HIOKI

MEMORY HiLOGGER - LR8400 provider

Version 1.0.1

User's guide

April 20, 2020

Remarks:

【 revision history 】

Version	Date	Content
1.0.0	2019-10-7	First edition
1.0.1	2020-04-20	Bug fixed. memory leak.

【 Operation check model 】

Model	Notes
LR8400 series	It corresponds to virtual COM (cereal) connection by TCP/IP and USB.

Contents

1. はじめに

This book is an user's guide of the CAO provider that sends and receives data by connecting TCP/IP or the cereal with MEMORY HiLOGGER (LR8400) made of HIOKI, and using the communication command. CAO provider (CaoProvHIOKILR8400.dll) that treats in this book is called LR8400 provider.

Chapter 2 gives an overview of the LR8400 provider and details on variables and commands. Refer to the LR8400 command correspondence table for the correspondence status of communication commands implemented by the LR8400 provider. Refer to the communication command handling explanation in [roga-yu-tei;ritei;] (CD-R) of "Communication command" of the manual of the LR8400 series and the main body of LR8400 attachment for the communication command.

Connected image of PC and LR8400 is shown in the following. The driver's installation is necessary for USB connection. Moreover, set IP when connecting it by LAN.

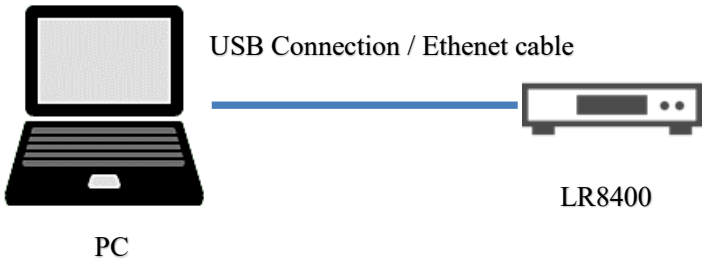

Figure1-1Connected image

2. Outline of provider

2.1. Outline

[1]The LR8400 provider is CAO provider that sends and receives data by connecting TCP/IP or the cereal with MEMORY HiLOGGER made of HIOKI, and using the communication command.

The file format is DLL(Dynamic Link Library), and when using it from the CAO engine, it is dynamically loaded. When the LR8400 provider is used, it is necessary to install ORiN2SDK or to register the registry by the hand work referring to the table below.

Table2-1LR8400 provider

File name	CaoProvHIOKILR8400.dll
ProgID	CaoProv.HIOKI.LR8400
Registry registration	regsvr32 CaoProvHIOKILR8400.dll
Blotting out of registry registration	regsvr32 /u CaoProvHIOKILR8400.dll

[1] For details on communication commands, refer to the logger utility (CD-R) included with the LR8400. Description of communication commands is saved in HTML format.

2.2. Limitations

Use it noting the following when you use this provider.

- [2]The analysis of the response data from the main body equipment is the header and is made for the LR8400 provider by assumption. Do not set the setting of the header except turning on.
- It becomes the post-winning by the specification of the main body of LR8400 equipment at the TCP connection. When other PC etc. connect TCP with LR8400 while connecting TCP in this provider, the connection of this provider is cut.
- Do not change the network transmission setting of the main body equipment after it connects it in the LR8400 provider.

  After of the reactivation of the main body equipment, try to connect it again in the LR8400 provider when changing.
- Communication command[1]It drinks and the real-time data cannot be acquired in a short recording interval of one second in the specification. However, the data after the measurement stops for one second at short recording intervals can be acquired.

[2] For header settings, refer to System> Communication Settings of the main unit.

2.3. Method property

2.3.1. CaoWorkspace::AddController method

The LR8400 provider sets connected parameter for the communication at AddController, and connects the communication.

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

<bstrCtrlName>	:	In controller name
<bstrProvName>	:	In provider name. "CaoProv.HIOKI.LR8400"
<bstrPcName>	:	Execution machine name of in provider (unused)
<bstrOption>	:	In optional character string

Example

```
// Generation of engine
var eng = new CCaoEngine();
// Generation of workspace
var ws = eng.AddWorkspace("SampleWorkSpace","");

// At the TCP connection
var option =
"Conn=TCP:192.168.1.1:8802,Delimiter=1,ConnTimeout=5000,TimeOut=4000";
// At the cereal connection
// var option =
"Conn=COM:3:9600:N:8:1,Delimiter=1,ConnTimeout=5000,TimeOut=4000";

// Connection to LR8400
this.ctrl = ws.AddController("Sample","CaoProv.HIOKI.LR8400","",option);
```

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. (Please refer to chapter 2.3.1.1 for details.)
Delimiter [= <number of delimiter>]	[3] Specify the delimiter when the command is communicated. (default: 1) 0 : LF 1 : CR+LF
ConnTimeout [=< timeout period >]	Specify the timeout period (millisecond) when it connects it. (default: 3000)
Timeout [=< timeout period >]	Specify the timeout period of the reply command (millisecond). (default: 3000)

[3] デリリミタの設定は，本体機器のシステム＞通信設定を参照してください

2.3.1.1. 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 device 】

"Conn=com:<COM Port>"

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

【 Ethernet device 】

TCP/IP

"Conn=TCP:<Dest IP Address>:<Dest Port No>[:<Src IP Address>:<Src Port No>]"

<Dest IP Address> : Internet Protocol address of connection destination.

<Dest Port No> : Port number of connection destination.

- Specify two for the last 1 digit by the specification of the main body equipment.

Example) 880X ⇒ 8802

<Src IP Address> : Internet Protocol address. 127.0.0.1

<Src Port No> : Port number. 5001

2.3.2. CaoController::GetVariableNames property

Get the list of system variable names shown in 2.4.

2.3.3. CaoController::AddVariable method

Make the variable object to read LR8400 data.

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

<bstrVariableName> : In variable identifier
<bstrOption> : In optional character string

Example

```
var result = this.ctrl .AddVariable("@MAKER_NAME","");
```

2.3.4. CaoController::Execute method

The Execute method of CaoController class is a method for executing commands. Details of each command Refer to the command reference.

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

<bstrCommandName> : [in] Command
<vntParam> : [in] Parameter

Example

```
var result = this.ctrl.Execute("Send", ":START");
```

2.3.5. CaoVariable::get_Value property

Send the value passed by the argument and send the reading command to the data to be accessed according to optional specification.

2.4. Variable list

2.4.1. CaoController class

2.4.1.1. System variable

Table2-3CaoController class system variable list

Variable identifier	Data type	Explanation	Attribute	
			get	put
@MAKER_NAME	VT_BSTR	Return manufacturer name = HIOKI.	✔	–
@VERSION	VT_BSTR	Return the provider version.	✔	–
@TITLE_COMMENT	VT_BSTR	Return the comment on the title of LR8400. (Transmit the inquiry command of the title comment, and return the content of the response.)	✔	–

2.4.1.2. User variable

Table2-4CaoController class user variable list

Only the get operation is possible by all variables.

[4]Variable identifier	Data type	Explanation	Attribute	
			get	put
UI<??>	VT_BSTR	Acquire an input kind of set value of the specified channel. ..(.. reference:2.4.1.2.1)	✔	–
UR<??>	VT_R8	Acquire a set value of the measurement cooking stove of the specified channel. ..(.. reference:2.4.1.2.2)	✔	–
CCH<??>	VT_BSTR	Acquire the setting of the comment character of the specified channel. ..(.. reference:2.4.1.2.3)	✔	–
MVREAL<??>	VT_R8	Acquire the value in which the value of the output of real-time data of the specified channel (voltage, temperature, humidity, resistance, multiplication, and rotational speed) is converted into the voltage value. ..(.. reference:2.4.1.2.4)	✔	–

[4] <??>には，任意の文字列を指定してください.

[4]Variable identifier	Data type	Explanation	Attribute	
			get	put
MTVREAL<??>	VT_ARRAY |VT_R8	The value of the output of real-time data of the specified unit (It is all ..the channel of measurement ON..) is acquired and the converted value is acquired the voltage value. ..(.. reference:2.4.1.2.5)	✔	–
MTVRCH<??>	VT_ARRAY |VT_BSTR	Acquire all the channel names of measurement ON of the specified unit. ..(.. reference:0)	✔	–

2.4.1.2.1. UI <??>

It is a variable to acquire an input kind of set value of the specified channel.

Table 2-5 shows the required options when creating a variable, and Table 2-6 shows the index of the acquired value and the data contents.

Table2-5 Indispensability is UI optional.

Option	Range of value	Explanation
CH	CH1_1 ～ CH4_15	Channel to acquire input kind Example) CH = CH1_1

Table2-6 UI acquisition value

Explanation
Input kind VOLTAGE : Voltage TC : Thermo-couple RTD : Resistance temperature sensor HUMIDITY : Humidity RESIST : Resistance

Example

```
var userVar= this.ctrl.AddVariable("UI1","CH=CH1_1");
Debug.WriteLine(userVar.Value.ToString());

// example of → display result: VOLTAGE
```

2.4.1.2.2. UR <??>

It is a variable to acquire a set value of the kind of the measurement cooking stove of the specified channel.

Table 2-7 lists the required options when creating variables, and Table 2-8 lists the acquired value indexes and data contents.

Table2-7Indispensability is UR optional.

Option	Range of value	Explanation
CH	CH1_1 ～ CH4_15	Channel to acquire kind of measurement cooking stove Example) CH = CH1_1

Table2-8UR acquisition value

Explanation
Value in which measurement cooking stove of specified channel is converted into floating point. - When 1-5V is specified with the main body equipment, the value of 15 is set.

Example

```
var userVar= this.ctrl.AddVariable("UR1","CH=CH1_1");
Debug.WriteLine(userVar.Value.ToString());

// When the main body equipment setting is 10mV →Example of display result: 0.01
// When the main body equipment setting is 20mV →Example of display result: 0.02
// When the main body equipment setting is 100mV →Example of display result: 0.1
// When the main body equipment setting is 200mV →Example of display result: 0.2
// When the main body equipment setting is 1V →Example of display result: 1
// When the main body equipment setting is 2V →Example of display result: 2
// When the main body equipment setting is 10V→Example of display result: 10
// When the main body equipment setting is 20V→Example of display result: 20
// When the main body equipment setting is 100V →Example of display result: 100
// When the main body equipment setting is 1-5V →Example of display result: 15
```

2.4.1.2.3. CCH <??>

It is a variable to acquire a set value of the comment character string of the specified channel.

Table 2-9 lists the required options when creating a variable, and Table 2-10 lists the index of the acquired value and the data contents.

Table2-9Indispensability is CCH optional.

Option	Range of value	Explanation
CH	CH1_1 ～ CH4_15 PLS1 ～ PLS8 W1 ～ W30	Channel to acquire comment character string Example) CH = PLS1

Table2-10CCH acquisition value

Explanation
Character string of form that encloses it with [daburuko-te-shon] before and behind comment character The input of a special character is displayed as follows though the one-byte character that can be treated as a comment character (maximum em-size 20 characters and normal-width 40 characters) is a character that can be input with the main body of LR8400. ^2 : The second power ^3 : The third power ~c : ° ~e : ε ~u : μ ~o : Ω ^^ : ^ ~~ : ~ ~, : ' ~; : " Replace it with space excluding the above-mentioned.

Example

```
var userVar= this.ctrl.AddVariable("CCH1","CH=CH1_1");
Debug.WriteLine(userVar.Value.ToString());

// Example of → display result "Test comment"
```

2.4.1.2.4. MVREAL <??>

It is a variable to acquire the value in which the value of the output of real-time data of the specified channel (voltage, temperature, humidity, resistance, multiplication, and rotational speed) is converted into the voltage value.

Table 2-11 lists the required options when creating variables, and Table 2-12 lists the data contents of the acquired values.

Table2-11Indispensability is MVREAL optional.

Option	Range of value	Explanation
CH	CH1_1 ～ CH4_15 PLS1 ～ PLS8 LOG ALARM W1 ～ W30	Specify the channel and the pulse number, etc. in which real-time data is output. The data output by the specified value is different. The combination of the specified value and the output data is the following. CH1_1 ～ CH4_15 : Voltage, temperature, humidity, and resistance ..drinking.. analog data PLS1 ～ PLS8 : Multiplication and rotational speed ..drinking.. [parusude-ta] LOG : 0～255 ..drinking.. [dei;jitaruinde-ta] ALARM : 0～15 ..drinking.. [ara-muautode-ta] W1～W30 : Wavy operation ..drinking.. wavy operation data

Table2-12MVREAL acquisition value

Explanation
Voltage value of value input to channel and pulse number, etc. specified with indispensable, optional CH.

Example

```
var userVar= this.ctrl.AddVariable("MVREAL1","CH=CH1_1");
Debug.WriteLine(userVar.Value.ToString());

// example of → display result: 0.02328
```

2.4.1.2.5. MTVREAL <??>

It is a variable in the voltage value to acquire the converted value as for the value of the output of real-time data of the specified unit (It is all ..the channel of measurement ON..).

Table 2-13 lists the required options when creating variables, and Table 2-14 lists the indexes of the acquired values and the data contents.

Table2-13Indispensability is MTVREAL optional.

Option	Range of value	Explanation
UNIT	UNIT1 UNIT2 UNIT3 UNIT4 PLS&ALM CALC1 CALC2	Specify the unit that outputs real-time data. The data output with each channel is different. The combination of each channel and the output data is the following. CH1_1 ～ CH4_15 : Voltage, temperature, humidity, and resistance ..drinking.. analog data PLS1 ～ PLS8 : Multiplication and rotational speed ..drinking.. [parusude-ta] LOG : 0～255 ..drinking.. [dei;jitaruinde-ta] ALARM : 0～15 ..drinking.. [ara-muautode-ta] W1 ～ W30 : Wavy operation ..drinking.. wavy operation data

Table2-14MTVREAL acquisition value

[5]Index	Explanation
0～14	Voltage value of value input to channel and pulse number, etc. specified with indispensable, optional UNIT

[5]Since only channels with measurement ON are output, the index range is variable.
For example, when CH1_2 is removed from the measurement target on the LR8400 body,
When UNIT1 is specified as an option, CH1_2 of CH1_1 to CH1_15 is excluded, so the output index is 0 to 13.

Example

```
var userVar= this.ctrl.AddVariable("MTVREAL1","UNIT=UNIT1");
foreach(var item in userVal as Array)
{
Debug.WriteLine(item.Value.ToString());
}

// example of → display result: 0.066065
                0.023062
                -3276.8
                0.0025615
                0.0028485
                0.0022975
                0.0031365
                0.003715
                0.0020615
                0.001176
                0.0001085
                0.0001085
                -0.0004455
                -0.0008855
                -0.0017835
```

2.4.1.2.6. MTVRCH<??>

It is a variable to acquire all the channel names of measurement ON of the specified unit.

Table 2 15 lists the required options when creating variables, and Table 2 16 lists the indexes and data contents of the acquired values.

Table2-15Indispensability is MTVRCH optional.

Option	Range of value	Explanation
UNIT	UNIT1 UNIT2 UNIT3 UNIT4 PLS&ALM CALC1 CALC2	Unit to be confirmed

Table2-16MTVRCH acquisition value

Index[5]	Explanation
0 ～ 14	Channel name

Example

```
var userVar= this.ctrl .AddVariable("MTVRCH","UNIT=UNIT1");
foreach(var item in userVal as Array)
{
Debug.WriteLine(item.Value.ToString());
}

    // example of → display result: CH1_1
                    CH1_2
                    CH1_3
                    CH1_4
                    CH1_5
                    CH1_6
                    CH1_7
                    CH1_8
                    CH1_9
                    CH1_10
                    CH1_11
                    CH1_12
                    CH1_13
                    CH1_14
                    CH1_15
```

3. Command reference

3.1. CaoController class

Table3-1CaoController class command list

Command	Function	Page
General-purpose command transmission		
Send	Transmit specified command statement [azaretsu]. To command statement [azaretsu]?However, after it transmits, [fuku] [mareteiru] situation receives data.	P. 24
Execution processing command transmission		
Start	Begin the measurement (wavy taking operation).	P. 24
Status	Acquire the state of storage.	P. 25
Stop	Stop measuring when you complete the measurement (wavy taking operation).	P. 25
Abort	Cancel the measurement (wavy taking operation). Stop measuring even if it doesn't complete the measurement.	P. 26
Error	Acquire the number of the error or the warning that occurs by the main body.	P. 26

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

It is command statement [azaretsu] in LR8400 specified for the argument. [1]Is transmitted [wo], and do to command statement [azaretsu]?However, [fuku] [mareteiru] situation receives the content of the response.

Format Send(<command>)

command : Command statement [azaretsu] that transmits in. (VT_BSTR)

Return value : To command statement [azaretsu] that transmits out?However, it is time when is [fuku] [mareteinai]
None. (VT_EMPTY)

To transmitted command statement [azaretsu]?However, [fuku] [mareteiru] situation :.
Content of response of LR8400. (VT_BSTR)
- When transmitted command statement [azaretsu] is wrong etc.
The time-out error occurs when the content of the response cannot be acquired.

Example

```
this.ctrl.Execute("Send", ":START");
var result = this.ctrl.Execute("Send", ":STATUS?");
```

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

Transmit the start processing command to LR8400, and begin the measurement (wavy taking operation).

Format Start()

Return value : None

Example

```
this.ctrl.Execute("Start", null);
```

3.1.3. CaoController::Execute ("Status") command

Transmit the inquiry command stored in LR8400, and acquire the state of storage.

Format Status()

Return value : Return the state of the out storage by the numerical value. (VT_UI1)
The response is : of three situations.
It means the start, and storage.
The meaning of each bit value is the following.

bit0	:	Start
bit1	:	It is storing it.
bit2	:	Trigger waiting
bit3	:	[Puritoriga] waiting
bit4	:	Unused
bit5	:	Save inside

Example

```
var result = this.ctrl.Execute("Status",null);
```

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

Transmit the stop processing command to LR8400. End when you complete the measurement (wavy taking operation).

Stop after it measures it at the record time when you execute this command once when a continuous record is turning off.

Stop measuring immediately when a continuous record is turned on, and this command is executed twice at the time of turning off.

Format Stop()

Return value : None

Example

```
this.ctrl.Execute("Stop",null);
```

3.1.5. CaoController::Execute ("Abort") command

Transmit the abort processing command to LR8400, and cancel the measurement (wavy taking operation). Stop measuring even if it doesn't complete the measurement.

Format Abort()

Return value : None

Example

```
this.ctrl.Execute("Abort",null);
```

3.1.6. CaoController::Execute ("Error") command

Transmit the inquiry command of the main body error number to LR8400, and acquire the number of the error or the warning that occurs by the main body.

Format Error()

Return value : Return the number of the error or the warning that occurs by the main body of out. (VT_UI2)
- Refer to the manual of the main body for the error or the warning.

Example

```
var result = this.ctr.Execute("Error",null);
```

4. Sample program

As follows..various..variable..acquire..command..various..execute..sample..show.

List 4-1

```
// Abbreviation.
using ORiN2.ManagedCAO ;

namespace HIOKI_LR8400_Sample
{
        public partial class Sample : Form
        {
                private CCaoEngine eng;
                private CCaoWorkspace ws;
                private CCaoWorkspaces wss;
                private CCaoControllers ctrls;
                private CCaoController ctrl;

                public Sample()
                {
                        InitializeComponent();
                }

                private void Sample_Load(object sender, EventArgs e)
                {
                        // Generation of Cao engine
                        this.eng = new CCaoEngine();
                        this.wss = this.eng.Workspaces;
                        this.ws = this.wss[0];
                        this.ctrls = this.ws.Controllers;

                        // At the TCP connection
                        var option =
"Conn=TCP:192.168.2.43:8802,Delimiter=1,ConnTimeout=5000,TimeOut=4000";

                        // Do as follows at the cereal connection.
                        // var option = "Conn=COM:3,ConnTimeout=5000,TimeOut=4000";
```

```
        // Connection to LR8400
        this.ctrl = this.ws.AddController("Sample",
        "CaoProv.HIOKI.LR8400",
        "",
        option);
}

private void btnGetVariable_Click(object sender, EventArgs e)
{
        // Addition of various Variable
        // System variable
        var tempVal1 = this.ctrl.AddVariable("@MAKER_NAME", null);
        var tempVal2 = this.ctrl.AddVariable("@VERSION", null);
        var tempVal3 = this.ctrl.AddVariable("@TITLE_COMMENT", null);
        // User variable
        var tempVal4 = this.ctrl.AddVariable("UI_1", "CH=CH1_1");
        var tempVal5 = this.ctrl.AddVariable("UR_1", "CH=CH1_1");
        var tempVal6 = this.ctrl.AddVariable("CCH_1", "CH=CH1_1");
        var tempVal7 = this.ctrl.AddVariable("MVREAL_CH1_1", "CH=CH1_1");
        var tempVal8 = this.ctrl.AddVariable("MTVREAL_1", "UNIT=UNIT1");
        var tempVal9 = this.ctrl.AddVariable("MTVRCH_1", "UNIT=UNIT1");

        // GetValue of various Variable
        // System variable
        var val1 = tempVal1.Value;
        var val2 = tempVal2.Value;
        var val3 = tempVal3.Value;
        // User variable
        var val4 = tempVal4.Value;;
        var val5 = tempVal5.Value;;
        var val6 = tempVal6.Value;;
        var val7 = tempVal7.Value;;
        var val8 = tempVal8.Value;;
        var val9 = tempVal9.Value;;
}
```

```
        private void btnExecute_Click(object sender, EventArgs e)
        {
                // Execute various Execute.
                var tempResult1 = this.ctrl.Execute("Send", ":START");
                var tempResult2 = this.ctrl.Execute("Send", ":STATUS?");
                var tempResult3 = this.ctrl.Execute("Start", null);
                var tempResult4 = this.ctrl.Execute("Status", null);
                var tempResult5 = this.ctrl.Execute("Stop", null);
                var tempResult6 = this.ctrl.Execute("Abort", null);
                var tempResult7 = this.ctrl.Execute("Error", null);
        }

        private void Sample_FormClosed(object sender, FormClosedEventArgs e)
        {
                // Opening of object related to Cao
                if (this.eng != null)
                {
                        this.eng.Dispose();
                        this.eng = null;
                }
        }
    }
}
```

5. Table for command of LR8400

Variable identifier	get_Value	set_Value
@TITLE_COMMENT	:COMMent:TITLe?	-
UI<??>	:UNIT:INMOde? ch$	-
UR<??>	:UNIT:RANGe? ch$	-
CCH<??>	:COMMent:CH? ch$	-
MVREAL<??>	:MEMory:VREAl? ch$	-
MTVREAL<??>	:MEMory:TVREAl? unit$	-
MTVRCH<??>	:MEMory:TVRCH? unit$	-

Method name of Execute	Command name
Start	:STARt
Status	:STATUS?
Stop	:STOP
Abort	:ABORT
Error	:ERRor?