Mitutoyo
U-WAVE providers
User's Guide

Version 1.0.1

February 2, 2023

NOTE:

[Revision History]

Version	Date	Description
1.0.0	2020-10-26	First edition
	2021-08-04	Typo correction & sample code correction
1.0.1	2022-08-08	Correction of internal processing
	2023-02-02	Added driver information to chapter 1

[Operation check model]

Model		Version	Notes
U-WAVE-R	02AZD810D	-	
U-WAVE-T	U-WAVE-TM(264-622)	-	
	U-WAVE-TC(264-620)	-	

Contents

1. Introduction 6
2. Setting Up Your Environment for Application Development 7
2.1. Preparing for Connection 7
2.1.1. Connecting U-WAVE-R to a Client-PC 7
2.1.2. Installing U-WAVEPAK Application 7
2.1.3. Configuring Group IDs, Channels, and Band IDs for U-WAVE-R and U-WAVE-T 7
2.1.4. Measurement-mode settings 8
2.1.5. Setting the Data Missing Monitoring Level 9
3. Command Reference 10
3.1. Method/Property List 10
3.2. Method properties 10
3.2.1. CaoWorkspace classes 10
3.2.1.1. AddController method 10
3.2.2. CaoController classes 12
3.2.2.1. Extensions Properties 12
3.2.2.2. GetVariableNames method 12
3.2.2.3. Variables Properties 12
3.2.2.4. AddExtension method 12
3.2.2.5. AddVariable method 13
3.2.2.6. OnMessage event 13
3.2.3. CaoExtension classes 13
3.2.3.1. GetVariableNames method 13
3.2.3.2. Variables Properties 14
3.2.3.3. AddVariable method 14
3.2.4. CaoVariable classes 14
3.2.4.1. Value Properties 14
3.3. Variable list 14
3.3.1. CaoController class-variable 14
3.3.1.1. @MAKER_NAME 15
3.3.1.2. @VERSION 15
3.3.1.3. @RDEVICEINFO 16
3.3.2. CaoExtension class-variable 17
3.3.2.1. @TDEVICEINFO 17
3.3.2.2. @MEASUREMENT 18

3.4. List of Messages 19
3.4.1.1. Packet Analysis Error Messages 19
3.4.1.2. Status packet received message 20
4. Programming by provider 22
4.1. Sample Programming to Retrieve Measurement Data Messages 22
4.1.1. Sample program 23
5. Provider error code 25
Appendix A. Communication packet correspondence table 26

1. Introduction

This manual is a user's guide for providers connecting to Mitutoyo's U-WAVE-R. Fig. 1-1 shows the overall configuration of this provider and the device. From now on, the provider is simply referred to as the provider. This provider requires a driver that is included in the setting software "U-WAVEPAK".Providers get U-WAVE-R and U-WAVE-T info by connecting to U-WAVE-R via USBs. The providers have been created with reference to U-WAVEPAK User's Manual (MANUAL NO. 99MAL216J8 P/N 02ARB114).

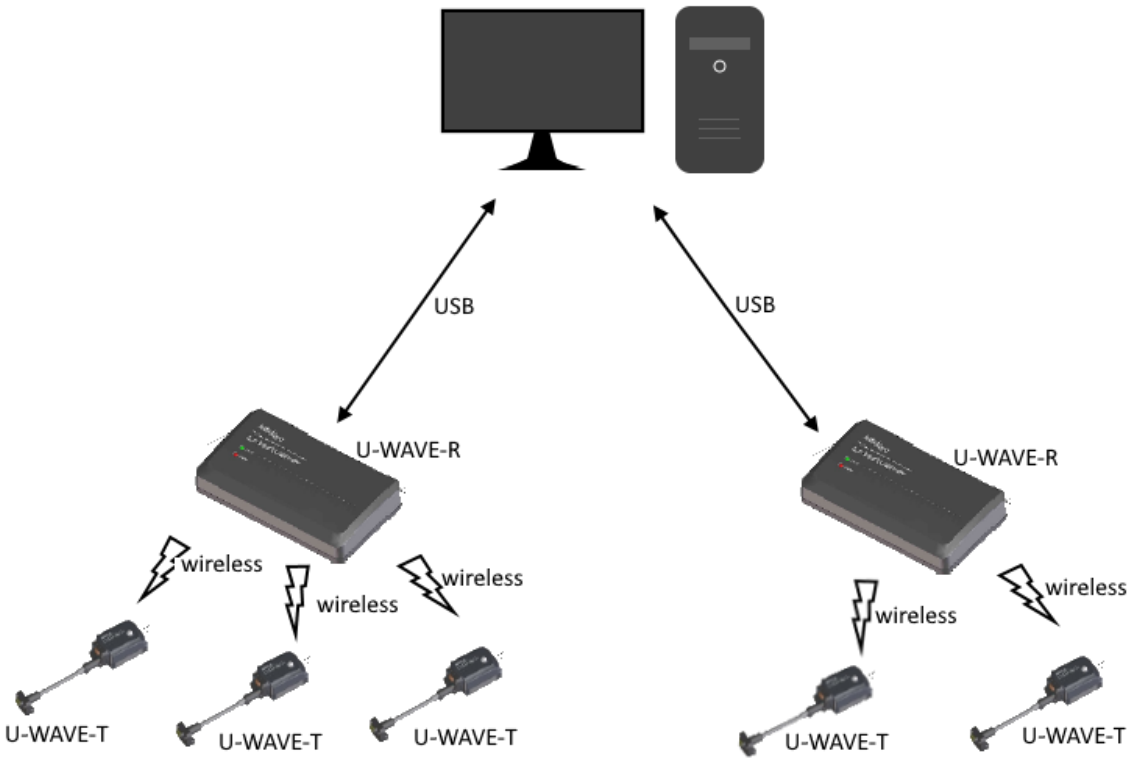

Fig. 1-1 Configuration Diagram

2. Setting Up Your Environment for Application Development

2.1. Preparing for Connection

This section describes how to connect the client PC to U-WAVE-R.

2.1.1. Connecting U-WAVE-R to a Client-PC

Connect U-WAVE-R to the client PC with the USB cable, and install the communication driver referring to U-WAVEPAK User's Manual. U-WAVEPAK install disc is required to install communication drivers.

2.1.2. Installing U-WAVEPAK Application

Install U-WAVEPAK application referring to U-WAVEPAK User's Manual.

2.1.3. Configuring Group IDs, Channels, and Band IDs for U-WAVE-R and U-WAVE-T

Use U-WAVEPAK application to set the group ID, channel, and band ID of U-WAVE-R,U-WAVE-T to be connected according to the environment. The figure below shows the conceptual diagram of the group ID and channel.

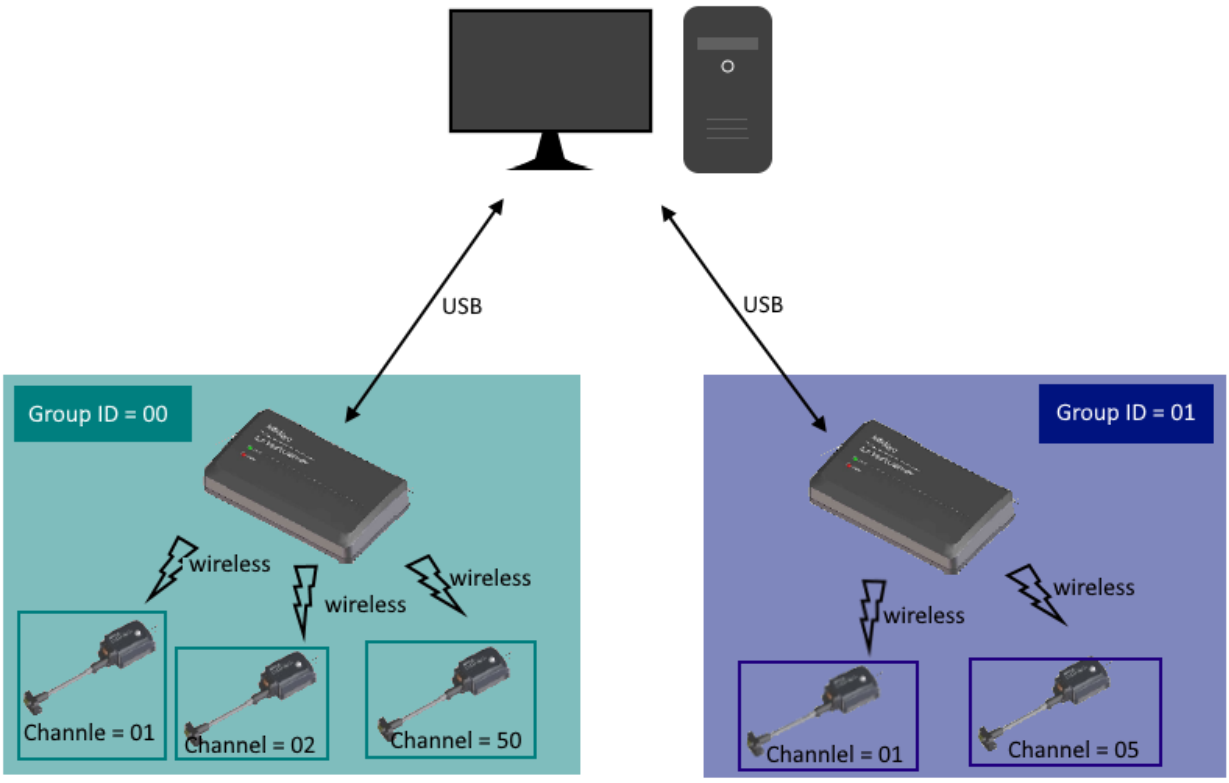

Fig. 2-1 Relationship between group IDs and channels

Group ID

IDs for grouping U-WAVE-R and U-WAVE-T. U-WAVE-R and U-WAVE-T with the same identity can communicate. If you want to use more than one U-WAVE-R within a radio area, you must specify a different ID.

The value set here must be specified in GroupId option of CaoWorkspace::AddController.

-channel

Channel number specified by the user for U-WAVE-T. If you connect more than one U-WAVE-T to the same U-WAVE-R, you must specify different channels for each U-WAVE-T. The value specified here must be specified in Channel option of CaoController::AddExtension.

Band ID

The band identifier assigned to the radio communication between U-WAVE-R and U-WAVE-T. U-WAVE-R and U-WAVE-T with the same band-ID can communicate with each other. If two or more U-WAVE-R use the same band ID, communication with each other may be interrupted. Use a different band ID.

Band ID	Center Frequency (Bandwidth)
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. Measurement-mode settings

Use U-WAVEPAK application to specify the measuring mode for individual U-WAVE-T. When the provider receives data from U-WAVE-R, it generates a corresponding message event.

2.1.5. Setting the Data Missing Monitoring Level

In the communication between U-WAVE-T and U-WAVE-R, the measured data has a sequence number. U-WAVE-R detects data loss by monitoring this sequence number.

U-WAVE-R detects a data loss and sends a status packet to the provider when the following conditions are met: The receiving provider raises a status message reception event.

$$\text{Number of missing sequence numbers} > (9 - L)$$

$$L = \text{Data loss monitoring level}$$

You can use U-WAVEPAK application to adjust the monitoring level of missing data.

3. Command Reference

3.1. Method/Property List

Table 3-1 List of methods and properties

Category	Methods/Properties[1]		Function	See Also
CaoWorkspace				
	AddController	M	Connected to controller	P.10
CaoController				
	GetExtensionNames	M	Obtaining a list of expansion board names that can be connected	P.12
	Extensions	P	Retrieving an Expansion Board Collection Retained by the Controller	P.12
	GetVariableNames	M	Get a list of variable names that can be connected	P.12
	Variables	P	Retrieving Variable Collections Holded by the Controller	P.12
	AddExtension	M	Adding an Expansion Board Object	P.12
	AddVariable	M	Adding Variable Objects	P.13
	OnMessage	E	Message reception event	P.13
CaoExtenstion				
	GetVariableNames	P	Get a list of variable names that can be connected	P.13
	Variables	P	Retrieving Variable Collections Retained by a Task	P.14
	AddVariable	M	Adding Variable Objects	P.14
CaoVariable				
	Value	P	Get/set value	P.14

3.2. Method properties

3.2.1. CaoWorkspace classes

3.2.1.1. AddController method

Add controller objects to CaoWorkspace. The following are the specifics of AddController method: The controller objects that you add correspond to U-WAVE-R.

SYNOPSIS

AddController

(

```
    "<controller name>",            // Controller name (optional)
    "CaoProv.Mitutoyo.U-WAVE",      // Provider name (fixed)
```

[1] M: Indicates methods, P: properties, and E: events, respectively.

```
    "<machine name>",              // Provider execution machine name (unused)
    "<Option>"                     // Option character string
)
```

Option

The following is an optional specification for Option character string: Option character string is a comma (,) string consisting of the options listed below.

Option	Required	Description	Value Range
Conn	✓	Specifies the COM-port number used to connect to U-WAVE-R.	See section 3.2.1.1.1.
Group	✓	Specifies the group-ID of the connected U-WAVE-R. For details on the group ID, see 2.1.3.	0 - 99
Timeout	--	Specify the communication timeout (ms). When using the @RDEICEINFO variable described later, specify 11000 or more because it takes a long time to acquire the value.	1-65535

Usage example(C#)

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

3.2.1.1.1. CONN Optional

The following is a Conn optional connection parameter string: Here, braces ("[]") are optional, and the underlined part in the description of each parameter indicates the default value when no options are specified.

"Conn=COM:<COM Port>[:BaudRate>[:<Parity>:<DataBits>:<StopBits>[:Flow]]]"

<COM Port>	:	COM port number '1' -COM1, '2' - COM2, ...
<BaudRate>	:	Baud rate. 4800, 9600, 19200, 38400, 57600, 115200
<Parity>	:	Parity 'N'-NONE, 'E'-EVEN, 'O'-ODD
<DataBits>	:	Number of data bits '7'-7bit, '8'-8bit
<StopBits>	:	No. of Stop Bits '1'-1bit, '2'-2bit
<Flow>	:	Flow control. '0'-None, '1'-Xon/Xoff, '2'-Hardware control It can be specified by taking OR.

3.2.2. CaoController classes

3.2.2.1. Extensions Properties

Gets the expansion board collection held by the controller.

Usage example(C#)

```
// Expansion Board Collection Acquisition
ORiN2.ManagedCAO.CCaoExtensions extensions = controller.Extensions;
// Expansion board acquisition
ORiN2.ManagedCAO.CCaoExtension extension = extensions[0];
```

3.2.2.2. GetVariableNames method

Gets a list of variable names that can be connected. The variable name obtained by this property can be used as the first argument of AddVariable method described later.AddVariable method

SYNOPSIS

GetVariableNames

```
(
        "<Option>"                  // Option character string (not used)
)
```

Usage example(C#)

```
// Get variable name list
String[] variableNames = controller.GetVariableNames("");
```

3.2.2.3. Variables Properties

Gets a collection of variables that the controller holds.

Usage example(C#)

```
// Variable Collection Retrieval
ORiN2.ManagedCAO.CCaoVariables variables = controller.Variables;
// Variable acquisition
ORiN2.ManagedCAO.CCaoVariable variable = variables[0];
```

3.2.2.4. AddExtension method

Add the corresponding objectto U-WAVE-T connected to the connected U-WAVE-R. The following are the specifics of AddExtension method:

SYNOPSIS

AddExtension

```
(
        "<expansion board name>",   // Expansion Board Name (Optional)
        "<Option>"                  // Option character string
)
```

Option

The following is an optional specification for Option character string: Option character string is a comma (,) string consisting of the options listed below.

Option	Required	Description
Channel	✓	Corresponding U-WAVE-T channels: 0-99 For details of the channel, refer to section 2.1.3.

Usage example(C#)

```
// Adding an expansion board
ORiN2.ManagedCAO.CCaoExtension caoExt = controller.AddExtension("T_00", "channel=0");
```

3.2.2.5. AddVariable method

Adds a variable object to CaoController. Only the variable names shown in 3.3.1 can be used.

AddVariable is specifi ed as follows.

SYNOPSIS

AddVariable

```
(
        "<variable name>",          // Variable name
        "<Option>"                  // Option character string (optional)
)
```

3.2.2.6. OnMessage event

You can receive controller error notifications and status changes as OnMessage events. See 3.4 for the events that can be received.

3.2.3. CaoExtension classes

3.2.3.1. GetVariableNames method

SYNOPSIS

GetVariableNames

```
(
        "<Option>"                  // Option character string (not used)
 )
```

Usage example(C#)

```
// Get variable name list
String[] variableNames = caoExt.GetVariableNames("");
```

3.2.3.2. Variables Properties

Gets the variable collection held by the expansion board.

Usage example(C#)

```
// Variable Collection Retrieval
ORiN2.ManagedCAO.CCaoVariables variables = caoExt.Variables;
// Variable acquisition
ORiN2.ManagedCAO.CCaoVariable variable = variables[0];
```

3.2.3.3. AddVariable method

Adds a variable object to CaoExtension. Only the variable names shown in 3.3.2 can be used.

AddVariable is specifi ed as follows.

SYNOPSIS

AddVariable

```
(
        "<variable name>",          // Variable name
        "<Option>"                  // Option character string (optional)
 )
```

3.2.4. CaoVariable classes

3.2.4.1. Value Properties

Gets/sets the data from the connected U-WAVER/U-WAVE-T. The behavior depends on the variable name. For details, refer to section 3.3, Variable List.

3.3. Variable list

Defines a list of variables that can be used in each class. Variables refer to objects of CaoVariable classes.

3.3.1. CaoController class-variable

Variable name	Description	Value		See Also
		Get	Put	
@MAKER_NAME	Obtain the manufacturer's name.	✓	-	P.15
@VERSION	Get the DLL version.	✓	-	P.15

Variable name	Description	Value		See Also
		Get	Put	
@RDEVICEINFO	Get U-WAVE-R info.	✓	-	P.16

3.3.1.1. @MAKER_NAME

Obtain the manufacturer's name.

Data Type

Type Description	
VT_BSTR	Obtain the manufacturer's name.

Usage example(C#)

```
// Add Variable
ORiN2.ManagedCAO.CCaoVariable var = controller.AddVariable("@MAKER_NAME","");
// Acquisition of Values
String value = var.Value.ToString();
```

3.3.1.2. @VERSION

Get the version of the provider.

Data Type

Type Description	
VT_BSTR	Get the version of provider. *.*.*

Usage example(C#)

```
// Add Variable
ORiN2.ManagedCAO.CCaoVariable var = controller.AddVariable("@VERSION","");
// Acquisition of Values
String value = var.Value.ToString();
```

3.3.1.3. @RDEVICEINFO

Get U-WAVE-R info. Since it takes at least 10 seconds to acquire this variable, specify enough time for Timeout option of AddController when using this variable.

Data Type

Type Description		
VT_ARRAY | VT_VARIANT		
0	VT_UI1	Grouping ID: 0-99 If no data is set, the value is 255.
1	VT_BSTR	Device ID: 00000000-1999999999 If no data is set, an FFFFFFFF occurs.
2	VT_UI1	BandID: 11- 25
3	VT_UI1	Data Missing Monitoring Level: 0 to 9 For the data loss monitoring level, see section 2.1.5.
4	VT_UI1	U-WAVE-R duplication status 0: There are no duplicate U-WAVE-R. 1: There are duplicate U-WAVE-R. If duplicates are detected, separate the wireless area or change the group ID and band ID so that they do not overlap.
5	VT_UI1	Band ID11 noise level: 0-255 The lower the value, the better the communication status. A value of 255 indicates that the noise level is indeterminate.
6	VT_UI1	Band ID12 noise level: 0-255
:	:	:
18	VT_UI1	Band ID24 noise level: 0-255
19	VT_UI1	Band ID25 noise level: 0-255
20	VT_UI1	Dummy: 0 - 255

Usage example(C#)

```
// Add Variable
ORiN2.ManagedCAO.CCaoVariable var = controller.AddVariable("@RDEVICEINFO","");
// Acquisition of Values
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 class-variable

Variable name	Description	Value		See Also
		Get	Put	
@TDEVICEINFO	Get U-WAVE-T info.	✓	-	P.17
@MEASUREMENT	Gets the last measurement data.	✓	-	P.18

3.3.2.1. @TDEVICEINFO

Get U-WAVE-T info.

Data Type

<table>
<tr><th colspan="3">Type Description</th></tr>
<tr><td colspan="2">VT_ARRAY | VT_VARIANT</td><td></td></tr>
<tr><td>0</td><td>VT_UI1</td><td>U-WAVE-T status
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>U-WAVE-T is not registered in U-WAVE-R</td></tr>
<tr><td>1</td><td>U-WAVE-T is registered in U-WAVE-R (not connected)</td></tr>
<tr><td>2</td><td>U-WAVE-T connected to U-WAVE-R</td></tr>
<tr><td>3</td><td>U-WAVE-T data is being edited</td></tr>
<tr><td>4</td><td>U-WAVE-T is being edited (source channels)</td></tr>
<tr><td>5</td><td>U-WAVE-T data is being edited (destination channels)</td></tr>
</table></td></tr>
<tr><td>1</td><td>VT_UI1</td><td>Channel: 0-99
If no data is set, the value is 255.</td></tr>
<tr><td>2</td><td>VT_UI1</td><td>Grouping ID: 0-99
If no data is set, the value is 255.</td></tr>
<tr><td>3</td><td>VT_BSTR</td><td>U-WAVE-T device-ID: 00000000-19999999</td></tr>
<tr><td>4</td><td>VT_UI1</td><td>U-WAVE-T banding ID: 11-25</td></tr>
<tr><td>5</td><td>VT_UI1</td><td>Measurement mode
0: Button drive</td></tr>
</table>

Usage example(C#)

```
// Add Variable
ORiN2.ManagedCAO.CCaoVariable var = extension.AddVariable("@TDEVICEINFO","");
// Acquisition of Values
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

Retrieves the last measured data sent from U-WAVE-T. It is VT_EMPTY if no measured data has been sent from U-WAVE-T at all.

Data Type

Type Description		
VT_ARRAY | VT_VARIANT		
0	VT_UI1	Grouping ID: 0-99 If no data is set, the value is 255.
1	VT_UI1	Channel: 0-99 If no data is set, the value is 255.
2	VT_R8	Measurement data
3	VT_BSTR	Measurement data unit M:mm I:Inch 0: No unit

Usage example(C#)

```
// Add Variable
ORiN2.ManagedCAO.CCaoVariable var = extension.AddVariable("@MEASUREMENT","");
// Acquisition of Values
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. List of Messages

You can receive controller error notifications and status changes as OnMessage events.

No.	Description
0	**Packet Analysis Error Messages** For the datatype of Value property, see 3.4.1.1.
1	**Measurement Packet Receive Message** For the datatype of Value property, see 3.3.2.2.
2	**Status packet received message** For information about the datatypes of Value properties, see. If the corresponding status packet is received while @TDEVICEINFO or @RDEVICEINFO is being acquired, the value masked by 0x8010**** is returned as the error code resulting from obtaining the value of @TDEVICEINFO or @RDEVICEINFO.
3	**U-WAVE-T data packet reception message** See 3.3.2.1 for the datatype of Value property. If the corresponding U-WAVE-T data packet is received while the @TDEVICEINFO is being acquired, no OnMessage occurs.
4	**U-WAVE-R data packet reception message** See 3.3.1.3 for the datatype of Value property. If the corresponding U-WAVE-R data packet is received while the @RDEVICEINFO is being acquired, no OnMessage occurs.

3.4.1.1. Packet Analysis Error Messages

This message is sent when an unexpected error occurs when parsing a received packet. If this message occurs, it may be a device that is not supported by the provider.

Data Type

Type Description		
VT_ARRAY | VT_VARIANT		
0	VT_I4	Provider error code (see 5)
1	VT_BSTR	The received packet is retained.

Usage example(C#)

```
///<summary>
/// Example of processing method when error message is received
///</summary>
///<param name="errorData><erroneous data</ param>
Private void OnReceivedError(object[] errorData)
{
    Int32? errorCode = errorData[0] as Int32?;
```

```
    String packet = errorData[1] as String;
}
```

3.4.1.2. Status packet received message

This message is sent when a status (error) packet is received. If the corresponding status packet is received while @TDEVICEINFO or @RDEVICEINFO is being acquired, this message is not generated because the value masked by 0x8010**** is returned as the error code resulting from @TDEVICEINFO or @RDEVICEINFO acquisition.

The meaning of the data differs depending on the status code for the group ID and channel. The group ID of each code and the meaning of the channel are described in the status code list.

Data Type

Type Description		
VT_ARRAY | VT_VARIANT		
0	VT_UI1	Grouping ID: 0-99 If no data is set, the value is 255.
1	VT_UI1	Channel: 0-99 If no data is set, the value is 255.
2	VT_BSTR	Device ID
3	VT_UI2	Status code

Status code

The following describes the list of status codes. In the explanation column, the group ID at that time and what value is stored in the channel are also shown.

Value	Message	Discovery source	Description
0	U-WAVE-T power down	U-WAVE-T	The batteries in U-WAVE-T have dropped. Group ID: Value of U-WAVE-T Channels: Values for U-WAVE-T
1	Instrument unresponsive	U-WAVE-T	The instrument is not responding. Group ID: Value of U-WAVE-T Channels: Values for U-WAVE-T
2	Unregistered U-WAVE-T detected	U-WAVE-R	A U-WAVE-T that is not registered in U-WAVE-R was detected. Group ID: Value of U-WAVE-R Channel: 0xFF
3	Missing measurement data	U-WAVE-R	Missing measurement data packets sent from U-

			WAVE-T to U-WAVE-R. Group ID: Value of U-WAVE-T Channels: Values for U-WAVE-T
4	U-WAVE-T not connected	U-WAVE-R	U-WAVE-T is not connected to U-WAVE-R. Group ID: Value of U-WAVE-T Channels: Values for U-WAVE-T
5	No measurement data	U-WAVE-R	The measured data of U-WAVE-T in U-WAVE-R has been cleared. Group ID: Value of U-WAVE-T Channels: Values for U-WAVE-T
50	Request packet error	U-WAVE-R	The request packet from the PC is invalid. An unregistered channel or a different group ID has been specified. Group ID: Value of the request packet Channel: Request packet value
51	Finish U-WAVE-T search	U-WAVE-R	U-WAVE-T was searched, but it was not found. Group ID: Value of the request packet Channel: Request packet value
99	Data cancel	U-WAVE-T	A request to cancel the previous measured data was received from U-WAVE-T. Group ID: Value of U-WAVE-T Channels: Values for U-WAVE-T

Usage example(C#)

```
///<summary>
/// Example of Processing Method for Status Packet Reception
///</summary>
///<param name="errorData><erroneous data</ 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. Programming by provider

The provider can connect the client PC to U-WAVE-R by following the steps below.

- Creating a CaoEngine
- Creating a CaoWorkspace
- Creating a CaoController

Once connected to U-WAVE-R, the measured data can be retrieved through OnMessage event of CaoController.

4.1. Sample Programming to Retrieve Measurement Data Messages

This example shows a sample program that retrieves measured data packets from U-WAVE-T through OnMessage event of CaoController. **Table 4-1** describes the requirements of the sample program.

Table 4-1 Sample program requirements

Requirements	Description
Host	Connect using a USB cable
	The group ID of U-WAVE-R is 0.
	U-WAVE-T channels connected to U-WAVE-R are 0,1.
Process Description	A measurement data message is received.
	Save the received measurement data for each channel.

4.1.1. Sample program

The following is an overview of the sample program.

Sample **ReceivedMeasurement.cs**

```
using ORiN2.ManagedCAO;
 using System;

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

        // Create CaoEngine object
        caoEngine = new CCaoEngine();
        // Create CaoWorkspace object
        caoWorkspace = caoEngine.AddWorkspace("NewWrks", "");
        // Create CaoController object
        caoController = caoWorkspace.AddController("uwave0",
                                    "CaoProv.Mitutoyo.U-WAVE",
                                    "",
                                     "Conn=Com:1, Group=0");

        // Register OnMessage Event Handler
        caoController.OnMessage += OnMessage;

        // Receive OnMessage event until ESC is pressed
        while (Console.ReadKey().Key != ConsoleKey.Escape) { }

          // End processing
        caoController.OnMessage -= OnMessage;
        caoEngine.Dispose();
        caoEngine = null;
    }
```

```
        /// <summary>
        /// CaoController Message Receive Event Handler
        /// </summary>
        /// <param name="sender">Sender </param>
        /// <param name="e">Message Infomation</param>
        static private void OnMessage(object sender, OnMessageEventArgs e)
        {
            switch (e.Message.Number)
            {
                case 0:
                    // Packet Analysis Error Messages. See 3.4.1.1 in detail.
                    break;
                case 1:
                    // Measurement Packet Receive Message
                    {
                        object[] data = e.Message.Value as object[];
                        Console.WriteLine("Group=" + data[0] + ",Channel=" + data[1] + ",MeasurementValue=" +
data[2] + ",Unit=" + data[3]);
                    }
                    break;
                case 2:
                        // Status packet received message. See 3.4.1.2 in detail.
                    break;
                case 3:
                        // U-WAVE-T data packet reception message. See 3.3.2.1 in detail.
                    break;
                case 4:
                        // U-WAVE-R data packet reception message. See 3.3.1.3 in detail.
                    break;
            }
        }
    }
```

5. Provider error code

This provider has the following unique error codes masked with the 0x8011****. (Refer to **Table 5-1** Unique Error Codes Table.)

For information about common ORiN2 errors, see the Error Codes section of ORiN2 Programming Guide.

Table 5-1 Unique Error Codes

Error Number	Description
0x80110101	CONN option was specified incorrectly. See the optional explanation for AddController.
0x80110102	Timeout option was specified incorrectly. See the optional explanation for AddController.
0x80110103	Group option was specified incorrectly. See the optional explanation for AddController.
0x80110104	Channel option was specified incorrectly. See the optional explanation for AddExtension.
0x80110202	An error occurred during data analysis of the measurement data packet. Operation may not be supported.
0x80110203	An error occurred during data analysis of the status packet. Operation may not be supported.
0x80110204	An error occurred during data analysis of U-WAVE-R data packet. Operation may not be supported.
0x80110205	An error occurred during data analysis of U-WAVE-T data packet. Operation may not be supported.
0x80110206	An unknown packet was received. Operation may not be supported.
0x8011FFFF	Internal program error. Please contact us.

This provider also masks the status code of the status packet with "0x8010****" and returns it. Refer to the user manual for status codes.

Appendix A. Communication packet correspondence table

The correspondence of communication packets with variables/messages is shown below.

Variable

Variable	Communication packet
@RDEVICEINFO	Normal: U-WAVE-R info-packet
	Error: Status packet
@TDEVICEINFO	Normal: T-WAVE-R info-packet
	Error: Status packet
@MEASUREMENT	Measurement data packet

Message

Variable	Communication packet
Measurement Packet Receive Message	Measurement data packet
Status packet received message	Status packet
U-WAVE-T data packet reception message	U-WAVE-T info-packet
U-WAVE-R data packet reception message	U-WAVE-R info-packet