

OMRON
ZN-CTX21 プロバイダ
ユーザーズ ガイド

Version 1.0.0

December 17, 2021

備考:

© 2018 DENSO WAVE INCORPORATED

この取扱説明書の著作権は、株式会社デンソーウェーブにあります。

本書に掲載されている会社名や製品は、一般に各社の商標または登録商標です。

この取扱説明書の一部または全部を無断で複製・転載することはお断りします。

- この説明書の内容は将来予告なしに変更することがあります。
- 本書の内容については、万全を期して作成いたしましたが、万一ご不審の点や誤り、記載もれなど、お気づきの点がありましたらご連絡ください。
- 運用した結果の影響については、上項にかかわらず責任を負いかねますのでご了承ください。

【改版履歴】

バージョン	日付	内容
1.0.0	2021-12-17	初版.

【対応機種】

機種	バージョン	注意事項
ZN-CTX21		

【動作確認機種】

機種	バージョン	注意事項
ZN-CTX21		

目次

1. はじめに.....	6
2. アプリケーション開発のための環境セットアップ.....	7
2.1. PC 開発環境のセットアップ.....	7
2.1.1. ZN-CTX21 プロバイダの手動インストール.....	7
3. コマンドリファレンス.....	8
3.1. メソッド/プロパティ一覧.....	8
3.2. メソッド・プロパティ.....	8
3.2.1. CaoWorkspace クラス.....	8
3.2.1.1. AddController メソッド.....	8
3.2.2. CaoController クラス.....	10
3.2.2.1. Index プロパティ.....	10
3.2.2.2. Name プロパティ.....	10
3.2.2.3. GetVariableNames メソッド.....	10
3.2.2.4. Variables プロパティ.....	10
3.2.2.5. AddVariable メソッド.....	10
3.2.2.6. Execute メソッド.....	11
3.2.3. CaoVariable クラス.....	13
3.2.3.1. Index プロパティ.....	13
3.2.3.2. Name プロパティ.....	13
3.2.3.3. Value プロパティ.....	13
3.3. 変数一覧.....	13
3.3.1. システム変数とユーザー変数.....	13
3.3.2. CaoController クラス変数.....	14
3.3.2.1. @CURRENT_STATUS の値の構成.....	15
3.3.2.2. @CURRENT_DATA の値の構成.....	16
4. ZN-CTX21 プロバイダによるプログラミング.....	17
4.1. ZN-CTX21 プロバイダの変数の値を収集するサンプルプログラミング.....	17
5. ZN-CTX21 プロバイダエラーコード.....	20

1. はじめに

本書は、OMRON 社の ZN-CTX21 に対して通信を行い、値を収集するプロバイダのユーザーズガイドです。図 1-1 が本プロバイダとデバイスの全体構成図になります。以降本プロバイダを ZN-CTX21 プロバイダと呼称します。

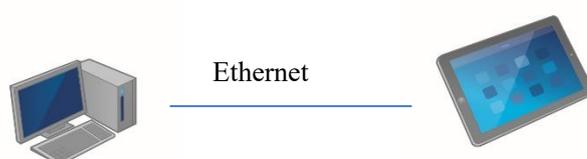


図 1-1 構成図

また、本プロバイダ及びデバイスそれぞれの対応を図 1-2 に表します。
(※一例です。全てを表しているわけではありません。)

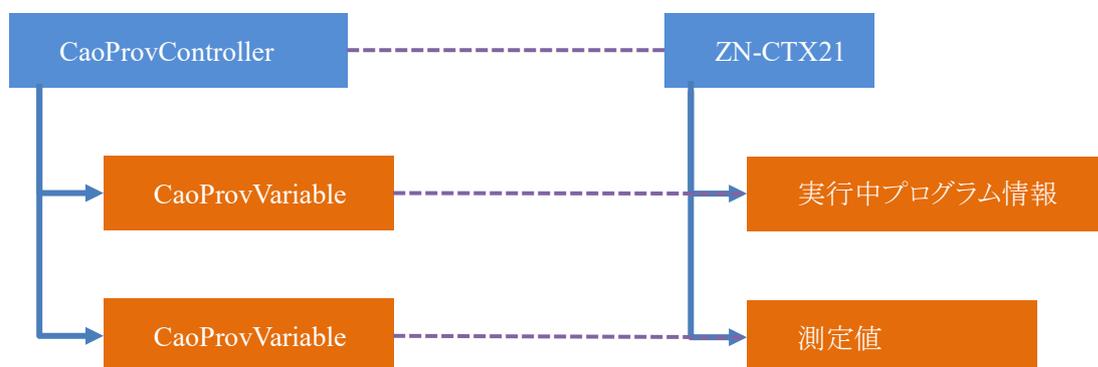


図 1-2 プロバイダの構成とデバイス情報との対応図

2. アプリケーション開発のための環境セットアップ

2.1. PC 開発環境のセットアップ

2.1.1. ZN-CTX21 プロバイダの手動インストール

ZN-CTX21 プロバイダを手動でインストールする場合は下記レジストリ登録を行う必要があります。レジストリ登録を行う場合は、管理者権限でコマンドプロンプトを起動し、regsvr32 コマンドを実行してください。実行する際には、ファイルのあるパスまで移動するか、ファイルパスを指定して実行してください。

表 2-1 ZN-CTX21 プロバイダ

ファイル名	CaoProvOMRONZNCTX21.dll
ProgID	CaoProv.OMRON.ZN-CTX21
レジストリ登録	regsvr32 CaoProvOMRONZNCTX21.dll
レジストリ登録の抹消	regsvr32 /u CaoProvOMRONZNCTX21.dll

3. コマンドリファレンス

3.1. メソッド/プロパティ一覧

表 3-1 メソッド/プロパティ一覧

カテゴリ	メソッド/プロパティ ¹	機能	参照
CaoWorkspace			
	AddController	M コントローラに接続	P.8
CaoController			
	Index	P コントローラ番号の取得	P.10
	Name	P コントローラ名の取得	P.10
	GetVariableNames	M 接続可能な変数名リストの取得	P.10
	Variables	P コントローラが保持する変数コレクションの取得	P.10
	AddVariable	M 変数オブジェクトの追加	P.10
	Execute	M 拡張コマンドの実行	P.11
CaoVariable			
	Index	P 変数番号の取得	P.13
	Name	P 変数名の取得	P.13
	Value	P 値の取得/設定	P.13

3.2. メソッド・プロパティ

3.2.1. CaoWorkspace クラス

3.2.1.1. AddController メソッド

CaoWorkspace に、コントローラオブジェクトを追加します。ZN-CTX21 プロバイダでは、AddController メソッド実行時に渡されたパラメータを参照し、該当する ZN-CTX21 と接続を行います。以下に、AddController メソッドの仕様を示します。

書式

```
AddController
(
"<コントローラ名>",           // コントローラ名(任意)
"CaoProv.OMRON.ZN-CTX21",     // プロバイダ名(固定)
"<マシン名>",                 // プロバイダ実行マシン名(未使用)
"<オプション>"                // オプション文字列
```

¹ M:メソッド, P:プロパティ, E:イベントをそれぞれ示します。

)

オプション

以下にオプション文字列に指定するオプションを示します。オプション文字列は下記に示す各オプションをカンマ(,)でつなげた文字列となります。

オプション	必須	説明	値範囲	デフォルト値
CONN	○	接続先の IP アドレスおよびポート番号を指定します。 ポート番号を省略した場合はポート番号は 2323 を指定したことになります。		
TIMEOUT	-	接続タイムアウトを指定します。	0~4294967295	500

使用例(C#)

// Engine オブジェクト

```
ORiN2.ManagedCAO.CCaoEngine engine = new ORiN2.ManagedCAO.CCaoEngine();
```

// Workspace オブジェクト

```
ORiN2.ManagedCAO.CCaoWorkspace workspace = engine.AddWorkspace("NewWrks", "");
```

// Controller オブジェクト

```
ORiN2.ManagedCAO.CCaoController controller= workspace.AddController(
    "ZN-CTX21",
    "CaoProv.OMRON.ZN-CTX21",
    "",
    "CONN=TCP:192.168.0.20");
```

3.2.1.1.1. CONN オプション

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。ここで中括弧("[]")内は省略可能なことを、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時のデフォルト値をそれぞれ示します。

```
"CONN=TCP:<IP>[:<Port>]"
```

<IP> : 接続先 IP アドレス。

<Port> : 接続先ポート。 (2323)

3.2.2. CaoController クラス

3.2.2.1. Index プロパティ

コントローラ番号を Long 型(4 バイト整数型)で取得します。この番号は, CaoWorkspace クラスの保持するコントローラコレクション内で該当のコントローラを識別するための番号を示します。

使用例(C#)

```
// Index 取得
long index = controller.Index;
```

3.2.2.2. Name プロパティ

CaoWorkspace クラスの AddController メソッドで指定されたコントローラ名を取得します。

使用例(C#)

```
System.Diagnostics.Debug.WriteLine(controller.Name);
```

3.2.2.3. GetVariableNames メソッド

接続可能な変数名リストを取得します。

オプションは使用しません。

書式

```
GetVariableNames(
    "<オプション>"           // オプション文字列(未使用)
)
```

使用例(C#)

```
// 変数名リスト取得
string[] variableNames = controller.GetVariableNames("");
```

3.2.2.4. Variables プロパティ

コントローラが保持する, 変数コレクションを取得します。

使用例(C#)

```
// 変数コレクション取得
ORiN2.ManagedCAO.CCaoVariables variables = controller.Variables;

// 変数取得
ORiN2.ManagedCAO.CCaoVariable variable = variables[0];
```

3.2.2.5. AddVariable メソッド

CaoController に変数オブジェクトを追加します。変数名には 3.3.2 に示すもののみ使用できます。

以下に, AddVariable の仕様を示します。

書式

```
AddVariable
(
"<変数名>",           // 変数名
"<オプション>"       // オプション文字列(未使用)
)
```

3.2.2.6. Execute メソッド

CaoController の拡張コマンドを実行します。以下に、Execute の仕様を示します。

書式

```
Execute
(
"<拡張コマンド名>",   // 拡張コマンド名
"<オプション文字列>" // オプション文字列(省略可能)
)
```

以下に、Execute で指定できる拡張コマンド一覧を示します。使用例は拡張コマンドの詳細で記述しています。

コマンド	説明	参照
ProviderCancel	プロバイダの動作をキャンセルします。	P.11
ProviderClear	ProviderCancel 動作をクリアします。	P.12
StartRec	測定値の記録を開始します。	P.12
StopRec	測定値の記録を停止し、測定結果を SD カードに書き出します。	P.12
ClearError	発生しているエラーを解除します。	P.12
ClearAlarm	発生しているアラームを解除します。	P.12
GetVersion	機器のバージョンを取得します。	P.13

3.2.2.6.1. ProviderCancel コマンド

プロバイダの動作をキャンセルします。

プロバイダの動作をキャンセルすると、ZN-CTX21 との接続を閉じます。

このとき、返信を待機しているコマンドはすべて失敗して戻ります。

3.2.2.6.2. ProviderCancelClear コマンド

ProviderCancel 動作をクリアします。

ZN-CTX21 プロバイダではコマンドは実装されていますが、呼び出しても特に何も行いません。

3.2.2.6.3. StartRec コマンド

ZN-CTX21 で測定値の記録を開始します。

引数及び戻り値はありません。

使用例(C#)

```
// StartRec 実行
controller.Execute("StartRec", null);
```

3.2.2.6.4. StopRec コマンド

ZN-CTX21 で測定値の記録を停止し、測定結果を SD カードに書き出します。

引数及び戻り値はありません。

StopRec コマンドは SD カードへの書き出しが完了するまで返信が戻ってきませんので、デフォルトのタイムアウト(500ms)では処理が完了せずにコマンドがタイムアウトする可能性があります。

StopRec コマンドを使用する場合はタイムアウトを長めに設定することをお勧めします。

使用例(C#)

```
// StartRec 実行
controller.Execute("StopRec", null);
```

3.2.2.6.5. ClearError コマンド

ZN-CTX21 で発生しているエラーを解除します。

引数及び戻り値はありません。

使用例(C#)

```
// StartRec 実行
controller.Execute("ClearError", null);
```

3.2.2.6.6. ClearAlarm コマンド

ZN-CTX21 で発生しているアラームを解除します。

引数及び戻り値はありません。

使用例(C#)

```
// StartRec 実行
controller.Execute("ClearAlarm", null);
```

3.2.2.6.7. GetVersion コマンド

ZN-CTX21 の機器のバージョン情報を取得します。

GetVersion で取得できる値は VT_VARIANT|VT_ARRAY で、各要素の意味は以下の通りです。

表 3-2 GetVersion 要素一覧

インデックス	型	説明
0	VT_UI1	メジャーバージョン
1	VT_UI1	マイナーバージョン
2	VT_UI1	リリース番号
3	VT_BSTR	シリアル番号
4	VT_UI1	モデル ID

3.2.3. CaoVariable クラス

3.2.3.1. Index プロパティ

変数番号を Long 型(4 バイト整数型)で取得します。この番号は CaoController クラスの保持する変数コレクションで該当の変数を識別する番号を示します。

使用例(C#)

```
// Index 取得
int index = caoVar.Index;
```

3.2.3.2. Name プロパティ

CaoContrller クラスの AddVariable メソッドで指定された変数名を取得します。

使用例(C#)

```
System.Diagnostics.Debug.WriteLine(caoVar.Name);
```

3.2.3.3. Value プロパティ

接続した ZN-CTX21 からデータを取得します。変数名によって動作が異なります。詳細は、3.3.変数一覧を参照してください。

3.3. 変数一覧

各クラスで使用可能な変数一覧を定義します。なお変数は、CaoVariable クラスのオブジェクトを指します。

3.3.1. システム変数とユーザー変数

ZN-CTX21 プロバイダではシステム変数のみが使用できます。

システム変数

その変数を保持するオブジェクト内で唯一の情報にアクセスするための変数です。システム変数はしばしば静的データである場合があります。システム変数は名前の先頭に "@" が付いています。

例)プロバイダバージョン, デバイス製造元, 現在時刻

3.3.2. CaoController クラス変数**表 3-3 CaoController クラス変数一覧**

変数名	説明	値の型
@MAKER_NAME	メーカー名	VT_BSTR
@VERSION	プロバイダのバージョン情報	VT_BSTR
@ERROR_CODE	エラーコード	VT_UI2
@STATUS	動作状態	VT_UI1
@RECORD_DATA_COUNT	記録データ数	VT_UI2
@SD_INSERT_STATUS	SD 挿入状態	VT_UI1
@ALARM_OUTPUT_STATUS	アラーム出力状態	VT_UI1
@CURRENT_STATUS	状態一覧 (表 3-4 @CURRENT_STATUS 要素一覧 参照)	VT_VARIANT VT_ARRAY
@INSTANTANEOUS_POWER	瞬時電力(kWh)	VT_R8
@ELECTRIC_ENERGY	電力量(kW)	VT_R8
@INSTANTANEOUS_CURRENT_CH1	瞬時電流 CH1(A)	VT_R8(※1)
@INSTANTANEOUS_CURRENT_CH2	瞬時電流 CH2(A)	VT_R8(※1)
@INSTANTANEOUS_CURRENT_CH3	瞬時電流 CH3(A)	VT_R8(※1)
@MEASURED_TIME	測定日時	VT_DATE
@CURRENT_DATA	最新の測定値 (表 3-5 @CURRENT_DATA 要素一覧 参照)	VT_VARIANT VT_ARRAY
@OPERATION_MODE	動作モード(設定値)	VT_UI4
@RECORD_MODE	記録モード(設定値)	VT_UI4
@RECORD_INTERVAL	記録間隔(設定値)	VT_UI4
@INTEGRATED_ELECTRIC_ENERGY_RESET_INTERVAL	積算電力量リセット間隔(設定値)	VT_UI4
@APPLICABLE_CIRCUIT	適用回路(設定値)	VT_UI4

@DEDICATED_CT_TYPE	専用 CT 種別(設定値)	VT_UI4
@NUMBER_OF_CHANNELS_USED	使用チャンネル数(設定値)	VT_UI4
@VOLTAGE_TO_BE_MEASURED	測定対象電圧(設定値)	VT_R8
@POWER_FACTOR	力率(設定値)	VT_R8
@FREQUENCY	周波数(設定値)	VT_UI4
@RATED_PRIMARY_SIDE_CURRENT_VALUE	定格一次側電流値(設定値)	VT_UI4
@LOW_CUT_CURRENT	ローカット電流(設定値)	VT_R8
@MEASUREMENT_RANGE	測定レンジ(設定値)	VT_UI4
@PRICE_PER_CO2_CONVERSION_VALUE_SETTING	料金/CO2 換算値設定(設定値)	VT_R8
@CONVERSION_VALUE_UNIT_SETTING	換算値単位設定(設定値)	VT_UI4
@INTEGRATED_ELECTRIC_ENERGY_THRESHOLD	閾値(積算電力量) (設定値)	VT_UI4
@START_TRIGGER	開始トリガ(設定値)	VT_UI4
@START_TIME	開始時刻(設定値)	VT_BSTR
@END_TRIGGER	終了トリガ(設定値)	VT_UI4
@END_TIME	終了時刻(設定値)	VT_BSTR
@ELAPSED_TIME	経過時間(設定値)	VT_UI4
@REC_RETURN_FUNCTION_AT_STARTUP	起動時 REC 復帰機能(設定値)	VT_UI4
@PRESENCE_OR_ABSENCE_OF_NETWORK_FUNCTION	ネットワーク機能の有無(設定値)	VT_UI4
@IP_ADDRESS	IP アドレス(設定値)	VT_BSTR
@SUBNET_MASK	サブネットマスク(設定値)	VT_BSTR
@DEFAULT_GATEWAY	デフォルトゲートウェイ(設定値)	VT_BSTR
@PORT_NUMBER	ポート番号(設定値)	VT_UI4

※1: チャンネルが存在しない場合は値が VT_EMPTY になります。

3.3.2.1. @CURRENT_STATUS の値の構成

@CURRENT_STATUS は ZN-CTX21 の状態を一回の通信でまとめて取得するための変数です。

@CURRENT_STATUS で取得できる値は VT_VARIANT|VT_ARRAY で、各要素の意味は以下の通りです。

表 3-4 @CURRENT_STATUS 要素一覧

インデックス	型	説明
--------	---	----

0	VT_UI2	エラーコード
1	VT_UI1	動作状態
2	VT_UI2	記録データ数
3	VT_UI1	SD 挿入状態
4	VT_UI1	アラーム出力状態

3.3.2.2. @CURRENT_DATA の値の構成

@CURRENT_STATUS は ZN-CTX21 の状態を一回の通信でまとめて取得するための変数です。

@CURRENT_STATUS で取得できる値は VT_VARIANT|VT_ARRAY で、各要素の意味は以下の通りです。

表 3-5 @CURRENT_DATA 要素一覧

インデックス	型	説明
0	VT_R8	瞬時電力(kWh)
1	VT_R8	電力量(kW)
2	VT_R8	瞬時電流 CH1(A)(※2)
3	VT_R8	瞬時電流 CH2(A) (※2)
4	VT_R8	瞬時電流 CH3(A) (※2)
5	VT_DATE	測定日時

※2: チャンネルが存在しない場合は値が VT_EMPTY になります。

4. ZN-CTX21 プロバイダによるプログラミング

ZN-CTX21 プロバイダでは、以下の手順でクライアント PC と ZN-CTX21 を接続することができます。

- CaoEngine の作成
- CaoWorkspace の作成
- CaoController の作成

ZN-CTX21 に接続した後は、CaoVariable オブジェクトを生成することで、ZN-CTX21 の情報にアクセスすることができます。

4.1. ZN-CTX21 プロバイダの変数の値を収集するサンプルプログラミング

ここでは例として ZN-CTX21 プロバイダの変数の値を収集するサンプルプログラムを示します。

Sample	Program.cs
--------	------------

```
using System;
using System.Linq;
using ORiN2.ManagedCAO;

namespace ZNCTX21Sample
{
    class Program
    {
        /// <summary>
        /// オブジェクトを表示用の文字列に変換します
        /// </summary>
        /// <param name="obj">表示するオブジェクト</param>
        /// <returns>表示用の文字列</returns>
        static string Dump(object obj)
        {
            if (obj is Array arr)
            {
                return $"[{string.Join(",", arr.Cast<object>().Select(x => Dump(x)))]";
            }
            else
            {
                return Convert.ToString(obj);
            }
        }
    }
}
```

```
}

static void Main(string[] args)
{
    try
    {
        // CaoEngine の作成
        // engine を解放すると配下のオブジェクトはすべて解放されます
        // using 節で囲った部分を抜けると engine は自動的に解放されます
        using (CCaoEngine engine = new CCaoEngine())
        {
            // コントローラを追加します
            var ctrl = engine.Workspaces[0].AddController(
                "ZNCTX21",
                "CaoProv.OMRON.ZN-CTX21",
                null,
                "CONN=TCP:192.168.0.20");

            // 定義されている変数名の一覧を取得します
            string[] varNames = ctrl.GetVariableNames(null);

            foreach (string varName in varNames)
            {
                // 変数を追加します
                var variable = ctrl.AddVariable(varName, null);
                // 変数の値を取得します
                object value = variable.Value;
                // 変数の名前と値を画面に出力します
                Console.WriteLine("[{0}] Value: {1}", variable.Name, Dump(value));
            }
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex);
    }
    // キーが押されるまでプログラムの終了を待機します
}
```

```
        Console.ReadKey();
    }
}
}
```

5. ZN-CTX21 プロバイダエラーコード

本プロバイダには、0x8011****でマスクした以下の独自エラーコードが存在します。(表 5-1 独自エラーコード表参照)

ORiN2 の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 5-1 独自エラーコード表

エラー番号	説明
0x80110001	応答フォーマットが異常
0x80111001	パラメータナンバー範囲外
0x80111002	設定値範囲外
0x80111003	設定値読み出しエラー
0x80111004	設定値書き込みエラー
0x80111005	時刻設定範囲外
0x80111006	時刻取得エラー
0x80111007	時刻設定エラー
0x8011100A	実行出来ないモード
0x8011100B	エラー解除失敗
0x80111010	センサエラー
0x80111011	ハードエラー
0x80111020	SD カード未挿入
0x80111021	SD カード書き込みエラー
0x80111022	SD カード読み込みエラー
0x80111023	SD カードアクセス不可 電池電圧低下
0x80111024	SD カードライトプロテクト
0x80111025	SD カードアクセス中
0x80111026	SD カード認識エラー
0x80111083	入力値範囲外
0x80112002	フォーマットエラー
0x80112003	CRC エラー
0x80112004	データ長エラー
0x80112002	未定義コマンド