PATLITE WDR-PRO provider

Version 1.2.0

User's Guide

July 28, 2022

[Remark]

[Revision History]

Version	Date	Description
1.0.0	2020-06-03	First edition
	2021-04-09	Specify the protocol used. Fixed typo.
1.1.0	2021-09-08	Supports the WDT-LR-Z2 series so that the counter value can be used.
	2021-10-22	Corrected typos in system variable names. @LIGHTS_BUZZER -> @LIGHT_BUZZER
1.2.0	2022-07-19	Added WDT-6M-Z2,WDT-5E-Z2 series to Series options. Supports WDT-6M-Z2,WDT-5E-Z2 series. It can also acquire the counter value.
	2022-07-28	Fixed the problem that "0" is missing when the MAC address of WDT starts with "0" when executing GetExtensionNames.

[Compatible devices]

Model	Version	Notes
WDR-L-Z2	1.00 ~	AC adapter attached
WDR-LE-Z2	1.00 ~	No AC adapter (for overseas use)
WDR-L-Z2-PRO	1.00 ~	Included with domestic AC adapter
WDR-LE-Z2-PRO	1.00 ~	No AC adapter for overseas use
WDR-L-Z2-PRO-L	1.00 ~	(Lite version) Included with domestic AC-Adapter
WDR-LE-Z2-PRO-L	1.00 ~	(Lite version) No AC-adapter for foreign use
WDT-6LR-Z2-PRO		
WDT-6M-Z2	2.00~	(1.00 to, 3.03 to are not released.)
WDT-5E-Z2	2.00~	(1.00 to, 3.03 to are not released.)
WDT-6LR-Z2	3.04~	(1.00 to 3.03 are not released.) ※Counter value is 3.06 to
WDT-5LR-Z2	3.04~	(1.00 to 3.03 are not released.) ※Counter value is 3.06 to
WDT-4LR-Z2	3.04~	(1.00 to 3.03 are not released.) ※Counter value is 3.06 to

[Operation Check Model]

Model	Version	Notes
WDR-L-Z2	2.04	
WDR-L-Z2-PRO	1.00	
WDR-LE-Z2	2.04	
WDT-5LR-Z2	3.06	
WDT-6M-Z2	2.02	
WDT-5E-Z2	2.02	
WDT-6LR-Z2-PRO	1.01	

Table of Contents

1. Introduction

This manual is a user's guide for WDR-PRO provider that acquire and control signaling information from PATLITE's WDR-L (E)-Z2 (hereinafter referred to as WDR), WDR-L (E)-Z2-PRO,WDR-L (E)-Z2-PRO-L (hereinafter referred to as WDR-PRO). The WDR (-PRO) is a receiver that can receive multiple signaling information wirelessly. Fig. 1-1 shows the configuration of WDR-PRO provider. Connect Ethernet cable between the PC and WDR (-PRO) and communicate wirelessly between the WDR (-PRO) and the signal light. Up to 30 signal light information can be acquired with a single WDR (-PRO). The signaling lights must be fitted with WDT-6LR-Z2,WDT-5LR-Z2,WDT-6M-Z2,WDT-5M-Z2 (WDTs) and WDT-6LR-Z2-PRO (hereafter WDT-PRO), which are data transmitters. By attaching the base unit WDB-D80S-PRO (WDB) to WDT-PRO, you can connect to the scanner, etc. via RS232C cable. The scanner can transmit scanned data, etc.

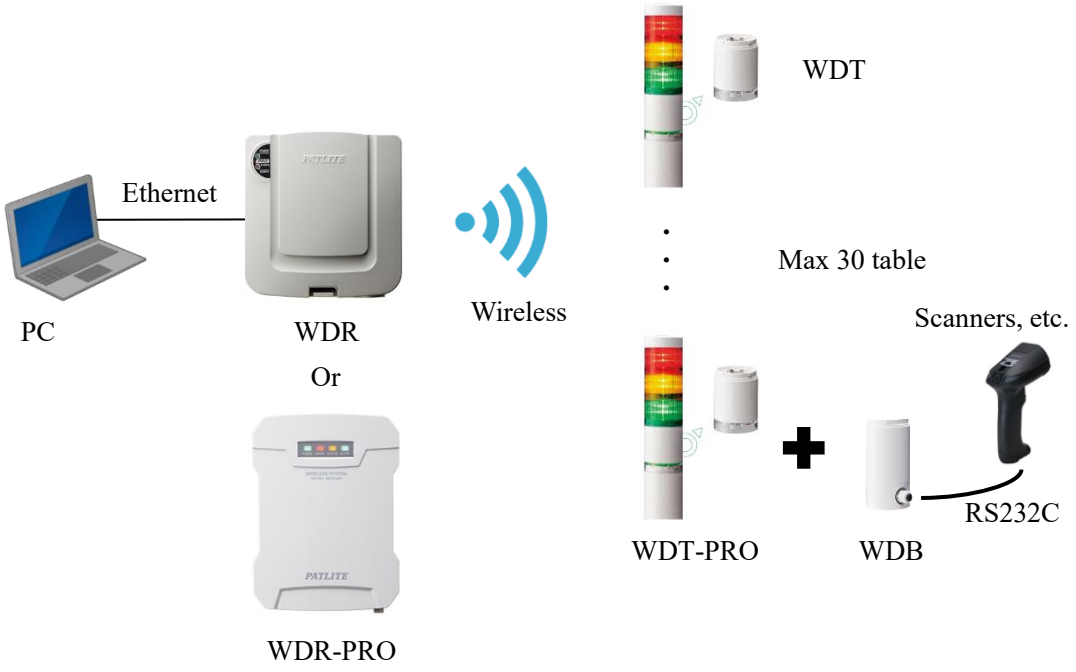


Fig. 1-1 Diagram: Configuration of WDR-PRO provider

2. Combination of Compatible Models

Table 2-1 shows the combinations of receivers and transmitters supported by WDR-PRO provider. Since the WDR-PRO provider is developed according to the WDR protocol (old protocol), it is possible to communicate with all combinations.

Table 2-1 WDR-PRO provider Correspondence Table

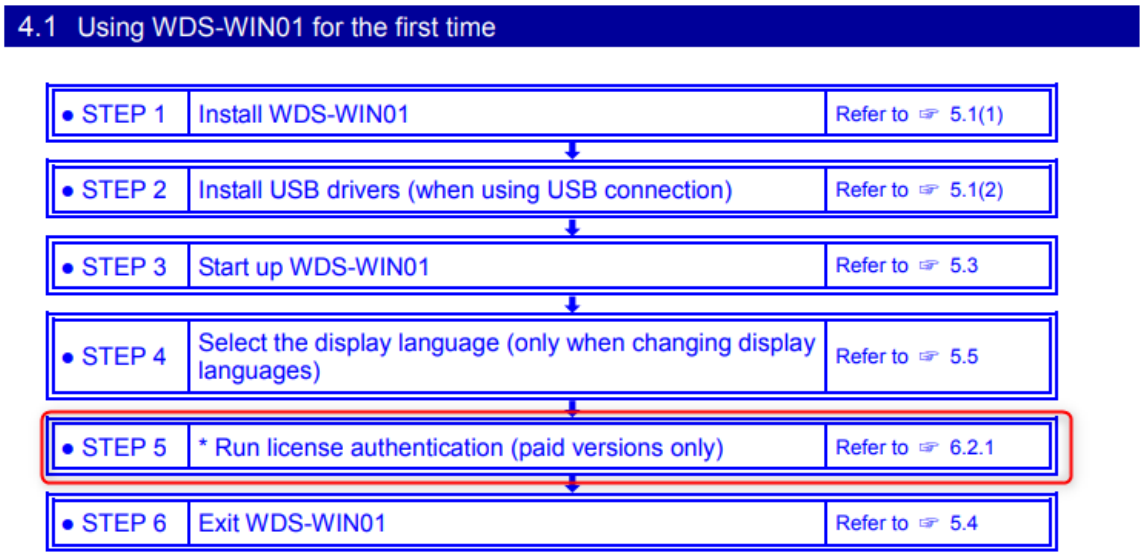
Transmitter / Receiver	WDT-6M-Z2, WDT-5E-Z2 Series	WDT-LR-Z2 Series	WDT-PRO Series
WDR-Z2 Series	○	○	○
WDR-Z2-PRO Series	○	○	○

3. Setting Up Your Environment

To use WDR-PRO provider, you must set up WDR (-PRO) and WDT (-PRO). For setup, please use the system operation software "WDS-WIN01" from PATLITE. You can use the DL WDS-WIN01 from the HP My Page of PATLITE.

For how to set up, refer to the "Wireless Data Communication System Operation Software MODEL:WDS-WIN01 Comprehensive Operation Manual". Complete the setup by following the steps of "4. Using 1 WDS-WIN01 for the first time" and "4. Configuring 2 WDT and WDRs".

If you only use WDR-PRO provider, you do not need the steps described in Section 4.1 "Enabling STEP5 Licensing (for paid versions only)".



In "4. Setting 2 WDT and WDR" and "Setting STEP5 WDT", set WDT-PRO and WDT settings as follows in order for WDR-PRO provider to obtain the correct values.

< WDT-PRO >

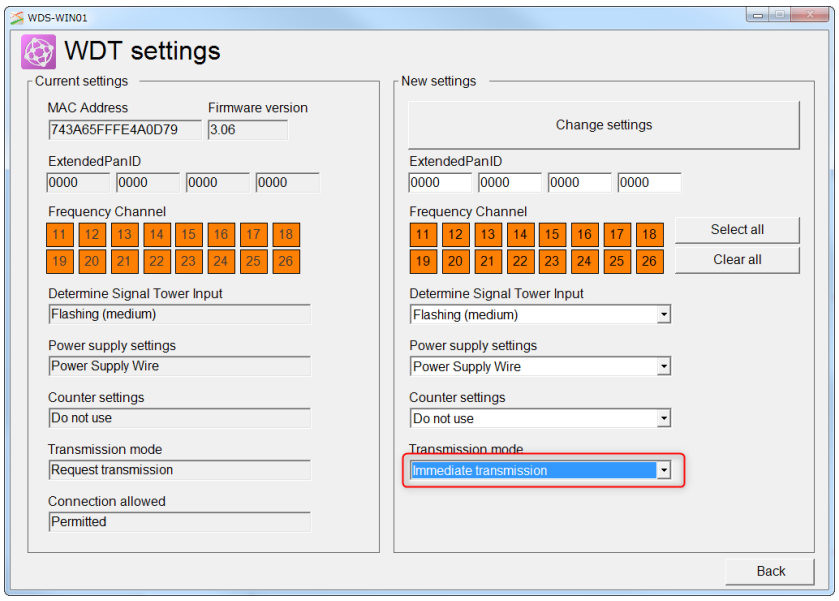
Transmit mode: Immediate transmission

Periodic transmission: None

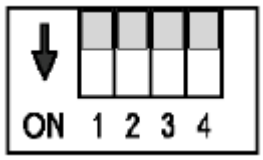
Input transmission method: WDT-PRO method

< WDT >

Transmit mode: Immediate transmission



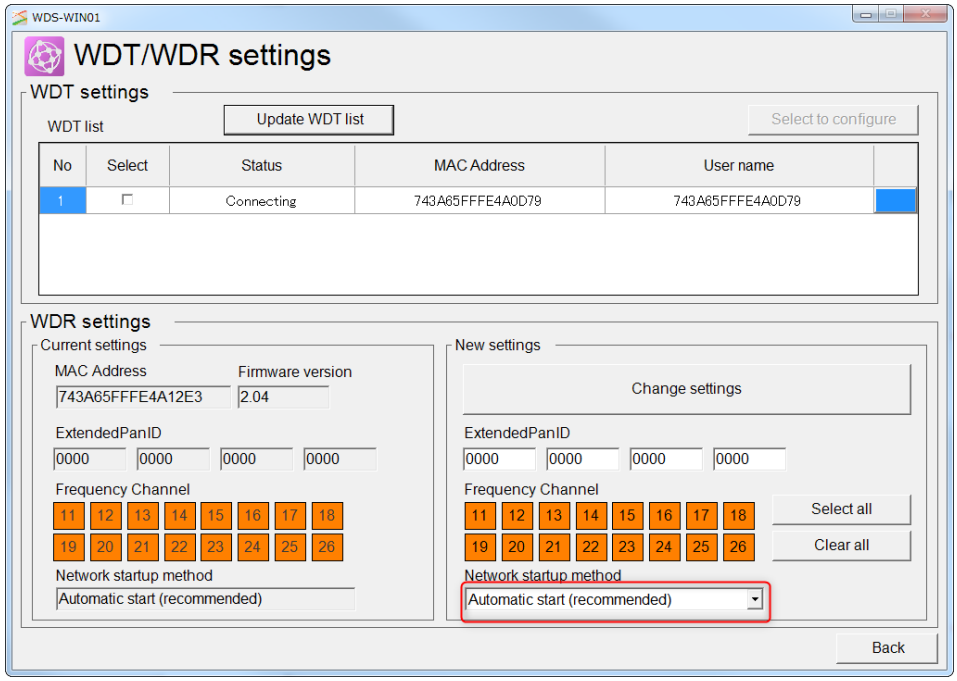
WDT-6LR-Z2,WDT-5LR-Z2 can add buzzer information in addition to signalling light information. When adding buzzer information, turn on the setting switch 3 located on the bottom of WDT-6LR-Z2,WDT-5LR-Z2 shown in Fig. 3-1.



[1]

Fig. 3-1 Setting switch

In section 4.2, "Setting STEP6 WDR", be sure to select "Automatic start (recommended)" for how to start the network. When "Manual start" is selected, WDR-PRO provider and the WDR (-PRO) cannot be connected.



[1] From PATLITE's "WDT-5LR-Z2,WDT-6LR-Z2,WDR-L(E)-Z2 General Operation Manual"

4. Provider Overview

4.1. Introduction

Fig. 4-1 shows the correspondence between each class of the provider and WDR(-PRO) and WDT(-PRO). CaoProvController corresponds to WDR(-PRO) main unit and CaoProvExtension corresponds to WDT(-PRO). CaoProvVariable supports the signal light information, external input information, and counter value held by WDT-PRO. Since the external input information cannot be retained in the case of WDT, only the signal light information and counter value information are available.

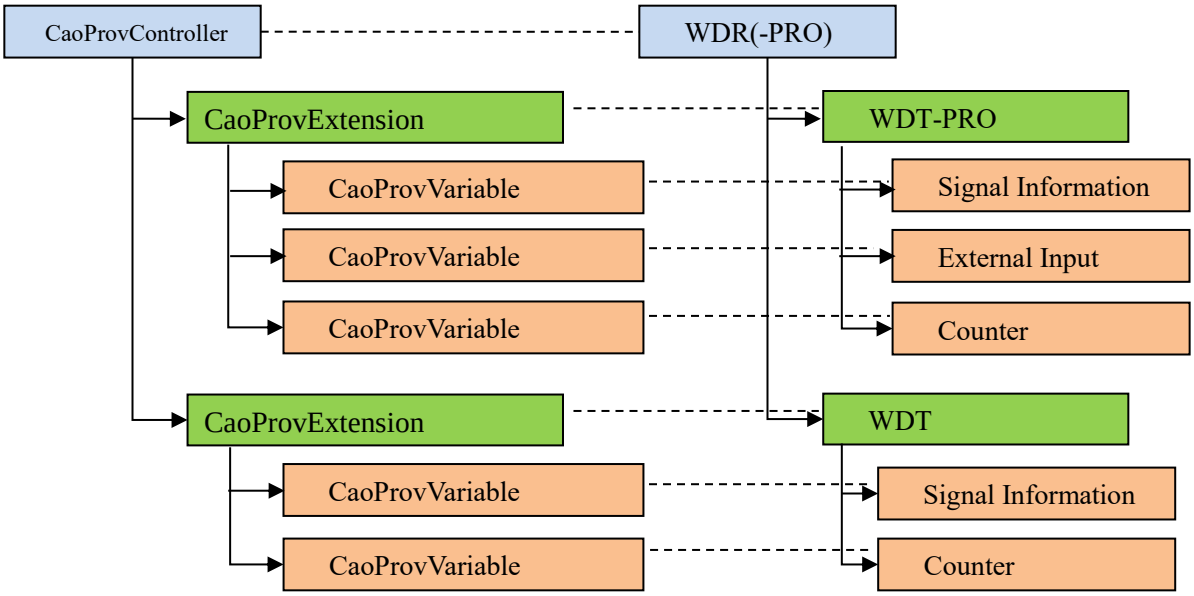


Fig. 4-1 Correspondence between each provider class and WDR (-PRO)

The file format of WDR-PRO provider is DLLs (Dynamic Link Library). Table 4-1 shows the details.

Table 4-1 Tabular WDR-PRO provider

File name	CaoProvPatliteWdr-Pro.dll
ProgID	CaoProv.PATLITE.WDR-PRO
Register registry[2]	Regsvr32 CaoProvPatliteWdr-Pro.dll
Delete Registry	Regsvr32 /u CaoProvPatliteWdr-Pro.dll

[2] If it is installed by ORiN SDK, it does not need to be registered/deleted manually.

4.1.1. State transition of WDT (-PRO)

WDT(-PRO) changes to 3 states of unconnected, connected and disconnected. WDT(-PRO) is unconnected before joining to WDR(-PRO) network. In this state, signal light information cannot be acquired from WDT(-PRO). When WDT(-PRO) joins the network, it becomes connected and the signal light information of WDT(-PRO) can be acquired (1). When the connected WDT (-PRO) and WDR (-PRO) become unable to communicate, this provider regards WDT (-PRO) as having failed, and transitions from being disconnected (2). When WDT (-PRO) returns from the abnormal state and joins the WDR (-PRO) network again, it returns to the connected state (3). The state of WDT (-PRO) is determined by the value of the "@STATUS" variable described later. Returns "-1" for the disconnected state, "0" for the connected state, and "-2" for the disconnected state.

When CaoWorkspace::AddController of this provider is executed, the network of WDR (-PRO) is reset. Therefore, immediately after executing AddController, all WDTs (-PRO) are disconnected. It may take up to 100 seconds for the WDT (-PRO) to enter the connection state from the unconnected state.

Also, due to the communication specifications, the provider value may not switch from the unconnected state to the connected state. If this happens, re-execute CaoWorkspace::AddController.

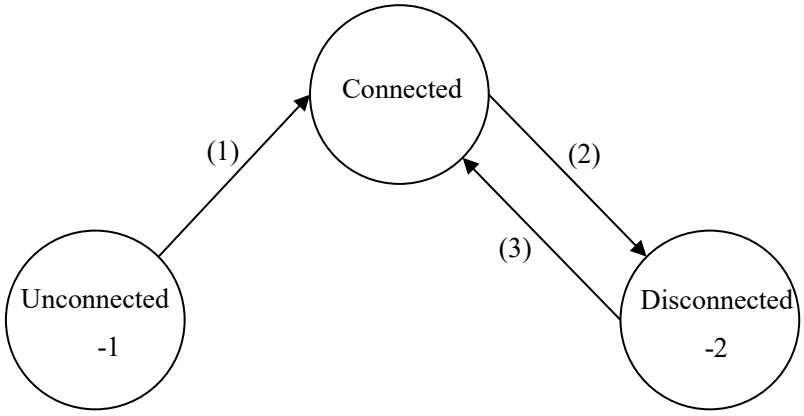


Fig. 4-2 WDT (-PRO) State Transition Diagram

4.1.2. Provider's internal behavior

Fig. 4-3 shows the internal behavior of the WDR-PRO provider. In the immediate transmission mode, WDT(-PRO) sends signal data to WDR(-PRO) at the timing when signal light information/external input information changes (1). When the reception is completed, WDR(-PRO) sends the received data to the PC and records the signal light information and the time when the PC received in the memory space in the WDR-PRO provider (2). When GetValue is executed, the WDR-PRO provider acquires and displays the signal lamp information recorded in the memory space and the reception time (3).

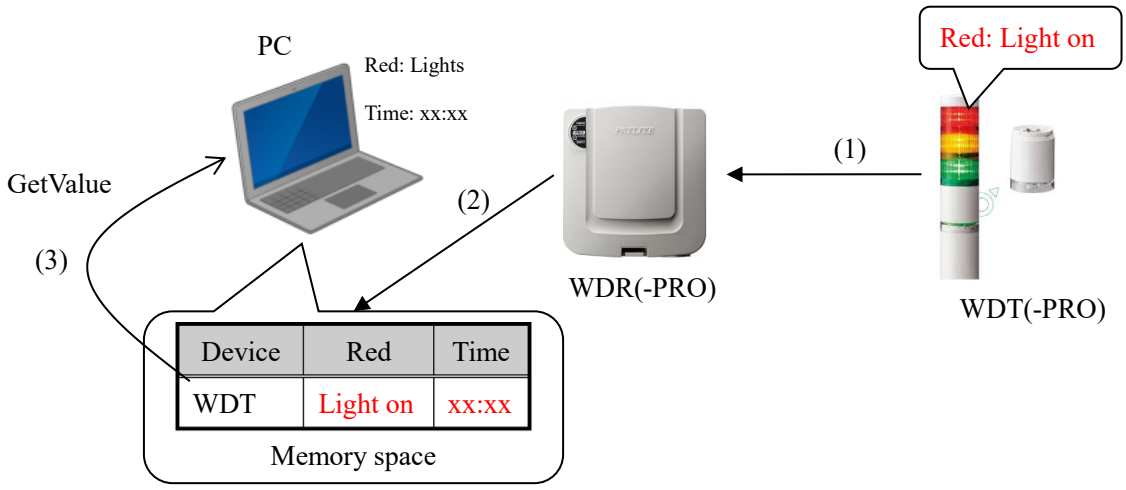


Fig. 4-3 Diagram WDR-PRO provider' Interior Behavior

4.1.3. Timing of the separation determination

In the transition transmission mode, signal information is transmitted at the timing when the signal light changes, so even if the WDT (-PRO) is disconnected from the WDR (-PRO) wireless network, it cannot be recognized. Therefore, the WDR (-PRO) periodically communicates with the WDT (-PRO) to monitor whether the WDT (-PRO) is in a normal state. Fig. 44 shows the alive check function between WDR (-PRO) and WDT (-PRO). When the communication between WDR (-PRO) and WDT (-PRO) is lost for 80 seconds (1), the WDT (-PRO) determines that it is disconnected from the network and notifies WDR-PRO provider (2). When the provider receives a detachment notification, the WDT (-PRO) determines that it is detached (3)

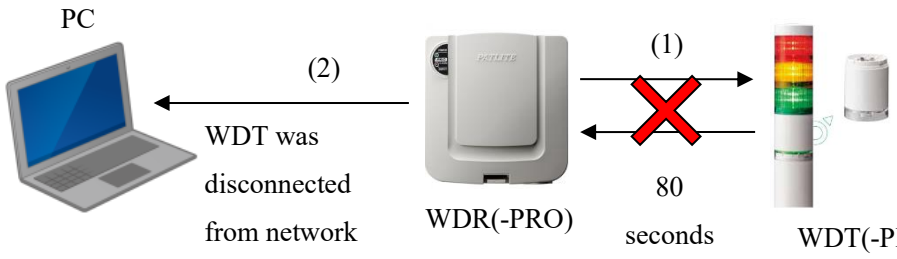


Fig. 4-4 Timing of decision on disconnection between WDR(-PRO) and WDT(-PRO)

4.1.4. Keep alive function

If you want to detect an abnormal WDT(-PRO) condition earlier than the specified disconnected decision timing (80 seconds) described in "4.1.3 Timing of disconnected decision", use WDR-PRO provider' survival-check function. Fig. 4-5 shows WDR-PRO provider alive check function. WDR-PRO provider periodically send a WDT(-PRO) and alive check request via WDR(-PRO) to monitor whether the WDT(-PRO) is in normal status (1). When WDT(-PRO) receives a request successfully, it returns a response (-PRO)(2). When the PC receives a response, it determines that the WDT (-PRO) is connected (2).

<Normal>

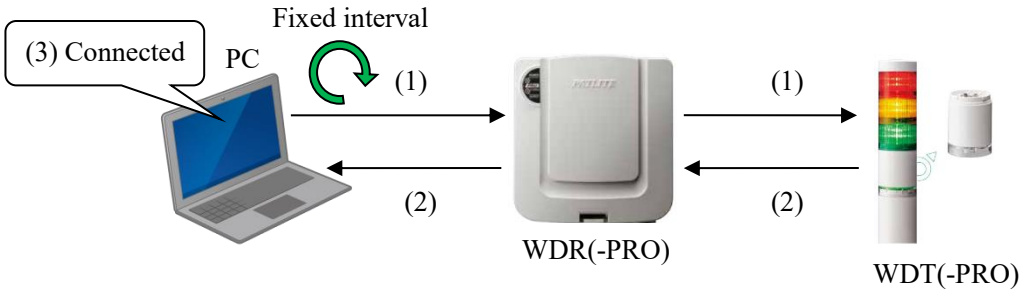


Fig. 4-5 WDR-PRO provider's keep alive function (when normal)

If there is no response for 17 seconds (WDR-(PRO) specified time) after the alive confirmation request, the request is considered to have failed. When the number of failed retransmissions reaches the maximum limit, the WDT (-PRO) regards it as an abnormal state and determines that it is disconnected from the network. The maximum number of retransmissions can be specified with Retry option during CaoController::AddExtension. In addition, you can specify the interval at which the alive check request is issued by using Interval option during CaoController::AddExtension, and this function is not executed by specifying "-1" as Interval option.

<In case of error>

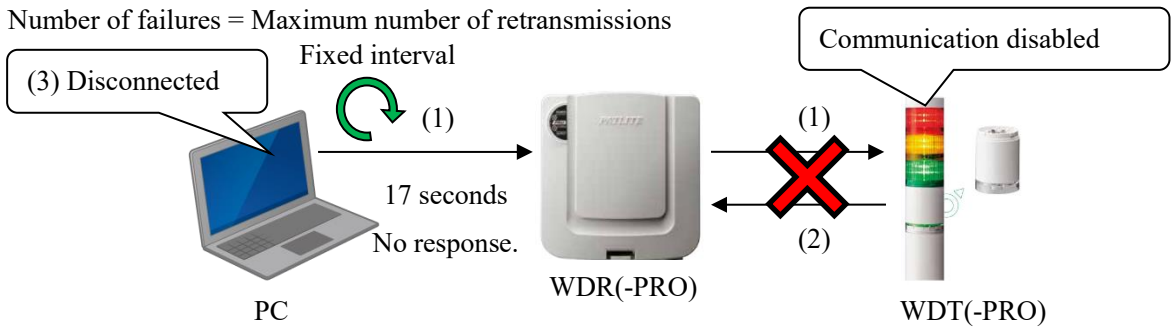


Fig. 4-6 WDR-PRO provider alive check function (in case of error)

4.2. Method properties

4.2.1. CaoWorkspace::AddController method

In this provider, the connection parameters are set in CaoWorkspace::AddController and communication(TCP) is connected. AddController specifications are shown below. Immediately after AddController is executed, all WDTs (-PRO) are disconnected.

Format

```
AddController(
        BSTR bstrCtrlName,      // [in] Controller name (enter any controller name)
        BSTR bstrProvName,      // [in] Provider name. Fixed value = " CaoProv.PATLITE.WDR-PRO "
        BSTR bstrPcName,        // [in] Provider's running machine name
        BSTR bstrOption         // [in] Option string
);
```

Table 4-2 CaoWorkspace::AddController Option character string

Setting items	Option String	Required	Description	Range	Default Value
Access point	Conn	○	Specify the connection destination in the following format. "Conn = tcp:<IP address>[:<port number>]"[3]	Port number: 1 ~ 65535	Port number: 10001
Timeout	Timeout	-	You can specify the number of seconds to time out with the connection destination in ms.	0 ~ 4294967295	3000

E.g. 1) When connecting with IP address 192.168.10.1, port number 10001, and timeout 3000 ms
"Conn = tcp:192.168.10.1"
E.g. 2) When connecting with IP address 192.168.10.1, port number 20001, and timeout 1000 ms
"Conn = tcp:192.168.10.1:20001, Timeout = 1000"

E.g. (C#)

```
using ORiN2.ManagedCAO;
CCaoEngine eng = new CCaoEngine();
CCaoWorkspace ws = eng.AddWorkspace("SampleWorkspace", "");
CCaoController ctrl = ws.AddController("controller1","CaoProv.PATLITE.WDR-PRO",
                                        string.Empty,"Conn=tcp:192.168.10.1");
```

[3] Square brackets ("[]") are optional parameters.

4.2.2. CaoController::get_VariableNames property

Retrieves a list of variables available to WDR-PRO provider. You can obtain the list in Table 4-3.

Return Value Type	Description
VT_ARRAY | VT_BSTR	You can get a list of variables that can be used for AddVariable variable names in CaoController.

Table 4-3 CaoController Class System Variable List

Variable name	Data Type	Description	Attribute	
			Get	Put
@MAKER_NAME	VT_BSTR	Returns the manufacturer named "PATLITE".	○	-
@VERSION	VT_BSTR	Returns the version of the provider.	○	-

E.g. (C#)

```
string[] variableNames = ctrl.GetVariableNames(string.Empty);
Debug.WriteLine(variableNames[0]);          // @MAKER_NAME
Debug.WriteLine(variableNames[1]);          // @VERSION
```

4.2.3. CaoController::AddVariable method

Creates a CaoVariable from a CaoController. Enter the variable name from the system variable list in Table 4-3. You do not need to enter an option string.

Format

```
AddVariable (
        BSTR bstrVarName,   // [in] Variable name (specify system variable name)
        BSTR bstrOption     // [in] Option string
 );
```

E.g. (C#)

```
CCaoVariable varMakerName = ctrl.AddVariable("@MAKER_NAME", string.Empty);
CCaoVariable varVersion = ctrl.AddVariable("@VERSION", string.Empty);
```

4.2.4. CaoController::get_ExtensionNames property

The MAC address list of the WDT (-PRO) connected or disconnected can be obtained.

Return Value Type	Description
VT_ARRAY | VT_BSTR	You can get a list of variables that can be used for AddExtension variable names in CaoController.

The Mac address of the WDT (-PRO) is described on the device label (Fig. 4-7).

Fig. 4-7 Print position of Mac address of WDT(-PRO)

E.g. (C#)

```
string[] variableNames = ctrl.GetExtensionNames(string.Empty);
Debug.WriteLine(variableNames[0]);          // e.g.,6CE4DAFFFE019CB9
Debug.WriteLine(variableNames[1]);          // e.g.,743A65FFFE4A0D79
```

4.2.5. CaoController::AddExtension

Creates a CaoExtension from a CaoController. For the variable name, specify the MAC address of the WDT (-PRO) you want to connect, and for the option string, use the option string in Table 4-4. If you want to notify WDR-PRO provider of the data read from the scanner, etc. by WDT-PRO, specify Mode,CodePage,Header,Term of the option string. For WDT, you do not need to specify it.

Format

```
AddExtension (
        BSTR bstrVarName,   // [in] Variable name (specifying the MAC address of WDT(-PRO))
        BSTR bstrOption     // [in] Option string
);
```

Table 4-4 CaoController::AddExtension Option character string

Setting items	Option String	Required	Description	Range	Default Value
Series	Series	○	Select the WDT series you want to connect to.	0: WDT-LR-Z2-PRO 1: WDT-LR-Z2 2: WDT-6M-Z2 WDT-5E-Z2	-
Send interval	Interval	-	You can specify the interval [ms] for sending the alive check function. Range: * If-1 is specified, the alive check function is disabled. If this happens, you cannot obtain the @COUNT (counting value) value described later.	-1, 500 ~ 2147483647	20000
Maximum number of retransmissions	Retry	-	You can specify how many consecutive retransmissions of the alive check function will cause the WDT (-PRO) to determine that it is in the offline state.	1 ~ 7	3
Mode	Mode	-	Optional string used by WDT-PRO that allows you to specify the output method for data sent from RS232C cable.	0: Text 1: Binary	0
Code page	CodePage	-	Option string used by WDT-PRO.You can specify the decoding method for data sent from RS232C cable.This option string is ignored when Mode option is 1:binary. A typical code page is described below. 0: Today's WindowsANSI codepage 932:Shift-JIS 1200:UTF-16 12000:UTF-32	0 ~ 65535	0

			65000:UTF-7 65001:UTF-8		
Header	Header	-	Optional string used by WDT-PRO that allows you to specify the header of the data sent from RS232C cable. Data is printed without the specified header. If Mode option is 1: binary, this option string is ignored.	0: None 1:STX	0
Terminator	Term	-	Optional string used by WDT-PRO that allows you to specify a terminator for data sent from RS232C cable. Data with the specified terminator removed is output. If Mode option is 1: binary, this option string is ignored.	0:CR 1:LF 2:CR+LF 3:ETX	0

E.g. 1) Series: WDT-LR-Z2, transmission interval of alive check function: 20000 ms, retransmission count: 3 times
"Series=1"

E.g.s 2) Series: WDT-LR-Z2-PRO, no viability check function, data-output method: text, decoding method: UTF-16, header: STX, terminator: CR+LF
"Series=0, Interval=-1, CodePage=1200, Header=1, Term=2"

E.g. 3) Series: WDT-LR-Z2-PRO, transmission intervals of viability checking function: 500 ms, number of re-sends: 7 times, data output method: binary
"Series=0, Interval=500, Retry=7, Mode=1"

E.g. (C#)

```
CCaoExtension extWdtPro = ctrl.AddExtension("6CE4DAFFFE019CB9",
                                            " Series=0, Interval=-1, CodePage=1200, Header=1, Term=2");
CCaoExtension extWdt = ctrl.AddExtension ("743A65FFFE4A0D79", " Series=1");
```

4.2.6. CaoController::OnMessage event

When WDR-PRO provider receives data from RS232C data read from a scanner, etc., it passes the data to the client as a OnMessage event of CaoController class. Note that if the source WDT (-PRO) does not exist as a CaoExtension object, the received RS232C data will be discarded.

Table 4-5 shows the data portion of OnMessage event. Message::Number is 0 and Message::Source indicates the MAC address of WDT-PRO from which the data is sent. Message::Value type is VT_BSTR when Mode option at AddExtension is in text mode and VT_ARRAY|VT_I1 when in binary mode.

Table 4-5 Data section of the table OnMessage event

Item	Data Type	Value
Number	VT_I4	0
DateTime	VT_DATE	Time the provider received the data
Destination	VT_BSTR	None
Source	VT_BSTR	MAC address of WDT(-PRO)
Value	(in text mode): VT_BSTR (in binary mode):VT_ARRAY | VT_I1	RS 232 C data
Description	VT_BSTR	None

4.2.7. CaoExtension:: get_VariableNames property

Gets a list of variables available in CaoExtension.You can get the list of Table 4-6. All system variables can be used in WDT-PRO, but @EXTERNAL_INPUTS, @EXTERNAL_INPUT1~8 cannot be used in WDT.

Return Value Type	Description
VT_ARRAY | VT_BSTR	You can get a list of variables that can be used for AddVariable variable names in CaoExtension.

Table 4-6 CaoExtension Class System Variable List

<table>
<tr><th rowspan="2">Variable name</th><th rowspan="2">Data Type</th><th rowspan="2">Description</th><th colspan="2">Attribute WDT-PRO</th><th colspan="2">Attribute WDT</th></tr>
<tr><th>Get</th><th>Put</th><th>Get</th><th>Put</th></tr>
<tr><td>@STATUS</td><td>VT_I1</td><td>Indicates the connection status of the WDT (-PRO).

<table><tr><th>Status</th><th>Value</th></tr><tr><td>Unconnected</td><td>-1</td></tr><tr><td>Connected</td><td>0</td></tr><tr><td>Disconnected</td><td>-2</td></tr></table></td><td>○</td><td>-</td><td>○</td><td>-</td></tr>
<tr><td>@LQI</td><td>VT_I2</td><td>Indicates the communication quality with the WDT (-PRO). It is expressed in the range of 0 to 255. The value is updated at Interval interval of the alive check function, the color of the signal light, and the timing when the external input changes.

<table><tr><th>Status</th><th>Value</th></tr><tr><td>Unconnected</td><td>-1</td></tr><tr><td>Connected</td><td>0~50: Bad
51~100: Slightly bad
101~150: Good
151~255: Best</td></tr><tr><td>Disconnected</td><td>-2</td></tr></table></td><td>○</td><td>-</td><td>○</td><td>-</td></tr>
<tr><td>@LIGHT_BUZZER</td><td>VT_ARRAY | VT_I1</td><td>Indicates the color of the signal light and the buzzer information. Put allows simultaneous control of the 5-color signal light and buzzer. Control from external contacts is not possible during control. To perform control using external contacts, put non-control (0).

<Signal light>
<table><tr><th>Status</th><th>Value</th></tr><tr><td>Unconnected</td><td>-1</td></tr><tr><td>Connected</td><td>0: Non-control (PUT only)
1: Not used (PUT disabled)</td></tr></table></td><td>○</td><td>○</td><td>○</td><td>-</td></tr>
</table>

<table>
<tr><td></td><td colspan="2"></td><td>
<table>
<tr><td></td><td>2: Lights off
3: Lit
4: Blinking
5: Triple Flash</td></tr>
<tr><td>Disconnected</td><td>-2</td></tr>
</table>

<Buzzer>
<table>
<tr><th>Status</th><th>Value</th></tr>
<tr><td>Unconnected</td><td>-1</td></tr>
<tr><td>Connected</td><td>0: Non-control (PUT only)
1: Non-sounding
2: Dubbing
3: Intermittent blowing</td></tr>
<tr><td>Disconnected</td><td>-2</td></tr>
</table>
</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>0</td><td>VT_I1</td><td>Indicates the red status of the signal light.</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>1</td><td>VT_I1</td><td>Indicates the yellow status of the signal light.</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>2</td><td>VT_I1</td><td>Indicates the green status of the signal light.</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>3</td><td>VT_I1</td><td>Indicates the blue status of the signal light.</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>4</td><td>VT_I1</td><td>Indicates the white status of the signal light.</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>5</td><td>VT_I1</td><td>Indicates the buzzer status.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>@RED</td><td colspan="2">VT_I1</td><td>Indicates the status of the red light. Use the Put operation to control the red light. The values for each status are the same as those for the <Signal light> of the @ LIGHT_BUZZER.</td><td>○</td><td>○</td><td>○</td><td>-</td></tr>
<tr><td>@YELLOW</td><td colspan="2">VT_I1</td><td>Indicates the status of the yellow signal light. The Put operation can control the yellow signal light. The values for each status are the same as those for the <Signal light> of the @ LIGHT_BUZZER.</td><td>○</td><td>○</td><td>○</td><td>-</td></tr>
<tr><td>@GREEN</td><td colspan="2">VT_I1</td><td>Indicates the status of the green signal light. The Put operation can control the green signal light. The values for each status are the same as those for the <Signal light> of the @ LIGHT_BUZZER.</td><td>○</td><td>○</td><td>○</td><td>-</td></tr>
<tr><td>@BLUE</td><td colspan="2">VT_I1</td><td>Indicates the status of the blue signal light. Use the Put operation to control the blue signal light. The values for each status are the same as those for the <Signal light> of the @ LIGHT_BUZZER.</td><td>○</td><td>○</td><td>○</td><td>-</td></tr>
</table>

<table>
<tr><td>@WHITE</td><td colspan="2">VT_I1</td><td>Indicates the status of the white signal light. Use the Put operation to control the white signal light. The values for each status are the same as those for the <Signal light> of the @ LIGHT_BUZZER.</td><td>○</td><td>○</td><td>○</td><td>-</td></tr>
<tr><td>@BUZZER</td><td colspan="2">VT_I1</td><td>Indicates the state of the beeper. The Put operation can be used to control the beeper. The values in each status are the same as the <buzzer> of the @LIGHT_BUZZER.</td><td>○</td><td>○</td><td>○</td><td>-</td></tr>
<tr><td rowspan="9">@EXTERNAL_INPUTS</td><td colspan="2">VT_ARRAY | VT_I1</td><td>Only WDT-PRO series can be used. Indicates external inputs 1 to 8. Cannot be controlled by Put.

<External input>
<table><tr><th>Status</th><th>Value</th></tr><tr><td>Unconnected</td><td>-1</td></tr><tr><td>Connected</td><td>0:OFF
1:ON</td></tr><tr><td>Disconnected</td><td>-2</td></tr></table></td><td>○</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>0</td><td>VT_I1</td><td>Indicates the status of external input 1.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>1</td><td>VT_I1</td><td>Indicates the status of external input 2.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>2</td><td>VT_I1</td><td>Indicates the status of external input 3.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>3</td><td>VT_I1</td><td>Indicates the status of external input 4.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>4</td><td>VT_I1</td><td>Indicates the status of external input 5.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>5</td><td>VT_I1</td><td>Indicates the status of external input 6.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>6</td><td>VT_I1</td><td>Indicates the status of external input 7.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>7</td><td>VT_I1</td><td>Indicates the status of external input 8.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>@EXTERNAL_INPUT1</td><td colspan="2">VT_I1</td><td>Indicates the status of external input 1. The values for each condition are the same as for the @EXTERNAL_INPUTS.</td><td>○</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>@EXTERNAL_INPUT2</td><td colspan="2">VT_I1</td><td>Indicates the status of external input 2. The values for each condition are the same as for the @EXTERNAL_INPUTS.</td><td>○</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>@EXTERNAL_INPUT3</td><td colspan="2">VT_I1</td><td>Indicates the status of external input 3. The values for each condition are the same as for the @EXTERNAL_INPUTS.</td><td>○</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>@EXTERNA</td><td colspan="2">VT_I1</td><td>Indicates the status of external input 4. The values for</td><td>○</td><td>-</td><td>-</td><td>-</td></tr>
</table>

<table>
<tr><td>L_INPUT4</td><td></td><td>each condition are the same as for the @EXTERNAL_INPUTS.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>@EXTERNAL_INPUT5</td><td>VT_I1</td><td>Indicates the status of external input 5. The values for each condition are the same as for the @EXTERNAL_INPUTS.</td><td>○</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>@EXTERNAL_INPUT6</td><td>VT_I1</td><td>Indicates the status of external input 6. The values for each condition are the same as for the @EXTERNAL_INPUTS.</td><td>○</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>@EXTERNAL_INPUT7</td><td>VT_I1</td><td>Indicates the status of external input 7. The values for each condition are the same as for the @EXTERNAL_INPUTS.</td><td>○</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>@EXTERNAL_INPUT8</td><td>VT_I1</td><td>Indicates the status of external input 8. The values for each condition are the same as for the @EXTERNAL_INPUTS.</td><td>○</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>@COUNT</td><td>VT_I8</td><td>Indicates the counter value. The counter value can be rewritten by the Put operation. The value is updated at the interval set by Interval option or when the Put operation is performed.

<Counter value>
<table><tr><th>Status</th><th>Value</th></tr><tr><td>Unconnected</td><td>-1</td></tr><tr><td>Connected</td><td>0~4294967295</td></tr><tr><td>Disconnected</td><td>-2</td></tr></table></td><td>○</td><td>○</td><td>○</td><td>○</td></tr>
</table>

4.2.8. CaoExtension::AddVariable method

Create a CaoVariable object from CaoExtension.

Enter the variable name from the system variable list in Table 4-6. For the option character string, use the character string shown in Table 4-7. The TimeStamp option can be used with five types of system variables. Table 4-8 shows the data types and values of system variables when the TimeStamp option is used. The availability of get/put operations is the same as in Table 4-6.

Format

```
AddVariable (
        BSTR bstrVarName,   // [in] Variable name (specify system variable name)
        BSTR bstrOption     // [in] Option string
 );
```

Table 4-7 CaoExtension::AddVariable Option character string

Setting items	Option String	Required	Description	Range	Default Value
Time stamp	TimeStamp	-	Specifies whether to include timestamp information when retrieving data. This option is available in five system variables: @STATUS, @LQI, @LIGHT_BUZZER, @EXTERNAL_INPUTS, @COUNT.	False: None True:Has a timestamp	False

e.g.1) With timestamp

"TimeStamp=true"

E.g. (C#)

```
CCaoVariable varStatus = ctrl.AddVariable("@STATUS", "");
CCaoVariable varLightBuzzer = ctrl.AddVariable("@LIGHT_BUZZER", "TimeStamp=true");
```

Table 4-8 CaoExtension class system variable (when TimeStamp option is used)

<table>
<tr><th>Variable name</th><th colspan="2">Data Type</th><th>Description</th></tr>
<tr><td rowspan="3">@STATUS</td><td colspan="2">VT_ARRAY
| VT_VARIANT</td><td>Indicates the connection status and timestamp of the WDT (-PRO). For the connection state value is the same as in Table 4-6.</td></tr>
<tr><td>0</td><td>VT_I1</td><td>Connection status.</td></tr>
<tr><td>1</td><td>VT_DATE</td><td>The timestamp of the connection state.
<table><tr><th>Status</th><th>Value</th></tr><tr><td>Unconnected</td><td>1970/1/1 00:00:00</td></tr><tr><td>Connected</td><td>Time of transition to connection status</td></tr><tr><td>Disconnected</td><td>Time of transition to the detached state</td></tr></table></td></tr>
<tr><td rowspan="3">@LQI</td><td colspan="2">VT_ARRAY
| VT_VARIANT</td><td>Indicates the communication quality and timestamp with the WDT (-PRO). The value is updated at Interval interval of the alive check function, the timing when the color of the signal light and external input change, and the timing when the counter value is put. For the communication quality value is the same as in Table 4-6.</td></tr>
<tr><td>0</td><td>VT_I2</td><td>Communication quality.</td></tr>
<tr><td>1</td><td>VT_DATE</td><td>Indicates the last time the LQI was updated.

Initial value: 1970/1/1 00:00:00</td></tr>
<tr><td rowspan="7">@LIGHT_BUZZER</td><td colspan="2">VT_ARRAY
| VT_VARIANT</td><td>Indicates signal light/buzzer information and time stamp. Put operation can control 5 colors of signal light and buzzer at the same time. Control from external contacts is not possible during control. To perform control using external contacts, put non-control (0). For the value of the signal light and buzzer

is the same as in Table 4-6.</td></tr>
<tr><td>0</td><td>VT_I1</td><td>Status of the red signal light</td></tr>
<tr><td>1</td><td>VT_I1</td><td>Status of the yellow signal light</td></tr>
<tr><td>2</td><td>VT_I1</td><td>Status of green signal light</td></tr>
<tr><td>3</td><td>VT_I1</td><td>Status of the blue signal light</td></tr>
<tr><td>4</td><td>VT_I1</td><td>Status of the white signal light</td></tr>
<tr><td>5</td><td>VT_I1</td><td>Buzzer status</td></tr>
</table>

	6	VT_DATE	Indicates the signal light and the last modified time of the buzzer. It is updated when either the signal light or the buzzer changes. Initial value: 1970/1/1 00:00:00
@EXTERNAL_INPUTS	VT_ARRAY | VT_VARIANT		Indicates the external inputs 1 to 8 and the time stamp. Only WDT-PRO series can be used. For the value of the external input is the same as in Table 4-6.
	0	VT_I1	Indicates the status of external input 1.
	1	VT_I1	Indicates the status of external input 2.
	2	VT_I1	Indicates the status of external input 3.
	3	VT_I1	Indicates the status of external input 4.
	4	VT_I1	Indicates the status of external input 5.
	5	VT_I1	Indicates the status of external input 6.
	6	VT_I1	Indicates the status of external input 7.
	7	VT_I1	Indicates the status of external input 8.
	8	VT_DATE	Indicates the last updated time of external input 1 to 8. It is updated when any of the external inputs changes. Initial value: 1970/1/1 00:00:00
@COUNT	VT_ARRAY | VT_VARIANT		Indicates the counter value. The counter value can be rewritten by Put operation. The value is updated at interval of the alive check function or at the timing when the counter value is put. For the counter value is the same as in Table 4-6.
	0	VT_I8	Indicates the counter value.
	1	VT_DATE	Indicates the last update time of the counter. Initial value: 1970/1/1 00:00:00

4.2.9. CaoVariable::get_Value Property

Get the status of the variable created by the AddVariable method. The return type depends on the variable name specified in AddVariable. The get_Value property can be used for the system variables in Table 4-3, Table 4-6, and Table 4-8 that are marked with "○". The return value depends on each system variable.

E.g. (C#)

```
Debug.WriteLine("@STATUS=" + varStatus.Value);      // @STATUS=0 (In connected)
object[] status = (object[])varLightBuzzer.Value;
Debug.WriteLine("RED=" + status[0]);                // RED=2(LightOff)
Debug.WriteLine("YELLOW=" +status[1]);              // YELLOW=3(LightOn)
Debug.WriteLine("GREEN=" +status[2]);               // GREEN=4(Flash)
Debug.WriteLine("BLUE=" +status[3]);                // BLUE=2(LightOff)
Debug.WriteLine("WHITE=" +status[4]);               // WHITE=2(LightOff)
Debug.WriteLine("BUZZER=" +status[5]);              // BUZZER=1(Non-buzzing)
Debug.WriteLine("TimeStamp=" +status[6]);           // TimeStamp=2020/01/01 00:00:00
```

4.2.10. CaoVariable::put_Value Property

Set the value to the variable created by AddVariable method. The put_Value property can be used for the system variables in Table 4-6 and Table 4-8 that are marked with "○". The argument of put_Value depends on each system variable.

For a system variable with TimeStamp option enabled, put_Value without timestamp information, as shown in the following use case.

E.g. (C#)

```
// Red, triple flush; yellow, flash; green, lights on; blue, lights off; white, uncontrolled
// Buzzer: uncontrolled
object[] newStatus = new object[]{5,4,3,2,0,0};
varLightBuzzer.Value = newStatus;
```

4.3. Error code

This provider defines a unique error code. Unique error codes are shown in Table 4-9. For ORiN2 common error, refer to the Error Codes section of ORiN2 Programming Guide.

Table 4-9 Unique error codes

Error Number	Description
0x80100001	The option string Conn is incorrectly specified. Specify the IP address in the following format: Conn = tcp:<IP address>[:<port number>]. Specify an IP address in the range of 0.0.0.0 to 255.255.255.255, and specify a port number in the range of 1 to 65535.
0x80100002	The optional character string Timeout specification is invalid. Specify a value in the range of 0 to 4294967295 ms.
0x80100003	The optional character string Series specification is invalid. Specify 0 or 1.
0x80100004	The optional character string Interval specification is invalid. Specify a value in the range of-1500 to 4294967295 ms.
0x80100005	The optional character string Retry specification is invalid. Specify a value in the range of 1 to 7 times.
0x80100006	The optional character string Mode specification is invalid. Specify 0 or 1.
0x80100007	The optional character string Header specification is invalid. Specify 0 or 1.
0x80100008	The optional character string Term specification is invalid. Specify a value in the range of 0 to 3.
0x80100009	The option-string TimeStamp is invalid. Specify the value of false or true.
0x8010000A	The variable-name of AddExtension is invalid. Enter the correct MAC address in the variable name.
0x8010000B	WDR (-PRO) failed to start. Check the IP address or port number of the connection destination again.
0x8010000C	WDR (-PRO) network start failed. Restart WDR (-PRO).
0x8010000D	WDT (-PRO) rejoin request failed. Restart WDR (-PRO).
0x8010000E	The Put value of the signal light/buzzer is incorrect. Specify the signal lamp in the range of 0,2 to 5 and the buzzer in the range of 0 to 3.
0x8010000F	The alive check function is in progress. Please wait for some time before executing the Put operation.
0x80100010	Put was executed while the WDT (-PRO) was not connected. Before executing Put, make a state transition during connection.
0x80100011	Put was executed when the WDT (-PRO) was detached. Before executing Put,

	make a state transition during connection.
0x80100012	17 seconds have not elapsed since the last send command failed. Please wait for some time before executing Put.
0x80100013	A timeout occurred during communication between WDR (-PRO) and WDT (-PRO) during Put operation. Check the connection status of WDT (-PRO) and execute Put operation at a stable distance.
0x80110010	[WDT-PRO only] A command error (no acquisition information) was notified. There is no applicable acquisition information for the request. If necessary, re-acquire or check that the requested information is correct. If that still happens, please contact us.
0x80110080	[WDT-PRO only] A command error was reported. The request command is incorrect. This is an error that the provider does not expect, so if it occurs, please contact us.
0x80110081	[WDT-PRO only] A command error (mode error) was notified. The request command mode is incorrect. This is an error that the provider does not expect, so if it occurs, please contact us.
0x80110082	[WDT-PRO only] A command error (data error) was reported. The data specified by the request command is out of the setting range or invalid. Set the correct value and make a request.
0x80110083	[WDT-PRO only] A command error (connection unit error) was reported. A communication error may have occurred between the WDT unit and the WDB unit. Check whether to recover by turning the power on again. If it does not recover, initialize it and if it does not recover, contact Patlite (there is a possibility of hardware failure).
0x80110084	[WDT-PRO only] A command error (wireless module response error) has been notified. Check whether to recover by turning the power on again. Otherwise, the wireless communication module may be faulty. Contact Patlite.
0x80110086	[WDT-PRO only] A command error (data acquisition error) was reported. You will be notified when data acquisition is performed while changing settings. Perform data acquisition after the setting change is completed.
0x801100C0	[WDT-PRO only]

	A command error (initialization error) was reported. You are notified when initialization fails. The hardware may be defective. Contact Patlite.
0x801100FF	[WDT-PRO] A command error (exception error) was reported. An unexpected error has occurred. If this happens frequently, contact Patlite. [WDT] A command error was notified. It is possible that you are sending an unsupported command. Make sure that the series you are connecting to matches the Series option.

Table 4-10 shows common error codes when using WDR-PRO provider.

Table 4-10 Common Error Codes

Error Number	Description
0x8091274D	Error code of Windows socket. Other software may be connected to the WDR (-PRO). Disconnect the WDR (-PRO) from the target software and try connecting again. Or, the port may be incorrect. Check the port number of the connection destination.

5. Sample Code(C#)

```
using ORiN2.ManagedCAO;
using System;

namespace ConsoleApp
{
    class Program
    {
        static void Main(string[] args)
        {
            CCaoEngine eng = new CCaoEngine();
            CCaoWorkspace ws = eng.AddWorkspace("sample",string.Empty);
            CCaoController ctrl = null;

            try{
                ctrl = ws.AddController("ctrl1", "CaoProv.PATLITE.WDR-PRO", string.Empty,
"Conn=tcp:192.168.10.1,Timeout=3000");
            }
            catch{
                if (eng != null){
                    eng.Dispose();
                }
                eng = null;
                ws = null;
                ctrl = null;
                return;
            }

            Console.WriteLine("Wait until one or more WDT(-PRO)s join the network");
            Console.WriteLine("Press ESC to exit");

            // Break loop when one or more WDT(-PRO)s join the network or escape key is pressed
            while (true){
                if (ctrl.GetExtensionNames(string.Empty) != null){
                    break;
```

```
        }

        if (!Console.KeyAvailable){
            continue;
        }

        if(Console.ReadKey().Key == ConsoleKey.Escape){
            if (eng != null){
                eng.Dispose();
            }
            eng = null;
            ws = null;
            ctrl = null;
            return;
        }
    }

    string[] extensionNames = ctrl.GetExtensionNames(string.Empty);

    CCaoExtension extWdt = ctrl.AddExtension(extensionNames[0], "Series=0");            // WDT-PRO
    //CCaoExtension extWdt = ctrl.AddExtension(extensionNames[0], "Series=1");          // WDT

    CCaoVariable varStatus = extWdt.AddVariable("@STATUS", string.Empty);
    Console.WriteLine("@STATUS=" + varStatus.Value);                  // @STATUS=0(CONNECTED)

    CCaoVariable varLightBuzzer = extWdt.AddVariable("@LIGHT_BUZZER", "TimeStamp=true");
    object[] status = (object[])varLightBuzzer.Value;
    Console.WriteLine("RED=" + status[0]);                // RED=2(LightOff)
    Console.WriteLine("YELLOW=" +status[1]);              // YELLOW=3(LightOn)
    Console.WriteLine("GREEN=" +status[2]);               // GREEN=4(Flash)
    Console.WriteLine("BLUE=" +status[3]);                // BLUE=2(LightOff)
    Console.WriteLine("WHITE=" +status[4]);               // WHITE=2(LightOff)
    Console.WriteLine("BUZZER=" +status[5]);              // BUZZER=1(Non-buzzing)
    Console.WriteLine("TimeStamp=" +status[6]);           // TimeStamp=2020/01/01 00:00:00

    if (eng != null){
```

```
                    eng.Dispose();
                }
                eng = null;
                ws = null;
                ctrl = null;
            }
        }
    }
```

6. Communication protocol command correspondence table

Table 6-1 shows the correspondence between the functions implemented in this provider and the communication commands of PATLITE's Application Note for Wireless Data Communication System WD PRO Series.

Table 6-1 Correspondence between communication commands

Method/Function/Variable Name	Get/put	Object	Communication Commands
AddController	-	WDT-LR-Z2/ WDT-LR-Z2-PRO	Wireless module reset request WDT disconnected/participation request
Keep alive function	-	WDT-LR-Z2/ WDT-LR-Z2-PRO	Count-value outputting (with register function) requesting
	-	WDT-6M-Z2/ WDT-5E-Z2	Count value output request
Status change	-	WDT-LR-Z2	Signal light information notification
		WDT-LR-Z2-PRO	Notification of input information Info (RS-232C) notification
@LIGHT_BUZZER	Put	WDT-LR-Z2-PRO	Signal lamp indicator control request
@RED	Put	WDT-LR-Z2-PRO	Signal lamp indicator control request
@YELLOW	Put	WDT-LR-Z2-PRO	Signal lamp indicator control request
@GREEN	Put	WDT-LR-Z2-PRO	Signal lamp indicator control request
@BLUE	Put	WDT-LR-Z2-PRO	Signal lamp indicator control request
@WHITE	Put	WDT-LR-Z2-PRO	Signal lamp indicator control request
@BUZZER	Put	WDT-LR-Z2-PRO	Signal lamp indicator control request
@COUNT	Put	WDT-LR-Z2/ WDT-LR-Z2-PRO	Count value output (with register function) request
	Put	WDT-6M-Z2/ WDT-5E-Z2	Count value registration request