

PATLITE WDR-PRO プロバイダ

Version 1.2.0

ユーザーズ ガイド

July 28, 2022

【備考】

【改版履歴】

バージョン	日付	内容
1.0.0	2020-06-03	初版.
	2021-04-09	使用プロトコルを明記, 誤字修正
1.1.0	2021-09-08	WDT-LR-Z2 シリーズでもカウンタ値を使用できるように対応.
	2021-10-22	システム変数名の誤記修正. @LIGHTS_BUZZER→@LIGHT_BUZZER
1.2.0	2022-07-19	Series オプションに WDT-6M-Z2, WDT-5E-Z2 シリーズを追加. WDT-6M-Z2, WDT-5E-Z2 シリーズでもカウンタ値を取得できるように対応.
	2022-07-28	GetExtensionNames 実行時に WDT の MAC アドレスが「0」始まりの場合, 「0」が欠落した形で表示される問題を修正.

【対応機器】

機種	バージョン	注意事項
WDR-L-Z2	1.00 ~	AC アダプタ付属
WDR-LE-Z2	1.00 ~	AC アダプタ無し(海外向け)
WDR-L-Z2-PRO	1.00 ~	国内専用 AC アダプタ付属
WDR-LE-Z2-PRO	1.00 ~	海外用 AC アダプタ無し
WDR-L-Z2-PRO-L	1.00 ~	(Lite 版)国内専用 AC アダプタ付属
WDR-LE-Z2-PRO-L	1.00 ~	(Lite 版)海外用 AC アダプタ無し
WDT-6LR-Z2-PRO		
WDT-6M-Z2	2.00~	(1.00~, 3.00~はリリースなし)
WDT-5E-Z2	2.00~	(1.00~, 3.00~はリリースなし)
WDT-6LR-Z2	3.04~	(1.00~3.03 はリリースなし ※カウンタ値対応は 3.06~
WDT-5LR-Z2	3.04~	(1.00~3.03 はリリースなし ※カウンタ値対応は 3.06~
WDT-4LR-Z2	3.04~	(1.00~3.03 はリリースなし ※カウンタ値対応は 3.06~

【動作確認機種】

機種	バージョン	注意事項
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	

目次

1. はじめに	5
2. 対応機種組み合わせ	6
3. 環境のセットアップ	6
4. プロバイダの概要	10
4.1. 概要	10
4.1.1. WDT(-PRO)の状態遷移	11
4.1.2. プロバイダの内部動作	12
4.1.3. 離脱判断のタイミング	12
4.1.4. 生存確認機能	13
4.2. メソッド・プロパティ	14
4.2.1. CaoWorkspace::AddController メソッド	14
4.2.2. CaoController::get_VariableNames プロパティ	15
4.2.3. CaoController::AddVariable メソッド	15
4.2.4. CaoController::get_ExtensionNames プロパティ	16
4.2.5. CaoController::AddExtension	16
4.2.6. CaoController::OnMessage イベント	19
4.2.7. CaoExtension::get_VariableNames プロパティ	19
4.2.8. CaoExtension::AddVariable メソッド	23
4.2.9. CaoVariable::get_Value プロパティ	26
4.2.10. CaoVariable::put_Value プロパティ	26
4.3. エラーコード	27
5. サンプルコード(C#)	30
6. 通信プロトコルコマンド対応表	33

1. はじめに

本書は PATLITE 社製 WDR-L(E)-Z2(以下, WDR), WDR-L(E)-Z2-PRO, WDR-L(E)-Z2-PRO-L(以下, WDR-PRO)から信号灯情報を取得・制御する WDR-PRO プロバイダのユーザーズガイドです. WDR(-PRO)は複数の信号灯情報を無線で受信できる受信機です. 図 1-1 は WDR-PRO プロバイダの環境構成です. PC と WDR(-PRO)間を Ethernet ケーブルで接続し, WDR(-PRO) と信号灯間は無線通信を行います. 1 台の WDR(-PRO)で最大 30 台の信号灯情報を取得可能です. 信号灯にはデータ送信機である WDT-6LR-Z2, WDT-5LR-Z2, WDT-6M-Z2, WDT-5M-Z2(以下, WDT), WDT-6LR-Z2-PRO(以下, WDT-PRO)を取り付ける必要があります. また, WDT-PRO にベースユニット WDB-D80S-PRO(以下, WDB)を取り付けることで, RS232C ケーブルを介してスキャナーなどと接続でき, スキャナーで読み取ったデータなどを送信することが可能です.

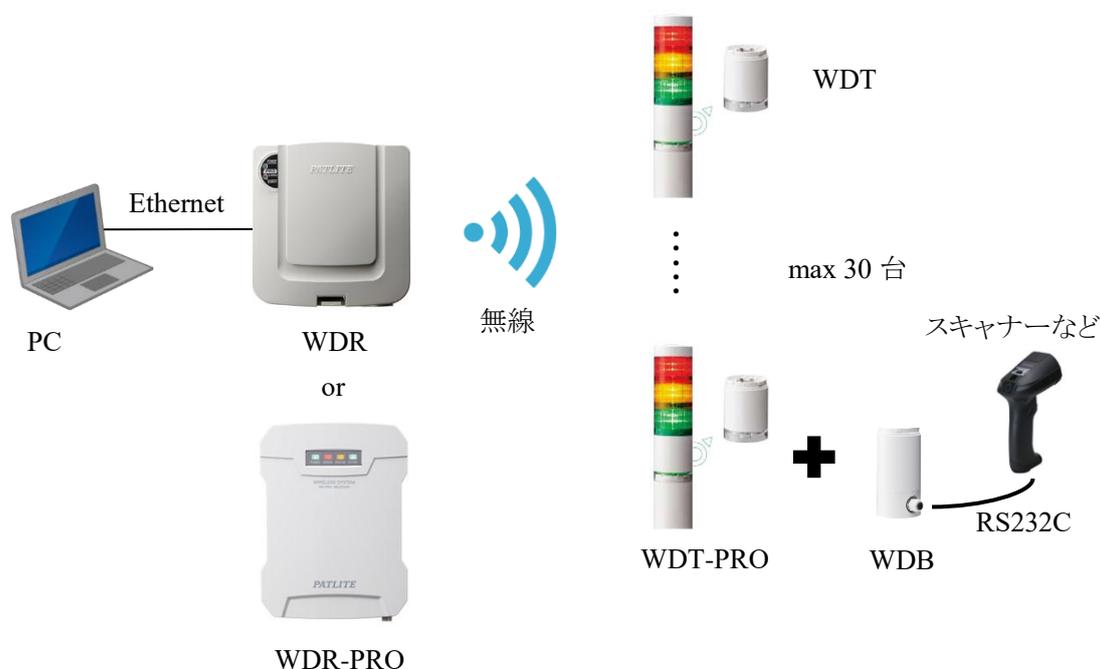


図 1-1 WDR-PRO プロバイダの環境構成

2. 対応機種のご組み合わせ

WDR-PRO プロバイダが対応している受信機及び送信機の組み合わせを表 2-1 に示します。WDR-PRO プロバイダは WDR プロトコル(旧プロトコル)に沿って開発しているため、すべての組み合わせに対して通信が可能です。

表 2-1 WDR-PRO プロバイダの対応表

送信機 受信機	WDT-6M-Z2, WDT-5E-Z2 シリーズ	WDT-LR-Z2 シリーズ	WDT-LR-Z2=PRO シリーズ
WDR-Z2 シリーズ	○	○	○
WDR-Z2-PRO シリ ーズ	○	○	○

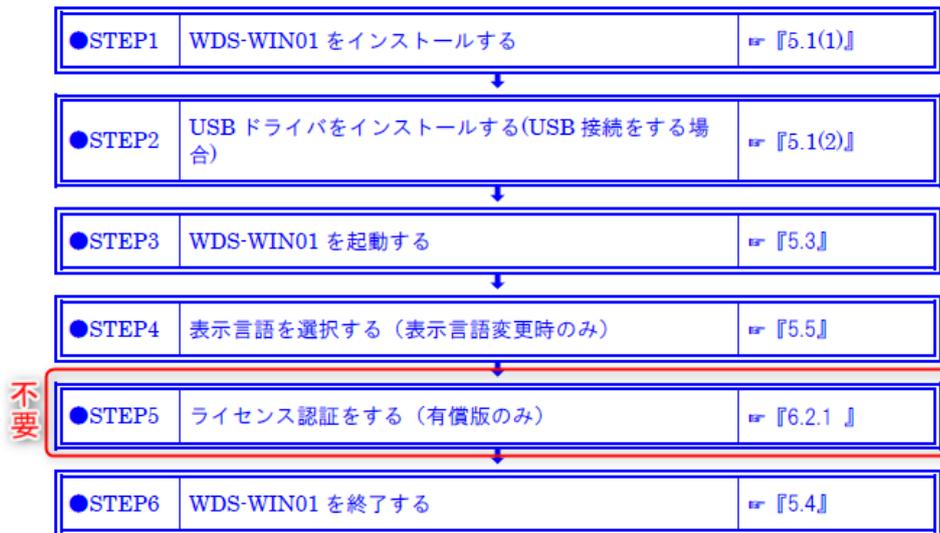
3. 環境のセットアップ

WDR-PRO プロバイダを使用するためには、WDR(-PRO)および WDT(-PRO)のセットアップをする必要があります。セットアップには、パトライト社製システム運用ソフトウェア「WDS-WIN01」をご使用ください。WDS-WIN01 はパトライト社の HP のマイページより DL 可能です。

セットアップ方法は、「ワイヤレス・データ通信システム システム運用ソフトウェア MODEL: WDS-WIN01 総合取扱説明書」をご覧ください。「4.1 WDS-WIN01をはじめて使用する」と「4.2 WDTとWDR の設定をおこなう」の手順に従い、セットアップを完了してください。

WDR-PRO プロバイダを使用するだけであれば、4.1 章の「STEP5 ライセンス認証をする(有償版のみ)」の手順は不要です。

4.1 WDS-WIN01 をはじめて使用する



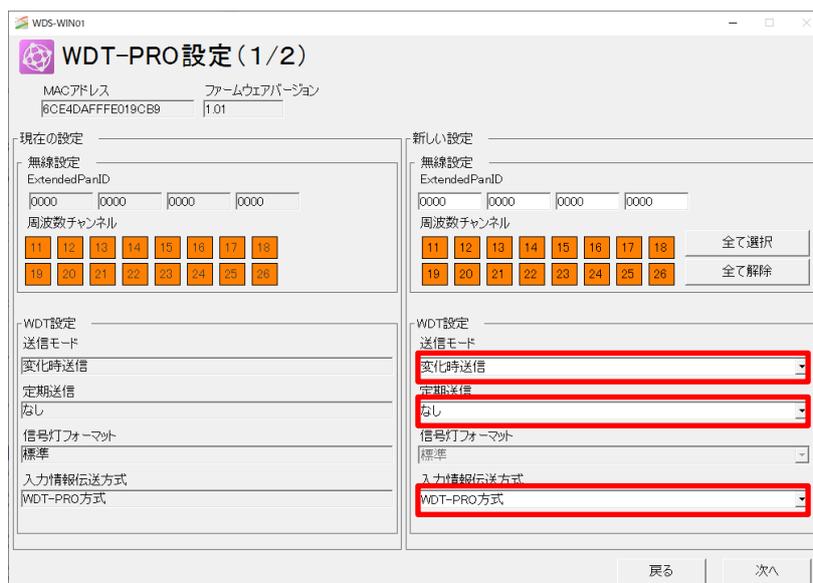
「4.2 WDT と WDR の設定をおこなう」の「STEP5 WDT の設定をする」において、WDR-PRO プロバイダが正常に値を取得するために、WDT-PRO および WDT の設定は以下の通りに設定してください。

< WDT-PRO >

送信モード: **変化時送信**

定期送信: **なし**

入力情報伝送方式: **WDT-PRO 方式**



< WDT >

送信モード: **変化時送信**



WDT-6LR-Z2, WDT-5LR-Z2 は、信号灯情報の他にブザー情報を追加することができます。ブザー情報を追加する場合、図 3-1 に示す WDT-6LR-Z2, WDT-5LR-Z2 の底面に配置されている設定スイッチ 3 を ON にしてください。

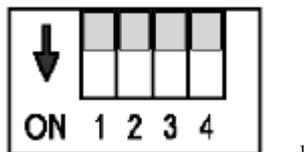


図 3-1 設定スイッチ

4.2 章の「STEP6 WDR の設定をする」において、ネットワーク開始方法は必ず「**自動起動(推奨)**」を選択してください。「手動起動」を選択している場合、WDR-PRO プロバイダと WDR(-PRO)の接続が開始できなくなります。



¹ PATLITE 社「WDT-5LR-Z2, WDT-6LR-Z2, WDR-L(E)-Z2 総合取扱説明書」より

4. プロバイダの概要

4.1. 概要

図 4-1 はプロバイダの各クラスと WDR(-PRO)および WDT(-PRO)の対応図です。CaoProvController は WDR(-PRO)本体に、CaoProvExtension は WDT(-PRO)に対応しています。CaoProvVariable は WDT-PRO が保持する信号灯情報、外部入力情報、カウンタ値などに対応しています。WDT の場合は外部入力情報を保持できないため、信号灯情報とカウンタ値情報のみになります。

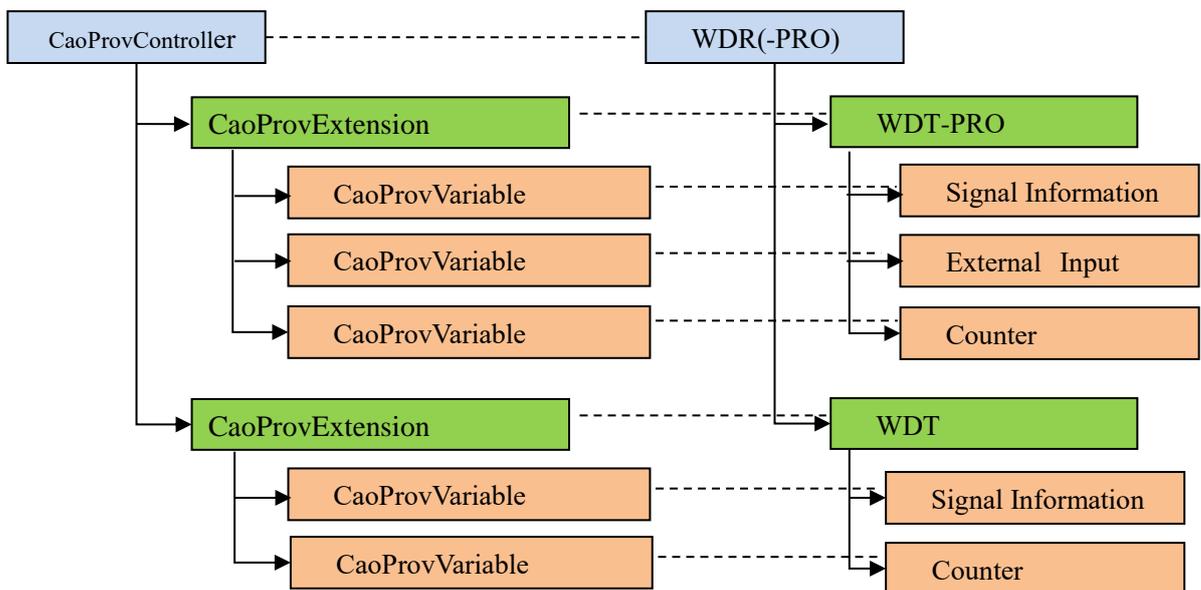


図 4-1 プロバイダの各クラスと WDR(-PRO)の対応図

WDR-PRO プロバイダのファイル形式は DLL(Dynamic Link Library)です。詳細を表 4-1 に示します。

表 4-1 WDR-PRO プロバイダ

ファイル名	CaoProvPatliteWdr-Pro.dll
ProgID	CaoProv.PATLITE.WDR-PRO
レジストリ登録 ²	regsvr32 CaoProvPatliteWdr-Pro.dll
レジストリ登録の抹消	regsvr32 /u CaoProvPatliteWdr-Pro.dll

² ORiN SDK でインストールした場合は手動で登録/抹消する必要はありません。

4.1.1. WDT(-PRO)の状態遷移

WDT(-PRO)は未接続、接続中、離脱中の3つに状態遷移します。WDT(-PRO)がWDR(-PRO)のネットワークに参加する前は未接続状態であり、この状態ではWDT(-PRO)から信号灯情報などを取得することができません。WDT(-PRO)がネットワークに参加すると接続中状態となり、WDT(-PRO)の信号灯情報を取得することができます①。接続中のWDT(-PRO)とWDR(-PRO)が通信不能になると、本プロバイダはWDT(-PRO)に異常が発生したとみなし、接続中から離脱中に状態遷移します②。WDT(-PRO)が異常状態から復帰し、再度WDR(-PRO)のネットワークに参加すると、接続中状態に戻ります③。WDT(-PRO)の状態は後述する「@STATUS」変数の値で判断します。未接続状態の場合「-1」、接続中の場合「0」、離脱中の場合「-2」を返します。

本プロバイダのCaoWorkspace::AddControllerを実行すると、WDR(-PRO)のネットワークがリセットされます。そのため、AddController 実行直後、すべてのWDT(-PRO)は未接続状態となります。未接続状態からWDT(-PRO)が接続状態になるまで、最大100秒かかる場合があります。

また通信仕様上、プロバイダの値が未接続状態から接続中状態に切り替わらない可能性があります。その場合はCaoWorkspace::AddController を再実行してください。

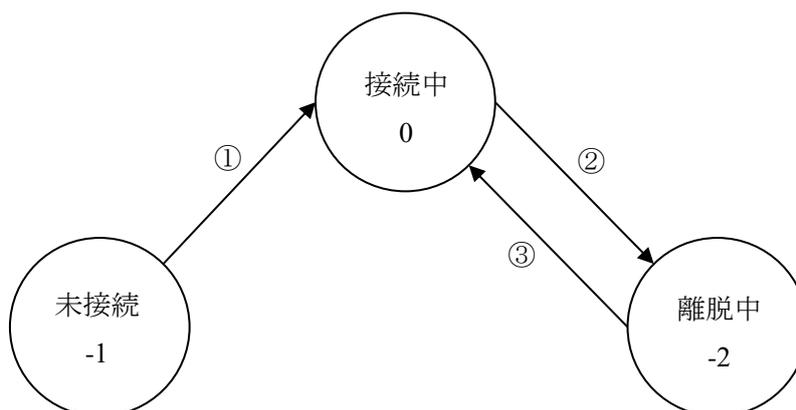


図 4-2 WDT(-PRO)の状態遷移図

4.1.2. プロバイダの内部動作

図 4-3 に WDR-PRO プロバイダの内部動作を示します。変化時送信モードでは、信号灯情報・外部入力情報が変化するタイミングで、WDT(-PRO)が WDR(-PRO)に対して信号データを送信します(①)。受信が完了すると、WDR(-PRO)が PC に対して受信データを送信し、WDR-PRO プロバイダ内のメモリ空間に信号灯情報と PC が受信した時刻を記録します(②)。WDR-PRO プロバイダは、GetValue が実行されると、メモリ空間に記録した信号灯情報と受信時刻を取得して表示します(③)。

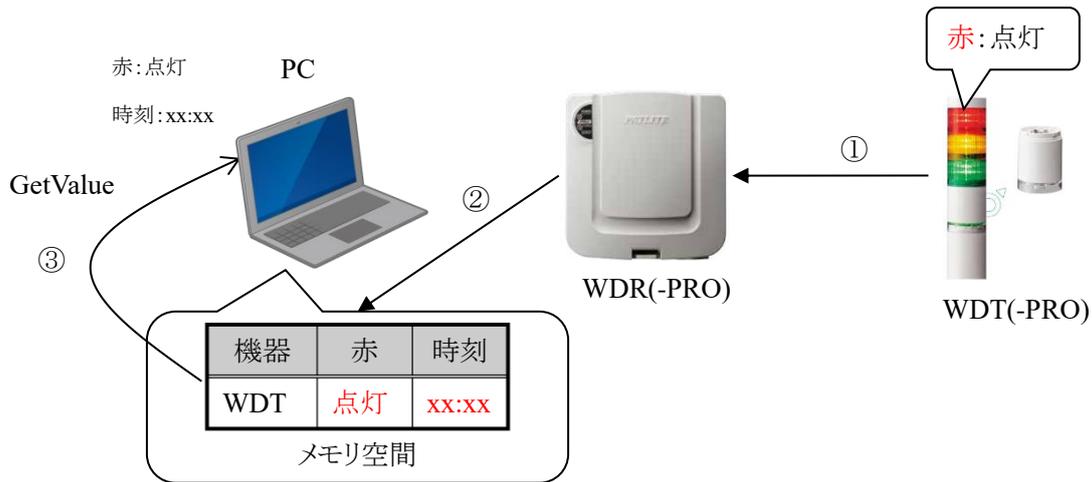


図 4-3 WDR-PRO プロバイダの内部動作

4.1.3. 離脱判断のタイミング

変化時送信モードは信号灯が変化したタイミングで信号情報を送信するため、WDT(-PRO)が WDR(-PRO)の無線ネットワークから切断されても認識ができません。そのため、WDT(-PRO)が正常状態かどうか監視するために、WDR(-PRO)は定期的に WDT(-PRO)と通信しています。図 4-4 は WDR(-PRO)と WDT(-PRO)間の生存確認機能を示します。WDR(-PRO)と WDT(-PRO)の通信が 80 秒間途絶えたとき(①)、WDT(-PRO)はネットワークから離脱したと判断して WDR-PRO プロバイダに通知します(②)。離脱通知をプロバイダが受け取ると、WDT(-PRO)が離脱中と判断します(③)。



図 4-4 WDR(-PRO)と WDT(-PRO)間の離脱判断タイミング

4.1.4. 生存確認機能

「4.1.3 離脱判断のタイミング」で説明した規定の離脱判断タイミング(80 秒)より早く WDT(-PRO)の異常状態を検知したい場合、WDR-PRO プロバイダの生存確認機能を使用します。図 4-5 に WDR-PRO プロバイダの生存確認機能を示します。WDR-PRO プロバイダは WDT(-PRO)が正常状態かどうか監視するために、WDR(-PRO)を介して定期的に WDT(-PRO)と生存確認要求を行います(①)。WDT(-PRO)が要求を正常に受け取ると、応答を返します(②)。応答を PC が受け取ると WDT(-PRO)が接続中の状態であると判断します(②)。

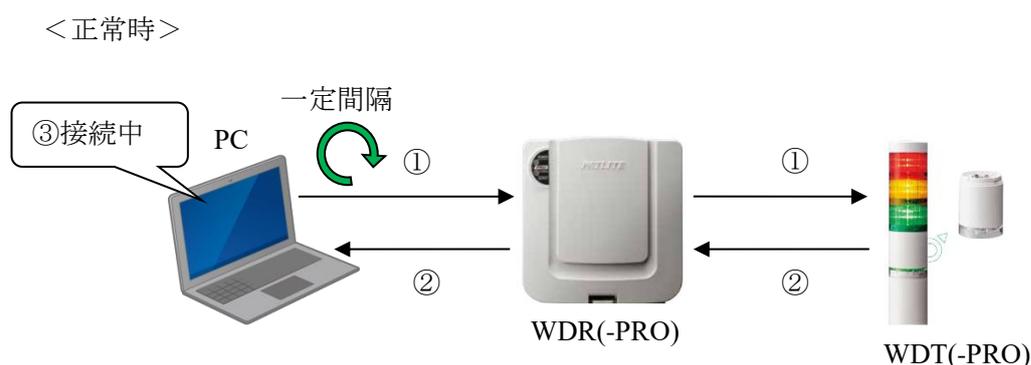


図 4-5 WDR-PRO プロバイダの生存確認機能(正常時)

生存確認要求をしてから 17 秒間(WDR(-PRO)の規定時間)応答がない場合は、要求が失敗したと見なします。失敗回数が再送回数の上限に達したとき、WDT(-PRO)は異常状態とみなし、ネットワークから離脱したと判断します。再送回数の上限は CaoController::AddExtension 時の Retry オプションで指定できます。また、生存確認要求を出す周期は CaoController::AddExtension 時の Interval オプションで指定できます。Interval オプションに「-1」を指定することで、本機能は実行されません。



図 4-6 WDR-PRO プロバイダの生存確認機能(異常時)

4.2. メソッド・プロパティ

4.2.1. CaoWorkspace::AddController メソッド

本プロバイダでは、CaoWorkspace::AddController で接続パラメータを設定して通信(TCP)の接続を行います。以下に AddController の仕様を示します。AddController 実行直後、すべての WDT(-PRO)は未接続状態になります。

書式

```
AddController(
    BSTR bstrCtrlName,           // [in] コントローラ名(任意のコントローラ名を入力して下さい)
    BSTR bstrProvName,          // [in] プロバイダ名. 固定値 = "CaoProv.PATLITE.WDR-PRO "
    BSTR bstrPcName,            // [in] プロバイダの実行マシン名
    BSTR bstrOption              // [in] オプション文字列
);
```

表 4-2 CaoWorkspace::AddController のオプション文字列

設定項目	オプション文字列	必須	説明	設定範囲	デフォルト値
接続先	Conn	○	接続先を以下の形式で指定してください。 「Conn = tcp:<IP アドレス>[:<ポート番号>] ³ 」	ポート番号: 1 ~ 65535	ポート番号: 10001
タイムアウト	Timeout	-	接続先とのタイムアウトの秒数を ms で指定できます。	0 ~ 4294967295	3000

例 1) IP アドレス 192.168.10.1, ポート番号 10001, タイムアウト 3000 ms で接続する場合

```
"Conn = tcp:192.168.10.1"
```

例 2) IP アドレス 192.168.10.1, ポート番号 20001, タイムアウト 1000 ms で接続する場合

```
"Conn = tcp:192.168.10.1:20001, Timeout = 1000"
```

使用例(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 角括弧("[]")内は省略可能なパラメータです。

4.2.2. CaoController::get_VariableNames プロパティ

WDR-PRO プロバイダで使用可能な変数リストを取得します。表 4-3 のリストが取得できます。

戻り値の型	説明
VT_ARRAY VT_BSTR	CaoController の AddVariable の変数名に使用できる変数一覧が取得できます。

表 4-3 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「PATLITE」を返します。	○	-
@VERSION	VT_BSTR	プロバイダのバージョンを返します。	○	-

使用例(C#)

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

4.2.3. CaoController::AddVariable メソッド

CaoController から CaoVariable オブジェクトを生成します。表 4-3 のシステム変数一覧から変数名を入力してください。オプション文字列は入力不要です。

書式

```
AddVariable (
    BSTR bstrVarName, // [in] 変数名(システム変数名を指定)
    BSTR bstrOption   // [in] オプション文字列
);
```

使用例(C#)

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

4.2.4. CaoController::get_ExtensionNames プロパティ

接続中または離脱中の WDT(-PRO)の MAC アドレス一覧が取得できます。

戻り値の型	説明
VT_ARRAY VT_BSTR	CaoController の AddExtension の変数名に使用できる変数一覧が取得できます。

WDT(-PRO)の Mac アドレスは機器のラベルに記述されています(図 4-7)。

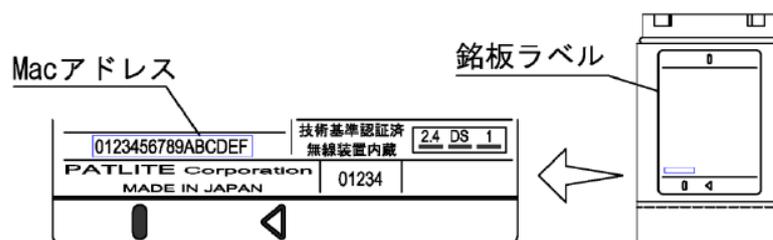


図 4-7 WDT(-PRO)の Mac アドレスの印字箇所⁴

使用例(C#)

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

4.2.5. CaoController::AddExtension

CaoController から CaoExtension オブジェクトを生成します。変数名には接続したい WDT(-PRO)の MAC アドレスを指定し、オプション文字列には表 4-4 のオプション文字列を使用します。WDT-PRO でスキャナーなどから読み取ったデータを WDR-PRO プロバイダに通知したい場合、オプション文字列の Mode, CodePage, Header, Term の指定をしてください。WDT の場合は、指定する必要はありません。

書式

```
AddExtension (
    BSTR bstrVarName, // [in] 変数名(WDT(-PRO)の MAC アドレスを指定)
    BSTR bstrOption   // [in] オプション文字列
);
```

⁴ PATLITE 社「WDT-5LR-Z2, WDT-6LR-Z2, WDR-L(E)-Z2 総合取扱説明書」より

表 4-4 CaoController::AddExtension のオプション文字列

設定項目	オプション文字列	必須	説明	設定範囲	デフォルト値
シリーズ	Series	○	接続先の WDT のシリーズを選択してください。	0: WDT-LR-Z2-PRO 1: WDT-LR-Z2 2: WDT-6M-Z2 WDT-5E-Z2	-
送信間隔	Interval	-	生存確認機能を送信する間隔[ms]を指定できます。 設定範囲: *-1 を指定した場合、生存確認機能が無効になります。その場合、後述する@COUNT(カウント値)の値を取得できません。	-1, 500 ~ 2147483647	20000
再送回数の上限	Retry	-	生存確認機能の再送信が連続して何回発生した場合、WDT(-PRO)が離脱状態と判断するかを指定できます。	1 ~ 7	3
モード	Mode	-	WDT-PRO で使用するオプション文字列です。RS232C ケーブルから送信されたデータの出力方式を指定できます。	0:テキスト 1:バイナリ	0
コードページ	CodePage	-	WDT-PRO で使用するオプション文字列です。RS232C ケーブルから送信されたデータのデコード方式を指定できます。Mode オプションが 1:バイナリするとき、このオプション文字列は無視されます。 以下、代表的なコードページを記述します。 0:現在の WindowsANSI コードページ 932:Shift-JIS 1200:UTF-16 12000:UTF-32 65000:UTF-7 65001:UTF-8	0 ~ 65535	0
ヘッダー	Header	-	WDT-PRO で使用するオプション文字列です。	0:なし	0

			RS232C ケーブルから送信されたデータのヘッダーを指定できます。指定したヘッダーを取り除いたデータが出力されます。Mode オプションが 1:バイナリ のとき、このオプション文字列は無視されます。	1:STX	
ターミネータ	Term	-	WDT-PRO で使用するオプション文字列です。RS232C ケーブルから送信されたデータのターミネータを指定できます。指定したターミネータを取り除いたデータが出力されます。Mode オプションが 1:バイナリ のとき、このオプション文字列は無視されます。	0:CR 1:LF 2:CR+LF 3:ETX	0

例 1) シリーズ: WDT-LR-Z2, 生存確認機能の送信間隔: 20000 ms, 再送回数: 3 回の場合
"Series=1"

例 2) シリーズ: WDT-LR-Z2-PRO, 生存確認機能なし, データ出力方式: テキスト, デコード方式: UTF-16, ヘッダー: STX, ターミネータ: CR+LF の場合
"Series=0, Interval=-1, CodePage=1200, Header=1, Term=2"

例 3) シリーズ: WDT-LR-Z2-PRO, 生存確認機能の送信間隔: 500 ms, 再送回数: 7 回, データ出力方式: バイナリ の場合
"Series=0, Interval=500, Retry=7, Mode=1"

使用例(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 イベント

スキャナーなどから読み取った RS232C データを WDR-PRO プロバイダがデータを受信すると、CaoController クラスの OnMessage イベントとしてクライアントにデータを受け渡します。送信元の WDT(-PRO)が CaoExtension オブジェクトとして存在していない場合、受信した RS232C データは破棄されますので、ご注意ください。

OnMessage イベントのデータ部を表 4-5 に示します。Message::Number は 0、Message::Source はデータ送信元の WDT-PRO の MAC アドレスを示します。また、Message::Value の型は AddExtension 時の Mode オプションがテキストモードの場合 VT_BSTR、バイナリモードの場合 VT_ARRAY|VT_I1 になります。

表 4-5 OnMessage イベントのデータ部

項目	データ型	値
Number	VT_I4	0
DateTime	VT_DATE	プロバイダがデータを受信した時刻
Destination	VT_BSTR	なし
Source	VT_BSTR	WDT(-PRO)の MAC アドレス
Value	(テキストモードの場合):VT_BSTR (バイナリモードの場合):VT_ARRAY VT_I1	RS232C データ
Description	VT_BSTR	なし

4.2.7. CaoExtension::get_VariableNames プロパティ

CaoExtension で使用可能な変数リストを取得します。表 4-6 のリストが取得できます。WDT-PRO ではすべてのシステム変数が使用可能ですが、WDT の場合は@EXTERNAL_INPUTS,@EXTERNAL_INPUT1~8 が使用できません。

戻り値の型	説明
VT_ARRAY VT_BSTR	CaoExtension の AddVariable の変数名に使用できる変数一覧が取得できます。

表 4-6 CaoExtension クラス システム変数一覧

変数名	データ型	説明	属性 WDT-PRO		属性 WDT									
			get	put	get	put								
@STATUS	VT_I1	<p>WDT(-PRO)の接続状態を示します。</p> <table border="1"> <thead> <tr> <th>状態</th> <th>値</th> </tr> </thead> <tbody> <tr> <td>未接続</td> <td>-1</td> </tr> <tr> <td>接続中</td> <td>0</td> </tr> <tr> <td>離脱中</td> <td>-2</td> </tr> </tbody> </table>	状態	値	未接続	-1	接続中	0	離脱中	-2	○	-	○	-
状態	値													
未接続	-1													
接続中	0													
離脱中	-2													
@LQI	VT_I2	<p>WDT(-PRO)との通信品質を示します。0～255の範囲で表されます。生存確認機能の Interval 間隔および信号灯の色・外部入力に変化したタイミングで値が更新されます。</p> <table border="1"> <thead> <tr> <th>状態</th> <th>値</th> </tr> </thead> <tbody> <tr> <td>未接続</td> <td>-1</td> </tr> <tr> <td>接続中</td> <td>0～50:悪 51～100:やや悪 101～150:良 151～255:最良</td> </tr> <tr> <td>離脱中</td> <td>-2</td> </tr> </tbody> </table>	状態	値	未接続	-1	接続中	0～50:悪 51～100:やや悪 101～150:良 151～255:最良	離脱中	-2	○	-	○	-
状態	値													
未接続	-1													
接続中	0～50:悪 51～100:やや悪 101～150:良 151～255:最良													
離脱中	-2													
@LIGHT_B UZZER	VT_ARRA Y VT_I1	<p>信号灯の各色およびブザー情報を示します。Put することで 5 色の信号灯とブザーを同時に制御できます。制御中は外部接点からの制御はできません。外部接点による制御を行うには、非制御(0)を Put してください。</p> <p><信号灯></p> <table border="1"> <thead> <tr> <th>状態</th> <th>値</th> </tr> </thead> <tbody> <tr> <td>未接続</td> <td>-1</td> </tr> <tr> <td>接続中</td> <td>0:非制御(PUTのみ) 1:未使用(PUT不可) 2:消灯 3:点灯</td> </tr> </tbody> </table>	状態	値	未接続	-1	接続中	0:非制御(PUTのみ) 1:未使用(PUT不可) 2:消灯 3:点灯	○	○	○	-		
状態	値													
未接続	-1													
接続中	0:非制御(PUTのみ) 1:未使用(PUT不可) 2:消灯 3:点灯													

		<table border="1"> <tr> <td></td> <td>4:点滅 5:トリプルフラッシュ</td> </tr> <tr> <td>離脱中</td> <td>-2</td> </tr> </table>		4:点滅 5:トリプルフラッシュ	離脱中	-2								
	4:点滅 5:トリプルフラッシュ													
離脱中	-2													
		<p><ブザー></p> <table border="1"> <thead> <tr> <th>状態</th> <th>値</th> </tr> </thead> <tbody> <tr> <td>未接続</td> <td>-1</td> </tr> <tr> <td>接続中</td> <td>0:非制御(PUTのみ) 1:非吹鳴 2:吹鳴 3:断続吹鳴</td> </tr> <tr> <td>離脱中</td> <td>-2</td> </tr> </tbody> </table>	状態	値	未接続	-1	接続中	0:非制御(PUTのみ) 1:非吹鳴 2:吹鳴 3:断続吹鳴	離脱中	-2				
状態	値													
未接続	-1													
接続中	0:非制御(PUTのみ) 1:非吹鳴 2:吹鳴 3:断続吹鳴													
離脱中	-2													
	0	VT_I1	信号灯の赤色の状態を示します。											
	1	VT_I1	信号灯の黄色の状態を示します。											
	2	VT_I1	信号灯の緑色の状態を示します。											
	3	VT_I1	信号灯の青色の状態を示します。											
	4	VT_I1	信号灯の白色の状態を示します。											
	5	VT_I1	ブザーの状態を示します。											
@RED		VT_I1	赤色の信号灯の状態を示します。Put 操作により、赤色の信号灯を制御できます。各状態における値は @LIGHT_BUZZER の<信号灯>と同様です。	○	○	○	-							
@YELLOW		VT_I1	黄色の信号灯の状態を示します。Put 操作により、黄色の信号灯を制御できます。各状態における値は @LIGHT_BUZZER の<信号灯>と同様です。	○	○	○	-							
@GREEN		VT_I1	緑色の信号灯の状態を示します。Put 操作により、緑色の信号灯を制御できます。各状態における値は @LIGHT_BUZZER の<信号灯>と同様です。	○	○	○	-							
@BLUE		VT_I1	青色の信号灯の状態を示します。Put 操作により、青色の信号灯を制御できます。各状態における値は @LIGHT_BUZZER の<信号灯>と同様です。	○	○	○	-							
@WHITE		VT_I1	白色の信号灯の状態を示します。Put 操作により、白色の信号灯を制御できます。各状態における値は @LIGHT_BUZZER の<信号灯>と同様です。	○	○	○	-							
@BUZZER		VT_I1	ブザーの状態を示します。Put 操作により、ブザーを制御できます。各状態における値は @LIGHT_BUZZER の<ブザー>と同様です。	○	○	○	-							

@EXTERNA L_INPUTS	VT_ARRA Y VT_I1	WDT-PRO シリーズのみ使用可能です。外部入力 1 ~8 を示します。Put での制御はできません。 <外部入力> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>状態</th> <th>値</th> </tr> </thead> <tbody> <tr> <td>未接続</td> <td>-1</td> </tr> <tr> <td>接続中</td> <td>0:OFF 1:ON</td> </tr> <tr> <td>離脱中</td> <td>-2</td> </tr> </tbody> </table>	状態	値	未接続	-1	接続中	0:OFF 1:ON	離脱中	-2	○	-	-	-
	状態	値												
	未接続	-1												
	接続中	0:OFF 1:ON												
	離脱中	-2												
	0	VT_I1	外部入力 1 の状態を示します。											
	1	VT_I1	外部入力 2 の状態を示します。											
	2	VT_I1	外部入力 3 の状態を示します。											
	3	VT_I1	外部入力 4 の状態を示します。											
4	VT_I1	外部入力 5 の状態を示します。												
5	VT_I1	外部入力 6 の状態を示します。												
6	VT_I1	外部入力 7 の状態を示します。												
7	VT_I1	外部入力 8 の状態を示します。												
@EXTERNA L_INPUT1	VT_I1	外部入力 1 の状態を示します。各状態における値は @EXTERNAL_INPUTS と同様です。	○	-	-	-								
@EXTERNA L_INPUT2	VT_I1	外部入力 2 の状態を示します。各状態における値は @EXTERNAL_INPUTS と同様です。	○	-	-	-								
@EXTERNA L_INPUT3	VT_I1	外部入力 3 の状態を示します。各状態における値は @EXTERNAL_INPUTS と同様です。	○	-	-	-								
@EXTERNA L_INPUT4	VT_I1	外部入力 4 の状態を示します。各状態における値は @EXTERNAL_INPUTS と同様です。	○	-	-	-								
@EXTERNA L_INPUT5	VT_I1	外部入力 5 の状態を示します。各状態における値は @EXTERNAL_INPUTS と同様です。	○	-	-	-								
@EXTERNA L_INPUT6	VT_I1	外部入力 6 の状態を示します。各状態における値は @EXTERNAL_INPUTS と同様です。	○	-	-	-								
@EXTERNA L_INPUT7	VT_I1	外部入力 7 の状態を示します。各状態における値は @EXTERNAL_INPUTS と同様です。	○	-	-	-								
@EXTERNA L_INPUT8	VT_I1	外部入力 8 の状態を示します。各状態における値は @EXTERNAL_INPUTS と同様です。	○	-	-	-								
@COUNT	VT_I8	カウンタ値を示します。Put 操作により、カウンタ値の 書き換えが可能です。Interval オプションで設定した	○	○	○	○								

		間隔または Put 操作時に値が更新されます。				
		<カウンタ値>				
		状態	値			
		未接続	-1			
		接続中	0~4294967295			
		離脱中	-2			

4.2.8. CaoExtension::AddVariable メソッド

CaoExtension から CaoVariable オブジェクトを生成します。表 4-6 のシステム変数一覧から変数名を入力してください。オプション文字列には表 4-7 に示す文字列を使用します。5 種類のシステム変数で TimeStamp オプションが使用可能です。TimeStamp オプションを使用した際のシステム変数のデータ型および値を表 4-8 に示します。get/put 操作の可否は表 4-6 と同様です。

書式

AddVariable (

BSTR bstrVarName, // [in] 変数名(システム変数名を指定)

BSTR bstrOption // [in] オプション文字列

);

表 4-7 CaoExtension::AddVariable のオプション文字列

設定項目	オプション文字列	必須	説明	設定範囲	デフォルト値
タイムスタンプ	TimeStam mp	-	データを取得する際にタイムスタンプ情報を含めるかを指定します。 このオプションが使用可能なシステム変数は@STATUS, @LQI, @LIGHT_BUZZER, @EXTERNAL_INPUTS, @COUNT の 5 種類です。	false:なし true:あり	false

例1) タイムスタンプありの場合

"TimeStamp=true"

使用例(C#)

```
CCaoVariable varStatus = ctrl.AddVariable("@STATUS", "");
```

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

表 4-8 CaoExtension クラス システム変数(TimeStamp オプション使用時)

変数名	データ型	説明
@STATUS	VT_ARRAY VT_VARIANT	WDT(-PRO)の接続状態とタイムスタンプを示します。接続状態の値は表 4-6 と同様です。
	0 VT_I1	接続状態。
	1 VT_DATE	接続状態のタイムスタンプ。
		状態
未接続		1970/1/1 00:00:00
接続中	接続状態に遷移した時刻	
離脱中	離脱状態に遷移した時刻	
@LQI	VT_ARRAY VT_VARIANT	WDT(-PRO)との通信品質とタイムスタンプを示します。生存確認機能の Interval 間隔, 信号灯の色・外部入力に変化したタイミング, カウンタ値を Put 操作したタイミングで値が更新されます。通信品質の値は表 4-6 と同様です。
	0 VT_I2	通信品質。
	1 VT_DATE	LQI の最終更新時刻を示します。 初期値: 1970/1/1 00:00:00
@LIGHT_B UZZER	VT_ARRAY VT_VARIANT	信号灯・ブザー情報とタイムスタンプを示します。Put 操作により, 5 色の信号灯とブザーを同時に制御できます。制御中は外部接点からの制御はできません。外部接点による制御を行うには, 非制御(0)を Put してください。信号灯・ブザーの値は表 4-6 と同様です。
	0 VT_I1	赤色信号灯の状態
	1 VT_I1	黄色信号灯の状態
	2 VT_I1	緑色信号灯の状態
	3 VT_I1	青色信号灯の状態
	4 VT_I1	白色信号灯の状態
	5 VT_I1	ブザーの状態
	6 VT_DATE	信号灯およびブザーの最終更新時刻を示します。信号灯またはブザーのいずれかが変化すると更新されます。 初期値: 1970/1/1 00:00:00

@EXTERNA L_INPUTS	VT_ARRAY VT_VARIANT	外部入力 1~8 とタイムスタンプを示します。WDT-PRO シリーズのみ使用可能です。外部入力の値は表 4-6 と同様です。	
	0	VT_I1	外部入力 1 の状態を示します。
	1	VT_I1	外部入力 2 の状態を示します。
	2	VT_I1	外部入力 3 の状態を示します。
	3	VT_I1	外部入力 4 の状態を示します。
	4	VT_I1	外部入力 5 の状態を示します。
	5	VT_I1	外部入力 6 の状態を示します。
	6	VT_I1	外部入力 7 の状態を示します。
	7	VT_I1	外部入力 8 の状態を示します。
8	VT_DATE	外部入力 1~8 の最終更新時刻を示します。外部入力のいずれかが変化すると更新されます。 初期値: 1970/1/1 00:00:00	
@COUNT	VT_ARRAY VT_VARIANT	カウンタ値を示します。Put 操作により、カウンタ値の書き換えが可能です。生存確認機能の Interval 間隔またはカウンタ値を Put 操作したタイミングで値が更新されます。カウンタの値は表 4-6 と同様です。	
	0	VT_I8	カウンタ値を示します。
	1	VT_DATE	カウンタの最終更新時刻を示します。 初期値: 1970/1/1 00:00:00

4.2.9. CaoVariable::get_Value プロパティ

AddVariable メソッドで作成した変数の状態を取得します。戻り値の型は AddVariable で指定した変数名によって異なります。表 4-3, 表 4-6 および表 4-8 のシステム変数で get に「○」がついている変数が get_Value プロパティを使用可能です。戻り値は各システム変数によって異なります。

使用例(C#)

```
Debug.WriteLine("@STATUS=" + varStatus.Value); // @STATUS=0 (接続中の場合)
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 プロパティ

AddVariable メソッドで作成した変数に値を設定します。表 4-6 および表 4-8 のシステム変数で put に「○」がついている変数が put_Value プロパティを使用可能です。put_Value の引数は各システム変数によって異なります。

TimeStamp オプションが有効のシステム変数は、以下の使用例で示すように、タイムスタンプの情報を除いた値を put_Value してください。

使用例(C#)

```
// 赤 : トリプルフラッシュ, 黄 : フラッシュ, 緑 : 点灯, 青 : 消灯, 白 : 非制御
// ブザー : 非制御
object[] newStatus = new object[] {5,4,3,2,0,0};
varLightBuzzer.Value = newStatus;
```

4.3. エラーコード

本プロバイダでは固有エラーコードを定義しています。固有エラーコードを表 4-9 に示します。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 4-9 固有エラーコード

エラー番号	説明
0x80100001	オプション文字列 Conn の指定が不正です。Conn = tcp:<IP アドレス>[:<ポート番号>]の形式で指定してください。IP アドレスを 0.0.0.0 ~ 255.255.255.255 の範囲, ポート番号を 1 ~ 65535 の範囲で指定してください。
0x80100002	オプション文字列 Timeout の指定が不正です。0 ~ 4294967295 ms の範囲で指定してください。
0x80100003	オプション文字列 Series の指定が不正です。0 or 1 の値を指定してください。
0x80100004	オプション文字列 Interval の指定が不正です。-1, 500 ~ 4294967295 ms の範囲で指定してください。
0x80100005	オプション文字列 Retry の指定が不正です。1 ~ 7 回の範囲で指定してください。
0x80100006	オプション文字列 Mode の指定が不正です。0 or 1 の値を指定してください。
0x80100007	オプション文字列 Header の指定が不正です。0 or 1 の値を指定してください。
0x80100008	オプション文字列 Term の指定が不正です。0 ~ 3 の範囲で指定してください。
0x80100009	オプション文字列 TimeStamp の指定が不正です。false or true の値を指定してください。
0x8010000A	AddExtension の変数名が不正です。変数名に正しい MAC アドレスを入力してください。
0x8010000B	WDR(-PRO)の起動に失敗しました。接続先の IP アドレスまたはポート番号を再度確認してください。
0x8010000C	WDR(-PRO)のネットワーク開始に失敗しました。WDR(-PRO)を再起動してください。
0x8010000D	WDT(-PRO)の再参加要求に失敗しました。WDR(-PRO)を再起動してください。
0x8010000E	信号灯・ブザーの Put 値が不正です。信号灯は 0,2~5, ブザーは 0~3 の範囲で指定してください。
0x8010000F	生存確認機能が実行中です。時間を空けてから Put 操作を実行してください。

0x80100010	WDT(-PRO)が未接続状態で Put を実行しました. 接続中に状態遷移してから, Put を実行してください.
0x80100011	WDT(-PRO)が離脱状態で Put を実行しました. 接続中に状態遷移してから, Put を実行してください.
0x80100012	前回の送信コマンドが失敗してから 17s 経過していません. 時間を空けてから Put を実行してください.
0x80100013	Put 操作時に WDR(-PRO)と WDT(-PRO)間の通信でタイムアウトが発生しました. WDT(-PRO)の接続状態を確認し, 通信が安定する距離で Put 操作を実行してください.
0x80110010	[WDT-PRO のみ] コマンドエラー (取得情報なし) が通知されました. 要求に対して該当する取得情報がありません. 必要に応じて再取得をおこなうか, 要求する取得情報が間違っていないか確認ください. それでも発生する場合は弊社までお問い合わせください.
0x80110080	[WDT-PRO のみ] コマンドエラー (コマンドエラー) が通知されました. 要求コマンドが間違っています. 本プロバイダでは想定していないエラーのため, 発生した場合は弊社までお問い合わせください.
0x80110081	[WDT-PRO のみ] コマンドエラー (モードエラー) が通知されました. 要求コマンドのモードが間違っています. 本プロバイダでは想定していないエラーのため, 発生した場合は弊社までお問い合わせください.
0x80110082	[WDT-PRO のみ] コマンドエラー (データエラー) が通知されました. 要求コマンドで指定したデータが設定範囲外もしくは不正な値です. 正しい値をセットの上, 要求をおこなってください.
0x80110083	[WDT-PRO のみ] コマンドエラー (接続ユニットエラー) が通知されました. WDT ユニット, WDB ユニット間で通信異常が発生している可能性があります. 電源を再投入して, 復旧するかどうか確認ください. 復旧しない場合は, 初期化を実施していただき, それでも復旧しない場合は, パトライト社までお問合せください (ハードウェア故障の可能性あります).
0x80110084	[WDT-PRO のみ] コマンドエラー (無線モジュール応答エラー) が通知されました. 電源を再投入して, 復旧するかどうか確認ください. 復旧しない場合は, 無線通信モジュールが故障している可能性があります. パトライト社までお問合せください.

0x80110086	[WDT-PRO のみ] コマンドエラー (データ取得エラー) が通知されました。設定変更中実施中に、データ取得がおこなわれた場合に、通知されます。設定変更完了後、データ取得をおこなってください。
0x801100C0	[WDT-PRO のみ] コマンドエラー (初期化異常) が通知されました。初期化に失敗した場合に通知されます。ハードウェアが故障している可能性があります。パトライト社までお問合せください。
0x801100FF	[WDT-PRO] コマンドエラー (例外異常発生) が通知されました。想定外のエラーが発生しています。頻発するようであれば、パトライト社までお問合せください。 [WDT] コマンドエラーが通知されました。未対応コマンドを送信している可能性があります。接続先のシリーズと Series オプションが一致しているか確認してください。

WDR-PRO プロバイダ使用時によく発生するエラーコードを表 4-10 に示します。

表 4-10 よく発生するエラーコード

エラー番号	説明
0x8091274D	Windows ソケットのエラーコードです。他のソフトウェアが WDR(-PRO) に接続している可能性があります。対象のソフトウェアと WDR(-PRO) を切断し、再度接続を試みてください。または、ポートが誤っている可能性があるため、接続先のポート番号を確認してください。

5. サンプルコード(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. 通信プロトコルコマンド対応表

本プロバイダで実装している機能とパトライト社の『ワイヤレス・データ通信システム WD PRO シリーズ向け アプリケーションノート』の通信コマンドとの対応表を表 6-1 に示します。

表 6-1 通信コマンド対応表

メソッド/機能/変数名	get/put	対象	通信コマンド
AddController	-	WDT-LR-Z2/ WDT-LR-Z2-PRO	無線モジュールリセット要求 WDT 離脱/参加要求
生存確認機能	-	WDT-LR-Z2/ WDT-LR-Z2-PRO	カウント値出力(登録機能付き)要求
	-	WDT-6M-Z2/ WDT-5E-Z2	カウント値出力要求
状態変化時	-	WDT-LR-Z2	信号灯情報通知
		WDT-LR-Z2-PRO	入力情報通知 入力情報(RS-232C)通知
@LIGHT_BUZZER	put	WDT-LR-Z2-PRO	信号灯表示制御要求
@RED	put	WDT-LR-Z2-PRO	信号灯表示制御要求
@YELLOW	put	WDT-LR-Z2-PRO	信号灯表示制御要求
@GREEN	put	WDT-LR-Z2-PRO	信号灯表示制御要求
@BLUE	put	WDT-LR-Z2-PRO	信号灯表示制御要求
@WHITE	put	WDT-LR-Z2-PRO	信号灯表示制御要求
@BUZZER	put	WDT-LR-Z2-PRO	信号灯表示制御要求
@COUNT	put	WDT-LR-Z2/ WDT-LR-Z2-PRO	カウント値出力(登録機能付き)要求
	put	WDT-6M-Z2/ WDT-5E-Z2	カウント値登録要求