Using Application Programs for Audio Analyzer R&S® UPV in Remote Control

Application Note

Products:

- R&S®UPV
- R&S®UPV66
- R&S®UPV-K4
- R&S®UPP200
- R&S®UPP400
- R&S®UPP800

Most application notes for audio analyzer R&S® UPV include an application program which runs locally using the universal sequence controller option UPV-K1. This application note explains how sequence control programs can be modified to run on a remote PC using remote control option UPV-K4.

1GA60

Thomas Lechner

27th June 2012-0e

ROHDE&SCHWARZ

1 Principles of Remote Control

1.1 Local Sequence Control

Local sequence control on the audio analyzer R&S® UPV, further herein below called “UPV”, uses a dll called “UPx Server” to communicate with the UPV firmware via TCP. The dll is included in the software project as reference and provides methods to open and close the connection, to write ASCII data to the firmware and read ASCII data from the firmware. This mechanism is used to send SCPI commands from the application program to the UPV firmware and to read responses to SCPI queries from the UPV firmware.

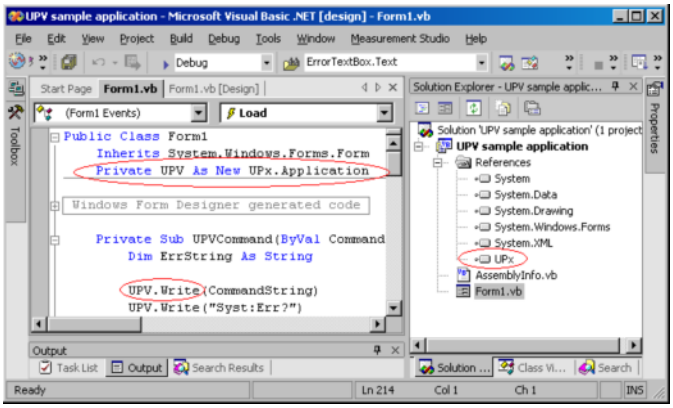

Figure 1: Local UPV sequence control via UPx server

1.2 Remote Control using UPV Drivers

UPV drivers provide a DLL with methods for specific instrument commands and queries which send the appropriate SCPI commands via VISA and remote control interface (GPIB, TCP/IP or USB) to the instrument. A language interface “rsupv.vb” provides a class library which can be included into a VB.NET project to call the driver functions.

As local sequence control works with SCPI commands and does not use specific drivers, this version of remote control programming is not compatible with the application programs provided for local sequence control.

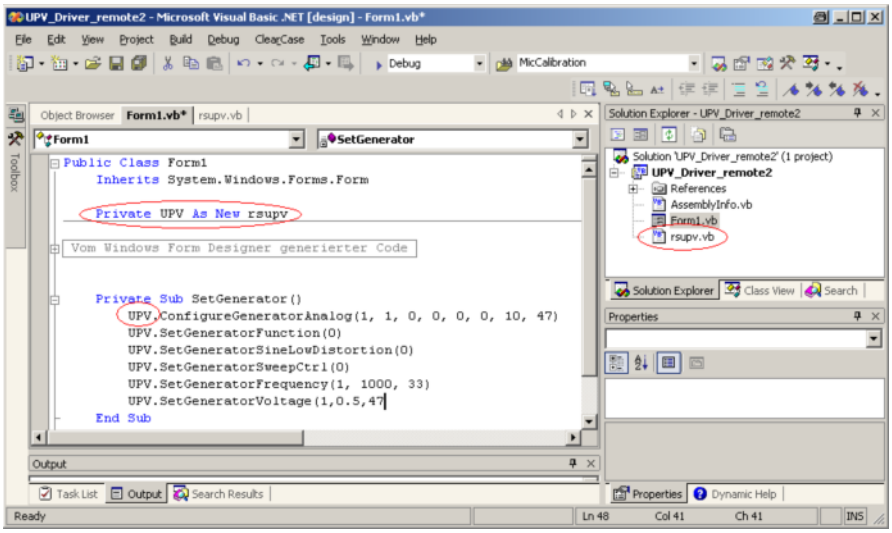

Figure 2: Remote control using the VXIplug&play driver and language interface "rsupv.vb"

1.3 Remote Control using VISA calls

VISA ("Virtual Instrument Software Architecture") is a software messaging layer between instrument drivers or remote control programs and the drivers of remote control interfaces. It provides standardized function calls which can be used for data transfer on the interface.

The class library provided with this application note uses a simplistic approach which is sufficient to provide the functionality of the UPx server. It calls functions of "InteropBASICFORMATTEDIOLib.dll" and "Interop.VisaComLib.dll" which are registered as COM objects when VISA software is installed.

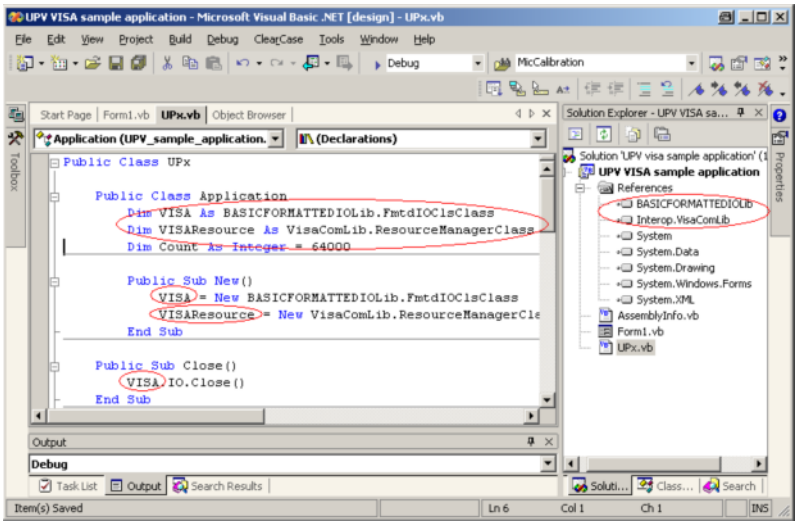

Figure 3: Class "UPx.VB" which connects to the VISA

The class library provides procedures “InitVisa”, “Write”, “Read” and “Close” and property “Timeout” corresponding to those of the UPx server. These procedures can be called in the application program for communication to the UPV firmware.

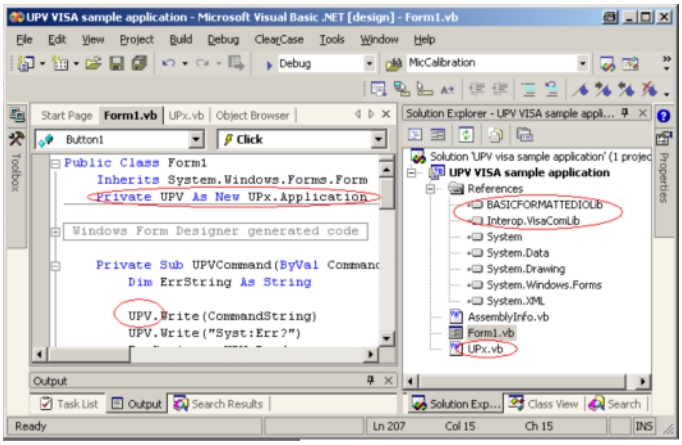

Figure 4: Remote control using VISA and class library “UPx.VB”

```
Public Class Application
    Dim VISA As BASICFORMATTEDIOLib.FmtdIOClsClass
    Dim VISAResource As VisaComLib.ResourceManagerClass
    Dim Count As Integer = 64000

    Public Sub New()
        VISA = New BASICFORMATTEDIOLib.FmtdIOClsClass
        VISAResource = New VisaComLib.ResourceManagerClass
    End Sub

    Public Sub Close()
        VISA.IO.Close()
    End Sub

    Public Sub InitVisa(ByVal ResourceName As String)
        VISA.IO = VISAResource.Open(ResourceName)
    End Sub

    Public Function Read() As String
        Return VISA.IO.ReadString(Count)
    End Function

    Public Property Timeout() As Integer
        Get
            Return VISA.IO.Timeout
        End Get
        Set(ByVal Value As Integer)
            VISA.IO.Timeout = Value
        End Set
    End Property

    Public Sub Write(ByVal Command As String)
        VISA.IO.WriteString(Command)
    End Sub
End Class
```

2 Example Program

2.1 Prerequisites

For remote control the UPV must be equipped with option UPV-K4.

The controlling PC must have VISA software installed. The application programs use version 2003, 2005 or 2008 of Visual Basic.NET. A development environment for one of these versions must be installed on the controlling PC. If a newer version of Visual Basic.NET is to be used, the project files can be automatically converted when the project is opened for the first time.

2.2 Encapsulated Communication Procedures

The following procedures and functions are used in the sample applications and application programs for communication with the UPV firmware.

2.2.1 Instantiate the Communication Server

An instance of the communication server is created by calling the constructor of the “UPx.Application” class:

```
Private UPV As New UPx.Application  ' Instantiate the driver
(Must be added to references first!)
```

2.2.2 Establish Connection

In sequence control, a connection is established by calling the “InitTCP” function:

```
UPV.InitTCP("localhost")                    ' Connect UPV
```

The string parameter must be “localhost”.

In remote control, the connection is established with

```
UPV.InitVISA("GPIB0::20::INSTR")            ' Connect UPV
```

The string parameter specifies the interface (GPIB 0) and address (20) of the instrument.

2.2.3 Set Timeout

The timeout specifies the time in milliseconds which the interface waits for a response from the instrument to a query. It is set with the procedure

```
UPV.Timeout = 100000
```

In this example the timeout is 100 seconds. If synchronization is achieved with the SCPI command “*WAI”, the timeout must be longer than the longest expected measurement duration.

2.2.4 Send Command

A SCPI command is sent as string data via the established connection. The encapsulated “UPVCommand” procedure additionally queries and checks the error queue:

```
Private Sub UPVCommand(ByVal CommandString As String)
'Sends a command string and returns success ("0") or error
string
    UPV.Write("Syst:Err?")          'Query error queue
    UPVErrorString = UPV.Read       'Read top entry of error
queue
    UPVError = Not UPVErrorString.StartsWith("0")
End Sub
```

2.2.5 Send Query and Read Response

The query command, again, is sent as string and the response is read also as string. After this has been completed, the error queue is checked:

```
Private Function UPVQuery(ByVal QueryString As String) As String
'Sends a query string and returns the answer string
   Dim AnswerString As String
   Dim ErrString As String

   UPV.Write(QueryString)
   AnswerString = UPV.Read
   UPV.Write("Syst:Err?")
'Query error queue
   ErrString = UPV.Read
   If Not ErrString.StartsWith("0") Then
'If an error exists, a message is displayed and the application
is ended
     MessageBox.Show("UPV Error" & ControlChars.CrLf & _
     ErrString, String.Empty, MessageBoxButtons.OK, _
     MessageBoxIcon.Exclamation, _
     MessageBoxDefaultButton.Button1, _
     MessageBoxOptions.DefaultDesktopOnly)
     Me.Close()
   End If
   Debug.WriteLine("UPV Query = " & QueryString & "  Answer = "_
   & AnswerString & "   Err = " & ErrString)
   Return AnswerString
End Function
```

2.2.6 Close Connection

The “Close” procedure breaks up the connection to the UPV firmware:

```
Private Sub Form1_Closing(ByVal sender As Object, ByVal e _
As System.ComponentModel.CancelEventArgs) Handles MyBase.Closing
    UPV.Close()
End Sub
```

2.3 Restrictions

Application programs which use the local file system of the UPV or the Windows® clipboard for transferring data between the UPV firmware and the application program cannot be directly converted for remote control.

File selector boxes (OpenFileDialog and SaveFileDialog) show the file system of the controlling PC instead of the UPV file system. These controls must be removed from the application program when converting it to a remote control program. File path names must be entered directly into text boxes.

2.4 Conversion of an Example Program from Sequence Control to Remote Control

In the following, step-by-step instructions are given how to convert a sequence control program written for the UPV in Visual Basic .NET to a remote control program. As an example, the “UPV Sample Application” is used, which can be found in “D:\UPV\VBexamples” on the UPV hard disk.

2.4.1 Remove Reference to UPxServer

Open the “References” in the solution explorer and right-click on “UPx”. Click “Remove” to remove the UPx Server from the project.

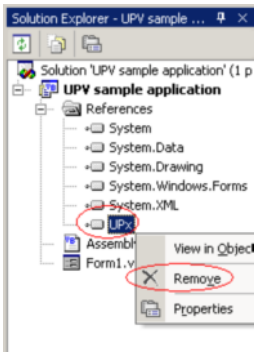

Figure 5: Removing the reference to the UPx server

2.4.2 Add Project “UPxVisaServer” to Solution

Project “UPxVisaServer” is provided as zip file with this application note. It must be unpacked and pasted to the Visual Studio Projects folder on the controlling PC.

Right-click on the solution in the solution explorer and click Add → Existing Project.

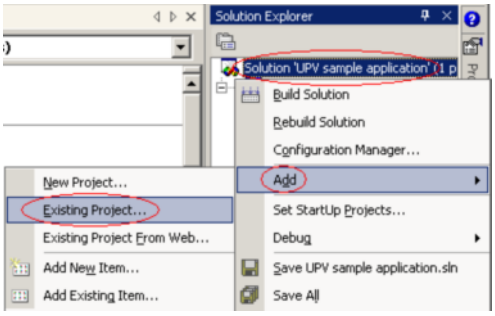

Figure 6: Adding an existing project to a solution

Browse to the file “UPxVisaServer.vbproj” and click “OK”. The project is added to the solution:

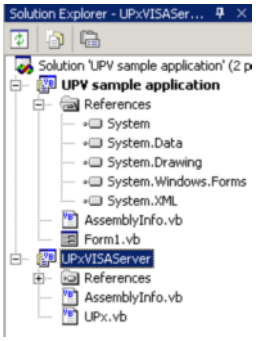

Figure 7: Solution with UPxVisaServer added

2.4.3 Add Reference to Project “UPxVisaServer” in Project

To make use of the functionality of the added project in the main project, a reference must be set.

Right-click on “References” of the main project and click “Add Reference”:

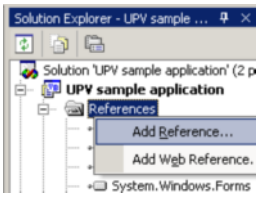

Figure 8: Adding a reference to the main project

Click the “Projects” Tab, click on the “UPxVisaServer” project and click the “Select” button. The project now appears in the “Selected Components” field. Click “OK” to finish.

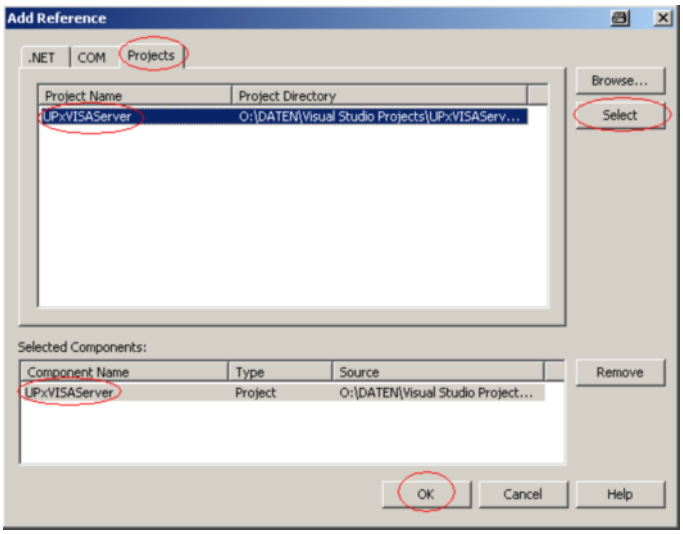

Figure 9: Selecting a project as reference

2.4.4 Include Namespace of Project “UPxVisaServer” in Project

The class library is now available in the main project. In order to be able to reference the functions directly without the project name as prefix, the namespace is included using the “Imports” statement on top of the source code.

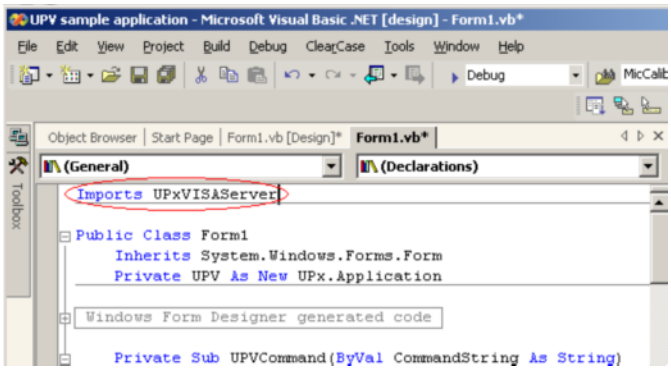

Figure 10: Including a namespace using the “Imports” statement

2.4.5 Change “InitTCP” to “InitVisa” and Enter an Ident-String for Instrument

The procedure “InitTCP” is not available in the UPxVisaServer. This call which is to open the remote control connection to the UPV firmware has to be replaced by “InitVisa”. The argument for the “InitVisa” call is a Visa-conformal ident string. For example, an instrument with remote address 20 connected to GPIB interface 0 is addressed by the ident string “GPIB0::20::INSTR”.

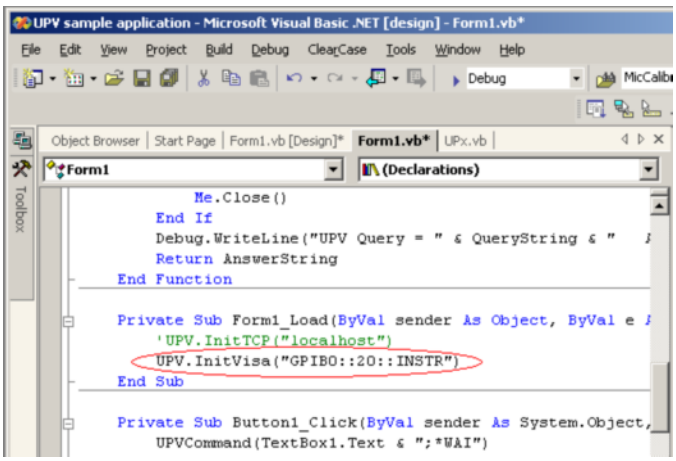

Figure 11: Initiating a remote connection to the instrument via GPIB

The program is not ready to run for remote control of a UPV with remote address 20 at GPIB interface 0.

3 Literature

- UPV Manual
- John M. Pieper: “Automatic Measurement Control, A tutorial on SCPI and IEEE 488.2”, Rohde & Schwarz
- Application Note 1GA47: “Calibrating Measuring Microphones and Sound Sources for Acoustic Measurements with Audio Analyzer R&S® UPV”
- Application Note 1GA50: “Calibration Tool for PESQ® Speech Quality Tests”
- Application Note 1GA52: “Averaging of 1/n Octave Spectra”
- Application Note 1GA56: “Swept Harmonic Distortion Measurements”

4 Ordering Information

Audio Analyzer		
Type	**Designation**	**Order No.**
UPV	Audio analyzer, analog interfaces, DC to 250 kHz	1146.2003.02
UPV66	Audio analyzer, without display, keyboard and CD drive	1146.2003.66
UPV-K1	Universal sequence controller	1401.7009.02
UPV-K4	Remote control for IEC625 /IEEE488, USB and LAN	1401.9001.02
UPP200	Audio Analyzer two channels	1411.1003.02
UPP400	Audio Analyzer four channels	1411.1003.04
UPP800	Audio Analyzer eight channels	1411.1003.08

About Rohde & Schwarz
Rohde & Schwarz is an independent group of companies specializing in electronics. It is a leading supplier of solutions in the fields of test and measurement, broadcasting, radiomonitoring and radiolocation, as well as secure communications. Established more than 75 years ago, Rohde & Schwarz has a global presence and a dedicated service network in over 70 countries. Company headquarters are in Munich, Germany.

Environmental commitment

- Energy-efficient products
- Continuous improvement in environmental sustainability
- ISO 14001-certified environmental management system

Certified Quality System
ISO 9001

Regional contact
Europe, Africa, Middle East
+49 89 4129 12345
customersupport@rohde-schwarz.com

North America
1-888-TEST-RSA (1-888-837-8772)
customer.support@rsa.rohde-schwarz.com

Latin America
+1-410-910-7988
customersupport.la@rohde-schwarz.com

Asia/Pacific
+65 65 13 04 88
customersupport.asia@rohde-schwarz.com

China
+86-800-810-8228 /+86-400-650-5896
customersupport.china@rohde-schwarz.com

Rohde & Schwarz GmbH & Co. KG
Mühldorfstraße 15 | D - 81671 München
Phone + 49 89 4129 - 0 | Fax + 49 89 4129 – 13777

www.rohde-schwarz.com