OMROM
Providers for ZN-PD-S series

Version 1.0.0

User's Guide

February 5, 2020

NOTE:

[Revision History]

Version	Date	Description
1.0.0	2020-02-05	First edition

[Operation check device]

Model	Notes
ZN-PD50-S	Without temperature/humidity sensor Firmware version: 3.2.0

Table of Contents

1. Introduction

This document is a user's guide for OMRON Environmental Sensor EQUO Series air particle sensors and CAO-providers to acquire status and measured values from ZN-PD03-S,ZN-PD50-S. The CAO providers (CaoProvOMRONZNPDS.dll) described in this manual are referred to as ZN-PD-S providers, and the devices to be connected are referred to as air particle sensors.

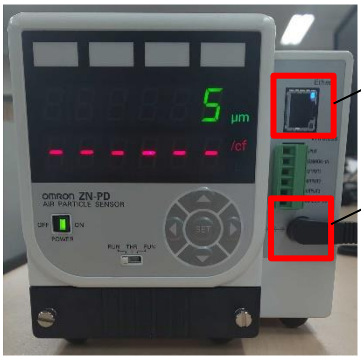
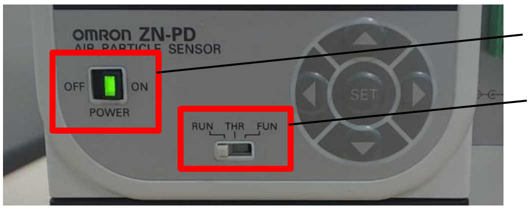
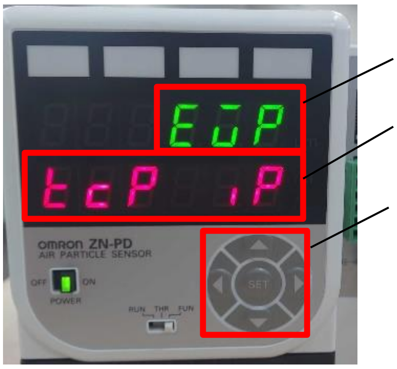
ZN-PD-S providers are developed based on the "Environmental Sensor_EQUO Series_Communication Command Manual D.pdf" provided by OMRON. Commands are sent and received in server/client communication by TCP/IP using the provider as the client and the air particle sensor as the server.

2. Setting Up Your Environment for Application Development

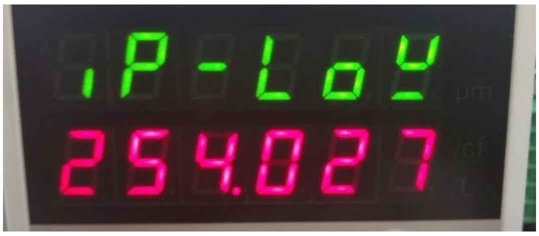
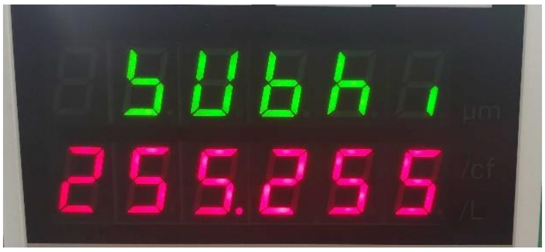
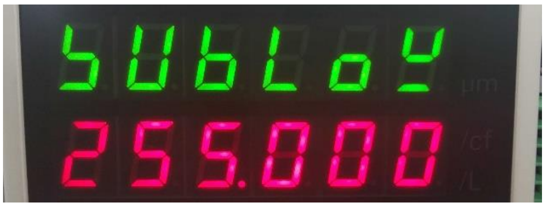
2.1. Communication setting of air particle sensor

Before connecting to the client PC, set the communication settings of the air particle sensor.

The following shows the procedure for setting the IP address [192.168.254.27] and subnet mask [255.255.255.0 (initial value)].

1	Connect the LAN cable to [LAN Port] on the front of the air particle sensor.[1] Connect the supplied AC adapter to the [AC adapter terminal].	
2	Set [Power switch] on the front of the air particle sensor to ON. Switch the mode switch to FUN.	
3	Operate the right cursor key of [Operation key] until [EXP] is displayed on [Sub 7-segment indicator]. Operate the lower cursor keys of [Control keys] until [TCP IP] is displayed on [Main 7-segment indicator]. Press [SET] once to confirm.	 ※ The image status is [EXP].
4	Press the right-cursor key of [Operation Key] once to display [IP-HI]. Check that [Main 7-segment indicator] is 192.168.	

[1] Use a straight-through LAN cable when connecting to a client PC via a hub, or a cross-over LAN cable when connecting directly.

5	Press the right-cursor key of [Operation Key] once to display [IP-LOW]. Operate the [Operation key] so that the [Main 7-segment indicator] has the desired address. Up/down cursor keys: Change value Left/Right Cursor Keys: Move digits SET key: Set value	
6	Switch the display with the right cursor key of [Operation key]. [SUBHI] And make sure that [SUBLOW] is the same as the picture on the right. If you need to change the setting, use the cursor keys to set it. The operation of each cursor key is the same as in step 5.	[SUBHI]  [SUBLOW] 
7	Power cycle the air particle sensor.	

2.2. Connecting Air Particle Sensors to Client PCs

Air Particle Sensors and Client PCs communicate on Ethernet. Connect via a hub as shown in Figure 2-1, or connect directly with a cross cable. The LAN port of the air particle sensor is located on the front of the instrument.

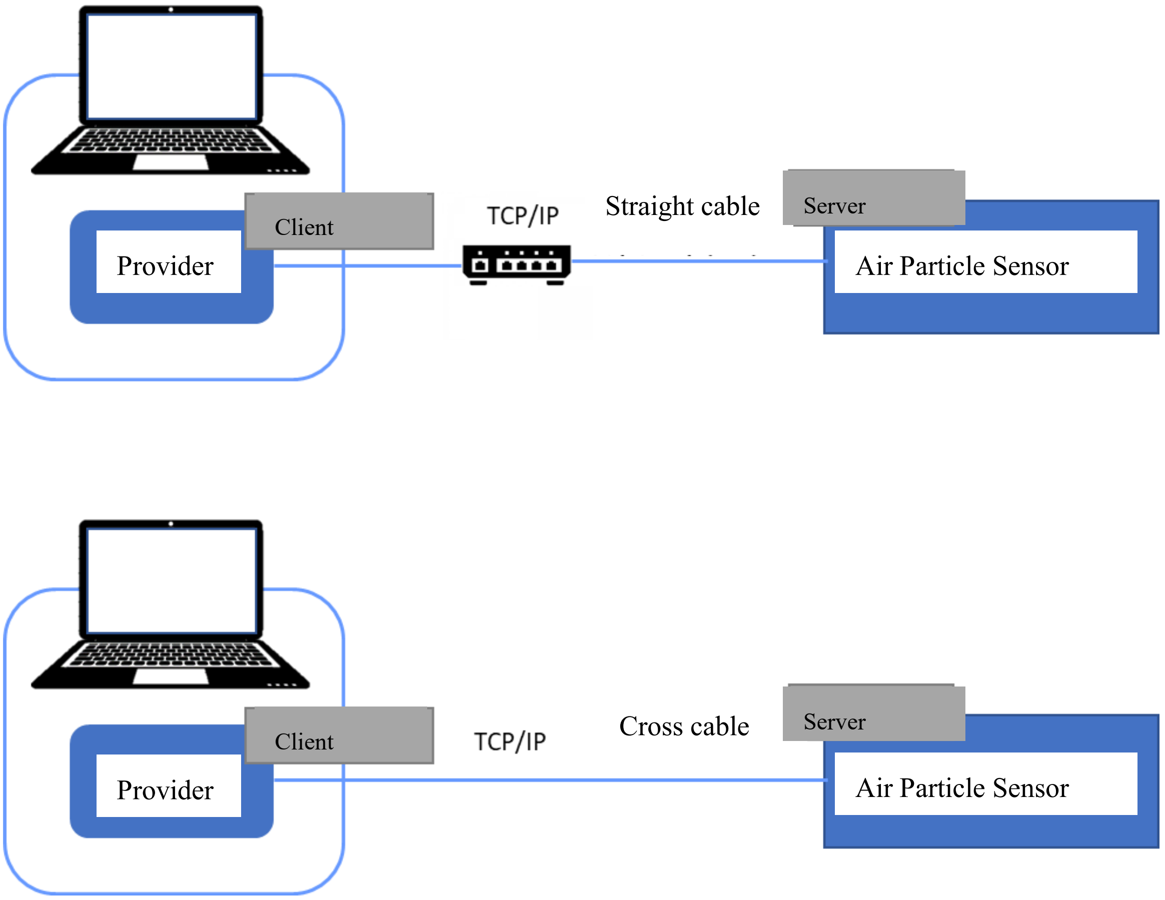


Figure 2-1 Configuration Diagram

2.3. Operating system installation

The file format for ZN-PD-S providers is DLLs (Dynamic Link Library), which are dynamically loaded when used by the CAO-engine. To use it, you must install ORiN2 SDKs or manually register the registry by referring to Table 2-1.

Table 2-1 ZN-PD-S Providers

File name	CaoProvOMRONZNPDS.dll
ProgID	CaoProv.OMRON.ZNPDS
Registry registration	Regsvr32 CaoProvOMRONZNPDS.dll
Deletion of Registry Registration	Regsvr32 /u CaoProvOMRONZNPDS.dll

3. Command Reference

3.1. Method properties

3.1.1. CaoWorkspace::AddController method

ZN-PD-S providers refer to the connection parameters during AddController and connect communications. At this time, use Option character string to set the access point and time-out period.

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

<bstrCtrlName> : [in] Controller name
<bstrProvName> : [in] Provider name. Fixed value = "CaoProv.OMRON.ZNPDS"
<bstrPcName> : [in] Provider's running machine name (unused)
<bstrOption> : [In] Option character string

The following is a listing for Option character string:

Table 3-1 Option character string in CaoWorkspace::AddController

Option	Meaning
Conn = <connection parameter>	Required. Set the communication mode and its connection parameters. (See 3.1.1.1.)3.1.1.1
ConnTimeout[=<timeout>]	Specifies the timeout in milliseconds for TCP connections. (Default: 3000)
Timeout[=<timeout>]	Specifies the timeout period in milliseconds when sending or receiving commands. (Default: 3000)

3.1.1.1. Conn Optional

The following is a Conn optional connection parameter string:

[Ethernet]
"Conn=ETH:<Dest IP Address>"
"Conn=TCP:<Dest IP Address>"
<Dest IP Address> : IP address of the air particle sensor.

The port is fixed at 2323 and does not need to be specified.

Usage example(C#)

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

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

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

3.1.2. CaoController::GetVariableNames Properties

Gets the list of system variable names shown in Table 3-2.

3.1.3. CaoController::AddVariable method

Create a variable object to acquire measured values and status from the air particle sensor. Only the variable names shown in Table 3-2 can be used as variable names.

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

<bstrVariableName>	:	[in] Variable name
<bstrOption>	:	[In] Option character string

Usage example(C#)

```
// Added a variable for acquiring the error status.
CaoVariable val;
val = caoCtrl.AddVariable("@ERROR_STATUS");
```

3.1.4. CaoVariable::get_Value Property

Raises the command corresponding to the variable name and acquires the data.

Usage example(C#)

```
// Added a variable for acquiring the error status.
CaoVariable val;
val = caoCtrl.AddVariable("@ERROR_STATUS");

// Get current error status
var value = val.Value;
```

3.2. Variable list

3.2.1. CaoController classes

Table 3-2 CaoController Class System Variable List

Variable name	Description	Attribute		Page
		Get	Put	
@MAKER_NAME	Obtain the manufacturer's name.	✓	-	10
@VERSION	Get the provider version.	✓	-	10
@DEVICE_VERSION	Gets the firmware version.	✓	-	11
@ERROR_STATUS	Gets the current error status.	✓	-	11
@MODE	Gets the current mode switch status.	✓	-	11
@MEASUREMENT	Gets the most recent measurement.	✓	-	11

3.2.1.1. Provider information-related variables

Gets the info held by ZN-PD-S providers.

3.2.1.1.1. @MAKER_NAME

Obtain the manufacturer's name.

Data Type	Description
VT_BSTR	OMRON

3.2.1.1.2. @VERSION

Get the provider version.

Data Type	Description
VT_BSTR	*,*,*,*

3.2.1.2. Variable of information obtained from the air particle sensor

This variable sends commands to the air particle sensor to obtain the status and measured values.

3.2.1.2.1. @DEVICE_VERSION

U-Ver(0x8004) Sends a command and obtains the firmware version.

Data Type	Description
VT_BSTR	*.*.*

3.2.1.2.2. @ERROR_STATUS

S-Re(0x1001) Sends a command to obtain the error status that has occurred in the main unit.

Data Type	Description
VT_UI1	0x00 : No error 0x01 : Hardware error (E-Hard) 0x02 : Memory error (E-Mem)

3.2.1.2.3. @MODE

S-Rm(0x1003) Sends a command and obtains the status of the mode switch.

Data Type	Description
VT_UI1	0x00 : RUN mode 0x01 : THR mode 0x02 : FUN mode

3.2.1.2.4. @MEASUREMENT

N-Req(0x5100) Sends a command and gets the most recent measurement.

Data Type		Data name	Description
VT_ARRAY | VT_VARIANT			
0	VT_UI1	Size of the data	0x81 : Normal termination 0x80 : Error or non-measurement
1	VT_UI1	Response code	0x00 : Normal termination 0x01 : Device error 0x02 FROM error
2	VT_UI1	Model ID	0x10 : ZN-PD03-S (without temperature/humidity sensor) 0x11 : ZN-PD50-S (without temperature/humidity sensor) 0x50 ZN-PD03-S (with temperature/humidity sensor)

			0x51 ZN-PD50-S (with temperature/humidity sensor)
3	VT_UI4	Measured value 1 (small particle)	0x00000000 : Measurement error Otherwise : Measured value
4	VT_UI4	Measured value 2 (medium particle)	Same as small particles
5	VT_UI4	Measured value 3 (large particle)	Same as small particles
6	VT_I2	Temperature (10 times value)	0x7FFF : Measurement error 0x7FFE : Without temperature/humidity sensor Otherwise : Measured value
7	VT_I2	Humidity (10 times)	Same as the temperature
8	VT_I2	Dewpoint temperature (10x value)	Same as the temperature

3.3. Error code

ZN-PD-S providers return the following unique failures:

Error name	Error Number	Description
No response	0x80100000	Returns when data cannot be received from the air particle sensor.
Receive data error	0x80100001	Returns the result when data is missing or there is an error in the data.
CRC check error	0x80100002	Returned when the CRC code calculated from the received data does not match.
Data conversion error	0x80100003	Returned when the type conversion of the received data failed.
Grammar check error	0x801001XX	Returns when a syntax check error is received as the response result of the command. The hexadecimal error code received from the air particle sensor is inserted into XX. e.g. 0x02 → 0x80100102

3.4. Restrictions

Up to four clients can be connected to the air particle sensor at the same time. If four clients are already connected, the air particle sensor will not respond to new connection requests.[2]

[2] Firmware version 2.10 or later. In earlier versions, only one client can be connected.

4. Sample program(C#)

The following shows the sample program (C#) that uses this provider to communicate with the air particle sensor and acquire measured values.

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
            {
                // Create Cao object
                caoEng = new CCaoEngine();
                CCaoWorkspaces caoWss = caoEng.Workspaces;
                CCaoWorkspace caoWs = caoWss[0];
                CCaoControllers caoCtrls = caoWs.Controllers;



                var connOpt = "Conn=TCP:192.168.0.10,ConnTimout=5000,Timout=5000";

                // Connect device
                CCaoController caoCtrl = caoWs.AddController("ZNPDS_Sample",
                                                             "CaoProv.OMRON.ZNPDS",
                                                             null,
                                                             connOpt);

                // Get error status
                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);

                // Get mode switch status
                CCaoVariable caoVarMode;
                caoVarMode = caoCtrl.AddVariable("@MODE", null);
                var mode = caoVarMode.Value;
                value = (mode != null) ? Convert.ToString(mode) : string.Empty;
                Console.WriteLine("@MODE:" + value);

                // Get measurement
                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. Appendix

5.1. Correspondence between Provider Variables and Device Commands

Variables in CaoController classes of ZN-PD-S providers correspond to commands in devices as follows:

Table 5-1 Supported commands

Variable name	Command name	Command code
@DEVICE_VERSION	U-Ver	0x8004
@ERROR_STATUS	S-Re	0x1001
@MODE	S-Rm	0x1003
@MEASUREMENT	N-Req	0x5100