



IVI-COM 計測器ドライバ プログラミング・ガイド

(Windows Scripting Host / VBS 編)

Mar 2004 Revision 1.0

1- 概要

1-1 Windows Scripting Host での IVI-COMドライバの運用

Windows Scripting Host (WSH) は、Internet Explorer 4.x 以上で付いてくるスクリプト・エンジンです。本ガイドブックでは、Visual Basic Script (VBS)を使用して計測器ドライバを操作する方法を説明します。

VBS は Visual Basic 6.0 (VB6) と非常に良く似た BASIC 言語のスクリプトですが、本来 IVI-COM 計測器ドライバを使うのには適していません。それは、VBS では COM サーバへのアクセスに関して実行時バインディングしか使用できないからです。従って、VBS で制御したい COM サーバは IDispatch インターフェイス(別名オートメーション・インターフェイス)を装備する必要があります。残念ながら IVI-COM 計測器ドライバは IDispatch インターフェイスを装備せず、IUnknown から直接派生するカスタム・インターフェイスだけを装備するので、VBS から直接利用することはできません。

しかし、カスタム・インターフェイスを IDispatch インターフェイスにラップする特殊なツールが存在します。それが Script Adapter です。これを利用する事で、VBS から IVI-COM 計測器ドライバを使うことができます。

Notes:

ScriptAdapterはDLL形式のフリーウェアであり、
<http://homepage.interaccess.com/~hollp/ScriptAdapter.htm>からソースコード付で入手することができます。
弊社製 IVI-COM 計測器ドライバの最新版(VER1.1.x.x以降)にはビルド済みのCoScriptAdapter.DLLが含まれており、ドライバのセットアップ時に自動的にインストールされます。

CoScriptAdapter.DLL を手作業でセットアップする場合は、ハードディスクの任意のディレクトリにそれをコピーし、REGSVR32.EXE などを使用して自己登録(DllRegisterServer の呼び出し)を行う必要があります。
CoScriptAdapter.DLL が IVI-COM 計測器ドライバのセットアップ・プログラムによってインストールされる場合は、この作業は必要ありません。通常このファイルは/Program Files/IVI/BIN ディレクトリに置かれます。

本ガイドブックでは、IVI-COM KikusuiPlz 計測器ドライバ(KIKUSUI PLZ-4W/4WA シリーズ電子負荷装置)を使用する例を示します。他機種用の IVI-COM 計測器ドライバでも、ほぼ同様の手順で使用できます。

1-2 アプリケーションの作成

スクリプトの編集には任意のテキスト・エディターを使用することができますが、ここでは Notepad(メモ帳)を使用します。スクリプトのデバッグを行いたい場合は、Visual Basic 6.0 統合環境や、Visual Studio 2003.NET を使用することをお勧めします。Visual Basic 6.0 は BASIC 言語の仕様として VBS のスーパーセットになっています。

2- サンプルコード

Notepad 又は任意のテキスト・エディタを使用して下記のようなコードを書いてください。

```
Option Explicit

Dim adapter
Set adapter = CreateObject("ScriptAdapter.Adapter")

Dim ki
Set ki =
adapter.CreateAndWrap("Kikusui Plz. Kikusui Plz").QueryInterface("IKikusuiPlz")
ki.Initialize "ASRL1::INSTR", True, True, ""

Dim kiPws
Set kiPws = ki.Inputs

Dim kiPw
Set kiPw = kiPws.Item("")

kiPw.Function = 1
kiPw.CurrentLimit = 1.2
kiPw.SlewRate = 0.5
kiPw.Enabled = True

ki.Close
```

VBS のスクリプトは、HTML ドキュメントに埋め込むことも可能です。その場合には、<Script>タグを使用して、スクリプト部分を囲みます。スクリプトを HTML に埋め込んだ場合の利点は、Visual Studio.NET 又は Visual InterDev 6.0 でデバッグを行える事です。

```
<SCRIPT LANGUAGE="VBS">

' ここからスクリプトを書く
Option Explicit
Dim adapter
Set adapter = CreateObject("ScriptAdapter.Adapter")
...
...
ki.Close
' スクリプトはここまで

</SCRIPT>
```

2-1 オブジェクトの作成

まず最初に ScriptAdapter オブジェクトを作成する必要があります。オブジェクトを作成するには、CreateObject ステートメントを使います。Program ID として"ScriptAdapter.Adapter"を指定します。

引き続き、計測器ドライバ・オブジェクトを作成します。ここでは ScriptAdapter の CreateAndWrap メソッドを使います。これは、通常の CreateObject メソッドを呼び出し、更に ScriptAdapter の WrapObject メソッドを呼び出す事と等価です。CreateObject によって作成されたドライバ・オブジェクトは一旦 IIviDriver インターフェースを返しますが、ScriptAdapter は指定された COM インターフェース(この場合 IKikusuiPlz)を経由してラップされた IDispatch インターフェースを返します。

この部分は下記のように書き換えることもできます。

```

Dim o
Set o = CreateObject("Kikusui PI z. Kikusui PI z")
Dim ki
Set ki = adapter.WrapObject(o).QueryInterface("IKikusui PI z")

```

2-2 セッションの開始

計測器とのセッションを開始するには、Initialize メソッドを使用します。ここで、Initialize メソッドのパラメータについて説明しましょう。全ての IVI-COM 計測器ドライバは、IVI 仕様書で定義された Initialize メソッドを持っています。このメソッドには、以下のようなパラメータがあります。

Table 2-1 Initialize メソッドのパラメータ

パラメータ	タイプ	説明
ResourceName	String	VISA リソース名の文字列。計測器が接続されている I/O インターフェース、アドレスなどによって決定される。例えば、GPIB ボード 0 に接続されたプライマリ・アドレス 3 の計測器であれば、GPIB0::3::INSTR となる。
IdQuery	Boolean	TRUE を指定した場合、計測器に対して ID クエリを行う。
Reset	Boolean	TRUE を指定した場合、計測器の設定をリセットする。
OptionString	String	RangeCheck Cache Simulate QueryInstrStatus RecordCoercions Interchange Check に関する設定を、デフォルト以外に指定できる。更に、計測器ドライバが DriverSetup 機能をサポートする場合、その設定を行うことができる。

ResourceName には VISA リソースを指定します。IdQuery に TRUE を指定した場合は、計測器に対して "*IDN?" クエリなどを発行して機種情報を問い合わせます。Reset に TRUE を指定した場合は、"*RST" コマンドなどを発行して計測器の設定をリセットします。

OptionString には、2 つの機能があります。1 つは RangeCheck, Cache, Simulate, QueryInstrStatus, RecordCoercions, Interchange Check, などの IVI 定義の動作を設定します。もう 1 つは、計測器ドライバ毎に独自に定義される DriverSetup を指定します。OptionString は文字列パラメータなので、これらの設定は下のサンプルのような書式でなければなりません。

```
QueryInstrStatus = TRUE , Cache = TRUE , DriverSetup=12345
```

設定したい機能の名称及び設定値はケース・インセンシティブ(大文字と小文字の区別なし)です。設定値は Boolean 型なので、TRUE、FALSE、1、0 の何れかが有効です。複数の項目を設定する場合は、コンマで区切ります。OptionString パラメータで特に設定値を指定しない場合、IVI 仕様書で定義されたデフォルト値が適用されます。IVI 仕様書で定義されたデフォルト値は、RangeCheck と Cache だけが TRUE で、その他は全て FALSE です。

計測器ドライバによっては、DriverSetup パラメータが意味を持つ場合もあります。これは、IVI 仕様書では定義されない項目を Initialize の呼び出し時に指定するもので、利用目的や書式はドライバ依存です。従って DriverSetup の指定を行う場合、それは OptionString の最後

の項目として指定される必要があります。DriverSetup の指定内容はドライバ毎に異なるので、ドライバの Readme 文書又はオンライン・ヘルプなどを参照してください。

Notes:

上記の表では、パラメータのタイプとして String 型や Boolean 型と表記しましたが、これらは VB6 のような厳密な型を使用する言語での表現です。VBS では全ての変数は Variant 型となります。

2-3 チャンネルへのアクセス

IVI-COM 計測器ドライバでは一般に、電源装置や電子負荷装置などの計測器の場合、複数のチャンネルが装備されている事を前提に設計されています。従って、計測器のパネル設定に関する操作を計測器ドライバのルート・インターフェース(上記の例の場合、変数 `ki` がそれに相当する)から直接行うことは通常はありません。計測器の個々のチャンネルにアクセスするには、一旦 `Outputs`(又は `Inputs`)プロパティを通じてコレクションへの参照を取得し、更に `Item` メソッドで指定されたチャンネルへの参照を取得します。

```
Dim ki Pws
Set ki Pws = ki . Inputs
```

```
Dim ki Pw
Set ki Pw = ki Pws . Item("")
```

この例では電子負荷装置を操作する KikusuiPlz ドライバを使用するので、`Inputs` プロパティを使用します。PLZ-4W/4W シリーズは単一チャンネルの電子負荷装置なので、`Item` メソッドに渡すチャンネル名は空白(長さゼロの文字列)とします。複数チャンネルをサポートする環境では、"CH1" のような具体的なチャンネル名を使用する必要があります。実際に使用できるチャンネル名の詳細は、オンライン・ヘルプなどを参照してください。

特定チャンネルへの参照を取得できれば、あとは具体的な計測器設定を行うことができます。

```
ki Pw . Function = 1
ki Pw . CurrentLimit = 1.2
ki Pw . SlewRate = 0.5
ki Pw . Enabled = True
```

ここでは、ファンクションを CC モードに設定し、電流設定を 1.2A、スルーレートを 0.5 A/μs に設定し、そして入力を ON します。本来 `Function` プロパティは列挙型の数値を受け取りますが、スクリプト環境では列挙型のシンボルを使用できないので、直接数値を指定する必要があります。`Function = 1` は CC モードを意味します。使用できるプロパティやメソッドについての詳細はオンライン・ヘルプを参照してください。

2-4 セッションのクローズ

計測器ドライバによるセッションをクローズするには、`Close` メソッドを使います。

```
ki . Close
```

2-5 スクリプトの保存と実行

ここまでのコードだけで、とりあえず実行する事は可能です。まず、Notepad で編集したスクリプトを保存してください。その場合、拡張子をデフォルトの TXT ではなく VBS にします。(例えば `ex01.VBS` 等。) あとは保存された VBS ファイルを Explorer から実行するだけです。

プログラムを実行すると、即座に計測器との通信が開始されます。実際に計測器が接続されていて `Initialize` メソッドが成功した場合、このスクリプトは直ぐに終了します。通信に失敗した場合や、

VISA ライブラリの設定が正しく行われていない場合などは、COM 例外(WSH ランタイム・エラー)を発生します。

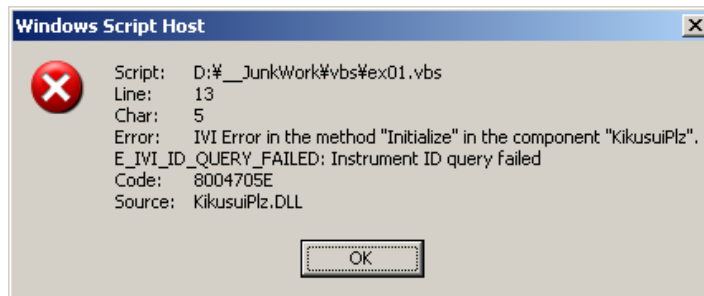


Figure 2-1 COM 例外

3- エラー処理

これまで示したサンプルでは、エラー処理を何も行っていませんでした。しかし実際には、範囲外の値をプロパティに設定したり、サポートされていない機能呼び出ししたりすると、計測器ドライバがエラーを発生する事があります。特に VBS のような実行時バインディングの環境ではプロパティやメソッドの名前やパラメータ個数を間違えても、実行時までチェックされません。その為、単純な記述ミスやスペルミスによるエラーも発生しやすくなっています。また、どんなに堅牢に設計・実装されたアプリケーションでも、計測器との I/O 通信エラーは避けることが出来ません。

IVI-COM 計測器ドライバでは、計測器ドライバ内で発生したエラーは全て COM 例外としてクライアント・プログラムに伝えられます。VBS の場合、COM 例外は通常実行時エラーになってしまいますが、On Error Resume Next ステートメントを使ってそれを抑制する事ができます。エラーが発生したかどうかの識別は、Err オブジェクトを通じて行うことができます。

```
On Error Resume Next
...
...

kiPw.CurrentLimit = 1.2
If err.Number <> 0 Then
    'エラー発生!!!
End If
```

IVI-COM 計測器ドライバ・プログラミング・ガイド

本ガイドブックに登場する製品名・会社名等は各社の商標または登録商標です。

©2004 Kikusui Electronics Corp. All Rights Reserved.