USBRH Provider
Starwberry Linux USBRH-FG

Version 1.0.0

User's guide

September 13, 2016

Notes:

[Revision history]

Version	Date	Contents
1.0.0	2016-9-13	First edition.

Contents

1. Introduction

This is a user's guide of USBRH provider that is a CAO provider exclusively developed for USBRH-FG temperature humidity meters from Strawberry Linux. USBRH provider issues an instruction to USB-connected USBRH-FG through API of USBMeter.dll in order to obtain temperature and humidity information, control LEDs and heater, and obtain firmware version information.

Chapter 2 describes the outline of this provider and function specification, and chapter 3 describes commands implemented in CaoController::Execute.

2. Outline of provider

2.1. Outline

USBRH provider executes a command by using CaoController::Execute. CaoController::Execute can execute commands that are provided by USBMeter.dll.

Table 2-1 USBRH Provider

File Name	CaoProv.StrawberryLinux.USBRH2.dll
ProgID	CaoProv.StrawberryLinux.USBRH2
Registration	regsvr32 CaoProv.StrawberryLinux.USBRH2.dll
Deregistration	regsvr32 /u CaoProv.StrawberryLinux.USBRH2.dll

2.2. Method and property

2.2.1. CaoWorkspace::AddController method

USBRH provider establishes a connection to one USBRH-FG unit at the timing of AddController. USBRH provider cannot establish connections for multiple USBRH-FG units

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

bstrCtrlName : [in] Controller name (any name)
bstrProvName : [in] Provider name. Fixed to "CaoProv.StrawberryLinux.USBRH2"
bstrPcName : [in] Computer name where provider runs. (not used)
bstrOption : [in] Option strings (not used)

Example

```
// Connect to the controller
caoCtrl = caoWs.AddController("USBRH", "CaoProv.StrawberryLinux.USBRH2", "", "");
```

2.2.2. CaoController::Execute method

This method sends and receives a command in native mode. Specify a command name for the first argument and command parameter for the second argument. For details and usage of each command, refer to Chaper 3 Command Reference.

Syntax Execute(<bstrCommandName: VT_BSTR>, <vntName: VT_VARIANT>)

bstrCommandName : [in] Command name

vntParam : [in] Parameter

2.2.3. Error Code

The processing result from USBRH-FG at the method execution will be returned as HRESULT.

- If the method is executed properly (OK), S_OK(0) is returned.
- If the method is not executed properly (ER), an error code (from 0x80100000 to 0x80100004) is returned.

For provider common errors, refer to ORiN2 SDK Programming guide.

Table 2-2 Error code list

Error name	Error No.	Description
E_USBRHERROR_FINDUSB	0x80100000	Failed to identify USBRH-FG.
E_USBRHERROR_GETVERS	0x80100001	Failed to obtain the firmware version information of USBRH-FG.
E_USBRHERROR_CTRLIO	0x80100002	Failed to switch an LED.
E_USBRHERROR_SETHTR	0x80100003	Failed to switch the heater.
E_USBRHERROR_GETTEMP	0x80100004	Failed to obtain temperature and humidity information.

3. Command reference

This chapter describes each command of CaoController::Execute method. For detailed information about each command, refer to Function reference for USBRH module of Strawberry Linux.

Table 3-1 CaoController::Execute command list

Commands provided by USBMeter.dll	Command	Function	Page
GetVers	GetVers	Obtain the firmware version information of USBRH-FG.	7
ControlIO	ControlIO	Turn ON/OFF an LED.	8
SetHeater	SetHeater	Turn ON/OFF the heater.	8
GetTempHumidTrue	GetTempHumidTrue	Obtain the temperature and humidity information.	9

3.1. Controller class

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

Obtain the firmware version information of USBRH-FG.

Syntax caoCtrl.Execute("GetVers", "")

Argument : None

Return value : [out] Version information (VT_BSTR)

Example C# sample program

```
// Obtain the version information and display in to the message box.
object ver;
ver = caoCtrl.Execute("GetVers", "");

MessageBox.Show(ver.ToString());
```

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

Turn ON/OFF an LED. Both LEDs are OFF-status when USBRH-FG is connected.

Syntax caoCtrl.Execute("ControlIO", Array(<iLedNo>, <iLedState>))

<iLedNo> : [in] LED number
0: Green (factory default setting)
1: Red (factory default setting)

<iLedState> : [in] LED state
0: OFF
1: ON

Return value : None

Example C# sample program

```
// Turn On the green LED.
int[] ledNoState = new int[2]{0, 1};
caoCtrl.Execute("ControlIO", ledNoState);
```

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

Turn ON/OFF the heater. The heater is OFF-state when USBRH-FG is connected.

Syntax caoCtrl.Execute("SetHeater", <iHeaterState>)

< iHeaterState > : [in] Heater state
0: OFF
1: ON

Return value : None

Example C# sample program

```
// Turn ON the heater.
caoCtrl.Execute("SetHeater", 1);
```

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

Obtain the temperature and humidity information.

Syntax caoCtrl.Execute("GetTempHumidTrue", "")

Argument	:	None
Return value	:	[out] temperature [℃], humidity [%RH] (VT_R8 | VT_ARRAY)

Example C# sample program

```
// Obtain the temperature and humidity information and display it to the text box.
object tmp;
double[] TempHumid = new double[2];

tmp = caoCtrl.Execute("GetTempHumidTrue", "");
TempHumid = (double[])tmp;

tBoxTemp.Text = TempHumid[0].ToString();
tBoxHumid.Text = TempHumid[1].ToString();
```