KEYENCE
ILDLEP1 provider

Version 1.0.0

User's guide

October 12, 2018

Remarks:

This document uses the machine translation.

【 revision history 】

Version	Date	Content
1.0.0	2018-10-12	First edition.

【Operation confirmation model 】

Model	Version	Notes
IL series		Communication unit (DL-EP1) + amplifier unit (IL-1000 etc.) Composition of + sensor unit (IL-S100 etc.)

Contents

1. Introduction

The KEYENCE ILDLEP1 provider (henceforth ILDLEP1 provider) is ORiN2 CAO provider that accesses the laser made of KEYENCE type displacement sensor (IL series).

[in] this document, it explains the outline of the ILDLEP1 provider and the mounted CAO interface (function specification).

2. Outline of provider

2.1. Installation

The ILDLEP1 provider module is composed of following DLL. The installation work is unnecessary when installing it with the installer of ORiN2 SDK. Please refer to the table 2-1, when you install it by hand power.

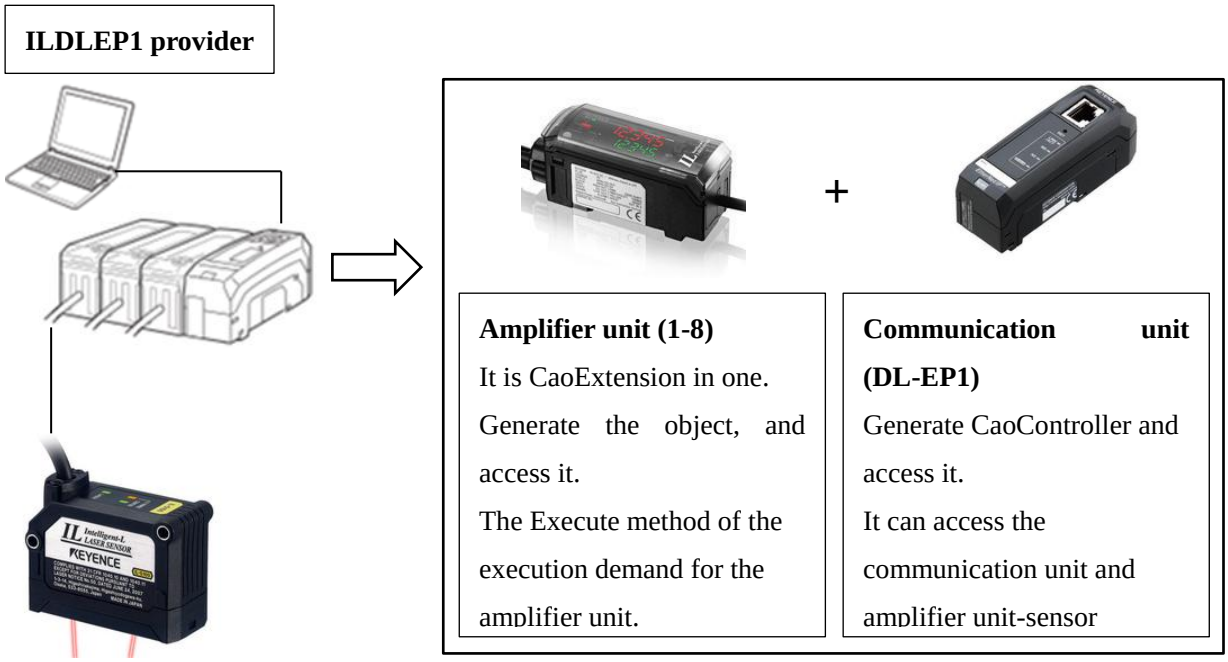
Table2-1ILDLEP1 provider

File name	CaoProvKEYENCEILDLEP1.dll
ProgID	CaoProv.KEYENCE.ILDLEP1
Registry registration	regsvr32 CaoProvKEYENCEILDLEP1.dll.dll
Blotting out of registry registration	regsvr32 /u CaoProvKEYENCEILDLEP1.dll.dll

2.2. Outline

The ILDLEP1 provider accesses DL-EP1 by using the CIP protocol.

Generate the CaoExtension object of each amplifier unit connected with DL-EP1, and access doing sensor information reading and writing to the Execute method or CaoVariable.



3. Method property

3.1. CaoWorkspace::AddController method

The ILDLEP1 provider connects the communication referring to connected parameter for the communication at AddController.

Format AddController(<bstrCtrlName:BSTR>,<bstrProvName:BSTR>,<bstrPCName:BSTR>, <bstrOption:BSTR>)

bstrCtrlName : [in] controller name

bstrProvName : [in] provider name. Fixed value = "CaoProv.KEYENCE.ILDLEP1"

bstrPCName : Execution machine name of in provider

bstrOption : [in] optional character string

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

Table3-1Optional character string of CaoWorkspace::AddController

Option ([1])	Explanation
Conn=< connected parameter >	Indispensability. Communication form and connected parameter. (Refer to chapter 3.1.1)
MyIP[=< Internet Protocol address >]	Internet Protocol address. (two or more NIC usage) (default: There is no specification.)
ConnTimeout[=<Connected time-out time>]	Timeout period when connecting it. (millisecond) (default: 3000)
TimeOut[=< timeout period >]	Timeout period when sending and receiving it. (millisecond) (default: 3000)

3.1.1. Conn is optional.

Specify a communication form and connected parameter.

Connected parameter character string of optional Conn is shown as follows.

Ethernet/IP device

"Conn=ETH:<Dest IP Address>[:<Dest Port No>[:<Src IP Address>[:<Src Port No>]]]"

"Conn=TCP:<Dest IP Address>[:<Dest Port No>[:<Src IP Address>[:<Src Port No>]]]"

[1] In the square bracket, a possible omission is shown. Moreover, the underlined part under the explanation of each parameter becomes a default value when the option is not specified.

<Dest IP Address>	:	Internet Protocol address TCP/IP connection ahead. Example:"127.0.0.1","192.168.0.1"
<Dest Port No>	:	Port number TCP/IP connection ahead Example: 44818,5006,5007
<Src IP Address>	:	Internet Protocol address. (two or more NIC usage) ([2])
<Src Port No>	:	Port number. (two or more NIC usage)

3.2. CaoController::Execute method

The Execute method of the CaoController class is a method for the execution of the command. Specify the command name in the first argument and specify the parameter of the command for the second argument. Refer to the chapter 4.1 for details.

Format [<vntRet:VARIANT> =]Execute(<bstrCmd:BSTR>[,<vntParam:VARIANT>])

<bstrCmd>	:	[in] command name
<vntParam>	:	[in] parameter

3.3. CaoController::AddVariable method

The AddVariable method of the CaoController class is a method for making the variable object. Refer to the chapter5.1 for the variable list that can be used.

Format AddVariable(<bstrVariableName:BSTR>[,<bstrOption:BSTR>])

<bstrVariableName>	:	[in] variable identifier
<bstrOption>	:	[in] optional character string

3.4. CaoController::AddExtension method

The AddExtension method of the CaoController class is a method for making the Extension object of each amplifier unit connected with DL-EP1.

Format AddExtension(<bstrChannelNo:BSTR>)

<bstrChannelNo>	:	[in] Identification number of in amplifier unit. Specify it by the format of "CH< identification number >". (identification number: 1-8)

[2] If Internet Protocol address is specified by both of optional Conn option and MyIP, it becomes an error. Please specify either by one when using it.

3.5. CaoExtension::Execute method

The Execute method of the CaoExtension class is a method for the execution of the command. It specifies parameters in the first argument and it specifies parameters for the command name and the second argument. Please refer to the chapter 4.2 for details.

Format [<vntRet:VARIANT> =]Execute(<bstrCmd:BSTR>[,<vntParam:VARIANT>])

<bstrCmd>	:	[in] command name
<vntParam>	:	[in] parameter

3.6. CaoExtension::AddVariable method

The AddVariable method of the CaoExtension class is a method for making the variable object. Please refer to the chapter 5.2 for the variable list that can be used.

Format AddVariable(<bstrVariableName:BSTR>[,<bstrOption:BSTR>])

<bstrVariableName>	:	[in] variable identifier
<bstrOption>	:	[in] optional character string

3.7. CaoController::get_VariableNames property

Acquire the list of the variable identifier of the CaoController class.

3.8. CaoController::get_ExtensionNames property

Acquire the Extension person list of the CaoController class.

3.9. CaoExtension::get_VariableNames property

Acquire the list of the variable identifier of the CaoExtension class.

3.10. CaoVariable::put_ID property

Set ID of CaoVariable. A set value sets the value specified by the argument.

Use it for the ID specification when it accesses a part of variable.

3.11. CaoVariable::get_ID property

Acquire ID of CaoVariable.

3.12. CaoVariable::put_Value property

Set the value of a variable of CaoVariable specified by the variable identifier. A set value sets the value specified by the argument.

3.13. CaoVariable::get_Value property

Acquire the value of a variable of CaoVariable specified by the variable identifier.

4. Command list

4.1. CaoController class

Table4-1CaoController class command list

Command name	Function	
Raw	Send a general-purpose command.	P11

4.1.1. CaoController::Execute("Raw") command

Send a general-purpose command.

Format Raw(<vntDataArray:VARIANT>)

<vntDataArray> : [in] command data
(VT_ARRAY | VT_UI1)
[0] Service code
[1] Class ID
[2] Instance ID
[3] Attribute ID(Low)
[4] Attribute ID(High)
[n] Service data

- Specify the High side with 0x00 when attribute ID is 8Bit.
- Treat since element 5 as service data.
- ..Low side.. ..8Bit.. specify the service data.
- The NULL(0x00) padding is done internally when the service data is odd number Byte.

Return value : [out] response data
(VT_ARRAY | VT_UI1)

- The response data restores and VT_EMPTY is restored to a no potato.
- The response data ..Low side.. ..8Bit.. is stored.

example

```
Byte param = new Byte[5];
```

```
param[0] = 0x0E;
param[1] = 0x67;
param[2] = 0x00;
param[3] = 0x64;
param[4] = 0x00;
Object result = m_CaoController.Execute("Raw", param);
```

4.2. CaoExtension class

Table4-2CaoExtension class command list

Command name	Function	
ZeroShift	Do 0 shift execution demand.	P13
ZeroShiftReset	Do 0 shift reset demand.	P13
Reset	Do the reset demand.	P14
InitReset	Do the initial reset demand.	P14
SystemParametersSet	Do the system parameter set demand.	P14
ToleranceTuning	Do the allowance tuning demand.	P15
DecideHIGH1stPTOf2PTTuning	Do the first point on tuning HIGH side in two points decision operation demand.	P15
DecideHIGH2ndPTOf2PTTuning	Demand the second point on tuning HIGH side in two points decision operation (HIGH side setting value decision).	P15
DecideLOW1stPTOf2PTTuning	Do the first point on tuning LOW side in two points decision operation demand.	P16
DecideLOW2ndPTOf2PTTuning	Demand the second point on tuning LOW side in two points decision operation (LOW side setting value decision).	P16
DecideCALSET1	Do calibration SET1 decision operation demand.	P17
DecideCALSET2	Demand calibration SET2 decision operation (calibration execution).	P17
DecideCALSET1OfCALCV2PT	Do calibration SET1 decision of operation value two points operation demand.	P17
DecideCALSET2OfCALCV2PT	Demand calibration SET2 decision of operation value two points operation (calibration execution of the operation value two points).	P18

DecideCALSET1OfCALCV3PT	Do calibration SET1 decision of operation value three points operation demand.	P18
DecideCALSET2OfCALCV3PT	Do calibration SET2 decision of operation value three points operation demand.	P18
DecideCALSET3OfCALCV3PT	Demand calibration SET3 decision of operation value three points operation (calibration execution of the operation value three points).	P19
Req1PTTuningForStepCountFilter	Do one point tuning demand for the difference count filter.	P19
Decide1stPTOf2PTTuningForStepCountFilter	Do the first tuning point in two points decision operation demand for the difference count filter.	P19
Decide2ndPTOf2PTTuningForStepCountFilter	Demand the second tuning point in two points decision operation for the difference count filter (HIGH side/LOW side setting value decision).	P20

4.2.1. CaoExtension::Execute("ZeroShift") command

Do 0 shift execution demand.

Format ZeroShift ()

Argument : It is not.
Return value : It is not.

Usage

example

```
m_CaoExtension.Execute("ZeroShift");
```

4.2.2. CaoExtension::Execute("ZeroShiftReset") command

Do 0 shift reset demand.

Format ZeroShiftReset ()

Argument : It is not.
Return value : It is not.

Usage

example

```
m_CaoExtension.Execute("ZeroShiftReset");
```

4.2.3. CaoExtension::Execute("Reset") command

Do the reset demand.

Format Reset ()

Argument : It is not.
Return value : It is not.

Usage

example

```
m_CaoExtension.Execute("Reset");
```

4.2.4. CaoExtension::Execute("InitReset") command

Do the initial reset demand.

Format InitReset ()

Argument : It is not.
Return value : It is not.

Usage

example

```
m_CaoExtension.Execute("InitReset");
```

4.2.5. CaoExtension::Execute("SystemParametersSet") command

Do the system parameter set demand.

Format SystemParametersSet ()

Argument : It is not.
Return value : It is not.

Usage

example

```
m_CaoExtension.Execute("SystemParametersSet");
```

4.2.6. CaoExtension::Execute("ToleranceTuning") command

Do the allowance tuning demand.

Format ToleranceTuning ()

Argument : It is not.

Return value : It is not.

Usage

example

```
m_CaoExtension.Execute("ToleranceTuning");
```

4.2.7. CaoExtension::Execute("DecideHIGH1stPTOf2PTTuning") command

Do the first point on tuning HIGH side in two points decision operation demand.

Format DecideHIGH1stPTOf2PTTuning ()

Argument : It is not.

Return value : It is not.

Usage

example

```
m_CaoExtension.Execute("DecideHIGH1stPTOf2PTTuning");
```

4.2.8. CaoExtension::Execute("DecideHIGH2ndPTOf2PTTuning") command

Demand the second point on tuning HIGH side in two points decision operation (HIGH side setting value decision).

Format DecideHIGH2ndPTOf2PTTuning ()

Argument : It is not.
Return value : It is not.

Usage
example

```
m_CaoExtension.Execute("DecideHIGH2ndPTOf2PTTuning");
```

4.2.9. CaoExtension::Execute("DecideLOW1stPTOf2PTTuning") command

Do the first point on tuning LOW side in two points decision operation demand.

Format DecideLOW1stPTOf2PTTuning ()

Argument : It is not.
Return value : It is not.

Usage
example

```
m_CaoExtension.Execute("DecideLOW1stPTOf2PTTuning");
```

4.2.10. CaoExtension::Execute("DecideLOW2ndPTOf2PTTuning") command

Demand the second point on tuning LOW side in two points decision operation (LOW side setting value decision).

Format DecideLOW2ndPTOf2PTTuning ()

Argument : It is not.
Return value : It is not.

Usage
example

```
m_CaoExtension.Execute("DecideLOW2ndPTOf2PTTuning");
```

4.2.11. CaoExtension::Execute("DecideCALSET1") command

Do calibration SET1 decision operation demand.

Format DecideCALSET1 ()

Argument : It is not.

Return value : It is not.

Usage

example

```
m_CaoExtension.Execute("DecideCALSET1");
```

4.2.12. CaoExtension::Execute("DecideCALSET2") command

Demand calibration SET2 decision of operation value two points operation (calibration execution of the operation value two points).

Format DecideCALSET2 ()

Argument : It is not.

Return value : It is not.

Usage

example

```
m_CaoExtension.Execute("DecideCALSET2");
```

4.2.13. CaoExtension::Execute("DecideCALSET1OfCALCV2PT") command

Do calibration SET1 decision of operation value two points operation demand.

Format DecideCALSET1OfCALCV2PT ()

Argument : It is not.

Return value : It is not.

Usage

example

```
m_CaoExtension.Execute("DecideCALSET1OfCALCV2PT");
```

4.2.14. CaoExtension::Execute("DecideCALSET2OfCALCV2PT") command

Demand calibration SET2 decision of operation value two points operation (calibration execution of the operation value two points).

Format DecideCALSET2OfCALCV2PT ()

Argument : It is not.
Return value : It is not.

Usage

example

```
m_CaoExtension.Execute("DecideCALSET2OfCALCV2PT");
```

4.2.15. CaoExtension::Execute("DecideCALSET1OfCALCV3PT") command

Do calibration SET1 decision of operation value three points operation demand.

Format DecideCALSET1OfCALCV3PT ()

Argument : It is not.
Return value : It is not.

Usage

example

```
m_CaoExtension.Execute("DecideCALSET1OfCALCV3PT");
```

4.2.16. CaoExtension::Execute("DecideCALSET2OfCALCV3PT") command

Do calibration SET2 decision of operation value three points operation demand.

Format DecideCALSET2OfCALCV3PT ()

Argument : It is not.
Return value : It is not.

Usage

example

```
m_CaoExtension.Execute("DecideCALSET2OfCALCV3PT");
```

4.2.17. CaoExtension::Execute("DecideCALSET3OfCALCV3PT") command

Demand calibration SET3 decision of operation value three points operation (calibration execution of the operation value three points).

Format DecideCALSET3OfCALCV3PT ()

Argument : It is not.

Return value : It is not.

Usage

example

```
m_CaoExtension.Execute("DecideCALSET3OfCALCV3PT");
```

4.2.18. CaoExtension::Execute("Req1PTTuningForStepCountFilter") command

Do one point tuning demand for the difference count filter.

Format Req1PTTuningForStepCountFilter ()

Argument : It is not.

Return value : It is not.

Usage

example

```
m_CaoExtension.Execute("Req1PTTuningForStepCountFilter");
```

4.2.19. CaoExtension::Execute("Decide1stPTOf2PTTuningForStepCountFilter") command

Do the first tuning point in two points decision operation demand for the difference count filter.

Format Decide1stPTOf2PTTuningForStepCountFilter ()

Argument : It is not.
Return value : It is not.

Usage
example

```
m_CaoExtension.Execute("Decide1stPTOf2PTTuningForStepCountFilter");
```

4.2.20. CaoExtension::Execute("Decide2ndPTOf2PTTuningForStepCountFilter") command

Demand the second tuning point in two points decision operation for the difference count filter (HIGH side/LOW side setting value decision).

Format Decide2ndPTOf2PTTuningForStepCountFilter ()

Argument : It is not.
Return value : It is not.

Usage
example

```
m_CaoExtension.Execute("Decide2ndPTOf2PTTuningForStepCountFilter");
```

5. Variable list

5.1. CaoController class

5.1.1. System variable

Table5-1CaoController class system variable list

Variable identifier	Data type	Explanation	Attribute	
			get	put
@MAKER_NAME	VT_BSTR	Return "KEYENCE maker name =".	✓	-
@VERSION	VT_BSTR	Return provider version information.	✓	-
@LAST_RESPONSE_STATUS	VT_ARRAY | VT_I4	Return error reply detailed information immediately before. (VT_ARRAY | VT_I4) [0] General status [1] Additional status (0 sets in case of not being)	✓	-

5.2. CaoExtension class

5.2.1. System variable

Table5-2CaoExtension class system variable list

Variable identifier	Data type	Explanation	Attribute	
			get	put
@ErrStatus	VT_I4	State of sensor amplifier error When the error occurs, the corresponding bit becomes ON(1). (initial value :---) bit0 : Overcurrent error (ErC) bit1 : EEPROM..error. (ErE) bit2 : Head error (ErH) bit7 : Floodlight projector laser error (ErH LASEr) bit8 : Model disagreement error (ErH tyPE) bit11 : Communication error	✓	-

		between amplifiers (Er.Com) bit12 : Error of the number of unit (Er.Unt) bit13 : Operation error (Er.CAL) Excluding the above-mentioned : 0 fixation		
@ControlOutput	VT_I4	Output status of sensor When the output is turning on, the corresponding bit becomes ON(1). (initial value :---) bit0 : HIGH judgment output bit1 : LOW judgment output bit2 : GO judgment output bit3 : Alarm output	✓	-
@CriteriaValue	VT_R8	[3]Judgment value (P.V . value) () -99.999 -+99.999 (initial value :---)	✓	-
@RawValue	VT_R8	Internal judgment value (R.V . value) ([3]) -99.999 -+99.999 (initial value :---)	✓	-
@PeakValue	VT_R8	Peak holding value (of holding[3]) ・ Things except the sample holding Peak holding value for sampling period -99.999 -+99.999 (initial value :---) ・ When hold sample -99.998	✓	-
@BottomValue	VT_R8	Bottom holding value (of holding[3]) ・ Things except the sample holding Peak holding value for sampling period -99.999 -+99.999 (initial value :---) ・ When hold sample -99.998	✓	-
@CalcValue	VT_R8	Operation value (CALC value) ([3]) -99.999 ～ +99.999	✓	-

[3] IL-300/IL-600 使用時, -999.99 ~ +999.99, IL-2000 使用時, -9999.9 ~ +9999.9 になります.

@AnalogValue	VT_R8	Analog of present output value (Only the main phone :.) (initial value :---) Voltage : -5.000 ～ +5.000 At ..(.. error : +5.500) 4 ? 20mA : +4.00 ～ +20.00 At ..(.. error : +3.00) OFF : 0 fixation	✓	-
@SettingError	VT_I4	Setting and abnormality states 0 - 1 (initial value :---) 0 : The setting is normal. 1 : The setting is abnormal.	✓	-
@ExternalInputState	VT_I4	State of external input When an external input line of the sensor amplifier or an external input of a cyclic communication is turning on, the corresponding bit becomes ON(1). (initial value :---) bit0 : External input 1 bit1 : External input 2 bit2 : External input 3 bit3 : External input 4	✓	-
@EEPROMWriteResult	VT_I4	Result 0 - 2 (initial value :---) 0 : It is writing it. 1 : Normal termination 2 : Abnormal termination	✓	-
@ZeroShiftResult	VT_I4	Execution result of 0 shifts 0-shift reset 0 - 2 (initial value :---) 0 : It is executing it. 1 : Normal termination 2 : Abnormal termination (SHiFt Err)	✓	-
@ResetResult	VT_I4	Execution result of reset 0 - 2 (initial value :---) 0 : It is executing it.	✓	-

		1 : Normal termination 2 : Abnormal termination		
@TuningResult	VT_I4	Execution result of tuning 0 - 2 (initial value :---) 0 : It is executing it. 1 : Normal termination 2 : Abnormal termination	✓	-
@CalibrationResult	VT_I4	Execution result of calibration 0 - 2 (initial value :---) 0 : It is executing it. 1 : Normal termination 2 : Abnormal termination	✓	-
@Bank_HIGH	VT_R8	HIGH side setting value (BANK0 - 3) (^{3}) -..99.999 -+.. 99.999 (value :+ of initial5.000) - Bank No specifies in the ID property (ID=0 - 3).	✓	✓
@Bank_LOW	VT_R8	LOW side setting value (BANK0 - 3) (^{3}) -..99.999 -+.. 99.999 (value :- of initial5.000) - Bank No specifies in the ID property (ID=0 - 3).	✓	✓
@Bank_ShiftTarget	VT_R8	Shift target value (BANK0 - 3) (^{3}) -99.999 -+99.999 (Initial value: 0) - Bank No specifies in the ID property (ID=0 - 3).	✓	✓
@Bank_AnalogOutputUpperLimit	VT_R8	Analog output upper bound value (BANK0 - 3) (^{3}) (Only the main phone :.) -..99.999 -+.. 99.999 (value :+ of initial10.000) - Bank No specifies in the ID property (ID=0 - 3).	✓	✓
@Bank_AnalogOutputLowerLimit	VT_R8	Analog output lower bound value (BANK0 - 3) (^{3}) (Only the main phone :.) -..99.999 -+.. 99.999 (value :- of	✓	✓

		initial10.000) - Bank No specifies in the ID property (ID=0 - 3).		
@KeyLock	VT_I4	Key lock function 0 - 1 (Initial value: 0) 0 : Unlock 1 : Key lock	✓	✓
@Bank	VT_I4	Bank function At Get: The operating bank number. 0 - 3 (Initial value: 0) At Put: Change the bank number. 0 : Bank 0 1 : Bank 1 2 : Bank 2 3 : Bank 3	✓	✓
@Timing	VT_I4	At Get: The operating state of timing. 0 - 1 (Initial value: 0) 0 : It is sampling it. 1 : It is non-sampling it. At Put: Set the timing input signal. 0 : Timing input signal OFF 1 : Timing input signal ON	✓	✓
@ProjectionStop	VT_I4	At Get: The operating [**hika] halt condition. 0 - 1 (Initial value: 0) 0 : [**hika] inside 1 : [**hika] is stopping ([**hika] stop input ON/laser error/head error). At Put: Set the laser [**hika] stop input. 0 : [**hika] stop input OFF 1 : [**hika] stop input ON	✓	✓
@SubDisplay	VT_I4	Display in subdisplay part 0 - 5 (Initial value: 0) 0 : R.V . Value 1 : Analog value	✓	✓

		2 : HI setting value 3 : LO setting value 4 : 0 shift value 5 : CALC value		
@SystemParameters	VT_I4	At Get: The present state of the system parameter. The bit that corresponds according to the state becomes ON(1) now. (initial value :---) At Put After a set writing, the reflection of the setting :. "SystemParametersSet" Command It is necessary to execute it drinking. bit0 : 0 : NPN 1 : PNP bit1,2,3 : (000 fixation only of the main phone of the cordless handset.) 000 : Analog output OFF 001 : 0 ～ +5V 010 : -5 ～ +5V 011 : +1 ～ +5V 100 : 4 ～ 20mA	✓	✓
@ToleranceSettingWidth	VT_R8	Set width (of allowance of allowance tuning[3]) 0.000 - 99.999 (Initial value: 0.200)	✓	✓
@CAL	VT_I4	Calibration function 0 - 1 (Initial value: 0) 0 : Initial state 1 : User setting	✓	✓
@CALSET1	VT_R8	Targeted value (of calibration of the first point[3]) -99.999 -+99.999 (Initial value: 0.000)	✓	✓
@CALSET2	VT_R8	Targeted value (of calibration of the second	-	-

		point[3]) -..99.999 -+.. 99.999 (value :+ of initial5.000)		
@CALCVCAL	VT_I4	Operation value calibration function (Only the main phone :.) 0 - 2 (Initial value: 0) 0 : Initial state 1 : Operation two point calibration 2 : Operation three point calibration	✓	✓
@CACLV2PTCALSET1	VT_R8	Targeted value (of the first point of operation value two point calibration[3]) (Only the main phone :.) -..99.999 -+.. 99.999 (value :+ of initial5.000)	✓	✓
@CALCV2PTCALSET2	VT_R8	Targeted value (of the second point of operation value two point calibration[3]) (Only the main phone :.) -..99.999 -+.. 99.999 (value :+ of initial10.000)	✓	✓
@CALCV3PTCALSET1	VT_R8	Targeted value (of the first point of operation value three point calibration[3]) (Only the main phone :.) -..99.999 -+.. 99.999 (value :+ of initial5.000)	✓	✓
@CALCV3PTCALSET3	VT_R8	Targeted value (of the third point of operation value three point calibration[3]) (Only the main phone :.) -..99.999 -+.. 99.999 (value :+ of initial10.000)	✓	✓
@CalcMode	VT_I4	Calculation function (Only the main phone :.) 0 - 2 (Initial value: 0) 0 : OFF 1 : Addition 2 : Subtraction	✓	✓

@MeasureDirect	VT_I4	Direction of measurement 0 - 1 (Initial value: 0) 0 : Usually 1 : Reversing	✓	✓
@SamplingCycle	VT_I4	Sampling period 0 - 4 (Initial value: 0) 0 : Default 1 : 0.33ms 2 : 1ms 3 : 2ms 4 : 5ms	✓	✓
@AverageTimes	VT_I4	Average frequency and difference count filter high-pass filter 0 - 14 (Initial value: 4) 0 : Once 1 : Twice 2 : Four times 3 : Eight times 4 : 16 times 5 : 32 times 6 : 64 times 7 : 128 times 8 : 256 times 9 : 512 times 10 : 1024 times 11 : 2048 times 12 : 4096 times 13 : Difference count filter 14 : High-pass filter	✓	✓
@Output	VT_I4	Output style 0 - 1 (Initial value: 0) 0 : Normal opening 1 : Normal close	✓	✓
@Hold	VT_I4	Setting of hold facility 0 - 5 (Initial value: 0) 0 : Sample holding	✓	✓

		1 : Peak holding 2 : Bottom holding 3 : Peak to peak holding 4 : Auto peak holding 5 : Auto bottom holding		
@HoldSettings	VT_R8	Auto peak holding or auto bottom holding trigger level ([3]) -..99.999 -+.. 99.999 (value :+ of initial1.000)	✓	✓
@TimingInput	VT_I4	Timing input setting 0 - 1 (Initial value: 0) 0 : Level 1 : Edge	✓	✓
@DelayTimerSetting	VT_I4	Delay timer 0 - 3 (Initial value: 0) 0 : OFF 1 : On DeLay 2 : Off DeLay 3 : One shot	✓	✓
@TimerTime	VT_I4	Time of timer (Unit: Ms) 5 - 9999 (Initial value: 60)	✓	✓
@Hysteresis	VT_R8	Hysteresis ([3]) 0.000 - 99.999 (Initial value: 0.000)	✓	✓
@AnalogSetting	VT_I4	Analog output scaling (Only the main phone :.) 0 - 2 (Initial value: 0) 0 : Initial state 1 : Free cooking stove 2 : Bank	✓	✓
@AnalogOutput_Hi	VT_R8	Analog output upper bound value ([3]) (Only the main phone :.) -..99.999 -+.. 99.999 (value :+ of initial10.000)	✓	✓
@AnalogOutput_Lo	VT_R8	Analog output lower bound value ([3]) (Only the main phone :.) -..99.999 -+.. 99.999 (value :- of	✓	✓

		initial10.000)		
@ExternalInput	VT_I4	External input 1 - Function allocation of four 0 - 1 (Initial value: 0) 0 : Initial state 1 : User setting	✓	✓
@ExternalInput1	VT_I4	Function that allocates it in external input 1 0 - 4 (Initial value: 0) 0 : 0 shift input 1 : Bank A input 2 : Bank B input 3 : Laser [**hika] stop input 4 : Do not use it.	✓	✓
@ExternalInput2	VT_I4	Function that allocates it in external input 2 0 - 4 (Initial value: 0) 0 : Reset input 1 : Bank A input 2 : Bank B input 3 : Laser [**hika] stop input 4 : Do not use it.	✓	✓
@ExternalInput3	VT_I4	Function that allocates it in external input 3 0 - 4 (Initial value: 0) 0 : Timing input signal 1 : Bank A input 2 : Bank B input 3 : Laser [**hika] stop input 4 : Do not use it.	✓	✓
@ExternalInput4	VT_I4	Function that allocates it in external input 4 0 - 3 (Initial value: 0) 0 : 0 shift input 1 : Bank A input 2 : Bank B input 3 : Laser [**hika] stop input	✓	✓
@BankSwitching	VT_I4	Method of bank changing 0 - 1 (Initial value: 0) 0 : Button 1 : External input	✓	✓

@ZeroShiftMem	VT_I4	0 shift value storage function 0 - 1 (Initial value: 0) 0 : OFF 1 : ON	✓	✓
@InterferencePrevention	VT_I4	Mutual interference prevention function 0 - 1 (Initial value: 0) 0 : Interference prevention OFF 1 : Interference prevention ON	✓	✓
@DisplayDigit	VT_I4	Display digit (initial value: 0) 0 : Initial state 2 : 0.001 3 : 0.01 4 : 0.1 5 : 1	✓	✓
@EcoMode	VT_I4	Power saving mode 0 - 2 (Initial value: 0) 0 : OFF 1 : Half 2 : Oar	✓	✓
@HeadDisp	VT_I4	Head display mode 0 - 2 (Initial value: 0) 0 : Initial state 1 : OK/NG display 2 : OFF	✓	✓
@DisplayColor	VT_I4	Display color of amplifier 0 - 2 (Initial value: 0) 0 : GO green 1 : GO red 2 : Red always	✓	✓
@OneShotTime	VT_I4	One shot output ON time of difference count filter (Unit: Ms) 2 - 9999 (Initial value: 10)	✓	✓
@CutoffFrequency	VT_I4	Cutoff frequency of high-pass filter 0 - 9 (Initial value: 0) 0 : 0.1Hz	✓	✓

		1 : 0.2Hz 2 : 0.5Hz 3 : 1Hz 4 : 2Hz 5 : 5Hz 6 : 10Hz 7 : 20Hz 8 : 50Hz 9 : 100Hz		
@AlarmSetting	VT_I4	Alarm setting 0 - 2 (Initial value: 0) 0 : Initial state 1 : Clamping 2 : User setting	✓	✓
@NumberOfAlarms	VT_I4	Alarm frequency 2 - 1000 (Initial value: 7)	✓	✓
@LAST_RESPONSE_STATUS	VT_ARRAY | VT_I4	Return error reply detailed information immediately before. (VT_ARRAY | VT_I4) [0] General status [1] Additional status (0 sets in case of not being)	✓	✓

6. Error code

[in] the ILDLEP1 provider, an error code that as follows and is peculiar is defined.

About the ORiN2 commonness error, Refer to the chapter of the error code of "ORiN2SDK user's guide".

Table6-1Peculiar error code

Error name	Error number	Explanation
The CIP response is abnormal. (format error)	0x80100000	It returns when the response packet of the CIP communication is an abnormal format outside assumption.
The CIP response is abnormal. (Error Status Code)	0x80100001	It returns when the response packet of the CIP communication is status abnormal.
The CIP response is abnormal. (Error General Status)	0x80100002	It returns when the response packet of the CIP communication is an error reply. When this error is received, the detailed error code can be obtained by reading @LAST_RESPONSE_STATUS of the system variable.

Table6-2Detailed error code

General status	Additional status	Explanation
0x00	0x0000	Normal termination.
0x05	0x0000	Instance ID is outside the range.
0x08	0x0000	This service is non-correspondence in specified instance ID or it is not possible to execute it.
0x09	0x0000	The written data is outside the range.
0x0C	0x0000	・ The parameter writing was executed the function that controlled by a cyclic communication. ・ It failed in the execution of the operation instruction. The sensor amplifier is a state that the operation instruction can be executed or confirm it.
0x0E	0x0000	It tries to write it in attribute ID in attribute ID that cannot be

		written or the state that cannot be written.
0x10	0x0000	It tries to read it from attribute ID in attribute ID that cannot be read or the state that cannot be read.
0x13	0x0000	The size of the service data is following regulations. The provided data size is stored in additional status.
0x14	0x0000	Attribute ID is outside the range.
0x16	0x0000	The sensor amplifier corresponding to specified instance ID is not connected.
0x1F	0xC350	It is not possible to use it though this service is supported by specified instance ID by specified attribute ID.
	0xC351	The specified set value cannot be written in the parameter in a present mode.
	0xC352	This machine is initializing the communication.
0xFE	0x0000	System error.

7. Sample program

Sample code (C#) is shown as follows.

7.1. Connection and generation of object

```
// Connect with the communication unit, and generate the object accessed the amplifier unit
and the sensor head.
private void Connect()
{
 try
 {
// Generation of CAO engine
 m_CaoEngine = new CCaoEngine();
// Workspace acquisition
 m_CaoWorkspace = m_CaoEngine.Workspaces[0];
// Connection to communication unit
 m_CaoController = m_CaoWorkspace.AddController("Sample",
 "CaoProv.KEYENCE.ILDLEP1", "", "Conn=TCP:192.168.0.20");
// Amplifier unit object generation
 m_CaoExtension = m_CaoController.AddExtension("CH1", "");
// Variable object generation
 m_BankHigh = m_CaoExtension.AddVariable("@Bank_HIGH", "");
 m_CriteriaValue = m_CaoExtension.AddVariable("@CriteriaValue", "");
 }
 catch (Exception e)
 {
 MessageBox.Show(this, e.Message);
 }
}
```

7.2. Sensor ID=1 reset

```
// Reset sensor ID=1.
private void Reset()
{
 try
 {
// Reset execution demand
 m_CaoExtension.Execute("Reset");
 }
 catch (Exception e)
 {
 MessageBox.Show(this, e.Message);
 }
}
```

7.3. Bank 2 of sensor ID=1 Setting of HIGH side setting value

```
// bank 2 of sensor ID=1 Setting of HIGH side setting value
private void PutBank_HIGH()
{
 try
 {
// Setting of bank No
 m_BankHigh.ID = 2;
// Setting of value
 Double val;
 if (Double.TryParse(txtPutBank_HIGH.Text, out val))
 {
 m_BankHigh.Value = val;
 }
 }
 catch (Exception e)
 {
 MessageBox.Show(this, e.Message);
 }
}
```

7.4. Acquisition of judgment value of sensor ID=1 (P.V . value)

```
// Acquire the judgment value of sensor ID=1 (P.V . value).
private void GetCriteriaValue()
{
 try
 {
// Acquisition of value
 Object val = m_CriteriaValue.Value;
// Reflection of result
 txtGetCriteriaValue.Text = val.ToString();
 }
 catch (Exception e)
 {
 MessageBox.Show(this, e.Message);
 }
}
```

7.5. Acquisition of judgment value of sensor ID=1 (P.V . value)(Raw command)

```
// Acquire the judgment value of sensor ID=1 (P.V . value) by using the Raw command.
private void ExecuteRawGetCriteriaValue()
{
 try
 {
 txtRawResult.Text = "";
```

```
// Command data creation
 Byte[] param = new Byte[5];
 param[0] = 0x0E;    // service code: Parameter reading
 param[1] = 0x67;    // class ID: DL Object class
 param[2] = 0x01;    // instance ID: Sensor ID=1
 param[3] = 0x25;    // attribute ID(Low): Judgment value (P.V . value) (0x0325)
 param[4] = 0x03;    // attribute ID(High): ↑
// Raw command execution
 Object result = m_CaoController.Execute("Raw", param);
It ..//.. reflects it.
 String strResult = String.Empty;
 foreach (Byte b in (Byte[])result)
 {
 strResult += " ";
 strResult += b.ToString();
 }
 txtRawResult.Text = strResult;
 }
 catch (Exception e)
 {
 MessageBox.Show(this, e.Message);
 }
}
```

7.6. Acquisition of status of communication unit (Raw command)

```
// Acquire the status of the communication unit by using the Raw command.
private void ExecuteRawGetStatus()
{
 try
 {
 txtRawResult.Text = "";
// Command data creation
 Byte[] param = new Byte[5];
 param[0] = 0x0E;    // service code: Parameter reading
 param[1] = 0x67;    // class ID: DL Object class
 param[2] = 0x00;    // instance ID: Communication unit =0
 param[3] = 0x64;    // attribute ID(Low): Status (0x0064)
 param[4] = 0x00;    // attribute ID(High): ↑
// Raw command execution
 Object result = m_CaoController.Execute("Raw", param);
It ..//.. reflects it.
 String strResult = String.Empty;
 foreach (Byte b in (Byte[])result)
 {
 strResult += " ";
 strResult += b.ToString();
 }
 txtRawResult.Text = strResult;
 }
 catch (Exception e)
 {
```

```
            MessageBox.Show(this, e.Message);
            }
        }
```