Dai-ichiSeiko

ESTORQ provider

Version 1.0.0

User's Guide

September 30, 2019

Remark

Revision history

Version	Date	Content
1.0.0	2019-09-30	First edition

Compatible device

Model	Version	Note
ESTORQ		
ES-Gripper		

Operation check device

Model	Version	Note
ESTORQ	4.19	

Table of contents

1. Introduction

This document is a user's guide of CAO providers that reads the torquevalues into ESTORQ and ES-Gripper made by Dai-ichi Seiko Co.,Ltd.. The description of ES-Gripper is omitted below.

The CAO provider (CaoProvDai-ichiSeikoESTORQ. dll) described in this document is called the ESTORQ provider. ESTORQ Provider is developed in accordance with the Electrostatic Capacity-type Torque Sensor ESTORQ Communication Specifications (USB, RS422) provided by Dai-ichi Seiko Co.,Ltd.. This document describes the features of the ESTORQ providers and the methods implemented.

2. Provider Overview

2.1. Overview

Fig. 2-1 shows the correspondence between classes of providers and ESTORQ. CaoProvController corresponds to the ESTORQ itself, and CaoProvVariable corresponds to theTorque and Versioninformation.

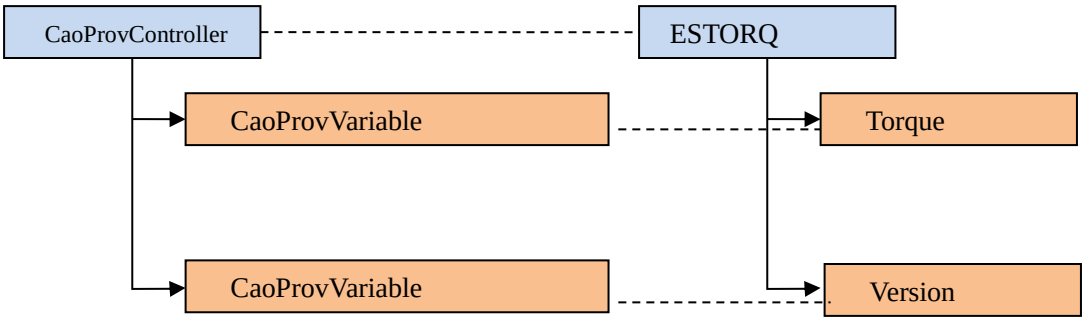

Fig. 2-1 Correspondence Diagram of Classes of Providers and ESTORQ

Fig. 2-2 shows the connection diagram with ESTORQ. Connect to the target ESTORQ with RS-422A. The default baud rate is 307.2 kbps. Please contact Dai-ichi Seiko Co.,Ltd. to change the ESTORQ baud rate to 115.2 kpbs.

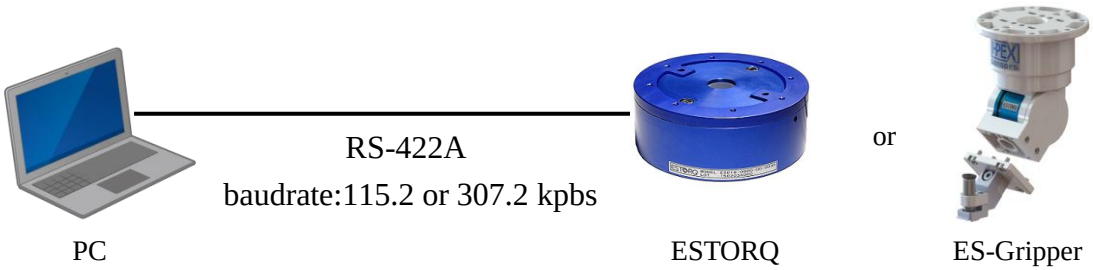

Fig. 2-2 Connecting to the ESTORQ

The file format of the ESTORQ provider is DLL (Dynamic Link Library) and is dynamically loaded when used from the CAOengine.To use the ESTORQ provider, you must either install ORiN2SDK or manually register the registry, as shown in Table 2-1.

Table 2-1 File Formats for ESTORQ Provider

File name	CaoProvDai-ichiSeikoESTORQ.dll
ProgID	CaoProv.Dai-ichiSeiko.ESTORQ
Registry registration[1]	Regsvr32 CaoProvDai-ichiSeikoESTORQ.dll
Unregistering the Registry	Regsvr32 /u CaoProvDai-ichiSeikoESTORQ.dll

[1] You do not need to manually register or delete the software if it is installed by ORiN2 SDK.

2.2. Method Properties

2.2.1. CaoWorkspace::AddController method

In this provider, communication is connected by setting connection parameters by CaoWorkspace ::AddController. The AddController specifications are shown below.

Format

```
AddController(
        BSTR bstrCtrlName,      // [in] Controller name (enter any controller name)
        BSTR bstrProvName,      // [in] The provider name. Fixed = "CaoProv.Dai-ichiSeiko.ESTORQ"
        BSTR bstrPcName,        // [in] Running machine name of the provider
        BSTR bstrOption         // [in] Option string
);
```

Table 2-2 CaoWorkspace::AddController Optional Strings

Setting item	Option String	Required	Description
Destination	Conn	○	Write in the format com:<port number>[:<baud rate>].[2] Port number: 1 to 256 Baud rate[3][bps]: 115200/307200
Rated torque	RatedTorque	○	Specify the rated torques of the connected ESTORQ. 0.0 < rated torque ≦ 100000.0.
Time-out	Timeout	-	Specify a timeout from 1 to 4294967295 ms. Default value: 500 ms

Ex.1) Connecting at a COM1, baud rate of 307200 bps, rated torque10 Nm

"Conn = com:1, RatedTorque = 10"

Ex.2) Connecting at COM256, baudrate of 115200 bps, rated torque of 0.5 Nm, and timeout of 1000 ms

"Conn = com:256:115200, RatedTorque = 0.5, Timeout=1000"

Example (C#)

```
using ORiN2.ManagedCAO;
CCaoEngine eng = new CCaoEngine();
CCaoWorkspace ws = eng.AddWorkspace("SampleWorkspace", "");
CCaoController ctrl = ws.AddController("controller1","CaoProv.Dai-ichiSeiko.ESTORQ",
                                        string.Empty,"Conn=com:1,RatedTorque=10,Timeout=1000");
```

[2] Square brackets ("[]") enclose optional parameters.
[3] The underlined value is the default value.

2.2.2. CaoController::Execute method

Executes a provider-specific command belonging to a CaoController class.The arguments of the Execute method specify the command as a BSTR type and the parameters as a VARIANT type. The parameters are optional.

Format

```
Execute (
        BSTR bstrCmd            // [in] command name
        [,VARIANT vntParam]     // [in] parameter
);
```

Argument	Description
bstrCmd	Specify Execute method names in the GetTorque,GetDeviceVersion,Reset commands.
vntParam	Specifies the parameters to use in the command name.

2.2.2.1. CaoController::Execute("GetTorque") Command

Gets the current Torque value.

Argument type	Description
Without	-

Return Value Type	Description
VT_R8	Current Torque Value [Nm]. Measure in the following range using the rated torque specified in AddController. - (Rated Torque × 1.1) [Nm] ～ +(Rated Torque × 1.1) [Nm]

Example (C#)

```
double torque =    (double)ctrl.Execute("GetTorque", "");
Debug.WriteLine(torque + "[Nm]");              // 0.00134293737028446[Nm]
```

2.2.2.2. CaoController::Execute("GetDeviceVersion") Command

Gets the ESTORQ deviceversion.

Argument type	Description
Without	-

Return Value Type	Description
VT_BSTR	The ESTORQ device version.

Example (C#)

```
string deviceVersion = ctrl.Execute("GetDeviceVersion", "").ToString();
Debug.WriteLine(deviceVersion);     // TRQ Ver 4.19 2018/02/06
```

2.2.2.3. CaoController::Execute("Reset") Command

Set the Torque value at the time this command is executed to 0 Nm. The setting persists after a power cycle.

Argument type	Description
Without	-

Return Value Type	Description
Without	-

Example (C#)

```
ctrl.Execute("Reset", "");
```

2.2.3. CaoController::get_VariableNames Properties

Retrieve lists of variables available to ESTORQ providers. You can obtain the list in Table 2-3.

Return Value Type	Description
VT_ARRAY | VT_BSTR	You can obtain a list of variables that can be used for AddVariable variable names in the CaoController.

Example (C#)

```
string[] variableNmaes = ctrl.GetVariableNames(string.Empty);
Debug.WriteLine(variableNmaes[0]);          // @MAKER_NAME
Debug.WriteLine(variableNmaes[1]);          // @VERSION
Debug.WriteLine(variableNmaes[2]);          // @TORQUE
Debug.WriteLine(variableNmaes[3]);          // @DEVICE_VERSION
```

Table 2-3 CaoController Class System Variables

Variable name	Data type	Description	Attribute	
			Get	Put
@MAKER_NAME	VT_BSTR	Returns the manufacturer Dai-ichiSeiko.	○	-
@VERSION	VT_BSTR	Returns the provider version.	○	-
@TORQUE	VT_R8	Returns the current Torque value.	○	-
@DEVICE_VERSION	VT_BSTR	Returns the ESTORQ device version.	○	-

2.2.4. CaoController::AddVariable method

Create CaoVariable objects from the CaoController. Enter the variable name from the system variable list in Table 2-3. No option string is required.

Format

```
AddVariable (
        BSTR bstrVarName,   // [in] Variable name (specifies the system variable name)
        BSTR bstrOption     // [in] Option string
 );
```

Example (C#)

```
        CCaoVariable varTorque = ctrl.AddVariable("@TORQUE", "");
        CCaoVariable varDeviceVersion = ctrl.AddVariable("@DEVICE_VERSION", "");
```

2.2.5. CaoVariable::get_Value Properties

Gets the status of the variables created by the AddVariable method. The type of the return value depends on the name of the variable or variables specified in the AddVariable. Refer to the system variables in Table 2-4 for the return values of each variable.

Example (C#)

```
        Debug.WriteLine(varTorque.Value + "[Nm]");          // 0.00134293737028446[Nm]
        Debug.WriteLine(varDeviceVersion.Value);            // TRQ Ver 4.19 2018/02/06
```

2.3. Error-code

This provider defines specific error codes. Unique error codes are shown in Table 2-4. For ORiN2 common errors, refer to the error codes section in the ORiN2 Programming Guide.

Table 2-4 Unique Error Codes

Error Number	Description
0x80110000	Rated torque parameter is invalid during AddController. Please specify the rated torque within the following range 0.0 < rated torque ≦ 100000.0.
0x80110001	The length of the response packet is invalid. Shorten the transmission distance because noise may have occurred.
0x80110002	Checksum is invalid. Shorten the transmission distance because noise may have occurred.

3. Sample Code (C#)

```
using ORiN2.ManagedCAO;

CCaoEngine eng = new CCaoEngine();
CCaoWorkspace ws = eng.AddWorkspace("sample", "");
CCaoController ctrl = ws.AddController("ctrl1", "CaoProv.Dai-ichiSeiko.ESTORQ", string.Empty,
                                       "Conn=com:1:115200,RatedTorque=10,Timeout=1000");

double torque = (double)ctrl.Execute("GetTorque", "");
Debug.WriteLine(torque + "[Nm]");                    // 0.00134293737028446[Nm]

string deviceVersion = ctrl.Execute("GetDeviceVersion", "").ToString();
Debug.WriteLine(deviceVersion);                      // TRQ Ver 4.19 2018/02/06

string[] variableNmaes = ctrl.GetVariableNames(string.Empty);

CCaoVariable varTorque = ctrl.AddVariable(variableNmaes[2].ToString(), "");
CCaoVariable varDeviceVersion = ctrl.AddVariable(variableNmaes[3].ToString(), "");

Debug.WriteLine(varTorque.Value + "[Nm]");           // 0.00134293737028446[Nm]
Debug.WriteLine(varDeviceVersion.Value);             // TRQ Ver 4.19 2018/02/06

if(eng != null)
{
    eng.Dispose();
}
eng = null;
ws = null;
ctrl = null;
```

4. Communication protocol command correspondence table

Table 4-1 shows the correspondence between the Execute method and Variable variables implemented in this provider and the communication commands in the "Electrostatic Capacitance-type Torque Sensor ESTORQ Communication Specifications (USB, RS422)".

Table 4-1 Communication Command Support

Execute method	Variable name	Get/put	Communications command
GetTorque	@TORQUE	Get	R
GetDeviceVersion	@DEVICE_VERSION	Get	V
Reset	-	Put	O