

KEYENCE FSN40NUEP1 プロバイダ

Version 1.0.0

ユーザーズ ガイド

May 13, 2020

備考:

【改版履歴】

バージョン	日付	内容
1.0.0	2018-7-26	初版.
	2020-5-13	誤植修正 DL-EP1 → NU-EP1

【動作確認機種】

機種	バージョン	注意事項
FS-N40		通信ユニット(NU-EP1) + センサユニット(FS-N40) の構成

目次

1. はじめに	5
2. プロバイダの概要	6
2.1. インストール.....	6
2.2. 概要.....	6
3. メソッド・プロパティ	7
3.1. CaoWorkspace::AddController メソッド	7
3.1.1. Conn オプション	7
3.2. CaoControllere::Execute メソッド	8
3.3. CaoController::AddVariable メソッド	8
3.4. CaoController::AddExtension メソッド	9
3.5. CaoExtension::Execute メソッド	9
3.6. CaoExtension::AddVariable メソッド	9
3.7. CaoController::get_ExtensionNames プロパティ.....	9
3.8. CaoController::get_VariableNames プロパティ.....	9
3.9. CaoExtension::get_VariableNames プロパティ	9
3.10. CaoVariable::put_Value プロパティ	10
3.11. CaoVariable::get_Value プロパティ.....	10
4. コマンド一覧.....	11
4.1. CaoController クラス.....	11
4.1.1. CaoController::Execute(“Raw”) コマンド	11
4.2. CaoExtension クラス.....	12
4.2.1. CaoExtension::Execute(“Initialize”) コマンド	12
4.2.2. CaoExtension::Execute(“CustomSave”) コマンド.....	13
4.2.3. CaoExtension::Execute(“MaxSensitivityCalibration”) コマンド	13
4.2.4. CaoExtension::Execute(“PercentageCalibration”) コマンド.....	13
4.2.5. CaoExtension::Execute(“TwoPointCalibration1st”) コマンド	13
4.2.6. CaoExtension::Execute(“TwoPointCalibration2nd”) コマンド	13
4.2.7. CaoExtension::Execute(“PositioningCalibration1st”) コマンド.....	14
4.2.8. CaoExtension::Execute(“PositioningCalibration2nd”) コマンド	14
4.2.9. CaoExtension::Execute(“StartFullAutoCalibration”) コマンド	14
4.2.10. CaoExtension::Execute(“FinishFullAutoCalibration”) コマンド	14

4.2.11. CaoExtension::Execute(“ZeroShift”) コマンド	14
4.2.12. CaoExtension::Execute(“CancelZeroShift”) コマンド	15
4.2.13. CaoExtension::Execute(“Preset”) コマンド	15
4.2.14. CaoExtension::Execute(“CancelPreset”) コマンド	15
4.2.15. CaoExtension::Execute(“SaturationCanceling”) コマンド	15
4.2.16. CaoExtension::Execute(“CancelSaturationCanceling”) コマンド	15
4.2.17. CaoExtension::Execute(“ResetHoldValue”) コマンド	16
4.2.18. CaoExtension::Execute(“WorkPreset”) コマンド	16
4.2.19. CaoExtension::Execute(“StartFullAutoPreset”) コマンド	16
4.2.20. CaoExtension::Execute(“FinishFullAutoPreset”) コマンド	16
4.2.21. CaoExtension::Execute(“MaxSensitivityPreset”) コマンド	16
4.2.22. CaoExtension::Execute(“ResetUser”) コマンド	17
4.2.23. CaoExtension::Execute(“CancelKeyLock”) コマンド	17
4.2.24. CaoExtension::Execute(“KeyLock”) コマンド	17
4.2.25. CaoExtension::Execute(“PINKeyLock”) コマンド	17
4.2.26. CaoExtension::Execute(“CommunicationKeyLock”) コマンド	17
5. 変数一覧	19
5.1. CaoController クラス	19
5.1.1. システム変数	19
5.1.2. ユーザ変数	19
5.2. CaoExtension クラス	20
5.2.1. システム変数	20
5.2.2. ユーザ変数	31
6. エラーコード	32
7. サンプルプログラム	34
7.1. 接続とオブジェクトの生成	34
7.2. Variable を用いた値の取得	34
7.3. Variable を用いた値の設定	35
7.4. Extension の Execute を用いたコマンドの実行	35
7.5. Controller の Execute を用いたコマンドの実行	35

1. はじめに

KEYENCE FSN40NUEP1 プロバイダ(以下 FSN40NUEP1 プロバイダ)は, KEYENCE 製デジタルファイバセンサ(FS-N40)に対し, 通信ユニット(NU-EP1)を通じてアクセスを行う ORiN2 CAO プロバイダです.

本ドキュメントでは, FSN40NUEP1 プロバイダの概要と, 実装されている CAO インタフェース(関数仕様)について説明しています.

2. プロバイダの概要

2.1. インストール

FSN40NUEP1 プロバイダモジュールは、下記の DLL で構成されています。ORiN2 SDK のインストーラでインストールした場合は、インストール作業は不要です。手動でインストールする場合は、表 2-1 のように実行してください。

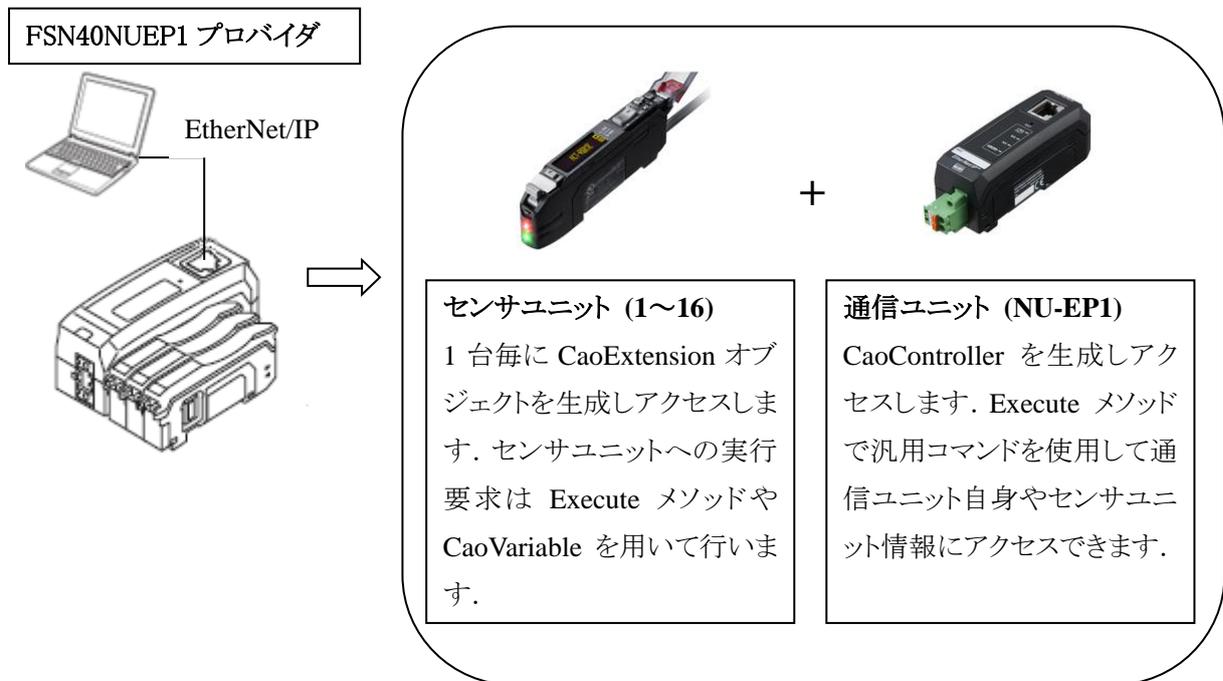
表 2-1 FSN40NUEP1 プロバイダ

ファイル名	CaoProvKEYENCE FSN40NUEP1. dll
ProgID	CaoProv.KEYENCE. FSN40NUEP1
レジストリ登録	regsvr32 CaoProvKEYENCE FSN40NUEP1. dll
レジストリ登録の抹消	regsvr32 /u CaoProvKEYENCE FSN40NUEP1. dll

2.2. 概要

FSN40NUEP1 プロバイダは、CIP プロトコルを用いて NU-EP1 へのアクセスを行います。

NU-EP1 に接続されたセンサユニット(FS-N40)毎に CaoExtension オブジェクトを生成し、Execute メソッドの実行または CaoVariable への読み書きでセンサの情報へアクセスします。



3. メソッド・プロパティ

3.1. CaoWorkspace::AddController メソッド

FSN40NUEP1 プロバイダは AddController 時に通信用の接続パラメータを参照し、通信の接続を行います。



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

bstrCtrlName : [in] コントローラ名
 bstrProvName : [in] プロバイダ名. 固定値 = "CaoProv.KEYENCE. FSN40NUEP1"
 bstrPcName : [in] プロバイダの実行マシン名
 bstrOption : [in] オプション文字列

以下にオプション文字列に指定するリストを示します。

表 3-1 CaoWorkspace::AddController のオプション文字列

オプション (1)	説明
Conn=<接続パラメータ>	必須. 通信形態と接続パラメータ. (参照 3.1.1)
MyIP[=<自 IP アドレス>]	自 IP アドレス. (複数 NIC 用途) (デフォルト: 指定なし)
ConnTimeout [=<タイムアウト時間>]	TCP 接続時のタイムアウト時間. (ミリ秒) (デフォルト: 3000)
TimeOut[=<タイムアウト時間>]	送受信時のタイムアウト時間. (ミリ秒) (デフォルト: 5000)

3.1.1. Conn オプション

通信形態と接続パラメータを指定します。

以下に Conn オプションの接続パラメータ文字列を示します。

Ethernet/IP デバイス

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

< Dest IP Address > : TCP/IP 接続先 IP アドレス.
 例: “127.0.0.1”, “192.168.0.1”

¹ 角括弧(“[]”)内は省略可能を示します。また、各パラメータの解説中の下線部はオプションを指定しなかったときのデフォルト値になります。

<Dest Port No>	:	TCP/IP 接続ポート番号. 例:44818, 5006, 5007
<Src IP Address>	:	自 IP アドレス. (複数 NIC 用途) ²⁾
<Src Port No>	:	自ポート番号. (複数 NIC 用途)

3.2. CaoController::Execute メソッド

CaoController クラスの Execute メソッドは、コマンドを実行するためのメソッドです。第 1 引数にコマンド名、第 2 引数にコマンドのパラメータを指定します。各コマンドの詳細は 4.1 章を参照してください。



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

< bstrCmd >	:	[in] コマンド名
< vntParam >	:	[in] パラメータ

3.3. CaoController::AddVariable メソッド

CaoController クラスの AddVariable メソッドは、変数オブジェクトを作成するためのメソッドです。使用できる変数一覧は 5.1 章を参照してください。



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

<bstrVariableName>	:	[in] 変数名
<bstrOption>	:	[in] オプション文字列

²⁾ Conn オプションと MyIP オプションの両方で自 IP アドレスを指定するとエラーになります。利用する場合は必ずどちらか片方で指定するようにしてください。

3.4. CaoController::AddExtension メソッド

CaoController クラスの AddExtension メソッドは、NU-EP1 に接続されたセンサユニット毎の Extension オブジェクトを作成するためのメソッドです。

書式 AddExtension(<bstrChannelNo:VT_BSTR>)

<bstrChannelNo> : [in] センサユニットの ID 番号.
“CH<ID 番号>”の書式で指定. (ID 番号:1~16)

3.5. CaoExtension::Execute メソッド

CaoExtension クラスの Execute メソッドは、コマンドを実行するためのメソッドです。第1引数にコマンド名、第2引数にパラメータを指定します。各コマンドの詳細は 4.2 章を参照してください。

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

<bstrCmd> : [in] コマンド名
<vntParam> : [in] パラメータ

3.6. CaoExtension::AddVariable メソッド

CaoExtension クラスの AddVariable メソッドは、変数オブジェクトを作成するためのメソッドです。使用できる変数一覧は 5.2 章を参照してください。

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

<bstrVariableName> : [in] 変数名
<bstrOption> : [in] オプション文字列

3.7. CaoController::get_ExtensionNames プロパティ

Extension 名リストを取得します。

3.8. CaoController::get_VariableNames プロパティ

CaoController から生成される変数名リストを取得します。

3.9. CaoExtension::get_VariableNames プロパティ

CaoExtension から生成される変数名リストを取得します。

3.10. CaoVariable::put_Value プロパティ

変数名で指定したデータを設定します。設定値は引数で指定した値を設定します。

3.11. CaoVariable::get_Value プロパティ

変数名で指定したデータを取得します。

4. コマンド一覧

4.1. CaoController クラス

表 4-1 CaoController クラス コマンド一覧

コマンド名	機能	
Raw	汎用コマンドを送出します.	P11

4.1.1. CaoController::Execute(“Raw”) コマンド

汎用コマンドを送出します.



Raw (<vntDataArray>)

<vntDataArray> : [in] コマンドデータ
(VT_UI1 | VT_ARRAY)
[0] サービスコード
[1] クラス ID
[2] インスタンス ID
[3] アトリビュート ID(Low)
[4] アトリビュート ID(High)
[n] サービスデータ

- アトリビュート ID が 8Bit の場合は High 側を 0x00 で指定します.
- 要素 5 以降はサービスデータとして扱います.
- サービスデータは Low 側から 8Bit ずつ指定します.
- サービスデータが奇数 Byte の場合は内部で NULL(0x00) パディングされます.

戻り値 : [out] 応答データ
(VT_UI1 | VT_ARRAY)

- 応答データが無いものは VT_EMPTY が返ります.
- 応答データは Low 側から 8Bit ずつ格納されます.

4.2. CaoExtension クラス

表 4-2 CaoExtension クラス コマンド一覧

コマンド名	機能	
Initialize	初期化を実行します。	P12
CustomSave	現在の設定を、センサの「カスタムセーブ領域」に書き込みます。	P13
MaxSensitivityCalibration	最大感度チューニングを実行します。	P13
PercentageCalibration	パーセントチューニングを実行します。	P13
TwoPointCalibration1st	2点チューニングの1点目を指定します。	P13
TwoPointCalibration2nd	2点チューニングの2点目を指定します。	P13
PositioningCalibration1st	位置決めチューニングの1点目を指定します。	P14
PositioningCalibration2nd	位置決めチューニングの2点目を指定します。	P14
StartFullAutoCalibration	フルオートチューニングを開始します。	P14
FinishFullAutoCalibration	フルオートチューニングを終了します。	P14
ZeroShift	ゼロシフトを実行します。	P14
CancelZeroShift	ゼロシフトを解除します。	P15
Preset	プリセットを実行します。	P15
CancelPreset	プリセットを解除します。	P15
SaturationCanceling	飽和回避を実行します。	P15
CancelSaturationCanceling	飽和回避を解除します。	P15
ResetHoldValue	ピーク値やボトム値などのホールド値をリセットします。	P16
WorkPreset	ワークプリセットを実行します。	P16
StartFullAutoPreset	フルオートプリセット(もしくは最大感度プリセット)を開始します。	P16
FinishFullAutoPreset	フルオートプリセット(もしくは最大感度プリセット)を終了します。	P16
MaxSensitivityPreset	最大感度プリセットを実行します。(開始と終了を指定せずに実行できます。)	P16
ResetUser	「カスタムセーブ領域」に保存した設定を読み出します。	P17
CancelKeyLock	キーロック解除を行います。	P17
KeyLock	キーロックを行います。	P17
PINKeyLock	暗証番号付きキーロックを行います。	P17
CommunicationKeyLock	通信キーロックを行います。	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

”) コマンド

初期化を実行します。



Initialize ()

引数 : 無し
戻り値 : 無し

4.2.2. CaoExtension::Execute(“CustomSave”) コマンド

現在の設定を、センサの「カスタムセーブ領域」に書き込みます。

 CustomSave ()

引数 : 無し
戻り値 : 無し

4.2.3. CaoExtension::Execute(“MaxSensitivityCalibration”) コマンド

最大感度チューニングを実行します。

 MaxSensitivityCalibration ()

引数 : 無し
戻り値 : 無し

4.2.4. CaoExtension::Execute(“PercentageCalibration”) コマンド

パーセントチューニングを実行します。

 PercentageCalibration ()

引数 : 無し
戻り値 : 無し

4.2.5. CaoExtension::Execute(“TwoPointCalibration1st”) コマンド

2点チューニングの1点目を指定します。

 TwoPointCalibration1st ()

引数 : 無し
戻り値 : 無し

4.2.6. CaoExtension::Execute(“TwoPointCalibration2nd”) コマンド

2点チューニングの2点目を指定します。

 TwoPointCalibration2nd ()

引数 : 無し
戻り値 : 無し

4.2.7. CaoExtension::Execute(“PositioningCalibration1st”) コマンド

位置決めチューニングの 1 点目を指定します。

 PositioningCalibration1st ()

引数 : 無し
戻り値 : 無し

4.2.8. CaoExtension::Execute(“PositioningCalibration2nd”) コマンド

位置決めチューニングの 2 点目を指定します。

 PositioningCalibration2nd ()

引数 : 無し
戻り値 : 無し

4.2.9. CaoExtension::Execute(“StartFullAutoCalibration”) コマンド

フルオートチューニングを開始します。

 StartFullAutoCalibration ()

引数 : 無し
戻り値 : 無し

4.2.10. CaoExtension::Execute(“FinishFullAutoCalibration”) コマンド

フルオートチューニングを終了します。

 FinishFullAutoCalibration ()

引数 : 無し
戻り値 : 無し

4.2.11. CaoExtension::Execute(“ZeroShift”) コマンド

ゼロシフトを実行します。

 ZeroShift ()

引数 : 無し
戻り値 : 無し

4.2.12. CaoExtension::Execute(“CancelZeroShift”) コマンド

ゼロシフトを解除します。

 CancelZeroShift ()

引数 : 無し
戻り値 : 無し

4.2.13. CaoExtension::Execute(“Preset”) コマンド

プリセットを実行します。

 Preset ()

引数 : 無し
戻り値 : 無し

4.2.14. CaoExtension::Execute(“CancelPreset”) コマンド

プリセットを解除します。

 CancelPreset ()

引数 : 無し
戻り値 : 無し

4.2.15. CaoExtension::Execute(“SaturationCanceling”) コマンド

飽和回避を実行します。

 SaturationCanceling ()

引数 : 無し
戻り値 : 無し

4.2.16. CaoExtension::Execute(“CancelSaturationCanceling”) コマンド

飽和回避を解除します。

 CancelSaturationCanceling ()

引数 : 無し
戻り値 : 無し

4.2.17. CaoExtension::Execute(“ResetHoldValue”) コマンド

ピーク値やボトム値などのホールド値をリセットします。

 ResetHoldValue ()

引数 : 無し
戻り値 : 無し

4.2.18. CaoExtension::Execute(“WorkPreset”) コマンド

ワークプリセットを実行します。

 WorkPreset ()

引数 : 無し
戻り値 : 無し

4.2.19. CaoExtension::Execute(“StartFullAutoPreset”) コマンド

フルオートプリセット(もしくは最大感度プリセット)を開始します。

 StartFullAutoPreset ()

引数 : 無し
戻り値 : 無し

4.2.20. CaoExtension::Execute(“FinishFullAutoPreset”) コマンド

フルオートプリセット(もしくは最大感度プリセット)を終了します。

 FinishFullAutoPreset ()

引数 : 無し
戻り値 : 無し

4.2.21. CaoExtension::Execute(“MaxSensitivityPreset”) コマンド

最大感度プリセットを実行します。(開始と終了を指定せずに実行できます。)

 MaxSensitivityPreset ()

引数 : 無し
戻り値 : 無し

4.2.22. CaoExtension::Execute(“ResetUser”) コマンド

「カスタムセーブ領域」に保存した設定を読み出します。

 ResetUser ()

引数 : 無し
戻り値 : 無し

4.2.23. CaoExtension::Execute(“CancelKeyLock”) コマンド

キーロック解除を行います。

 CancelKeyLock ()

引数 : 無し
戻り値 : 無し

4.2.24. CaoExtension::Execute(“KeyLock”) コマンド

キーロックを行います。

 KeyLock ()

引数 : 無し
戻り値 : 無し

4.2.25. CaoExtension::Execute(“PINKeyLock”) コマンド

暗証番号付きキーロックを行います。

 PINKeyLock ()

引数 : 無し
戻り値 : 無し

4.2.26. CaoExtension::Execute(“CommunicationKeyLock”) コマンド

通信キーロックを行います。

 CommunicationKeyLock()

引数 : 無し
戻り値 : 無し

5. 変数一覧

5.1. CaoController クラス

5.1.1. システム変数

表 5-1 CaoController クラス システム変数一覧

変数名	データ型	説明	属性	
			get	put
@MAKER_NAME	VT_BSTR	メーカー名="KEYENCE"を返す.	○	—
@VERSION	VT_BSTR	プロバイダバージョン情報を返す.	○	—
@LAST_RESPONSE_STATUS	VT_I4 VT_ARRAY	直前のエラー応答詳細情報. (VT_I4 VT_ARRAY) [0] 一般ステータス [1] 追加ステータス(無い場合は0セット)	○	—

5.1.2. ユーザ変数

表 5-2 CaoController クラス ユーザ変数一覧

変数名	データ型	説明	属性	
			get	put
無し			—	—

5.2. CaoExtension クラス

5.2.1. システム変数

表 5-3 CaoExtension クラス システム変数一覧

変数名	データ型	説明(●は初期値)	属性	
			get	put
@ErrorInfomation	VT_UI2	エラー情報. Bit0 : 過電流エラー Bit1 : システム予約 Bit2 : EEPROM エラー Bit3 : システム予約 Bit4 : システム予約 Bit5 : DATUM 補正エラー Bit6 : APC 補正エラー Bit7 : センサ間通信エラー	○	—
@WarningInfomation	VT_UI2	ワーニング情報. Bit0 : 予知保全検	○	—
@WarningFunction	VT_UI2	ワーニング機能動作状態. Bit0 : 予知保全検	○	—
@Output	VT_UI2	出力. Bit0 : 出力 1 Bit1 : 出力 2	○	—
@CurrentValue	VT_UI2	受光量現在値. 0 ~ 65535	○	—
@RawValue	VT_UI2	受光量生値. 0 ~ 65535	○	—
@PeakHoldValue	VT_UI2	ピークホールド値. 0 ~ 65535	○	—
@BottomHoldValue	VT_UI2	ボトムホールド値. 0 ~ 65535	○	—
@PeakHoldMinValue	VT_UI2	ピークホールド最小値. 0 ~ 65535	○	—
@PresetDatum	VT_UI2	プリセット/DATUM 内部設定値. 0 ~ 65535	○	—

@DisplayCustomize	VT_I2	表示値補正状態. 0 : 未使用 / DATUM 1 : プリセット 2 : ゼロシフト 3 : ワークプリセット 4 : ワークプリセット(反転)	○	—
@SlideSwitchStatus	VT_I2	スライドスイッチ状態. 0 : [SEL]/[1ch] を選択中 1 : [TERA]/[2ch] を選択中	○	—
@SaturationCancelingStatus	VT_I2	飽和回避実行状態. 0 : OFF 1 : ON	○	—
@OperationMode	VT_I2	動作モード状態. 0 : 通常動作中 1 : 投光 OFF 中 2 : スリープ中 3 : 出力固定中	○	—
@ SetOperationMode	VT_I2	動作モード設定. 0 : 通常動作 ● 1 : 投光 OFF 2 : スリープ 3 : 出力固定	○	○
@SettingValueHi	VT_UI2	設定値. (出力 1)/(エリア上限値) 0 ~ 65535(初期値 : 100)	○	○
@SettingValueLow	VT_UI2	設定値. (エリア下限値) 0 ~ 65535(初期値 : 100)	○	○
@LimitDetection	VT_I2	予知保全検出. 0 : OFF ● 1 : ON	○	○
@LimitDetectionPoint	VT_I2	予知保全設定値. 0 ~ 100(初期値 : 5)	○	○
@ResetLimitDetection	VT_I2	予知保全検出リセットモード. 0 : ユーザーリセット ● 1 : オートリセット	○	○

@CSettingValueHi	VT_UI2	[カスタム]設定値. (出力1)/(エリア上限値) 0 ~ 65535(初期値 : 100)	○	○
@CSettingValueLow	VT_UI2	[カスタム]設定値. (エリア下限値) 0 ~ 65535(初期値 : 100)	○	○
@CLimitDetection	VT_I2	[カスタム]予知保全検出. 0 : OFF ● 1 : ON	○	○
@CLimitDetectionPoint	VT_I2	[カスタム]予知保全設定値. 0 ~ 100(初期値 : 5)	○	○
@CResetLimitDetection	VT_I2	[カスタム]予知保全検出リセットモード. 0 : ユーザーリセット ● 1 : オートリセット	○	○
@KeyLockStatus	VT_I2	キーロック状態. 0 : 解除 ● 1 : キーロック 2 : 暗証番号キーロック 3 : 通信ロック	○	○
@PINCode	VT_I2	キーロック暗証番号. 0 ~ 9999(初期値 : 0)	○	○
@SlideSwitchLock	VT_I2	スライドスイッチロック. 0 : ロックしない ● 1 : ロックする	○	○
@L-onD-on	VT_UI2	L-on/D-on. (N.O./N.C.) Bit0 : 出力 1 Bit1 : 出力 2 ※1 (初期値 : 0)	○	○

@PowerModes	VT_I2	<p>パワーモード.</p> <p>0 : S-HSPD (応答時間 23us)</p> <p>1 : HSPD (応答時間 50us)</p> <p>2 : FINE (応答時間 250us) ●</p> <p>3 : TURBO (応答時間 500us)</p> <p>4 : SUPER (応答時間 1ms)</p> <p>5 : ULTRA (応答時間 4ms)</p> <p>6 : MEGA (応答時間 16ms)</p> <p>7 : TERA (応答時間 64ms)</p>	○	○
@PercentageCalibrationTarget	VT_I2	<p>パーセントチューニング目標値.</p> <p>-99 ~ 99 (初期値 : -10)</p>	○	○
@AreaPercentageTarget	VT_I2	<p>エリアパーセント目標値.</p> <p>1 ~ 99 (初期値 : 10)</p>	○	○
@Hysteresis	VT_I2	<p>ヒステリシス.</p> <p>0 : 最小</p> <p>1 : 小</p> <p>2 : 標準 ●</p> <p>3 : 大</p> <p>4 : 最大</p>	○	○
@DisplayBar	VT_I2	<p>バー表示.</p> <p>0 : OFF ●</p> <p>1 : ON</p>	○	○
@KeyLockMethod	VT_I2	<p>キーロック方式.</p> <p>0 : 標準 ●</p> <p>1 : 暗証番号付き</p>	○	○
@CKeyLockStatus	VT_I2	<p>[カスタム]キーロック状態.</p> <p>0 : 解除 ●</p> <p>1 : キーロック</p> <p>2 : 暗証番号キーロック</p> <p>3 : 通信ロック</p>	○	○
@CPINCode	VT_I2	<p>[カスタム]キーロック暗証番号.</p> <p>0 ~ 9999 (初期値 : 0)</p>	○	○
@CSlideSwitchLock	VT_I2	<p>[カスタム]スライドスイッチロック.</p> <p>0 : ロックしない ●</p> <p>1 : ロックする</p>	○	○

@CL-onD-on	VT_UI2	[カスタム]L-on/D-on. (N.O./N.C.) Bit0: 出力 1 Bit1: 出力 2 ※1 (初期値 : 0)	○	○
@CPowerModes	VT_I2	[カスタム]パワーモード. 0 : S-HSPD (応答時間 23us) 1 : HSPD (応答時間 50us) 2 : FINE (応答時間 250us) ● 3 : TURBO (応答時間 500us) 4 : SUPER (応答時間 1ms) 5 : ULTRA (応答時間 4ms) 6 : MEGA (応答時間 16ms) 7 : TERA (応答時間 64ms)	○	○
@CPercentageCalibrationTarget	VT_I2	[カスタム]パーセントチューニング目標値. -99 ~ 99 (初期値 : -10)	○	○
@CAreaPercentageTarget	VT_I2	[カスタム]エリアパーセント目標値. 1 ~ 99 (初期値 : 10)	○	○
@CHysteresis	VT_I2	[カスタム]ヒステリシス. 0 : 最小 1 : 小 2 : 標準 ● 3 : 大 4 : 最大	○	○
@CDisplayBar	VT_I2	[カスタム]バー表示. 0 : OFF ● 1 : ON	○	○
@CKeyLockMethod	VT_UI2	[カスタム]キーロック方式. 0 : 標準 ● 1 : 暗証番号付き	○	○

@OutputTimer	VT_I2	タイマ機能. (出力 1) 0: タイマオフ ● 1: オフディレイタイマ 2: オンディレイタイマ 3: ワンショットタイマ 4: オンディレイオフディレイ 5: オンディレイワンショット	○	○
@TimerDuration1	VT_I2	タイマ時間. (出力 1) 1 ~ 9999(初期値 : 10)	○	○
@TimerDuration2	VT_I2	タイマ時間. (出力 2) 1 ~ 9999(初期値 : 10)	○	○
@DetectionMode	VT_I2	検出モード. (出力 1) 0: 標準 ● 1: プリセット 2: パーセントチューニング 3: ゼロシフトチューニング 4: エリア検出 5: エリアパーセント 6: DATUM 7: 立上りエッジ検出 8: 立下りエッジ検出	○	○
@DATUMSpeed	VT_I2	DATUM 補正間隔. 0: LEVEL1:(遅い) ● 1: LEVEL2:(普通) 2: LEVEL3:(速い)	○	○
@DATUMWarningLevel	VT_I2	DATUM 警報出力レベル. 0 ~ 100(初期値 : 50)	○	○
@ExternalInput	VT_I2	外部入力設定. 0: 外部入力を使用しない ● 1: 外部チューニング 2: ゼロシフト 3: リセット 4: 投光 OFF 5: 出力固定 6: スリープモード	○	○

@PauseMode	VT_I2	出力固定種別. 0 : OFF 固定 ● 1 : ON 固定 2 : 外部入力時出力で固定	○	○
@Attenuation	VT_I2	アッテネート. 1 ~ 100(初期値 : 100)	○	○
@COutputTimer	VT_I2	[カスタム]タイマ機能. (出力 1) 0 : タイマオフ ● 1 : オフディレイタイマ 2 : オンディレイタイマ 3 : ワンショットタイマ 4 : オンディレイオフディレイ 5 : オンディレイワンショット	○	○
@CTimerDuration1	VT_I2	[カスタム]タイマ時間. (出力 1) 1 ~ 9999(初期値 : 10)	○	○
@CTimerDuration2	VT_I2	[カスタム]タイマ時間. (出力 2) 1 ~ 9999(初期値 : 10)	○	○
@CDetectionMode	VT_I2	[カスタム]検出モード. (出力 1) 0 : 標準 ● 1 : プリセット 2 : パーセントチューニング 3 : ゼロシフトチューニング 4 : エリア検出 5 : エリアパーセント 6 : DATUM 7 : 立上りエッジ検出 8 : 立下りエッジ検出	○	○
@CDATUMSpeed	VT_I2	[カスタム]DATUM 補正間隔. 0 : LEVEL1 : (遅い) ● 1 : LEVEL2 : (普通) 2 : LEVEL3 : (速い)	○	○
@CDATUMWarningLevel	VT_I2	[カスタム]DATUM 警報出力レベル. 0 ~ 100(初期値 : 50)	○	○

@CEXternalInput	VT_I2	[カスタム]外部入力設定. 0: 外部入力を使用しない ● 1: 外部チューニング 2: ゼロシフト 3: リセット 4: 投光 OFF 5: 出力固定 6: スリープモード	○	○
@CPauseMode	VT_I2	[カスタム]出力固定種別. 0: OFF 固定 ● 1: ON 固定 2: 外部入力時出力で固定	○	○
@CAttenuation	VT_I2	[カスタム]アッテネート. 1 ~ 100(初期値 : 100)	○	○
@FlipDisplay	VT_I2	表示反転. 0: 通常表示 ● 1: 反転表示	○	○
@SubDisplay	VT_I2	サブ表示. 0: OFF ● 1: 余裕度表示 2: ホールド表示 3: 余裕度ホールド 4: TAG 文字列表示	○	○
@DisplayHold	VT_I2	ホールド表示. 0: 通常表示 ● 1: ピーク最大値/ ピーク最小値 2: ボトム最大値/ ボトム最小値 3: ピーク最小値/ ボトム最大値 4: ピーク最大値/ ボトム最小値	○	○
@PresetSaturationLevel	VT_I2	プリセット飽和レベル. 100 ~ 200(初期値 : 110)	○	○
@CurrentDisplay	VT_I2	表示画面選択. 0: 通常表示 ● 1: バー表示 2: サブ表示	○	○

@DisplayBrightness	VT_I2	画面明るさ. 0: 明るさ :1 1: 明るさ :2 2: 明るさ :3 ●	○	○
@ActiveReceiverSetting	VT_I2	アクティブレシーバ設定. 0: 停止 1: 出力連動 ● 2: 出力反転 3: 常時 ON	○	○
@Language	VT_I2	言語選択. 0: 英語 ● 1: 日本語 2: 簡体字 3: ドイツ語	○	○
@CFlipDisplay	VT_I2	[カスタム]表示反転. 0: 通常表示 ● 1: 反転表示	○	○
@CSubDisplay	VT_I2	[カスタム]サブ表示. 0: OFF ● 1: 余裕度表示 2: ホールド表示 3: 余裕度ホールド 4: TAG 文字列表示	○	○
@CDisplayHold	VT_I2	[カスタム]ホールド表示. 0: 通常表示 ● 1: ピーク最大値/ ピーク最小値 2: ボトム最大値/ ボトム最小値 3: ピーク最小値/ ボトム最大値 4: ピーク最大値/ ボトム最小値	○	○
@CPresetSaturationLevel	VT_I2	[カスタム]プリセット飽和レベル. 100 ~ 200(初期値 :110)	○	○
@CCurrentDisplay	VT_I2	[カスタム]表示画面選択. 0: 通常表示 ● 1: バー表示 2: サブ表示	○	○

@CDisplayBrightness	VT_I2	[カスタム]画面明るさ. 0: 明るさ :1 1: 明るさ :2 2: 明るさ :3 ●	<input type="radio"/>	<input type="radio"/>
@CActiveReceiverSetting	VT_I2	[カスタム]アクティブレシーバ設定. 0: 停止 1: 出力連動 ● 2: 出力反転 3: 常時 ON	<input type="radio"/>	<input type="radio"/>
@CLanguage	VT_I2	[カスタム]言語選択. 0: 英語 ● 1: 日本語 2: 簡体字 3: ドイツ語	<input type="radio"/>	<input type="radio"/>
@TAGStrings	VT_BSTR	TAG 文字列. 8 文字以下(初期値 : "")	<input type="radio"/>	<input type="radio"/>
@ECO	VT_I2	エコ機能. 0: OFF ● 1: 表示消灯 2: 表示消灯+低動作速度 3: 全消灯	<input type="radio"/>	<input type="radio"/>
@APC	VT_I2	APC 機能. 0: OFF ● 1: ON	<input type="radio"/>	<input type="radio"/>
@DisplayGain	VT_I2	光量拡大設定. 0: 通常 ● 1: 4 倍	<input type="radio"/>	<input type="radio"/>
@InterferencePrevention	VT_I2	干渉防止台数. 0: 通常 ● 1: 2 倍	<input type="radio"/>	<input type="radio"/>
@EEPROMSave	VT_I2	外部入力 EEPROM 書き込み. 0: 保存しない 1: 保存する ●	<input type="radio"/>	<input type="radio"/>
@CTAGStrings	VT_BSTR	[カスタム]TAG 文字列. 8 文字以下(初期値 : "")	<input type="radio"/>	<input type="radio"/>

@CECO	VT_I2	[カスタム]エコ機能. 0 : OFF ● 1 : 表示消灯 2 : 表示消灯+低動作速度 3 : 全消灯	○	○
@CAPC	VT_I2	[カスタム]APC 機能. 0 : OFF ● 1 : ON	○	○
@CDisplayGain	VT_I2	[カスタム]光量拡大設定. 0 : 通常 ● 1 : 4 倍	○	○
@CInterferencePrevention	VT_I2	[カスタム]干渉防止台数. 0 : 通常 ● 1 : 2 倍	○	○
@CEEPROMSave	VT_I2	[カスタム]外部入力 EEPROM 書き込み. 0 : 保存しない 1 : 保存する ●	○	○
@IDNumber	VT_I2	ID 番号占有数. FS-N40/42* : 1 FS-N44* : 2	○	—
@ProductCode	VT_I2	プロダクトコード. FS-N40/N42* : 2025 FS-N44* : 2026	○	—
@Revision	VT_I2	リビジョン. 0100H ~ FFFFH	○	—
@ProductName	VT_BSTR	型式名. FS-N40/N42* : "FS-N40/N42*" FS-N44* : "FS-N44*"	○	—
@SeriesCode	VT_I2	シリーズコード. FS-N40/N42* : 2025 FS-N44* : 2026	○	—
@SeriesVersion	VT_I2	シリーズバージョン. 1	○	—

@DeviceType	VT_I2	デバイスタイプ. 0	○	—
@LAST_RESPONSE_STATUS	VT_I4 VT_ARRAY	直前のエラー応答詳細情報. (VT_I4 VT_ARRAY) [0] 一般ステータス [1] 追加ステータス(無い場合は 0 セット)	○	—

5.2.2. ユーザ変数

表 5-4 CaoController クラス ユーザ変数一覧

変数名	データ型	説明	属性	
			get	put
無し			—	—

6. エラーコード

FSN40NUEP1 プロバイダでは、以下の固有エラーコードが定義されています。

ORiN2 共通エラーについては、「[ORiN2 プログラミングガイド](#)」のエラーコードの章を参照してください。

表 6-1 固有エラーコード

エラー名	エラー番号	説明
CIP 応答異常 (フォーマットエラー)	0x80100000	CIP 通信の応答パケットが想定外の異常なフォーマットであった場合に返ります。
CIP 応答異常 (Error Status Code)	0x80100001	CIP 通信の応答パケットがステータス異常であった場合に返ります。
CIP 応答異常 (Error General Status)	0x80100002	CIP 通信の応答パケットがエラー応答であった場合に返ります。 本エラーを受信した場合は、システム変数の @LAST_RESPONSE_STATUS を読み出すことで詳細エラーコードを得ることができます。

表 6-2 詳細エラーコード

一般ステータス	追加ステータス	説明
0x00	0x0000	正常終了。
0x05	0x0000	インスタンス ID が範囲外です。
0x08	0x0000	指定したインスタンス ID ではこのサービスは非対応であるか、実行できません。
0x09	0x0000	書き込まれたデータが範囲外です。
0x0C	0x0000	動作指令の実行に失敗しました。センサアンプが、動作指令が実行できる状態か確認してください。
0x0E	0x0000	書き込みできないアトリビュート ID に書き込もうとしています。
0x10	0x0000	システム起動中です。再度実行してください。
0x13	0x0000	サービスデータのサイズが規定以下です。
0x14	0x0000	アトリビュート ID が範囲外です。
0x16	0x0000	指定されたインスタンス ID に対応するセンサアンプが接続されていません。
0x1F	0x0000	指定されたインスタンス ID でこのサービスはサポートされていますが、指定されたアトリビュート ID では使用できません。

0xFE	0x0000	システムエラー.
------	--------	----------

7. サンプルプログラム

以下にサンプルコード(C#)を示します。

7.1. 接続とオブジェクトの生成

```
// 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. Variable を用いた値の取得

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

7.3. Variable を用いた値の設定

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

7.4. Extension の Execute を用いたコマンドの実行

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

7.5. Controller の Execute を用いたコマンドの実行

```
// 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);
```
