WACOH
DynPick provider

Version 1.0.1

User's guide

March 27, 2018

Remarks:

DENSO WAVE does not take any responsibility for the mistranslation of this document.

[Revision History]

Version	Date	Content
1.0.0	2013-01-25	First edition.
1.0.1	2013-04-11	Added Error code
1.0.1	2013-04-22	Added hardware
1.0.1	2013-10-10	Added hardware
1.0.1	2017-02-06	Added hardware
1.0.1	2018-03-27	Modified the measurement value obtainment process

[Hardware]

Model	Version	Notes
WEF-3A200-1.5-U		Three-axis force sensor USB type Load rating: Fz 200N, Mx, and My 1.5Nm
WEF-3A200-1.5-R		Three-axis force sensor RS422 type Load rating: Fz 200N, Mx, and My 1.5Nm
WEF-6A100-2-UG5		Six-axis force sensor USB type Load rating: Fx, Fy, Fz 100N, Mx, My, and Mz 2Nm
WEF-6A100-2-RG5		Six-axis force sensor RS422 type Load rating: Fx, Fy, Fz 100N, Mx, My, and Mz 2Nm
WEF-6A200-4-UG5		Six-axis force sensor USB type Load rating: Fx, Fy, Fz 200N, Mx, My, and Mz 4Nm
WEF-6A200-4-RG5		Six-axis force sensor RS422 type Load rating: Fx, Fy, Fz 200N, Mx, My, and Mz 4Nm
WEF-6A1000-30-UG5		Six-axis force sensor USB type Load rating: Fx, Fy, Fz 1000N, Mx, My, and Mz 30Nm
WEF-6A1000-30-RG5		Six-axis force sensor RS422 type Load rating: Fx, Fy, Fz 1000N, Mx, My, and Mz 30Nm
WEF-6A200-4-EGP		Six-axis force sensor Ethernet type Load rating: Fx, Fy, Fz 200N, Mx, My, and Mz 4Nm
WEF-6A200-4-RC5		Six-axis force sensor RS422 type(5V)

		Load rating: Fx, Fy, Fz 200N, Mx, My, and Mz 4Nm
WEF-6A500-10-RC5		Six-axis force sensor RS422 type(5V) Load rating: Fx, Fy, Fz 500N, Mx, My, and Mz 10Nm
WEF-6A1000-30-RC5-B		Six-axis force sensor RS422 type(5V) Load rating: Fx, Fy, Fz 1000N, Mx, My, and Mz 30Nm
WEF-6A200-4-RC5-B		Six-axis force sensor RS422 type (TypeS series)(5V) Load rating: Fx, Fy, Fz 200N, Mx, My, and Mz 4Nm
WEF-6A500-10-RC5-B		Six-axis force sensor RS422 type (TypeS series)(5V) Load rating: Fx, Fy, Fz 500N, Mx, My, and Mz 10Nm
WGF-3A50-5-RG5		Three-axis force sensor RS422 type(5V) Load rating: Fz 50N, Mx, and My 5Nm
WEF-6A200-4-RC24		Six-axis force sensor RS422 type(24V) Load rating: Fx, Fy, Fz 200N, Mx, My, and Mz 4Nm
WEF-6A500-10-RC24		Six-axis force sensor RS422 type(24V) Load rating: Fx, Fy, Fz 500N, Mx, My, and Mz 10Nm
WEF-6A1000-30-RC24-B		Six-axis force sensor RS422 type(24V) Load rating: Fx, Fy, Fz 1000N, Mx, My, and Mz 30Nm
WEF-6A200-4-RC24-B		Six-axis force sensor RS422 type (TypeS series)(24V) Load rating: Fx, Fy, Fz 200N, Mx, My, and Mz 4Nm
WEF-6A500-10-RC24-B		Six-axis force sensor RS422 type (TypeS series)(24V) Load rating: Fx, Fy, Fz 500N, Mx, My, and Mz 10Nm
WGF-3A50-5-RG24		Three-axis force sensor RS422 type(24V) Load rating: Fz 50N, Mx, and My 5Nm

Contents

1. Introduction

This document is a user's guide of the CAO provider for WACOH-TECH force sensor DynPick series. CAO provider (CaoProvDynPick.dll) used in this document is called DynPick provider.

The next chapter describes the overview of DynPick provider and Chapter 3 lists the command reference.

2. Overview of provider

2.1. Overview

DynPick provider is CAO provider that absorbs the WACOH-TECH force sensor-dependent parts and offers CAO provider interface-specified functions.

The file format of the DynPick provider is DLL (Dynamic Link Library), and it is automatically loaded from CAO engine when using. To use DynPick provider, install ORiN2SDK or register it manually with referring to the table below.

Table2-1 DynPick provider

File name	CaoProvDynPick.dll
ProgID	CaoProv.WACOH.DynPick
Registration	regsvr32 CaoProvDynPick.dll
Delete the registration	regsvr32 /u CaoProvDynPick.dll

DynPick provider has the following three operation modes.

・ Standard mode

Obtain the measurement data with CaoVariable::get_Value(). Processing starts once the measurement data is received from DynPick.

・ High-speed mode

Obtain the measurement data with CaoVariable::get_Value(). In this mode, the return value is the data where DynPick has obtained the last time; therefore, if the data obtainment interval is too long, it may fail to obtain data.

・ Cycle mode

Obtain the measurement data with OnMessage event. To specify the obtainment interval, use Interval option of CaoController::AddController(). Note that the data obtainment with CaoVariable::get_value() is not available in this mode.

2.2. Method property

2.2.1. CaoWorkspace::AddController method

The communication is connected in the DynPick provider referring to connected parameter for the communication at AddController. DynPick provider establishes the communication connection with referring to the communication connection parameters at AddController.

In the option elements, specify the type of communication, time-out period, and a measurement interval with OnMessage event.

Format AddController(<bstrCtrlName:BSTR>,<bstrProvName:BSTR>,
<bstrPCName:BSTR>,<bstrOption:BSTR>)

<bstrCtrlName> : [in] Controller name (given name)

<bstrProvName> : [in] Option character string
Fixed value = "CaoProv.WACOH.DynPick

<bstrPcName> [in] Machine name to execute provider

<bstrOption> [in] Option character string

Table 2-2 shows the elements specified in the option character string.

Table2-2 Option character string of CaoWorkspace::AddController

Option	Description
Conn=< connection parameter >	This item is not omittable. Specify the type of communication and the connection parameter.. Please refer to 2.2.1.1 for details.
Mode=< mode >	Specify the operation mode. 0: Standard mode 1: High-speed mode 2: Cycle mode
Timeout=< timeout period >	Specify the timeout period by milliseconds at sending and receiving. (default: 500) The value of this option is disregarded if the high-speed mode (Mode=1) is selected.
Interval=< interval >	Specify an interval of the OnMessage event data obtainment by milliseconds. (default: 1000) The value of this option is disregarded if the Standard mode (Mode=0) or high-speed mode (Mode=1) is selected.

2.2.1.1. Conn option

The following shows the connection parameter character strings of Conn option. Parameters enclosed with square brackets ([]) are omittable. Underlined parts in each parameter’s description show the default values when the options are not specified.

・ **For TCP**

“Conn=TCP:<Dest IP>[:<Dest Port>[:<Src IP>[:<Src Port>]]]"

< Dest IP>: IP address of connection destination. (default: 127.0.0.1)

< Dest Port>: TCP port number of connection destination. (default: 5001)

< Src IP>: IP address of connection source. (default: 255.255.255.255)

When "255.255.255.255" is specified for the IP address of connection source, a local IP address will be specified automatically.

< Src Port>: Port number of connection source. (default: 0)

When "0" is specified for the port number of connection source, available port numbers will be specified automatically

・ **For UDP**

“Conn=UDP:<Dest IP>[:<Dest Port>[:<Src IP>[:<Src Port>]]]"

< Dest IP>: IP address of connection destination. (default: 127.0.0.1)

< Dest Port>: TCP port number of connection destination. (default: 5001)

< Src IP>: IP address of connection source. (default: 255.255.255.255)

When "255.255.255.255" is specified for the IP address of connection source, a local IP address will be specified automatically.

< Src Port>: Port number of connection source. (default: 0)

When "0" is specified for the port number of connection source, available port numbers will be specified automatically.

・ **For RS422**

“Conn=COM:[<ComPort>"[:<BaudRate>[:<Parity>:<DataBits>:<StopBits>]]]"

< ComPort>: COM port number. '1' - COM1, '2' - COM2, ...(default: 1)

< BaudRate>: Transmission rate. (default: 115200)

< ByteSize>: Parity. 'N' - NONE, ‘E' - EVEN, ‘O' - ODD (default: N)

< DataBits>: Number of data bit. '7'- 7 bits, ‘8’- 8 bits. (default: 8)

< StopBits>: Number of stop bit. ‘1'- 1 bit, ‘2'- 2 bits (default: 0)

2.2.2. CaoController::Execute method

Execute commands.

To specify arguments in Execute method, use BSTR for a command and VARIANT array for parameters. For details about commands, refer to 3.1.

Format [<vntRet:VT_VARIANT>=]Execute(<bstrCmd:VT_BSTR>[,<vntParam:VT_VARIANT>])

< vntRet >	:	[out] Return value of command
< bstrCmd >	:	[in] Command
< vntParam >	:	[in] Parameter

2.2.3. CaoController::AddVariable method

Create variable objects.

For about system variables implemented, refer to 2.3.1.

Format AddVariable(<bstrVariableName:VT_BSTR>[,<vntOption:VT_BSTR>])

< bstrVariableName >	:	[in] Variable name
<bstrOption>	:	[in] Option character string

2.2.4. CaoVariable::get_Value property

Obtain variable values.

For details about obtained values, refer to 2.3.

This property is available when Standard mode or High-speed mode is selected.

2.2.5. CaoMessage::get_Value method

Obtain the measurement value stored in the message.

The measurement value is stored in the array in the following order.

< record number >, < Fx >, < Fy >, < Fz >, < Mx >,< My >, < Mz >

The data type is stored with Long type array (VT_I4 | VT_ARRAY).

< Fx> to <Mz > store digital output values.

2.3. Variable list

2.3.1. Controller class

Table2-3 Controller class System variable list

Variable identifier	Data type	Description	Attribute	
			get	put
@Data	VT_I4 | VT_ARRAY	Measurement value Values are stored in the array in the following order. < record number >, < Fx >, < Fy >, < Fz >, < Mx >, < My >, < Mz > < Fx> to <Mz > store the digital output values.	-	-

2.4. Error code

The following specific error code is defined in DynPick provider. For about ORiN2 common errors, refer to the error code-related page in "ORiN2 Programming guide".

Table2-4 Specific error code list

Error name	Error number	Description
E_RECV_DATA_BROKEN	0x80100001	Receive data had been damaged.

3. Command reference

This chapter explains each command of the CaoController::Execute method.

3.1. Controller class

Table3-1 CaoController::Execute command list

Command	Function	
OffsetReset	Offset reset	P.11

3.1.1. CaoController::Execute(“OffsetReset") command

Execute the offset reset of the DynPick series. For details, refer to "DYNPICK manual"..

Format OffsetReset()

Return value : None

Example

```
caoCtrl.Execute(“OffsetReset″)     Execute offset-reset
```