

SINTOKOGIO ZYXer provider

Version 1.0.0

User's guide

November 09, 2020

Remarks:

[Revision history]

Version	Date	Description
1.0.0	4/23/2020	First edition.
	11/9/2020	Removed "@Filter" variable due to ZYXer's specification change.

[Supported model]

Models	Version	Notes
Low load model		Six-axis force torque sensor Rated load : Fx,Fy $\pm 50\text{N}$, Fz $\pm 100\text{N}$, Mx,My,Mz $\pm 0.5\text{Nm}$
Standard model		Six-axis force torque sensor Rated load : Fx,Fy,Fz $\pm 500\text{N}$, Mx,My,Mz $\pm 20\text{Nm}$
Middle load model		Six-axis force torque sensor Rated load : Fx,Fy,Fz $\pm 1000\text{N}$, Mx,My,Mz $\pm 30\text{Nm}$
High load model		Six-axis force torque sensor Rated load : Fx,Fy $\pm 5000\text{N}$, Fz $\pm 10000\text{N}$ Mx,My $\pm 500\text{Nm}$, Mz $\pm 250\text{Nm}$

Table of contents

1. Introduction	4
2. Overview of provider	5
2.1. Overview	5
2.2. Method and property	6
2.2.1. CaoWorkspace::AddController method	6
2.2.1.1. Conn option	7
2.2.2. CaoController::Execute method	8
2.2.3. CaoController::AddVariable method	8
2.2.4. CaoVariable:: get_Value property	8
2.2.5. CaoVariable:: put_Value property	8
2.2.6. CaoMessage::get_Value property	8
2.3. Variable list.....	10
2.3.1. Controller class	10
2.4. Error code	10
3. Command reference	12
3.1. Controller class	12
3.1.1. CaoController::Execute("Stop") command.....	12
3.1.2. CaoController::Execute("ReStart") command.....	12
3.1.3. CaoController::Execute("ImageClear") command	13

1. Introduction

This is a CAO provider user's guide for Force torque sensors manufactured by SINTOKOGIO. Hereafter, the CAO provider (CaoProvZYXer.dll) described in this user's guide is called ZYXer provider.

The next chapter describes the overview of this ZYXer provider, and the chapter 3 lists the command reference.

2. Overview of provider

2.1. Overview

ZYXer provider is a CAO provider that eliminates Force torque sensor-dependent functions and offers CAO provider interface specification-defined functions.

The file format of ZYXer provider is DLL (Dynamic Link Library) and it is automatically loaded from CAO engine when it is used. When using ZYXer provider, you need to install OriN2SDK or, manually register the registration with reference to the following table.

Table 2-1 ZYXer provider

File name	CaoProvZYXer.dll
ProgID	CaoProv.SINTOKOGIO.ZYXer
Registry registration	regsvr32 CaoProvZYXer.dll
Un-registration	regsvr32 /u CaoProvZYXer.dll

ZYXer provider has two types of operation mode depending on the acquisition method of the measurement data of the translational force of each axis (F_x , F_y , F_z), and torque force (M_x , M_y , M_z).

- Normal mode

Acquires the measurement data by `CaoVariable.GetValue()`. Process is suspended until the response data is received from ZYXer.

- Cycle mode

Acquires the measurement data in every cycle time with `OnMessage` event. The cycle time of the measurement data acquisition is specified by `Interval` option of `CaoController::AddController()`.

In this mode, data acquisition by `CaoVariable::GetValue()` is not executable.

2.2.1.1. Conn option

The following shows connection parameter strings of Conn option. Parameters enclosed by square brackets (“[]”) can be omitted. An underlined selection in each parameter is the default value when there is no entry for the parameter.

- **For TCP**

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

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

<Dest Port>: TCP port number of the connection destination. (Default: 10001)

<Dest IP> : IP address of the connection source (Default: 255.255.255.255)

If “255.255.255.255” is entered as connection source IP address, the IP address of the local machine will be automatically set as a source IP address.

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

If “0” is entered as a connection source port number, available port number will be automatically set.

- **For UDP**

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

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

<Dest Port>: TCP port number of the connection destination. (Default: 10001)

<Dest IP> : IP address of the connection source (Default: 255.255.255.255)

If “255.255.255.255” is entered as connection source IP address, the IP address of the local machine will be automatically set as a source IP address.

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

If “0” is entered as a connection source port number, available port number will be automatically set.

- **For RS422**

“Conn=COM:[<ComPort>]’[:<BaudRate>]”

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

<BaudRate> : communication speed (Default: 460800)

Enter the baud rate according to the force torque sensor specification.

For RS422 communication, the following connection parameters are fixed to the default setting values. You cannot change them.

Connection parameter name	Value
Parity	NONE

Data bit count	Bit 8
Stop bit count	Bit 1

2.2.2. CaoController::Execute method

Executes a command.

For the argument of Execute method, specify a command with BSTR, specify a parameter with VARIANT array.

For information about commands, refer to 3.1.

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

`< vntRet >` : [out] Command return value
`< bstrCmd >` : [in] command
`< vntParam >` : [In] Parameter

2.2.3. CaoController::AddVariable method

Generates a variable object.

For about implemented system variables, refer to 2.3.1.

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

`< bstrVariableName >` : [in] Variable Name
`<bstrOption>` : [in] Option string

2.2.4. CaoVariable:: get_Value property

Obtains value of variable.

For details, refer to 2.3.

This property is available only in the normal mode.

2.2.5. CaoVariable:: put_Value property

Sets the value of variable.

For details about value to set, refer to 2.3.

This property is available only in the normal mode.

2.2.6. CaoMessage::get_Value property

Acquires the measurement value stored in the message.

Values are stored in an array by the following order.

`<Sensor status>`,`<Fx>`,`<Fy>`,`<Fz>`,`<Mx>`,`<My>`,`<Mz >`,

Data is stored as long type array data (VT_I4 | VT_ARRAY).

Digital output values are stored in the area from <Fx> to <Mz>.

The data acquisition with this property is available only in the cycle mode.

2.3. Variable list

2.3.1. Controller class

Table 2 Controller class system variable list

Variable name	Data type	Description	Attribute	
			get	put
@Data	VT_I2 VT_ARRAY	Measurement value Values are stored in an array by the following order. <Sensor status> , <Fx> , <Fy> , <Fz> , <Mx> , <My> , <Mz> Digital output values are stored in the area from <Fx> to <Mz>. <Sensor status> stores the sensor status values listed in the SINTOKOGIO Force torque sensor communication specification. For details, please refer to the force torque sensor communication specification provided by SINTOKOGIO.	✓	-
@Version	VT_BST VT_ARRAY	The product information of the force torque sensor is stored in an array by the following order. <Product model>,<Serial number>, <Firmware version>,<Baud rate>	✓	
@RatedValue	VT_R4 VT_ARRAY	The maximum rated values of the force torque sensor are stored in an array by the following order. <Fx>,<Fy>,<Fz>,<Mx>,<My>,<Mz >, For each element (<Fx> to <Mz>), floating point number data (positive value) is stored.	✓	

2.4. Error code

In ZYXer provider, the following original error codes are designated. For about ORiN2 common errors, refer to the error code section of ORiN2 programming guide.

Table 2-3 Original error code list

Error name	Error number	Description
E_RECV_DATA_BROKEN	0x80100001	Received data was damaged.

E_RECV_NAK	0x80100002	Force torque sensor denied the command.
	0x801000XX XX: Response result	Response result is abnormal. The lowest one byte of the error number stores the response result value which is described in the SINTOKOGIO Force torque sensor communication specification. For details, please refer to the force torque sensor communication specification provided by SINTOKOGIO.

3. Command reference

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

3.1. Controller class

Table 3-1 CaoController::Execute command list

Command	Function	
Stop	Stops the data measurement. (available only in the cycle mode)	P. 12
ReStart	Resumes the data measurement being stopped (available only in the cycle mode)	P. 10
BufferClear	Clears reception buffer.	P. 12

3.1.1. CaoController::Execute("Stop") command

Stops the data measurement.

This command is available only in the cycle mode.

Syntax Stop ()

Return value : none

Example

```
caoCtrl.Execute("Stop") 'Stop the data measurement
```

3.1.2. CaoController::Execute("ReStart") command

Resumes the data measurement stopped by Stop command.

This command is available only in the cycle mode.

Syntax ReStart()

Return value : none

Example

```
caoCtrl.Execute("ReStart") 'Resume the data measurement.
```

3.1.3. GaoController::Execute("ImageClear") command

Clears the received buffer and then executes the error clear process.

Syntax BufferClear()

Return value : none

Example

```
caoCtrl.Execute("BufferClear") ' Clear the reception buffer
```
