

# PATLITE WDR プロバイダ

Version 1.0.0

## ユーザーズ ガイド

August 3, 2022

**【備考】**

\* 本マニュアルの WDR, WDT の画像は, 株式会社パトライト社により提供されたものです.

**【改版履歴】**

バージョン	日付	内容
1.0.0	2019-02-01	初版.
	2020-07-22	動作確認機種, 対応機種を組み合わせを追加.
	2022-08-03	プロバイダ名の誤記修正.

**【対応機器】**

機種	バージョン	注意事項
WDR-L-Z2	1.0.0~	ACアダプタ付属
WDR-LE-Z2	1.0.0~	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

**【動作確認機種】**

機種	バージョン	注意事項
WDR-L-Z2	2.04	

## 目次

1. はじめに .....	4
2. 対応機種を組み合わせ .....	4
3. 環境のセットアップ .....	5
4. プロバイダの概要 .....	8
4.1. 概要 .....	8
4.1.1. WDT の状態遷移 .....	9
4.1.2. 送信モードによる使用方法 .....	10
4.2. メソッド・プロパティ .....	13
4.2.1. CaoWorkspace::AddController メソッド .....	13
4.2.2. CaoController::get_VariableNames プロパティ .....	14
4.2.3. CaoController::AddVariable メソッド .....	15
4.2.4. CaoVariable::get_Value プロパティ .....	16
4.3. エラーコード .....	18
5. サンプルコード(CaoScript).....	19

## 1. はじめに

本書は PATLITE 社製 WDR-L-Z2, WDR-LE-Z2(以下, WDR)から信号灯情報を取得する WDR プロバイダのユーザーズガイドです。WDR 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 プロバイダの環境構成です。PC-WDR 間を Ethernet ケーブルで接続し, WDR - 信号灯間は無線通信を行います。1 台の WDR 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 または WDT-5M-Z2(以下, WDT)を取り付ける必要があります。



図 1-1 WDR プロバイダの環境構成

## 2. 対応機種のご組み合わせ

WDR 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 プロバイダは送信機 WDT-PRO シリーズには対応していません。WDT-PRO シリーズをご使用される場合、後継プロバイダである WDR-PRO プロバイダをご使用ください。

表 2-1 WDR プロバイダの対応表

送信機 \ 受信機	WDT シリーズ	WDT-PRO シリーズ
WDR シリーズ	○	×
WDR-PRO シリーズ	○	×

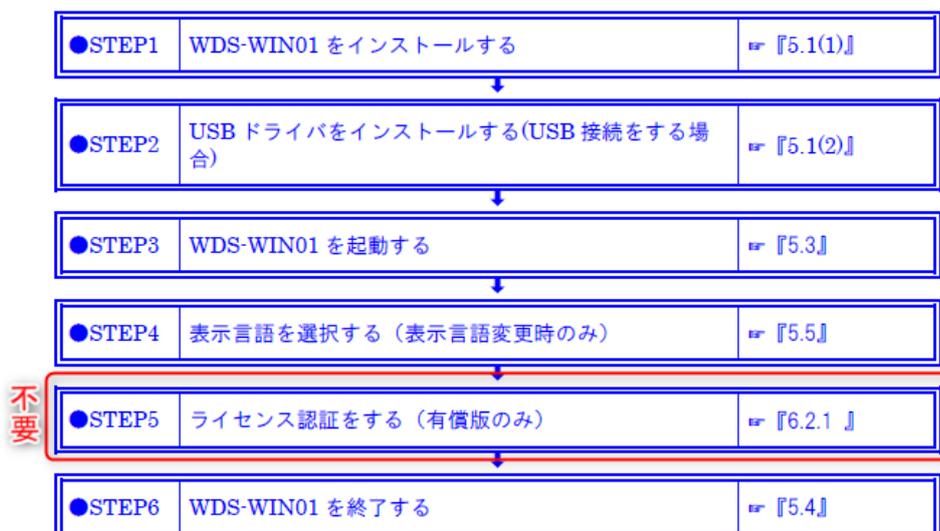
### 3. 環境のセットアップ

WDR プロバイダを使用するためには、WDR および WDT 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」の順に従い、セットアップを完了してください。

WDR 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



1

4.2 章の「STEP5 WDT の設定をする」において、WDR の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 プロバイダと WDR の接続が開始できなくなります。

<sup>1</sup> PATLITE 社「WDS-WIN01 総合取扱説明書」より



また、WDT の送信モードは「変化時送信」、 「応答送信」どちらでも対応可能です。詳細については、本プロバイダマニュアルの 4.1.2 章を参照ください。



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.

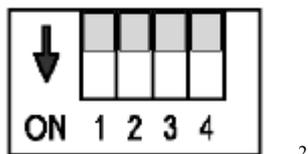


図 3-1 設定スイッチ

<sup>2</sup> 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 および WDT の対応図です。CaoProvController は WDR に、CaoProvVariable は WDT に対応しています。1 台の WDR に接続可能な CaoProvController は 1 つまでです。CaoProvVariable には信号灯の状態を格納する Signal 変数と、信号灯状態の時刻を格納する TimeStamp 変数があります。TimeStamp 変数の作成は任意ですが、Signal 変数は必ず作成してください。TimeStamp 変数のみを作成した場合、4.1.2 章で後述している、WDR プロバイダが一定間隔で信号灯情報を要求しないため、TimeStamp 変数の時刻が更新されません。

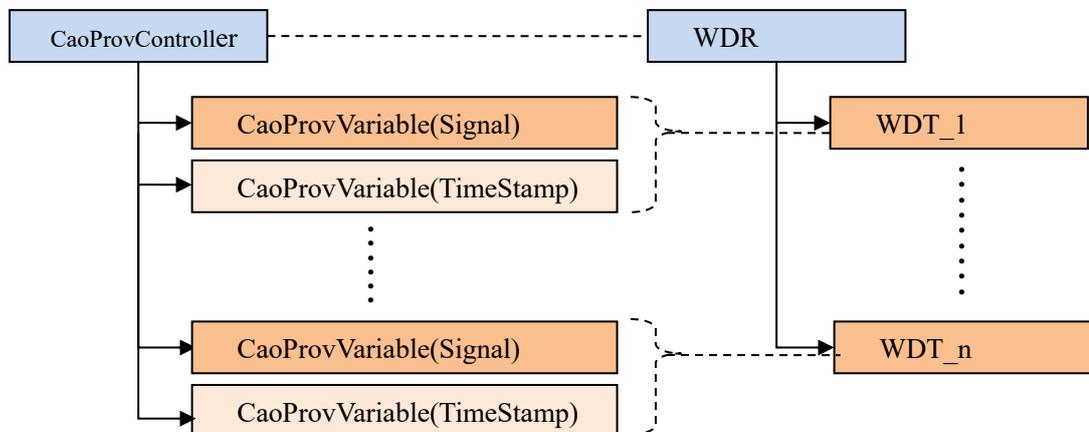


図 4-1 プロバイダの各クラスと WDR の対応図

Signal 変数, TimeStamp 変数を作成するには、WDT の Mac アドレスを使用します。WDT の Mac アドレスは機器のラベルに記述されています(図 4-2)。

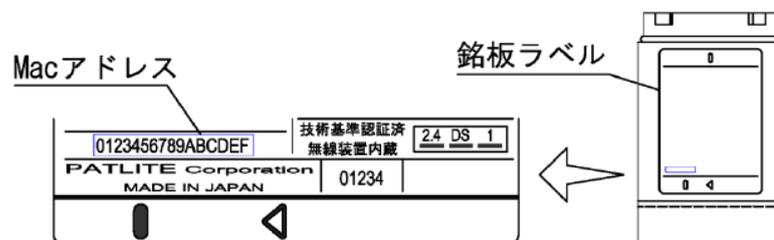


図 4-2 WDT の Mac アドレスの印字箇所<sup>3</sup>

<sup>3</sup> PATLITE 社「WDT-5LR-Z2, WDT-6LR-Z2, WDR-L(E)-Z2 総合取扱説明書」より

WDR 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 プロバイダ

ファイル名	CaoProvPatliteWDR.dll
ProgID	CaoProv.PATLITE.WDR
レジストリ登録 <sup>4</sup>	regsvr32 CaoProvPatliteWDR.dll
レジストリ登録の抹消	regsvr32 /u CaoProvPatliteWDR.dll

#### 4.1.1. WDT の状態遷移

WDT は未接続、接続中、離脱中の 3 つに状態遷移します。WDT が WDR 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 から信号灯情報を取得することができません。WDT がネットワークに参加すると、接続中状態となり、WDT の信号灯情報を取得することができます(①)。接続中の WDT と WDR が通信不能になると、本プロバイダは WDT 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 が異常状態から復帰し、再度 WDR 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 の状態は Signal 変数の値で判断します。未接続状態の場合「-1」、離脱中の場合「-2」を返します。接続中の状態は 4.2.4 章を参照ください。

4.2.1 章の AddController を実行すると、WDR 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 は未接続状態となります。未接続状態から WDT が接続状態になるまで、最大 100 秒かかる場合があります。

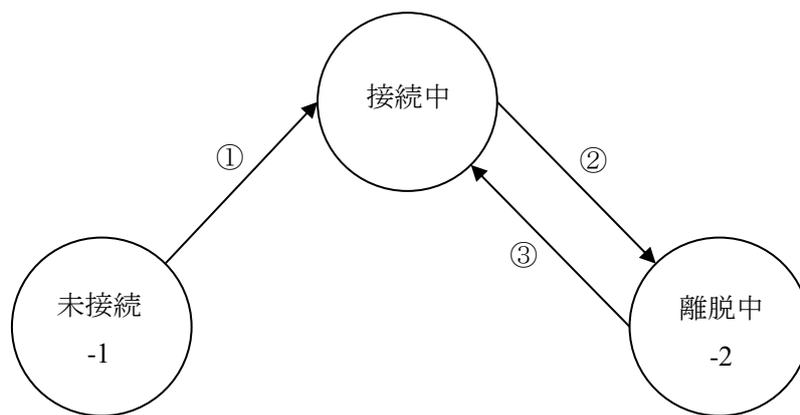


図 4-3 WDT の状態遷移図

<sup>4</sup> 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。

#### 4.1.2. 送信モードによる使用方法

WDR プロバイダは WDT の送信モードによって 2 通りの使用方法があります。信号灯の最新の状態を知りたい場合は「変化時送信モード」を、指定した周期でのみ信号情報を監視したい場合は「応答送信モード」を選択してください。

##### ○WDT が変化時送信モードの場合

図 4-4 は WDT が変化時送信モードのときの WDR プロバイダの使用法を示します。変化時送信モードでは、信号灯情報が変化すると、WDT が WDR 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 が PC に対して受信データを送信し、WDR 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 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を受信時刻を取得して表示します③。

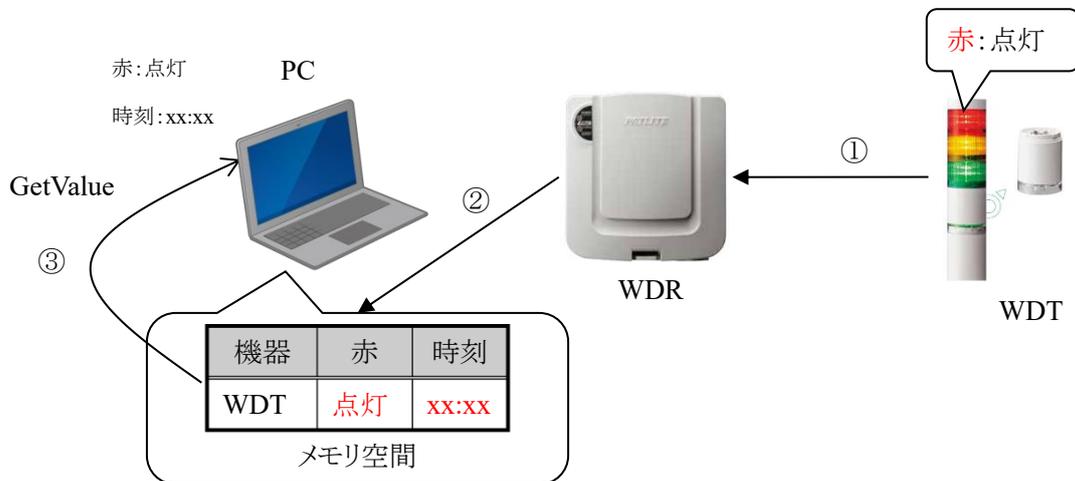


図 4-4 変化時送信モードの WDR プロバイダ使用方法

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 が切断されても認識できません。そのため、WDT が正常状態かどうか監視するために、WDR は定期的に WDT と通信しています。図 4-5 は WDR-WDT 間の生存確認機能を示します。WDR と WDT の通信が 80 秒間途絶えたとき①、WDR はネットワークから離脱したと判断して WDR プロバイダに通知します②。



図 4-5 WDR-WDT 間の生存確認機能

WDR-WDT 間の生存確認機能を使用することで WDT が正常状態か監視することができますが、より早いタイミングで WDT の異常状態を知りたい場合、WDR プロバイダの生存確認機能を使用します。図 4-6 は WDR プロバイダ-WDT 間の生存確認機能を示します。WDR プロバイダは WDT が正常状態かどうか監視するために、WDR を介して定期的に WDT と生存確認要求を行います(①)。WDT が要求を正常に受け取ると応答を返し、メモリ空間の受信時刻を更新します(②)。生存確認要求をしてから 17 秒間応答がない場合は、要求が失敗したと見なします。失敗が規定回数に達したとき、WDT 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。失敗の規定回数は AddController 時の FailureCount 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 AddController 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。

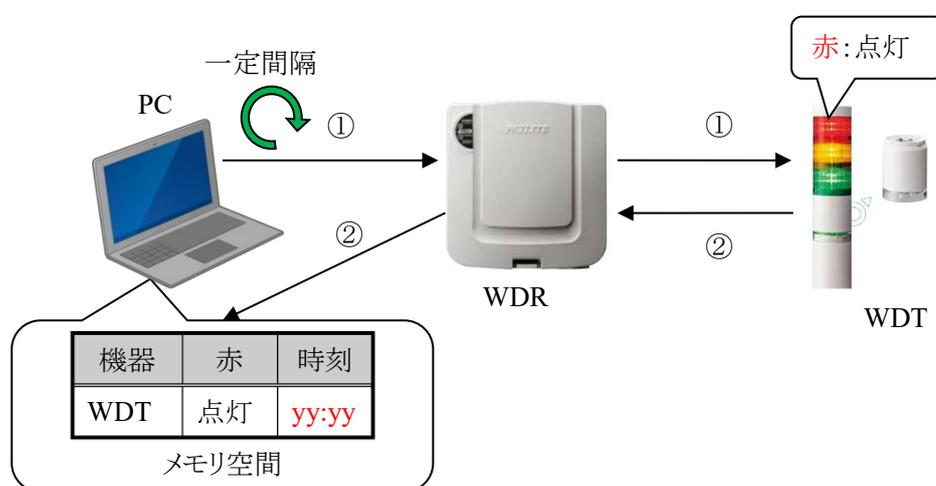


図 4-6 WDR プロバイダ-WDT 間の生存確認機能

### ○WDT が応答送信モードの場合

図 4-7 は WDT が応答送信モードの場合の WDR プロバイダの使用方法を示します。応答送信モードでは、変化時に信号灯情報が送信されません。そのため、WDR プロバイダは WDR を介して定期的に WDT の信号情報を要求します①。WDT が要求を受け取ると、そのときの信号情報を返し、PC 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 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。信号情報を要求をしてから 17 秒間応答がない場合は、要求が失敗したと見なします。失敗が規定回数に達したとき、WDR 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。失敗の規定回数は AddController 時の FailureCount オプションで指定できます。情報を要求する間隔は、AddController 時に Interval オプションで指定できます。応答送信モードで使用する場合、Interval オプションは必ず 500～2147483647 ミリ秒の範囲で指定してください。

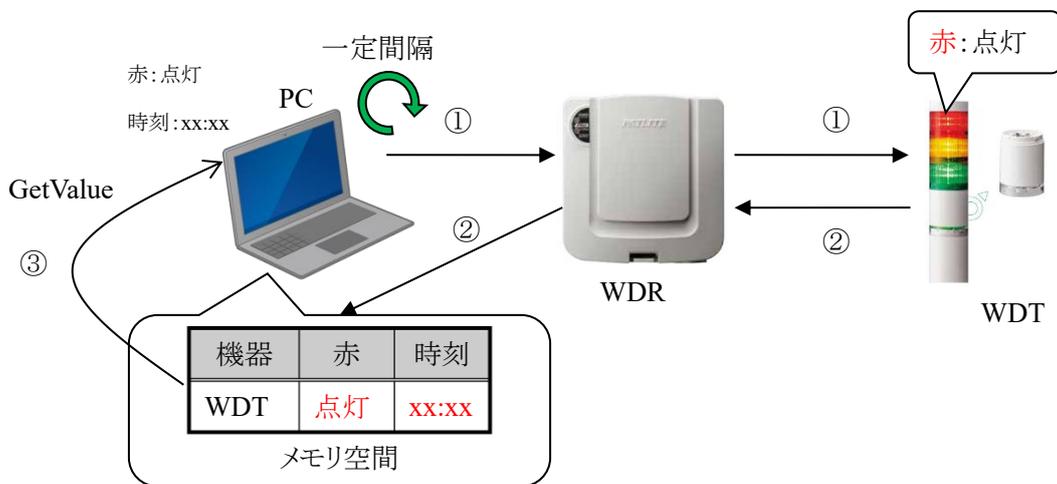


図 4-7 応答送信モードの WDR プロバイダ使用方法

応答送信モードの場合も WDT が正常状態かどうか監視するために、WDR は定期的に WDT と通信しています。図 4-5 は WDR-WDT 間の生存確認機能を示します。WDR と WDT の通信が 80 秒間途絶えたとき①、WDT はネットワークから離脱したと判断して WDR プロバイダに通知します②。WDR プロバイダの方が先に WDT がネットワークから離脱したと判断した場合、この通知は行われません。



図 4-8 WDR-WDT 間の生存確認機能

## 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の接続を行います。以下に AddController の仕様を示します。3 章で設定した WDR の IP アドレスの情報が必要となります。AddController 実行直後、すべての WDT は未接続状態になります。

#### 書式

```
AddController(
    BSTR bstrCtrlName,           // [in] コントローラ名
    BSTR bstrProvName,         // [in] プロバイダ名
    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
);
```

引数	説明			
bstrCtrlName	任意のコントローラ名を入力してください。 例) “Ctrl1”			
bstrProvName	”CaoProv.PATLITE.WDR”を入力してください。			
bstrPcName	プロバイダの実行マシン名を入力します。本プロバイダがローカルにある場合は、空白を指定してください。			
bstrOption	接続先と WDT 生存確認信号送信間隔が指定できます。			
	設定項目	オプション文字列	必須	説明
	接続先	Conn	○	接続するデバイスの IP アドレスとポート番号などを指定します。以下の書式で入力します。中括弧 (“[]”)内は省略可能なパラメータです。オプションを指定しなかったときの接続先ポートは、10001 がデフォルト値となります。  Eth: <接続先 IP>[:<接続先ポート>[:<ローカル IP>[:<ローカルポート>]]”
WDT 信号情報要求間隔 (WDT 生存確認要求間隔)	Interval	-	WDT のモードが変化時送信モードの場合は生存確認要求間隔、応答送信モードの場合は信号情報要求間隔になります。送信間隔をミリ秒単位で 500 ~ 2147483647 ミリ秒の範囲で指定してください。	

			い. -1 を指定した場合, 本機能は実行されません. 値を指定しない場合, 20000 ミリ秒がデフォルト値となります. <b>注: 応答送信モードの場合は-1 を指定せず, 必ず 500 ~ 2147483647 ミリ秒の範囲で指定してください.</b>	
	送信要求の 失敗回数	FailureCount	-	異常状態の WDT を離脱中と判断するための, 送信要求の失敗回数を指定します. 1 ~ 2147483647 回の範囲で指定してください. 指定しない場合, 3 回がデフォルト値となります.
<p>例) “Conn = Eth:192.168.0.1:10001”</p> <p>“Conn = Eth:192.168.0.1:10001, Interval = 1000”</p> <p>“Conn = Eth:192.168.0.1:10001, Interval = 1000, FailureCount = 1”</p>				

**使用例(CaoScript)**

Dim ctrl

Set ctrl = cao.AddController("ctrl1", "CaoProv.PATLITE.WDR", "",

"Conn = Eth:192.168.0.1:10001, Interval = 1000, FailureCount = 1")

**4.2.2. CaoController::get\_VariableNames プロパティ**

WDR と接続済みの WDT の MAC アドレスを取得します. AddController メソッドを実行後, WDT が WDR に接続するまで最大 100 秒かかります. 接続済みの WDT がない場合, Empty 型を返します.

**書式**

```
get_VariableNames(
    BSTR bstrOption           // [in] オプション文字列
);
```

引数	説明
bstrOption	使用しません. 空白を指定してください.

戻り値の型	説明
ARRAY   BSTR	WDR と接続済みの WDT の MAC アドレス, "TimeStamp_" + MAC アドレスを BSTR の配列で返します. 接続済みの WDT がない場合, Empty 型を返します.

**使用例(CaoScript)**

```
Dim varNamesList
varNamesList = dat.ToVar(ctrl.VariableNames)
dbg.output dat.ToVar(ctrl.VariableNames)(0)           '⇒表示結果例:743a65ffe4a0d79
dbg.output dat.ToVar(ctrl.VariableNames)(1)           '⇒表示結果例:TimeStamp_743a65ffe4a0d79
```

**4.2.3. CaoController::AddVariable メソッド**

WDT の信号情報を取得する CaoVariable オブジェクトを生成します。

\* TimeStamp 変数を使用する場合の注意点

TimeStamp 変数とともに必ず Signal 変数を作成してください。TimeStamp 変数のみを作成した場合、一定間隔で信号灯情報を要求しないため、日時が更新されません。

**書式**

```
AddVariable (
    BSTR bstrVarName,           // [in] 変数名
    BSTR bstrOption             // [in] オプション文字列
);
```

引数	説明
bstrVarName	<p>WDT の MAC アドレスまたは”TimeStamp_”+MAC アドレスを指定してください。 MAC アドレスを指定した場合、WDT の信号情報を取得する変数(Signal 変数)を生成します。 ”TimeStamp_”+MAC アドレスを指定した場合、WDT の信号情報が更新された日時を取得する変数(TimeStamp 変数)を生成します。get_VariableNames で取得した値を使用することができます。なお、MAC アドレスは大文字小文字を区別しません。</p> <p>例 1) 743a65ffe4a0d79                      MAC アドレス 例 2) TimeStamp_743a65ffe4a0d79        ”TimeStamp_”+MAC アドレス</p>
bstrOption	使用しません。空白を指定してください。

**使用例(CaoScript)**

```
Dim varSignal
Set varSignal = cao.Controllers("ctrl1").AddVariable(varNamesList (0), "")
Dim varTimeStamp
Set varTimeStamp = cao.Controllers("ctrl1").AddVariable(varNamesList (1), "")
```

#### 4.2.4. 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。返り値は Signal 変数と TimeStamp 変数によって異なります。詳しくは返り値の表を参照ください。

##### 書式

get\_Value ();

引数	説明
なし	-

返り値	型	説明																
Signal 変数	ARRAY   I4	<p>配列要素が7つの I4 型を返します。要素は左から「赤, 黄, 緑, 青, 白, ブザー, LQI」を示します。それぞれの値は状態に応じて以下の表の値を示します。</p> <p>*LQI は無線通信の品質を表す値です。0~255 の数値で表されます。</p> <table border="1"> <thead> <tr> <th>状態</th> <th>色情報</th> <th>ブザー情報</th> <th>LQI</th> </tr> </thead> <tbody> <tr> <td>未接続</td> <td>-1</td> <td>-1</td> <td>-1</td> </tr> <tr> <td>接続中</td> <td>0:未使用 1:消灯 2:点灯 4:点滅</td> <td>0:消音 1:鳴響</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> <td>-2</td> <td>-2</td> </tr> </tbody> </table>	状態	色情報	ブザー情報	LQI	未接続	-1	-1	-1	接続中	0:未使用 1:消灯 2:点灯 4:点滅	0:消音 1:鳴響	0 ~ 50:悪 51 ~100:やや悪 101~150:良 151~255:最良	離脱中	-2	-2	-2
状態	色情報	ブザー情報	LQI															
未接続	-1	-1	-1															
接続中	0:未使用 1:消灯 2:点灯 4:点滅	0:消音 1:鳴響	0 ~ 50:悪 51 ~100:やや悪 101~150:良 151~255:最良															
離脱中	-2	-2	-2															
TimeStamp 変数	DATE	<p>DATE 型を返します。TimeStamp 変数は状態に応じて以下の表の値を示します。</p> <table border="1"> <thead> <tr> <th>状態</th> <th>TimeStamp 変数</th> </tr> </thead> <tbody> <tr> <td>未接続</td> <td>1970/1/1 0:00:00</td> </tr> <tr> <td>接続中</td> <td>WDT の信号灯情報を受信した時刻</td> </tr> <tr> <td>離脱中</td> <td>WDT を離脱と判断した時刻</td> </tr> </tbody> </table>	状態	TimeStamp 変数	未接続	1970/1/1 0:00:00	接続中	WDT の信号灯情報を受信した時刻	離脱中	WDT を離脱と判断した時刻								
状態	TimeStamp 変数																	
未接続	1970/1/1 0:00:00																	
接続中	WDT の信号灯情報を受信した時刻																	
離脱中	WDT を離脱と判断した時刻																	

\* WDT を応答送信モードに設定した場合、WDT が接続中に状態遷移してから最初に情報要求されるまでの間に `get_Value` を実行すると、「0, 0, 0, 0, 0, 0, 0」を示します(図 4-9 参照).

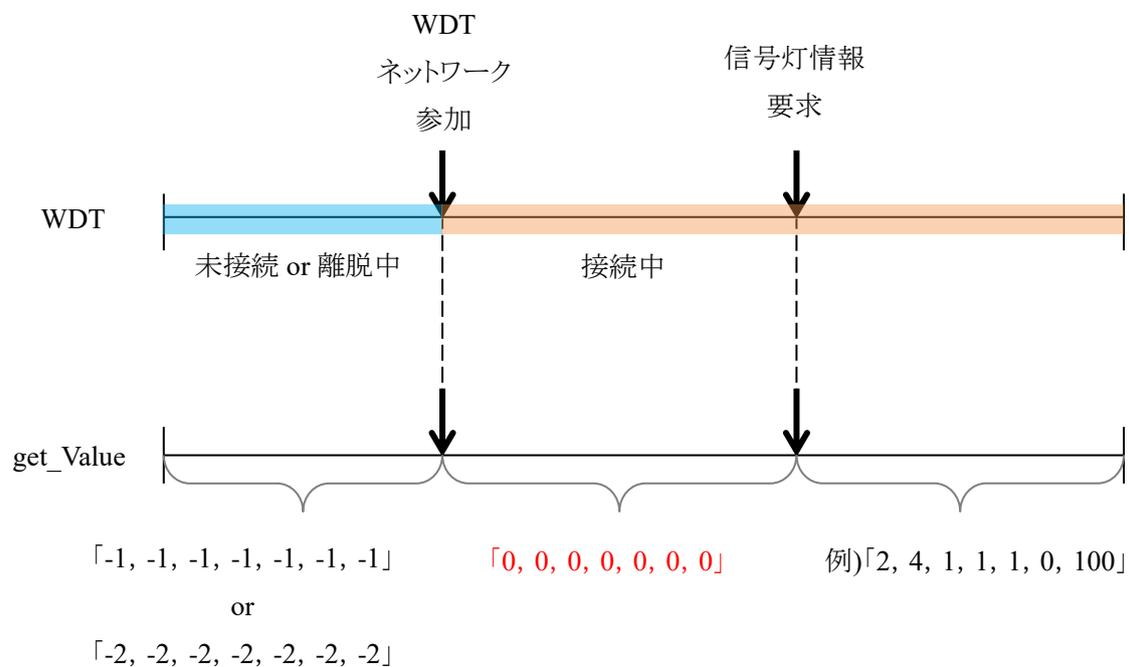


図 4-9 応答送信モードの `get_Value` 結果

#### 使用例(CaoScript)

```
dbg.output DAT.BstrFromVariant(varSignal.Value)
```

```
'⇒表示結果例: -1,-1,-1, -1, -1, -1, -1      未接続中
```

```
'          2, 1, 4, 1, 0, 0, 98      接続中
```

```
'          -2,-2,-2, -2, -2, -2, -2      離脱中
```

```
dbg.output DAT.BstrFromVariant(varTimeStamp.Value)
```

```
'⇒表示結果例: 1/1/2019 0:00:00
```

### 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 4-2 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 4-2 固有エラーコード

エラー番号	説明
0x80110000	オプション文字列 Conn の指定が不正です。
0x80120000	オプション文字列 Interval の指定が不正です。値が適切な範囲になっているか確認してください。
0x80130000	オプション文字列 FailureCount の指定が不正です。値が適切な範囲になっているか確認してください。
0x801400xx	ネットワーク開始で失敗しました。詳細については、下位 2 桁の番号を PATLITE ワイヤレス・データ通信システムアプリケーションノート 0x2D00 ネットワーク開始通知のステータスを参照してください。
0x80150000	WDT 参加要求コマンドを送信したとき、WDR が正しく受信できませんでした。
0x80160000	不適切な受信データが存在しています。WireSharkなどで受信パケットを確認してください。

WDR プロバイダ使用時によく発生するエラーコードを表 4-3 に示します。

表 4-3 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 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 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(CaoScript)

Sub Main

'Create controller

Dim ctrl

```
Set ctrl = cao.AddController("wdr", "CaoProv.PATLITE.WDR", "",  
                             "Conn=Eth:192.168.0.1:10001,Interval=1000, FailureCount = 3")
```

'Wait until a WDT is connected

Dim counter

counter = 0

Dim varNamesList

Do

    Ext.Delay(1000)

    counter = counter+1

    dbg.Output counter & "s"

    varNamesList = dat.ToVar(ctrl.VariableNames)

    If IsEmpty(varNamesList) = false Then

        Exit Do

    End if

Loop

'Create variables

Dim signalVar

Set signalVar = ctrl.AddVariable("743a65fffe4a0d79", "")

Dim timeStampVar

Set timeStampVar = ctrl.AddVariable("TimeStamp\_743a65fffe4a0d79", "")

Ext.Delay(2000)

'Output value

dim signalOutputVal

signalOutputVal = Dat.ChangeType(signalVar.value , 12) 'CaoScript can use only variant type

dbg.Output "Signal:" & Join(signalOutputVal)

dbg.Output "TimeStamp:" & timeStampVar.value

**Delete variables & controller**

Call cao.Controllers("wdr").Variables.Remove(0)

Set signalVariable = Nothing

Call cao.Controllers("wdr").Variables.Remove(1)

Set timeStampSignalVar = Nothing

**Delete controller**

Call cao.Controllers.Remove(0)

Set ctrl = Nothing

End Sub