

OMROM

ZN-PD-S シリーズ用 プロバイダ

Version 1.0.0

ユーザーズ ガイド

February 5, 2020

備考:

【改版履歴】

バージョン	日付	内容
1.0.0	2020-02-05	初版

【動作確認機器】

機種	注意事項
ZN-PD50-S	温湿度センサ無し ファームウェアバージョン:3.2.0

目次

1. はじめに.....	4
2. アプリケーション開発のための環境セットアップ.....	5
2.1. エアパーティクルセンサの通信設定.....	5
2.2. エアパーティクルセンサとクライアント PC との接続.....	7
2.3. 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.....	8
3.1.1. CaoWorkspace::AddController メソッド.....	8
3.1.1.1. Conn オプション.....	8
3.1.2. CaoController::GetVariableNames プロパティ.....	9
3.1.3. CaoController::AddVariable メソッド.....	9
3.1.4. CaoVariable::get_Value プロパティ.....	9
3.2. 変数一覧.....	10
3.2.1. CaoController クラス.....	10
3.2.1.1. プロバイダ情報関連変数.....	10
3.2.1.1.1. @MAKER_NAME.....	10
3.2.1.1.2. @VERSION.....	10
3.2.1.2. エアパーティクルセンサから取得する情報の変数.....	11
3.2.1.2.1. @DEVICE_VERSION.....	11
3.2.1.2.2. @ERROR_STATUS.....	11
3.2.1.2.3. @MODE.....	11
3.2.1.2.4. @MEASUREMENT.....	11
3.3. エラーコード.....	12
3.4. 制限事項.....	12
4. サンプルプログラム.....	13
5. 付録.....	15
5.1. プロバイダの変数とデバイスコマンドの対応.....	15

1. はじめに

本書は、OMRON 環境センサ EQUO シリーズのエアパーティクルセンサ, ZN-PD03-S, ZN-PD50-S から状態や測定値を取得するための CAO プロバイダのユーザーズガイドです。以降本書で扱う CAO プロバイダ (CaoProvOMRONZNPDS.dll) を ZN-PD-S プロバイダ, 接続対象の機器をエアパーティクルセンサと呼びます。

ZN-PD-S プロバイダは、OMRON 社から提供された「環境センサ_EQUO シリーズ_通信コマンドマニュアル D.pdf」を参考にして開発しています。プロバイダをクライアント, エアパーティクルセンサをサーバとした TCP/IP によるサーバ/クライアント通信でコマンドを送受信します。

2. アプリケーション開発のための環境セットアップ

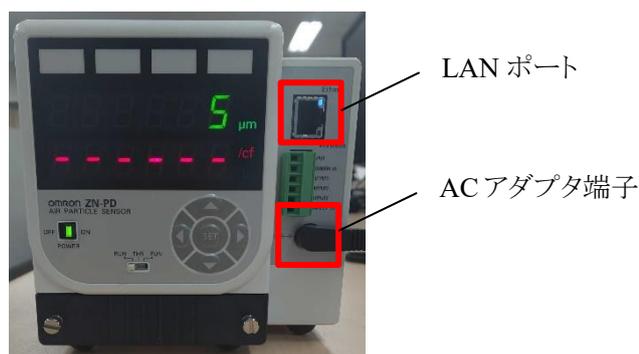
2.1. エアパーティクルセンサの通信設定

クライアント PC と接続する前に、エアパーティクルセンサの通信設定を行います。

IPアドレス[192.168.254.27]、サブネットマスク[255.255.255.0(初期値)]を設定する場合を例に、手順を示します。

- 1 エアパーティクルセンサ前面の[LAN ポート]に LAN ケーブル¹を接続します。

[AC アダプタ端子]に付属 AC アダプタを接続します。



- 2 エアパーティクルセンサ前面の[電源スイッチ]を ON にします。

[モード切替スイッチ]を FUN にします。



- 3 [サブ 7 セグ表示灯]に[EXP]が表示されるまで[操作キー]の右カーソルキーを操作します。

[メイン 7 セグ表示灯]に[TCP IP]が表示されるまで、[操作キー]の下カーソルキーを操作します。

[SET キー]を 1 回押して確定します。



※画像の状態が[EXP]です

- 4 [操作キー]の右カーソルキーを 1 回押して[IP-HI]を表示します。

[メイン 7 セグ表示灯]が 192. 168 であることを確認します。



¹ クライアント PC とハブを介して接続する場合は LAN のストレートケーブル、直接接続する場合は LAN のクロスケーブルを使用してください。

- 5 [操作キー]の右カーソルキーを 1 回押しして[IP-LOW]を表示します。
[メイン 7 セグ表示灯]を任意のアドレスになるよう[操作キー]を操作します。
上下カーソルキー: 値の変更
左右カーソルキー: 桁の移動
SET キー: 設定値確定



- 6 [操作キー]の右カーソルキーで表示を切り替えます。
[SUBHI]および[SUBLOW]の値が右の画像と同じであることを確認します。
変更が必要な場合は、カーソルキーを操作して設定してください。各カーソルキーの動作は手順 5 と同様です。

[SUBHI]



[SUBLOW]



- 7 エアパーティクルセンサの電源を入れなおします。

2.2. エアパーティクルセンサとクライアント PC との接続

エアパーティクルセンサとクライアント PC は Ethernet で通信します。図 2-1 のようにハブを介して接続するか、直接クロスケーブルで接続してください。エアパーティクルセンサの LAN ポートは機器の前面にあります。

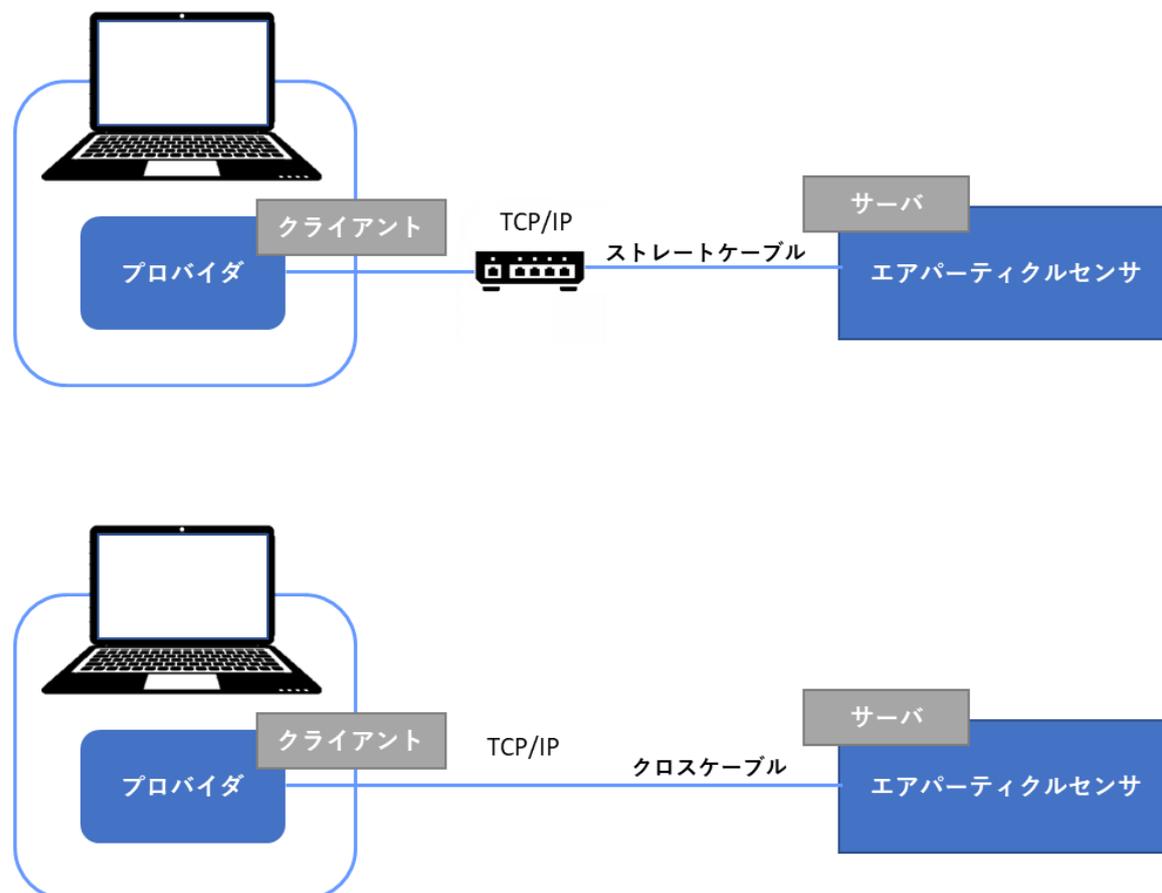


図 2-1 構成図

2.3. インストール

ZN-PD-S プロバイダのファイル形式は DLL(Dynamic Link Library)であり、CAO エンジンから使用時に動的ロードされます。使用するにあたっては ORiN2SDK をインストールするか、表 2-1 を参照して手作業でレジストリ登録を行う必要があります。

表 2-1 ZN-PD-S プロバイダ

ファイル名	CaoProvOMRONZNPDS.dll
ProgID	CaoProv.OMRON.ZNPDS
レジストリ登録	regsvr32 CaoProvOMRONZNPDS.dll
レジストリ登録の抹消	regsvr32 /u CaoProvOMRONZNPDS.dll

3. コマンドリファレンス

3.1. メソッド・プロパティ

3.1.1. CaoWorkspace::AddController メソッド

ZN-PD-S プロバイダは AddController 時に接続パラメータを参照し、通信の接続を行います。このときオプション文字列で接続先とタイムアウト時間を設定します。



AddController (<bstrCtrlName:VT_BSTR>,<bstrProvName:VT_BSTR>,
<bstrPcName:VT_BSTR>,<bstrOption:VT_BSTR>)

<bstrCtrlName> : [in] コントローラ名
 <bstrProvName> : [in] プロバイダ名. 固定値 ="CaoProv.OMRON.ZNPDS"
 <bstrPcName> : [in] プロバイダの実行マシン名 (未使用)
 <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	意味
Conn=<接続パラメータ>	必須. 通信形態とその接続パラメータを設定します. (参照:3.1.1.1)
ConnTimeout[=<タイムアウト時間>]	TCP 接続時のタイムアウト時間をミリ秒で指定します. (デフォルト:3000)
Timeout[=<タイムアウト時間>]	コマンド送受信時のタイムアウト時間をミリ秒で指定します. (デフォルト:3000)

3.1.1.1. 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。

【Ethernet】

"Conn=ETH:<Dest IP Address>"

"Conn=TCP:<Dest IP Address>"

<Dest IP Address> : エアパーティクルセンサの IP アドレス.

ポートは 2323 固定のため、指定する必要はありません。

使用例(C#)

```

CaoEngine caoEng;
CaoWorkspaces caoWss;
CaoWorkspace caoWs;
CaoControllers caoCtrls;
CaoController caoCtrl;

caoEng = new CaoEngine();
caoWss = caoEng.Workspaces;
caoWs = caoWss.Item(0);
caoCtrls = caoWs.Controllers;

// 接続
caoCtrl = caoWs.AddController("ZN-PD-S", "CaoProv. OMRON. ZNPDS", null,
                             "Conn=TCP:192.168.0.10, ConnTimeout=5000, Timeout=5000");

```

3.1.2. CaoController::GetVariableNames プロパティ

表 3-2 に示しているシステム変数名の一覧を取得します。

3.1.3. CaoController::AddVariable メソッド

エアパーティクルセンサから測定値や状態の取得を行うための変数オブジェクトを作成します。変数名は、表 3-2 に示す変数名のみ使用できます。

書式

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

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

使用例(C#)

```

// エラー状態取得用の変数を追加
CaoVariable val;
val = caoCtrl.AddVariable("@ERROR_STATUS");

```

3.1.4. CaoVariable::get_Value プロパティ

変数名に対応するコマンドを送出し、データを取得します。

使用例(C#)

```

// エラー状態取得用の変数を追加
CaoVariable val;
val = caoCtrl.AddVariable("@ERROR_STATUS");

// 現在のエラー状態を取得
var value = val.Value;

```

3.2. 変数一覧

3.2.1. CaoController クラス

表 3-2 CaoController クラス システム変数一覧

変数名	説明	属性		ページ
		get	put	
@MAKER_NAME	メーカー名を取得します。	○	-	10
@VERSION	プロバイダバージョンを取得します。	○	-	10
@DEVICE_VERSION	ファームウェアバージョンを取得します。	○	-	11
@ERROR_STATUS	現在のエラー状態を取得します。	○	-	11
@MODE	現在のモードスイッチの状態を取得します。	○	-	11
@MEASUREMENT	最新の測定値を取得します。	○	-	11

3.2.1.1. プロバイダ情報関連変数

ZN-PD-S プロバイダが保持している情報を取得する変数です。

3.2.1.1.1. @MAKER_NAME

メーカー名を取得します。

データ型	内容
VT_BSTR	OMRON

3.2.1.1.2. @VERSION

プロバイダバージョンを取得します。

データ型	内容
VT_BSTR	*,*,*,*

3.2.1.2. エアパーティクルセンサから取得する情報の変数

エアパーティクルセンサにコマンドを送信し、状態や測定値を取得する変数です。

3.2.1.2.1. @DEVICE_VERSION

U-Ver(0x8004)コマンドを送出し、ファームウェアバージョンを取得します。

データ型	内容
VT_BSTR	*.*.*

3.2.1.2.2. @ERROR_STATUS

S-Re(0x1001)コマンドを送出し、本体で発生しているエラー状態を取得します。

データ型	内容
VT_UI1	0x00 : エラーなし 0x01 : ハードエラー(E-Hard) 0x02 : メモリエラー(E-Mem)

3.2.1.2.3. @MODE

S-Rm(0x1003)コマンドを送出し、モードスイッチの状態を取得します。

データ型	内容
VT_UI1	0x00 : RUN モード 0x01 : THR モード 0x02 : FUN モード

3.2.1.2.4. @MEASUREMENT

N-Req(0x5100)コマンドを送出し、最新の測定値を取得します。

データ型	データ名	内容
VT_ARRAY VT_VARIANT		
0	VT_UI1	データサイズ 0x81 : 正常終了時 0x80 : エラー時, または未計測時
1	VT_UI1	レスポンスコード 0x00 : 正常終了 0x01 : 機器異常 0x02 : FROM 異常
2	VT_UI1	機種 ID 0x10 : ZN-PD03-S(温湿度センサ無し) 0x11 : ZN-PD50-S(温湿度センサ無し) 0x50 : ZN-PD03-S(温湿度センサ有り) 0x51 : ZN-PD50-S(温湿度センサ有り)
3	VT_UI4	計測値 1(小粒子) 0x00000000 : 測定エラー それ以外 : 測定値

4	VT_UI4	計測値 2(中粒子)	小粒子と同じ
5	VT_UI4	計測値 3(大粒子)	小粒子と同じ
6	VT_I2	温度(10 倍値)	0x7FFF : 測定エラー 0x7FFE : 温湿度センサ無し それ以外 : 測定値
7	VT_I2	湿度(10 倍値)	温度と同じ
8	VT_I2	露点温度(10 倍値)	温度と同じ

3.3. エラーコード

ZN-PD-S プロバイダは、以下の固有エラーを返します。

エラー名	エラー番号	説明
応答なし	0x80100000	エアパーティクルセンサからデータが受信できない場合に返します。
受信データ異常	0x80100001	データが欠損している、データの内容に不備がある場合などに返します。
CRC チェックエラー	0x80100002	受信データから算出した CRC コードが一致しない場合に返します。
データ変換エラー	0x80100003	受信データの型変換に失敗した場合に返します。
文法チェックエラー	0x801001XX	コマンドの応答結果として文法チェックエラーを受信した場合に返します。 エアパーティクルセンサから受信した 16 進数のエラーコードが XX に挿入されます。 例)0x02 → 0x80100102

3.4. 制限事項

エアパーティクルセンサに同時に接続できるクライアントは 4 台まで²です。既に 4 台のクライアントが接続している場合、エアパーティクルセンサは新規の接続要求に応答しません。

² ファームウェアバージョン 2.10 以降。それ以前のバージョンでは接続可能なクライアントは 1 台です。

4. サンプルプログラム(C#)

以下に、本プロバイダを使用してエアパーティクルセンサと通信を行い、測定値を取得するサンプルプログラム(C#)を示します。

Sample

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using ORiN2.ManagedCAO;

namespace Sample
{
    class Program
    {
        static void Main(string[] args)
        {
            CCaoEngine caoEng = null;

            try
            {
                // Cao関連オブジェクト生成
                caoEng = new CCaoEngine();
                CCaoWorkspaces caoWss = caoEng.Workspaces;
                CCaoWorkspace caoWs = caoWss[0];
                CCaoControllers caoCtrls = caoWs.Controllers;

                var connOpt = "Conn=TCP:192.168.0.10,ConnTimeout=5000,Timeout=5000";

                // エアパーティクルセンサと接続
                CCaoController caoCtrl = caoWs.AddController("ZNPDS_Sample",
                                                            "CaoProv.OMRON.ZNPDS",
                                                            null,
                                                            connOpt);

                // エラー状態を取得
                CCaoVariable caoVarErrorStatus;
                caoVarErrorStatus = caoCtrl.AddVariable("@ERROR_STATUS", null);
                var errStatus = caoVarErrorStatus.Value;
                var value = (errStatus != null) ? Convert.ToString(errStatus) : string.Empty;
                Console.WriteLine("@ERROR_STATUS:" + value);

                // モードスイッチの状態を取得
                CCaoVariable caoVarMode;
                caoVarMode = caoCtrl.AddVariable("@MODE", null);
                var mode = caoVarMode.Value;
                value = (mode != null) ? Convert.ToString(mode) : string.Empty;
                Console.WriteLine("@MODE:" + value);

                // 測定値を取得
                CCaoVariable caoVarMeasurement;
                caoVarMeasurement = caoCtrl.AddVariable("@MEASUREMENT", null);
                var measurement = caoVarMeasurement.Value;
                value = (measurement != null && measurement is Array)
                    ? string.Join(",", (object[])measurement)
                    : string.Empty;
                Console.WriteLine("@MEASUREMENT:" + value);
            }
        }
    }
}
```

```
        catch (Exception ex)
        {
            Console.WriteLine(ex.Message);
        }
        finally
        {
            if (caoEng != null)
            {
                caoEng.Dispose();
                caoEng = null;
            }
        }
    }
}
}
```

5. 付録

5.1. プロバイダの変数とデバイスコマンドの対応

ZN-PD-S プロバイダの CaoController クラスの変数とデバイスのコマンドの対応は以下の通りです.

表 5-1 コマンド対応一覧

変数名	コマンド名	コマンドコード
@DEVICE_VERSION	U-Ver	0x8004
@ERROR_STATUS	S-Re	0x1001
@MODE	S-Rm	0x1003
@MEASUREMENT	N-Req	0x5100