

KEYENCE FSN40NUEP1 provider

Version 1.0.0

User's guide

May 13, 2020

Remarks:

This document uses the machine translation.

【 revision history 】

Version	Date	Content
1.0.0	2018-7-26	First edition.
	2020-5-13	Fixed a typo.

【 ..confirming the operation.. model 】

Model	Version	Notes
FS-N40		Communication unit (NU-EP1) Composition of + sensor unit (FS-N40)

Contents

1. Introduction.....	5
2. Outline of provider.....	6
2.1. Installation.....	6
2.2. Outline.....	6
3. Method property.....	7
3.1. CaoWorkspace::AddController method.....	7
3.1.1. Conn is optional.....	7
3.2. CaoControllere::Execute method.....	8
3.3. CaoController::AddVariable method.....	8
3.4. CaoController::AddExtension method.....	9
3.5. CaoExtension::Execute method.....	9
3.6. CaoExtension::AddVariable method.....	9
3.7. CaoController::get_ExtensionNames property.....	9
3.8. CaoController::get_VariableNames property.....	9
3.9. CaoExtension::get_VariableNames property.....	10
3.10. CaoVariable::put_Value property.....	10
3.11. CaoVariable::get_Value property.....	10
4. Command list.....	11
4.1. CaoController class.....	11
4.1.1. CaoController::Execute("Raw") command.....	11
4.2. CaoExtension class.....	12
4.2.1. CaoExtension::Execute("Initialize") command.....	12
4.2.2. CaoExtension::Execute("CustomSave") command.....	13
4.2.3. CaoExtension::Execute("MaxSensitivityCalibration") command.....	13
4.2.4. CaoExtension::Execute("PercentageCalibration") command.....	13
4.2.5. CaoExtension::Execute("TwoPointCalibration1st") command.....	13
4.2.6. CaoExtension::Execute("TwoPointCalibration2nd") command.....	13
4.2.7. CaoExtension::Execute("PositioningCalibration1st") command.....	14
4.2.8. CaoExtension::Execute("PositioningCalibration2nd") command.....	14
4.2.9. CaoExtension::Execute("StartFullAutoCalibration") command.....	14
4.2.10. CaoExtension::Execute("FinishFullAutoCalibration") command.....	14
4.2.11. CaoExtension::Execute("ZeroShift") command.....	14
4.2.12. CaoExtension::Execute("CancelZeroShift") command.....	15
4.2.13. CaoExtension::Execute("Preset") command.....	15
4.2.14. CaoExtension::Execute("CancelPreset") command.....	15
4.2.15. CaoExtension::Execute("SaturationCanceling") command.....	15

4.2.16. CaoExtension::Execute("CancelSaturationCanceling") command	15
4.2.17. CaoExtension::Execute("ResetHoldValue") command	16
4.2.18. CaoExtension::Execute("WorkPreset") command	16
4.2.19. CaoExtension::Execute("StartFullAutoPreset") command.....	16
4.2.20. CaoExtension::Execute("FinishFullAutoPreset") command.....	16
4.2.21. CaoExtension::Execute("MaxSensitivityPreset") command	16
4.2.22. CaoExtension::Execute("ResetUser") command.....	17
4.2.23. CaoExtension::Execute("CancelKeyLock") command.....	17
4.2.24. CaoExtension::Execute("KeyLock") command	17
4.2.25. CaoExtension::Execute("PINKeyLock") command	17
4.2.26. CaoExtension::Execute("CommunicationKeyLock") command	17
5. Variable list	19
5.1. CaoController class.....	19
5.1.1. System variable	19
5.1.2. User variable	19
5.2. CaoExtension class	20
5.2.1. System variable	20
5.2.2. User variable	31
6. Error code.....	32
7. Sample program.....	34
7.1. Connection and generation of object	34
7.2. Acquisition of value that uses Variable	34
7.3. Setting of value that uses Variable	34
7.4. Execution of command that uses Execute of Extension	35
7.5. Execution of command that uses Execute of Controller.....	35

1. Introduction

KEYENCE FSN40NUEP1 provider is an ORiN2 CAO provider that accesses KEYENCE digital fiber sensor (FS-N40) through communication unit (NU-EP1).

In this document, it explains the outline of the FSN40NUEP1 provider and the mounted CAO interface (function specification).

2. Outline of provider

2.1. Installation

The FSN40NUEP1 provider module is composed of following DLL. The installation work is unnecessary when installing it with the installer of ORiN2 SDK.

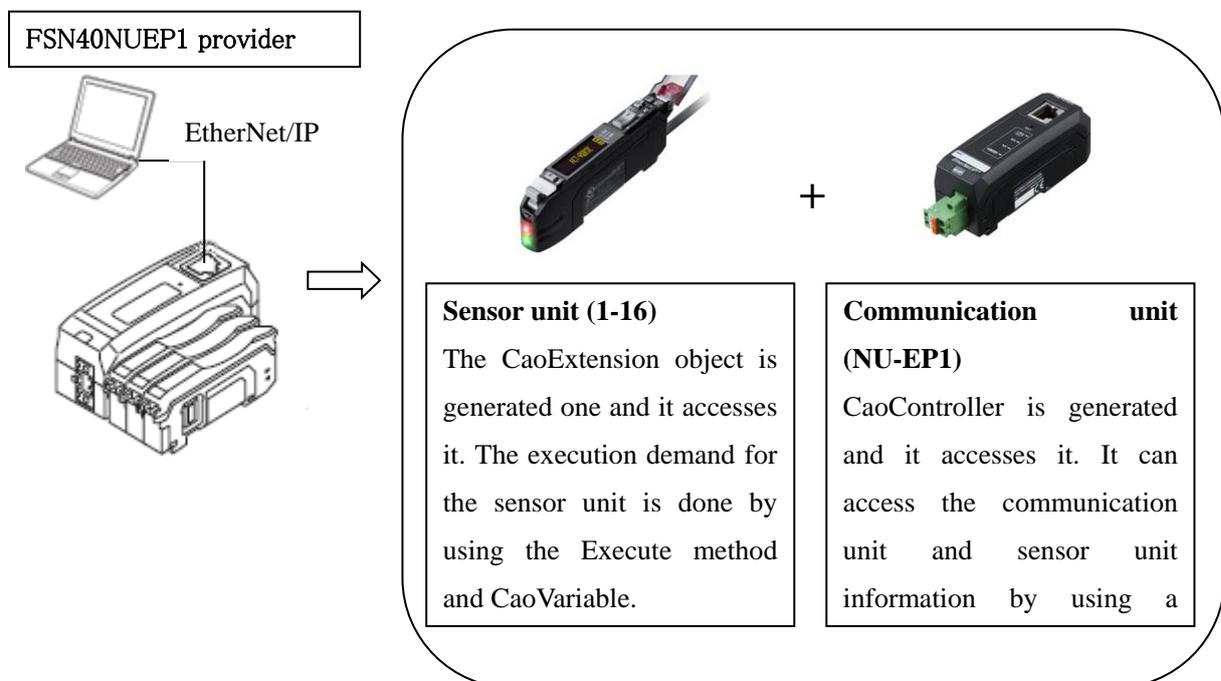
Table2-1FSN40NUEP1 provider

File name	CaoProvKEYENCE FSN40NUEP1. dll
ProgID	CaoProv.KEYENCE. FSN40NUEP1
Registry registration	regsvr32 CaoProvKEYENCE FSN40NUEP1. dll
Blotting out of registry registration	regsvr32 /u CaoProvKEYENCE FSN40NUEP1. dll

2.2. Outline

The FSN40NUEP1 provider accesses NU-EP1 by using the CIP protocol.

The CaoExtension object of each sensor unit (FS-N40) connected with NU-EP1 is generated, and it accesses information on the sensor by the execution of the Execute method or reading and writing to CaoVariable.



3. Method property

3.1. CaoWorkspace::AddController method

The FSN40NUEP1 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. FSN40NUEP1"

bstrPcName : Execution machine name of in provider

bstrOption : In optional character string

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

Table3-1 Optional character string of CaoWorkspace::AddController

¹ Option ()	Explanation
Conn=< connected parameter >	Indispensability. Communication form and connected parameter. (refer to 3.1.1)
MyIP[=< Internet Protocol address >]	Internet Protocol address. (two or more NIC usage) (default: There is no specification.)
ConnTimeout [=< timeout period >]	Timeout period when TCP is connected. (millisecond) (default: 3000)
Timeout [=< timeout period >]	Timeout period when sending and receiving it. (millisecond) (default: 5000)

3.1.1. Conn is optional.

A communication form and connected parameter are specified.

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

¹ Items enclosed with square brackets ("[]") are omissible. Underlined part shows the default value when the option is not specified..

Ethernet/IP device

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

- < Dest IP Address > : Internet Protocol address TCP/IP connection ahead.
Example: "127.0.0.1", "192.168.0.1"
- <Dest Port No> : TCP/IP connection port number.
Example: 44818,5006,5007
- <Src IP Address> : ²Internet Protocol address. (two or more NIC usage) ()
- <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. The command name is specified in the first argument and the parameter of the command is specified for the second argument. Details of each command, please refer to the chapter 4.1.

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. The variable list that can be used, please refer to the chapter 5.1.

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

- <bstrVariableName> : In variable identifier
- <bstrOption> : In optional character string

² An error occurs if own IP address is specified both in the Conn option and in the MyIP option. When you specify an IP address, be sure to specify it in one option only.

3.4. CaoController::AddExtension method

The AddExtension method of the CaoController class is Extension of each sensor unit connected with NU-EP1.

It is a method for making the object.

Format AddExtension(<bstrChannelNo:VT_BSTR>)

< bstrChannelNo> : [In] Identification number of in sensor unit.
It specifies it by the format of "CH < identification number >".
(identification number: 1-16)

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. Details of each command, please refer to the chapter 4.2.

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. The variable list that can be used, please refer to the chapter 5.2.

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

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

3.7. CaoController::get_ExtensionNames property

The Extension person list is acquired.

3.8. CaoController::get_VariableNames property

The variable identifier list generated from CaoController is acquired.

3.9. CaoExtension::get_VariableNames property

The variable identifier list generated from CaoExtension is acquired.

3.10. CaoVariable::put_Value property

The data specified by the variable identifier is set. A set value sets the value specified by the argument.

3.11. CaoVariable::get_Value property

The data specified by the variable identifier is acquired.

4. Command list

4.1. CaoController class

Table4-1CaoController class command list

Command name	Function	
Raw	A general-purpose command is sent.	P11

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

A general-purpose command is sent.

Format Raw (<vntDataArray>)

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

- When attribute ID is 8Bit, the High side is specified with 0x00.
- It treats since element 5 as service data.
- The service data ..Low side.. ..8Bit.. is specified.
- The NULL(0x00) padding is done internally when the service data is odd number Byte.

Return value : Out response data
(VT_UI1 | VT_ARRAY)

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

4.2. CaoExtension class

Table4-2CaoExtension class command list

Command name	Function	
Initialize	Initialization is executed.	P12
CustomSave	A present setting is written in "Custom save area" of the sensor.	P13
MaxSensitivityCalibration	The maximum sensitivity tuning is executed.	P13
PercentageCalibration	Percent tuning is executed.	P13
TwoPointCalibration1st	The first point of two point tuning is specified.	P13
TwoPointCalibration2nd	The second point of two point tuning is specified.	P13
PositioningCalibration1st	The first point of the positioning tuning is specified.	P14
PositioningCalibration2nd	The second point of the positioning tuning is specified.	P14
StartFullAutoCalibration	The full auto tuning begins.	P14
FinishFullAutoCalibration	The full auto tuning is ended.	P14
ZeroShift	0 shifts are executed.	P14
CancelZeroShift	0 shifts are released.	P15
Preset	The preset is executed.	P15
CancelPreset	The preset is released.	P15
SaturationCanceling	The saturation evasion is executed.	P15
CancelSaturationCanceling	The saturation evasion is released.	P15
ResetHoldValue	The holding values of the peak value and the bottom value, etc. are reset.	P16
WorkPreset	The work preset is executed.	P16
StartFullAutoPreset	The full auto preset (Or, maximum sensitivity preset) is begun.	P16
FinishFullAutoPreset	The full auto preset (Or, maximum sensitivity preset) is ended.	P16
MaxSensitivityPreset	The maximum sensitivity preset is executed. (It is possible to execute it specifying neither beginning nor the end.)	P16
ResetUser	The setting preserved in "Custom save area" is read.	P17
CancelKeyLock	The key lock release is done.	P17
KeyLock	The key is locked.	P17
PINKeyLock	The key lock with the password is done.	P17
CommunicationKeyLock	The communication key is locked.	P17

4.2.1. C

a
o
E
x
t
e
n
s
i
o
n
:
E
x
e
c
u
t
e
(
"
I
n
i
t
i
a
l
i
z
e

) command

Initialization is executed.

Format Initialize ()

Argument : None.
Return value : None.

4.2.2. **CaoExtension::Execute("CustomSave")** command

A present setting is written in "Custom save area" of the sensor.

Format CustomSave ()

Argument : None.
Return value : None.

4.2.3. **CaoExtension::Execute("MaxSensitivityCalibration")** command

The maximum sensitivity tuning is executed.

Format MaxSensitivityCalibration ()

Argument : None.
Return value : None.

4.2.4. **CaoExtension::Execute("PercentageCalibration")** command

Percent tuning is executed.

Format PercentageCalibration ()

Argument : None.
Return value : None.

4.2.5. **CaoExtension::Execute("TwoPointCalibration1st")** command

The first point of two point tuning is specified.

Format TwoPointCalibration1st ()

Argument : None.
Return value : None.

4.2.6. **CaoExtension::Execute("TwoPointCalibration2nd")** command

The second point of two point tuning is specified.

Format TwoPointCalibration2nd ()

Argument : None.
Return value : None.

4.2.7. **CaoExtension::Execute("PositioningCalibration1st") command**

The first point of the positioning tuning is specified.

Format PositioningCalibration1st ()

Argument : None.
Return value : None.

4.2.8. **CaoExtension::Execute("PositioningCalibration2nd") command**

The second point of the positioning tuning is specified.

Format PositioningCalibration2nd ()

Argument : None.
Return value : None.

4.2.9. **CaoExtension::Execute("StartFullAutoCalibration") command**

The full auto tuning begins.

Format StartFullAutoCalibration ()

Argument : None.
Return value : None.

4.2.10. **CaoExtension::Execute("FinishFullAutoCalibration") command**

The full auto tuning is ended.

Format FinishFullAutoCalibration ()

Argument : None.
Return value : None.

4.2.11. **CaoExtension::Execute("ZeroShift") command**

0 shifts are executed.

Format ZeroShift ()

Argument : None.
Return value : None.

4.2.12. **CaoExtension::Execute("CancelZeroShift")** command

0 shifts are released.

Format CancelZeroShift ()

Argument : None.
Return value : None.

4.2.13. **CaoExtension::Execute("Preset")** command

The preset is executed.

Format Preset ()

Argument : None.
Return value : None.

4.2.14. **CaoExtension::Execute("CancelPreset")** command

The preset is released.

Format CancelPreset ()

Argument : None.
Return value : None.

4.2.15. **CaoExtension::Execute("SaturationCanceling")** command

The saturation evasion is executed.

Format SaturationCanceling ()

Argument : None.
Return value : None.

4.2.16. **CaoExtension::Execute("CancelSaturationCanceling")** command

The saturation evasion is released.

Format CancelSaturationCanceling ()

Argument : None.
Return value : None.

4.2.17. **CaoExtension::Execute("ResetHoldValue") command**

The holding values of the peak value and the bottom value, etc. are reset.

Format ResetHoldValue ()

Argument : None.
Return value : None.

4.2.18. **CaoExtension::Execute("WorkPreset") command**

The work preset is executed.

Format WorkPreset ()

Argument : None.
Return value : None.

4.2.19. **CaoExtension::Execute("StartFullAutoPreset") command**

The full auto preset (Or, maximum sensitivity preset) is begun.

Format StartFullAutoPreset ()

Argument : None.
Return value : None.

4.2.20. **CaoExtension::Execute("FinishFullAutoPreset") command**

The full auto preset (Or, maximum sensitivity preset) is ended.

Format FinishFullAutoPreset ()

Argument : None.
Return value : None.

4.2.21. **CaoExtension::Execute("MaxSensitivityPreset") command**

The maximum sensitivity preset is executed. (It is possible to execute it specifying neither beginning nor the end.)

Format MaxSensitivityPreset ()

Argument : None.
Return value : None.

4.2.22. **CaoExtension::Execute("ResetUser")** command

The setting preserved in "Custom save area" is read.

Format ResetUser ()

Argument : None.
Return value : None.

4.2.23. **CaoExtension::Execute("CancelKeyLock")** command

The key lock release is done.

Format CancelKeyLock ()

Argument : None.
Return value : None.

4.2.24. **CaoExtension::Execute("KeyLock")** command

The key is locked.

Format KeyLock ()

Argument : None.
Return value : None.

4.2.25. **CaoExtension::Execute("PINKeyLock")** command

The key lock with the password is done.

Format PINKeyLock ()

Argument : None.
Return value : None.

4.2.26. **CaoExtension::Execute("CommunicationKeyLock")** command

The communication key is locked.

Format CommunicationKeyLock()

Argument : None.

Return value : None.

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	"KEYENCE maker name =" is returned.	✓	-
@VERSION	VT_BSTR	Provider version information is returned.	✓	-
@LAST_RESPONSE_STATUS	VT_I4 VT_ARRAY	Error reply detailed information immediately before. (VT_I4 VT_ARRAY) [0] General status [1] Additional status (0 sets in case of not being)	✓	-

5.1.2. User variable

Table5-2CaoController class user variable list

Variable identifier	Data type	Explanation	Attribute	
			get	put
None.			-	-

5.2. CaoExtension class

5.2.1. System variable

Table5-3CaoExtension class system variable list

Variable identifier	Data type	Explanation (* is an initial value).	Attribute	
			get	put
@ErrorInfomation	VT_UI2	Error information. Bit0: Overcurrent error. Bit1: System reservation. Bit2: EEPROM error. Bit3: System reservation. Bit4: System reservation. Bit5: DATUM correction error Bit6: APC correction error Bit7: Communication error between sensors	✓	-
@WarningInfomation	VT_UI2	Warning information. Bit0: Prospect maintenance	✓	-
@WarningFunction	VT_UI2	State of warning function operation. Bit0: Prospect maintenance	✓	-
@Output	VT_UI2	Output. Bit0: Output 1 Bit1: Output 2	✓	-
@CurrentValue	VT_UI2	Receiving quantities of light present value. 0 ~ 65535	✓	-
@RawValue	VT_UI2	Receiving quantities of light life value. 0 ~ 65535	✓	-
@PeakHoldValue	VT_UI2	Peak holding value. 0 ~ 65535	✓	-
@BottomHoldValue	VT_UI2	Bottom holding value. 0 ~ 65535	✓	-
@PeakHoldMinValue	VT_UI2	Minimum value of peak holding. 0 ~ 65535	✓	-

@PresetDatum	VT_UI2	Preset/DATUM internal and setting value. 0 ~ 65535	✓	-
@DisplayCustomize	VT_I2	State of indicated value correction. 0 : Unused/ DATUM 1: Preset 2: 0 shifts 3: Work preset 4: Work preset (reversing)	✓	-
@SlideSwitchStatus	VT_I2	State of slide switch. 0 : [SEL]/[1ch] Selected. 1: [TERA]/[2ch] selected.	✓	-
@SaturationCancelingStatus	VT_I2	Saturation evasion running state. 0 : OFF 1 : ON	✓	-
@OperationMode	VT_I2	State of operation mode. 0 : It is operating usually. 1 : flood light off 2 : sleep 3: The output is being fixed.	✓	-
@ SetOperationMode	VT_I2	Operation mode setting. 0 : It is operating usually. 1 : flood light off 2 : sleep 3: The output is being fixed.	✓	✓
@SettingValueHi	VT_UI2	Set value. (output 1) /(area upper bound value) 0 - 65535 (initial value: 100)	✓	✓
@SettingValueLow	VT_UI2	Set value. (area lower bound value) 0 - 65535 (initial value: 100)	✓	✓
@LimitDetection	VT_I2	Prospect maintenance detection. 0 : OFF 1 : ON	✓	✓
@LimitDetectionPoint	VT_I2	Prospect maintenance setting value. 0 - 100 (initial value: 5)	✓	✓

@ResetLimitDetection	VT_I2	Prospect maintenance detection reset mode. 0 :User reset 1: Self resetting	✓	✓
@CSettingValueHi	VT_UI2	Custom setting value. (output 1) /(area upper bound value) 0 - 65535 (initial value: 100)	✓	✓
@CSettingValueLow	VT_UI2	Custom setting value. (area lower bound value) 0 - 65535 (initial value: 100)	✓	✓
@CLimitDetection	VT_I2	Custom prospect maintenance detection. 0 : OFF 1 : ON	✓	✓
@CLimitDetectionPoint	VT_I2	Custom prospect maintenance setting value. 0 - 100 (initial value: 5)	✓	✓
@CResetLimitDetection	VT_I2	Custom prospect maintenance detection reset mode. 0 :User reset 1: Self resetting	✓	✓
@KeyLockStatus	VT_I2	State of key lock. 0 :Release 1: Key lock 2: Password key lock 3: Communication lock	✓	✓
@PINCode	VT_I2	Key lock password. 0 - 9999 (initial value: 0)	✓	✓
@SlideSwitchLock	VT_I2	Slide switch lock. 0 : not locked 1: It locks.	✓	✓
@L-onD-on	VT_UI2	L-on/D-on.(N.O./N.C.) Bit0: Output 1 Bit1: Output 2 1 (initial value: 0)	✓	✓

@PowerModes	VT_I2	Power mode. 0 : S-HSPD (response time 23us) 1: HSPD (response time 50us) 2: FINE (response time 250us) * 3: TURBO (response time 500us) 4: SUPER (response time 1ms) 5: ULTRA (response time 4ms) 6: MEGA (response time 16ms) 7 : TERA (response time 64ms)	✓	✓
@PercentageCalibrationTarget	VT_I2	Percent tuning target value. -99 - 99 (initial 10 ..value :-..)	✓	✓
@AreaPercentageTarget	VT_I2	Area percent target value. 1 - 99 (initial value: 10)	✓	✓
@Hysteresis	VT_I2	Hysteresis. 0 : Minimum 1 : Smallness 2: Standard * 3: Largeness 4: Maximum	✓	✓
@DisplayBar	VT_I2	Bar display. 0 : OFF * 1 : ON	✓	✓
@KeyLockMethod	VT_I2	Key lock method. 0 :Standard * 1: With password	✓	✓
@CKeyLockStatus	VT_I2	State of custom key lock. 0 :Release * 1: Key lock 2: Password key lock 3: Communication lock	✓	✓
@CPINCode	VT_I2	[Custom] key lock password. 0 - 9999 (initial value: 0)	✓	✓
@CSlideSwitchLock	VT_I2	[Custom] slide switch lock. 0 : unlock 1 : lock	✓	✓

@CL-onD-on	VT_UI2	[Custom] L-on/D-on.(N.O./N.C.) Bit0: Output 1 Bit1: Output 2 *1 (initial value: 0)	✓	✓
@CPowerModes	VT_I2	[Custom] power mode. 0 : S-HSPD (response time 23us) 1: HSPD (response time 50us) 2: FINE (response time 250us) * 3: TURBO (response time 500us) 4: SUPER (response time 1ms) 5: ULTRA (response time 4ms) 6: MEGA (response time 16ms) 7: Response time 64 TERA (ms)	✓	✓
@CPercentageCalibrationTarget	VT_I2	[Custom] percent tuning target value. -99 - 99 (initial 10)	✓	✓
@CAreaPercentageTarget	VT_I2	[Custom] area percent target value. 1 - 99 (initial value: 10)	✓	✓
@CHysteresis	VT_I2	[Custom] hysteresis. 0 : Minimum 1 : Smallness 2: Standard * 3: Largeness 4: Maximum	✓	✓
@CDisplayBar	VT_I2	[Custom] bar display. 0 : OFF * 1 : ON	✓	✓
@CKeyLockMethod	VT_UI2	[Custom] key lock method. 0 :Standard * 1: With password	✓	✓
@OutputTimer	VT_I2	Timer function. (output 1) 0 :Off timer * 1: Off delay timer 2: On delay timer 3: One shot timer 4: Off on DeLay DeLay 5: DeLay on one shot	✓	✓

@TimerDuration1	VT_I2	Time of timer. (output 1) 1 - 9999 (initial value: 10)	✓	✓
@TimerDuration2	VT_I2	Time of timer. (output 2) 1 - 9999 (initial value: 10)	✓	✓
@DetectionMode	VT_I2	Detection mode. (output 1) 0 :Standard * 1: Preset 2: Percent tuning 3: 0 shift tuning 4: Area detection 5: Area percent 6: DATUM 7: on edge detection 8: off edge detection	✓	✓
@DATUMSpeed	VT_I2	DATUM correction interval. 0 : LEVEL1 (slow)* 1: LEVEL2 (usually) 2: LEVEL3 (fast)	✓	✓
@DATUMWarningLevel	VT_I2	DATUM warning output level. 0 - 100 (initial value: 50)	✓	✓
@ExternalInput	VT_I2	External input setting. 0 : don't use external input. 1 : External tuning 2: 0 shifts 3: Reset 4: light OFF 5: Output fixation 6: Sleep mode	✓	✓
@PauseMode	VT_I2	Output fixation type. 0 : Turning off fixation * 1: turning on fixation 2: It fixes by the output when the external is input.	✓	✓
@Attenuation	VT_I2	[Attene-to]. 1 - 100 (initial value: 100)	✓	✓

@COutputTimer	VT_I2	[Custom] timer function. (output 1) 0 :Off timer * 1: Off delay timer 2: On delay timer 3: One shot timer 4: Off on DeLay DeLay 5: DeLay on one shot	✓	✓
@CTimerDuration1	VT_I2	Time of custom timer. (output 1) 1 - 9999 (initial value: 10)	✓	✓
@CTimerDuration2	VT_I2	Time of custom timer. (output 2) 1 - 9999 (initial value: 10)	✓	✓
@CDetectionMode	VT_I2	[Custom] detection mode. (output 1) 0 :Standard * 1: Preset 2: Percent tuning 3: 0 shift tuning 4: Area detection 5: Area percent 6: DATUM 7: on edge detection 8: off edge detection	✓	✓
@CDATUMSpeed	VT_I2	[Custom] DATUM correction interval. 0 : LEVEL1 (It is slow)* 1: LEVEL2 (usually) 2: LEVEL3 (It is fast.)	✓	✓
@CDATUMWarningLevel	VT_I2	[Custom] DATUM warning output level. 0 - 100 (initial value: 50)	✓	✓
@CEXternalInput	VT_I2	The custom outside input setting. 0 :don't use external input. * 1 : External tuning 2: 0 shifts 3: Reset 4: light OFF 5: Output fixation 6: Sleep mode	✓	✓

@CPauseMode	VT_I2	[Custom] output fixation type. 0 : Turning off fixation * 1: turning on fixation 2: It fixes by the output when the external is input.	✓	✓
@CAttenuation	VT_I2	[Custom] [attene-to]. 1 - 100 (initial value: 100)	✓	✓
@FlipDisplay	VT_I2	Display reversing. 0 : usually Display * 1: Reversing display	✓	✓
@SubDisplay	VT_I2	Subdisplay. 0 : OFF * 1: Room degree display 2: Holding display 3: Room degree holding 4: TAG character string display	✓	✓
@DisplayHold	VT_I2	Holding display. 0 : usually Display * 1: Peak peak maximum value/minimum value 2: Maximum in bottom value/minimum value of bottom 3: peak minimum value / Maximum value of bottom 4: Peak maximum value/minimum value of bottom	✓	✓
@PresetSaturationLevel	VT_I2	Preset saturation level. 100 - 200 (initial value: 110)	✓	✓
@CurrentDisplay	VT_I2	Display selection. 0 : usually Display * 1: Bar display 2: Subdisplay	✓	✓
@DisplayBrightness	VT_I2	Screen brightness. 0 :Brightness: 1 1: Brightness: 2 2: Brightness: 3 *	✓	✓

@ActiveReceiverSetting	VT_I2	Active receiver setting. 0 :Stop 1: Output synchronization * 2: Output reversing 3: Turning on always	✓	✓
@Language	VT_I2	Language choice. 0 :English * 1: Japanese 2: Simplified Chinese characters 3: German	✓	✓
@CFlipDisplay	VT_I2	[Custom] display reversing. 0 : usually Display * 1 :Reversing display	✓	✓
@CSubDisplay	VT_I2	[Custom] subdisplay. 0 : OFF * 1: Room degree display 2: Holding display 3: Room degree holding 4: TAG character string display	✓	✓
@CDisplayHold	VT_I2	[Custom] holding display. 0 : usually Display * 1: Peak maximum value/ peak minimum value 2: bottom Maximum value / bottom minimum value 3: peak maximum value / bottom maximum value 4: Peak maximum value / bottom minimum value	✓	✓
@CPresetSaturationLevel	VT_I2	[Custom] preset saturation level. 100 - 200 (initial value: 110)	✓	✓
@CCurrentDisplay	VT_I2	[Custom] display selection. 0 : usually Display * 1: Bar display 2: Subdisplay	✓	✓

@CDisplayBrightness	VT_I2	[Custom] screen brightness. 0 :Brightness 1 1: Brightness 2 2: Brightness 3 *	✓	✓
@CActiveReceiverSetting	VT_I2	[Custom] active receiver setting. 0 :Stop 1: Output synchronization * 2: Output reversing 3: Turning on always	✓	✓
@CLanguage	VT_I2	[Custom] language selection. 0 :English * 1: Japanese 2: Simplified Chinese characters 3: German	✓	✓
@TAGStrings	VT_BSTR	TAG character string. Initial value ("".eight character or less..)	✓	✓
@ECO	VT_I2	Eco-function. 0 : OFF * 1: Display turning off 2: Low display turning off + operation speed 3: All turning off	✓	✓
@APC	VT_I2	APC function. 0 : OFF * 1 : ON	✓	✓
@DisplayGain	VT_I2	Quantities of light expansion setting. 0 : usually * 1: Four times	✓	✓
@InterferencePrevention	VT_I2	The number of interference prevention. 0 : usually * 1: Twice	✓	✓
@EEPROMSave	VT_I2	External input EEPROM writing, Including. 0 : not preserved 1 : Preserved *	✓	✓

@CTAGStrings	VT_BSTR	[Custom] TAG character string. Initial value ("".eight character or less..)	✓	✓
@CECO	VT_I2	[Custom] eco-function. 0 : OFF * 1: Display turning off 2: Low display turning off + operation speed 3: All turning off	✓	✓
@CAPC	VT_I2	[Custom] APC function. 0 : OFF * 1 : ON	✓	✓
@CDisplayGain	VT_I2	[Custom] quantities of light expansion setting. 0 : usually * 1: Four times	✓	✓
@CInterferencePrevention	VT_I2	The number of custom interference prevention. 0 :* 1 usually: Twice	✓	✓
@CEEPROMSave	VT_I2	[Custom] outside input EEPROM writing, Including. 0 : not preserved 1 : Preserved *	✓	✓
@IDNumber	VT_I2	Number of identification number occupations. FS-N40/42* : 1 FS-N44* : 2	✓	-
@ProductCode	VT_I2	Product code. FS-N40/N42* : 2025 FS-N44* : 2026	✓	-
@Revision	VT_I2	[Ribijon]. 0100H ~ FFFFH	✓	-
@ProductName	VT_BSTR	Model name. FS-N40/N42* : "FS-N40/N42*" FS-N44* : "FS-N44*"	✓	-

@SeriesCode	VT_I2	Series code. FS-N40/N42* : 2025 FS-N44* : 2026	✓	-
@SeriesVersion	VT_I2	Series version. 1	✓	-
@DeviceType	VT_I2	Device type. 0	✓	-
@LAST_RESPONSE_STATUS	VT_I4 VT_ARRAY	Error reply detailed information immediately before. (VT_I4 VT_ARRAY) [0] General status [1] Additional status (0 sets in case of not being)	✓	-

5.2.2. User variable

Table5-4CaoController class user variable list

Variable identifier	Data type	Explanation	Attribute	
			get	put
None.			-	-

6. Error code

In the FSN40NUEP1 provider, the following and peculiar the error code is defined.

About the ORiN2 commonness error, Please refer to the chapter of the error code of "ORiN2 programing 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. It bites and , cannot be executed.
0x09	0x0000	The written data is outside the range.
0x0C	0x0000	Execution of operation command failed. Check if the sensor amplifier can execute the operation command.
0x0E	0x0000	It tries to write it in attribute ID that cannot be written.
0x10	0x0000	It is a system startup. Please execute it again.
0x13	0x0000	The size of the service data is following regulations.

0x14	0x0000	Attribute ID is outside the range.
0x16	0x0000	The sensor amplifier corresponding to specified instance ID is not connected.
0x1F	0x0000	It is not possible to use it though this service is supported by specified instance ID by specified attribute ID.
0xFE	0x0000	System error.

7. Sample program

Sample code (C#) is shown as follows.

7.1. Connection and generation of object

```
// Cao objects
CaoEngine caoEng = null;
CaoWorkspace caoWs = null;
CaoController caoCtrl = null;
CaoExtension caoExt = null;
CaoVariable caoVal = null;

// Create CaoEngine
caoEng = new CaoEngine();
// Create CaoWorkspace
caoWs = caoEng.AddWorkspace("ws");
// Create CaoController and connect with a device
caoCtrl = caoWs.AddController(
    "ctrl",
    "CaoProv. KEYENCE. FSN40NUEP1",
    "",
    "conn=eth:192.168.0.21"
);
// Create CaoExtension and associate with a sensor of ch1
caoExt = caoCtrl.AddExtension("CH1");
// Create CaoVariable and associate with a command(ex. @DisplayBrightness)
caoVal = caoExt.AddVariable("@DisplayBrightness");
```

7.2. Acquisition of value that uses Variable

```
// Get value of DisplayBrightness
int DisplayBrightness = caoVal.Value;
```

7.3. Setting of value that uses Variable

```
// Put value of DisplayBrightness
caoVal.Value = 2;
```

7.4. Execution of command that uses Execute of Extension

```
// Execute initialization
caoExt.Execute("Initialize");
```

7.5. Execution of command that uses Execute of Controller

```
// Instance ID=1, Get RawValue
byte[] RawParam = new byte[5];
RawParam[0] = 0x0E; // Service code : Read parameter
RawParam[1] = 0x66; // Class ID : DL Object class
RawParam[2] = 0x01; // Instance ID : Instance(Sensor) ID = 1
RawParam[3] = 0x25; // Attribute ID(Low) : RawValue(0x0325)
RawParam[4] = 0x03; // Attribute ID(High)
object RetVal = caoCtrl.Execute("Raw", RawParam);
```
