

ミットヨ  
U-WAVE プロバイダ  
ユーザーズ ガイド

Version 1.0.1

February 2, 2023

備考：

© 2018 DENSO WAVE INCORPORATED

この取扱説明書の著作権は、株式会社デンソーウェーブにあります。

本書に掲載されている会社名や製品は、一般に各社の商標または登録商標です。

仕様は予告なく変更することがあります。

**【改版履歴】**

バージョン	日付	内容
1.0.0	2020-10-26	初版.
	2021-08-04	誤字修正&サンプルコード修正
1.0.1	2022-08-08	内部処理の修正
	2023-02-02	1章にドライバについて追記

**【動作確認機種】**

機種		バージョン	注意事項
U-WAVE-R	02AZD810D	-	
U-WAVE-T	U-WAVE-TM (264-622)	-	
	U-WAVE-TC (264-620)	-	

この取扱説明書の一部または全部を無断で複製・転載することはお断りします。

- この説明書の内容は将来予告なしに変更することがあります。
- 本書の内容については、万全を期して作成いたしましたが、万一ご不審の点や誤り、記載もれなど、お気づきの点がございましたらご連絡ください。
- 運用した結果の影響については、上項にかかわらず責任を負いかねますのでご了承ください。

## 目次

1. はじめに.....	6
2. アプリケーション開発のための環境セットアップ.....	7
2.1. 接続準備.....	7
2.1.1. クライアント PC と U-WAVE-R との接続.....	7
2.1.2. U-WAVEPAK アプリケーションのインストール.....	7
2.1.3. U-WAVE-R と U-WAVE-T のグループ ID, チャネル, バンド ID の設定.....	7
2.1.4. 測定モードの設定.....	9
2.1.5. データ抜け監視レベルの設定.....	9
3. コマンドリファレンス.....	10
3.1. メソッド/プロパティ一覧.....	10
3.2. メソッド・プロパティ.....	11
3.2.1. CaoWorkspace クラス.....	11
3.2.1.1. AddController メソッド.....	11
3.2.2. CaoController クラス.....	12
3.2.2.1. Extensions プロパティ.....	12
3.2.2.2. GetVariableNames メソッド.....	12
3.2.2.3. Variables プロパティ.....	13
3.2.2.4. AddExtension メソッド.....	13
3.2.2.5. AddVariable メソッド.....	14
3.2.2.6. OnMessage イベント.....	14
3.2.3. CaoExtension クラス.....	14
3.2.3.1. GetVariableNames メソッド.....	14
3.2.3.2. Variables プロパティ.....	14
3.2.3.3. AddVariable メソッド.....	15
3.2.4. CaoVariable クラス.....	15
3.2.4.1. Value プロパティ.....	15
3.3. 変数一覧.....	15
3.3.1. CaoController クラス変数.....	15
3.3.1.1. @MAKER_NAME.....	16
3.3.1.2. @VERSION.....	16

3.3.1.3. @RDEVICEINFO .....	16
3.3.2. CaoExtension クラス変数 .....	18
3.3.2.1. @TDEVICEINFO .....	18
3.3.2.2. @MEASUREMENT .....	19
3.4. メッセージ一覧 .....	20
3.4.1.1. パケット解析時エラーメッセージ .....	21
3.4.1.2. ステータスパケット受信メッセージ .....	21
<b>4. プロバイダによるプログラミング .....</b>	<b>24</b>
4.1. 測定データメッセージを取得するサンプルプログラミング .....	24
4.1.1. サンプルプログラム .....	25
<b>5. プロバイダエラーコード .....</b>	<b>27</b>
<b>付録.A 通信パケット対応表 .....</b>	<b>28</b>

## 1. はじめに

本書は、ミットヨの U-WAVE-R に対して接続するプロバイダのユーザーズガイドです。図 1-1 が本プロバイダとデバイスの全体構成図になります。以降本プロバイダを単にプロバイダと呼称します。本プロバイダは「U-WAVEPAK」という設定ソフトに同梱されているドライバが必要になります。プロバイダは USB により U-WAVE-R と接続することで、U-WAVE-R および U-WAVE-T の情報を取得します。プロバイダは U-WAVEPAK ユーザーズマニュアル (MANUAL NO. 99MAL216J8 P/N 02ARB114) を参考にして作成されています。

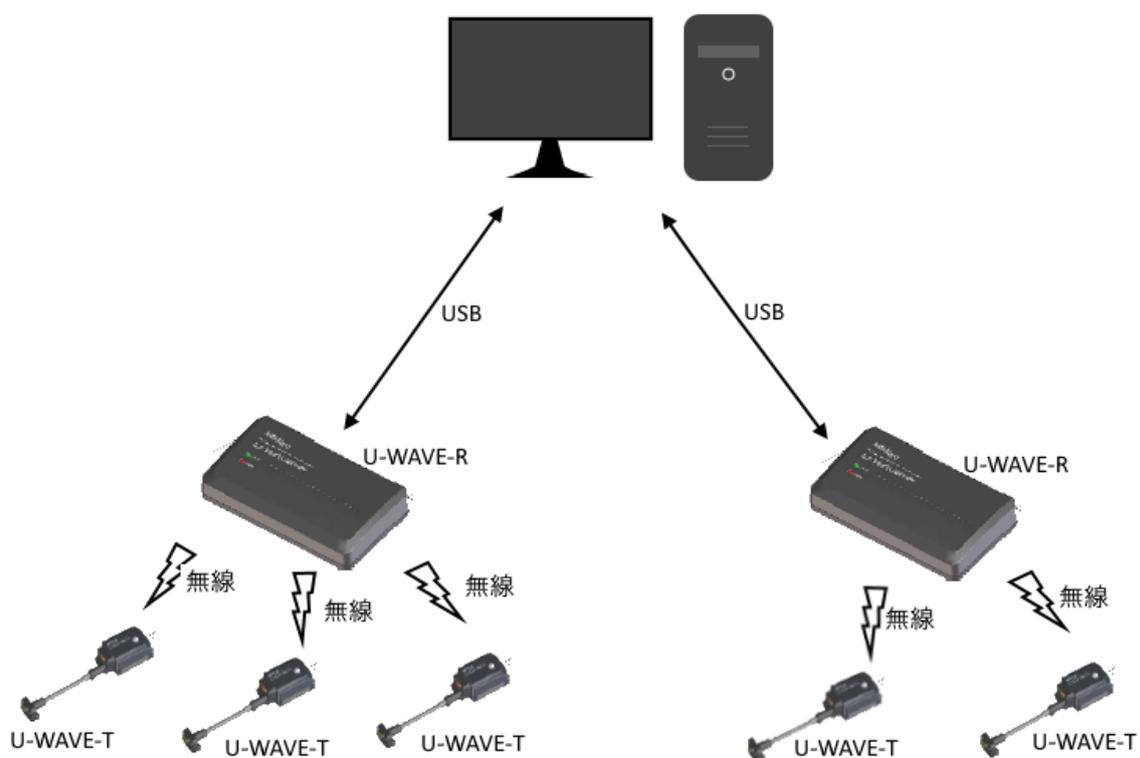


図 1-1 構成図

## 2. アプリケーション開発のための環境セットアップ

### 2.1. 接続準備

クライアント PC と U-WAVE-R との接続手順について記述します。

#### 2.1.1. クライアント PC と U-WAVE-R との接続

クライアント PC と U-WAVE-R を USB ケーブルで接続し、U-WAVEPAK ユーザーズマニュアルを参考にして通信ドライバをインストールします。通信ドライバのインストールには U-WAVEPAK のインストールディスクが必要です。

#### 2.1.2. U-WAVEPAK アプリケーションのインストール

U-WAVEPAK ユーザーズマニュアルを参考にして、U-WAVEPAK アプリケーションをインストールしてください。

#### 2.1.3. U-WAVE-R と U-WAVE-T のグループ ID, チャンネル, バンド ID の設定

U-WAVEPAK アプリケーションを利用して、接続する U-WAVE-R, U-WAVE-T のグループ ID, チャンネル, バンド ID を環境に合わせて設定します。下図はグループ ID とチャンネルの概念図になります。

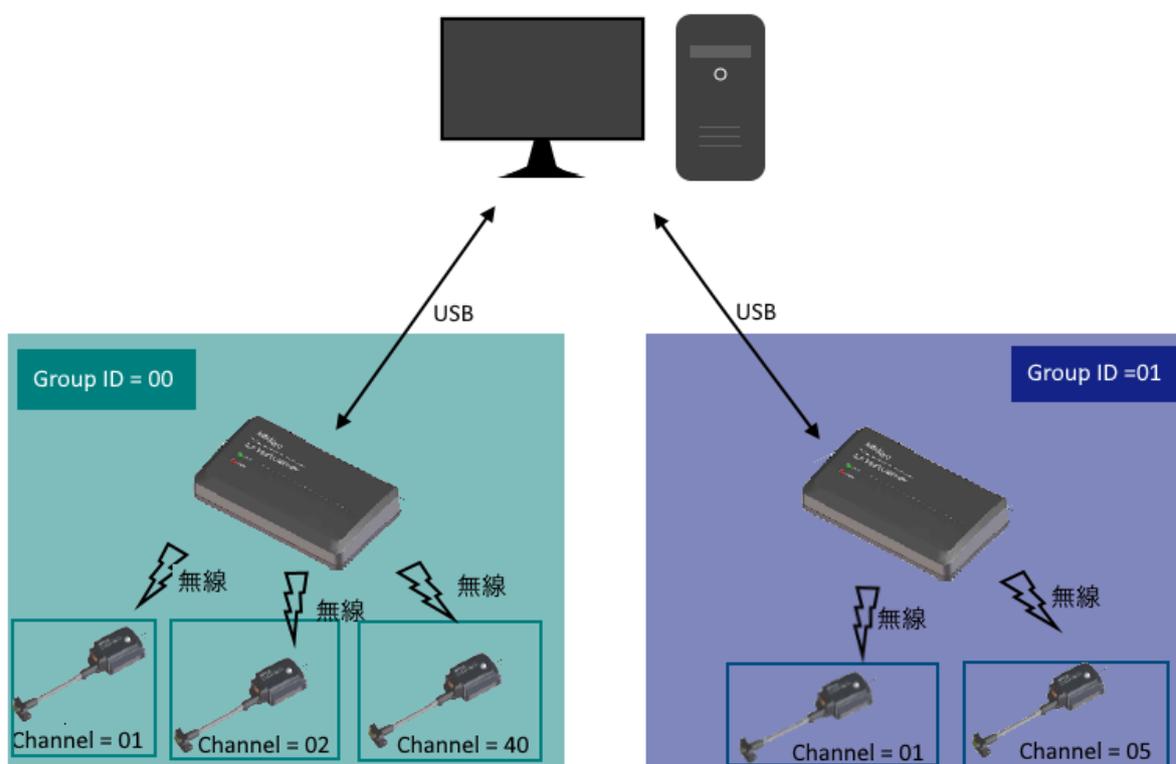


図 2-1 グループ ID とチャンネルの関係

**グループ ID**

U-WAVE-R と U-WAVE-T をグルーピングするための ID です。同じ ID を持つ U-WAVE-R と U-WAVE-T が通信可能です。無線エリア内に複数の U-WAVE-R を使用する場合は、異なる ID を指定する必要があります。ここで設定した値は CaoWorkspace::AddController の GroupId オプションで指定する必要があります。

**チャンネル**

U-WAVE-T に対してユーザが指定するチャンネル番号です。同じ U-WAVE-R に複数の U-WAVE-T を接続する場合、各 U-WAVE-T には異なるチャンネルの値を指定する必要があります。ここで指定した値は CaoController::AddExtension の Channel オプションで指定する必要があります。

**バンド ID**

U-WAVE-R と U-WAVE-T の間の無線通信に割り当てるバンド ID です。同じバンド ID をもつ U-WAVE-R と U-WAVE-T が通信可能です。複数の U-WAVE-R が同じバンド ID を利用しているとお互いの通信を妨害する可能性がありますので異なるバンド ID をご使用ください。

バンド ID	中央周波数(バンド幅)
11	2405 MHz
12	2410 MHz
13	2415 MHz
14	2420 MHz
15	2425 MHz
16	2430 MHz
17	2435 MHz
18	2440 MHz
19	2445 MHz
20	2450 MHz
21	2455 MHz
22	2460 MHz
23	2465 MHz
24	2470 MHz
25	2475 MHz

#### 2.1.4. 測定モードの設定

U-WAVEPAK アプリケーションを利用して、各 U-WAVE-T に対して測定モードを指定します。プロバイダは U-WAVE-R からのデータを受信すると対応するメッセージイベントを発生させます。

#### 2.1.5. データ抜け監視レベルの設定

U-WAVE-T と U-WAVE-R 間の通信では測定データにシーケンス番号を持っています。U-WAVE-R はこのシーケンス番号を監視することでデータ抜けを検出しています。

次の条件を満たした場合に、U-WAVE-R はデータ抜けを検出し、ステータスパケットをプロバイダに送信します。受信したプロバイダはステータスメッセージ受信イベントを発生させます。

$$\text{シーケンス番号の抜け数} > (9 - L)$$

$$L = \text{データ抜け監視レベル}$$

U-WAVEPAK アプリケーションを利用してデータ抜け監視レベルを調整することができます。

### 3. コマンドリファレンス

#### 3.1. メソッド/プロパティ一覧

表 3-1 メソッド/プロパティ一覧

カテゴリ	メソッド/プロパティ <sup>1</sup>	機能	参照
<b>CaoWorkspace</b>			
	AddController	M コントローラに接続	P. 11
<b>CaoController</b>			
	GetExtensionNames	M 接続可能な拡張ボード名リストの取得	P. 12
	Extensions	P コントローラが保持する拡張ボードコレクションの取得	P. 12
	GetVariableNames	M 接続可能な変数名リストの取得	P. 12
	Variables	P コントローラが保持する変数コレクションの取得	P. 13
	AddExtension	M 拡張ボードオブジェクトの追加	P. 13
	AddVariable	M 変数オブジェクトの追加	P. 14
	OnMessage	E メッセージ受信イベント	P. 14
<b>CaoExtension</b>			
	GetVariableNames	P 接続可能な変数名リストの取得	P. 14
	Variables	P タスクが保持する変数コレクションの取得	P. 14
	AddVariable	M 変数オブジェクトの追加	P. 15
<b>CaoVariable</b>			
	Value	P 値の取得/設定	P. 15

<sup>1</sup> M:メソッド, P:プロパティ, E:イベントをそれぞれ示します。

## 3.2. メソッド・プロパティ

### 3.2.1. CaoWorkspace クラス

#### 3.2.1.1. AddController メソッド

CaoWorkspace に、コントローラオブジェクトを追加します。以下に、AddController メソッドの仕様を示します。追加するコントローラオブジェクトはU-WAVE-Rに対応します。

#### 書式

```
AddController
(
    "<コントローラ名>",           // コントローラ名(任意)
    "CaoProv.Mitutoyo.U-WAVE",   // プロバイダ名(固定)
    "<マシン名>",               // プロバイダ実行マシン名(未使用)
    "<オプション>"              // オプション文字列
)
```

#### オプション

以下にオプション文字列に指定するオプションを示します。オプション文字列は下記に示す各オプションをカンマ(,)でつなげた文字列となります。

オプション	必須	説明	値範囲
Conn	○	U-WAVE-R との接続に使用する COM ポート番号を指定します。	3.2.1.1.1 を参照してください。
Group	○	接続U-WAVE-R のグループ ID を指定します。グループ ID の詳細は 2.1.3 を参照してください。	0 - 99
Timeout	--	通信タイムアウト (ms) を指定します。後述する @RDEICEINFO 変数を使用する場合は、値取得に時間がかかるため、11000 以上を指定してください。	1-65535

#### 使用例 (C#)

```
// Engineオブジェクト
ORiN2.ManagedCAO.CCaoEngine engine = new ORiN2.ManagedCAO.CCaoEngine();
// Workspaceオブジェクト
ORiN2.ManagedCAO.CCaoWorkspace workspace = engine.AddWorkspace("NewWrks", "");
// Controllerオブジェクト
ORiN2.ManagedCAO.CCaoController controller =
    workspace.AddController("conn",
        "CaoProv.Mitutoyo.U-WAVE",
        "", "Conn=Com:1, Group=0, Timeout=11000");
```

### 3.2.1.1.1. CONN オプション

以下に Conn オプションの接続パラメータ文字列を示します。ここで中括弧(“[]”)内は省略可能なことを、各パラメータの解説中の下線部はオプションを指定しなかった時のデフォルト値をそれぞれ示します。

```
“Conn=COM:<COM Port>[:<BaudRate>[:<Parity>:<DataBits>:<StopBits>[:<Flow>]]]”
```

<COM Port>	: COM ポート番号. '1' -COM1, '2' - COM2, ...
<BaudRate>	: 通信速度. <u>4800</u> , 9600, 19200, 38400, 57600, 115200
<Parity>	: パリティ. ' <u>N</u> ' -NONE, 'E' -EVEN, 'O' -ODD
<DataBits>	: データビット数. '7' -7bit, ' <u>8</u> ' -8bit
<StopBits>	: ストップビット数. '1' -1bit, ' <u>2</u> ' -2bit
<Flow>	: フロー制御. ' <u>0</u> ' -None, '1' -Xon/Xoff, '2' -ハードウェア制御 OR をとって指定できます。

## 3.2.2. CaoController クラス

### 3.2.2.1. Extensions プロパティ

コントローラが保持する、拡張ボードコレクションを取得します。

#### 使用例 (C#)

```
// 拡張ボードコレクション取得
ORiN2.ManagedCA0.CCaoExtensions extensions = controller.Extensions;
// 拡張ボード取得
ORiN2.ManagedCA0.CCaoExtension extension = extensions[0];
```

### 3.2.2.2. GetVariableNames メソッド

接続可能な変数名リストを取得します。本プロパティで取得した変数名は後述する

3.2.2.5AddVariable メソッドの第一引数に使用することができます。

#### 書式

#### GetVariableNames

```
(
  “<オプション>” // オプション文字列(未使用)
)
```

#### 使用例 (C#)

```
// 変数名リスト取得
string[] variableNames = controller.GetVariableNames(“”);
```

### 3.2.2.3. Variables プロパティ

コントローラが保持する、変数コレクションを取得します。

#### 使用例 (C#)

```
// 変数コレクション取得
ORiN2.ManagedCA0.CCaoVariables variables = controller.Variables;
// 変数取得
ORiN2.ManagedCA0.CCaoVariable variable = variables[0];
```

### 3.2.2.4. AddExtension メソッド

接続している U-WAVE-R に接続している U-WAVE-T に対応するオブジェクトを追加します。以下に、AddExtension メソッドの仕様を示します。

#### 書式

##### AddExtension

```
(
    "<拡張ボード名>", // 拡張ボード名(任意)
    "<オプション>"    // オプション文字列
)
```

#### オプション

以下にオプション文字列に指定するオプションを示します。オプション文字列は下記に示す各オプションをカンマ(,) でつなげた文字列となります。

オプション	必須	説明
Channel	○	対応する U-WAVE-T のチャンネル: 0 - 99 チャンネルの詳細は 2.1.3 を参照してください。

#### 使用例 (C#)

```
// 拡張ボードの追加
ORiN2.ManagedCA0.CCaoExtension caoExt = controller.AddExtension("T_00", "channel=0");
```

### 3.2.2.5. AddVariable メソッド

CaoController に変数オブジェクトを追加します。変数名には 3.3.1 で示す変数名のみ使用できます。

以下に、AddVariable の仕様を示します。

#### 書式

##### AddVariable

```
(  
    "<変数名>",           // 変数名  
    "<オプション>"       // オプション文字列(省略可能)  
)
```

### 3.2.2.6. OnMessage イベント

コントローラのエラー通知や状態の変化を OnMessage イベントとして受け取ることが可能です。受け取れるイベントについては 3.4 を参照してください。

### 3.2.3. CaoExtension クラス

#### 3.2.3.1. GetVariableNames メソッド

#### 書式

##### GetVariableNames

```
(  
    "<オプション>"           // オプション文字列(未使用)  
)
```

#### 使用例(C#)

```
// 変数名リスト取得  
string[] variableNames = caoExt.GetVariableNames("");
```

#### 3.2.3.2. Variables プロパティ

拡張ボードが保持する、変数コレクションを取得します。

#### 使用例(C#)

```
// 変数コレクション取得  
ORiN2.ManagedCA0.CCaoVariables variables = caoExt.Variables;  
// 変数取得  
ORiN2.ManagedCA0.CCaoVariable variable = variables[0];
```

### 3.2.3.3. AddVariable メソッド

CaoExtension に変数オブジェクトを追加します。変数名には 3.3.2 で示す変数名のみ使用できます。

以下に、AddVariable の仕様を示します。

#### 書式

##### AddVariable

```
(
  "<変数名>",           // 変数名
  "<オプション>"       // オプション文字列(省略可能)
)
```

### 3.2.4. CaoVariable クラス

#### 3.2.4.1. Value プロパティ

接続した U-WAVER / U-WAVE-T からデータを取得/設定します。変数名によって動作が異なります。詳細は、3.3. 変数一覧を参照してください。

### 3.3. 変数一覧

各クラスで使用可能な変数一覧を定義します。なお変数は、CaoVariable クラスのオブジェクトを指します。

#### 3.3.1. CaoController クラス変数

変数名	説明	Value		参照
		Get	put	
@MAKER_NAME	メーカー名を取得します。	○	-	P. 16
@VERSION	DLL バージョンを取得します。	○	-	P. 16
@RDEVICEINFO	U-WAVE-R 情報を取得します。	○	-	P. 16

### 3.3.1.1. @MAKER\_NAME

メーカー名の取得をします。

#### データ型

型説明	
VT_BSTR	メーカー名を取得します。

#### 使用例 (C#)

```
// 変数追加
ORiN2.ManagedCAO.CCaoVariable var = controller.AddVariable("@MAKER_NAME", "");
// 値取得
string value = var.Value.ToString();
```

### 3.3.1.2. @VERSION

プロバイダのバージョンの取得をします。

#### データ型

型説明	
VT_BSTR	プロバイダ L のバージョンを取得します。 *. *.*

#### 使用例 (C#)

```
// 変数追加
ORiN2.ManagedCAO.CCaoVariable var = controller.AddVariable("@VERSION", "");
// 値取得
string value = var.Value.ToString();
```

### 3.3.1.3. @RDEVICEINFO

U-WAVE-R 情報を取得します。取得には最低 10 秒はかかるため本変数を使用する場合は AddController の Timeout オプションに十分な時間を指定してください。

#### データ型

型説明		
VT_ARRAY   VT_VARIANT		
0	VT_UI1	グループ ID: 0 - 99 データが設定されていないときは 255 となります。
1	VT_BSTR	デバイス ID: 00000000 - 199999999 データが設定されていないときは FFFFFFFF となります。
2	VT_UI1	バンド ID: 11- 25

型説明		
3	VT_UI1	データ抜け監視レベル: 0 - 9 データ抜け監視レベルについては, 2.1.5 を参照してください.
4	VT_UI1	U-WAVE-R 重複状況 0: 重複する U-WAVE-R は存在しない 1: 重複する U-WAVE-R が存在する 重複が確認された場合は, 無線エリアを分離するか, グループ ID とバンド ID を重複しないものに変更してください.
5	VT_UI1	バンド ID11 のノイズレベル: 0 - 255 値が小さいほど通信状態が良好であることを示します. 255 の場合はノイズレベルが不確定であることを示します.
6	VT_UI1	バンド ID12 のノイズレベル: 0 - 255
:	:	:
18	VT_UI1	バンド ID24 のノイズレベル: 0 - 255
19	VT_UI1	バンド ID25 のノイズレベル: 0 - 255
20	VT_UI1	ダミー: 0 - 255

**使用例 (C#)**

// 変数追加

```
ORiN2.ManagedCA0.CCaoVariable var = controller.AddVariable("@RDEVICEINFO", "");
```

// 値取得

```
object[] rInfos = var.Value as object[];
```

```
Byte? group = rInfos[0] as Byte?;
```

```
String deviceId = rInfos[1] as String;
```

```
Byte? bandId = rInfos[2] as Byte?;
```

```
Byte? watchingLevel = rInfos[3] as Byte?;
```

```
Byte? duplication = rInfos[4] as Byte?;
```

```
Byte?[] noiseLevels = new Byte?[rInfos.Length - 5];
```

```
for (int i = 5, j = 0; i < rInfos.Length; ++i, ++j)
```

```
{
    noiseLevels[j] = rInfos[i] as Byte?;
```

```
}
```

### 3.3.2. CaoExtension クラス変数

変数名	説明	Value		参照
		get	put	
@TDEVICEINFO	U-WAVE-T 情報を取得します.	○	-	P. 18
@MEASUREMENT	直近の測定データを取得します.	○	-	P. 19

#### 3.3.2.1. @TDEVICEINFO

U-WAVE-T 情報を取得します.

##### データ型

型説明																
VT_ARRAY   VT_VARIANT																
0	VT_UI1	U-WAVE-T の状態 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>値</th> <th>意味</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>U-WAVE-T は U-WAVE-R に未登録</td> </tr> <tr> <td>1</td> <td>U-WAVE-T は U-WAVE-R に登録済み(但し未接続)</td> </tr> <tr> <td>2</td> <td>U-WAVE-T は U-WAVE-R に接続済み</td> </tr> <tr> <td>3</td> <td>U-WAVE-T 情報は編集中</td> </tr> <tr> <td>4</td> <td>U-WAVE-T 情報は編集中(変更元のチャンネル)</td> </tr> <tr> <td>5</td> <td>U-WAVE-T 情報は編集中(変更先のチャンネル)</td> </tr> </tbody> </table>	値	意味	0	U-WAVE-T は U-WAVE-R に未登録	1	U-WAVE-T は U-WAVE-R に登録済み(但し未接続)	2	U-WAVE-T は U-WAVE-R に接続済み	3	U-WAVE-T 情報は編集中	4	U-WAVE-T 情報は編集中(変更元のチャンネル)	5	U-WAVE-T 情報は編集中(変更先のチャンネル)
値	意味															
0	U-WAVE-T は U-WAVE-R に未登録															
1	U-WAVE-T は U-WAVE-R に登録済み(但し未接続)															
2	U-WAVE-T は U-WAVE-R に接続済み															
3	U-WAVE-T 情報は編集中															
4	U-WAVE-T 情報は編集中(変更元のチャンネル)															
5	U-WAVE-T 情報は編集中(変更先のチャンネル)															
1	VT_UI1	チャンネル: 0 - 99 データが設定されていないときは 255 となります.														
2	VT_UI1	グループ ID: 0 - 99 データが設定されていないときは 255 となります.														
3	VT_BSTR	U-WAVE-T のデバイス ID: 00000000 - 19999999														
4	VT_UI1	U-WAVE-T のバンド ID: 11 -25														
5	VT_UI1	測定モード 0: ボタン駆動														

##### 使用例 (C#)

```
// 変数追加
ORiN2.ManagedCA0.CCaoVariable var = extension.AddVariable("@TDEVICEINFO", "");
// 値取得
object[] tInfos = var.Value as object[];

Byte? status = tInfos[0] as Byte?;
Byte? channel = tInfos[1] as Byte?;
```

```
Byte? group = tInfos[2] as Byte?;
String deviceId = tInfos[3] as String;
Byte? bandId = tInfos[4] as Byte?;
Byte? measurementMode = tInfos[5] as Byte?;
```

### 3.3.2.2. @MEASUREMENT

U-WAVE-T から送信された直近の測定データを取得します。一度も U-WAVE-T から測定データが送られていない場合は VT\_EMPTY となります。

#### データ型

型説明		
VT_ARRAY   VT_VARIANT		
0	VT_UI1	グループ ID: 0 - 99 データが設定されていないときは 255 となります。
1	VT_UI1	チャンネル: 0 - 99 データが設定されていないときは 255 となります。
2	VT_R8	測定データ
3	VT_BSTR	測定データの単位 M:mm I:Inch 0:無単位

#### 使用例 (C#)

```
// 変数追加
ORiN2.ManagedCAO.CCaoVariable var = extension.AddVariable("@MEASUREMENT", "");
// 値取得
object[] measurement = var.Value as object[];
if (measurement != null)
{
    Byte? group = measurement[0] as Byte?;
    Byte? channel = measurement[1] as Byte?;
    Double? measurementValue = measurement[2] as Double?;
    String unit = measurement[3] as String;
}
```

### 3.4. メッセージ一覧

コントローラのエラー通知や状態の変化を OnMessage イベントとして受け取ることが可能です。

番号	説明
0	<p><b>パケット解析時エラーメッセージ</b></p> <p>Value プロパティのデータ型は 3.4.1.1 を参照してください。</p>
1	<p><b>測定パケット受信メッセージ</b></p> <p>Value プロパティのデータ型は 3.3.2.2 を参照してください。</p>
2	<p><b>ステータスパケット受信メッセージ</b></p> <p>Value プロパティのデータ型については参照してください。</p> <p>@TDEVICEINFO や@RDEVICEINFO の値取得中に、該当するステータスパケットを受信した場合は、@TDEVICEINFO や@RDEVICEINFO の値取得結果のエラーコードとしてステータスコードを 0x8010**** でマスクした値を返します。</p>
3	<p><b>U-WAVE-T 情報パケット受信メッセージ</b></p> <p>Value プロパティのデータ型については 3.3.2.1 を参照してください。</p> <p>@TDEVICEINFO の値取得中に、該当する U-WAVE-T 情報パケットを受信した場合、OnMessage イベントは発生しません。</p>
4	<p><b>U-WAVE-R 情報パケット受信メッセージ</b></p> <p>Value プロパティのデータ型については 3.3.1.3 を参照してください。</p> <p>@RDEVICEINFO の値取得中に、該当する U-WAVE-R 情報パケットを受信した場合、OnMessage イベントは発生しません。</p>

### 3.4.1.1. パケット解析時エラーメッセージ

受信したパケットの解析時に予期せぬエラーが発生した際に送信するメッセージです。このメッセージが発生した場合はプロバイダの動作対象外の機器の可能性があります。

#### データ型

型説明		
VT_ARRAY   VT_VARIANT		
0	VT_I4	プロバイダエラーコード(5 参照)
1	VT_BSTR	受信したパケットをそのまま保持します。

#### 使用例 (C#)

```

/// <summary>
/// エラーメッセージ受信時の処理メソッド例
/// </summary>
/// <param name="errorData">エラーデータ</param>
private void OnReceivedError(object[] errorData)
{
    Int32? errorCode = errorData[0] as Int32?;
    String packet = errorData[1] as String;
}

```

### 3.4.1.2. ステータスパケット受信メッセージ

ステータス(エラー)パケットを受信した際に送信するメッセージです。@TDEVICEINFO や @RDEVICEINFO の値取得中に、該当するステータスパケットを受信した場合は、@TDEVICEINFO や @RDEVICEINFO の値取得結果のエラーコードとして、ステータスコードを 0x8010\*\*\*\*でマスクした値を返すため本メッセージは発生しません。

グループ ID とチャンネルはステータスコードによりデータの意味が異なります。コード毎のグループ ID とチャンネルの意味はステータスコード一覧に記述しています。

#### データ型

型説明		
VT_ARRAY   VT_VARIANT		
0	VT_UI1	グループ ID: 0 - 99 データが設定されていないときは 255 となります。
1	VT_UI1	チャンネル: 0 - 99 データが設定されていないときは 255 となります。
2	VT_BSTR	デバイス ID
3	VT_UI2	ステータスコード

**ステータスコード**

以下にステータスコードの一覧を記述します。説明欄にはその時のグループ ID、チャンネルにどのような値が格納されているかも載せています。

値	メッセージ	検出元	説明
0	U-WAVE-T 電源低下	U-WAVE-T	U-WAVE-T の電池電圧が低下しました。 グループ ID: U-WAVE-T の値 チャンネル: U-WAVE-T の値
1	測定器無応答	U-WAVE-T	測定器が無応答です。 グループ ID: U-WAVE-T の値 チャンネル: U-WAVE-T の値
2	未登録の U-WAVE-T を検出	U-WAVE-R	U-WAVE-R に登録されていない U-WAVE-T を検出しました。 グループ ID: U-WAVE-R の値 チャンネル: 0xFF
3	測定データ抜け	U-WAVE-R	U-WAVE-T から U-WAVE-R に送られる測定データパケットに抜けが発生しました。 グループ ID: U-WAVE-T の値 チャンネル: U-WAVE-T の値
4	U-WAVE-T 未接続	U-WAVE-R	U-WAVE-T が U-WAVE-R に接続されていません。 グループ ID: U-WAVE-T の値 チャンネル: U-WAVE-T の値
5	測定データなし	U-WAVE-R	U-WAVE-R 内の U-WAVE-T の測定データがクリア済みです。 グループ ID: U-WAVE-T の値 チャンネル: U-WAVE-T の値
50	要求パケットエラー	U-WAVE-R	PC からの要求パケットが不正です。 未登録のチャンネルや、異なるグループ ID を指定しています。 グループ ID: 要求パケットの値 チャンネル: 要求パケットの値
51	U-WAVE-T 検索終了	U-WAVE-R	U-WAVE-T を検索しましたが、発見されませんでした。 グループ ID: 要求パケットの値 チャンネル: 要求パケットの値
99	データキャンセル	U-WAVE-T	U-WAVE-T から直前の測定データのキャンセル要求を受信しました。

			グループ ID: U-WAVE-T の値 チャンネル: U-WAVE-T の値
--	--	--	--

**使用例 (C#)**

```

/// <summary>
/// ステータスパケット受信時の処理メソッド例
/// </summary>
/// <param name="errorData">エラーデータ</param>
private void OnReceivedStatus(object[] statusData)
{
    Byte? group = statusData[0] as Byte?;
    Byte? channel = statusData[1] as Byte?;
    String deviceId = statusData[2] as String;
    Byte? statusCode = statusData[3] as Byte?;
}
    
```

## 4. プロバイダによるプログラミング

プロバイダでは、以下の手順でクライアント PC と U-WAVE-R を接続することができます。

- CaoEngine の作成
- CaoWorkspace の作成
- CaoController の作成

U-WAVE-R に接続した後は、CaoController の OnMessage イベントを通して測定データを取得することができます。

### 4.1. 測定データメッセージを取得するサンプルプログラミング

ここでは例として U-WAVE-T からの測定データパケットを CaoController の OnMessage イベントを通して取得するサンプルプログラムを示します。表 4-1 にサンプルプログラムの要件を記述しています。

表 4-1 サンプルプログラムの要件

要件	説明
接続先	USB ケーブルで接続する。
	U-WAVE-R のグループ ID は 0 である。
	U-WAVE-R に接続されている U-WAVE-T のチャンネルは 0, 1 である。
処理内容	測定データメッセージを受信する。
	受診した測定データをチャンネルごとに保存する。

#### 4.1.1. サンプルプログラム

以下にサンプルプログラムの全体像を示します。

Sample	ReceivedMeasurement.cs
--------	------------------------

```
using ORiN2.ManagedCAO;
using System;

class Program
{
    static void Main(string[] arg)
    {
        CCaoEngine caoEngine = null;
        CCaoWorkspace caoWorkspace = null;
        CCaoController caoController = null;

        // CaoEngineオブジェクトの生成
        caoEngine = new CCaoEngine();
        // CaoWorkspaceオブジェクトの生成
        caoWorkspace = caoEngine.AddWorkspace("NewWrks", "");
        // CaoControllerオブジェクトの生成
        caoController = caoWorkspace.AddController("uwave0",
            "CaoProv.Mitutoyo.U-WAVE",
            "",
            "Conn=Com:1, Group=0");

        // OnMessageイベントハンドラの登録
        caoController.OnMessage += OnMessage;

        // ESCが押されるまでOnMessageイベントを受信
        while (Console.ReadKey().Key != ConsoleKey.Escape) { }

        // 終了処理
        caoController.OnMessage -= OnMessage;
        caoEngine.Dispose();
        caoEngine = null;
    }
}
```

```
/// <summary>
/// CaoControllerメッセージ受信イベントハンドラ
/// </summary>
/// <param name="sender">送信元</param>
/// <param name="e">メッセージ情報</param>
static private void OnMessage(object sender, ORiN2.ManagedCA0.OnMessageEventArgs e)
{
    switch (e.Message.Number)
    {
        case 0:
            // エラー発生メッセージです。詳細は3.4.1.1を参照してください。
            break;
        case 1:
            // 測定データメッセージ
            {
                object[] data = e.Message.Value as object[];
                Console.WriteLine("Group=" + data[0] + ", Channel=" + data[1] +
                    ", MeasurementValue=" + data[2] + ", Unit=" + data[3]);
            }
            break;
        case 2:
            // ステータスパケット受信メッセージです。詳細は3.4.1.2を参照してください。
            break;
        case 3:
            // U-WAVE-T情報パケット受信メッセージです。詳細は3.3.2.1を参照してください。
            break;
        case 4:
            // U-WAVE-R情報パケット受信メッセージです。詳細は3.3.1.3を参照してください。
            break;
    }
}
}
```

## 5. プロバイダエラーコード

本プロバイダには、0x8011\*\*\*\*でマスクした以下の独自エラーコードが存在します。（表 5-1 独自エラーコード表参照）

ORiN2 の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 5-1 独自エラーコード表

エラー番号	説明
0x80110101	CONNオプションの指定が間違っています。 AddControllerのオプション説明を参照してください。
0x80110102	Timeoutオプションの指定が間違っています。 AddControllerのオプション説明を参照してください。
0x80110103	Groupオプションの指定が間違っています。 AddControllerのオプション説明を参照してください。
0x80110104	Channelオプションの指定が間違っています。 AddExtensionのオプション説明を参照してください。
0x80110202	測定データパケットのデータ解析中にエラーが発生しました。 動作対応機種ではない可能性があります。
0x80110203	ステータスパケットのデータ解析中にエラーが発生しました。 動作対応機種ではない可能性があります。
0x80110204	U-WAVE-R情報パケットのデータ解析中にエラーが発生しました。 動作対応機種ではない可能性があります。
0x80110205	U-WAVE-T情報パケットのデータ解析中にエラーが発生しました。 動作対応機種ではない可能性があります。
0x80110206	不明なパケットを受信しました。 動作対応機種ではない可能性があります。
0x8011FFFF	プロバイダの内部エラーです。 弊社までお問い合わせください。

また、本プロバイダは、ステータスパケットのステータスコードコードを「0x8010\*\*\*\*」でマスクして返します。ステータスコードについてはユーザーマニュアルを参照してください。

## 付録.A 通信パケット対応表

以下に変数/メッセージとの通信パケットの対応を示します。

### 変数

変数	通信パケット
@RDEVICEINFO	正常時: U-WAVE-R情報パケット
	異常時: ステータスパケット
@TDEVICEINFO	正常時: T-WAVE-R情報パケット
	異常時: ステータスパケット
@MEASUREMENT	測定データパケット

### メッセージ

変数	通信パケット
測定パケット受信メッセージ	測定データパケット
ステータスパケット受信メッセージ	ステータスパケット
U-WAVE-T 情報パケット受信メッセージ	U-WAVE-T情報パケット
U-WAVE-R 情報パケット受信メッセージ	U-WAVE-R情報パケット