

# USB RH プロバイダ Strawberry Linux USB RH-FG 用プロバイダ

Version 1.0.0

## ユーザーズ ガイド

July 17, 2012

【備考】

**【改版履歴】**

バージョン	日付	内容
1.0.0	2012-5-25	初版.
	2012-7-17	ドキュメントのバージョンルールを変更

## 目次

1. はじめに.....	4
2. プロバイダの概要.....	5
2.1. 概要.....	5
2.2. メソッド・プロパティ.....	5
2.2.1. CaoWorkspace::AddController メソッド.....	5
2.2.2. CaoController::Execute メソッド.....	5
2.2.3. CaoController::AddVariable メソッド.....	6
2.2.4. CaoVariable::put_value プロパティ.....	6
2.2.5. CaoVariable::get_value プロパティ.....	6
2.3. 変数一覧.....	6
2.3.1. コントローラクラス.....	6
3. コマンドリファレンス.....	8
3.1. 基本通信.....	8
3.1.1. CaoController::Execute(“GetTempHumidTrue”) コマンド.....	8
3.1.2. CaoController::Execute(“ControlIO”) コマンド.....	8
3.1.3. CaoController::Execute(“SetHeater”) コマンド.....	9

## 1. はじめに

本書は Strawberry Linux 社製 USB 温度・湿度計キットである USBRH-FG 用のプロバイダです。USB RH プロバイダは USB 接続された USBRH シリーズに対して USBMeter.dll の API 経由で命令を発行し、温度・湿度の取得と LED・ヒーターの制御を行うことができます。

## 2. プロバイダの概要

### 2.1. 概要

USB RH プロバイダは、コマンドの実行方法として `CaoController::Execute` と `CaoVariable` による 2 通りの方法を提供しています。

`CaoController::Execute` は `USBMeter.dll` で提供されているコマンドを実行する事ができます。

`CaoVariable` は温度と湿度を変数クラスとして実装しています。

### 2.2. メソッド・プロパティ

#### 2.2.1. `CaoWorkspace::AddController` メソッド

USB RH プロバイダでは `AddController` 時に、番号指定により複数の USB RH に接続する事ができます。オプション指定を省略した場合は初めに検出した 1 台目に接続します。

**書式** `AddController( <bstrCtrlName:VT_BSTR>,<bstrProvName:VT_BSTR>,<bstrPcName:VT_BSTR > [,<bstrOption:VT_BSTR>])`

`bstrCtrlName` : [in] コントローラ名 任意

`bstrProvName` : [in] プロバイダ名 固定値 =” `CaoProv.StrawberryLinux.USB RH`”

`bstrPcName` : [in] プロバイダの実行マシン名

`bstrOption` : [in] オプション文字列

以下にオプション文字列に指定するリストを示します。

**表 2-1 `CaoWorkspace::AddController` のオプション文字列**

オプション	意味
<code>Num=&lt;モジュール番号&gt;</code>	<p><code>Num:0~</code></p> <p>モジュール番号を指定します。</p> <p><code>USBMeter.dll</code> の <code>FindUSB</code> コマンドで見つけたモジュールに接続できます。どの順番にモジュールを見つけるかは <code>FindUSB</code> コマンドに依存します。</p> <p>(デフォルト:0)</p>

#### 2.2.2. `CaoController::Execute` メソッド

ネイティブモードのコマンドの送受信を行います。第 1 引数にコマンド名、第 2 引数にコマンドのパラメータを指定します。各コマンドの詳細は 3 章コマンドリファレンスを参照してください。

**書式** Execute ( <bstrCommandName:VT\_BSTR>,[<vntParam : VT\_VARIANT>])

bstrCommandName: [in] コマンド名  
vntParam : [in] パラメータ

Eecute メソッド実行した際の In-Sight からのステータスコードの戻り値は HRESULT として返されます。  
USBMeter.dll で発生したエラーは全て E\_FAIL で返るように実装されています。

### 2.2.3. CaoController::AddVariable メソッド

湿度と温度を単体で取得する変数を実装しています。作成された変数の get\_value が実行されると、内部で GetTempHumidTrue コマンドが実行され作成したオブジェクトに合わせて値を返しています。温度・湿度両方を取得する場合はなるべく CaoController::Execute の GetTempHumidTrue を利用してください。

**書式** AddVariable( <bstrVariableName:VT\_BSTR>,[< bstrOption: VT\_BSTR >])

bstrVariableName : [in] 変数名  
bstrOption : [in] オプション文字列

#### 使用例

```
Dim Temp as Object
Dim sngVal as Single
Set Temp = caoCtrl.AddVariable("@TEMPERATURE", "")
sngVal = Temp.Value

sngVal: 32.45
```

### 2.2.4. CaoVariable::put\_value プロパティ

現在変数クラスでは put\_value プロパティをサポートしていません。

### 2.2.5. CaoVariable::get\_value プロパティ

画像は表 2-2 コントローラクラス システム変数一覧のフォーマットで取得できます。

## 2.3. 変数一覧

### 2.3.1. コントローラクラス

表 2-2 コントローラクラス システム変数一覧

変数名	データ型	説明	属性	
			get	put

---

@TEMPERATURE	VT_R4	温度を取得します。	○	-
@HUMIDITY	VT_R4	湿度を取得します。	○	-

### 3. コマンドリファレンス

本章では CaoController::Execute メソッドの各コマンドについて解説します。各コマンドの詳細動作については Strawberry Linux 社の USB 温度・湿度計モジュール関数リファレンスを参照してください。

表 3-1 CaoController::Execute コマンド一覧

USBMeter.dll コマンド	コマンド	機能	
GetVers	-	未実装	-
GetTempHumidTrue	GetTempHumidTrue	温度と湿度を取得します。	P8
ControlIO	ControlIO	LED の制御を行います。	P8
SetHeater	SetHeater	ヒーターの制御を行います。	P9

#### 3.1. 基本通信

##### 3.1.1. CaoController::Execute(“GetTempHumidTrue”) コマンド

温度と湿度を取得します。

**書式**

GetTempHumidTrue()

引数 : 無し

戻り値 : [out] 温度(°C)と湿度(%RH) (VT\_R8 | VT\_ARRAY)

温度と湿度を配列で取得します。要素 1 が温度で要素 2 が湿度になります。

**使用例**

```
Dim vntResult as Variant
vntResult = caoCtrl.Execute("GetTempHumidTrue")
```

```
vntResult(0) : 34.12
vntResult(1) : 40.2360
```

##### 3.1.2. CaoController::Execute(“ControlIO”) コマンド

モジュールの LED を制御します。

**書式**

ControlIO (<ILEDNum>, <ILEDState>)

<ILEDNum> : [in] LED の番号 (VT\_I4)

0: 緑 (完成品出荷の場合)  
1: 赤 (完成品出荷の場合)

<lLEDState> : [in] LED の状態 (VT\_I4)

0: OFF  
1: ON

戻り値 : なし

**使用例**

---

```
' 緑色の LED を点灯させます  
caoCtrl.Execute "ControlIO", Array(0, 1)
```

---

**3.1.3. CaoController::Execute ("SetHeater") コマンド**

ヒーターを制御します。

**書式**

SetHeater (<lState> )

<lState> : [in] ヒーターの状態(VT\_I4)

0: OFF  
1: ON

戻り値 : なし

**使用例**

---

```
' ヒーターをつけます  
caoCtrl.Execute "SetHeater", 1
```

---