

VISA COM provider

For DC power PA made of industry Kikusui electronic S Series

Version 1.0.0

User's guide

October 2, 2019

arks:

This document is translated into English using machine translation.

Contents

1. Introduction	4
2. Outline of provider	5
2.1. Outline.....	5
2.2. Method property.....	6
2.2.1. CaoWorkspace::AddController method	6
2.2.1.1. Conn is optional.....	6
2.2.2. CaoController::GetVariableNames property	7
2.2.3. CaoController::AddVariable method	7
2.2.4. CaoController::Execute method.....	8
2.2.5. CaoVariable::get_Value property	8
2.2.6. CaoVariable::put_Value property	8
2.3. Variable list.....	10
2.3.1. CaoController class.....	10
2.4. Error code	11
2.5. Limitations.....	13
3. Command reference	14
3.1. CaoController class	14
3.1.1. CaoController::Execute ("Clear") command.....	14
3.1.2. CaoController::Execute ("PowerOff") command.....	15
4. Sample program.....	16
5. Appendix	18
5.1. Table for device message	18
5.2. Reference URL	19

1. Introduction

This book is an user's guide of the CAO provider to control as for DC power made of industry electronic of Kikusui (PA S Series) through the power supply controller (PIA4800 series).

CAO provider (CaoProvKIKUSUIVISACOM.dll) that treats in this book is called VISA COM provider. It explains the function and the variable that the provider offers since the chapter.

2. Outline of provider

2.1. Outline

It connects it with DC power PA S Series through power supply controller PIA4800 series made of industry electronic of Kikusui, and the VISA COM provider is an instrument setup and CAO provider that acquires the value now.

¹Use VISA COM API mounted according to standard issue (VISA) with the measuring instrument connection software for the communication with the power supply controller. VISA COM API becomes possible to use by installing the KI-VISA library that Kikusui electronic industry offers it. Download and install version 5.0.9 or more from WEB site (<http://www.kikusui.co.jp/download/>) of Kikusui electronic industry. Refer to the chapter of Setup of "KI-VISA Library VISA COM guide book" for the method of installing the KI-VISA library.

It is VISA COM provider and a whole of device block diagram as follows.

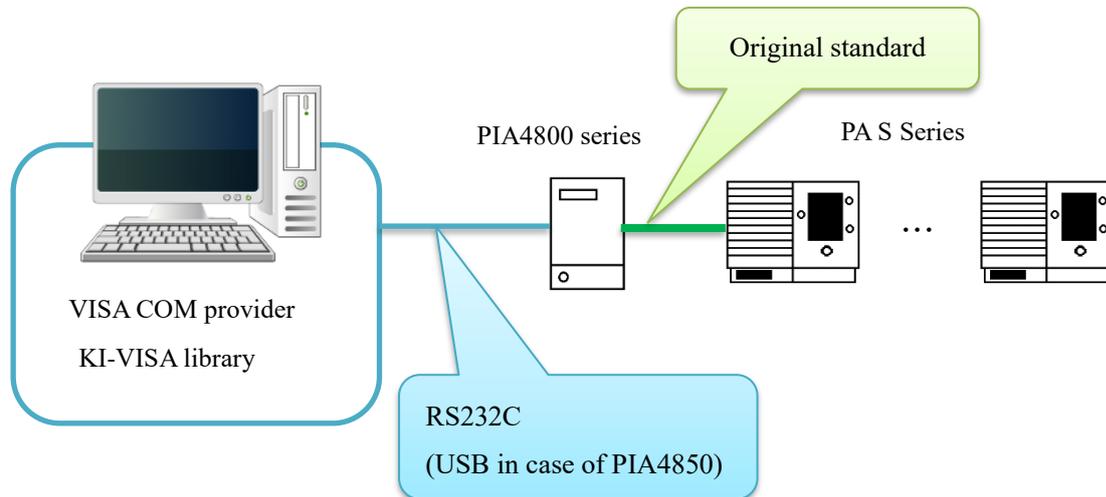


Figure2-1 Whole block diagram

The file format of the VISA COM provider is DLL(Dynamic Link Library), and when using it from the CAO engine, it is dynamically loaded. Is installed ORiN2SDK when you use it?Table2-1It is necessary to register the registry by the hand work referring to [wo].

Table2-1 VISA-COM provider

File name	CaoProvKIKUSUIVISACOM.dll
-----------	---------------------------

¹ Virtual Instrument Software Architecture の略

ProgID	CaoProv.KIKUSUI.VISACOM
² Registry registration	RegistAsm.bat CaoProvKIKUSUIVISACOM.dll
Blotting out of registry registration	UnregistAsm.bat CaoProvKIKUSUIVISACOM.dll

2.2. Method property

2.2.1. CaoWorkspace::AddController method

The VISA COM provider connects the communication referring to connected parameter at AddController.

Format AddController (<bstrCtrlName:VT_BSTR>,<bstrProvName:VT_BSTR>,
<bstrPcName:VT_BSTR>, [<bstrOption:VT_BSTR>])

<bstrCtrlName> : In controller name
 <bstrProvName> : In provider name. "CaoProv.KIKUSUI.VISACOM fixed value ="
 <bstrPcName> : Execution machine name of in provider (unused)
 <bstrOption> : In optional character string

The list specified for an optional character string is shown as follows.

Table2-2Optional character string of CaoWorkspace::AddController

Option	Meaning
Conn=< connected parameter >	Indispensability. Set a communication form and the connected parameter.
Timeout =< timeout period >	Specify the timeout period when the command is sent and received by the millisecond. (default: 3000)

2.2.1.1. Conn is optional.

Connected parameter character string of optional Conn is shown as follows. A possible omission is shown here in the square bracket (""). Moreover, the underlined part under the explanation of each parameter shows the default value when the option is not specified.

【 cereal 】

"Conn=COM:<COM Port>[:<BaudRate>]"

<COM Port> : COM port number. '1' -COM1, '2' -COM2, '3' -COM3.

<BaudRate> : Transmission rate. 2400,4800,9600,19200.

² RegistAsm.bat および UnregistAsm.bat は {ORiN2 インストールフォルダ}\¥DotNet¥BAT 以下に配置されています。

³The parity bit, the data bit, the stop bit, and the flow control need not be specified for fixation.

【USB】

```
"Conn=USB:<SerialNo>[:<ModelNo>]"
```

<SerialNo> : Serial number of PIA series equipment to be connected.

<ModelNo> : ⁴Serial number of PIA series equipment to be connected. PIA4850.

Usage example

```
CaoEngine caoEng;
CaoWorkspaces caoWss;
CaoWorkspace caoWs;
CaoControllers caoCtrls;
CaoController caoCtrl;

caoEng = new CaoEngine();
caoWss = caoEng.Workspaces;
caoWs = caoWss.Item(0);
caoCtrls = caoWs.Controllers;

// When connecting it by using serial communications
caoCtrl = caoWs.AddController("VISACOM_SAMPLE", "CaoProv.KIKUSUI.VISACOM", null,
                             "Conn=COM:1:19200,Timeout=5000");

// When connecting it by using the USB communication
caoCtrl = caoWs.AddController("VISACOM_SAMPLE", "CaoProv.KIKUSUI.VISACOM", null,
                             "Conn=USB:J009999:PIA4850,Timeout=5000");
```

2.2.2. CaoController::GetVariableNames property

Table2Acquire the list of the shown system variable identifier of [ni].

2.2.3. CaoController::AddVariable method

Make the variable object setting/to acquire the value to the equipment.

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

<bstrVariableName> : In variable identifier

<bstrOption> : In optional character string

Usage example

```
// Add the variable for the output current acquisition and the setting.
CaoVariable val;
val = caoCtrl.AddVariable("@ISET");
```

³ VISACOM はフロー制御(Xon/Xoff)を使用します(機器の推奨設定).

⁴ 現時点では PIA4850 のみ対象としています.

2.2.4. CaoController::Execute method

The Execute method of the CaoController class is a method for the execution of the command. Please refer to the Command reference for details of each command.

Format Execute (<bstrCommandName:VT_BSTR>,[<vntParam:VT_VARIANT>])

<bstrCommandName> : In command name
<vntParam> : In parameter

2.2.5. CaoVariable::get_Value property

⁵Send [kuerimesse-ji] corresponding to the variable identifier, and return the lead backing data by the character string. The setting of the lead backing data returns data that adds the header and the unit when there are a header and unit data. Check the 2.5 Limitations before use.

Usage example

```
// Add the variable for the output current acquisition and the setting.
CaoVariable val;
val = caoCtrl.AddVariable("@ISET");

// Acquire the output current setting value.
var value = val.Value;

/*
 * Data when there are header and unit data
 * ISET 1.25A
 *
 * Data when there are neither header nor unit data
 * 1.25
 */
```

2.2.6. CaoVariable::put_Value property

Send the command written in the device to be accessed after converting the value passed by the argument according to optional specification. Check the 2.5 Limitations before use.

Usage example

```
// Add the variable for the output current acquisition and the setting.
CaoVariable val;
val = caoCtrl.AddVariable("@ISET");

// Set the output current value to 13.20A.
val.Value = "13.20";
```

⁵ リードバックデータの設定は@HEAD 変数で切替可能です。

2.3. Variable list

2.3.1. CaoController class

Table23 CaoController class system variable list

Variable identifier	Data type	Explanation	Attribute	
			get	put
@MAKER_NAME	VT_BSTR	Return "KIKUSUI manufacturer name =".	-	-
@VERSION	VT_BSTR	Return the provider version.	-	-
@NODE	VT_BSTR	Specification/acquire the node address. Return "0" at the node address unsetting. Set value : 1~36	-	-
@IOUT	VT_BSTR	Acquire the monitor value of the output current of the connected DC power device.	-	-
@ISET	VT_BSTR	Setting/acquire the output current of the connected DC power device. Set value: Minimum value-maximum value of connected power-supply unit	-	-
@VOUT	VT_BSTR	Acquire the output voltage monitor value of the connected DC power device.	-	-
@VSET	VT_BSTR	Setting/acquire the output voltage of the connected DC power device. Set value: Minimum value-maximum value of connected power-supply unit	-	-
@STS	VT_BSTR	Acquire the content of the status register.	-	-
@STB	VT_BSTR	Acquire the content of status byte register.	-	-
@FAU	VT_BSTR	Acquire the content of the fault register.	-	-
@ERR	VT_BSTR	Acquire the content of the error register. After reading, the error is reset.	-	-
@FUNMASK	VT_BSTR	Set/reset each bit of the fault register. Specify it by the decimal number or the hexadecimal number. Set value : 0-255 or 00h-FFh	-	-
@UNMASK	VT_BSTR	Set/reset each bit of the service request enable register. Specify it by 0 -adic numbers or the hexadecimal number.	-	-

Variable identifier	Data type	Explanation	Attribute	
			get	put
		Set value : 0-255 or 00h-FFh		
@HEAD	VT_BSTR	Specification/acquire the presence of the header (Contain the channel header) of the lead backing data and the unit data. 0 : Apply neither header nor the unit data. (default) 1 : Apply the header and the unit data.	-	-
@LOCK	VT_BSTR	Set effective/invalidity of the panel operation. 0 : Panel effective operation (default) 1 : Panel operation invalidity	-	-
@OUT	VT_BSTR	Set on/off of the output of the connected DC power device. 0 : Outputting off 1 : Outputting on	-	-
@OCSET	VT_BSTR	Setting/acquire the OCP operation point of the connected DC power device. Set value : Rated output current About 110%- about 10 %	-	-
@OVSET	VT_BSTR	Setting/acquire the OVP operation point of the connected DC power device. Set value : 10 of voltage ratings %-110 %	-	-
@IDN	VT_ARRAY VT_BSTR	Acquire power supply controller's model name in the array. 0 : Name of the company 1 : Model name 2 : Cereal No (unused) 3 : Version number	-	-

2.4. Error code

The error code defined in the KI-VISA library might be returned though there is no error code peculiar to the VISA COM provider. Refer to the chapter of the VISA COM API status code of "KI-VISA Library VISA COM guide book" for the error of KI-VISA.

Moreover, refer to the chapter of the error code of "ORiN2 programming guide" for the ORiN2 commonness error.

2.5. Limitations

It connects it with DC power PA S Series, and the instrument setup and it is necessary to set the node address to the equipment of the PA S Series to acquire the value now. It fails in the operations of the system variable other than "@MAKER_NAME" "@ VERSION" "@NODE" in the node address unsetting. The node address must be initialized when you drop the power supply of the equipment of the PA S Series, and start (reactivation) and set the node address again.

The node address can set whether to specify the node address by system variable "@NODE" of the VISACOM provider by operating the panel of the equipment of the PA S Series.

3. Command reference

3.1. CaoController class

Table3-1CaoController class command list

Command	Function	Page
Clear	Initialize the value of the program data and the register.	14
PowerOff	Turn off the power supply of the DC power device.	15

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

Transmit the CLR message, and initialize the value of the program data and the register. Moreover, turn off the output of the DC power device. About an initial valueTable3-2Refer to [wogo].

Format Clear()

Argument : None

Return value : None

Usage example

```
// Initialize the program data register.
caoCtrl.Execute("Clear");
```

Table3-2Program data and register initial value

Message header	Initial value
NODE	0
TRM(DELIMITER)	0
FUNMASK	0h
HEAD	0
SILENT	1
*SRE(UNMASK)	0h

Register	Each bit
Status byte register	0
Fault register	0
Error register	0

3.1.2. GaoController::Execute ("PowerOff") command

Transmit the POW message, and turn off the power supply of the DC power device.

Format PowerOff ()

Argument : None

Return value : None

Usage example

```
// Turn off the power supply.  
caoCtrl.Execute("PowerOff");
```

4. Sample program

It communicates with the PA S Series by using this provider as follows, and the execution of the Execute command of the output current value ..reading and finishing come.. shows sample program (C#). Please check the 2.5 Limitations before running.

Sample	Program.cs
--------	------------

```
using System;
using System.Runtime.InteropServices;
using ORiN2.interop.CAO;

namespace VISACOMSample
{
    class Program
    {
        static void Main(string[] args)
        {
            CaoEngine caoEng;
            CaoWorkspaces caoWss;
            CaoWorkspace caoWs;
            CaoControllers caoCtrls;
            CaoController caoCtrl;
            CaoVariables caoVariables;

            caoEng = new CaoEngine();
            caoWss = caoEng.Workspaces;
            caoWs = caoWss.Item(0);
            caoCtrls = caoWs.Controllers;

            // Connect it by using serial communications.
            caoCtrl = caoWs.AddController("VISACOM_SAMPLE", "GaoProv.KIKUSUI.VISACOM", null,
                                        "Conn=COM:1:19200,Timeout=5000");

            caoVariables = caoCtrl.Variables;

            // Add the variable of the output current.
            CaoVariable val;
            val = caoCtrl.AddVariable("@ISET");

            // Acquire the output current setting value.
            var value = val.Value;

            // Set the output current value to 13.20A.
            val.Value = "13.20";

            // Initialize the program data register.
            caoCtrl.Execute("Clear");

            // Turn off the power supply of the DC power device.
            caoCtrl.Execute("PowerOff");

            // Liberating of COM object-----
            // Liberate CaoVariable.
            caoVariables.Clear();
            Marshal.ReleaseComObject(val);
        }
    }
}
```

```
    val = null;

    // Liberate CaoVariables.
    Marshal.ReleaseComObject(caoVariables);
    caoVariables = null;
    // Liberate CaoController.
    caoCtrls.Clear();
    Marshal.ReleaseComObject(caoCtrl);
    caoCtrl = null;

    // Liberate CaoControllers.
    caoWss.Clear();
    Marshal.ReleaseComObject(caoCtrls);
    caoCtrls = null;
    // Liberate CaoWorkspace.
    Marshal.ReleaseComObject(caoWs);
    caoWs = null;
    // Liberate CaoWorkspaces.
    Marshal.ReleaseComObject(caoWss);
    caoWss = null;

    // Liberate CaoEngine.
    Marshal.ReleaseComObject(caoEng);
    caoEng = null;
  }
}
```

5. Appendix

5.1. Table for device message

The system variable and the correspondence of the device message using VISA COM API are shown below.

Table5-1Correspondence of system variable and device message

Variable identifier	get_Value	set_Value
@NODE	NODE?	NODE 【 data 】 Example:NODE 5
@IOUT	IOUT?	-
@ISET	ISET?	ISET 【 data 】 Example:ISET 12.34
@VOUT	VOUT?	-
@VSET	VSET?	VSET 【 data 】 Example:VSET 12.34
@STS	STS?	-
@STB	*STB?	-
@FAU	FAU?	-
@ERR	ERR?	-
@FUNMASK	FUNMASK?	FUNMASK 【 data 】 Example:FUNMASK FFh
@UNMASK	UNMASK?	UNMASK 【 data 】 Example:UNMASK FFh
@HEAD	HEAD?	HEAD 【 data 】 Example:HEAD 1
@LOCK	-	LOCK 【 data 】 Example:LOCK 1
@OUT	OUT?	OUT 【 data 】 Example:OUT 1
@OCSET	OCSET?	OCSET 【 data 】 Example:OCSET 12.34
@OVSET	OVSET?	OVSET 【 data 】 Example:OVSET 12.34
@IDN	*IDN?	-

5.2. Reference URL

Connection & programming guide is open to the public on the site of Kikusui electronic industry. Refer together with the equipment manual.

https://www.kikusui.co.jp/kiku_manuals/P/PIA4800/index_J.html

The following content can be confirmed on the site.

- Connected manual with power-supply unit
- Explanation of device message
- PIA4800 series message list
- About the register
- Installation manual of KI-VISA
- IVI-COM measuring instrument driver programming guide