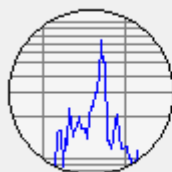




PIONEER HILL SOFTWARE



SPECTRA**PLUS**[®]-DT

Industrial Edition

FFT Spectrum Analyzer

ユーザースガイド

《 INDEX 》

1章 概要	頁
• Welcome to SpectraPLUS-DT Copy Protection	6
• スペクトラムアナライザーとは	
• 推奨環境	
2章 基本操作	7
基本機能	
• Modes(動作モード)	
• Plot(表示ビュー)	7
アナライザーのセットアップ	
• サウンドインターフェースデバイスのセット	
• デバイスの構成	8
• Frequency Range(周波数範囲)	
• Axis Scaling(グラフ軸スケール)	
• Open the Plot(ビューの開閉)	
• Select the operating mode(オペレーションモードの選択)	
• Start the analyzer(アナライザーのスタート)	
• Stop the analyzer and make a measurement(アナライザーの停止とカーソル測定)	9
• Print the results(データの印刷)	
• Save the configuration(定義ファイルの保存)	9
Cursor measurements(カーソル測定操作)	
• Left Mouse Button(マウス左ボタン)	
• Control Key + Left Mouse button(コントロールキーとマウス左ボタン)	
• Shift Key + Left Mouse button(シフトキーとマウス左ボタン)	
• Control Key + Shift Key + Left Mouse button(コントロールキーとシフトキーとマウス左ボタン)	
• Right Mouse Button(マウス右ボタン)	10
Right Mouse Click Actions(マウス右クリックメニュー)	
• Time Series Plot	
• Spectrum Plot	11
• Phase Plot	
• 3-D Surface Plot	
• Spectrogram Plot	
Printing the results(データの印刷)	13
Toolbars(ツールバー)	14
• Main Toolbar	
• Secondary Toolbar	
• Plot Toolbars	16
Status Bar(ステータスバー)	
Keyboard Shortcuts(キーボードショートカット)	
• Accelerator keys	
• The Return Key	
• The Focus	
3章 File Management(ファイル操作)	18
Opening and Saving Wave Files(WAV ファイル)	
• Enable Recording	
• Continuous Recording Capability(記録サイズ)	

Importing Files(ファイルのインポート)	19
<ul style="list-style-type: none"> • File Formats(ファイル仕様) • Sample Values(サンプル値) • File Header Options(ファイルヘッダオプション) • Data Scaling(データスケール) 	
Exporting Files(ファイルのエクスポート)	20
<ul style="list-style-type: none"> • File Formats(ファイル仕様) • Sample Values(サンプル値) 	
File Options(ファイルオプション)	
<ul style="list-style-type: none"> • Access modes(アクセスモード) • Make Backup(バックアップ) • Include Calibration Data in WAVfile(.WAV ファイル中の校正データ) • Real-Time Mode Data Buffering(リアルタイムモードでのデータバッファ) • Continuous Recording Capability(記録容量) 	
Set Paths(パス設定)	22
Configuration Files(定義ファイル)	
<ul style="list-style-type: none"> • Automatically Opening a Configuration file(定義ファイルの自動ロード) 	
4章 Editing(編集)	23
Cut/Copy/Paste commands(カット/コピー/ペースト)	
Play/Play Special commands(再生操作)	
Filtering(フィルタ)	24
<ul style="list-style-type: none"> • 5種類のフィルタ • Filter Sharpness • Filter Gain • フィルタレスポンスファイルの作成手順 	
Mute(ミュート)	25
Select All(全ての選択)	
Gain Adjust(ゲイン調整)	
DC Offset (DC オフセット)	
5章 Mode(動作モード)	27
Real-Time Mode(リアルタイムモード)	
Post-Process Mode(後処理モード)	
<ul style="list-style-type: none"> • Single step operation 	
Recorder Mode(レコーダモード)	
<ul style="list-style-type: none"> • Continuous Recording Capability(記録容量) 	
6章 Plots(表示グラフモード)	29
Time Series Plot(タイムシリーズ表示)	
Spectrum Plot(スペクトラム表示)	
Phase Plot(位相表示)	
Spectrogram Plot(スペクトログラム表示)	30
3-D Surface Plot(3次元表示)	
Single Channel and Composite Channel(シングルチャンネルとコンポジットチャンネル)	
<ul style="list-style-type: none"> • Selecting Single Channel Plots(シングルチャンネルモードの設定) • Composite Channel Processing(コンポジットチャンネル処理) <ul style="list-style-type: none"> Definitions(選択パラメータ) Composite Measurement Type(コンポジット測定タイプ) 	31

7章 Options (オプション)	33
Input Device Settings (入力デバイス設定)	
Output Device Settings (出力デバイス設定)	35
FFT Settings (FFT設定)	36
• FFT Size (FFT サイズ)	
• Smoothing Windows (窓関数)	
• Averaging Settings (アベレージ設定)	
Averaging Mode (アベレーシングモード)	
Averaging Type (アベレーシングタイプ)	
• Overlap Percentage (オーバーラップパーセンテージ)	
Scaling (Plot グラフのスケール)	39
• Amplitude Axis Scaling (Y 軸・レベルスケール)	
• Frequency Axis Scaling (X 軸・周波数スケール)	
• Power Spectral Density (PSD)	
• Standard Frequency Weighting (聴感補正カーブ)	40
• Transducer Compensation (周波数特性の補償)	
Calibration (アンプリチュードレベル校正)	
Triggering (トリガリング)	42
Run Control (ラン制御)	
8章 Utilities (ユーティリティ)	45
Signal Generator (シグナルジェネレータ)	
• Waveform types (波形の種類)	
• Save Signal To Wave File (WAV ファイルの保存)	
Peak Frequency (ピーク周波数)	48
Peak Amplitude (ピークアンプリチュード)	
Total Power (トータルパワー)	
Total Harmonic Distortion (全高調波ひずみ)	49
THD+N vs Frequency (全高調波ひずみ+ノイズ対周波数)	
InterModulation Distortion (IMD/混変調ひずみ)	51
Signal to Noise Ratio (SNR/SN 比)	
Signal to Noise and Distortion (SINAD)	
Reverberation Time (RT-60/残響時間)	52
• Reverberation Time	
• Reverberation Plots	
• Reverberation Options	
Equivalent Noise Levels (等価ノイズレベル)	54
Data Logging (データロギング)	
9章 Licensing (ライセンス)	57
• Software Key Method (ソフトウェアキー)	
• Hardware Key Method (ハードウェアキー)	
10章 Automation (自動化)	59
COM API – Initialization	
• Visual Basic	
• Visual C++	
• Microsoft Excel	
COM API - Control Functions	63
• Analyzer Operion	
•	

• Analyzer Configuration	63
• Calibration Commands	66
• Plot and Window Commands	67
• Overlay Control	69
• File Management	70
• File Positioning	
• File Post-Process	72
• Signal Generation	74
COM API - Read Data Functions	
• General Status	
• Peak Search Operation	
• Total Power Readings	76
• Distortion Readings	
• Marker Readings	
• Conversion Operations	
• Time Series Data	78
• Spectral Data	80
• Phase Data	82
• Leq Data	
	84
11章 Performance Issues (パフォーマンスの改善)	
• Processing Speed (処理速度)	
• Measurement Accuracy (測定確度・精度)	
• Aliasing (アリアシング)	
12章 Technical Support (サポート窓口)	85
サウンドデバイス ドライバーのインストール	86

1章 概要

Welcome to SpectraPLUS-DT:

SpectraPLUS-DT は高性能/パワフルなマルチチャンネル・スペクトラム・アナライザーです。本製品は Recording、Playback と Post-Process、Real-Time モードでの周波数特性、ひずみや伝達関数など種々のスペクトル解析機能を提供します。サウンドインターフェース(ADC)には指定デバイスをご使用頂きます。

ワイドレンジFFTサイズ、窓関数、デジタルフィルタ、オーバーラッププロセッシング、アベレージング、ピークホールド、トリガリング、デシメーション、ナローバンド/オクターブバンド(1/1,1/3,etc)スケーリングをサポートし、Time Series、Spectrum、Phase、3-D Surface、Spectrogram、THDとRT-60をグラフ表示します。そして、エクスポートとカラープリントアウトが可能です。シグナルジェネレータ機能はpink/white noise、swept sine、tones、pulses等の信号を生成します。すべての信号処理がCPUによって実行されますが、今日のコンピュータは十分な処理能力を持っています。

Copy Protection: このソフトウェアはライセンス情報を登録する「ハードウェアキー」(別名「 dongle」)を使用することで保護されています。ハードウェアキーはUSBポート装着タイプです。また、ご希望により「ソフトウェアキー」を選択することもできます。

スペクトラムアナライザーとは:

スペクトラムアナライザーはタイムドメイン(振幅時間)から周波数ドメイン(振幅頻度)に信号を変換するのに使用されます。オシロスコープになじみがあればタイムドメイン表示が何に似ているかを理解している筈です。周波数ドメイン表示はスペクトルとして知られています。単一トーンの測定でない場合、オシロスコープは周波数情報をわずかしか提供しません。しかしスペクトラムアナライザーは明確にこの情報を明らかにします。測定周波数限界は使用するサウンドインターフェースの能力に依存します。

推奨環境:

システム推奨条件は次の通りです。

CPU: 1GHz 以上

RAM: 1GB 以上

HD 空容量: 40MB 以上

モニター: 256色 VGA(1024 x 768 ドット)

ADC: 指定入力インターフェースデバイス(米・Data Translation DT-9800 シリーズ)

2章 基本操作

基本機能

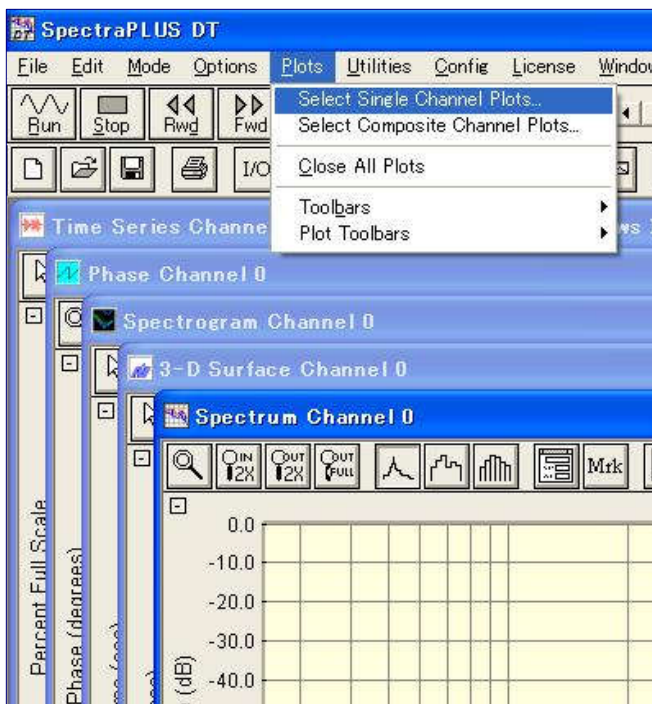
プログラムはサウンドインターフェースから信号を取得してFFT処理し、結果を表示します。プログラムは3つの動作モードと5つの表示ビュー(Plotと呼びます)を提供します。

Modes(動作モード):



- Real-Time: リアルタイムモードは、直接サウンドインターフェースから信号を取得して処理し、結果を表示します。
- Recorder: レコーダモードでは、デジタル化している信号をWAVファイル形式でハードディスクに保存することができます。サウンドインターフェース(標準的なサウンドカードの使用可)を経由してスピーカで信号を再生することができます。
- Post-Process: ポスト処理モードでは、ディスクに保存されたWAVファイルを後処理できます。このモードは他の2つのモードに比べ分析プロセスにより高い柔軟性があります。オーバーラップ処理ではSpectrogramと3-D Surface Plotの双方で時間分解を効果的にストレッチすることができます。

Plot(表示ビュー):



- Time Series Plot: デジタル化している信号波形を表示します。表示フォームはオシロスコープ(振幅対時間)と同様です。
- Spectrum Plot: 周波数対アンプリチュードを表示します。
- Phase Plot: 周波数対位相を表示します。
- Spectrogram Plot: 時系列のスペクトルをカラーかグレースケールでサーモグラフィ的(ソノグラフ)に表示します。3-D Surface Plotの俯瞰ビューです。
- 3-D Surface Plot: 時系列のスペクトルを3次元表示します。

この他に「THD+N vs Frequency」と「Reverbe(RT-60)」の2つのグラフィック表示ビューがあります。

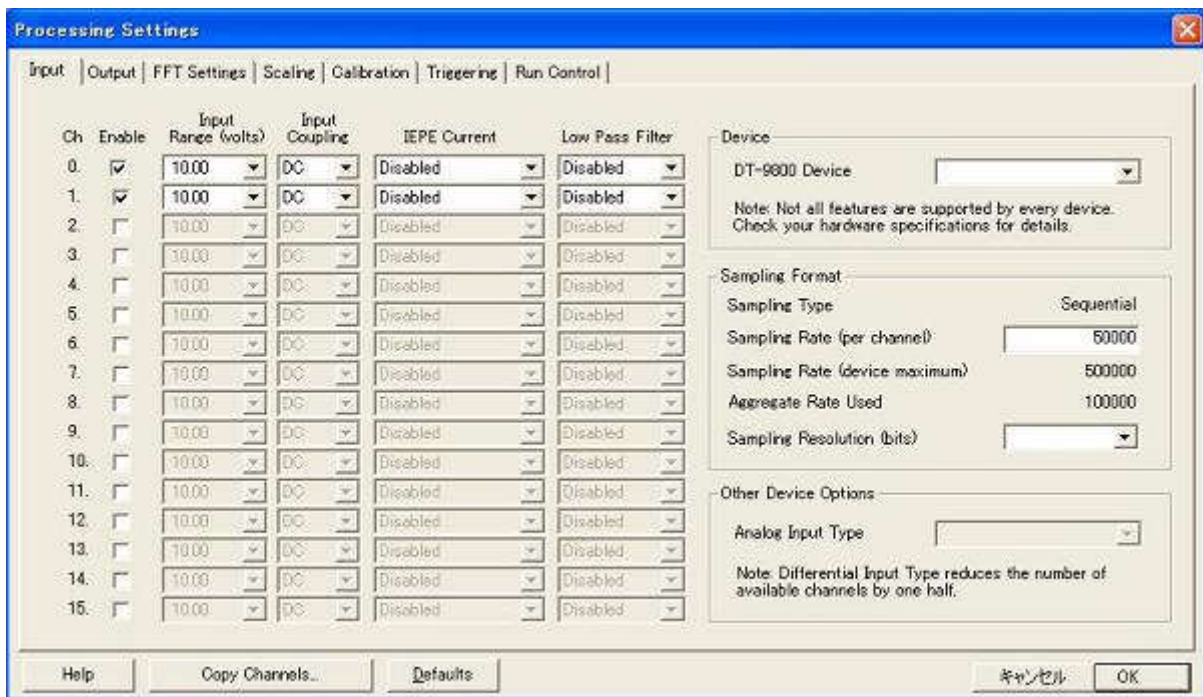
アナライザーのセットアップ

サウンドインターフェースデバイスのセット:

サウンドデバイスとそのドライバーをインストールします。そして、サウンドデバイスの信号入力ターミナルに被測定信号を接続します。ドライバーのインストールに関する記述は巻末を参照下さい。

デバイスの構成:

ハードウェアを構成するには<Options><Processing>メニューをクリックするか、<F4>キーを押します。もしくは「Input Device」セッティングダイアログを開くために[I/O]ツールバーボタンをクリックします。使用するDT-9800デバイスを選び、「Sampling Rate」を測定しようとする最も高い周波数の少なくとも2倍にセットします。そして、一つ以上の入力チャンネルを有効化(Enableをマーク)します。



Frequency Range (周波数範囲) :

測定に入る前に、必要な周波数帯域と周波数分解能を決める必要があります。ダイアログボックスの「Sampling Rate」を少なくとも測定上限周波数の2倍に設定します。そして「FFT Settings」タグで「FFT Size」を調整します。

Axis Scaling (グラフ軸スケール) :

「Scaling」タグで適切なスケーリングオプションを選択します。プログラムは「Logarithmic」スケーリングを既定値としています。

Open the Plot (ビューの開閉) :



<Plots><Select Single Channel Plots>メニュー（もしくは<F5>キー）をクリックして「Single Channel Plot Selection」ダイアログを開き、目的のチャンネルとそのPlot（ビュー）を選択します。ラン中にもPlotを開閉できますが、3-D SurfaceとSpectrogram Plotでは閉じると内容が消失します。

Select the operating mode (オペレーションモードの選択) :

後処理するために被測定信号を保存する場合は「Recorder」モードを選択します。信号を保存しないでモニターモードでランする場合は「Real-Time」モードを選択して下さい。「Recorder」モードはコンピュータのハードディスク空容量やメモリサイズによって制限されます(選択されたFile Optionsにもよりますが)、「Post-Process」モードでは記録された「.WAV」ファイルフォーマットが使われます。

Start the analyzer (アナライザーのスタート) :

ツールバーの<Run>か<Rec>ボタンをクリックするか、アクセラレータキー<Alt><R>を使用してプログラムを起動すると、Plot表示がアップデートし始めます。

Stop the analyzer and make a measurement (アナライザーの停止とカーソル測定):

ツールバーの<Stop>か、アクセラレータキー<Alt><S>をクリックするとプログラムが停止します。マウスを使用してグラフ上の任意のポイントで左ボタンを押して下さい。クリックポイントの数値を表示するカーソルボックスが現れます(詳細はCursor Measurements項参照)。

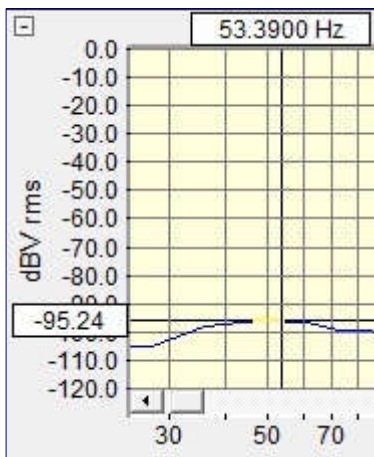
Print the results (データの印刷):

<File><Print>メニューを選択するとアクティブ(選択されている)Plotを印刷します。また、印刷ページに印字される注釈を入力することができます。

Save the configuration (定義ファイルの保存):

<Config><Save Configuration>メニューを使用することによって、現在設定されているアナライザーの構成パラメータをファイル保存することができます。ファイルにはビューサイズ、位置、開いている.WAVファイルなどの情報パラメータが書き込まれます。

Cursor Measurements (カーソル測定操作)



Left Mouse Button (マウス左ボタン):

Plot上でマウス左ボタンを押すと、ポイント位置のX軸とY軸の値をカーソルボックスに表示します。

Control Key + Left Mouse button (コントロールキーとマウス左ボタン):

<Ctrl>キーをホールドしながらPlot上でマウス左ボタンをクリックし、そしてドラッグすると、基点とマウスポイント間の相対値をカーソルボックスに表示します。これは2つのポイント間の差分を観るのに役立ちます。

Shift Key + Left Mouse button (シフトキーとマウス左ボタン):

<Shift>キーをホールドしながらPlot上でマウス左ボタンをクリックするとハーモニクカーソルが表示されます。このカーソルは10次の高調波を示しています。これは非常に複雑なスペクトル中の高調波成分を特定するのに役立ちます。

ちます。

Control Key + Shift Key + Left Mouse button (コントロールキーとシフトキーとマウス左ボタン):

<Ctrl>と<Shift>キーの両方をホールドしながらPlot上でマウス左ボタンをクリックすると、サイドバンドカーソルが表示されます。これらのカーソルはスタートポイント上下サイドの測定線を表示します。これは非常に複雑なスペクトルの中で変調するコンポーネントを特定することに役に立ちます。

Right Mouse Button (マウス右ボタン):

Plot上でマウスの右ボタンをクリックしてホールドすると、編集オプションを含むポップアップメニューが現れます。このポップアップメニューの内容はビューにより異なります(詳細はRight Click Action Menu項参照)。

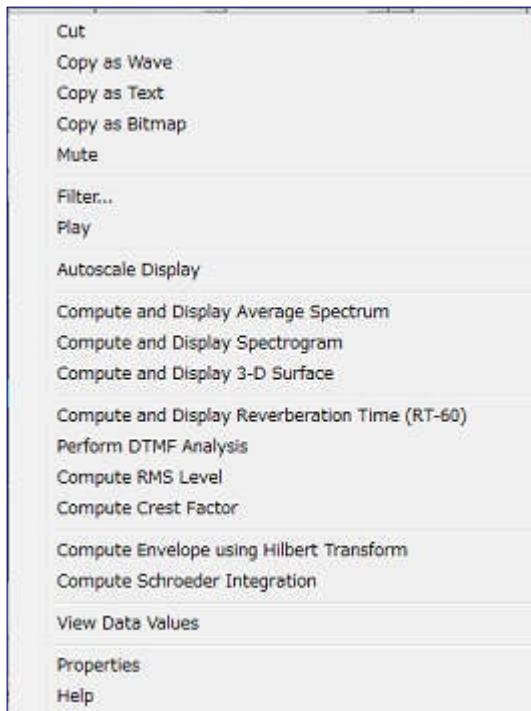
SpectrumとPhase Plotではアンプリチュード軸に沿ってカーソルをトレース可能です。表示オプションダイアログボックスで無効に設定することができます。

Notes:

- マウス操作している間はアナライザーを停止することをお勧めします。対応するキーボードコマンドはありません。
- 3-Dsurface Plotは正確な周波数を確定する為にグラフのX軸沿って測定しなければなりません、ピークの先端の表示は左に歪曲します。

Right Click Action Menus(右クリックポップアップメニュー)

各Plot上でマウス右ボタンをクリックするとポップアップメニューが現れます。アイテムのいくつかは慣れ親しんだクリップボード操作ですが、その他に多くの強力な後処理機能を提供します。



Time Series Plot:

- Cut - 選択されたタイムセグメントをカットし、.WAV ファイルフォーマットでそれをクリップボードにコピーします。
- Copy As Wave - 選択されたタイムセグメントを.WAV ファイルフォーマットでクリップボードにコピーします。
- Copy as Bitmap - Plot イメージをビットマップフォーマットでクリップボードにコピーします。
- Copy As Text - 選択されたタイムセグメントをタブで区切られたテキスト形式でクリップボードにコピーします。
- Save To Wave File - 選択したタイムセグメントを新しいウェーブファイルに保存します(ファイル名の入力を求められます)。
- Write To Bitmap File - プロット画像をビットマップファイルに書き込みます(ファイル名と場所の入力を求められます)。
- Mute - 選択されたタイムセグメントをミュート処理します。
- Filter - 選択されたタイムセグメントをフィルタ処理します。
- Play - 出力デバイスを介して、選択されたタイムセグメントを再生します。
- Autoscale Display - データ内容を評価し、Plot の「Plot Top/Range」を最適値に設定します。
- Pull Cursor To Trace - このオプションを有効にすると、測定カーソルは自動的に振幅軸に沿った波形トレースに「pull 追従」します。
- Compute and Display Average Spectrum - 「infinite」アベレージングを使い、選択されたタイムセグメントのアベレージング Spectrum を計算して表示します。もし必要なら Spectrum Plot が開かれます。
- Compute and Display Spectrogram - 現在の FFT サイズで、選択されたタイムセグメントの Spectrogram を処理して表示します。オーバーラップパーセンテージは、タイムセグメントが Spectrogram Plot を満たすように使用されます。もし必要なら Spectrogram Plot が開かれます。
- Compute and Display 3-D Surface - 現在の FFT サイズで、選択されたタイムセグメントの 3-D Surface を処理して表示します。オーバーラップパーセンテージは、タイムセグメントが 3-D Surface Plot を満たすように使用されます。もし必要なら 3-D Surface Plot が開かれます。
- Compute and Display RT-60 - 選択されたタイムセグメントを処理して RT-60 を表示します。もし必要なら Reverberation Time Plot が開かれます。
- Perform DTMF Analysis - 選択されたタイムセグメントの DTMF キー(タッチトーン)をデコードします。
- Compute RMS, Max, Min, and Peak to Peak Levels - 選択された時間セグメントの RMS レベル、最大/最小と Peak to Peak 値を計算します。
- Compute RMS Level - 選択されたタイムセグメントの RMS レベルを算出します。

- Compute Crest Factor - データセットのクレストファクターを処理して表示します。
- Compute Envelope Using Hilbert Transform - Hilbert Transform を使い、選択されたタイムセグメントの Envelope(大きさ)を計算します。結果は Time Series Plot に表示されます。[zoom out full]ボタンをクリックすると通常表示モードに戻ります。
- Compute Schroeder Integration - M.R.シュローダー法に基づくリバーシインテグレーション(right to left)を計算します。結果は Time Series Plot に表示されます。[zoom out full]ボタンをクリックすると通常表示モードに戻ります。
- Compute Cumulative Sound Exposure Level (C-SEL) - 選択した時間セグメントの前方積分のプロットを作成します。Time Series プロットでは、対数 amplitude スケーリングを使用して表示するのが最適です。この測定は、杭打ちなどの水中騒音評価でよく使用されます。
- Export Peak Values to Text File - 選択した時間セグメント内のすべてのイベントの時間、ピーク、および rms レベルを含むテキストファイルを作成します。閾値とファイル名の入力を求められます。
- View Data Values - 現在の Time Series 値をテーブル表示します。表示は常にアップデートされます。
- Properties - ディスプレイオプションダイアログボックスを起動します。
- Help - オンラインヘルプを表示します。

Spectrum Plot:

- Copy as Text - 選択されたタイムセグメントを、タブで区切られたテキスト形式でクリップボードにコピーします。
- Write to Text File - 表示された Spectrum 値を ASCII テキストファイルに書き込みます。
- Copy as Bitmap - Plot イメージをビットマップフォーマットでクリップボードにコピーします。
- Write To Bitmap File - プロット画像をビットマップファイルに書き込みます (ファイル名と場所を聞かれます)。
- Inverse FFT - Inverse FFT を実行して、結果を Time Series Plot に表示します。
- Compute Cepstrum - スペクトラルデータに FFT を実行します。
- Bandwidth and Q Factor - 最大ピークのハーフパワーバンド幅と Q Factor を計算して表示します。
- Clear Spectrum - Spectrum をクリアします。ピークホールド表示を観察したり、印刷する場合に効果的です。
- Clear Peak Hold - Spectrum をクリアします。ピークホールド表示なしでオーバーレイデータを観察したり、印刷する場合に効果的です。
- Autoscale Spectrum - データを評価して、Plot の「Plot Top/Range」を最適値に設定します。
- Set Marker N - マウスクリックポイントの周波数位置にマーカ N を設定します。
- Clear Marker N - マーカ N をクリアします。
- Marker Options - Marker Options ダイアログボックスを呼び出します。
- Smooth Spectrum N - スペクトラルデータ(Narrowband スケーリングと Logarithmic アンプリチュードスケールリング)をスムージングするために N を平均化します。
- View DataValues - 現在の Spectrum 値をテーブル表示します。表示は常にアップデートされます。

- Properties - ディスプレイオプションダイアログを起動します。
- Help - オンラインヘルプを表示します。

Phase Plot:

- Copy as Text - Phase 値をタブで区切られたテキスト形式でクリップボードにコピーします。
- Copy as Bitmap - Plot イメージをビットマップフォーマットでクリップボードにコピーします。
- Write To Bitmap File - プロット画像をビットマップファイルに書き込みます (ファイル名と場所の入力を求められます)。
- Pull Cursor To Trace - このオプションを有効にすると、測定カーソルは自動的に位相軸に沿ったトレースに「追従/プル」します。
- View Data Values - 現在の Phase 値をテーブル表示します。表示は常にアップデートされます。
- Properties - ディスプレイオプションダイアログを起動します。
- Help - オンラインヘルプを表示します。

3-D Surface Plot:

- Cut - 選択されたタイムセグメントをカットし、.WAV ファイルフォーマットでそれをクリップボードにコピーします。
- Copy as Wave - 選択されたタイムセグメントを.WAV ファイルフォーマットでクリップボードにコピーします。
- Copy as Bitmap - Plot イメージをビットマップフォーマットでクリップボードにコピーします。
- Write To Bitmap File - プロット画像をビットマップファイルに書き込みます (ファイル名と場所を聞かれます)。
- Mute - 選択されたタイムセグメントをミュート処理します。
- Filter - 選択されたタイムセグメントをフィルタ処理します。
- Play - 出力デバイスを介して、選択されたタイムセグメントを再生します。
- Autoscale 3-D Surface - データを評価して、「Plot Top/Range」値を最適値に設定します。
- Display Time Series - 選択されたセグメントに対応する Time Series を表示します。もし必要なら、Time Series Plot が開かれます。
- Compute and Display Average Spectrum - 「infinite」アベレージングを使い選択されたタイムセグメントのアベレージング Spectrum を計算して表示します。もし必要なら Spectrum Plot が開かれます。
- Compute and Display Spectrogram - 現在の FFT サイズで、選択されたタイムセグメントの Spectrogram を処理して表示します。オーバーラップパーセンテージは、タイムセグメントが Spectrogram Plot を満たすように使用されます。もし必要なら Spectrogram Plot が開かれます。
- Expand 3-D Surface - 選択したタイムセグメントのデータを再処理することによって、3-D Surface Plot を拡大します。オーバーラップパーセンテージは、タイムセグメントが 3-D Surface Plot を満たすように使用されます。
- Properties - ディスプレイオプションダイアログを起動します。
- Help - オンラインヘルプを表示します。

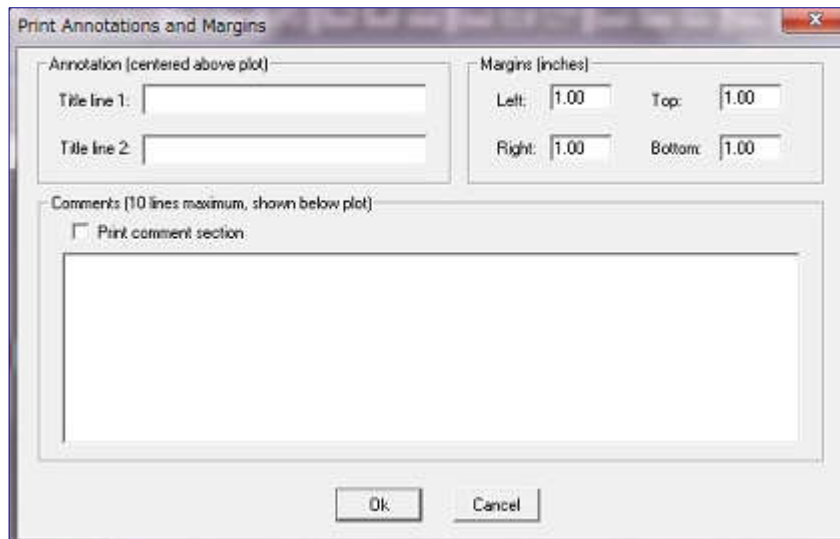
Spectrogram Plot:

- Cut - 選択されたタイムセグメントをカットし、.WAV ファイルフォーマットでそれをクリップボードにコピーします。
- Copy as Wave - 選択されたタイムセグメントを.WAV ファイルフォーマットでクリップボードにコピーします。
- Copy as Bitmap - Plot イメージをビットマップフォーマットでクリップボードにコピーします。
- Write To Bitmap File - プロット画像をビットマップファイルに書き込みます (ファイル名と場所を聞かれます)。
- Mute - 選択されたタイムセグメントをミュート処理します。
- Filter - 選択されたタイムセグメントをフィルタ処理します。
- Play - 出力デバイスを介して選択されたタイムセグメントを再生します。
- Autoscale Spectrogram - データを評価して、「Plot Top/Range」値を最適値に設定します。
- Display Time Series - 選択されたセグメントに対応する Time Series を表示します。もし必要なら Time Series Plot が開かれます。
- Compute and Display Average Spectrum - 「infinite」アベレージングを使い選択されたタイムセグメントのアベレージング Spectrum を計算して表示します。もし必要なら Spectrum Plot が開かれます。
- Expand Spectrogram - 選択したタイムセグメントのデータを再処理することによって、Spectrogram plot を拡大します。オーバーラップパーセンテージはタイムセグメントが Spectrogram Plot を満たすように使用されます。
- Compute and Display 3-D Surface - 現在の FFT サイズで、選択されたタイムセグメントの 3-D Surface を処理して表示します。オーバーラップパーセンテージは、タイムセグメントが 3-D Surface Plot を満たすように使用されます。もし必要なら 3-D Surface Plot が開かれます。
- Display Spectrum Plot - 現在のマウスの位置に対応する時刻に、基礎となるスペクトルデータを表示します (ダブルクリックと同じ)。
- Display Order Plot - 現在のマウスの位置に対応する RPM で、基本となるオーダベースのスペクトルデータを表示します (ダブルクリックと同じ)。このメニューコマンドを使用するには、Time Series 軸で RPM が選択されている必要があります。
- Properties - ディスプレイオプションダイアログを起動します。
- Help - オンラインヘルプを表示します。

Printing the results(測定データの印刷)

一度に1つのPlotを印刷できます。印刷対象Plotをクリックすることによってウィンドウをアクティブにします。<File>メニューからプリントオプションを選択し必要に応じてプリントオプションを設定します。Print Annotation and Merginsダイアログにアクセスすると以下の設定が行えます。

- Annotation: 2行の注釈スペースが提供されます。プリントシートのヘッダライン中央に印字されます。
- Margins: 余白スペースを任意に設定します。
- Comments: グラフ下部に最大10行の注釈を入れることができます。

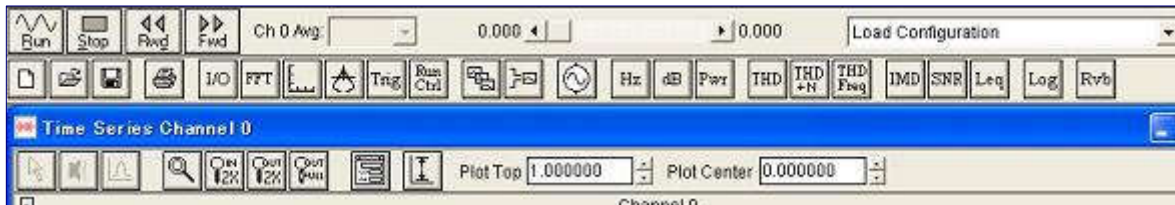


Notes:

白黒プリンタでカラーのSpectrogramを印刷するときは、オプションダイアログボックスでグレースケールオプションを選択してください。白黒プリンタでカラーの3-D Surfaceを印刷するときは、オプションダイアログボックスで背景色を白に、そしてラインを黒に設定して下さい。ペンプロッターはSpectrogramと3-D Surface Plotに対応しません。

Toolbars(ツールバー)

メインアプリケーションツールバー、セコンダリツールバーとインディビデュアルプロットツールバーの3つのツールバーが用意されています。



Main Toolbar(メインツールバー):

メイン画面とPlot画面それぞれにツールバーを配しています。メインアプリケーションツールバーの機能は動作モードで変わります。ツールバーにはプログラムをコントロールするアイコン(Run/Stop/etc)が配置されています。そして「Real-Time」と「Post-Process」モードではさらに、「Avg」と「Peak Hold」を設定するコントロールボックスが配されます(「Processing Settings」ダイアログボックスにも配されています)。「Recorder」と「Post-Process」モードでは水平スクロールバーが配置されます。バー左側は「.WAV」ファイルの現位置時間を、右側はトータル時間を表します。

Real-Time モード:

- <Run> - アナライザー(ジェネレータが起動していれば双方)をスタートします
- <Stop> - アナライザー(ジェネレータが起動していれば双方)を停止します
- <Avg> - アベレージングブロックサイズを設定します

Recorder モード:

- <Rec> - 測定データを記録します(ジェネレータが起動していればスタートします)
- <Stop> - 記録あるいは再生を停止します(ジェネレータが起動していれば停止します)
- <Fwd> - 現位置時間をファイルの最後尾に移動します(早送り)
- <Rwd> - 現位置時間をファイルの頭部に移動します(巻き戻し)

- ・ <Avg> - アベレージングブロックサイズを設定します
- ・ <Scroll> - 現位置時間を移動します

Post-Process モード:

- ・ <Run> - アナライザーをスタートします
- ・ <Stop> - アナライザーを停止します
- ・ <Rwd> - 現位置時間をファイルの頭部に移動します(巻き戻し)
- ・ <Avg> - アベレージングブロックサイズを設定します
- ・ <Scroll> - 現位置時間を移動します

Secondary Toolbar(セカンダリツールバー):

セカンダリツールバーはメインツールバー直下のアイコンボタンが配されているバーです。

- ・ Record New Wave File - File Managementダイアログボックスを開きます
- ・ Open one or more Wave File - File Managementダイアログボックスを開きます
- ・ Save one or more Wave file - File Managementダイアログボックスを開きます
- ・ Print Current Plot - アクティブなビューを印刷します(time series, spectrum, phase, spectrogram, or 3-D surface)
- ・ [I/O]Device Options – Processing SettingsダイアログボックスのInputタグを開きます
- ・ [FFT]FFT Processing Settings - Processing SettingsダイアログボックスのFFT Settingsタグを開きます
- ・ Scaling Options - Processing SettingsダイアログボックスのScalingタグを開きます
- ・ Amplitude Calibration Settings - Processing SettingsダイアログボックスのCalibrationタグを開きます
- ・ [Trig]Triggering Options - Processing SettingsダイアログボックスのTriggeringタグを開きます
- ・ [Run Ctrl]Run Control Options - Processing SettingsダイアログボックスのRun Controlタグを開きます
- ・ Select Single Channel Plots - Single Channel Plot Selectionダイアログボックスを開きます
- ・ Select Composite Channel Plots - Composite Channel Plot Settingsダイアログボックスを開きます
- ・ Signal Generator - Signal Generator機能を閉じます
- ・ [Hz]Peak Frequency - Peak Frequency表示機能を閉じます
- ・ [dB]Peak Amplitude – Peak Level表示機能を閉じます
- ・ [Pwr]Total power – Total power表示機能を閉じます
- ・ [THD]Total Harmonic Distortion - THD表示機能を閉じます
- ・ [THD+N]Total Harmonic Distortion + Noise - THD+N表示機能を閉じます
- ・ [THD Freq]THD+N vs Frequency - THD+N vs Frequency表示ウィンドウを開閉します
- ・ [IMD]InterModulation Distortion - IMD表示機能を閉じます
- ・ [SNR]Signal to Noise Ratio - SNR表示機能を閉じます
- ・ [Leq] – Equivalent Noise Levels – Leq Analysis表示機能を閉じます
- ・ [Log]Data Logging Setup - Data Logging Settingsダイアログボックスを開きます
- ・ 「Rvb」Reverberation Utility - Reverberation Time(RT-60) 表示ウィンドウを開きます

Plot Toolbars(プロットツールバー):

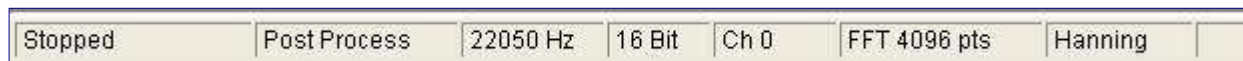
プロットツールバーは各Plotにアイコンボタンが配置されているバーです。前図はTime series Plotのバーを示しています。

- ・ Select a time segment for editing - 編集するタイムセグメントをグラフィカルに選択することができます矢印アイコンをクリックしデータの任意の範囲をクリックアンドドラッグします。マウスボタンをリリースすると指定したタイムセグメントを反転表示します。編集機能(playback, filtering, cut/copy/paste)を使用可能です
- ・ Play currently selected time segment - 選択したタイムセグメントを再生します。最初にタイムセグメントを指定して置かなければなりません。メニューバーの<Edit><Play>を使っても同様に機能します
- ・ Filter currently selected time segmen - 選択したタイムセグメントをフィルタ処理します。最初にタイム

セグメントを指定して置かなければなりません。メニューバーの<Edit><Filter>を使っても同様に機能します

- Zoom in along time axis - 選択した範囲を拡大(ズームイン)表示することができます。まずボタンをクリックし、表示される「□+」カーソルを任意のポジション(周波数/タイム軸)に移動します。そして任意の範囲をクリックアンドドラッグし、マウスボタンをリリースすると拡大表示します
- [IN 2x]Zoom In by a factor of 2x - 水平軸感度を倍に拡大します
- [OUT 2x]Zoom Out by factor of 2X - 水平軸感度を 1/2 倍戻します
- [OUT FULL]Zoom Out to show entire data set - 水平軸感度をノーマルに戻し、フルスパン表示します
- Set additional ... display options - Plot Options ダイアログボックスを起動します。<Options><Plot Options>メニューでもアクセスできます
- Display Spectrum as line graph - Spectrum を折れ線グラフモードで表示します
- Display Spectrum as stepped bar graph - Spectrum をステップバーグラフモードで表示します
- Set additional spectrum display options - Spectrum をバーグラフモードで表示します
- [Mrk]Marker Options - Marker Options ダイアログボックスを起動します
- Autoscale current ... display - 当該 Plot 垂直軸の感度/レンジを自動的に最適化します
- Plot Top - Plot のアンプリチュード/垂直軸のトップ(最上部)の感度を設定します。同様の機能は「Display Options」ダイアログボックスにもあります。上下矢印キーをクリックして調整します。もしくは直接キー入力します
- Plot Range - アンプリチュード/垂直軸の表示レンジを設定します。同様の機能は「Display Options」ダイアログボックスにもあります。上下矢印キーをクリックして調整します。もしくは直接キー入力します
- Hiding the Toolbar - ツールバー左下に配された小さな「+」「-」ボタンにより、バーの ON/OFF が可能です。<Plots><Plot Toolbars>メニューからもアクセス可能です

Status Bar(ステータスバー)



ステータスバーはメイン画面の下部に配置されます。バー上にはいくつかの設定パラメータが表示されます。

- Run Status: 動作状態を表示します(Stopped/Running/etc)
- Operating Mode: 設定されている動作モードを表示します
- Sampling Frequency: 設定されているサンプリング周波数を表示します
- Sampling Format: 設定されているサンプリングフォーマットを表示します
- Channel: アクティブなチャンネル ID (Single Channel: 0...15/Composit Channel: A...P)を表示します
- FFT Size: 設定されているFFTサイズを表示します
- Smoothing Window: 設定されているスムージングウィンドウ(窓関数)を表示します

Keyboard Shortcuts(キーボードショートカット)

操作アイコンはメインツールバーに配置されていますが、各アイコンのラベルには1文字だけ下線が引かれています。<Alt>キーと<下線文字>キーを一緒に押すと、そのアイコンと同一の機能を得られます(Windowsに準拠)。

ショートカットキーは各メニュー、サブメニューに対しても効果的に機能します。例えば<Alt>-<V>を押し、続けて<Alt>-<T>を押すとTime Series Plotにアクセスすることができます。

次にメニュー選択に便利なショートカットキーを紹介します。

Accelerator keys:

F1: Helpを起動します
F4: 「Processing Setting」ダイアログボックスを表示します
F5: 「Display Options」ダイアログボックスを表示します
F7: 「Save Configuration File」ダイアログボックスを表示します
F8: 「Load Configuration File」ダイアログボックスを表示します
F11: 「Signal Generator」機能を起動します

<Alt+R>: アナライザーをランします
<Alt+S>: アナライザーをストップします

<Ctrl+O>: 「Open Wave File」ダイアログボックスを表示します(Recorder、Post-Processモード時)
<Ctrl+S>: 上書き保存のため「Save Wave File」ダイアログボックスを呼び出します(Recorderモード時)
<Ctrl+A>: 保存のため「Save Wave File」ダイアログボックスを表示します(Recorderモード時)
<Ctrl+P>: 「印刷」ダイアログボックスを表示します
<Alt+F4>: プログラムを終了します

<Ctrl+Z>: 直前の編集を元に戻します(アンドウー)
<Ctrl+Del>: 編集指定部を切り取ります
<Ctrl+Ins>: 編集指定部をコピーします
<Shift+Ins>: 編集指定部を貼り付け挿入します
<Ctrl+L>: スペシャルループ再生を実行します

The Return Key:

「Real-Time」「Post-Process」モードで<Return/Enter>キーを押すと、シングルFFT(1回の)を実行して停止します。アベレージングは行いません。

The Focus:

スクロールバー上でマウスをクリックするとフォーカスされ、キーボード(矢印キー)操作が可能となります。例えば、ツールバーの「Avg:」コントロールをクリックするとアベレージパラメータをキー選択することができます。

3章 File Management(ファイル操作)

Opening and Saving Wave Files(WAVファイル操作)

<File><New/Open/Save/Save As>メニューで最大16チャンネルのモノラルファイル操作を行います。ステレオファイルの操作も可能です。現在は.WAVファイルのみに対応していますが、<File><Import>メニューを使用することで他の測定器で取得したテキストデータファイルをインポートすることも可能です。

Enable Recording - データを記録するか否かに関係なく、このオプションはRecorderモードだけで利用可能です。.WAVファイルが開かれていないと、アナライザーはタイトルのない「Untitled」ファイルに記録します。データを記録した後にファイルを保存するかまたは、保存しないでファイルを閉じることができます。

既存ファイルに記録する場合ハードウェアのサンプリングレートは、その.WAVファイルのサンプリングレートと整合しなければなりません。これは.WAVファイル形式がファイル全体で一定のサンプリングレートを必要とするからです。

Post-Processモードで異なったサンプリングレートを持つファイルを処理できます。

モードがReal-Timeに切り替わると現在のファイルは閉じられますので、もし必要なら保存します。必要なメモリ容量はサンプリングレートとフォーマット次第です。

Save All Files Using a Prefix

大量のファイルを扱う作業は面倒です。この機能では、開いているすべてのファイルを素早く保存するためのパターンとして使用する「プレフィックス」を指定することができます。Prefixにはチャンネル番号が付加されます。例えば、「Project 451」というプレフィックスを指定すると、以下のような名前のファイルが保存されます。

```
Project 451_Ch_0.wav  
Project 451_Ch_1.wav  
Project 451_Ch_2.wav  
Project 451_Ch_3.wav  
etc...
```

チャンネル名*は、プレフィックスを使用している場合は、ファイル名に含めることもできます。以下にいくつかの例を示します。:

```
Project 451_Accel_X_Ch_0.wav  
Project 451_Accel_Y_Ch_1.wav  
Project 451_Accel_Z_Ch_2.wav  
Project 451_Mic_H2_Ch_3.wav  
etc...
```

*チャンネル名は、処理設定のキャリブレーションタブに表示されているものです。

Wave File Groups

ダイアログボックスの下部にある対応するボタンを使って、ウェーブファイルのグループを保存したりロードしたりすることができます。これにより、各チャンネルの開いているwaveファイル名のリストを含む小さなテキストファイル(.wfg 拡張子)が作成されます。このファイルは、waveファイルと同じフォルダに保存されます。これは、大きなwaveファイルのセットを録音したり、後処理したりするのに便利です。

File Storage Requirements

1分間のデータに必要なメモリ量は、サンプリングレートやサンプリング形式によって大きく異なります。モノラル録音の代表的な値を以下に示します。

Sampling Rate(Hz)	Sampling Precision(bits)	Bytes /Minute
11,025	8	661,500
11,025	16	1,323,000
22,050	8	1,323,000
22,050	16	2,646,000
44,100	8	2,646,000
44,100	16	5,292,000
44,100	24	7,938,000
96,000	16	11,520,000
96,000	24	17,280,000

Continuous Recording Capability(記録サイズ):

.WAV ファイルは最大 2GB のデータを保存できます(対応するサンプリング仕様に依存します)。限界に達すると、プログラムは記録するのを止めてファイルを保存し、そして新しいファイルに再録し始めます。もしファイルに名前がなければ、コンピュータの日付データを使用して名前を付与します。

例: C:\¥spectraplus¥wave¥recording_2002_11_26_102011.wav (日時:2002/11/26, 10:20:11)

ファイルは<File><Set Paths>メニューで既設定されている「¥wave」フォルダに保存されます。この機能はハードディスクレコーディングモード時にのみ適用されます。

Importing Files(ファイルのインポート)

標準の.WAV ファイルは 1 ブロックのヘッダーを含むバイナリーファイルです。 .WAV ファイルヘッダーはサンプリングレート、チャンネルの数、サンプルあたりのビット数の情報を含んでいます。ステレオファイルは L と R が交互になります(L、R、L、R.)。

インポート機能は、外部のソースからデジタル化しているデータをインポートして.WAV ファイルを生成します。適切に.WAV ファイルヘッダーを造るため、データによって表されたサンプリングレートを指定しなければなりません。そしてサンプリングフォーマットを指定します。

File Formats(ファイル仕様):

- ASCII Integer - どんなテキストエディタでも読み込み可能です。モノタイプファイルでは各列は一つのサンプリング値を含んでいます。ステレオファイルでは各列はコンマによって分離された左右のサンプリング値を含んでいます
- ASCII Float - どんなテキストエディタでも読み込み可能ですが、値は浮動小数点形式(例えば、21.034)に変換されます。モノタイプファイルでは、各列は一つのサンプリング値を含んでいます。ステレオファイルでは、各列はコンマによって分離された左右のサンプリング値を含んでいます

Binary8 bit - 8 bit 整数値

Binary 16 bit - 16 bit 整数値

Binary32 bit - 32 bit 整数値

Binary floating point - 32 bit フローティング値

Sample Values(サンプル値):

8bitWAVdata: 0...255

16bitWAV data: -32768...+32768

24bitWav data: -8388607...++8388607

File Header Options(ファイルヘッダオプション):

データファイルがヘッダーを含んでいれば、列(ASCII ファイル)かバイト(バイナリーファイル)でヘッダーのサイズを指定できます。このヘッダデータはインポート処理ではスキップされます。

Data Scaling(データスケール):

- ・ 「Rescale(再スケール)」がチェックされると、ファイルの最大/最小値を検知して選択したサンプリングプレジジョンで最適になるようスケールを再設定します
- ・ 「Truncate(先端切)」がチェックされるとすべての値を受け入れ、指定されたサンプリングプレジジョンを超えていれば先端を切ります
- ・ データファイルに含まれた単位が分かっているならばこのオプションを使用して下さい。これはデータファイルを適切にスケールリングするための校正パラメータを生成します

Exporting Files(ファイルのエクスポート)

この機能では.WAVファイルをさまざまなファイル形式にエクスポートすることができます。これらのファイルはスプレッドシート、数学処理プログラムやカスタムアプリケーションで利用できます。

File Formats(ファイル仕様):

- ・ ASCII Integer - どんなテキストエディタでも読み込み可能です。モノタイプファイルでは各列は一つのサンプリング値を含んでいます。ステレオファイルでは各列はコンマによって分離された左右のサンプリング値を含んでいます
- ・ ASCII Float - どんなテキストエディタでも読み込み可能ですが、値は浮動小数点形式(例えば、21.034)に変換されます。モノタイプファイルでは、各列は一つのサンプリング値を含んでいます。ステレオファイルでは、各列はコンマによって分離された左右のサンプリング値を含んでいます

Binary8 bit - 8 bit 整数値

Binary 16 bit - 16 bit 整数値

Binary32 bit - 32 bit 整数値

Binary floating point - 32 bit フローティング値

Sample Values (サンプル値):

8bitWAVdata: 0...255

16bitWAV data: -32768...+32768

24bitWav data: -8388607...++8388607

Notes:

- ・ ステレオファイルをエクスポートするときは左のチャンネルから始まり、L, R交互になります
- ・ バイナリーファイルはテキストファイルより効率的です。ASCIIテキスト仕様は大サイズファイルには適しません
- ・ 大きい.WAVファイルから小さい時間セグメントをエクスポートする場合は、Time Series plotを使用してセグメントを指定し、そして<Edit><Copy>メニューを使用してそのセグメントをクリップボードにコピーします。次にファイルを閉じて、<Edit><Paste Insert>メニューでそのセグメントをペーストすることにより新しいファイルを作成して下さい。そしてこの小さい.WAVファイルをエクスポートします
- ・ ASCIIかBinary 8ビットファイルのどちらかに16ビットのWAVファイルをエクスポートすると、8ビットサンプリングレンジにフィットするために再スケールします

File Options(ファイルオプション)

ファイルオプションでは.WAV ファイルの保存とアクセス方法をコントロールします。

Access modes(アクセスモード):

- ・ RAM - WAVEデータはRAMサイズに適合しなければなりません
- ・ Hard Disk - WAVEデータは直接ハードディスクに記録され、再生されます。これは長時間録音を可能にします。メガバイト(MB)単位で最大記録サイズを指定できます(最大2GBのデータを保存できます。時

間はサンプリングフォーマットに依存します)。

Make Backup (バックアップ):

このオプションはハードディスクアクセスモードだけに適用されます。このオプションがチェックされると、一時バックアップファイルは元のファイルを再現できるように作成されます。<File><Set Paths>メニューで一時ファイルに使用するディレクトリを指定してください。アプリケーションを閉じるとすべてのバックアップファイルは消去されます。

Include Calibration Data in WAV file (.WAV ファイル中の校正データ):

このオプションを有効にすると校正値が.WAV ファイルに含まれます。そして、ファイルが再び開かれるときにはロードされます。これはデフォルトで有効に設定されています。校正データはファイルサイズに含まれません。.WAV ファイルで機能する多くのアプリケーションはこの追加データで問題を生じることはありませんが、多くはファイル保存時に校正データを維持しません。校正データ付き.WAV ファイルを読むことに問題を持つアプリケーションを使用する場合は、このオプションを無効にする必要があります。

Include Time Stamp Wave file

このオプションを有効にすると、タイムスタンプ情報がファイルに保存されます。これは、録画開始時のPC時刻に基づいています。

Apply Time Stamp to Plots

有効にすると、タイムスタンプ情報は、Time Series、Spectrogram、3-D Surface プロットの時間軸に使用されます。

Include Metadata in Wave file

このオプションを有効にすると、ユーザーが提供したメタデータがファイルに保存されます。 <Edit><Metadata>メニューをクリックして、メタデータを入力または表示します。

Prompt for Metadata when saving file

この機能を有効にすると、初めてウェーブファイルを保存するときに<Edit><Metadata>ダイアログボックスが表示されます。

Automatically reopen WAV files when program launches

この機能を有効にすると、プログラム終了時に開いたままになっていた wave ファイルを再度開きます。

Real-Time Mode Data Buffering (リアルタイムモードでのデータバッファ):

このオプションでは、Real-Timeモードでデータがバッファリングされるかどうかをコントロールします。データバッファリングは、Spectrogramや3-D Surface PlotスケールパラメータやPlotサイズの変化にともない再描画することやマウス右クリックメニュー機能を可能にします。さらに、Post-ProcessやRecorderモードでの処理のために.WAVファイルに変換できます。このオプションは既定で有効に設定されています。この機能はほとんどの状況で有用であることを理解して下さい。このオプションをオフすると若干の性能向上はもたらされますが、その効果は僅かです。データバッファリングは前述で選択されたものと同じファイルアクセス(RAMかHard Disk)を使用します。既定でのバッファ長は60秒ですが、必要に応じて変更することができます。

Continuous Recording Capability (記録容量):

.WAVファイルは最大2GBのデータを保存できます(対応するサンプリング仕様に依存します)。限界に達すると、プログラムは記録するのを止めてファイルを保存し、そして新しいファイルに再録し始めます。もしファイルに名前がなければ、コンピュータの日付データを使用して名前を付与します。

例: C:\\$spectraplus\$wave\$recording_2002_11_26_102011.wav (日時:2002/11/26, 10:20:11)

Notes:

- ・ 小サイズのファイルを扱う場合、RAMアクセスはより速い性能を提供します。ハードディスクアクセスはより大きいファイルに適応します

- ・ ハードディスクレコーディングの間、他のディスク操作(大きいアプリケーションの起動など)を実行しないで下さい。Windowsはマルチタスクオペレーティングシステムですが、それは個々のアプリケーションの協力を依存しており、システムリソースへの即座のアクセスを保証できません

Set Paths(パス設定)

このオプションはプログラムによって利用される各種ファイルの参照ディレクトリ/フォルダを設定します。パスに関する知識を持たない場合は既定の設定をお勧めします。既定ディレクトリは「Drv:¥SpectraPLUS-DT¥foldername」です。

Folder Name:

- ・ wave: .WAV ファイル
- ・ Temp: 一時ファイル
- ・ config: 定義ファイル
- ・ cal: 校正ファイル
- ・ miccomp: 入力系補正ファイル
- ・ overlay: データストレージファイル
- ・ log: データログファイル
- ・ filter: カスタムフィルタファイル
- ・ reverb: 残響データファイル
- ・ thd: THD+N vs Freq ファイル

Configuration Files(定義ファイル)

<Config>メニューでは現在設定されているアナライザーの定義構成値を保存し、また保存された構成値をロードして再現することができます。

定義ファイルはPlotの位置、オプション、サイズ情報などを含んでいます。ファイル拡張子は「.CFG」です。このファイルはWIN.INIやSYSTEM.INIファイルと同様、標準的なASCIIテキスト形式ファイルです。

メインツールバーの「Load Configuration」とラベルされたドロップダウンリストから、希望のファイルを選択することによって定義ファイルをすばやくロードできます。

Automatically Opening a Configuration file(定義ファイルの自動ロード):

アプリケーションのショートカットを作成して定義ファイルの名前を設定すると、ファイルを自動的に開くことができます。これをショートカットアイコンにセットするにはショートカットアイコンを右クリックし、プロパティメニューのショートカットタブでリンク先を設定します。

例えば、「C:¥SpectraPLUS_DT¥Bin¥SplusDT.exe Test1.cfg」のようにします。これは「test1.cfg」をロードします。

複数のショートカットをテストごとに作成できます。またアプリケーションは、マルチチャンネル DT-9800 ハードウェアをサポートするため同時にランさせることができます。

4章 Editing (編集)

Cut/Copy/Paste commands (カット/コピー/ペースト)

Cut/Copy/Pasteのコマンドは、アクティブなビューの内容をクリップボードにコピーする為に使用されます。データ形式はPlotによって異なります。

Plot	Cut	Copy	Paste
Spectrum	N/A	ASCII table	N/A
Phase	N/A	ASCII table	N/A
Time Series	.WAV format	.WAV format	.WAV format
Spectrogram	.WAV format	.WAV format	.WAV format
3-D Surface	.WAV format	.WAV format	.WAV format

Time Series、Spectrogram、3-D Surface Plotを使用する場合は最初に、編集したいタイムセグメント(時間帯)を選択しなければなりません。これをするには、Plotツールバーの矢印ボタンをクリックします。次に、Plot上の任意の位置をマウス左ボタンでクリックし、そしてドラッグして編集したいタイムセグメントを反転表示します。ここで<Edit>コマンドを選択して必要な操作を行います。Real-Timeモードではこの機能は無効です。

.WAVファイルはこの形式をサポートする他のアプリケーションとも互換性があります。

ASCIIテーブル形式は2つのコラムを生成します。左のコラムには周波数が、右のコラムにはアンプリチュードかフェーズが配置されます。コラムはタブによってセパレートされます。そして、このデータはスプレッドシートやテキストエディタに貼り付けて利用することができます。また、このファイルはスペクトルオーバーレイ(Overlay)と補償ファイル(Mic compensation)の両方と互換性があります。

<Undo><Edit>コマンドは直前の編集操作をキャンセルして回復することができます。これは<Edit><Cut>、<Edit> <Paste Over>、<Edit><Paste Insert>、<Edit Filter>に適用されます。

Play/Play Special commands (再生操作)

<Play>と<Play Special>メニューコマンドは、選択したタイムセグメントをスピーカーから再生します。

まず、編集したいタイムセグメントを選択しなければなりません。この操作はTime Series、Spectrogramか3-D Surface Plotのいずれかで可能です。最初にビューツールバーの矢印アイコンをクリックします。次にPlot上の任意の位置をマウス左ボタンでクリックし、そしてドラッグして編集したいタイムセグメントを選択します。ここで<Edit>コマンドが有効になり、必要な操作を選択できるようになります。

<Edit><Play>メニューは選択したサウンドデバイスを介して、選択されたセグメントを再生します。

<Edit><Play Special>メニューには4つの選択肢があります。

- Loop - 選択されたタイムセグメントを継続再生/リピートします(Stopボタンをクリックするとキャンセルされます)
- Half speed - 選択されたタイムセグメントを1/2倍のサンプリングレートで再生します
- Double speed - 選択されたタイムセグメントを2倍のサンプリングレートで再生します
- Arbitrary speed - 選択されたタイムセグメントを指定したサンプリングレートで再生します

Filtering(フィルタ)

.WAVファイルをフィルタ処理するにはまず、Time Series、Spectrogramまたは3-D Surface Plotでタイムセグメントを選択しなければなりません。ビューツールバーにある選択矢印アイコンをクリックします。そして、Plot上でマウス左ボタンをクリックしドラッグします。選択範囲がリバーズ表示されます。タイムセグメントが選択されれば、ツールバーのフィルタアイコンまたは、<Edit><Filter>メニューコマンドが有効になります。そして、フィルタリングパラメータダイアログボックスでフィルタ処理条件を選択することができます。

5種類のフィルタ:

- Low Pass - 指定されたカットオフ周波数より低域のすべての成分を通し、高域のコンポーネントを減衰させます
- High Pass - 指定されたカットオフ周波数より高域のすべての成分を通し、低域のコンポーネントを減衰させます
- Band Pass - 指定された帯域の周波数を通し、他の帯域を減衰させます
- Notch - 指定された帯域をリジェクトし、他の帯域を通します
- Custom - 指定されたフィルタレスポンスファイルをデータに適用します

Filter Sharpness - フィルタタップの数(係数)は、フィルタ応答の鋭さと、どれくらい理想的な応答に合っているかに直接影響します。より多くのフィルタタップがよりシャープなフィルタレスポンスをもたらしますが、より多くの処理時間を必要とします。

Filter Gain - オーバーオールゲインを調整可能です。(dB)単位で値を入力します。正数値はゲインを増加し、負数値は減衰します。

フィルタレスポンスファイルの作成手順:

1. 標準のASCIIテキスト・ファイルを作成できるNOTEPAD.EXEアプリケーション、スプレッドシートまたはワードプロセッサを起動します
2. フィルタレスポンス曲線を表す2つの数値をコラムに入力します
3. 左のコラムは(Hz)単位の周波数です
4. 右のコラムは(dB)単位のオフセットです
5. 昇り順で情報を入力してください
6. Tabキャラクタで各コラムを分離します

例:

```
-----  
100.0      -10.0  
500.0      -5.0  
1000.0     0.0  
2000.0     2.0  
10000.0    -5.0  
15000.0    -15.0
```

最大32,768、最小3ポイントの情報が必要です。アナライザーはデータポイント間のスペクトルを補間するため「キューブスプラインアルゴリズム」を使用します。セミコロンで始まる行は注釈行として扱われて無視されます。フィルタ応答ファイルの拡張子は「.FLT」です。

Note:

フィルタレスポンスファイル、Microphone CompensationファイルおよびOverlayファイルは、同一形式のフォーマットで互換性があります。これは正しくフィルタレスポンス情報を入れたことを確かめる場合役に立ちます。これを行うには、フィルタレスポンスファイルをSpectrum Plotのオーバーレイ機能でロードして下さい。尚、<File><Set Paths>メニューからSpectraPLUSが使用するフォルダ設定をカスタマイズできます。

Mute(ミュート)

タイムセグメントをミュートすると、レベル値はゼロに設定されます。

.WAV ファイルのセグメントをミュートするには、最初にタイムセグメントを選択しなければなりません。その操作は Time Series、Spectrogram または 3D Surface Plot で行います。Plot ツールバーにある矢印アイコンをクリックし、そして Plot 上の任意のポイントをマウスクリックしてドラッグします。選択範囲が反転表示されます。タイムセグメントを選択したら、<Edit><Mute>メニューコマンドをクリックします。選択されたタイムセグメントはゼロに設定されます。

Notes:

<Edit><Undo Edit>メニューコマンドはデータを直前の値に回復します。

Select All(全ての選択)

<Select All>メニューコマンドは、Time Series、Spectrogram、3D Surface各Plotの全タイムセグメントを選択します(反転表示されます)。

Notes:

編集のために.WAVファイル全体を選択したければ、最初にTime Series Plotのツールバーで「Zoom Out Full」ボタンをクリックして下さい。また、Time Seriesオプションダイアログの「Maximum Zoom Limit」値を増やす必要があるかも知れません。

Gain Adjust(ゲイン調整)

.WAV ファイルのゲインを調整するには、最初にタイムセグメントを任意に選択しなければなりません。その操作はTime Series、Spectrogramまたは3-D Surface Plotで行います。Plotツールバーにある矢印アイコンをクリックし、そしてPlot上の任意のポイントをマウスクリックしてドラッグします。選択範囲が反転表示されます。タイムセグメントを選択したら、<Edit><Gain Adjust>メニューコマンドをクリックして下さい。次に、加減するための利得の量をデシベル(dB)単位で入力してください。OKボタンをクリックして調整を実行します。

DC Offset(DC オフセット)

.WAVファイルのDC Offsetを調整するには、最初にタイムセグメントを任意に選択しなければなりません。その操作はTime Series、Spectrogramまたは3-D Surface Plotで行います。Plotツールバーにある矢印アイコンをクリックし、そしてPlot上の任意のポイントをマウスクリックしてドラッグします。選択範囲が反転表示されます。タイムセグメントを選択したら、<Edit><DC Offset>メニューコマンドをクリックして下さい。次に、加減するためのDCオフセットの量を入力します。OKボタンをクリックしてDCオフセットを実行して下さい。

Notes:

<Edit><Undo Edit>メニューコマンドはセグメントを直前の値に戻します。

Dynamic DC offset

この機能は、波形の平均オフセットを計算し、time series からそれを減算します。これにより、固定オフセットを適用するのではなく、DC オフセットを動的に削除します。

ダイナミック DC オフセットアルゴリズムをファイルに適用するには、まず時間セグメントを選択する必要があります。

これは、Time Series、Spectrogram、または 3D Surface プロットのいずれかから行います。プロットツールバーの選択矢印をクリックして、目的の時間セグメントをクリックしてドラッグします。すると、このセグメントが inverse video で表示されます。時間セグメントが選択されたら、<Edit><Dynamic DC Offset>メニューコマンドを選択します。次に、Dynamic DC Offset パラメータの入力を求められます。OK ボタンをクリックして計算を実行します。

Parameters:

- Cutoff Frequency - これはフィルタのカットオフ周波数を決定します。周波数が低いほど計算時間が長くなります。
- Recalculation Interval - これは、オフセットを再計算する頻度を制御します。間隔を小さくすると、計算時間が長くなります。

Notes:

<Edit><Undo Edit>メニューコマンドはセグメントを直前の値に戻します。

5章 Mode (動作モード)

Real-Time、Post-Process、Recorder の3つの動作モードを提供します。

Real-Time Mode (リアルタイムモード)

Real-Timeモードではプログラムは入力デバイスから直接、デジタル化されたアナログデータのFFTサイズブロックを取得し、スペクトラムを処理して表示します。プログラムはストップされるまで中断なく新しいデータを取得し、前のデータとそれを平均化して結果を表示します。ツールバーにはアベレージングとピークホールド条件を設定するコントロールが有ります。これは稼働中にも使用可能です。

デフォルトでは、最新の60秒間のデジタル化データは一時バッファに保存されます。RecorderかPost-Processモードに切り替えると、バッファデータを.WAVファイルにコンバートできます。

Notes:

コンピュータのCPU性能、選択されたサンプリングレート及びFFTサイズによっては、コンピュータは次のブロックが有効になる前にFFTを実行して結果を表示することができないかも知れません。この場合、処理データにギャップがあるでしょう。ギャップのない分析を実行するには記録後に、Post-Processモードによってデータを解析します。

最終的な測定をする前に、まずReal-Timeモードで予備の測定をし、次にRecorderモードに切り替えて実行することをお勧めします。

Post-Process Mode (後処理モード)

Post-Process モードは、記録・保存された.WAV ファイルを分析できます。スペクトルアップデートが優先しますので、スピーカを通して再生することはできません。再生は Recorder モードか、または<Play><Play Special>メニューを使用して下さい。

このモードはRecorderやReal-Timeモードより柔軟なコントロールを提供します。処理データにはギャップがなく、そしてオーバーラップ処理によりSpectrogramと3-D Surface Plotの時間軸をストレッチできます。さらに選択されたタイムセグメントを編集、再生可能です。

.WAVファイルを開くと、サンプリングレートとフォーマットはファイルに記録されたレートに変更されます。サンプリングレートは.WAVファイルに残されていなければなりません。現在のサンプリングレートは画面下部のステータスバーにいつも表示されます。

Single step operation:

<Enter>キーを押すとシングルFFTを実行し、そしてアナライザは自動的に停止します。アベレージングはシングルステップ操作の間にリセットされません。

Recorder Mode (レコーダモード)

このモードではマルチチャンネルの入力信号を個々のモノラルファイルに記録することができます。記録または再生している間も信号のスペクトラムを表示します。

プログラムはRAMファイルアクセスモードを使用するとき、録音のために十分なメモリを割り当てることを試みます。もし十分でなければ録音長を減少させます。録音の長さは「File Options」ダイアログボックスで指定されます。Hard Diskファイルアクセスモードでは、プログラムは直接ハードディスクにアクセスして録音、再生を行います。録音中は一時ファイルが作成されます。一時ファイルが使用するディレクトリは<File><Set Paths>メニュー

で指定してください。アプリケーションを終了するとすべての一時ファイルが消去されます。

必要なメモリ容量はサンプリングレートとフォーマット次第です。モノラル録音時の代表的な値を以下に示します。ステレオ時は倍になります。

Sampling Rate(Hz)	Sampling Precision(bits)	Bytes /Minute
11,025	8	661,500
11,025	16	1,323,000
22,050	8	1,323,000
22,050	16	2,646,000
44,100	8	2,646,000
44,100	16	5,292,000
44,100	24	7,938,000
96,000	16	11,520,000
96,000	24	17,280,000

.WAVファイルを開くと、サンプリングレートとフォーマットはファイルに記録されたレートに変更されます。サンプリングレートは.WAVファイルに残されていなければなりません。現在のサンプリングレートは下部のステータスバーにいつも表示されます。

Continuous Recording Capability(記録容量):

.WAVファイルは最大2GBのデータを保存できます(対応するサンプリング仕様に依存します)。限界に達すると、プログラムは記録するのを止めてファイルを保存し、そして新しいファイルに再び記録し始めます。もしファイルに名前がなければ、コンピュータの日付データを使用して名前を付与します。ファイルは<File><Set Paths>で規定している「¥wave」フォルダに保存されます。この機能はハードディスクレコーディングモード時にのみ適用されます。

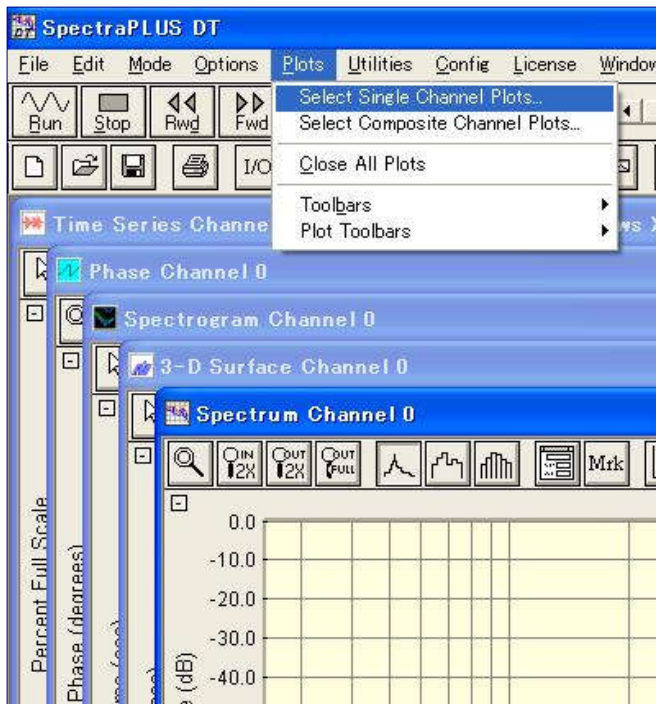
例: C:¥spectraplus¥wave¥recording_2002_11_26_102011.wav (日時:2002/11/26, 10:20:11)

Important:

ハードディスクレコーディングの間、他のディスク操作(大きいアプリケーションの起動など)を実行しないで下さい。Windowsはマルチタスクオペレーティングシステムですが、それは個々のアプリケーションの協力に依存しており、システムリソースへの即座のアクセスを保証できません。

6章 Plots(グラフ表示モード)

SpectraPLUS-DT には Time Series、Spectrum、Phase、Spectrogram、3-D Surface の5つと、THD+N vs Freq、Reverb(RT-60)の2つのグラフ表示ウィンドウがありますが、前出5つのウィンドウを Plot と呼びます。



Time Series Plot(タイムシリーズ表示)

デジタル化したアナログデータを表示します。オシロスコープと同様にアンプリチュードは垂直軸に、時間は水平軸に表示されます。

Real-Timeを使用している時、グラフは最新の60秒のデータを表示します。RecorderかPost-Processモードでは、ツールバーとスクロールバーコントロールを使用してデータファイル全体をズーム、スクロールできます。水平軸スケリングは秒単位です。マウス左ボタンでPlot上の任意のポイントをダブルクリックするとファイルの位置をポイントした位置に移動します。さらに、新しいポイントのスペクトラムを計算してSpectrum Plotに表示します。

Spectrum Plot(スペクトラム表示)

スペクトラムの2次元ビューです。水平軸は(Hz)単位で周波数を示します。垂直軸はそれぞれの周波数のアンプリチュード(レベル)を示しています。キャリブレーション(校正)オプションを使用することでアンプリチュードスケール(表示単位)を変更できます。

有効なスペクトラルの総数は「FFT Size」値の1/2です。従って、FFT Size = 1024ポイントでは512本のスペクトラルラインがあります。測定周波数範囲は0(Hz)から「Sampling Rate」の1/2(Hz)です。

最大表示スペクトラムは上下限で制限されます。最も下の3個は、DCコンポーネントを含んでいて表示されません。上限の1%は、しばしばエイリアスコンポーネントを含んでいるので表示されません。

ツールバーのズームボタンで表示スパンを容易に調整できます。しかしFFTアルゴリズムは、それらを表示するかどうかにかかわらずすべての周波数を計算する必要があります。

Plotから直接周波数とアンプリチュード情報を観察するのにマウスを使用できます。また、<Edit><Copy>メニューはスペクトル数値データを表様式でクリップボードにコピーします。これは直接スプレッドシートやテキストファイルに貼り付けて利用することができます。

Phase Plot(位相表示)

信号対周波数のフェーズ(位相)を表示します。チャンネル間のフェーズを調べるとき最も役に立ちます。この場合は<Plot><Select Composite Channel Plots>メニューを選択して下さい。

周波数スケールは設定されている「Scaling」オプションで決まります。ツールバーのズームコントロールは表示する周波数スパンを調整するのに使用できます。

Plotから直接周波数と位相情報を測定するのにマウスを使用できます。また、<Edit><Copy>メニューコマンドは現在のフェーズ数値データを表様式でクリップボードにコピーします。これは直接スプレッドシートやテキストファイルに貼り付けて利用することができます。

フェーズはスムージングウィンドウ(窓関数)の選択で大きく影響を受けます。「Uniform」を選択すると最もクリーンなフェーズレスポンスが生成されます。

Spectrogram Plot(スペクトログラム表示)

時系列のスペクトルデータをサーモグラフィのようにカラー表示します(ソノグラフ)。アンプリチュードは色調/階調で示されます。この表示方法に馴染みのないユーザでも少しの習得で、スペクトル解析に最も役に立つ表示パターンの一つであることを理解できるでしょう。3-D Surface Plot を俯瞰するようにアンプリチュードが表示されます。Plot の右側にアンプリチュードとカラー(色度/階調)の相関を示します。Spectrogram Plot 上でマウス左ボタンをダブルクリックすると、対応するスペクトルと Time Series データが表示されます。

3-D Surface Plot(3次元表示)

スペクトルデータの3次元時系列透視図です。周波数は水平軸、時間は垂直軸に示されます。このPlotはスペクトルデータの視覚的ダイナミックイメージの観察に役立ちます。

グラフをマウス左ボタンでダブルクリックすると、対応するスペクトルとTime Seriesデータが表示されます。

Single Channel と Composite Channel(シングルチャンネルとコンポジットチャンネル)

SpectraPLUS-DT には被測定信号取込解析用一次チャンネルと処理解析データ表示用二次チャンネルがあり、前者を Single Channel、後者を Composite Channel と呼びます。Plot ID は前者が「0 - 15」、後者が「A - P」で表されます。

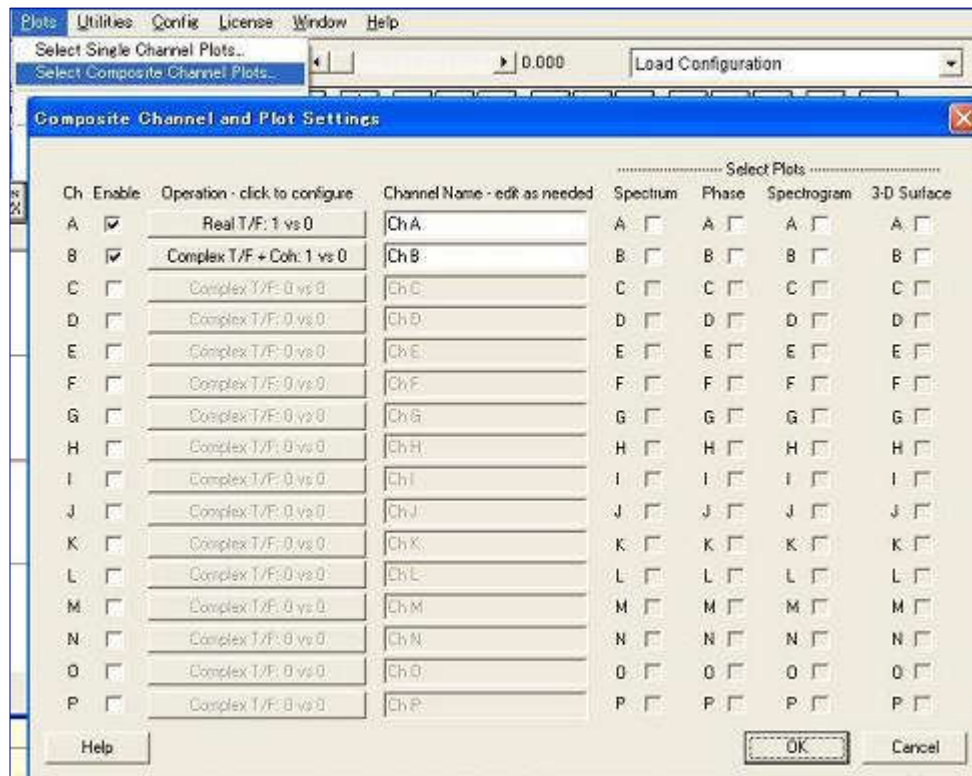
Selecting Single Channel Plots (シングルチャンネルモードの設定):



<Plots><Select Single Channel Plots>メニューでダイアログボックスを起動し、使用するチャンネルの Plot を選択します。Plot を選択する前に、そのチャンネルが有効になっているか、.WAV ファイルが開かれていなければなりません。チャンネルの有効化は <Options><Processing Settings>メニューから行います。最大 16 チャンネルまで選択できます。チャンネル番号は 0 - 15 です。この番号は使用する DT-9800 ハードウェアを反映します。

Composite Channel Processing(コンポジットチャンネル処理):

コンポジットチャンネル処理機能は種々の強力な拡張測定を提供します。この機能をセットするには<Plots><Select Composite Channel Plots>メニューをクリックします。最大 16 チャンネルまで対応します。チャンネルには A - P のアルファベットが割り当てられ識別されます。



Definitions (定義):

- Auto-spectrum - 複雑な結合 (正反対のフェーズ) にスペクトラムを掛けることによって計算します。オートスペクトラムはリアルで、マグニチュードレスポンスに一致します。オートスペクトラムはシングルチャンネルの Spectrum Plot に表示されます
- Cross-spectrum - 第二スペクトラムの複雑な結合にスペクトラムを掛けることによって計算します。クロススペクトラムは複雑です (リアルとイメージコンポーネントがあります)。クロススペクトラムのマグニチュードは両チャンネルに共通のパワーを表し、フェーズはチャンネル間の位相差を表します。例えば、2つの信号に 180 度の位相があればクロススペクトラムのマグニチュードはゼロです。もし2つのチャンネルが一致すれば、そのクロススペクトラムはオートスペクトラムと等価です
- Transfer Function - 2チャンネル間の比です。2つの方法で計算できます。トランスファー機能は非常にパワフルです。迅速かつ正確に周波数特性を算出します
 - ✓ Real - 大変シンプルな方法です。各チャンネル間のオートスペクトラム (マグニチュード) の比です。各チャンネルのフェーズは結果に関与しません
 - ✓ Complex - 2チャンネルのクロススペクトラムとリファレンスチャンネルのオートスペクトラムの比です。各チャンネルのフェーズは結果に関与します
- Coherence Function - クロススペクトラムのマグニチュード比で、両チャンネルのオートスペクトラムの産物です。チャンネル間のリアリティ度を測ります。統計学で使われる方形 (スクエア) 相関関数に類似しています。2つの完全なコヒーレント信号はコヒーレンス値 1.0 になります

Composite Measurement Type (コンポジットチャンネルの処理タイプ):

- Average Multiple Channels - 2つ以上のチャンネルのオートスペクトラムを平均化します
- Real Transfer Function - 2つのチャンネル間のリアルトランスファーを処理して表示します
- Complex Transfer Function - 2つのチャンネル間のコンプレックストランスファーを処理して表示します
- Complex Transfer Function and Coherence - 2つのチャンネル間のコンプレックストランスファーとコヒーレンスを処理して表示します
- Cross Spectrum - 2つのチャンネル間のクロススペクトラムを処理して表示します

Notes:

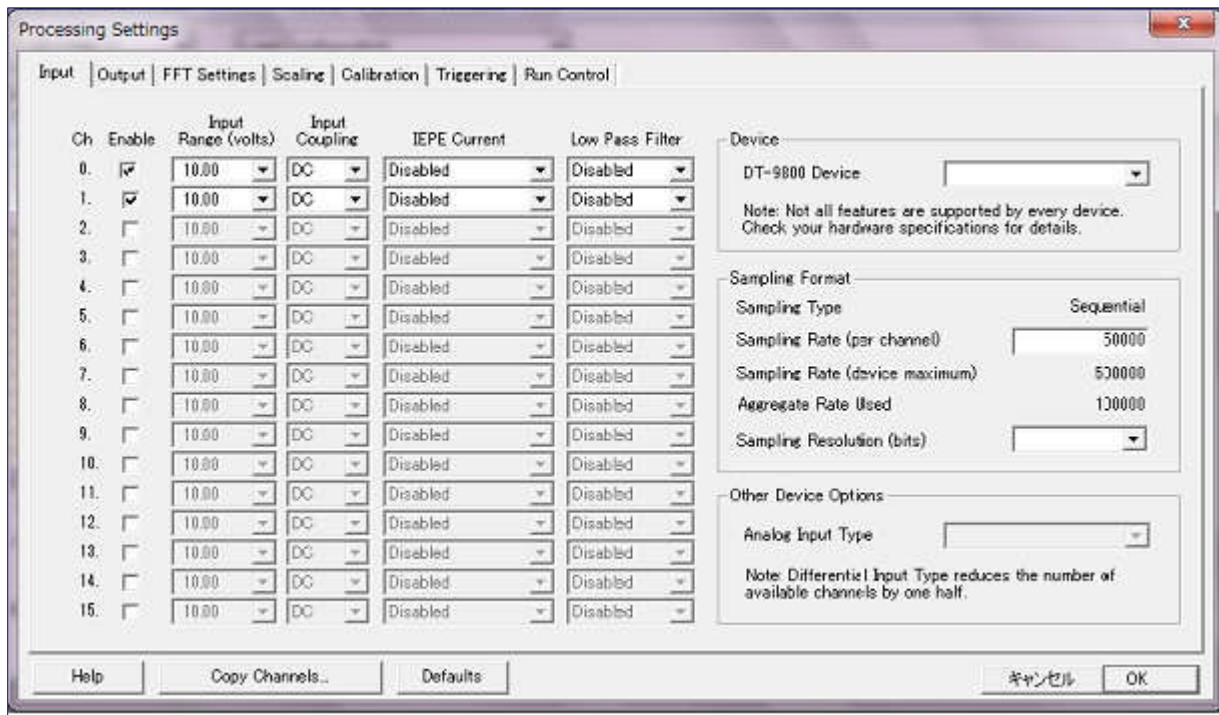
- Time Series plot には適応しません
- 対象チャンネルは同一のサンプリングレート、フォーマット、FFT サイズでなければなりません
- コンポジット Plot で使われるスケーリングは次の通りです
 - Average multiple channels: アベレージリストの最初のチャンネルに選択したスケーリングと同
 - All other types: "test channel"に選択したスケーリングと同

7章 Options(オプション)

Input Device Settings(入力デバイス設定)

入力デバイスは Record モードと Real-Time モード時に使われます。

<Options><Processing Settings>メニューもしくは F4 キーをクリックし、"Input"タブを選択して設定します。ツールバーの「I/O」アイコンをクリックしても行えます。



コンピュータに接続されている DT-9800 ハードウェア・モジュールがデバイスリストに示されます。アナライザーは選択されたデバイスを参照し、セッティング調節可能かを確認します。すべての DT-9800 デバイスがすべてのセッティング機能をサポートするわけではありません。機能のいくつかは使用不可であるかも知れません。詳細はハードウェアの資料を参照して下さい。

Enable: 入力チャンネルのコントロールは Real-Time か Record モードで可能となります。

Input Range: デバイスの入力電圧レンジ(ゲイン)を選択します。この値は Volts (peak)です。このレベルを超える信号は入力をオーバーロード(クリップ)します。オーバーロードしたチャンネルはステータスバーに赤で示します。すべての DT-9800 ハードウェアモジュールが入力ゲイン調整をサポートしている訳ではありません。

Input Coupling: 入力チャンネルの AC/DC カップリングを選択します。代表的な AC カップリングはオーディオ、バイブレーション測定で使われます。全ての DT-9800 デバイスがこの機能をサポートする訳ではありません。

IEPE Current: IEPE(constant current)パワーをオンするか否かをコントロールします。IEPE は多くのトランスデューサ(マイク、加速度計)で使われ、外部電源の必要性をなくします。全ての DT-9800 デバイスがこの機能をサポートする訳ではありません。

Low Pass Filter: デジタル化の前にローパスフィルタを適用するか否かをコントロールします。全ての DT-9800 デバイスがこの機能をサポートする訳ではありません。

Sampling Rate: すべてのチャンネルで使われるサンプリングレートをセットします。すべてのチャンネルが同一でなければなりません。

DT-9800 シリーズのサンプリング形式は次の2つのカテゴリーに分類されます。

- Sequential sampling – ひとつのデジタイザがすべてのチャンネルで使われます。各々のチャンネルには瞬時的時間差で割り振られます。チャンネル毎の最大サンプリングレートは「デバイスの最大サンプリングレート/有効チャンネル数」と等価です。例えば、「デバイスの最大サンプリングレート 500kHz/有効チャンネル数 10 = チャンネル毎の最大サンプリングレート 50kHz」となります。同様に有効チャンネル数が 2 ならば 250kHz です。
- Simultaneous sampling – チャンネル毎にそれぞれのデジタイザが使われます。各チャンネルは同時にデバイスの最大サンプリングレートでサンプリングすることができます。

Sampling Resolution: これはアナログ/デジタル(A/D)変換で使われる個々の数値を指定します。これは測定のダイナミックレンジに直接影響を及ぼします。リストには選択した DT-9800 デバイスによってサポートされるサンプリング値だけが表示されます。

Analog Input Type: アナログ入力タイプは Single Ended (アンバランス/不平衡)か Differential (バランス/平衡)を選択します。すべての DT-9800 デバイスが Differential をサポートするわけではありません。

- Single Ended – 片極がグランドです。
- Differential input channels – 独立したグランド極を持ち、グランドノイズに対して有効です。利用できる入力チャンネルの総数は 1/2 になります。配線の詳細はハードウェアの資料を参照して下さい。

Tachometer Channel: DT-9837、DT9839 と DT-9826 はタコメータ用入力チャンネルを持ちます。このチャンネルが選択されると、Input Device ダイアログでは "Tachometer Input Channel" とラベルされます。

- Ticks /Rev (音/回転)- このパラメータはシャフト一回転のタコメータパルスを設定します。

このチャンネルを有効にすると自動的に "RPM" キャリブレーションにセットされ、Time Series Plot は RPM を表示します。他の Plot (Spectrum, Spectrogram, 3-D surface, Phase) は利用可能ですがタコメータチャンネルは関与しません。

RPM を表示するユーティリティのウィンドウは <Utilities><Tachometer RPM>メニューで表示されます。ウィンドウはサイズ調整が可能です。

RPM 値は Data Logging オプションでも取得可能です。

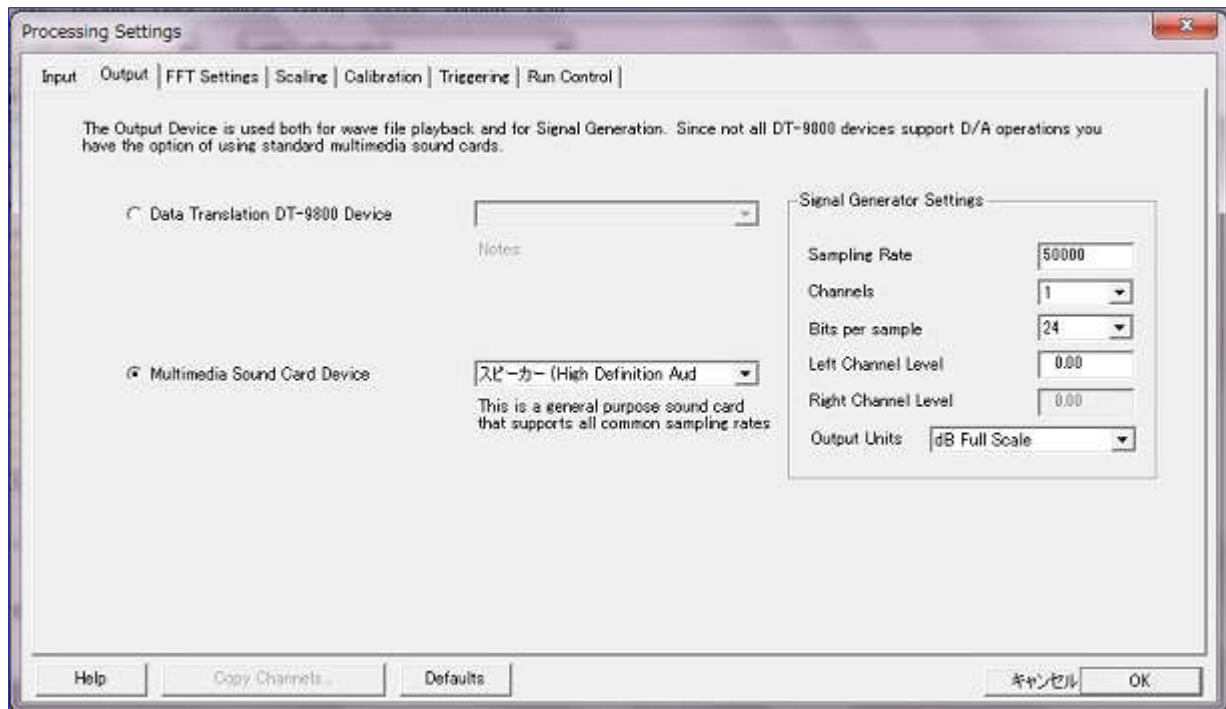
タコメータ信号については、ハードウェアのマニュアルを参照してください。

タコメータデータは記録され、後処理可能です。データ値は 100 回転数と等価の整数値として WAV ファイルにストアされます。最大 RPM は 83,886 です。

Output Device Settings (出力デバイス設定)

出力デバイスはプレイバック時にアナライザーによって、信号出力時にはシグナルジェネレータで使われます。D/A 変換機能はすべての DT-9800 デバイスがサポートしている訳ではありません。その代替として標準的な Windows 対応サウンドカード/デバイスを使うこともできます。

Output Device Settings ダイアログにアクセスするには、<Options><Processing Settings> メニュー(もしくは F4 キー)をクリックします。また、Signal Generator/シグナルジェネレータ機能の[I/O Settings]ボタンをクリックしても可能です。



コンピュータに装着されている DT-9800 デバイスとサウンドカードのすべてをデバイスリストに表示します。

Sampling Rate: シグナルジェネレータのサンプリングレートを設定します。WAV ファイル再生の間は WAV ファイルの設定値が使われます。

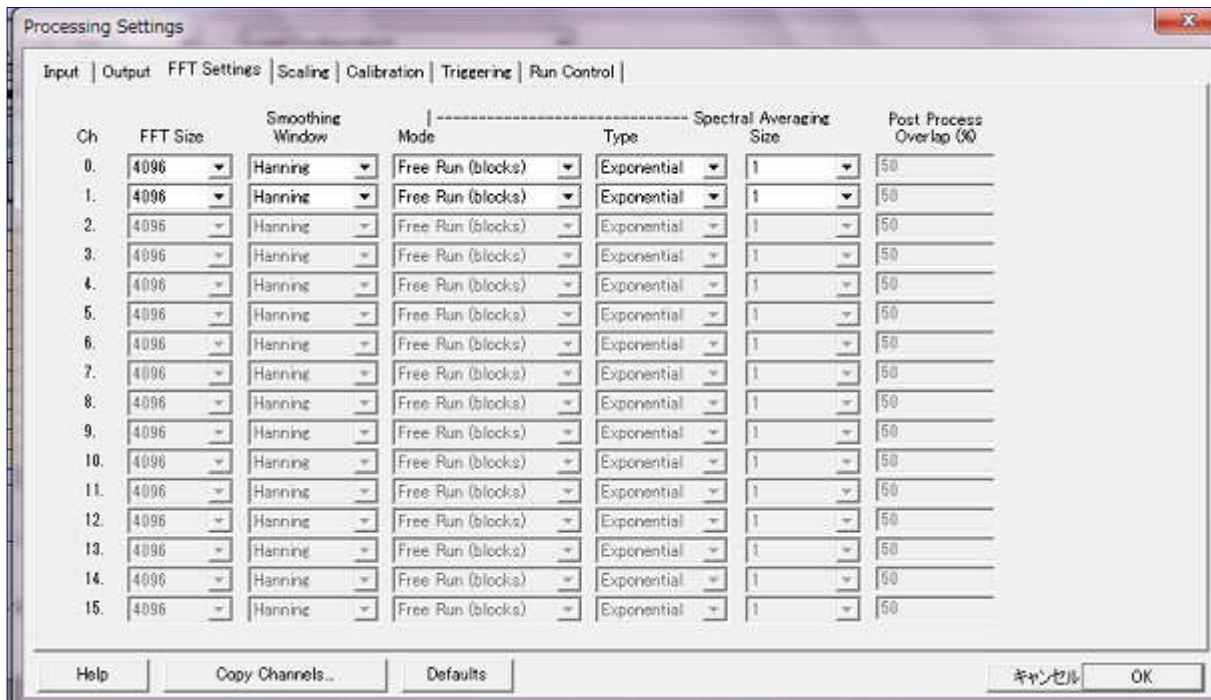
Channels: シグナルジェネレータのチャンネルを設定します。WAV ファイル再生の間はひとつのチャンネルのみ使われます。

Bits Per Sample: シグナルジェネレータのサンプリングプレジジョンを設定します。WAV ファイル再生の間は WAV ファイルの設定値が使われます。

Channel Levels: シグナルジェネレータ各チャンネルの出力レベルを設定します。WAV ファイル再生の間は WAV ファイルの設定値が使われます。この信号レベルはピュアトーンシグナル(非雑音)です。

Output Units: レベルの単位を設定します。Windows 対応サウンドカード/デバイスを選択した場合、単位はカードのフルスケール出力の相対単位(dB)です。DT-9800 デバイスの場合は指定単位で直接セットすることができます。

FFT Settings (FFT設定)



FFT Size (FFT サイズ):

選択した FFT サイズは直接スペクトラムの分解能に反映します。スペクトラルライン数は「FFT Size / 2」です。従って「FFT 1024 points」は「512」スペクトラルラインを生成します。

スペクトラルラインの周波数分解数は「Sampling Rate / FFT Size」と等価です。例えば、FFT サイズが「1024」で、サンプリングレートが「8192」であれば、スペクトラルラインの分解能は「8192 / 1024 = 8Hz」となります。

大きな FFT サイズは高いスペクトラル分解を得られますが、処理時間が長くなります。

現在の設定値は画面下部のステータスバーに表示します。また、アクティブなチャンネルはタイトルバーが強調されます。

Smoothing Windows (窓関数):

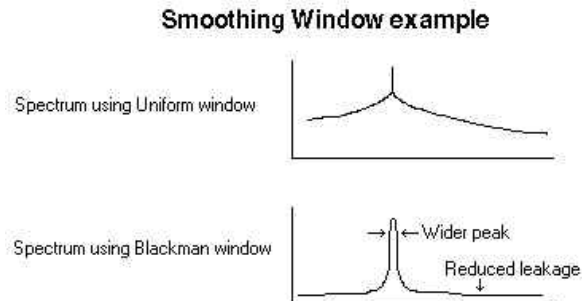
もし、正弦波がタイムシリーズの「始め」と「終わり」の部分でゼロを通過していれば、FFT スペクトラムの結果は正確なアンプリチュードと周波数によって一本の線になります。また、信号レベルがタイムシリーズの一端あるいは両端でゼロにならなければ、ウェーブフォームトランケーション(先端切)が発生し、その結果サンプリングした信号に不連続(切れ目)が発生します。FFT 処理に問題を生じさせスペクトラムの汚れとなります。これはリーケージと呼ばれ、本来は隣接したラインにあるリークのエネルギーです。もし、タイムシリーズのゼロ交差がサンプリングタイムと同期すればリーケージを回避することができますが、実際には不可能です。リーキースペクトラムのシェープ(形状)はシグナルトランケーションの量次第で、リアルシグナルに対しては予知できません。

リーケージの影響を避けるにはタイムシリーズの「始め」と「終わり」の部分でシグナルレベルをゼロにする必要があります。これは種々のシェープを持つ窓関数(Smoothing Window)機能によるマルチデータサンプリングによって行います。各窓関数間の差は、エッジ付近のローウェイトからシーケンスの中央付近のハイウェイトに移行する過程にあります。関数が使われない状態を、"Rectangular", "Flat"もしくは"Uniform"ウィンドウと呼びます。

窓関数が両端をゼロにしている間に、スペクトラのサイドバンドの結果であるタイムシリーズにひずみを加えます。サイドバンドやサイドローブはアナライザーの周波数分解能を効果的に減らします。ウェイトされた信号の

アンプリチュード測定値は、信号の一部が除去されているため正確ではありません。この除去に対して調整をするためにウィンドウアルゴリズムはシーケンスの中央付近の値に特別なウェイトをかけます。現在の設定値は画面下部のステータスバーに表示します。

下図は「Uniform」と「Blackman」ウィンドウ使用時のスペクトラムの比較です。「Uniform」はアン・ウェイティングです。



Characteristics of various smoothing windows

・Fair: 公正 ・Moderate: 適当 ・Good: 良 ・Excellent: 最良 ・Poor: 不的確

Window Type	Frequency Resolution	Amplitude Resolution	Leakage Suppression	Application (適用)
Bartlett	Fair	Fair	Moderate	
Blackman	Fair	Good	Excellent	Distortion 測定
Flattop	Poor	Excellent	Moderate	Amplitude 測定
Hamming	Fair	Fair	Fair	
Hanning	Fair	Excellent	Excellent	Distortion、Noise 測定
Kaiser	Fair	Fair	Poor	
Parzen	Fair	Fair	Poor	
Triangular	Fair	Fair	Poor	
Uniform	Excellent	Poor	Poor	高分解周波数測定

Notes:

- ・ Time Series Plot には他のビューに優先してデータを表示します
- ・ これらの機能はログモードスケール時に顕著に表れます

Averaging Settings (アベレージ設定):

この設定でスペクトラルデータのアベレージ方法をコントロールします。

Averaging Mode (アベレージングモード)

- ・ Free Run (blocks) - プログラムの既定値です。新しいFFTブロックが計算される毎にアナライザー表示をアップデートします。このモードでは、ムービングアベレージを計算するためにどれくらいのスペクトラブロックサイズをアベレージするかを決めます。例えば、アベレージングブロックサイズを4にセットすれば現在のスペクトラム表示値は直前4つのムービングアベレージです。

もし、被測定信号の変化が大きい場合は小さいブロックサイズを使うべきです。環境騒音が大きい場合は、埋没した信号を捕捉するために大きいブロックサイズを使います。

アベレージングサイズを「Infinite」にセットすると、アナライザーを停止するまで連続してアベレージを

実行します。アナライザーを停止あるいは、再スタートするとアベレージをリセットします。

- Sound Level Meter (SLM) - このモードでは標準サウンドレベルメータ(SLM)にマッチするようにアルゴリズムをセットします。アベレージスピードはスペクトラルアベレージングとディケイタイムを次のようにコントロールします。
 - ✓ Off - アベレージングしません
 - ✓ Fast - ディケイレート 32dB/sec でランニングアベレージを算出します
 - ✓ Medium - ディケイレート 20dB/sec でランニングアベレージを算出します
 - ✓ Slow - ディケイレート 4dB/sec でランニングアベレージを算出します
 - ✓ Forever - アナライザーを停止するまで継続計算します

このモードでは「Exponential」アベレージングを使用すべきです。

Averaging Type(アベレージングタイプ)

- Exponential - 最新のスペクトラデータをアベレージします。もし入力信号を切るとディケイレートはエクスポネンシャルカーブになります
- Linear - スペクトラオーバータイムのリニアなアベレージングです。各スペクトラブロックサイズはアベレージに均一に関与します。このアベレージングタイプは"Stable Averaging"として知られています
- Vector - 連続するスペクトラオーバータイムのコンプレックス(Vector)アベレージを実行します。ベクトルアベレージングはフェーズ成分を内包しますので、有意義な結果を求めるならばトリガーリングが使われなければなりません

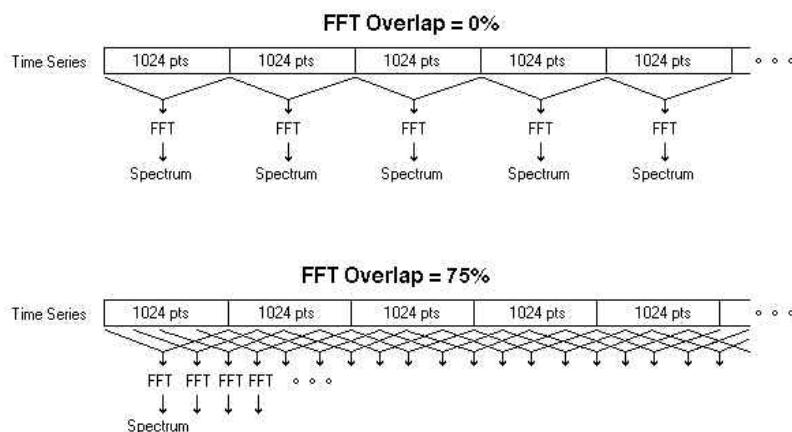
既定セットは最も効率的な「Exponential」です。

「Real-Time」と「Post-Process」の両モードでは動作中、ツールバーの「Avg」コントロールを使ってアベレージングブロックサイズを変更することができます。

Overlap Percentage(オーバーラップパーセンテージ) :

この強力な機能は、「Time Series」を再処理することにより「Spectrogram」と「3-D Surface」Plotの時間軸をストレッチすることを可能にします。設定値はデータポイントのFFTサイズブロックが次のトレースで使われる割合を決定します。高いOverlap Percentage(90+%)と低いFFT block Sizeの設定は時間軸の高分解を可能にしますが、データ処理時間が長くなります。この機能は「Post-Process」モードでのみ有効です。

大きいFFT Size値は周波数の高分解を実現しますが、低いFFT Size値と反対に時間分解が不十分です。オーバーラップ処理は周波数と時間軸双方の高分解をまかなうために使います。例えば、信号に2つの異なる周波数トーンが含まれていると、2つの周波数はしばしばシングルFFTサイズ・Time Seriesブロックに含まれるため、各トーンの適切なタイミングを見分けることが難しくなります。もしFFTを実行し、シリーズのFFTサイズブロックを進める代わりに少しだけ動かすと、各トーンのパルスタイミングを分離することができます。次の図はオーバーラップ処理の効果を示しています。これは分解能を4倍に増加させています。

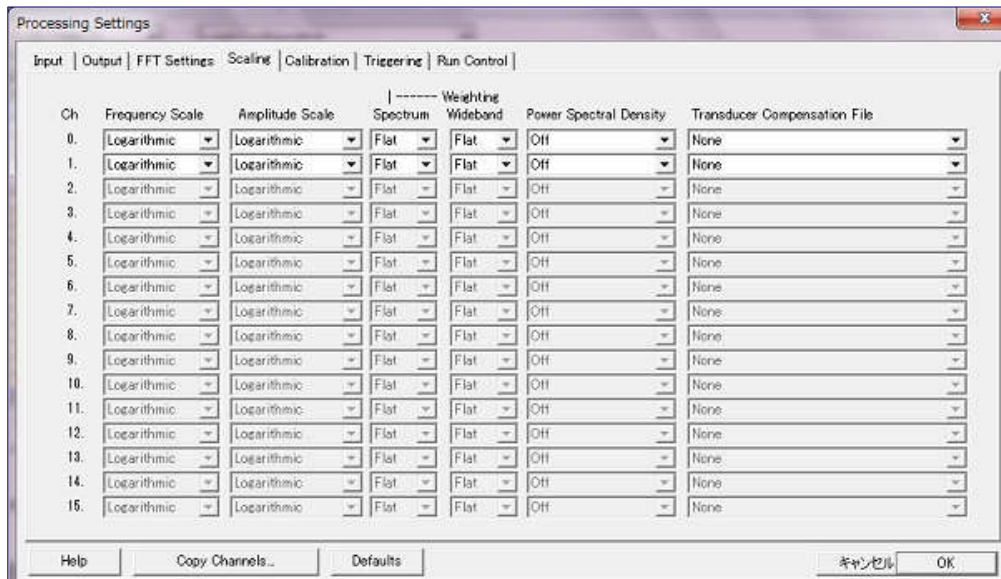


プログラムは自動的に計算して最適なオーバーラップパーセンテージを使います。Time Series, 3-D Surface, あるいは Spectrogram Plot のいずれかでタイムセグメントを選択し、そしてグラフ上でマウス右ボタンをクリックします。現れたポップアップメニューから「Compute and Display Spectrogram」を選択します。アナライザーは、選択したタイムセグメントでグラフを埋めるため自動的に必要なオーバーラップパーセンテージを計算します。詳細は「Right Click Action Menu/右クリックメニュー」の項を参照下さい。

Note:

高「Overlap Percentage」値を設定すると多くの処理時間が必要となります。

Scaling (Plot グラフのスケール)



各チャンネルの周波数とアンプリチュード両グラフ軸のスケールをコントロールします。これらの設定はTime SriesとPhase Plotの垂直軸以外のすべてに適用されます。

Amplitude Axis Scaling (垂直軸・レベルスケール):

基本的なスケールタイプは「Linear/リニア」と「Logarithmic/ログ」の 2 つです。これらの関係は次の通りです。

- リニア値 = 10^(ログ値/20)
- ログ値 = 20 * Log(リニア値)

リニアアンプリチュードスケールはスペクトラムコンポーネントを大きくし観易くします。しかし、非常に小さな部分まで見え過ぎるきらいがあります。従って、コンポーネントサイズが均一な時に適します。ログアンプリチュードスケールは強い信号の低いレベルをクリアに表示します。既定は「Logarithmic」です。

Frequency Axis Scaling (水平軸・周波数スケール):

周波数軸には「Narrowband」と「Octave」の 2 つのスケールモードが用意されています。「Narrowband」スケールでは「Linear」か「Logarithmic」、「Octave」スケールでは「Logarithmic」フォームで表示します。

- Narrowband Scaling - FFT アルゴリズムによって作られるスペクトラムはナローバンドスペクトラムとして知られています。スペクトラルラインは一定のバンドワイズで代表的な狭帯域です。ナローバンドスペクトラルデータはリニアかログスケールで表示しますが、周波数分解はサンプリングレートと FFT サイズで決まります (Resolution = Sampling Rate / FFT size)
- Octave Scaling - オクターブスケールは一定のプロポーション(比例した)バンドワイズとなります。1/1, 1/3, 1/6, 1/9, 1/12, 1/24, 1/48, 1/96(Oct.)フォームから選択できます。1/6, 1/9, 1/12, 1/24, 1/48, 1/96 はオプションです

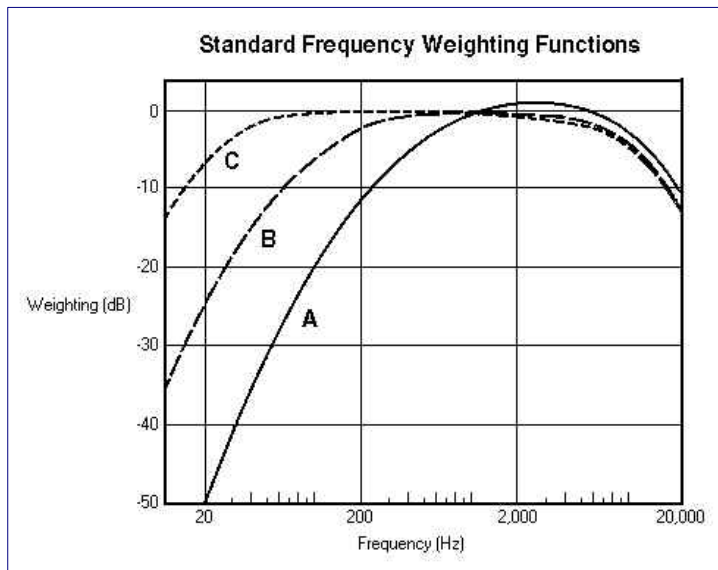
Power Spectral Density (PSD):

「Power Spectral Dencity」オプションはアナライザーを 1Hz バンドスペクトラムにセットします。これはノイズパワー測定に効果的です。例えば、PSD オプション選択をしないでノイズ信号を測ると、アンプリチュードは「FFT size」と「Sampling rate」により変化しますが、オンするとアンプリチュードは「FFT size」と「Sampling rate」に対し無関係になります。

「Units ^2/Hz」もしくは「Units/Sqrt Hz」を選択できますが、「Units ^2/Hz」が通常使用されます。

「PSD」オプションを使用する場合は Amlitude axis(垂直軸)を「Linear」、Frequency axis を「Narrowband」に設定して下さい。

Standard Frequency Weighting (聴感補正カーブ):



ANSI 規格 A、B、C ウェイティングカーブはスペクトルデータとトータルパワー測定に適用できます。ウェイティングが有効にされると、「Total Power」の設定はトータルパワーユーティリティと Spectrum Plot 右側のトータルパワーの値に反映します。

「Flat」はノーウェイティングです。重み関数は人間の耳への最も大きい感度領域である、500～10,000(Hz)の範囲に適応します。

Transducer Compensation (周波数特性の補償):

理想的な変換デバイスの条件は、完全にフラットな周波数特性を有していることです。しかし、実際にはこの条件をなかなか満たしません。本機能では簡単なテキストファイルを使用することにより補償データを作成し、平坦性の確保を実現します。変換デバイスの周波数特性を基に補正データを作成します。

補償ファイルの作成手順:

- ・ 「NOTEPAD.EXE」アプリケーションを起動します。標準的な ASCII テキストファイルに対応するスプレッドシートやワープロ・ソフトでも可能です
- ・ 2つのコラム(欄)にレスポンスデータを記述します
- ・ 先ず左のコラムに周波数を(Hz)単位で入力します
- ・ 右のコラムにはオフセットレベルを(dB)単位で入力します
- ・ データ(周波数)は昇り順に記述します
- ・ コラム間はタブキャラクターで分離します

例)

Freq	Tab	Amp. Data
100.0		-10.0
500.0		-5.0
1000.0		0.0
2000.0		2.0
10000.0		-5.0
15000.0		-15.0

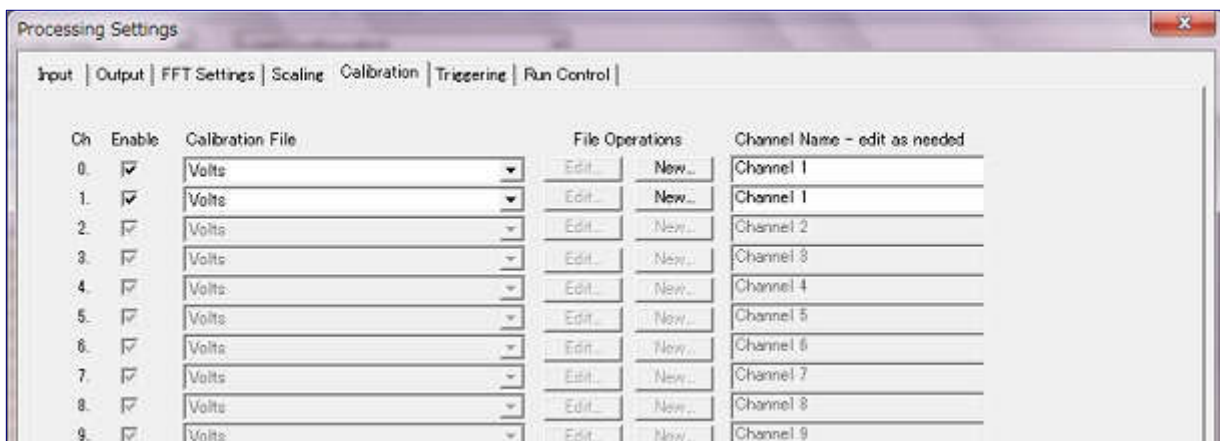
最大 32,768 ポイントのデータを認識します。最小 3 ポイントのデータが必要です。データポイント間は「キュービックスプラインアルゴリズム」で補間されます。最初のコラムにセミコロン(;)が記述されるとコメント行として認識します。「Transducer Compensation」ファイルの拡張子は「.MIC」です。

Notes:

- 「Transducer Compensation」ファイルと「Spectral Overlay」ファイルは同一フォームですから互換性があります。これは作成した「Transducer Compensation」ファイルが正しいか否かを確認するのにも便利です。
- 補償ファイルを手作業で作成すると補間アルゴリズム(Cubic Spline Interpolation)によって予想外の結果を招くことがあります。この場合、ファイルの先頭に「;SPLINE OFF」の記述を加えることによりアルゴリズムの適用をオフすることができます。

Calibration(アンプリチュード/レベル校正)

<Options><Processing Settings>メニュー(もしくは F4 キー)から、「Calibration」タブをクリックしてアクセスします。また、ツールバーのアイコンでもアクセス可能です。



既定では、アナライザーはVolts単位に校正されています。校正オプションを使用すると種々の単位に校正することができます。

Enable: オフ(disabled)にするとアナライザーは校正されず、アンプリチュード値は"Percent Full Scale"もしくは、0.0(dB)がウェーブフォームのクリッピングポイントである"dB Full Scale"になります。既定ではオン(enabled)になります。

Calibration File: どの校正ファイルを適用するか選択します。ファイルネーム"Volts"と"Millivolts"は予約名で、いつでも利用可能です。ユーザが作成する校正ファイルはこのリストに追加表示されます。

File Operations Edit/New: 校正ファイルの作成、編集が可能です。校正には2つの方法があります。

- 1) Directly specify transducer sensitivity – 変換デバイス(Transducer Type)のタイプを選び、そして実際の感度を入力します。例えばマイクロホンでは「45mV/PA」などとします。
- 2) Calibrate to an external source – 明らかになっている外部リファレンスに校正可能です。この情報で、正しいスケールファクタを処理することができます。手順は次の通りです:
 - 安定した基準信号(マイク校正器のような)を選択したチャンネルの入力につないで下さい(DC やノイズ信号を使用しないで下さい)。
 - 基準信号の適切な単位を選択します。もしくは"Custom"を選択し、その右の Amplitude Axis Label 欄にグラフ軸の単位を入力します。
 - 実際の基準信号レベル値を編集欄に入力します(例: 94.0 dB SPL)。

- ・ 基準信号のタイプを"Peak", "RMS", "Peak to Peak"から選択します(マイク校正器は RMS です)。
- ・ 「Measure Input Signal」ボタンを押すと、アナライザーは数秒間アナログデータを取得し、各チャンネルのスケールファクタを演算します。

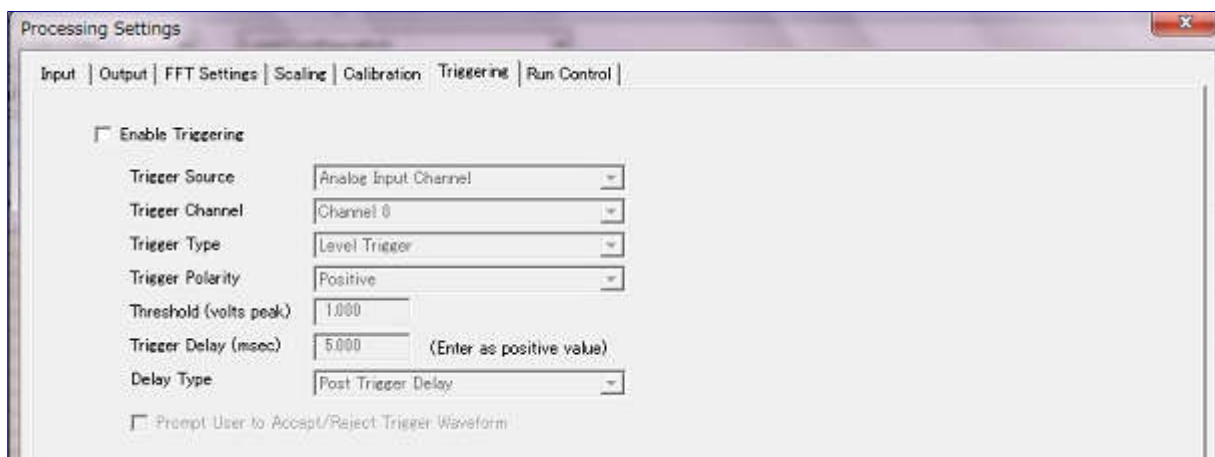
アクセロメータ/加速度計で校正を行うなら、G's (ft/sec²)の基準信号を入れてください。「Convert Spectrum To:」欄がアクティブになり、スペクトルをベロシティに変換します。この変換は Time Series Plot には適用されません。

アクセレーション/Acceleration 単位に較正すると、「Convert Acceleration To」欄がアクティブになり、スペクトラムのベロシティ(Velocity)変換、速度変換を可能にします。この変換は Time Series Plot には適用しません。

Lbel 欄で直接変更することによって、アンプリチュード軸のラベルをカスタマイズすることもできます。

Channel Name: 必要に応じてチャンネルの名前を入力します。これはすべての Plot 上に表示されます。

Triggering(トリガリング)



トリガリングが有効のときはFFTを実行する前に、プロセッサはトリガー状態がないかどうか現在のデータ・ブロックをテストします。

Trigger Source:

- ・ Analog channel - トリガーコンディションの有無を確認するために指定されたアナログ入力チャンネルをテストします。
- ・ External Trigger TTL - DT-9800 ハードウェア(すべてのモジュールが外部のトリガーチャンネルを提供するわけではありません)の専用の外部トリガー入力チャンネルで、TTL 信号(+5.0 V)の発生を待ちます。

Trigger Channel: トリガーコンディションをテストするためにアナログ入力チャンネルを指定します。

Trigger Type: トリガリングモードは「Level」か「edge」にセットできます

- ・ Level triggering - 値が合致しているか、スレッショルド(閾値)を超えている。
- ・ Edge triggering - 急激な変化がないかどうかデータ・ブロックをテストします(2つのサンプル間のレベルの違いがスレッショルドを超えている)。これはパルスタイプ信号のトリガリングに役立ちます。

Trigger Polarity: スレッショルド値の正負をコントロールします。

Trigger Threshold: トリガーレベルを Volts (peak)でセットします。

Trigger Delay: FFT ブロック中のトリガー位置を調整します。

Delay Type: 指定されたトリガーディレーの方向をコントロールします。「Pre-delay」はトリガーポイントを Time Series Plot の右へ、「Post-delay」は左へ移動します。「Delay」をゼロに設定すると、トリガーポイントは FFT データ・ブロックの始めに置かれます。

Prompt User to Accept/Reject Trigger Waveform:

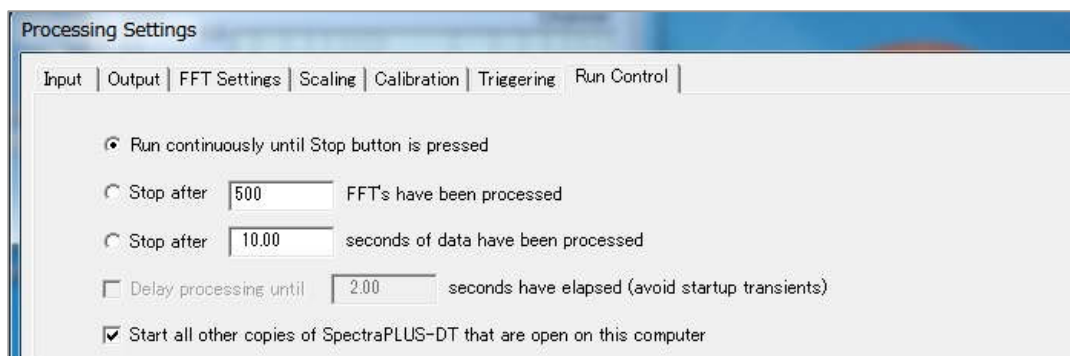
現在のトリガーの承諾の有無を確認するメッセージボックスを表示するようにします。プレビューのため現在のトリガー状況は Time Series Plot に表示されます。もし承諾すると、スペクトラルデータは処理され、そして前のデータと平均化されます。拒否するとトリガーは放棄され、アナライザーは他のトリガー条件に対し待機します。

これはインパクトテストに大変有効です。インパクトハンマーが弾まなかったことを確かめるためにトリガー波形のプレビューを可能にします。また、停止する前に実行するFFT数を設定するためにRun Controlオプションを使えます。承諾したトリガーだけがFFTを実行します。メッセージボックスで「Cancel」をクリックすると、アナライザーは停止します。

Note:

トリガーがオンの時、もし信号がスレッシュホールド以下ならばアナライザーは停止しているように見えるかも知れません。状態はステータスバーで確認して下さい。

Run Control(ラン制御)



Run Control オプションはすべてのモードでアナライザーによって使われます。<Options><Processing Settings>メニュー(もしくは F4 キー)をクリックし、「Run Control」タブを選択します。ツールバーの「Run Ctrl」ボタンでも可能です。

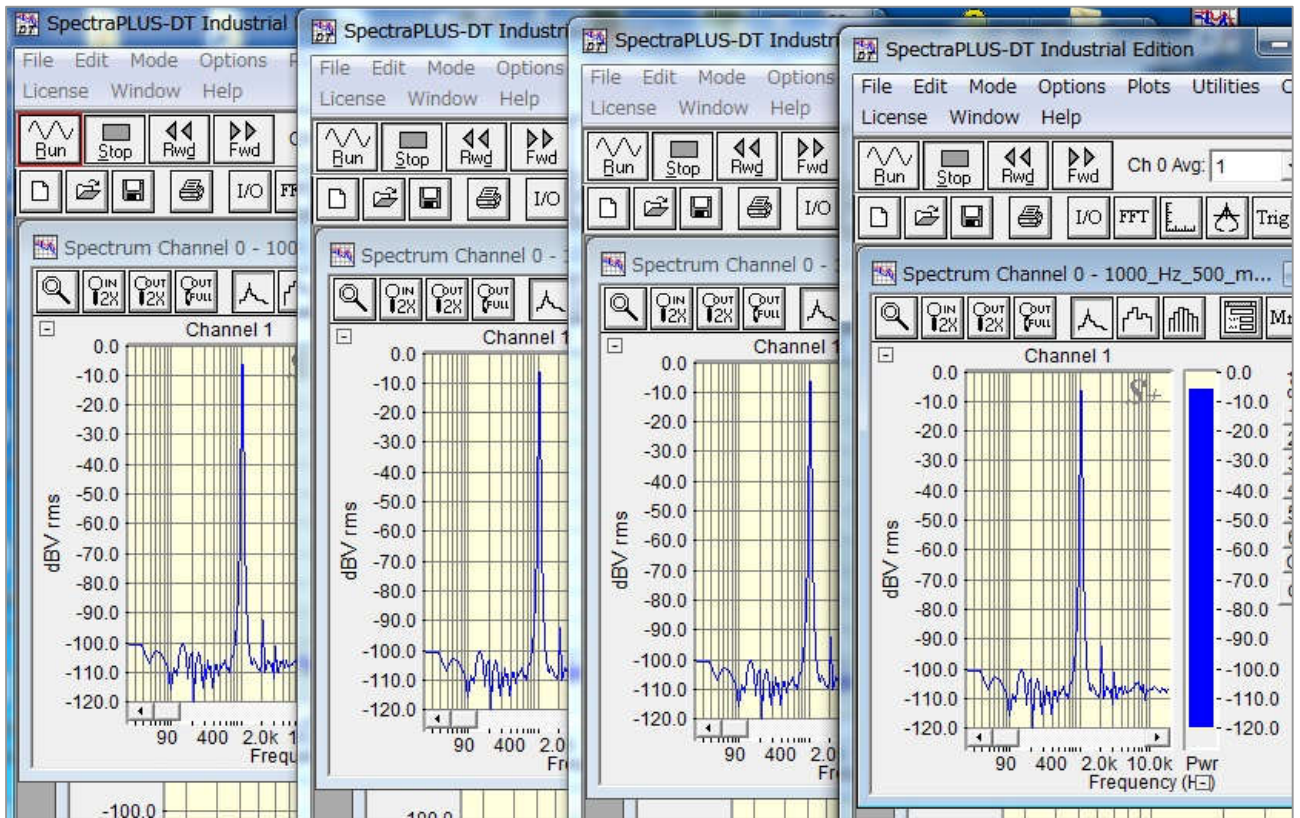
Run continuously until Stop button is pressed: 通常の操作環境で選択します。停止するまでアナライザーがラン/レコードするのを許可します。

Stop after N FFT's have been processed: 指定 N 回数の FFT を実行後、アナライザーを自動的に停止します。

Stop after N seconds of data have been processed: 指定したタイムリミット後、アナライザーを自動的に停止します。

Delay processing until N seconds have elapsed: アナライザーにN秒間ディレーを掛けて起動します。これはいくつかのテスト・シナリオで初期の過渡現象を避けることに役立ちます。

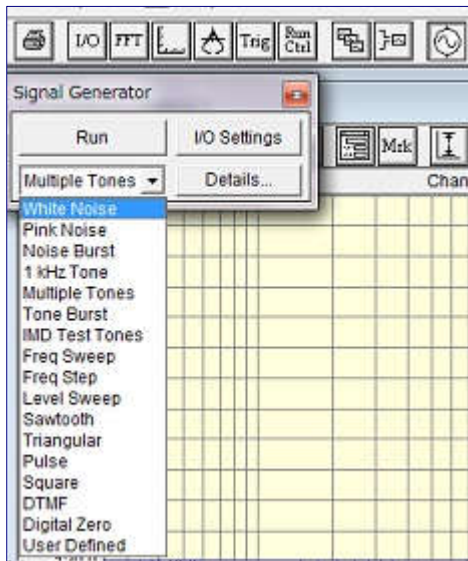
Start all other copies of SpectraPLUS-DT that are open on this computers: 多重起動されている SpectraPLUS-DT の同期「Run/Stop」操作を可能にします。「Run」もしくは「Record」ボタンは赤枠で明示されます。下図では左サイドの SpectraPLUS-DT が該当します。制御受動側の3つは当該オプションを有効にする必要はありません。



Enable the Soacebar key to toggle between Run and Stop: スペースキーによるRun/Stop操作を有効にします。

8章 Utilities(ユーティリティ)

Signal Generator(シグナルジェネレータ)



シグナルジェネレータ機能は種々のテスト信号を生成します。<Utilities><Signal Generator>メニューコマンドを選択するか、または<F11>キー、ツールバーのジェネレータアイコン(図右端)を押すとシグナルジェネレータが起動します。

シグナルジェネレータは DT-9800 デバイスのプレイバックチャンネル(D/A)を使って種々のテスト信号を供給します。ただ、すべての DT-9800 デバイスに対応している訳ではありませんので、[I/O Settings] ボタンをクリックして「Output Device Settings」にアクセスすることにより Windows 互換サウンドカードの利用も可能です。

Basic Operation(基本操作):

- ・ ウェーブフォームのリストをクリックすることによって使用するウェーブタイプを選びます
- ・ [Details]ボタンをクリックして条件を変更します(すべてのウェーブにユーザ・オプションが有る訳ではありません)
- ・ [I/O Setting]ボタンをクリックしてジェネレータ出力レベルを設定します
- ・ [Run]ボタンをクリックしてジェネレータをスタートします
- ・ ジェネレータがランしているときは「Stop」ボタンだけが表示されます

Waveform types(波形の種類):

White Noise:

ホワイトノイズはすべての周波数に対し一定のアンプリチュードを持つ一般的に使用されるランダムノイズ波形です。ナローバンドスペクトラムアナライザーで見ると、その平均スペクトラムはフラットです。カットオフ周波数はアナライザーで使用されるサンプリングレートによって決まります。

Pink Noise:

ピンクノイズは音響で一般的に使用されるランダムノイズ波形です。これには低域周波数により多くのエネルギーがあり、1オクターブあたり3(dB)のレートでアンプリチュードが減衰します。オクターブスペクトラムアナライザで見るとその平均スペクトラムは平坦です。カットオフ周波数はアナライザーで使用されるサンプリングレートによって決まります。

Noise Burst:

ノイズバーストを発生させることができます。「Burst Time」はノイズの長さを定義します。「Cycle Time」は回復の期間を指定します。例えば、Burst Timeが1000msec、Cycle Timeが2000msecであれば、ジェネレータは1秒のノイズ出力に続いて2秒の無音を出力します。ジェネレータを停止するまで繰り返されます。

1kHz Tone:

この1,000(Hz)正弦波信号はひずみ測定に広く使用されます。

Multiple Tones:

このオプションは最大10のトーンをセットすることができます。その周波数とアンプリチュードを設定できます。表示周波数の左のチェック・ボックスでトーンを有効にします。チェックされないトーンは発生しません。トーンの周波数は(Hz)単位で設定します。「Relative Power(相対パワー)」設定は他のトーンとのアンプリチュード比をデシベル(dB)で設定します。

Tone Burst:

トーン・バーストを発生させることができます。「Burst Time」はトーン時間を定義します。「Cycle Time」は反復の時間を指定します。例えば、Burst Timeが1000msec、Cycle Timeが2000msecに設定されると、ジェネレータは2秒間沈黙した後1秒間トーン信号を出力します。ジェネレータを停止するまで繰り返されます。

IMD Test Tones:

Inter Modulation Distortion(IMD/混変調ひずみ)テストは、異なったアンプリチュードの2つの独立しているトーンを必要とします。2つのトーンのうちの高域周波数のアンプリチュードは、低域周波数より通常 12dB(4:1比)低くなります。カスタムトーンを選択するために数個の標準ペアトーンを提供します。

Frequency Sweep:

時間とともに周波数が増加する正弦波を発生します。「Start」と「Stop」の周波数はスイープ周波数範囲を決定します。スタート周波数がストップ周波数より低い場合、スイープ周波数は時間とともに減少します。

スイープタイプは「Linear」か「Logarithmic」のどちらかです。これはスイープ周波数の増減率を決定します

- Jump Back to Start Frequency - ストップ周波数に達すると、スタート周波数から再開します
- Sweep Back to Start Frequency - ストップ周波数に達すると、スタート周波数にスイープバックします
- Stop After a Single Sweep - このオプションをマークすると、スイープを一度実行して停止します。マークを外すと、ジェネレータを停止するまで繰り返されます
- Sweep Time - スイープ時間を指定します。この値はミリセカンドで設定します

スイープ正弦波は周波数特性テストに役立ちます。「AVG」ブロックサイズを「1」に設定します。そして「Peak Hold」オプションを使用して、スイープ周波数のピークアンプリチュードを表示させます。「FFT size」が大きいほどスイープ時間は長くなります。大きい「FFT size」の時にあまり速くスイープすると、結果が不鮮明になります。

Frequency Step:

周波数ステップウェーブフォームは離散的なステップで周波数を変える正弦波を発生させます。スタートとストップ周波数は範囲を決定します。スタート周波数がストップ周波数より低い場合、スイープ周波数は時間とともに減少します。

「Step Size」は「Linear」か「Octave(対数)」のどちらかです。

- Linear Steps - ステップサイズを(Hz)で指定します。アナライザーでナローバンドスケールリングが使用される場合にフィットします
- Octave Steps - 使用するオクターブステップを指定します。アナライザーでオクターブバンドスケールリングが使用される場合にフィットします
- Jump Back to Start Frequency - ストップ周波数に達すると、スタート周波数から再開します
- Sweep Back to Start Frequency - ストップ周波数に達すると、スタート周波数にスイープバックします
- Stop After a Single Pass - このオプションをマークすると、スイープを一度実行して停止します。マークを外すと、ジェネレータを停止するまで繰り返されます
- Dwell Time - 各ステップの信号維持時間を指定します。この値はミリセカンドで設定します

ステップ正弦波は周波数特性テストに役立ちます。「AVG」ブロックサイズを「1」に設定します。そして「Peak Hold」オプションを使用してスイープ周波数のピークアンプリチュードを表示させます。

Level Sweep:

レベルスイープは時間とともにアンプリチュードが変化する正弦波を発生させます。スタートとストップレベルをデシベル(dB)で、そして周波数とスイープ時間を選択します。スタートレベルがストップレベルより低い場合、信号レベルは時間とともに減衰します。「Sweep Time」はスイープ時間をミリ秒で指定します。ジェネレータを停止するまでスイープは繰り返されます。

Sawtooth:

こぎり波を生成します。メイン画面で「Details」ボタンをクリックすると、のこぎり波の周波数とスロープを調整できます。「Frequency」は一秒にいくつのスロープが発生するかを指定します。「Slope」はスロープが時間とともに上昇するか、低下するかを指定します。

Triangular:

三角波を生成します。メイン画面で「Details」ボタンをクリックすると、三角波の周波数を調整できます。「Frequency」は一秒にいくつの三角波が発生するかを指定します。

Pulse:

Pulse はシングルパルスを生成します。「Details」ボタンをクリックすると、パルス幅と反復間隔を調整できます。「One Shot Pulse」オプションがチェックされると、パルス出力後にジェネレータは自動的に止まります。別のパルスを発生させるには[Run]ボタンを押します。

Square:

連続したパルスから成ります。メイン画面で「Details」ボタンをクリックすると、方形波の周波数とデューティサイクルを調整できます。「Frequency」は1秒に発生するパルスの数を設定します。「Duty Cycle」は間隔の割合(%)でパルス幅を決定します。

User Defined:

信号ソースとしてWAVフォーマットファイルを使用することができます。[Stop]ボタンを押すまで信号は生成されます。[Pick]ボタンをクリックしてWAVファイルを選びます。シグナルジェネレータがこのファイルをメモリにロードする為、コンピュータは十分なRAMを登録していなければなりません。SpectraPLUS-DTはASCIIテキストかバイナリーファイルからユーザーオリジナルのWAVファイルを造ることができます。詳細に関しては<File><Import>メニューの項を参照して下さい。

DTMF:

キーストリングのためのデュアルトーンパルス(touch tones)を生成します。またパルス幅、スペース及びツイストを調整できます。

「Twist」は(dB)で入力します。正の値は高域周波数のアンプリチュードが低域周波数より大きくなることを意味します。負の値は逆です。

完全な拡張DTMFキャラクターセット(1234567890*#ABCD)がサポートされます。

Digital Zero:

すべてゼロの静寂信号/無音状態を生成します。

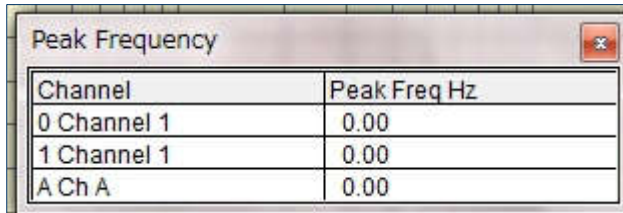
Save Signal To Wave File (WAV ファイルの保存):

<Utilities><Save Signal To Wave File>メニューをクリックすると、生信号をWAVファイルとして生成することができます。この機能を使うにはシグナルジェネレータが起動されていなければなりません。

信号をリピートすることによって生じるWAVファイルの長さを修正するオプションが与えられます。そして、ウェーブデータを含むファイルの名前を求められます。

これはカスタムウェーブフォームを作成するのに役立ちます。種々のコンポーネント(トーン、ノイズ、スイープ、パルスなど)で作成して下さい。そして、必要とする信号とタイミングでシングルウェーブファイルを生成するために編集機能を使用します。このカスタムファイルのロードは前出の「User Defined」を使用して行います。

Peak Frequency(ピーク周波数)



Channel	Peak Freq Hz
0 Channel 1	0.00
1 Channel 1	0.00
A Ch A	0.00

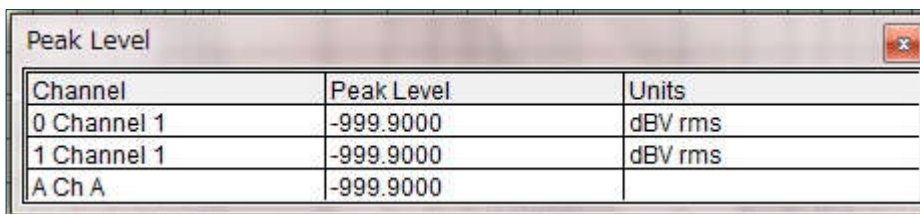
ピーク周波数ユーティリティは全有効レンジ(表示レンジではありません)の最も強いスペクトル成分の周波数を表示します。ユーザは必要に応じてこのユーティリティウィンドウをリサイズできます。新しいスペクトルが計算されると内容は常にアップデートされます。ウィンドウにはチャンネルインデックスと名前を表示します。

チャンネル名は「Calibration Settings」ダイアログで編集することができます。

Note:

ウィンドウ上で右クリックすると内容をクリップボードにコピーします。

Peak Amplitude(ピークアンプリチュード)



Channel	Peak Level	Units
0 Channel 1	-999.9000	dBV rms
1 Channel 1	-999.9000	dBV rms
A Ch A	-999.9000	

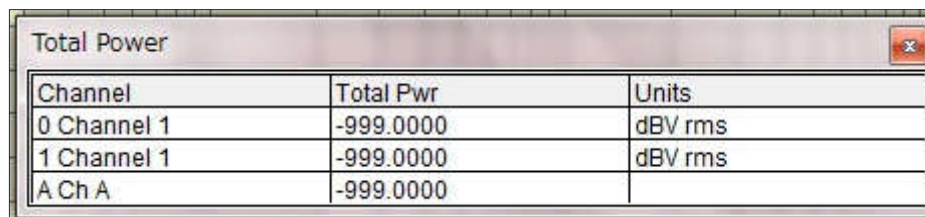
ピークアンプリチュードユーティリティは全有効レンジ(表示レンジではありません)の最も強いスペクトル成分のアンプリチュードを表示します。ユーザは必要に応じてこの

ユーティリティウィンドウをリサイズできます。新しいスペクトルが計算されると内容は常にアップデートされます。ウィンドウにはチャンネルインデックスと名前を表示します。チャンネル名は「Calibration Settings」ダイアログで編集することができます。

Note:

ウィンドウ上で右クリックすると内容をクリップボードにコピーします。

Total Power(トータルパワー)



Channel	Total Pwr	Units
0 Channel 1	-999.0000	dBV rms
1 Channel 1	-999.0000	dBV rms
A Ch A	-999.0000	

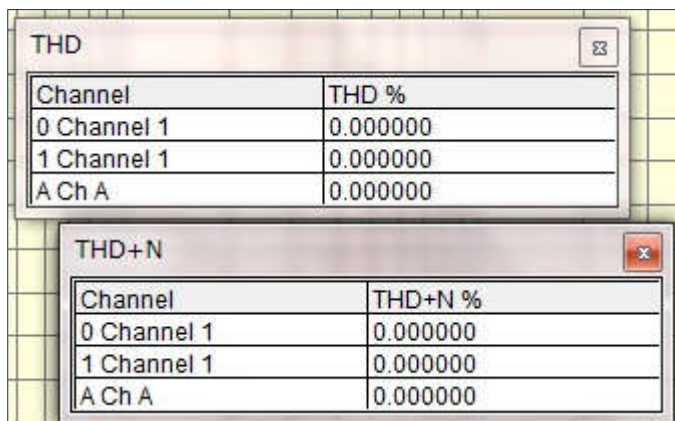
トータルパワーユーティリティは全スペクトルのトータル RMS パワーレベルを表示します。この値は Spectrum Plot 右側に表示されるのトータルパワーバーと等価です。このユーティリティに使用されるウェイトイングカーブ/聴感補正(Flat、A、B または C)は、「Scaling」オプションダイアログボックスで設定されます。これは SPL の単位に較正され、「Sound Level Meter」の表示と等価です。

ユーザは必要に応じてこのユーティリティウィンドウをリサイズできます。そして、新しいスペクトルが計算されると常にアップデートされます。ウィンドウにはチャンネルインデックスと名前を表示します。チャンネル名は「Calibration Settings」ダイアログで編集することができます。

Note:

ウィンドウ上で右クリックすると内容をクリップボードにコピーします。

Total Harmonic Distortion(全高調波ひずみ)



THD	
Channel	THD %
0 Channel 1	0.000000
1 Channel 1	0.000000
A Ch A	0.000000

THD+N	
Channel	THD+N %
0 Channel 1	0.000000
1 Channel 1	0.000000
A Ch A	0.000000

実際のオーディオ装置には「ひずみ」と呼ぶ望まれない影響をもたらす因子が存在します。

全高調波歪/Total Harmonic Distortion(THD)と(THD+Noise)はこの量の測定に広く使われる値です。

測定値は「%」で表されます。THD 値が低ければ低いほどクリアです。

Total Harmonic Distortion(THD):

高調波パワーと基本波パワーの比率です。ピーク周波数(基本波)を見つけるためにスペクトル全体を捜し、次に高調波周波数のトータルパワーを計算することによって処理されます。THD レベルは「全高調波のパワー」対「基本波パワー」の比率として計算されます。残留ノイズは含まれません。

Total Harmonic Distortion+Noise(THD+N):

「高調波パワー+ノイズ」と「基本波パワー」の比率です。ピーク周波数(基本波)を見つけるためにスペクトル全体を捜し、次に高調波周波数のトータルパワー+ノイズを計算することによって処理されます。THD レベルは「全高調波+ノイズのパワー」対「基本波パワー」の比率として計算されます。

同一信号では、THD+N 値は THD 値より大きくなります。

代表的な THD 測定は 1kHz の基準信号を使用します。この信号は被測定オーディオデバイス通過し、そしてスペクトルが算出されます。ランダムノイズコンポーネントをキャンセルして測定精度を高めるには長いアベレーシングの使用が有効です。

D/A コンバータと基準信号の両方の品質が測定結果に反映することを理解して下さい。被測定装置の性能を下回らない品質を確保するよう注意すべきです。シグナルジェネレータの品質もまた、使用されるサウンドカードの THD 性能に依存しています。

必要に応じてこのユーティリティウィンドウをリサイズできます。新しいスペクトルが計算されると内容はアップデートされます。ウィンドウにはチャンネルインデックスと名前を表示します。チャンネル名は「Calibration Settings」ダイアログで編集することができます。

ウィンドウ上で右クリックすると内容をクリップボードにコピーします。

Notes:

- ・ Smoothing Window/窓関数は「Hanning」か「Blackman」を推奨します
- ・ 適切なスペクトル分解を得るために 2048 以上の FFT サイズを推奨します

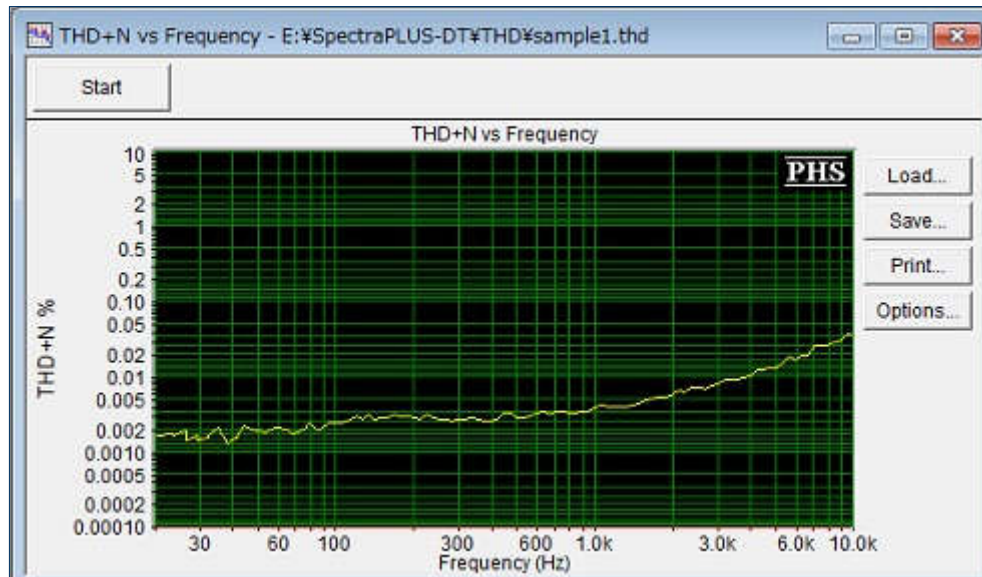
THD+N vs Frequency(全高調波ひずみ+ノイズ対周波数スイープ)

指定した出力デバイスを介してステップ周波数信号を発生させ、THD+N 対周波数特性を算出してPlotに表示します。このユーティリティによりひずみ特性をスピーディに測定することができます。

<Utilities>><THD+N vs Frequency>メニューをクリックするか、またはツールバーで[THD Freq]ボタンをクリックして起動します。ユーティリティはステップ周波数スイープ信号を生成します。信号は出力デバイスを介して被測定器に供給されます。テスト中、オペレータの操作は必要ありません。コンピュータはすべてのステップを実

行して結果を表示します。

測定結果は印刷あるいは、ディスク保存することができます。Plot を右クリックするとオプションメニューがポップアップし、他のアプリケーションで利用することができるデータやグラフィメージをクリップボードにコピー可能です。



Options:

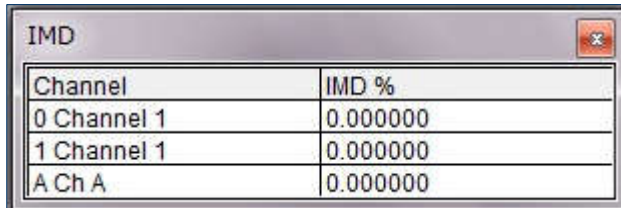
Plot 右側の「Options」ボタンをクリックすることによって、THD+N vs Frequency オプションダイアログボックスが起動します。ここで周波数スパン、表示カラー、および他のいくつかのオプションを変更することができます。

- Start and Stop Frequencies - 測定の周波数帯域を設定します。高域限界周波数はアナライザーで使用されるサンプリングレートの 1/2 未満であるべきです
- Channels - 2 チャンネルまで選択できます
- Frequency Step size - Hz でリニアステップサイズあるいは、オクターブスケール(1/1 □ 1/96)を選択します。ステップサイン波はステップあたり 1 □ 2 秒確保します。ステップサイズを減少させるとテスト実行時間は増大します
- Spectral Weighting - 使用するウェイトイング(聴感補正)カーブをセットします
- Plot Top - Plot アンプリチュード(垂直)軸のトップ(最上部)値を設定します。値は「%」でセットします
- Plot Bottom - Plot アンプリチュード(垂直)軸のボトム(最下部)値を設定します。値は「%」でセットします
- Frequency Axis Scaling - グラフ周波数(水平)軸のスケールフォームをセットします
- Amplitude Axis Scaling - グラフアンプリチュード(垂直)軸のスケールフォームをセットします
- Test Limits - 各チャンネルのリミット値を記述したファイルを指定できます。リミットデータは分離したトレースとして表示されます。リミットファイル形式はオーバーレイやマイクファイルと同じで、2 つのコラムとタブがあるシンプルな ASCII テキスト・ファイルです。最初のコラムは周波数、そして第 2 コラムは THD リミット値を含みます。周波数は順方向となります
- Prompt to Save - 新しいテストを始める前に、データを保存するように警告するか否かをコントロールします
- Signal Output Level - dB フルスケールでテスト信号出力レベルを設定します。0.0dB は、100%フルスケールサインウェーブを出力する信号レベルを生成します。ほとんどのサウンドカードには出力ボリュームを制御する能力があるので、フルスケールが必ずしも「最大出力」信号レベルを意味しません

Notes:

- テストソースとして、内蔵シグナルジェネレータが出力するステップ周波数信号を使用しなければなりません。

InterModulation Distortion(IMD/混変調ひずみ)



Channel	IMD %
0 Channel 1	0.000000
1 Channel 1	0.000000
A Ch A	0.000000

InterModulation Distortion(IMD)は 2 つのトーンの混合によって引き起こされるひずみです。

2 つの信号が装置に入力されると信号の変調や混合が起こります。まず IMD レベルは、スペクトルの 2 つの最も強いトーンの周波数とアンプリチュードを計算する

ことによって処理されます。そして、それぞれの混変調が作り出す周波数のトータルパワーが計算されます。IMD は「混変調パワー」対「トーンパワー」の rms 合計の比率です。

代表的な IMD 測定は、250Hz とそれより 12dB(4:1 比)低い 8020Hz トーンを使用します。ランダムノイズコンポーネントをキャンセルして測定精度を高めるには長いアベレージングの使用が有効です。

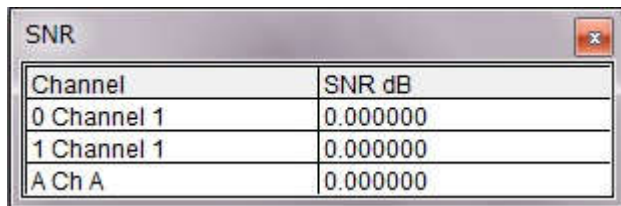
このユーティリティウィンドウは必要に応じリサイズできます。新しいスペクトルが計算されると内容はアップデートされます。ウィンドウにはチャンネルインデックスと名前を表示します。チャンネル名は「Calibration Settings」ダイアログで編集することができます。

ウィンドウ上で右クリックすると内容をクリップボードにコピーします。

Notes:

- Smoothing Window(窓関数)は「Hanning」か「Blackman」を推奨します。
- 適切なスペクトル分解を得るために 4096 以上の FFT サイズを推奨します。

Signal to Noise Ratio(SNR/SN 比)



Channel	SNR dB
0 Channel 1	0.000000
1 Channel 1	0.000000
A Ch A	0.000000

「ピークパワーレベル」対「トータルノイズレベル」の比率(dB)です。SNR は全スペクトルからピーク周波数を捜し、そして残りのスペクトルでトータルノイズパワーを計算することによって処理されます。次に「ノイズパワー」対「ピークパワー」の比率として計算され、デシベル(dB)で表示されます。

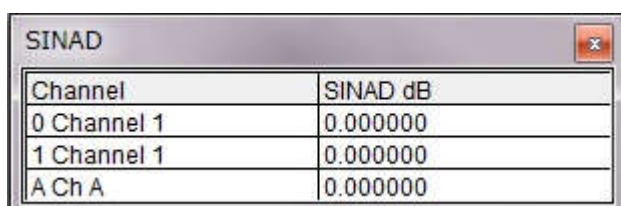
必要に応じてこのユーティリティウィンドウをリサイズできます。新しいスペクトルが計算されると内容はアップデートされます。ウィンドウにはチャンネルインデックスと名前を表示します。チャンネル名は「Calibration Settings」ダイアログで編集することができます。

ウィンドウ上で右クリックすると内容をクリップボードにコピーします。

Notes:

- Smoothing Window(窓関数)は「Hanning」か「Blackman」を推奨します。
- 適切なスペクトル分解を得るために 2048 以上の FFT サイズを推奨します。

Signal to Noise and Distortion(SINAD)



Channel	SINAD dB
0 Channel 1	0.000000
1 Channel 1	0.000000
A Ch A	0.000000

SINAD は通常、FM 受信機に関連している一般的な感度測定です。それは Signal、Noise、Distortion の頭文字であり、(S+N)/N と等価で(dB)で表されます。信号が雑音より強ければ SINAD 値は SNR 値に近づきます。さもなければ SINAD は SNR より大きくなります。

必要に応じてこのユーティリティウィンドウをリサイズできます。新しいスペクトルが計算されると内容はアップ

データされます。ウィンドウにはチャンネルインデックスと名前を表示します。チャンネル名は「Calibration Settings」ダイアログで編集することができます。

ウィンドウ上で右クリックすると内容をクリップボードにコピーします。

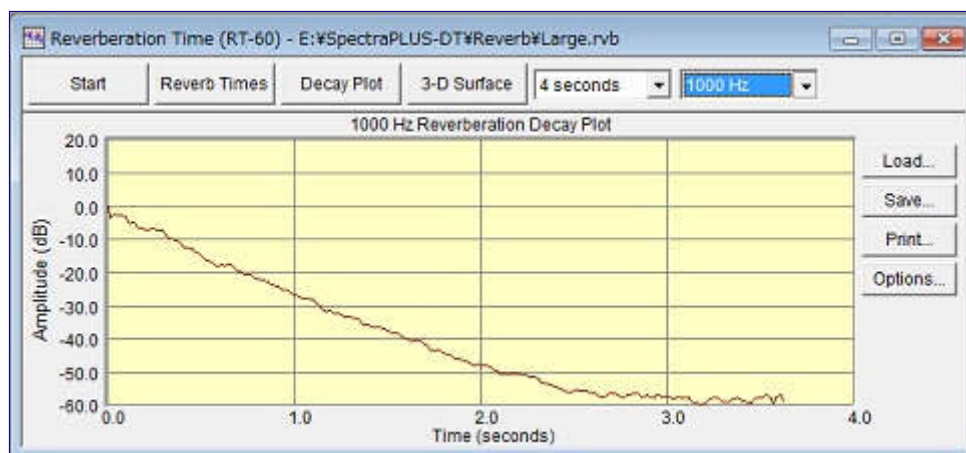
Notes:

- Smoothing Window (窓関数) は「Hanning」か「Blackman」を推奨します。
- 適切なスペクトル分解を得るために 2048 以上の FFT サイズを推奨します。

Reverberation Time (RT60/残響時間)

Reverberation Time:

RT は残響時間とスペクトラルの測定解析を提供する機能です。「Reverberation」はバースト後の信号のエネルギー減衰を意味し、その値は「RT-60」と呼ばれてエネルギーが「60dB」減衰するまでの時間で表されます。



このオプション

は RT-60 測定に使用する信号を出力するためのピンクノイズ・ジェネレータ機能を含んでいます。D/A デバイス出力からパワーアンプ経由で高品質なスピーカにそれを接続しなければなりません。通常、1台(複数のスピーカより)のスピーカがテストに使用されます。より正確な RT-60 データを得るには、環境を「飽和状態」にするようにアンプ出力レベルを調整する必要があります。しかし、アンプがクリップしたり、またはスピーカがひずみを発生させるほど高くするべきではありません。

テストの間、オペレータの操作は全く不要です。コンピュータはオーディオバンドのディケイカーブを生成するためにすべてのステップを実行します。まず環境を飽和状態にするのに十分な時間ノイズソースを放出します(すなわち、均衡に部屋中に充満するように)。そして音が停止され、信号の衰弱を捕らえて残響特性が詳細に分析されます。

内蔵ピンクノイズではなく、バロンポップやガンショットなどの信号を使用する場合の操作手順:

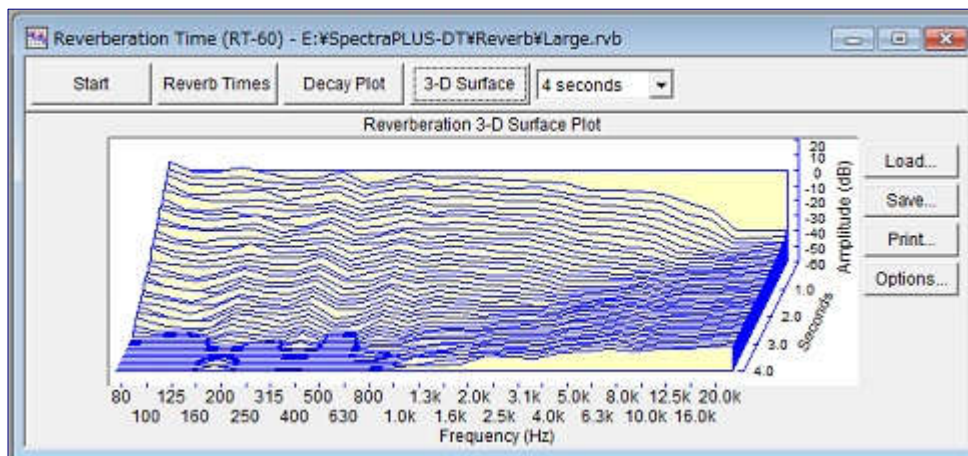
1. Recorder モードを使用して、信号とその減衰をキャプチャーします
2. Time Series Plot を使用して波形を表示します。そして矢印アイコンを選択し、Plot 上の適切な信号と減衰を示すレンジをクリックアンドドラッグします。ボタンをリリースすると選択された時間セグメントは反転表示されます
3. Plot 上で右ボタンをクリックします。そしてポップアップメニューから「Compute and Display RT-60」を選択します
4. Reverb Time 機能が起動されて処理結果を表示します
5. バーストのピークで Decay Plot を右クリックして「Set Decay Start Time」を選ぶことにより、測定が始まる時間を微調整することができます。そして新しい起点に基づき再処理を実行します

リバーブ信号が分析されるとグラフが表示され、そして印刷やディスク保存を行うことができます。グラフを右クリックすると、他のアプリケーションで使用するためにクリップボードにデータイメージをコピーするオプションメニュー

ユーがポップアップします。

Reverberation Plots:

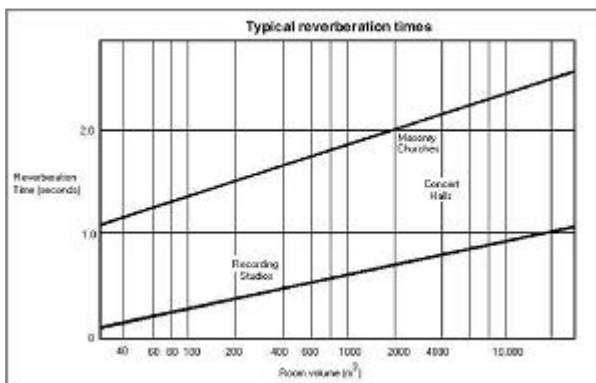
- Reverb Times - 「残響時間vs周波数」を示すバーグラフです。80Hz~20kHzを1/3 Oct.バンドで表示します。グラフ右端のバーはワイドバンドデータを示します。既定表示値は「RT-60」データですが、ツールバーのコントロールメニューで「RT-10/20/30/40/50」を任意に選択することができます。グラフ上のバーをダブルクリックすると、そのポジションの周波数バンド減衰特性を表示します
- Reverb Decay - 選択周波数バンドの「減衰レベルvs時間」を示すグラフです。周波数の選択、グラフ水平軸の時間レンジの変更はツールバーのコントロールメニューで行います。これにより例えば、特定の時間を観るために画像を徐々にズームすることができます
- 3-D Surface - 減衰特性を三次元グラフで表示しますので、全周波数バンドの減衰特性を俯瞰することができます。グラフ上の任意のポジションをダブルクリックすると、その周波数バンドの減衰特性を表示します。グラフ水平軸の時間レンジの変更はツールバーのコントロールメニューで行います



Reverberation Options:

オプションダイアログはグラフ右の[Option]ボタンをクリックすると現れます。いくつかのオプションとグラフ表示配色を設定することができます。

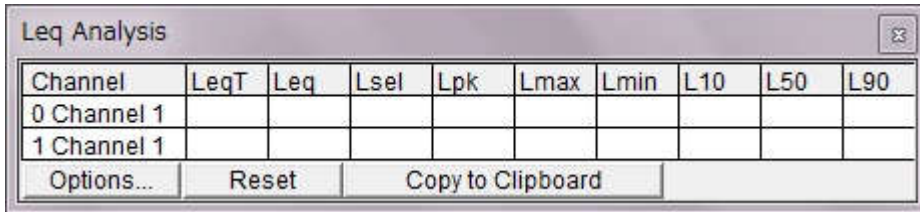
- PlotTop - Reverb Decay と 3-D Surface Plot のアンプリチュード軸の最大値を設定します
- Plot Range - Reverb Decay と 3-D Surface Plot のアンプリチュード軸のレンジ幅を設定します
- Maximum Decay Time - 測定時間長を設定します。この値を必要以上に長くするとディスク消費も大きくなり問題が生じますが、推定残響時間よりも長く設定することが肝要です。既定値は 4 秒です
- Extrapolation Method - 現実的な測定環境では、パースト信号レベルと暗騒音レベルの比を 60dB 以上確保するのは困難ですから、このオプションで「RT-10/20/30」を設定します。設定を変更するとデータを再処理して再描画します



- Smoothing Method - 残響エネルギーの実体は非常に複雑です。その結果不規則性を呈しますがより適切な評価を行うための技術がスムージングテクニックです。「None」は機能をオフします。「Average」は 100msec ムービングウィンドウ値を表示します。「Schroeder」はリバースインテグレーションを行います。既定値は「Average」です。設定を変更するとデータを再処理して再描画します
- Typical Reverberation Times - 以下のグラフは様々な環境の典型的な RT-60 を示しています

す

Equivalent Noise Levels (等価ノイズレベル)



Channel	LeqT	Leq	Lsel	Lpk	Lmax	Lmin	L10	L50	L90
0 Channel 1									
1 Channel 1									

Options... Reset Copy to Clipboard

<Utilities><Equivalent Noise Levels>メニューをクリックし Leq 分析ウィンドウを開きます。そして、アナライザーをランすると Leq 値が計算されます。

- Leq(t) - 等価ノイズレベルはユーザの指定した「参照時間」まで変換されます。例えば、90dBA ノイズの時間幅 3.6 秒を選択し、そして 1 時間(3600 秒)の参照時間を指定すると 60dBA の等価ノイズレベルを生成します
- Leq - 選択した時間セグメントの等価ノイズレベルです
- Lsel - 等価ノイズレベルは 1 秒間に変換されます
- Lpk - 選択した時間セグメントのピークレベルです。このパラメータにアベレージングは使用されません
- Lmax - 選択されたデータの最大レベルです
- Lmin - 選択されたデータの最小レベルです
- L10 - 時間の 10%を超えたレベルです
- L50 - 時間の 50%を超えたレベルです
- L90 - 時間の 90%を超えたレベルです

「Options」ボタンをクリックして、これらの計算のためのオプションを設定します。必要に応じてアベレージングとウェイトイング(聴感補正)カーブを選択できます。

- Calculation Interval: ノイズ値を再計算するための間隔を決めます
- Specific Intervals: Leq 値が指定された間隔で再計算されます
- When the Analyzer Stops: アナライザーの[Stop]ボタンが押されたときだけ Leq 値が再計算されます。アナライザーのスタートから停止時まで、すべてのデータが Leq 値に関与します(アナライザーが停止するまで値は利用できません)
- Channel: 各チャンネルは異なる Leq 設定が可能です。「Copy settings to all channels」機能を使うとすべてのチャンネルが一様に設定されます

Notes:

Leq と LeqT 値は、同じタイムサンプル(averaging と weighting だけでなく)が使われる場合は合致します。LeqT の参照間隔は「Leq options」ダイアログで直接設定します。Leq タイム間隔はアナライザーの[Run]ボタンを押してから[Stop]ボタンを押すまでの間隔と同等です。例えば、LeqT の参照時間を 60 秒にセットして Real-Time モードで 60 秒間ランすると、Leq と LeqT 値は同一(もしくは近似値)になります。計算の間隔は値が計算されて表示される時間を設定することを可能にしますが、Leq の間隔自体は変えられません。この例のようにも、計算の間隔を 4 秒にセットすると 60 秒に到達するに従って LeqT と Leq 値がゆっくりと一点に集まります。

「Calibration」設定と「Transducer Compensation」設定はこれらの計算に使用されます(もし有効になっていれば)。

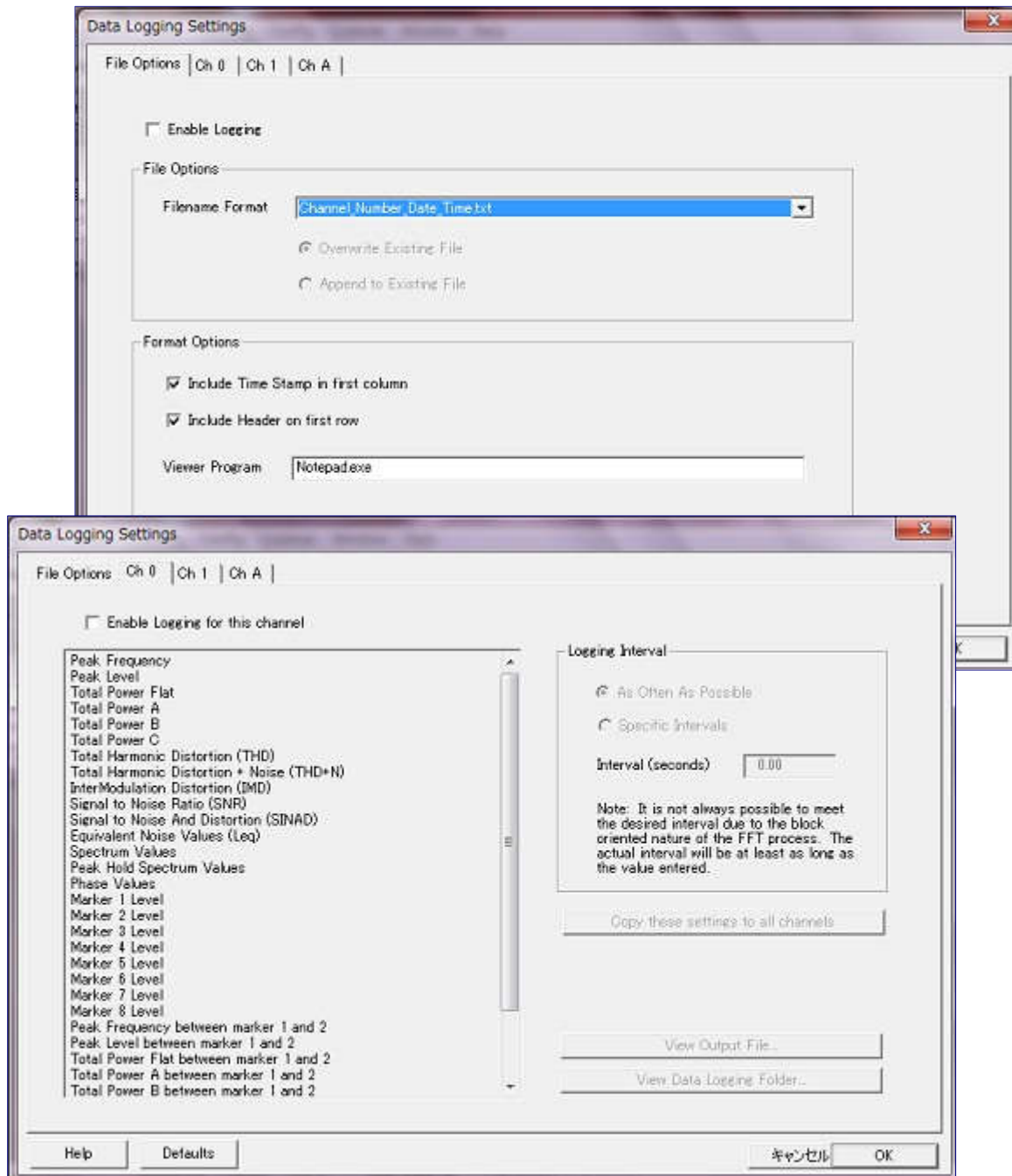
Data Logging(データロギング)

データロギングはデータを取得したタイムスタンプを付加し、スペクトラルパラメータを含むテキストデータファイルを生成する機能です。チャンネル毎にファイルが生成されます。

File Options:

- Enable Logging: ロギング状態をコントロールします
- Filename Format: 各チャンネルで使うファイルネームの仕様を選択します
- Overwrite Existing file: 同名ファイルが存在した場合上書きします

- Append to Existing file: 同名ファイルが存在した場合書き加えます
- Include Time Stamp in first column: ログファイルの各行にタイムスタンプを書き込みます
- Include Header on first row: ファイルの最初の行にヘッダーを生成します
- Viewer Program: データロギングファイル(ASCII テキスト)を観察表示するためのテキストエディタを選択します。既定アプリケーションは Notepad です



Logging Options for each cahnnel:

- Output Data Type: データファイルに書き込む内容を決めます
- Logging Interval: スペクトラデータがログファイルに何回書き込まれるかを設定します
- View Output File: 指定したアプリケーションでデータロギングファイルを開きます
- View Output Folder: エクスプローラを使いデータログフォルダを開きます。既定フォルダは <File><Set Paths>メニューで確認できます

Output example(ログファイル例) :

Data Logging On:Fri Sep3010:43:411995

Time Freq(Hz) Amplitude(dB)

10:43:41 527.56 -45.60
10:43:42 495.26 -54.03
10:43:43 495.26 -49.83
10:43:44 1399.66 -59.01
10:43:48 1399.66 -59.14
10:43:49 516.80 -56.38
10:43:50 549.10 -53.37
10:43:51 624.46 -43.66
10:43:52 559.86 -37.96

9章 Licensing(ライセンス)

SpectraPLUS ソフトウェアはオーソライゼーションキー(Authorization Key) 技術によって使用権がプロテクトされています。従って、取得した使用権(ライセンス)は一台のコンピュータにのみ発行されてご使用頂けることとなります。もし、複数のコンピュータで同時にプログラムをご使用になる場合は、コンピュータ台数相当分のライセンス(マルチライセンス)をご購入頂く必要があります。

次の3つのいずれかの方法でプロテクトされます。

- ・ ハードディスク上に保存されるソフトウェアキー
- ・ USB ポートに装着するハードウェアキー
- ・ ネットワーク上のコンピュータに装着するハードウェアキー

<License><Status and Authorization>メニューによってライセンス状態を調べることができます。利用可能なオプションとそれらの現在のライセンス状態(Authorized もしくは xx days)がリスト表示されます。プログラムはソフトウェアキーで使用することを基本としますが、頻繁にコンピュータ間のライセンス転送をする場合は、ライセンス失効事故を回避するためにも物理的なハードウェアキーを使用することをお勧めします。ハードウェアキーの場合は専用デバイス(HARD-KEY DONGLE)が必要となり、別途そのご購入代金が発生します。

Software Key Method(ソフトウェアキー):

この方式はコンピューター意の「Site Code」と「Authorization Key」技術を使用します。

「Site Code」は各コンピューター意のユニークなコード番号であり、プログラムの使用を許可するための「Authorization Key」を生成するために使われます。「Site Code」がコンピュータ固有のものですから、弊社あるいは販売代理店がお客様からそのコードを受領するまで、対応する「Authorization Key」を提供することができません。<License><Status and Authorization>メニューで表示されるダイアログの「Authorize」ボタンをクリックすることによって、容易にこの「Site Code」を確認することができます。

ライセンスをご購入頂くには、「Authorize」ボタンをクリックして「Site Code」を表示します。そして「Print Site Code」ボタンをクリックしてライセンス取得申請シートを印刷して下さい。そして、販売代理店あるいは弊社にお送り下さい。数日後、「Authorization Key」発行案内通知を受け取ったらこのダイアログボックスに戻り、キーコードを入力します。

「Site Code」はプログラムを終了しても変化しませんが、プログラムを再インストールすると変化します。従ってライセンス取得申請中は、再インストールをせずに「Authorization Key」発行案内をお待ち下さい。再インストールを実行すると「Authorization Key」が合致しなくなりその結果、ライセンス失効事故を誘発します。「Authorization Key」の無償再発行はお請け致しかねます。

Adding additional options to the Software Key(オプションの追加):

オプションをいつでも追加することができます。弊社あるいは販売代理店に連絡してオプションを購入し、そして前述同様にライセンス取得申請シートを印刷して送付します。その後、新しい「Authorization Key」をお知らせします。<License><Status and Authorization>メニューでダイアログを起動し、「Authorize」ボタンをクリックして下さい。そして、ダイアログに「Authorization Key」を入力します。これで購入したオプションにライセンスが発行されます。

Transferring the Software Key to another computer(コンピュータ間ライセンス転送/移植):

ソフトウェアキーを他のコンピュータに転送/移植するには、<License><Status and Authorization>メニューでダイアログを開き、「Transfer Key」ボタンをクリックし説明に従います。二つの基本的な方法があります。

- ・ LAN 経由
- ・ フロッピーディスクあるいは USB メモリー経由

両方法ともソフトウェアキーを受け取るコンピュータに、あらかじめ SpectraPLUS をインストールして置かな

ければなりません。2)の方法ではまず、転送先コンピュータにインストールされている SpectraPLUS の [Register a Floppy...]ボタンにより、フロッピーディスクや USB メモリーにライセンスされていない SpectraPLUS の固有情報を登録する必要があります。登録した後、ライセンスされている(転送元) SpectraPLUS の [Transfer License Out To...]ボタンによりライセンス情報をメディアに書き出します。そして、転送先 SpectraPLUS の [Transfer License In From...]ボタンによってライセンス情報をメディアからコンピュータに書き込みます。これでライセンスの転送が完了します。転送操作後のメディアにはライセンス情報は残りません。

ソフトウェアキーの転送操作は両方のコンピュータへのアクセスを必要とするため煩雑であり実際的な手段とは言えません。もし、頻繁に転送を繰り返したりまたは、コンピュータが異なった地理的場所にある場合は、安全性・信頼性の高いハードウェアキーのご利用をお勧めします。

Hardware Key Method(ハードウェアキー):

この方式は「ドングル」として知られている物理的なハードウェアデバイスを使用します。ハードウェアキーはソフトウェア(キーコード)で提供され、ハードウェアデバイスに書き込まれます。

- ・ ハードウェアキーは USB ポート仕様です
- ・ プログラムがキーを検索する前に、ハードウェアキー用デバイスドライバをインストールしなければなりません
- ・ デバイスドライバのインストールは管理者権限のあるアカウントでログインして行います。インストール後は一般のユーザーアカウントを利用可能です

Adding additional options to the Hardware Key(オプションの追加):

ハードウェアキーにオプションをいつでも追加することができます。弊社あるいは販売代理店に連絡してオプションを購入すると、「Field Activation Key」をご案内します。<License><Status and Authorization>メニューでダイアログを起動し、「Authorize」ボタンをクリックして下さい。そして、ダイアログに「Activation Key」を入力します。これで購入したオプションにライセンスが発行されます。

Network Hardware Key Method(ネットワークハードウェアキー):

この方式は USB ハードウェアデバイスを使用しますが、ネットワークの上で複数のライセンスを管理する能力を含んでいます。「Network Hardware Key」は複数のユーザが同時にソフトウェアを動かさず、一つの「Network Hardware Key」が必要です。キーは限定したユーザに提供され、そのユーザがプログラムを終了するまで他のユーザはプログラムを使用できません。SpectraPLUS を実行しているマシンがダウンするか、またはネットワークから取り外された場合は、ライセンスを開放する機会がありません。しかし、コミュニケーションがおよそ 5 分間以上失われると、ライセンスサーバーは自動的にライセンスをリリースします。

「Network Hardware Key」は LAN 上のどのマシンにもインストールできますが、「License Server」デバイスドライバのインストールを必要とします。このライセンスサーバーコンポーネントはネットワークキー付きで提供され、「Network Hardware Key」を持つマシンにインストールする必要があります。SpectraPLUS を実行させる他のマシンには通常のハードウェアキーデバイスドライバをインストールさせる必要があります。

- ・ ローカルマシンそれぞれに、SpectraPLUS をインストールしなければなりません
- ・ ローカルマシンそれぞれに、標準のハードウェアキードライバをインストールしなければなりません
- ・ ネットワークのマシンの一台は、Network Hardware Key をインストールしなければなりません。そしてライセンスサーバーに指定されます
- ・ ライセンスサーバーは、純粋な「サーバ」である必要はありません。単にピアツーピアネットワークにおけるコンピュータの一つでも支障ありません。「Network Hardware Key」と一緒に「ライセンスサーバー」ハードウェアキーデバイスドライバをインストールしなければなりません

追加ネットワークオプションは特定のサーバとプロトコルを指定することが可能ですが、多くの場合「Automatically find the license server on the network」とラベルされたオプションを使うべきです。

管理者が実行中のプログラム数をチェックするため、監視ツールが提供されます。

10章 Automation(自動化)

COM API

アナライザーは Visual Basic、C++ や Excel のような外部プログラムでコントロールし、データを処理することができます。サンプルプログラムは Web サイト上で提供します。

COM API - Initialization

Note: この機能にアクセスするには「Automation Tools」オプション(Opt/10)の搭載が必須です。「Accessed Denied」が表示される場合は当該オプションを追加ご購入下さい。

Visual Basic (Microsoft Visual Studio 2008)

- プロジェクトへのオートメーションインターフェースに参照を加えます
 - プロジェクトを選択 | 参照を追加...
 - 参照ダイアログで COM タブを選択
 - SpectraPLUS-DT Automation Typelib を選択
 - OK をクリック
- クラスのインスタンスを作成します

```
Dim SP As SpectraPLUS_DTAutomationLib.ISpectraPLUS_DT
SP = New SpectraPLUS_DTAutomationLib.SpectraPLUS_DTClass
```
- メソッドを呼び出します

```
SP.Run()
```

Return Values:

メソッドが HRESULTs を返すので、もしエラーが発生すれば Visual Basic は戻り値を例外にコンバートします。そのようにコードで例外を取り扱わなければなりません。

```
On Error GoTo Err_Trap
```

```
... call methods, etc.
```

```
Err_Trap:
```

```
Dim sMsg As String
sMsg = "The error returned: "
sMsg = sMsg & Err.Description
MsgBox sMsg, vbCritical, _
    "Error" & Str$(Err.Number)
Err.Clear
Resume Next
```

Accessing Arrays :

GetSpectrum()で SAFEARRAYS をどう使用するかに関する例を以下に述べます。freq と data の両方は単精度浮動小数点配列です。

```
Dim freq() As Single  
Dim data() As Single
```

```
SP.GetSpectrum(0, freq, data)
```

```
Dim n As Variant
```

```
For Each n In freq  
    Value = n  
Next n
```

```
For Each q In data  
    Value = q  
Next q
```

Visual C++ (Visual Studio 2008)

1. Type Library (TLB)をインポートします。
2. その Type のスマートポインタを作成します。
3. メソッドを呼び出します。

```
#import "C:\SpectraPLUS_DT\bin\SPDTAxnlfac.tlb" no_namespace
```

```
ColInitialize(NULL);
```

```
ISpectraPLUS_DTPtr ISP(__uuidof(SpectraPLUS_DTClass));
```

```
if ( ISP )  
{  
    ISP->Run();  
}
```

... execute remainder of program ...

```
ISP = 0;
```

戻り値は HRESULTS です。スマートポインタの Visual C++実装は戻り値を妨害し、エラーを除外します。このようにすべてのコードは try/catch ブロックに注意しなければなりません。除外されるタイプは `_com_error` です。

```
try  
{  
    ISP->Run();
```

```
}  
catch ( _com_error &error )  
{  
    MessageBox( error.ErrorMessage() );  
}
```

Microsoft Excel

1. Excel のデベロッパータブを有効にします

Excel 2007:

- 1) Office ボタンをクリック
- 2) オプションをクリック
- 3) Popular をクリック
- 4) "Show Developer tab in the Ribbon" ボックスをチェック

Excel 2010:

- 1) ファイルタブをクリック
- 2) オプション・コマンドをクリック
- 3) カスタマイズリボンカテゴリーをクリック
- 4) メインタブリストの"Developer"ボックスをチェック

2. オートメーションインターフェイスに参照を加えます

- 1) デベロッパータブをクリック
- 2) ビューコードをクリック
- 3) ツール| 参照を選択
- 4) スクロールダウンし"SpectraPLUS-DT Automation Typelib"をチェック
- 5) OK をクリック

3. クラスのインスタンスを作成します

```
Dim SP As SpectraPLUS_DTClass  
Set SP = CreateObject("SpectraPLUS_DT.Axn")
```

4. メソッドを呼び出します

```
SP.Run
```

以下はスペクトラルデータ列の取得方法とスプレッドシートでの利用例です。

```
Dim channel As Integer  
Dim startHz As Single  
Dim stopHz As Single  
  
' 周波数範囲とチャンネルを指定  
channel = 0  
startHz = 500#  
stopHz = 2000#
```

' スペクトラルデータ列変数の定義

Dim freq() As Single

Dim data() As Single

' アナライザーからデータの読み込み

SP.GetSpectrumInSpan channel, startHz, stopHz, freq, data

Dim n As Variant

Dim q As Variant

Dim col As Integer

Dim row As Integer

' セルにデータをロード

col = 1

row = 1

For Each n In freq

Sheet1.Cells(row, col).value = n

row = row + 1

Next n

col = 2

row = 1

For Each q In data

Sheet1.Cells(row, col).value = q

row = row + 1

Next q

COM API - Control Functions

以下の各機能はアナライザーをコントロールするために使われます。

Analyzer Operations (アナライザー操作)

Run()

アナライザーをランするか、レコーディングを開始します

Parameters: none

Return: S_OK = success, E_FAIL = unsuccessful

Stop()

アナライザーもしくはレコーディングを停止します

Parameters: none

Return: S_OK = success, E_FAIL = unsuccessful

Record()

レコーディングを開始します

Parameters: none

Return: S_OK = success, E_FAIL = unsuccessful

RecordPause()

レコーディングをポーズします。アナライザーは動作を継続しますが、データはWAVファイルに保存され
ます Parameters: none

Return: S_OK = success, E_FAIL = unsuccessful

RecordResume()

ポーズしたレコーディングを再開します

Parameters: none

Return: S_OK = success, E_FAIL = unsuccessful

FileStep(int numFFTSteps)

現在のファイル位置で指定した回数のFFTを実行します(Post-Processモードのみ)

Parameters: numFFTSteps: 1 or more

Return: S_OK = success, E_FAIL = unsuccessful

Analyzer Configuration (アナライザー構成)

SetMode(long mode)

モードをセットします

Parameters: mode: 0 = Real-Time, 1 = Recorder, 2 = Post-Process

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

ChannelEnable(long channel)

有効チャンネルを指定します

Parameters: channel = 0 - 15

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

ChannelDisable(long channel)

無効にするチャンネルを指定します

Parameters: channel = 0 - 15

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

RecordEnable(long channel)

レコーディングするチャンネルを指定します(前述 ChannelEnable()で有効にしておかなければなりません)

Parameters: channel = 0 - 15

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

RecordDisable(long channel)

指定したチャンネルのレコーディングを無効にします

Parameters: channel = 0 - 15

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

LoadConfig(BSTR cfg_filename)

指定した定義ファイル(.cfg)をロードします

Parameters: cfg_filename = filename of the configuration file. Full path or just filename if file located in /cfg folder

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

SaveConfig(BSTR cfg_filename)

現在の設定値を指定した定義ファイル(.cfg)に保存します

Parameters: cfg_filename = filename of the configuration file.

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

SetAverageSize(long channel, long size)

指定したチャンネルにアベレージサイズをセットします

Parameters: channel = 0 - 15; size = average size 1 - 10000 (free run mode), 0 = Off, 1 = Fast, 2 = Medium, 3 = Slow, 4 = Forever (SLM mode)

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

ResetAverage(long channel)

指定したチャンネルのアベレージ値をリセットします

Parameters: channel = 0 - 15

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

SetPeakHold(long channel_ext, long peak_hold_type)

指定したチャンネルのピークホールドタイプをセットします

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels; peak_hold_type: 0 = Off, 1 = Fast, 2 = Medium, 3 = Slow, 4 = Forever

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

ResetPeakHold(long channel_ext)

指定したチャンネルのピークホールドをリセットします

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels
Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

SetOverlapPercentage(long channel, long overlap_percent)

指定したチャンネルの overlap percentage 値をセットします

Parameters: channel = 0 - 15; overlap_percent = 1 - 99

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

SetCompensationFile(long channel, BSTR comp_filename)

指定したチャンネルの transducer compensation ファイルをセットします

Parameters: channel = 0 - 15; comp_filename = filename of the transducer file.

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

SetMarkerFreq(long channel_ext, long marker, float freqHz)

指定したチャンネルのマーカに周波数をセットします

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels; marker = 1 - 8,
freqHz = frequency in Hz

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

ResetOverloadCount()

すべてのチャンネルのオーバーロード検知をリセットします

Parameters: none

Return: S_OK = success, E_FAIL = unsuccessful

DataLoggingEnable()

data logging を有効にします

Parameters: none

Return: S_OK = success, E_FAIL = unsuccessful

DataLoggingDisable()

data logging を無効にします

Parameters: none

Return: S_OK = success, E_FAIL = unsuccessful

LoadCalibrationFile(long channel, BSTR cal_filename)

指定したチャンネルのキャリブレーションファイル(.cal)をロードします

Parameters: channel = 0 - 15; cal_filename = filename of the calibration file.

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

SaveCalibrationFile(long channel, BSTR cal_filename)

指定したチャンネルのキャリブレーションファイル(.cal)を保存します

Parameters: channel = 0 - 15; cal_filename = filename of the calibration file.

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Calibration Commands

LoadCalibrationFile(long channel, BSTR filename)

指定されたチャンネルのキャリブレーションファイルをロードします Parameters: channel: 0 = Left, 1 = Right;
filename = the filename of the calibration file

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Note: 短いファイル名を使用すると、<File><Set Paths>ダイアログボックスにリストされているキャリブレーションパスに保存されます

CalibrateToExternalSource(long channel, float source_level)

指定されたチャンネルで校正測定を実行します

Parameters: channel: 0 = Left, 1 = Right; source_level = value to calibrate to (SPL example: 94.0).

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Notes: 外部ソースにキャリブレーションするには、キャリブレーションオプションを使用している必要があります。以下の条件が満たされるまで測定を続けます:

- 1) 測定された信号レベルは、 ± 0.1 ミリボルト以内で安定。これは、一般的なマイク感度で約 ± 0.03 dB です
- 2) 測定された信号レベルは、元の値の 12%以内。これは約 ± 1 dB です

これらの条件の両方が 60 秒以内に満たされない場合、測定は停止し、E_FAIL エラーが発生します。これらの条件により、セtringが完了し、マイクキャリブレータがオンになり、正しいマイクとチャンネルが使用されることが保証されます

CalibrateToZeroDBr(long channel, float frequency)

選択した周波数でキャリブレーションを 0.0dBr に設定します

Parameters: channel: 0 = Left, 1 = Right; frequency = frequency at which the resulting calibration will become 0.0 dBr

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

SaveCalibrationFile(long channel, BSTR cal_filename)

指定したチャンネルのキャリブレーションファイルを保存します

Parameters: channel: 0 = Left, 1 = Right; cal_filename = filename of the calibration file.

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Note: 短いファイル名を使用すると、<File><Set Paths> ダイアログボックスにリストされている Cal パスに保存されます

Plot and Window Commands

OpenPlot(long channel, long plot_type)

指定されたチャンネルのプロットを開きます

Parameters: channel: 0 = Left, 1 = Right.

plot_type: 1 = Time Series, 2 = Spectrum, 3 = Phase, 4 = Spectrogram, 5 = 3-D Surface

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

ClosePlot(long channel, long plot_type)

指定されたチャンネルのプロットを閉じます

Parameters: channel: 0 = Left, 1 = Right.

plot_type: 1 = Time Series, 2 = Spectrum, 3 = Phase, 4 = Spectrogram, 5 = 3-D Surface

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

MinimizeApplication()

アプリケーションウィンドウ全体をアイコンに最小化する

Return: S_OK = success, E_FAIL = unsuccessful

MaximizeApplication()

画面上のアプリケーションウィンドウ全体を最大化する

Return: S_OK = success, E_FAIL = unsuccessful

RestoreApplication()

アプリケーションを以前のサイズに復元します

Return: S_OK = success, E_FAIL = unsuccessful

HideApplication()

アプリケーションウィンドウ全体とアイコンを非表示にします

Return: S_OK = success, E_FAIL = unsuccessful

ShowApplication()

アプリケーションウィンドウ全体とアイコンを表示します (HideApplication() から復元)

Return: S_OK = success, E_FAIL = unsuccessful

ClearSpectrum(long channel_ext)

指定されたチャンネルのスペクトルデータをクリアします

Parameters: channel_ext: 0 = Left, 1 = Right, 2 = Composite, 3 = Coherence.

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Note: コンポジットチャンネル=デュアルチャンネル操作(L&R 平均、伝達関数、またはクロススペクトル)

ClearPeakHoldSpectrum(long channel_ext)

指定されたチャンネルのピークホールドスペクトルデータをクリアします

Parameters: channel_ext: 0 = Left, 1 = Right, 2 = Composite, 3 = Coherence.

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Note: コンポジットチャンネル=デュアルチャンネル操作(L&R 平均、伝達関数、またはクロススペクトル)

SetSpectrumPlotFrequencySpan(long channel_ext, float startHz, float stopHz)

指定されたチャンネルのスペクトルプロットに表示される周波数スパンを設定します

Parameters: channel_ext: 0 = Left, 1 = Right, 2 = Composite, 3 = Coherence.

startHz = start frequency, stopHz = stop frequency

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Note: コンポジットチャンネル=デュアルチャンネル操作(L&R 平均、伝達関数、またはクロススペクトル)

SetSpectrumPlotTop(long channel_ext, float plot_top)

指定されたチャンネルのスペクトルプロットのアンプリチュードプロットトップ値を設定します

Parameters: channel_ext: 0 = Left, 1 = Right, 2 = Composite, 3 = Coherence.

plot_top = plot top value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Note: コンポジットチャンネル=デュアルチャンネル操作(L&R 平均、伝達関数、またはクロススペクトル)

SetSpectrumPlotRange(long channel_ext, float plot_range)

指定されたチャンネルのスペクトルプロットのアンプリチュードプロットレンジ値を設定します

Parameters: channel_ext: 0 = Left, 1 = Right, 2 = Composite, 3 = Coherence.

plot_range = plot range value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Note: コンポジットチャンネル=デュアルチャンネル操作(L&R 平均、伝達関数、またはクロススペクトル)

SetSpectrumPlotGraphType(long channel_ext, long graph_type)

指定したチャンネルのスペクトルプロットグラフタイプを設定します

Parameters: channel_ext: 0 = Left, 1 = Right, 2 = Composite, 3 = Coherence.

graph_type: 0 = Line, 1 = Bars, 2 = Stepped Bars

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Note: コンポジットチャンネル=デュアルチャンネル操作(L&R 平均、伝達関数、またはクロススペクトル)

SetTimeSeriesPlotTimeSpan(float start_time, float stop_time)

TimeSeries プロットに表示されるスパンを設定します

Parameters: start_time, stop_time = time span to display

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

WritePlotToBitmapFile(long channel_ext, long plot_type, BSTR bmp_filename)

プロット画像をビットマップファイルに保存します

Parameters: channel_ext: 0 = Left, 1 = Right, 2

plot_type: 1 = Time Series, 2 = Spectrum, 3 = Phase, 4 = Spectrogram, 5 = 3-D Surface
bmp_filename = full filename and path to store bitmap
Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Overlay Control

SetOverlay(long channel_ext, long overlay_index, int create_from_peak_hold, BSTR szLegend)

現在のスペクトルデータからオーバーレイを作成します

Parameters: channel_ext: 0 = Left, 1 = Right, 2

overlay_index = 1 - 14

create_from_peak_hold = 1 to create the overlay from the peak hold values (peak hold must be enabled)

szLegend = legend to apply to overlay

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

ShowOverlay(long channel_ext, long overlay_index)

指定されたオーバーレイを表示します

Parameters: channel_ext: 0 = Left, 1 = Right, 2

overlay_index = 1 - 14

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

HideOverlay(long channel_ext, long overlay_index)

指定されたオーバーレイを非表示にします

Parameters: channel_ext: 0 = Left, 1 = Right, 2

overlay_index = 1 - 14

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

SaveOverlay(long channel_ext, long overlay_index, BSTR ovl_filename)

オーバーレイをファイルに保存します

Parameters: channel_ext: 0 = Left, 1 = Right, 2

overlay_index = 1 - 14

ovl_filename = name of the overlay file.

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Note: 短いファイル名を使用すると、<File><Set Paths>ダイアログボックスにリストされているオーバーレイパスに保存されます。

LoadOverlay(long channel_ext, long overlay_index, BSTR ovl_filename)

ファイルからオーバーレイをロードします

Parameters: channel_ext: 0 = Left, 1 = Right, 2

overlay_index = 1 - 14

ovl_filename = name of the overlay file to load

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Note: 短いファイル名が使用されている場合、<File><Set Paths>ダイアログボックスに表示されているオーバーレイパスからロードされます

File Management (ファイル操作)

FileOpen(long channel, BSTR wav_filename)

指定したチャンネルで WAV ファイルを開きます

Parameters: channel = 0 - 15; wav_filename = full path to wave file

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

FileSave(long channel, BSTR wav_filename)

指定したチャンネルの WAV ファイルを保存します

Parameters: channel = 0 - 15; wav_filename = full path to wave file

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

FileClose(long channel)

指定したチャンネルの WAV ファイルを閉じます

Parameters: channel = 0 - 15

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

File Positioning (ファイル位置コントロール)

FileRewind(long channel)

指定したチャンネルの WAV ファイルの頭(スタート位置)に移動します

Parameters: channel = 0 - 15

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

FileForward(long channel)

指定したチャンネルの WAV ファイルの最終端(ストップ位置)に移動します

Description: Go to the end of the wave file on the specified channel

Parameters: channel = 0 - 15

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

FileGoToTime(long channel, float seconds)

指定したチャンネルの WAV ファイルの指定時間に移動します

Parameters: channel = 0 - 15; seconds = file position in seconds

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

FileGoToSample(long channel, unsigned long sample)

指定したチャンネルのサンプル位置に移動します

Description: Go to a sample position on the specified channel

Parameters: channel = 0 - 15; sample = file position in samples

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

File Post-Process (ファイルの後処理)

FilePlaySegmentTime(long channel, float start_seconds, float stop_seconds)

指定したチャンネルの指定したタイムセグメントを再生します

Parameters: channel = 0 - 15; start_seconds = segment start time; stop_seconds = segment stop time

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

FilePlaySegmentSample(long channel, unsigned long start_sample, unsigned long stop_sample)

指定したチャンネルの指定したサンプルを再生します

Parameters: channel = 0 - 15; start_sample = segment start sample position; stop_sample = segment stop sample position

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

FileFilterSegmentTime(long channel, float start_seconds, float stop_seconds)

指定したチャンネルの指定したタイムセグメントをフィルタ処理します

Parameters: channel = 0 - 15; start_seconds = segment start time; stop_seconds = segment stop time

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

FileFilterSegmentSample(long channel, unsigned long start_sample, unsigned long stop_sample)

指定したチャンネルの指定したサンプルをフィルタ処理します

Description: Filter the specified time segment for the selected channel

Parameters: channel = 0 - 15; start_sample = segment start sample position; stop_sample = segment stop sample position

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

FileProcessSegmentTime(long channel, float start_seconds, float stop_seconds)

指定したチャンネルの指定したタイムセグメントにスペクトラデータを処理します

Parameters: channel = 0 - 15; start_seconds = segment start time; stop_seconds = segment stop time

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

FileProcessSegmentSample(long channel, unsigned long start_sample, unsigned long stop_sample)

指定したチャンネルの指定したサンプルにスペクトラデータを処理します

Description: Process the spectral data for the specified time segment for the selected channel

Parameters: channel = 0 - 15; start_sample = segment start sample position; stop_sample = segment stop sample position

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Signal Generation (シグナルジェネレータ)

RunGenerator()

シグナルジェネレータをスタートします

Parameters: none

Return: S_OK = success, E_FAIL = unsuccessful

StopGenerator()

シグナルジェネレータをストップします

Parameters: none

Return: S_OK = success, E_FAIL = unsuccessful

SetGeneratorSignalType(long channel_sig, long signal_type)

指定したチャンネルに信号のフォーム/タイプをセットします

Parameters: channel = 0 - 1; signal_type - as follows:

- 0 = White Noise
- 1 = Pink Noise
- 2 = Noise Burst
- 3 = 1 kHz Tone
- 4 = Multiple Tones
- 5 = Tone Burst
- 6 = IMD Tones
- 7 = Freq Sweep
- 8 = Level Sweep
- 9 = Sawtooth
- 10 = Triangular
- 11 = Pulse
- 12 = Squarewave
- 13 = User Defined wave file
- 14 = DTMF
- 15 = Digital Zero
- 16 = Frequency Step

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

SetGeneratorLevel(long channel_sig, float volts)

指定したチャンネルの信号レベルをセットします

Parameters: channel = 0 - 1; volts = rms voltage level

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Note: このコマンドを使用すると、出力単位がボルトに変わります

SetGeneratorFreq(long channel_sig, float freqHz)

シグナルタイプをマルチトーンに設定し、指定された周波数で1つのトーンのみを有効にします。マルチトーンのレベルは 0 dBFS に設定されます

(出力レベルを設定するには SetGeneratorLevel() を使用します)

Parameters: channel_sig = 0 - 1; freqHz = frequency in Hz

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Note: このコマンドを使用すると、信号の種類が Multiple Tones に変更されます

LoadGeneratorMultiToneFile(long channel_sig, BSTR toneFilename)

ディスクから.tones ファイルをロードします-これにより、トーンのセットを変更できます

Parameters: channel_sig = 0 - 1; toneFilename = name of .tones file

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Note: このコマンドを使用すると、信号の種類が Multiple Tones に変更されます

Digital Input and Output Lines

SpectraDAQ-200 ハードウェアモジュールには、リレーなどの外部機器を制御および読み取るためのデジタル入力および出力ラインがあります。これらの関数を使用すると、これらの行の状態を読み書きできます。3つの入力ラインと3つの出力ラインがあります。

GetDigitalInputLineState(DWORD *state)

デジタル入力ラインの現在の状態を取得します

Parameters: state = pointer to value to receive the current state of the digital input lines

Return: S_OK = success, E_FAIL = unsuccessful

SetDigitalOutputLineState(DWORD state)

デジタル出力ラインの現在の状態を設定します

Parameters: state = value containing the new state to set the output lines

Return: S_OK = success, E_FAIL = unsuccessful

COM API - Read Data Functions

以下はアナライザーからデータ及びステータスを読む機能です。

General Status (状態)

IsRunning (int *runStatus)

アンライザーをリアルタイムモードもしくはレコーディングモードでランします

Parameters: runStatus = pointer to receive the analyzer run status (0 = stopped, 1 = running or recording)

Return: S_OK = success, E_FAIL

GetCurrentFileTime(long channel, float *seconds)

指定したチャンネルのファイルの現在位置を求めます

Parameters: channel = 0 - 15; seconds = pointer to variable to receive the time value in seconds

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetTotalFileTime(long channel, float *seconds)

指定したチャンネルのファイルのファイル長(時間)を求めます

Parameters: channel = 0 - 15; seconds = pointer to variable to receive the time value in seconds

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetCurrentFileSample(long channel, unsigned long *sample)

指定したチャンネルのファイルのサンプルポジションを求めます

Parameters: channel = 0 - 15; sample = pointer to variable to receive the sample position

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetTotalFileSamples(long channel, unsigned long *samples)

指定したチャンネルのファイルの総サンプル数を求めます

Parameters: channel = 0 - 15; samples = pointer to variable to receive the total samples value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetFFTCOUNT(long channel, long *fftcount)

ランカレコードが最後にスタートしてから実行された FFT の回数を求めます

Parameters: channel = 0 - 15; fftcount - pointer to variable to receive the the fft count

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetOverloadCount(long channel, long *overloadcount)

ランカレコードが最後にスタートしてから検出されたオーバーロード(クリップ)の回数を求めます

Parameters: channel = 0 - 15; overloadcount - pointer to variable to receive the the overload count

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Peak Search Operations (ピーク)

GetPeakFrequency(long channel_ext, float *freq)

指定したチャンネルの全スパン中のピーク周波数を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels; freq = pointer to variable to receive the peak frequency

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetPeakFrequencyNth(long channel_ext, long peak_number, float *freq)

指定したチャンネルの全スパン中の N 次ピーク周波数を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels; peak_number = 0 - 10; freq = pointer to variable to receive the peak frequency

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetPeakFrequencyNthSpan(long channel_ext, long peak_number, float startHz, float stopHz, float *freq)

指定したチャンネルの指定スパン中の N 次ピーク周波数を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels; peak_number = 0 - 10; startHz = span start frequency; stopHz = span stop frequency; freq = pointer to variable to receive the peak frequency

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetPeakAmplitude(long channel_ext, float *amplitude)

指定したチャンネルの全スパン中のピークアンプリチュードを求めます

Description: Get the peak amplitude in the total span for the specified channel

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels; amplitude = pointer to variable to receive the peak amplitude

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetPeakAmplitudeNth(long channel_ext, long peak_number, float *amplitude)

指定したチャンネルの全スパン中の N 次ピークアンプリチュードを求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels; peak amplitude

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetPeakAmplitudeNthSpan(long channel_ext, long peak_number, float startHz, float stopHz, float *amplitude)

指定したチャンネルの指定スパン中の N 次ピークアンプリチュードを求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels; peak_number = 0 - 10; startHz = span start frequency; stopHz = span stop frequency; amplitude = pointer to variable to receive the peak amplitude

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetPeakHoldFrequency(long channel_ext, float *freq)

指定されたチャンネルのピークホールドスペクトルの最大周波数を取得します

Parameters: channel_ext = 0 - 15 for main channels, 17 - 32 for composite channels;

freq = pointer to variable to receive the peak hold frequency

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetPeakHoldAmplitude(long channel_ext, float *amplitude)

指定されたチャンネルのピークホールドスペクトルの最大 amplitude を取得します

Parameters: channel_ext = 0 - 15 for main channels, 17 - 32 for composite channels;

amplitude = pointer to variable to receive the peak hold amplitude

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Total Power Readings (トータルパワー)

GetTotalPower(long channel_ext, long weighting, float *power)

Description: Get the total power in the total span for the specified channel and weighting

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;

weighting: 0 = Flat, 1 = A, 2 = B, 3 = C

power = pointer to variable to receive the total power value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetTotalPowerSpan(long channel_ext, long weighting, float startHz, float stopHz, float *power)

指定したチャンネルの指定周波数スパン中のトータルパワーを求めます。ウェイトの指定も可能です

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;

weighting: 0 = Flat, 1 = A, 2 = B, 3 = C

startHz = span start frequency; stopHz = span stop frequency;

power = pointer to variable to receive the total power

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetPeakHoldTotalPower(long channel_ext, long weighting, float *power)

指定されたチャンネルと重み付けのピークホールドスペクトルの total power を取得します

Parameters: channel_ext = 0 - 16 for main channels, 17 - 32 for composite channels;

weighting: 0 = Flat, 1 = A, 2 = B, 3 = C

power = pointer to variable to receive the peak hold total power value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Distortion Readings (ひずみ)

GetTHD(long channel_ext, float *thd)

指定したチャンネルの全スパンの THD を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;

thd = pointer to variable to receive the THD value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetTHDSpan(long channel_ext, float startHz, float stopHz, float *thd)

指定したチャンネルと周波数スパンの THD を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;

startHz = span start frequency; stopHz = span stop frequency;

thd = pointer to variable to receive the THD value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetTHD_N(long channel_ext, float *thd_n)

指定したチャンネルの全スパンの THD+N を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;

thd_n = pointer to variable to receive the THD+N value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetTHD_NSpan(long channel_ext, float startHz, float stopHz, float *thd_n)

指定したチャンネルと周波数スパンの THD+N を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;
startHz = span start frequency; stopHz = span stop frequency;
thd_n = pointer to variable to receive the THD+N value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetIMD(long channel_ext, float *imd)

指定したチャンネルの全スパンの IMD を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;
imd = pointer to variable to receive the IMD value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetSNR(long channel_ext, float *snr)

指定したチャンネルの全スパンの SNR を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;
snr = pointer to variable to receive the SNR value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetSNRSpan(long channel_ext, float startHz, float stopHz, float *snr)

指定したチャンネルと周波数スパンの SNR を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;
startHz = span start frequency; stopHz = span stop frequency;
snr = pointer to variable to receive the SNR value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetSINAD(long channel_ext, float *sinad)

指定したチャンネルの全スパンの SINAD を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;
sinad = pointer to variable to receive the SINAD value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetSINADSpan(long channel_ext, float startHz, float stopHz, float *sinad)

指定したチャンネルと周波数スパンの SINAD を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;
startHz = span start frequency; stopHz = span stop frequency;
sinad = pointer to variable to receive the SINAD value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Marker Readings(マーカ)

GetMarkerFrequency(long channel_ext, long marker, float *freq)

指定したマーカの周波数を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;
marker = 1 - 8;

freq = pointer to variable to receive the frequency value
Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetMarkerAmplitude(long channel_ext, long marker, float *amplitude)

指定したマーカのアンプリチュードを求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;
marker = 1 - 8;
amplitude = pointer to variable to receive the amplitude value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Conversion Operations(変換操作)

ConvertTimeToSample(long channel, float time_seconds, unsigned long *sample)

指定したチャンネルの秒をサンプルポジションに変換します

Parameters: channel_ext = 0 - 15; time_seconds = time value to convert
sample = pointer to variable to receive the sample position

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

ConvertSampleToTime(long channel, unsigned long sample, float *time_seconds)

指定したチャンネルのサンプルポジションを秒に変換します

Parameters: channel_ext = 0 - 15; sample = sample position to convert
time_seconds = pointer to variable to receive the time in seconds

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Time Series Data(タイムシリーズ)

GetTimeSeriesValueFromTime(long channel, float time_seconds, float *value)

指定した時間のタイムシリーズアンプリチュードを求めます

Parameters: channel_ext = 0 - 15; time_seconds = file position time in seconds
value = pointer to variable to receive the time series amplitude value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetTimeSeriesValueFromSample(long channel, unsigned long sample, float *value)

指定したサンプルポジションのタイムシリーズ値を求めます

Parameters: channel_ext = 0 - 15; sample = file position in samples
value = pointer to variable to receive the time series amplitude value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetMaxTimeSeriesValueInSegment(long channel, unsigned long startSample, unsigned long stopSample, float *value)

指定した時間レンジの最大タイムシリーズ値を求めます

Parameters: channel_ext = 0 - 15;
startSample = time segment start position in samples;
stopSample = time segment stop position in samples;
value = pointer to variable to receive the maximum time series amplitude value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetMinTimeSeriesValueInSegment(long channel, unsigned long startSample, unsigned long stopSample, float *value)

指定した時間レンジの最小タイムシリーズ値を求めます

Parameters: channel_ext = 0 - 15;

startSample = time segment start position in samples;

stopSample = time segment stop position in samples;

value = pointer to variable to receive the minimum time series amplitude value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetTimeSeriesFromFile(long channel, SAFEARRAY __RPC_FAR * __RPC_FAR *psaTimes, SAFEARRAY __RPC_FAR * __RPC_FAR *psaData)

指定したチャンネルのタイムシリーズデータ列を求めます

Parameters: channel_ext = 0 - 15;

psaTimes = pointer to array of time values

psaData = pointer to array of amplitude values

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetTimeSeriesFromFileSegmentSample(long channel, unsigned long startSample, unsigned long stopSample, SAFEARRAY __RPC_FAR * __RPC_FAR *psaTimes, SAFEARRAY __RPC_FAR * __RPC_FAR *psaData)

指定したチャンネルとサンプルレンジのタイムシリーズデータ列を求めます

Parameters: channel_ext = 0 - 15;

startSample = start sample of time series segment;

stopSample = stop sample of time series segment;

psTimes = pointer to array of time values

psaData = pointer to array of amplitude values

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetTimeSeriesFromFileSegmentTime(long channel, float startSecs, float stopSecs, SAFEARRAY __RPC_FAR * __RPC_FAR *psaTimes, SAFEARRAY __RPC_FAR * __RPC_FAR *psaData)

指定したチャンネルと時間レンジのタイムシリーズデータ列を求めます

Parameters: channel_ext = 0 - 15;

startSecs = start time of time series segment;

stopSecs = stop time of time series segment;

psaTimes = pointer to array of time values

psaData = pointer to array of amplitude values

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetMaxValueInTriggerSeries(long channel, float *value)

トリガサンプルのブロック内の最大 time series 値を取得します。トリガブロックの長さは、現在の FFT サイズと同じです

Parameters: channel: 0 = Left, 1 = Right

value = pointer to variable to receive the maximum amplitude value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetMinValueInTriggerSeries(long channel, float *value)

トリガサンプルのブロック内の最大 time series 値を取得します。トリガブロックの長さは、現在の FFT サイズと同じです

Parameters: channel: 0 = Left, 1 = Right

value = pointer to variable to receive the minimum amplitude value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetTriggerTimeSeries(long channel, SAFEARRAY __RPC_FAR * __RPC_FAR *psaTimes, SAFEARRAY __RPC_FAR * __RPC_FAR *psaData)

トリガサンプルのブロックを含む time series データの配列を取得します。トリガブロックの長さは、現在の FFT サイズと同じです

Parameters: channel: 0 = Left, 1 = Right

psaTimes = pointer to array of time values

psaData = pointer to array of amplitude values

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Spectral Data(スペクトラル)

GetSpectrumMinFrequency(long channel_ext, float *freq)

指定したチャンネルのスペクトラルデータの最小周波数を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;

freq = pointer to variable to receive the frequency value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetSpectrumMaxFrequency(long channel_ext, float *freq)

指定したチャンネルのスペクトラルデータの最大周波数を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;

freq = pointer to variable to receive the frequency value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetSpectrumValue(long channel_ext, float freqHz, float *value)

指定したチャンネルと周波数の、スペクトラルデータのアンプリチュードを求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;

freqHz = frequency to read the amplitude

value = pointer to variable to receive the amplitude value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetSpectrum(long channel_ext,SAFEARRAY __RPC_FAR * __RPC_FAR *psaFrequencies, SAFEARRAY __RPC_FAR * __RPC_FAR *psaData)

指定したチャンネルのスペクトラルデータ列を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;

psaFrequencies = pointer to array of frequency values

psaData = pointer to array of amplitude values

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetSpectrumInSpan(long channel_ext, float startHz, float stopHz, SAFEARRAY __RPC_FAR * __RPC_FAR *psaFrequencies, SAFEARRAY __RPC_FAR * __RPC_FAR *psaData)

指定したチャンネルと周波数レンジのスペクトラルデータ列を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;

startHz = start frequency of selected span;

stopHz = stop frequency of selected span;

psaFrequencies = pointer to array of frequency values

psaData = pointer to array of amplitude values

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetPeakHoldSpectrumValue(long channel_ext, float freqHz, float *value)

指定したチャンネルと周波数の、スペクトラルデータのピークアンプリチュードを求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;

freqHz = frequency to read the amplitude

value = pointer to variable to receive the amplitude value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetPeakHoldSpectrum(long channel_ext,SAFEARRAY __RPC_FAR * __RPC_FAR *psaFrequencies, SAFEARRAY __RPC_FAR * __RPC_FAR *psaData)

指定したチャンネルのピークホールドスペクトラルデータ列を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;

psaFrequencies = pointer to array of frequency values

psaData = pointer to array of amplitude values

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetPeakHoldSpectrumInSpan(long channel_ext, float startHz, float stopHz, SAFEARRAY __RPC_FAR * __RPC_FAR *psaFrequencies, SAFEARRAY __RPC_FAR * __RPC_FAR *psaData)

指定したチャンネルと周波数レンジのピークホールドスペクトラルデータ列を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;

startHz = start frequency of selected span;

stopHz = stop frequency of selected span;

psaFrequencies = pointer to array of frequency values

psaData = pointer to array of amplitude values

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetComplexSpectrum(long channel_ext,SAFEARRAY __RPC_FAR * __RPC_FAR *psaFrequencies, SAFEARRAY __RPC_FAR * __RPC_FAR *psaDataR, SAFEARRAY __RPC_FAR * __RPC_FAR *psaDataI)

指定されたチャンネルの complex スペクトルデータの配列を取得します

Parameters: channel_ext: 0 = Left, 1 = Right, 2 = Composite, 3 = Coherence.

psaFrequencies = pointer to array of frequency values

psaDataR = pointer to array of real component of complex values

psaDataI = pointer to array of imaginary component of complex values

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Note: コンポジットチャンネル=デュアルチャンネル動作(L&R 平均、トランスファーまたはクロススペクトル)

GetComplexSpectrumInSpan(long channel_ext, float startHz, float stopHz, SAFEARRAY __RPC_FAR * __RPC_FAR *psaFrequencies, SAFEARRAY __RPC_FAR * __RPC_FAR *psaDataR, SAFEARRAY __RPC_FAR * __RPC_FAR *psaDataI)

指定されたチャンネルの complex スペクトルデータの配列を取得します

Parameters: channel_ext: 0 = Left, 1 = Right, 2 = Composite, 3 = Coherence.

startHz = start frequency of selected span;

stopHz = stop frequency of selected span;

psaFrequencies = pointer to array of frequency values

psaDataR = pointer to array of real component of complex values

psaDataI = pointer to array of imaginary component of complex values

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Note: コンポジットチャンネル=デュアルチャンネル動作(L&R 平均、トランスファーまたはクロススペクトル)

Phase Data (位相)

GetPhaseValue(long channel_ext, float freqHz, float *value)

指定したチャンネルと周波数の位相を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;

freqHz = frequency to read the phase

value = pointer to variable to receive the phase value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetPhase(long channel_ext, SAFEARRAY __RPC_FAR * __RPC_FAR *psaFrequencies, SAFEARRAY __RPC_FAR * __RPC_FAR *psaData)

指定したチャンネルの位相データ列を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;

psaFrequencies = pointer to array of frequency values

psaData = pointer to array of phase values

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetPhaseInSpan(long channel_ext, float startHz, float stopHz, SAFEARRAY __RPC_FAR * __RPC_FAR *psaFrequencies, SAFEARRAY __RPC_FAR * __RPC_FAR *psaData)

指定したチャンネルと周波数レンジの位相データ列を求めます

Parameters: channel_ext = 0 - 15 for main channels, 16 - 31 for composite channels;

startHz = start frequency of selected span;

stopHz = stop frequency of selected span;

psaFrequencies = pointer to array of frequency values

psaData = pointer to array of phase values

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Leq Data (Leq)

GetLeqValues(long channel, float *LeqT, float *Leq, float *Lsel, float *Lpk, float *Lmax, float *Lmin, float *L10, float *L50, float *L90)

指定したチャンネルの Leq (Equivalent Noise Level)を求めます

Parameters: channel_ext = 0 - 15

LeqT = pointer to the equivalent noise level converted to the user specified "reference time period".

Leq = pointer to the equivalent noise level for the selected time segment.

Lsel = pointer to the equivalent noise level converted to a 1 second period.

Lpk = pointer to the peak level for the selected time segment. No averaging is used for this parameter.

Lmax = pointer to the maximum level for the selected data.

Lmin = pointer to the minimum level for the selected data.

L10 = pointer to the level exceeded 10% of the time.

L50 = pointer to the level exceeded 50% of the time.

L90 = pointer to the level exceeded 90% of the time.

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

Misc

GetDelayFinderValue(float *value)

Delay Finder ユーティリティの現在の値を取得します (ユーティリティは開いている必要があります)

Parameters: value = pointer to the delay value

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

GetMetadata(BSTR* metadata)

現在の wafe ファイルからメタデータテキストを取得します。この関数を実行するには、WAVE ファイルを開いていなければなりません

Parameters: metadata - up to 2000 characters max of ascii text

Return: S_OK = success, E_FAIL = unsuccessful, E_INVALIDARG = invalid parameter

11章 Performance Issues

Processing Speed (処理速度)

処理速度は多くの要素によって影響を受けます。まず「FFTサイズ」があります。速さを重視するなら小さいFFTサイズを選択して下さい。ただし、周波数分解は下がります。

ビデオの速度には他の要素が影響します。複数のビューが開かれているとビデオサブシステムに負荷がかかります。「Time Series」Plot 表示が最速です。「3-D Surface」「Spectrogram」Plot は「Time Series」や「Spectrum」Plot よりもスローです。もし、不要な Plot が開かれていれば閉じて下さい。アイコン化しても若干の改善が得られます。

Measurement Accuracy (測定確度・精度)

周波数:

周波数確度には DT-9800 モジュールのサンプリング周波数確度が直接反映します。

アンプリチュード:

DT-9800 は Volt で校正された値を提供します。使用する DT-9800 モジュールが提供する仕様を参照して下さい。

オクターブスケール:

歴史的に、スペクトラムアナライザーは各バンドにアナログフィルタを利用しました。そしてこれらのフィルタの性能を指定する規格(ANSI S1.11-1986)が設けられました。SpectraPLUS-DT はスペクトルデータを計算するのに高速フーリエ変換(FFT)を使用し、次にオクターブデータを引き出します。このアルゴリズムはアナログ機器と同じ ISO センター周波数と帯域幅を使用します。しかしながら、FFT フィルタメソッドははるかに急峻な「肩特性」を生成します。このデジタルフィルタは、伝統的なアナログフィルタの性能を満たすか、または超えています。

ダイナミックレンジ:

システムの理論ダイナミックレンジは次の通りです。

- 8 bit sampling precision=48 dB
- 16 bit sampling precision=96 dB
- 24 bit sampling precision=144 dB

実際のダイナミックレンジは電子ノイズなどの他のファクターが反映します。

Aliasing (アリアシング)

アリアシングは「サンプリングレートの 1/2」より大きな周波数を含んだ信号がサンプリングされた状態です。この場合、信号はアナライザーで解析されますがその周波数は不正確です。例えば、もしサンプリングレートが「6,000Hz」のとき「3,500Hz」の信号が入力されるとアナライザーは「2,500Hz」を表示します。

使用するDT-9800モジュールの仕様を参照して下さい。

12章 Technical Support(サポート窓口)

SONALYS 株式会社

TEL:050-3565-9626 / FAX:03-6701-7051

Email: support@spectrasoft.jp / support@sonalys.com

URL: www.spectrasoft.jp

Copyright 1993-2006,PHS

All Rights Reserved.

Legal Notices and Trademark attributions.

All trademarks are the property of their respective owners.

Pioneer Hill Software, 342 Lavender Meadows Drive Sequim WA 98382 USA

TEL:360697-3472 / FAX:360-697-7717

著作権:

本ドキュメントの著作権は Pioneer Hill Software 及び SONALYS 株式会社に帰属します。無断配布、転載はご遠慮申し上げます。

DT9800 デバイスドライバーのインストールについて

ほとんどのコンピュータにはUSBデバイスをダイレクト接続するいくつかのUSBポートが搭載されています。より多くのUSBポートを必要とする場合は拡張ハブを使用することができます。

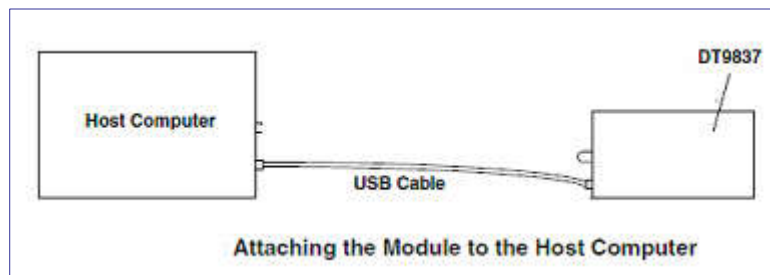
DT9800モジュールは基本的に外部電源供給は要しません。USBセルフパワーで駆動されます。

DT9800 モジュールをコンピュータに接続する前に、以下の手順に沿ってデバイスドライバをインストールします。

- 添付の「Omni CD」ディスクを CDドライブにセットします。
- 自動起動後表示されるメニューから「> Install from CD」を選択します。もし、最新のドライバーを必要とする場合は「> Install from Web」を選択して下さい。ダウンロードサイズは約 300MB です。
- インストールウィザードが開きます。その後 **NEXT** ボタンをクリックします。
- 「I accept ...」をチェックし、続いて **NEXT** ボタンをクリックします。
- インストール先既定フォルダを変更しなければ **NEXT** ボタンをクリックします。
- **INSTALL** ボタンをクリックします。
- 「InstallShield Wizard complete」が表示されたら **Finish** ボタンをクリックします。
- コンピュータを再起動します。

再起動後...

- USBケーブルの片端をモジュールのUSBポートにつなぎます。
- コンピュータのUSBポートにUSBケーブルのもう一端をつないで下さい。



- オペレーティングシステムは自動的にUSBモジュールを検出してウィザードを開始しますので画面の案内に従って進みます。
- もし必要なら、複数のDT9800モジュールをコンピュータに取り付けるためにこれらのステップを繰り返します。
- モジュールがコンピュータに接続されると、プログラムがモジュールにアクセスする際は自動的にパワーONし、LEDが青色に点灯します。そして、モジュールへのアクセスを終了するとOFFします。