
		0 - 11
Return value	:	Out measurement type (VT_I4)
		0 : Displacement
		1 : Speed
		2 : Acceleration

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetMeasureType", 1)
```

3.2.68. GaoController::Execute("GetSynchronization") command

Acquire a synchronous setting.

Format GetSynchronization (<ulOUTNo>)

ulOUTNo : In OUT number (VT_UI4)
0 - 11

Return value : Whether make to the out accumulation object (VT_I4) or not-
0 :Off the subject
1 :Object

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetSynchronization", 1)
```

3.2.69. GaoController::Execute("GetStorageTarget") command

Acquire the accumulation object of storage.

Format GetStorageTarget (<ulOUTNo>)

ulOUTNo : In OUT number (VT_UI4)
0 - 11

Return value : Whether make to the out accumulation object (VT_I4) or not-
0 :Off the subject
1 :Object

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetStorageTarget", 1)
```

3.2.70. GaoController::Execute("GetSamplingCycle") command

The sampling period is stored.

Format GetSamplingCycle ()

Argument	:	In none
Return value	:	Group of out mutual interference prevention (VT_I4)
		0 : 2.55us
		1 : 5us
		2 : 10us
		3 : 20us
		4 : 50us
		5 : 100us
		6 : 200us
		7 : 500us
		8 : 1ms

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetSamplingCycle")
```

3.2.71. GaoController::Execute("GetMutualInterferencePrevention") command

Acquire the mutual interference prevention.

Format GetMutualInterferencePrevention ()

Argument	:	In none
Return value	:	Out mutual interference prevention (VT_I4)
		0 : OFF
		1 : AB-ON
		2 : ABC-ON

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetMutualInterferencePrevention")
```

3.2.72. GaoController::Execute("GetToleranceComparatorOutputFormat") command

Acquire the mutual interference prevention.

Format GetToleranceComparatorOutputFormat ()

Argument : In none

Return value : Out judgment output form mode (VT_I4)

0 :Normality

1 :Holding

2 :Off DeLay

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetToleranceComparatorOutputFormat")
```

3.2.73. GaoController::Execute("GetDataStorage") command

Acquire the data storage (number of accumulation and accumulation cycle).

Format GetDataStorage ()

Argument : In none

Return value : Out data storage (VT_VARIANT | VT_ARRAY)

Index	Content
0	Accumulation point (VT_I4) 1 - 1200000
1	Accumulation cycle (VT_I4) 0 :Sampling period ×1

1	:Sampling period ×2
2	:Sampling period ×5
3	:Sampling period ×10
4	:Sampling period ×20
5	:Sampling period ×50
6	:Sampling period ×100
7	:Sampling period ×200
8	:Sampling period ×500
9	:Sampling period ×1000

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetDataStorage")
```

3.2.74. CaoController::Execute("GetAnalogChannel") command

Acquire analog output Ch.

Format GetAnalogChannel (<ulChNo>)

ulChNo : In analogue output Ch number (VT_UI4)
Return value : OUT number in which out is output (VT_I4)

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetAnalogChannel", 1)
```

3.2.75. CaoController::Execute("GetAlarmOutputForm") command

Acquire the alarm output form.

Format GetAlarmOutputForm ()

Argument	:	In none
Return value	:	Out alarm output form (VT_I4)
		0 :Only the system alarm
		1 :Only the measurements alarm
		2 :System alarm and measurements alarm

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetAlarmOutputForm")
```

3.2.76. CaoController::Execute("GetNumOfUsedHeads") command

Acquire the number of use heads.

Format GetNumOfUsedHeads ()

Argument	:	In none
Return value	:	Number of out use heads (VT_I4)
		2 - 12

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetNumOfUsedHeads")
```

3.2.77. CaoController::Execute("GetNumOfUsedOut") command

Acquire the number of use OUT.

Format GetNumOfUsedOut ()

Argument	:	In none
Return value	:	Number of out use OUT (VT_I4)

2 - 12

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetNumOfUsedOut")
```

3.2.78. CaoController::Execute("GetNumOfUsedAnalogCh") command

Acquire the number of use analogue outputs Ch.

Format GetNumOfUsedAnalogCh ()

Argument : In none
Return value : Number of out use analogue outputs Ch (VT_I4)

2 - 12

Example

```
// Example program
Dim vntRet
vntRet = caoCtrl.Execute("GetNumOfUsedAnalogCh")
```

3.3. Mode change command**3.3.1. CaoController::Execute("StartMeasure") command**

It changes to the measurement mode.

Format StartMeasure ()

Argument : None
Return value : None

Example

```
// Example program
call caoCtrl.Execute("StartMeasure")
```

3.3.2. GaoController::Execute("StopMeasure") command

It changes to a set (communication) mode.

Format StopMeasure ()

Argument : None

Return value : None

Example

```
// Example program  
call caoCtrl.Execute("StopMeasure")
```
