

オーディオ・アナライザ R&S[®]UPV 用印刷ツール アプリケーションノート

製品：

- | R&S[®]UPV
- | R&S[®]UPV66
- | R&S[®]UPV-K1

このアプリケーションノートでは、オーディオ・アナライザ R&S[®]UPV の測定結果のレポートを印刷するためのツールについて解説しています。この実行型ファイルは、結果のグラフや図をレポートに利用するために自由に構成を設定することができます。さらに、このファイルは、カスタム・アプリケーション・プログラムに組み込んで印刷機能を追加することができる、クラス・ライブラリの応用例でもあります。

目次

1	オーディオ・アナライザR&S®UPVの測定結果	4
1.1	数値結果	4
1.2	結果のグラフ	4
1.2.1	トレース・データの保存とクエリ	5
1.2.2	グラフへのトレース・データの書き戻し	5
1.2.3	グラフのハードコピー	6
1.2.4	グラフィック・プロファイル	7
1.2.4.1	グラフィック・プロファイルの目的	7
1.2.4.2	グラフィック・プロファイルの設定	7
1.2.4.3	グラフィック・プロファイルの編集と保存.....	7
2	VB.NETのWindows印刷機能	9
2.1	はじめに	9
2.2	PrintDocumentオブジェクト	9
2.3	PrintDialogコントロールとPrintPreviewコントロール.....	10
2.4	プリントアウトの開始	10
2.5	PrintPageイベント	10
2.6	ページの終了とドキュメントの終了	11
3	アプリケーション・プログラムの使用	12
3.1	インストール.....	12
3.2	アプリケーション・プログラムの起動.....	12
3.3	ユーザ・インタフェース.....	13
4	ソース・コードに関する作業	15
4.1	プロジェクト・ファイルの抽出と保存.....	15
4.2	ソリューションを開く	15
4.3	デバッガを使用したプログラム実行	15
5	他のアプリケーション・プログラムへのクラス・ライブラリ組み込み	16
5.1	ソース・コードへのクラス・ライブラリの追加	16
5.2	レポート・オブジェクトのインスタンス生成.....	16
5.3	プロパティ	16

5.4	印刷機能	17
5.5	イベント	19
6	参考文献.....	20
7	オーダー情報	21

1 オーディオ・アナライザ R&S®UPV の測定結果

1.1 数値結果

数値結果は、単一の数値からなる測定結果です。数値結果は、主要アナライザ機能の結果、入力モニタの結果、レベル・モニタの結果と周波数、周波数と位相の結果または周波数と群遅延の結果で構成されます。

それぞれの数値結果に対しては、対応する数値フィールドをダブルクリックすることによってコンビ・ディスプレイを表示することができます。コンビ・ディスプレイには数値結果の最大読取値と最小読取値が保存され、リミット値のチェックを行うことができます。最大および最小読取値は、それぞれ連続測定または単一測定の繰り返しによって得られた最大値と最小値です。これらの値は、「Start」キーを押すか、リモート・コマンド「INIT:CONT ON」を使用することによってリセットされます。さらに、実測値の最大値と最小値を棒グラフで表すことができます。

アプリケーション・プログラムにおける数値結果用の SCPI クエリ・コマンドは次の通りです。

SENSe<n>:DATA[1]|2,3,...16?

<n>は、結果の出所を指定します。

クエリ・パラメータ<n>					
装置タイプ	メイン・アナライザ機能	入力モニタ	周波数	位相／群遅延	レベル・モニタ
<n>	1	2	3	4	6

表 1: 数値結果のクエリ・パラメータ

2 番目のパラメータ [1]|2,3,...16 は、アナライザの入力チャンネルを指定します。

最大読取値と最小読取値は、次の SCPI コマンドを使用してクエリできます。

SENSe<n>:DATA[1]|2,3,...16?MIN

SENSe<n>:DATA[1]|2,3,...16?MAX

1.2 結果のグラフ

結果のグラフには、トレース、波形、スペクトラム (FFT、1/n オクターブ、高調波)、または MOS や時間経過に伴う遅延 (UPV-K61 オプションを使用した PESQ 測定) のような結果を含めることができます。UPV は、これらのデータを表示するためのグラフィック・ウィンドウを複数備えています。

1.2.1 トレース・データの保存とクエリ

トレース・データは、たとえばスプレッドシートへのエクスポートなどのために、それぞれのグラフ構成パネルの下側にあるフィールドから、ASCII フォーマットで保存することができます。グラフ構成パネルは、DispConfig メニューを使用するか、各グラフを右クリックして表示されたコンテキスト・メニューから「Config」を選ぶことによって、UPV ウィンドウ内に表示されます。

トレース・データは、リモート・コントロール・プログラムやシーケンス・コントロール・プログラムによるクエリが可能です (X 値と Y 値を個別にクエリ)。次のクエリ・コマンド、

TRACe:Subsys<i>:LOAD:AX|Y?

を使用すると、トレースのすべての X 値または Y 値を返します。「Subsys<i>」は特定のグラフを指定します。デフォルトでは、応答はカンマ区切りの ASCII フォーマットです。また、REAL フォーマットも可能です。このフォーマットはより効率的で、FFT スペクトラム・トレースや波形トレースのような大規模スキャンに適しています。詳細については、UPV マニュアルのリモート・コントロールに関する説明の「測定のトリガリングと結果の読み込み (Triggering Measurements and Reading Results)」という項を参照してください。

1.2.2 グラフへのトレース・データの書き戻し

アプリケーション・プログラムを使用して、そのアプリケーション・プログラムが計算または収集したトレース・データを、スイープ・グラフに表示したい場合もあります。一例としては、アプリケーション・プログラムが 2 つのトレースを読み込んでレベル差を計算し、その結果を新しいトレースに表示する場合があります。

スイープ・グラフにスイープを書き込むには、そのグラフを必要に応じて変更し、「X Source」を「Manual」に設定しなければなりません。これは、(SCPI) リモート・コマンドを使用するか、セットアップをロードすることによって行うことができます。対応する SCPI コマンドは次の通りです。

DISPlay:SWEp<i>:X:SOURce MAN

トレース・データは次のコマンドで書き込むことができます。

TRACe:Subsys<i>:STORE:AX|Y

スイープ・グラフ 1 にトレースを書き込むためのサンプル・コードを以下に示します。「UPVCommand」というメソッドは、UPV に引数文字列を送ってエラー状態をクエリします。

```
UPVCommand("DISP:SWE:X:SOUR MAN")
UPVCommand("DISP:SWE:X:SCAL MAN")
UPVCommand("DISP:SWE:X:AXIS FREQ")
UPVCommand("DISP:SWE:X:UNIT HZ")
UPVCommand("DISP:SWE:X:LEFT " & LeftValue.ToString & " HZ")
UPVCommand("DISP:SWE:X:RIGHT " & RightValue.ToString & " HZ")
UPVCommand("TRAC:SWE:STORE:AX " & SweepXString)
UPVCommand("TRAC:SWE:STORE:AY " & SweepYString)
```

X 値はスイープ・グラフにのみ書き込むことができます。FFT グラフや波形グラフの X 軸を希望通りスケールするには、Y データを書き込む前に FFT や波形を適切に変更する必要があります。つまり、サンプル・レートと FFT サイズや波形長を、それぞれ書き込む Y データのスケールに合わせなければなりません。

1.2.3 グラフのハードコピー

結果のグラフをプリントアウトしたりレポートに利用する場合は、ハードコピーを使用することができます。ハードコピーは UPV ウィンドウからビットマップ・コピーとして作成したり、現在アクティブなグラフのサイズと外観をさまざまな形式に変更することができます。ハードコピーは、Windows プリンタへ出力したりファイルへ保存したりできるほか、別の Windows プログラムでイメージとして使用するためにクリップボードへコピーすることも可能です。

ハードコピーの設定変更は「Config Panel」（設定パネル）で行います（メニュー項目は「Utilities → Config Panel」）。

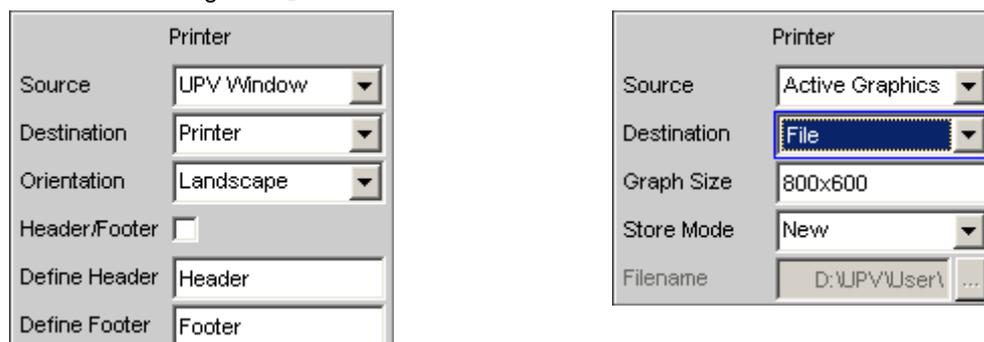


図 1：設定パネルによるハードコピー設定変更。左は UPV ウィンドウ全体を Windows プリンタへ出力。右はアクティブ・グラフをファイルへ保存。

ハードコピー機能をリモートで変更するための ASCII コマンドは、SCPI レコーダに簡単に記録できます（メニュー項目は「Sequence → SCPI Recording」）。記録は、フロントパネルの「Hcopy」キーを押して開始するか、ASCII コマンドを使用してリモートで開始することができます。

HCOPY[:IMMEDIATE]

アクティブ・グラフのハードコピーを作成するには、選択したグラフに Windows のフォーカスがなければなりません（ウィンドウのヘッダがグレーではなくブルー）。使用しているインタフェース上で各ウィンドウをクリックします。リモート・コントロールの場合にグラフ・ウィンドウにフォーカスを移動するには、リモートでそのウィンドウを開きます。ウィンドウが表示されている場合は、まずウィンドウを閉じるリモート・コマンドを送信してください：

DISPlay:SWEep:SHOW OFF

次に、下記のコマンドでウィンドウを開きます。

DISPlay:SWEep:SHOW ON

このウィンドウがプログラムが直近に扱ったウィンドウとなるので、フォーカスがこのウィンドウに移動し、ハードコピーが可能になります。

詳細については、UPV のマニュアルとアプリケーション・プログラムのソース・コードを参照してください。

1.2.4 グラフィック・プロファイル

1.2.4.1 グラフィック・プロファイルの目的

UPV グラフのフォーマット情報は、グラフィック・プロファイルと呼ばれるファイルに保存されます。このファイルの拡張子は「*.gpf」です。この情報には、すべてのオブジェクトの色、線の幅とスタイル、フォント、フォント・サイズ、プロパティなどが含まれます。

1.2.4.2 グラフィック・プロファイルの設定

グラフィック・プロファイルは、設定パネルを使用してスクリーン、プリンタ、ファイル、およびクリップボードを個別に設定することができます。スクリーン上での標準的な背景は黒ですが、印刷する場合は白の方が適しています。

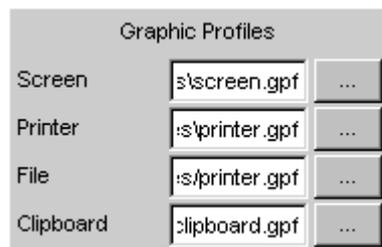


図 2 : UPV 設定パネルの「Graphic Profiles」フィールド

1.2.4.3 グラフィック・プロファイルの編集と保存

グラフィック・プロファイルの編集や作成を行うには、グラフィック・ウィンドウにフォーカスを移動して（マウスでヘッダをクリック）、「Utilities → Edit Graphic Profiles...」メニュー項目を選択します。グラフィック・プロファイル・エディタ・ウィンドウが開きます。このウィンドウには選択したグラフィック・ウィンドウのコピーが表示され、下側にはいくつかのコントロール機能が表示されます。



図3：グラフィック・プロファイル・エディタ

「Profile ...」ボタンをクリックして、既存のプロファイルを読みするか、新しいファイル名を指定してプロファイルを新規作成します。「Current Property」コンボボックスでプロパティを選択し、右側に値を入力するか、ボタンをクリックしてカラー・ダイアログまたはフォント・ダイアログを開きます。すべての変更は、コントロール機能の上に表示されたグラフィック・ウィンドウのコピーに反映されます。

必要な設定をすべて行ったら「OK」ボタンをクリックしてウィンドウを閉じ、現在のプロファイルを保存してください。

2 VB.NET の Windows 印刷機能

2.1 はじめに

Windows の印刷機能は、プリンタ・ドライバがインストールされたプリンタにプリントアウトを送信します。プリンタは、ローカル接続のものやネットワーク・プリンタ、あるいは Windows プリンタとしてインストールされた PDF Distiller を選択することができます。

Windows 印刷機能は VB.NET 環境を実装したオブジェクト指向の印刷機能です。プリンタへ文字をシリアル転送するのではなく、印刷ドキュメントを定義します。たとえば印刷ダイアログのコントロールを使用してプリントアウトを開始したり、印刷プレビューのコントロールを開くと、各印刷ページの開始時に「PrintPage」イベントが発生します。このイベントの引数は印刷ページのグラフィック用ハンドラを返答し、これに対しては、イベント・ハンドラ・コードによってアクセスと書き込みを行うことができます。プリントアウトの手順は以下の通りです：

- アプリケーション・プログラムは、印刷ドキュメントのインスタンスを生成します。
- アプリケーション・プログラムは、その印刷ドキュメントに対応する印刷ダイアログまたは印刷プレビュー・コントロールを開きます。
- PrintPage イベントは、印刷するページごとに発生します。
- アプリケーション・プログラムは PrintPage イベントのイベント・ハンドラを使って各ページに内容を書き込み、ページへの入力が完了すると印刷オブジェクトに制御を返答します。
- アプリケーション・プログラムは、グローバル変数を使って、1つのページの終了から次のページの開始までにプリントの進捗をメモリに記録しなければなりません。

印刷するドキュメントへの書き込みは、プリントアウトが開始されるまで行うことができません。データ（数値結果やスクリーンショット）が使用可能な状態になった時点からプリントアウトが開始される時点までの内容は、あらかじめ保存しておく必要があります。

以下では、アプリケーション・プログラムのコードを使った手順を示します。

2.2 PrintDocument オブジェクト

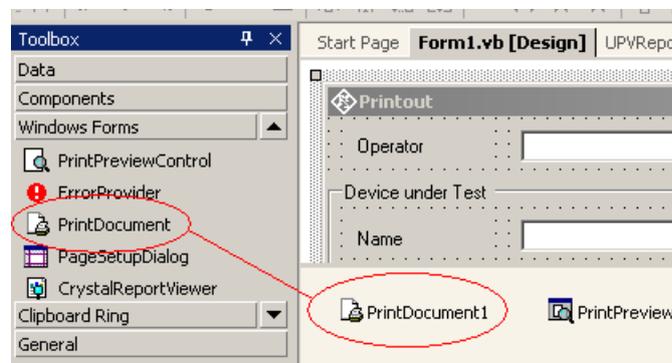


図4：Windows フォームへの PrintDocument 挿入

印刷ドキュメントは、ツールボックスからドラッグして、各ウィンドウのデザイナーにドロップします。

2.3 PrintDialog コントロールと PrintPreview コントロール

PrintDialog コントロールや PrintPreview コントロールは、ツールボックスからドラッグして各ウィンドウのデザイナーにドロップします。これは、次のコードによって開かれます。

```
With PrintDialog1
    .Document = PrintDocument1      'associate with PrintDocument
    .AllowSomePages = True          'allow printing of subset of pages
    If .ShowDialog = DialogResult.OK Then 'do the following of
the Ok button is clicked
        PrintDocument1.Print()      'print the document
    End If
End With
```

印刷ダイアログは、ユーザが「OK」か「Cancel」をクリックするまで開いたままになります。

```
With PrintPreviewDialog1
    .Document = PrintDocument1      'associate with PrintDocument
    .WindowState = FormWindowState.Maximized 'maximize print
preview window
    .ShowDialog()                    'show print preview window
End With
```

印刷プレビュー・ウィンドウは、ユーザの操作によって閉じられるまで開いたままになります。

2.4 プリントアウトの開始

ドキュメントのプリントアウトは、PrintDocument クラスの「Print」メソッドを使用することにより、PrintDialog や PrintPreviewDialog なしで開始することができます：

```
PrintDocument1.Print()
```

この時点では、PrintDocument の内容は空白です。ページの内容は、PrintPage イベントのイベント・ハンドラによって書き込まれます。

2.5 PrintPage イベント

PrintPage イベントのイベント・ハンドラは、PrintDocument のインスタンスを生成するコードの中に挿入されていなければなりません。イベント・ハンドラは、コード内で各イベントを発生させる際に呼び出されるメソッドです。



図5：PrintDocument の PrintPage イベントのためのイベント・ハンドラ

e は、印刷するページのグラフィック用のハンドルです。これによって、ページにグラフィック・コマンドを書き込むことができます。たとえば、

```
gr = e.Graphics      'graphics object of the page to be printed
```

```
gr.DrawString(HeadlineText, _HeadlineFont, _HeadlineBrush, _  
HeadlineRectangle) 'write formatted text  
  
gr.DrawLine(_LinePen, _PrintRect.Left, _CurrentYLocation, _  
_PrintRect.Right, _CurrentYLocation) 'draw horizontal line
```

といった例が考えられます。

2.6 ページの終了とドキュメントの終了

イベント・ハンドラが終了するとそのページも終了となり、実際のプリントアウトやプレビュー表示が開始されます。イベント・ハンドラの終了前に、さらにページが続くのかどうかを判断するために、「HasMorePages」というプロパティを設定する必要があります。

```
e.HasMorePages = True
```

このプロパティが真の場合は、イベント・ハンドラの終了後、次のページに対してもう一度「PrintPage」イベントが発生します。このプロパティが偽の場合は、プリントアウト・プロセスのローカル変数のクリーンアップやページ番号のリセットなどに使用できる「EndPrint」イベントが発生します。

3 アプリケーション・プログラムの使用

3.1 インストール

インストーラ「1GA57.msi」を UPV のハードディスクにコピーします（コピー先の例：D:\R&S_Software\Applications）。インストーラを実行して、表示される説明に従ってください。

デスクトップと、プログラム・メニューの「R&S UPV Applications」サブフォルダに、ショートカットがインストールされます。

このアプリケーション・プログラムを実行するには、UPV-K1 オプション「Universal Sequence Controller」をインストールする必要があります。さらに.NET Framework 2.0 が必要ですが、これはファームウェア・バージョン 2.0.0 以降のすべての UPV にインストールされています。このオプションとともに提供される Visual Basic .NET 開発環境は、インストールされたアプリケーション・プログラムの実行には必要ありません。

3.2 アプリケーション・プログラムの起動

プログラムは「1GA57 Printout」というショートカットから起動できます。このショートカットは、デスクトップの他、プログラム・メニューの「R&S UPV Applications」サブフォルダ内にもあります。

さらに、ツールバーにショートカット・ボタンを設定してプログラムを直接起動することも可能です。UPV ウィンドウ内のメニュー項目「Utilities → Quick Launch Config...」を使用し、行の右側にあるボタンをクリックして拡張子「*.exe」を選択し、さらに「C:\Program Files\Rohde&Schwarz\1GA57\Printout1.exe」を参照します。

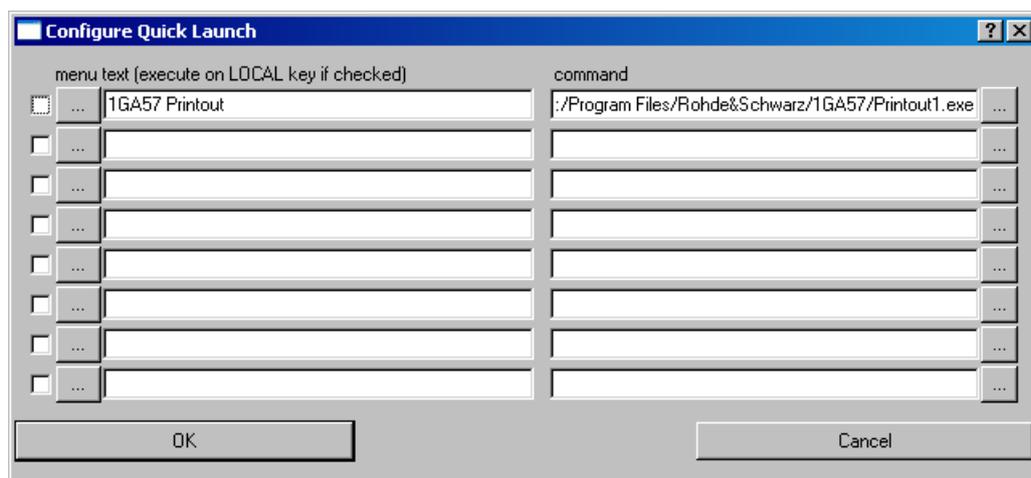


図6：クイック・ラウンチ・ボタン設定パネル

3.3 ユーザ・インタフェース

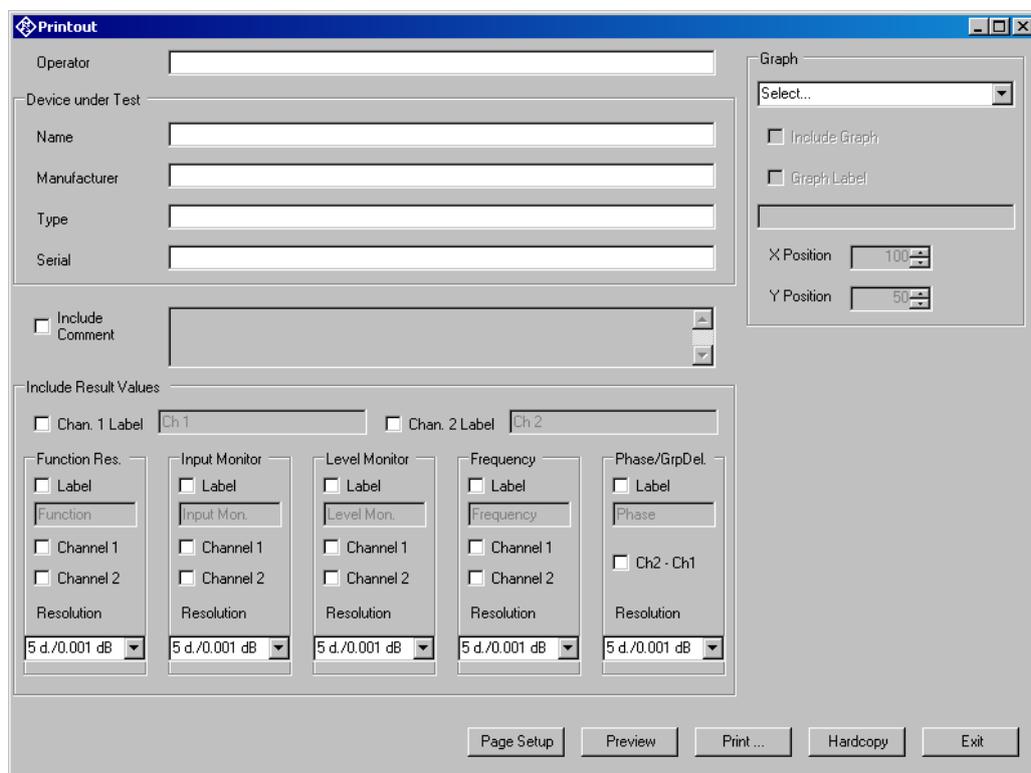


図7: アプリケーション・プログラムのユーザ・インタフェース

最上部にあるテキスト・ボックスには、オペレータや被測定物に関する情報のほか、必要に応じてコメントを入力することができます。

「Include Result Values」フィールドでは、レポートに含める数値結果を、ラベルやフォーマット等のオプションとともに指定することができます。ラベルは、UPV で設定された機能に関係なく使用できます。

「Graph」フィールドでは、1 つまたは複数の UPV グラフを選択して、レポートに含めることができます。ハードコピーにコメントを挿入することも可能です。

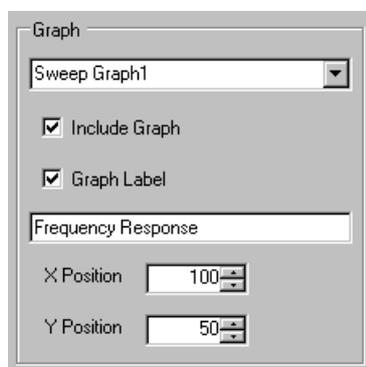


図8: ユーザ・インタフェースの「Graph」フィールド

一番下にある「Page Setup」ボタンを使用して、用紙のフォーマットやマージンなどを選択することができます。

「Preview」ボタンをクリックすると、レポートのプレビューが表示されます。印刷プレビュー・ウィンドウには、レポート印刷ボタンが表示されます。

「Print」ボタンをクリックすると Windows 印刷ダイアログが表示され、プリンタ、印刷範囲、およびページ数を選択することができます。

「Hardcopy」ボタンをクリックすると、レポートのプリントアウトを開始します。クリック以外の操作は不要です。

R&S Test Report



General

Date: 6/28/2011 5:51:54 PM
Operator: Thomas Lechner

Test Object

Name: Amplifier 1
Manufacturer: R&S
Type: RSAMP
Serial No.: 87658765

Comment

This is a sample printout

	RMS	Peak
Left	0.030834 V	0.14121 V
Right	0.030613 V	0.1402 V

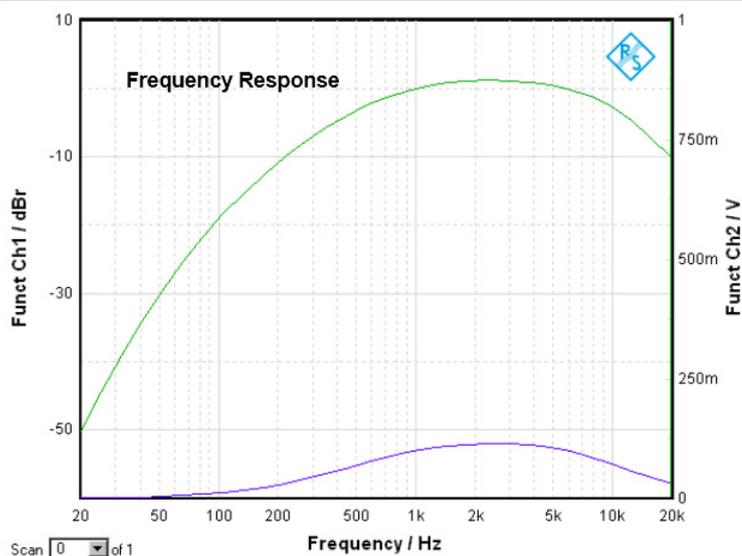


図9：アプリケーション・プログラムのプリントアウト例

4 ソース・コードに関する作業

4.1 プロジェクト・ファイルの抽出と保存

UPVにインストールされている Visual Basic .NET プログラミング環境のバージョン (xxxx) に応じ、zip 形式で圧縮されたソース・コード「1GA57_VBxxxx.zip」を選択してダウンロードします。UPV のハードディスク上で、この圧縮ファイルを解凍します。この際、パーテーションは「D:」を使用することを推奨します（「D:\Visual Studio Projects」または「D:\UPV\VBExamples」など）。

4.2 ソリューションを開く

プログラミング環境メニューの「File → Open Project」または、スタート・ページの「Open Project」ボタンを使用して「Printout1.sln」を探し、これを開きます。

ソース・コードを開くには、Solution Explorer で「Form1.vb」を右クリックして「View Code」をクリックします。グラフィック・ユーザ・インタフェースを編集するには、同じコンテキスト・メニューの「View Designer」を選択します。印刷クラス・ライブラリのコードは、「UPVReport.vb」に含まれています。

4.3 デバッガを使用したプログラム実行

デバッガを使用してプログラムを実行するには、IDE (Integrated Development Environment : 統合開発環境) の構成を「Debug」に設定する必要があります。メニュー項目「Debug → Start」か、デバッガのツールバーの起動ボタンを使用してプログラムを起動してください。デバッガを起動するごとにコードが新しくコンパイルされるので、ソース・コードの修正は直ちにプログラムの実行に反映されます。また、プログラムはデバッガ内で実行されるので、ソース・コードのエディタ・ウィンドウはロックされています。

5 他のアプリケーション・プログラムへのクラス・ライブラリ組み込み

5.1 ソース・コードへのクラス・ライブラリの追加

既存のプロジェクトにクラス・ライブラリを追加するには、Solution Explorer でプロジェクト名を右クリックしてコンテキスト・メニューから「Add」を選択し、サブメニューから「Add Existing Item ...」を選択します。さらに、このアプリケーションノートに付属しているソース・コードから「UPVReport.vb」を選択して、「OK」をクリックしてください。追加するプロジェクトのあるフォルダに、ファイルがコピーされます。

アプリケーション・プロジェクトへのクラス・ライブラリ組み込みの詳細例については、このアプリケーションノートに付属しているアプリケーション・プログラム「Form1.vb」のソース・コードを参照してください。

5.2 レポート・オブジェクトのインスタンス生成

コードを使用するには、プログラムのグローバル定義エリアにある「UPVReport」クラスからオブジェクトのインスタンスを生成しなければなりません。

```
Private WithEvents RPT As New UPVReport
```

クラス・ライブラリを使用するにはイベントが必要になるので、「WithEvents」を使ってインスタンスを生成することが不可欠です。

クラスには、PrintDocument と印刷機能に必要なすべてのコントロールが含まれています。さらに、ロゴ、見出し、表、イメージなどの内容をページに書き込むためのメソッドも備えています。

5.3 プロパティ

クラス定義「UPVReport」は、以下のパブリック・プロパティを提供します：

```
Public Property BoldTextFont() As System.Drawing.Font
Public Property CurrentPage() As Integer
Public Property CurrentYLocation() As Integer
Public Property HasMorePages() As Boolean
Public Property HeaderWidth() As Single
Public Property HeadlineBrush() As SolidBrush
Public Property HeadlineFont() As System.Drawing.Font
Public Property HeadlineSpace() As Single
Public Property ImageSpace() As Single
Public Property LabelBrush() As SolidBrush
Public Property LabelFont() As System.Drawing.Font
Public Property LinePen() As System.Drawing.Pen
```

```
Public Property LineSpace() As Single
Public Property Logo() As System.Drawing.Image
Public Property PreSubHeadlineSpace() As Single
Public Property SubHeadlineBrush() As SolidBrush
Public Property SubHeadlineFont() As System.Drawing.Font
Public Property SubHeadlineSpace() As Single
Public Property TableSpace() As Single
Public Property TextBrush() As SolidBrush
Public Property TextFont() As System.Drawing.Font
Public Property TextSpace() As Single
```

プロパティの設定または読み込みを行うには、レポート・オブジェクトの名前の最後にドット (.) を付けて入力し、プロパティの名前を続けます：

```
RPT.HasMorePages = True
```

5.4 印刷機能

以下に示すメソッドは、ドキュメントのセットアップ・ダイアログ、印刷ダイアログ、印刷プレビューの表示、印刷開始です：

```
Public Sub PageSetup()
Public Sub PrintDialog()
Public Sub PrintPreview()
Public Sub Print()
```

以下のメソッドは、印刷中のページへの内容書き込みに使用できます：

```
Public Function AppendImage(ByVal Img As System.Drawing.Image) _ As
PrintResult
```

```
Public Function AppendImage(ByVal Img As System.Drawing.Image, _ ByVal
Label As String, ByVal LabelXPosition As Integer, ByVal _
LabelYPosition As Integer) As PrintResult
```

```
Public Function AppendHeadline(ByVal HeadlineText As String) _ As
PrintResult
```

```
Public Function AppendSubHeadline(ByVal SubHeadlineText As _ String) As
PrintResult
```

```
Public Function AppendValueLine(ByVal HeaderText As String, _ ByVal
ValueText As String) As PrintResult
```

```
Public Function AppendTable(ByVal Tbl As ReportTable, ByVal _ Borders
As Boolean) As PrintResult
```

```
Public Function PlaceString(ByVal Text As String, ByVal _ RelXPosition
As Single, ByVal RelYPosition As Single, ByVal _ Width As Single) As
PrintResult
```

```
Public Function HorizontalLine() As PrintResult
```

戻り値「PrintResult」は、印刷操作の成功をレポートします：

```
Public Enum PrintResult
    Success
    PageFull
    PrintError
End Enum
```

「Success」は、そのページ上に情報を配置できたことを意味します。
「PageFull」は、情報を配置するための必要なスペースがページ上に残っていないことを意味します。この場合は「HasMorePages」を「True」に設定して現在のページの処理を完了させ、次のページに情報を配置する必要があります。
「PrintError」は、原因不明のエラーが発生してメソッドが現在のページ上に情報を配置できなかったことを意味します。

「ReportTable」は、表を定義して内容を書き込み、それぞれの印刷機能でそれを扱うための独自のクラス定義です：

```
Public Class ReportTable

    Structure TableRow
        Dim Cell() As String
    End Structure

    Public Sub New(ByVal ColumnCount As Byte, ByVal RowCount As _
        Byte, ByVal WithColumnHeader As Boolean, _
        ByVal WithRowHeader As Boolean)

    Public Property Cell(ByVal Row As Byte, ByVal Column As _
        Byte) As String

    Public Property ColumnHeader() As String()

    Public Property RowHeader() As String()

    Public Property HasRowHeader() As Boolean

    Public Property HasColumnHeader() As Boolean

    Public Property RowCount() As Byte

    Public Property ColumnCount() As Byte

End Class
```

表をページに出力するには、テーブル・オブジェクトのインスタンスを生成します：

```
Dim ResultValues As New UPVReport.ReportTable(ColumnCount, _
    RowCount, CatLabel, ChLabel)
```

次に、表のセル内容を書き込みます：

```
With ResultValues
    .ColumnHeader = ColumnHeader
    .RowHeader = RowHeader
    ...

End With
```

5.5 イベント

以下のイベントはクラスによって提供されます：

```
Public Event PrintPage()
Public Event EndPrint()
Public Event BeginPrint()
```

「PrintPage」イベントのイベント・ハンドラは、上記に示した印刷機能を使用して、現在印刷中のページの内容を書き込むために使用します。

```
Private Sub RPT_PrintPage() Handles RPT.PrintPage

    ... (Variable Definitions )

    Try
        With RPT
            If CurrentPosition = 0 Then
                .AppendHeadline("R&S Test Report")
                .AppendSubHeadline("General")
                .AppendValueLine("Date:", DateTime.Now.ToString)
                .AppendValueLine("Operator:", TextBox1.Text)
                .HorizontalLine()
                .AppendSubHeadline("Test Object")
                .AppendValueLine("Name:", TextBox2.Text)
                .AppendValueLine("Manufacturer:", TextBox3.Text)
                .AppendValueLine("Type:", TextBox4.Text)
                .AppendValueLine("Serial No.:", TextBox5.Text)
                .HorizontalLine()

                ...

            Catch ex As Exception
                Debug.WriteLine(ex.ToString)
                MessageBox.Show("Error in print preparation")
            End Try
        End With
    End Sub
```

6 参考文献

- R&S UPV オペレーティング・マニュアル
- Rod Stephens: “Visual Basic 2008 Programmer's Reference”, Wrox (February 5, 2008)
- Rakesh Rajan: “Visual Basic 2008 Recipes: A Problem-Solution Approach”, Apress (April 24, 2008)

7 オーダー情報

オーディオ・アナライザ		
タイプ	品名	オーダー番号
UPV	オーディオ・アナライザ、アナログ・インターフェース、DC~250kHz	1146.2003.02
UPV66	オーディオ・アナライザ（ディスプレイ、キーボード、CDドライブなし）	1146.2003.66
UPV-K1	汎用シーケンス・コントローラ	1401.7009.02

ローデ・シュワルツについて

ローデ・シュワルツ・グループ（本社：ドイツ・ミュンヘン）は、エレクトロニクス分野に特化し、電子計測、放送、無線通信の監視・探知および高品質な通信システムなどで世界をリードしています。

75 年以上前に創業し、世界 70 カ国以上で販売と保守・修理を展開している会社です。

ローデ・シュワルツ・ジャパン株式会社

本社／東京オフィス

〒160-0023 東京都新宿区西新宿 7-20-1

住友不動産西新宿ビル 27 階

TEL:03-5925-1288/1287 FAX:03-5925-1290/1285

神奈川オフィス

〒222-0033 神奈川県横浜市港北区新横浜 2-8-12

Attend on Tower 16 階

TEL : 045-477-3570 (代) FAX : 045-471-7678

大阪オフィス

〒564-0063 大阪府吹田市江坂町 1-23-20

TEK 第 2 ビル 8 階

TEL:06-6310-9651 (代) FAX:06-6330-9651

サービスセンター

〒330-0075 埼玉県さいたま市浦和区針ヶ谷 4-2-20

浦和テクノシティビル 3 階

TEL:048-829-8061 FAX:048-822-3156

E-mail: info.rsjp@rohde-schwarz.com

<http://www.rohde-schwarz.co.jp/>

Certified Quality System
ISO 9001
DQS REG. NO 1954 QM

Certified Environmental System
ISO 14001
DQS REG. NO 1954 UM

このアプリケーションノートと付属のプログラムは、ローデ・シュワルツのウェブサイトのダウンロード・エリアに記載されている諸条件に従ってのみ使用することができます。

掲載されている記事・図表などの無断転載を禁止します。

おことわりなしに掲載内容の一部を変更させていただくことがあります。あらかじめご了承ください。

ローデ・シュワルツ・ジャパン株式会社

〒160-0023 東京都新宿区西新宿 7-20-1 住友不動産西新宿ビル 27 階

TEL:03-5925-1288/1287 FAX:03-5925-1290/1285

www.rohde-schwarz.co.jp