



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



## VisualAnalog™ Converter Evaluation Tool Version 1.0 User Manual

### INTRODUCTION

VisualAnalog is a new way to test and characterize data converters, ADCs, and DACs alike. Whereas tools have been available in the past, they could perform a limited set of tests. Whereas these tools did provide many features, their flexibility was limited to just a few simple options. VisualAnalog provides the ability to customize the tests in a nearly limitless manner using a simple graphical user interface.

VisualAnalog interfaces seamlessly with the DAC pattern generator (DPG) for DAC evaluation and the following ADC data capture boards for ADC evaluation:

- HSC-ADC-EVALA
- HSC-ADC-EVALB
- HSC-ADC-EVALC

### PRODUCT HIGHLIGHTS

1. Quick set up of both ADC and DAC characterization
2. Easy testing of ADCs with reference DACs and DACs with reference ADCs
3. Easy configuration of custom signal flow tests for ADCs and DACs
4. Easy testing of converter models and comparison to real converter devices

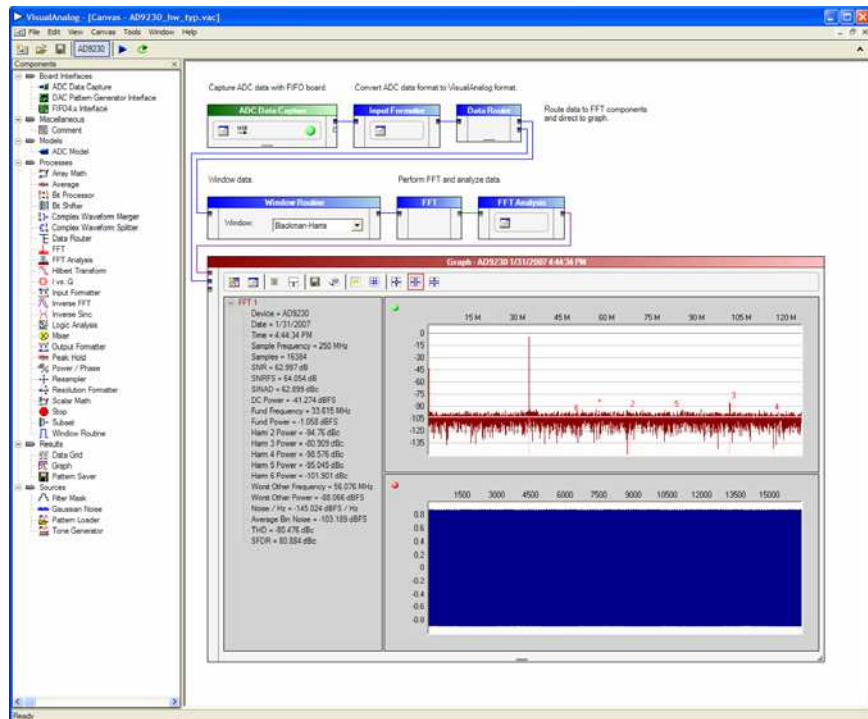


Figure 1. Typical VisualAnalog Canvas

## TABLE OF CONTENTS

Introduction .....	1	Bit Processor .....	24
Product Highlights .....	1	Bit Shifter .....	24
Installation of VisualAnalog.....	3	Comment.....	24
Instruction Notes.....	3	Complex Waveform Merger .....	25
Quick Start for ADC Evaluation.....	4	Complex Waveform Splitter.....	25
VisualAnalog Software.....	6	Data Router .....	25
Starting VisualAnalog.....	6	FFT .....	25
Using the Start-Up Form .....	6	FFT Analysis .....	25
Opening a Blank Canvas .....	7	Hilbert Transform .....	27
Using the Main Form.....	7	Input Formatter .....	27
Setting Canvas Properties.....	9	Inverse FFT .....	27
Placing Components.....	9	Inverse Sinc .....	27
Connecting Components .....	10	I vs. Q .....	27
Adjusting Component Parameters.....	12	Logic Analysis.....	27
Updating Results.....	13	Mixer.....	28
Making Layout Changes.....	14	Output Formatter .....	28
Using the Menu Bar .....	14	Peak Hold .....	28
Using the Tool Bar .....	15	Power/Phase.....	28
Using the Options Form.....	16	Resampler.....	29
Data Types Overview .....	17	Resolution Formatter .....	29
Real Waveform Data .....	17	Scalar Math .....	29
Complex Waveform Data.....	17	Stop.....	29
Real FFT Data .....	17	Subset .....	29
Complex FFT Data.....	17	Waveform Analysis .....	29
Analysis Data .....	17	Window Routine .....	29
Value Collection .....	17	Components Results .....	30
Numeric Value .....	17	Data Grid.....	30
Tone List .....	17	Graph .....	30
Components Overview.....	18	Pattern Saver .....	32
Board Interfaces.....	18	Components Sources .....	33
ADC Data Capture.....	19	Filter Mask.....	33
DAC Pattern Generator Interface.....	20	Gaussian Noise .....	33
DAC Pattern Generator Control Form.....	20	Pattern Loader .....	33
Pattern Limitations.....	21	Tone Generator .....	34
FIFO4.x Interface.....	21	VisualAnalog Example Canvases .....	35
Components Models .....	23	ADC with ADC Data Capture Board.....	35
ADC Model .....	23	ADIsimADC Model File .....	35
Components Processes .....	24	Loading the DPG with a Simple Vector .....	38
Array Math .....	24	Loading the DPG with a Complex Vector .....	39
Average.....	24		

## INSTALLATION OF VisualAnalog

To install VisualAnalog, you must have the following:

- Administrator privileges
- Microsoft® .NET Framework Version 1.1
- The latest **.NET Framework 1.1** service packs

The VisualAnalog installation package installs all of the items needed to use the VisualAnalog software along with the necessary drivers for the ADC and DAC hardware.

### INSTRUCTION NOTES

- Disconnect all Analog Devices, Inc. ADC data capture boards and/or the DPG from the computer before installing the software. Be sure to finish the installation of the software before attempting to connect any related hardware to ensure the proper installation and registration of the device drivers.
- Note that you need administrator privileges when installing this software package and when connecting the ADC and DAC hardware to the computer for the first time. If using the Hardware Wizard, follow through the instructions to install the software automatically. This allows Windows® to complete the driver installation process.

VisualAnalog is a Microsoft .NET application. You must have .NET Framework Version 1.1 on your machine to run VisualAnalog. The preferred way to obtain the .NET Framework is through Windows Update. Be sure to get the latest service packs available as well.

To determine if the .NET Framework 1.1 is already installed on your computer, click **Start**, select **Control Panel**, and click **Add or Remove Programs**. When the window appears, scroll through the list of applications. If you see **Microsoft .NET Framework 1.1** listed, you have the correct version and you do not need to install it.

1. To ensure proper installation of all the components you must have administrator privileges.
2. Disconnect all ADC data capture boards and/or the DPG from the computer.
3. Install the **.NET Framework 1.1** and the latest **.NET Service Pack**.
4. Run the VisualAnalog installation executable. Follow the on-screen instructions to install all of the necessary files. If interfacing with the DPG, be sure to launch the Hardware Wizard at the end of the installation process to set up the appropriate DPG device drivers.
5. Power up and connect the DPG and/or any ADC data capture board to the computer to finish the driver installation process. If using the DPG, be sure to connect the DPG before the ADC data capture board. If not using the DPG, you can plug in the ADC data capture board at any time after you install the software.



## QUICK START FOR ADC EVALUATION

With VisualAnalog, it is easy to bypass the canvas interface and begin ADC evaluation immediately. To begin interfacing to a particular ADC right away, use the following steps:

1. Connect and power the evaluation board, ADC data capture board, and any other required board used for data transfer. You can also supply the required clock and input signals to the ADC evaluation board.
2. Connect the ADC data capture board to the computer with a high speed USB cable. If a driver installation dialog appears, as happens when using the ADC data capture board for the first time, proceed through the dialog steps.

If the Hardware Wizard appears (see Figure 2), follow the instructions to install the software automatically. This allows Windows to complete the driver installation process.



Figure 2. Hardware Wizard

3. Start VisualAnalog. For more information, see the VISUALANALOG Software section.
4. The start-up form now appears. If an ADC was connected using the preceding steps, VisualAnalog attempts to detect it and selects the canvas template that supports the ADC on the start-up form. Note that the ADC must support SPI® functionality for the program to autodetect. If the ADC does not support SPI, or if the program does not detect the ADC for any reason, manually select the template. See the Using the Start-Up Form section for more information.

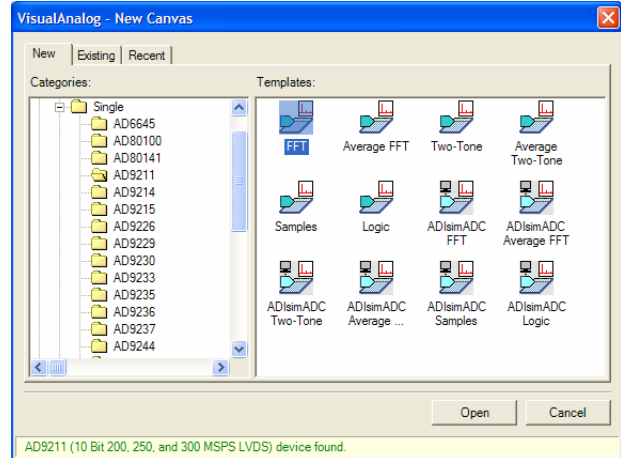


Figure 3. New Canvas Form

5. Select the FFT icon and click **Open**.
6. If you have an HSC-ADC-EVALC data capture board, a dialog box that asks for permission to configure the on-board FPGA may appear. If you prefer to use the current FPGA configuration, click **No** to bypass configuration. Otherwise, click **Yes** to configure the FPGA. See the Using the Start-Up Form section for more information.

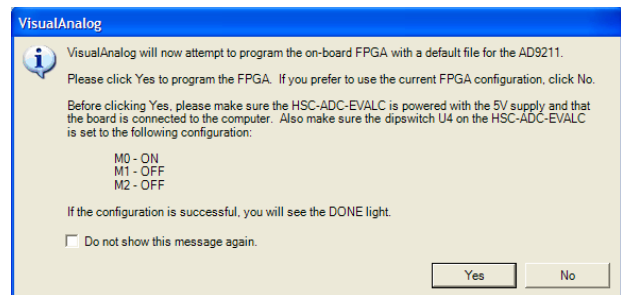


Figure 4. FPGA Configuration Dialog

7. The main form appears in collapsed mode with your canvas open and selected. Click **Update** to run the canvas.



Figure 5. Update Button

A **Graph** form should appear with the FFT results. If the graph does not appear, it is possible there was an error during processing. Check the board connection and try again. If this does not fix the problem, expand the main form and check the canvas settings. For more information, see the Using the Main Form and the Components Overview sections.

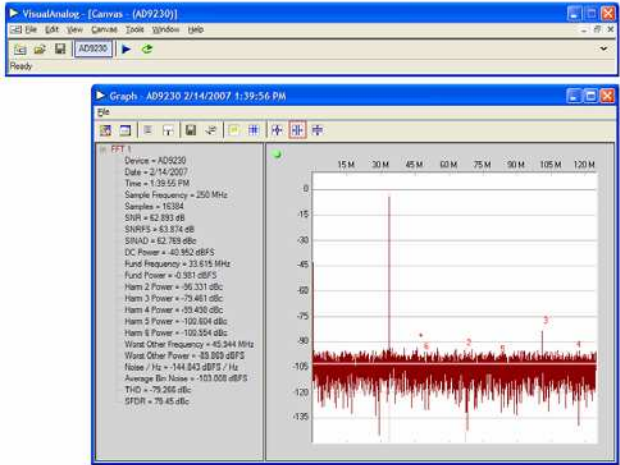


Figure 6. Main Form and Graph Form

To view the time domain representation of the captured data, click **Toggle Additional Plot** on the graph form. See the Components Overview section for more information.



Figure 7. Toggle Additional Plot Button

## VisualAnalog SOFTWARE

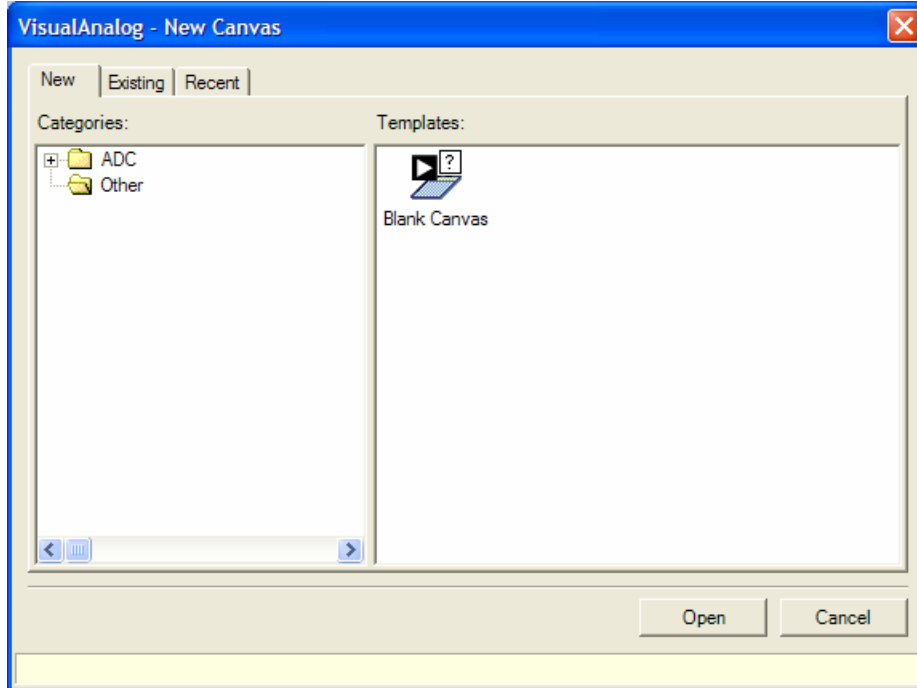


Figure 8. VisualAnalog Start-Up Form

### STARTING VISUALANALOG

After installing the VisualAnalog software, a **Start** menu item and a desktop icon should appear. To use the **Start** menu item, click **Start**, select **Programs**, select **Analog Devices**, click **VisualAnalog**, and select **VisualAnalog**.

To use the desktop icon, click the **VisualAnalog** icon from the desktop.

When the program starts, a splash screen appears while loading is in process, and the **Start-Up Form** appears after the splash screen disappears.

### USING THE START-UP FORM

Select the **New** tab at any time to load a blank canvas or a predefined canvas template. VisualAnalog maintains a list of templates that set up the canvas for running a common task or interface with a particular device.

To select a canvas template, expand the **Categories** tree until the appropriate device is visible. Then, select an icon in the **Templates** list and click **Open**. VisualAnalog opens the canvas.

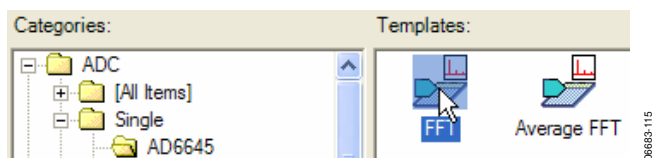


Figure 9. Selecting a Canvas Template

VisualAnalog can detect a connected ADC board if that ADC supports a programmable SPI interface. In addition, an ADC data capture board that supports SPI must be used. For this

autodetect to work correctly, both boards must be connected to the computer via a USB cable and powered up before the software is started. Windows must also recognize the ADC data capture board to ensure correct operation. If Windows does not recognize the board, then there is a USB problem. Refer to the data sheet of the particular board for more information.

If VisualAnalog detects an ADC board, the software displays information in the status bar of the start-up form. In addition, it finds an item in the category tree that supports the ADC device.

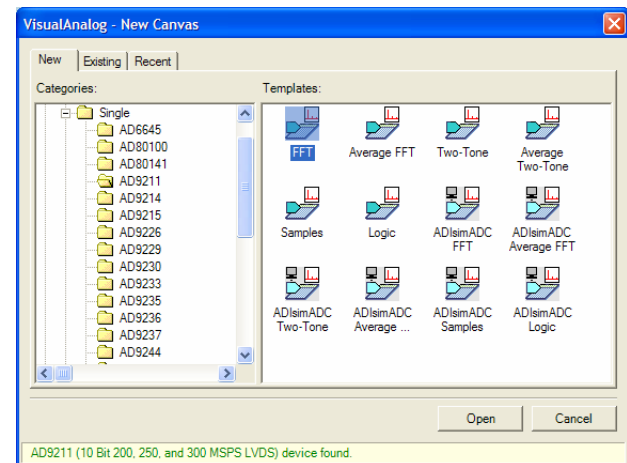


Figure 10. Visual Analog—New Canvas Window

If the software does not detect the ADC board at this point, manually select the correct category and template.

When choosing a template that represents the detected ADC board, with an interface with to HSC-ADC-EVALC data capture board, a dialog box may appear that asks for permission to configure the on-board FPGA. If the HSC-ADC-EVALC FPGA did not configure on power-up for the particular ADC you are evaluating, click **Yes**. Click **No** to bypass the FPGA configuration.

If you check **Do not show this message again**, the software automatically performs the last selected action in the future, when choosing an applicable canvas template. If you want to change this option, access these settings from the **Options** menu in VisualAnalog.

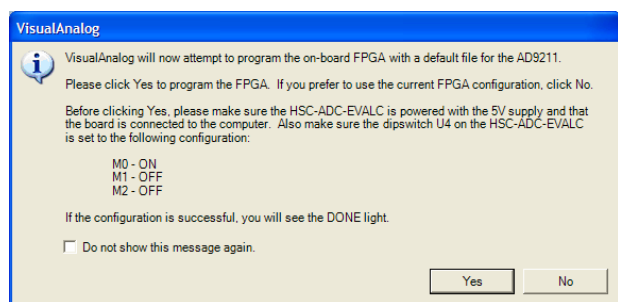


Figure 11. FPGA Configuration Dialog

VisualAnalog maintains a list of FPGA files that correspond with a device ID that exists in the SPI register map of the ADC. When the software detects a particular device ID that exists in the table, it can choose a default FPGA configuration file. Note that this process only occurs when using the HSC-ADC-EVALC.

Select the **Existing** tab to browse for an existing canvas file.

VisualAnalog maintains a list of the last five recently accessed canvas files. To access this list and open one of these recent files, select the **Recent** tab.

Click **Open**, VisualAnalog opens the selected canvas file or canvas template.

Click **Cancel** and the main form appears with no open canvases.

## OPENING A BLANK CANVAS

To open a blank canvas, navigate to **Other** in the category tree, and select the **Blank Canvas** template icon. Click **Open**.

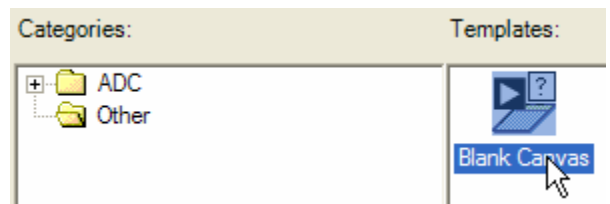


Figure 12. Opening a Blank Canvas

## USING THE MAIN FORM

The VisualAnalog main form maintains all currently open canvases in an MDI environment. The left side of the screen shows available **Components** (see Figure 14). This form docks to the left by default, but can be moved and docked to any of the corners of the VisualAnalog program. In addition, the **Components** form can be floated and moved anywhere on screen, even outside the VisualAnalog window.

### Canvas Buttons

As you open or create canvases, buttons are created across the tool bar, each button represent a canvas. These are canvas buttons. You can use these canvas buttons to select or clear a canvas for updating. When the button appears highlighted, the canvas runs on the next update. You must select the canvas in order to run it.

The text that appears on the canvas button is the **Display Name** for the canvas. Change this property by selecting **Canvas > Properties**.

It is possible to collapse the main form (see Figure 13). This is useful when you want to conserve screen area and you do not need to adjust canvas settings. To collapse the main form, click the arrow on the right side of the main forms tool bar. To expand the form, simply click this button again.



Figure 13. Collapsing and Expanding the Display



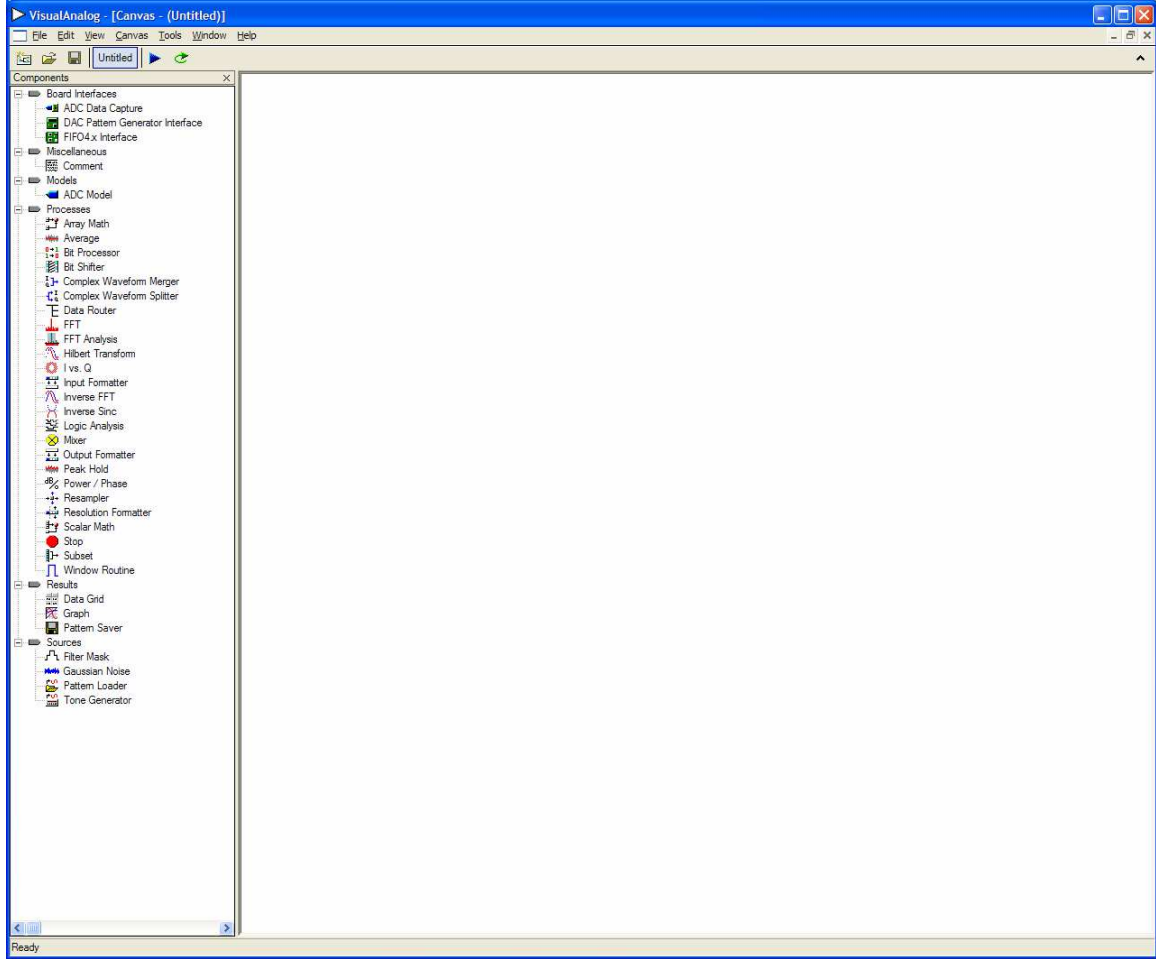


Figure 14. VisualAnalog Main Form

### SETTING CANVAS PROPERTIES

VisualAnalog allows you to set properties that describe the canvas and the way it behaves. To access the canvas properties, click **Canvas > (Display Name) Properties...** if a blank canvas is opened, **(Display Name)** appears as **Untitled**.

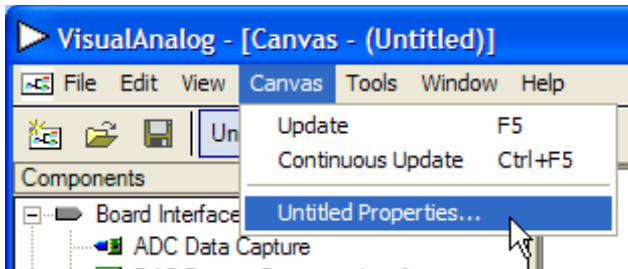


Figure 15. Canvas Properties Menu

Use the **Canvas Properties** form to adjust the properties of a particular canvas.

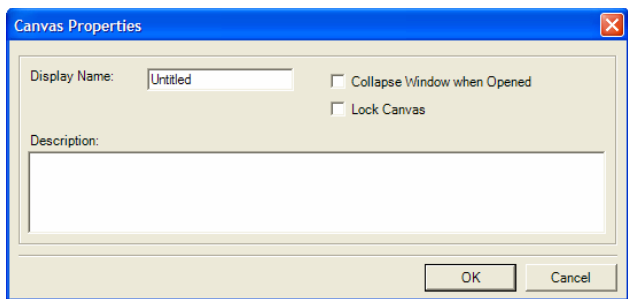


Figure 16. Canvas Properties Form

The **Display Name** refers to the name that appears on the canvas button. This is the title for the canvas.

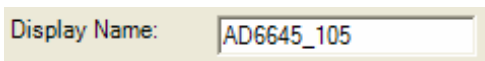


Figure 17. Display Name

You can also enter a description of the canvas.

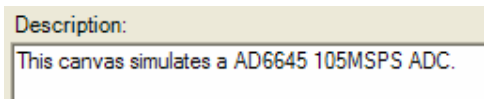


Figure 18. Description

**Collapse Window when Opened** allows you to collapse the main window when this canvas is opened in the future. It permits opening the canvas in a more formal mode without showing the canvas itself.

**Lock Canvas** allows you to lock the canvas from further layout changes. You can still change component settings when this is set, but you cannot alter the physical layout and connectivity of your canvas.

### PLACING COMPONENTS

To place a component on the canvas, either double click the item in the **Components** tree or drag the item and drop it on the canvas. Dragging the item gives more control over component placement.

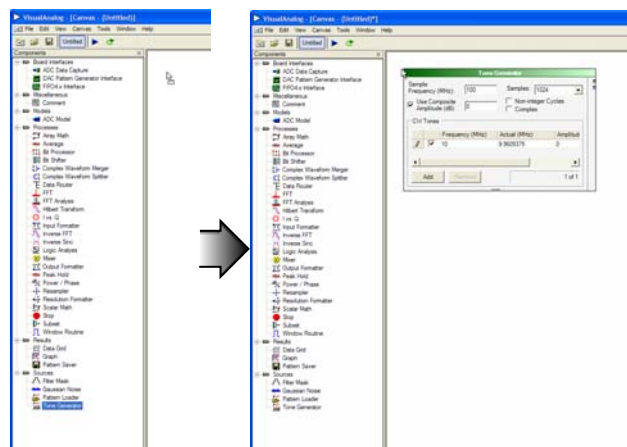


Figure 19. Placing a Component

Sample placements of components are shown in Figure 19 in the following order: **Tone Generator, ADC Model, Input Formatter, Data Router, Window Routine, FFT, FFT Analysis, and Graph.**

For a brief summary of the functionality of each component, see the Components Overview section. The canvas should look similar to Figure 20, disregarding any component placement differences.

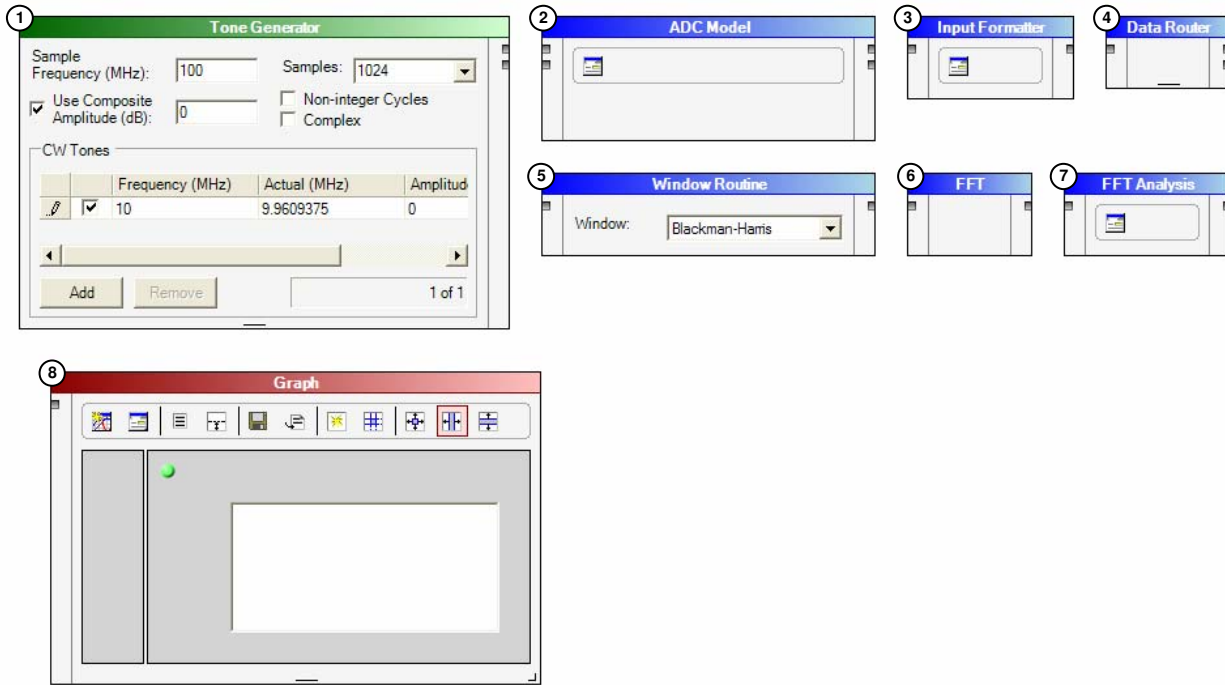


Figure 20. Sample Component Placement

## CONNECTING COMPONENTS

To connect two components together, place a wire from the output node of one component to the input node of another, or vice versa. To place a new wire, use one of the following techniques:

- Click the node, drag the new wire to another component, and click again to connect.
- Drag the new wire to another component.

If the wire is not connected, it appears red. As soon as you apply the connection, the color changes.

Figure 21 illustrates the process of connecting the first output node of the **Tone Generator** to the first input node of the **ADC Model**.

For this example, place eight more wires. Figure 22 shows the canvas after making the remainder of these connections. Your canvas should look similar to this Figure 22.

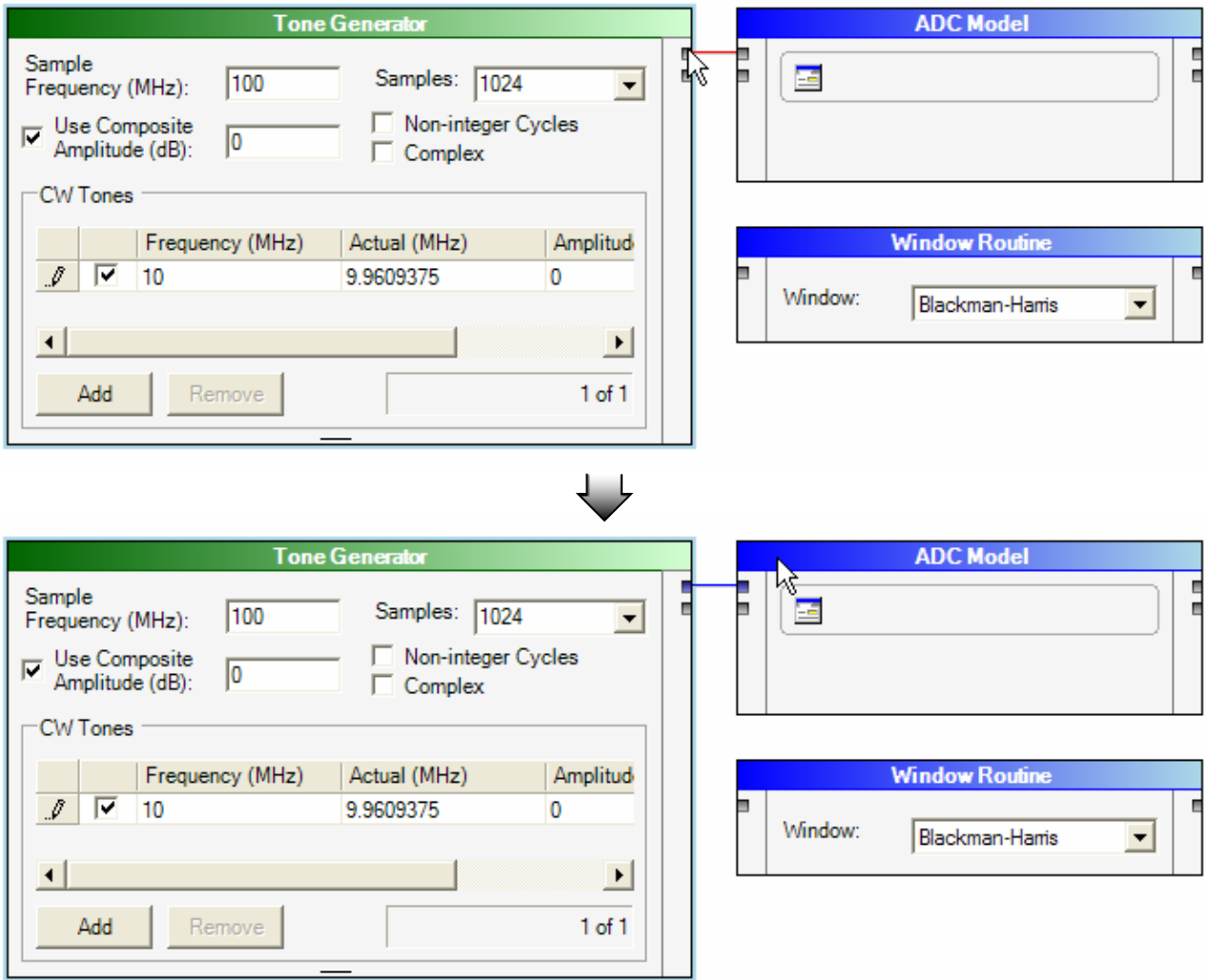


Figure 21. Connecting Components

06863-019

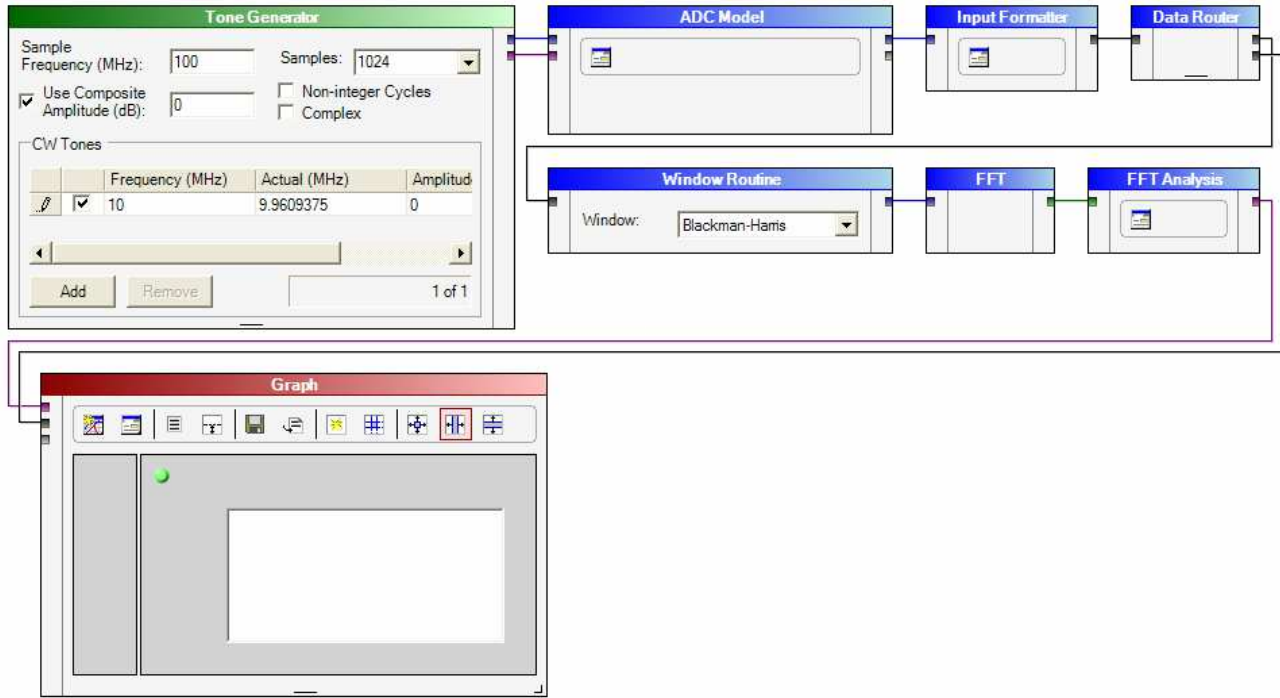


Figure 22. Sample Connections

When you connect and move components, wires route on the canvas automatically. If the diagram begins to look convoluted, try reorganizing some components to force the wires to reroute.

To connect the end of an existing wire to a new node, do one of the following:

- Click a node, drag the opposite end of the wire to another node, and click again to connect.
- Drag the selected end to another node.

Like physical wires, wires in VisualAnalog carry information between components. Although it is transparent to the user in most cases, a wire can transfer one of several different data types. See the Data Types Overview section for more information.

### ADJUSTING COMPONENT PARAMETERS

Some components have adjustable parameters. The steps that follow show how to generate a simple waveform as input for an ADC model and display FFT and time domain results.

1. In **Tone Generator**, set the **Sample Frequency (MHz)** to 105. Also, change the **Samples** text to 16384, by using the drop-down arrow or entering the text manually. Update the **Use Composite Amplitude (dB)** to -1 (because you are evaluating ADC performance).

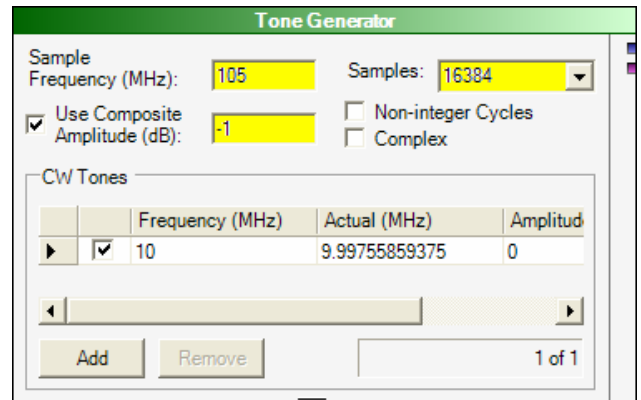


Figure 23. Tone Generator Settings

2. In **ADC Model**, click the **Settings...** button and click **Open** to browse for and select the **AD6645\_105.adc** model file. This file is located in the **Models\ADC** subdirectory in the VisualAnalog path. When the model opens, the file name appears in the **Model File** text box, along with information about that model in the **Properties** grid (see Figure 24). Click **OK**.



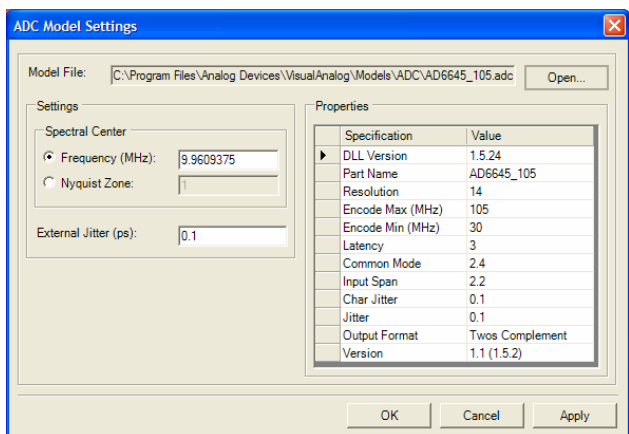
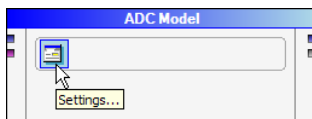


Figure 24. ADC Model and Settings Form

- In **Input Formatter** (see Figure 25 through Figure 27), click the **Settings...** button. Then change the **Number Format** to two's complement. Next, change both the **Resolution** and **Alignment** to 14. Click **OK**.

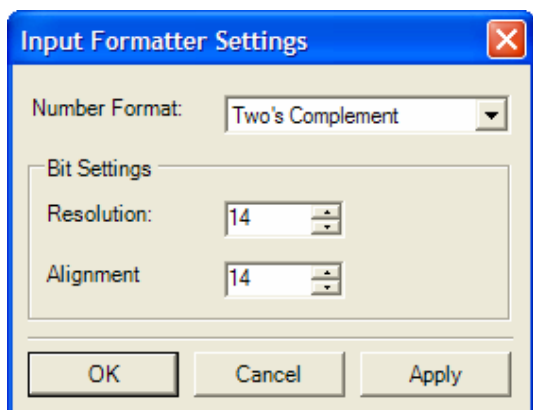
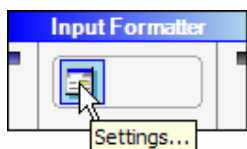


Figure 25. Input Formatter and Settings Form

- Resize the **Graph** component by positioning the cursor over the corner resize handle, drag the component out to a larger size (see Figure 26).

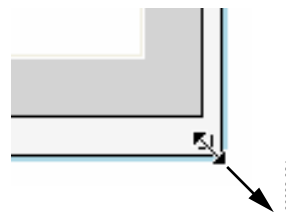


Figure 26. Resizing the Graph

Next, resize the **Analysis Results** panel on the left side of the **Graph** component by positioning the mouse cursor over the vertical divider bar and drag to a new location (see Figure 27).

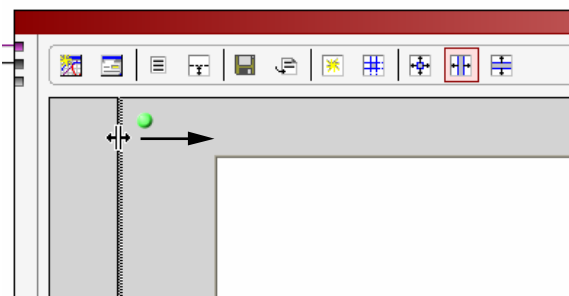


Figure 27. Resizing Analysis Results

Notice that there are two wires connected to the graph. This is because we have routed ADC samples to the graph as well as the FFT results. If you want to see the time domain representation of the data, you can click the **Toggle Additional Plot** button to view the second plot.

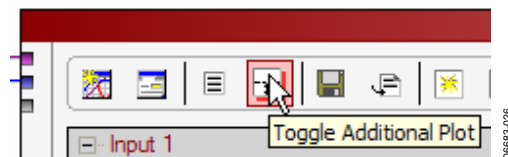


Figure 28. Toggle Additional Plot

### UPDATING RESULTS

After making all required adjustments, you are ready to update the results of the canvas. There are three ways to update the canvas

- Press the Shortcut Key F5 or Ctrl + F5.
- Select the menu command **Canvas**, click **Update or Canvas**, and select **Continuous Update**.
- Click **Update** or **Continuous Update** on the tool bar.

After updating, the canvas should look similar to Figure 29.

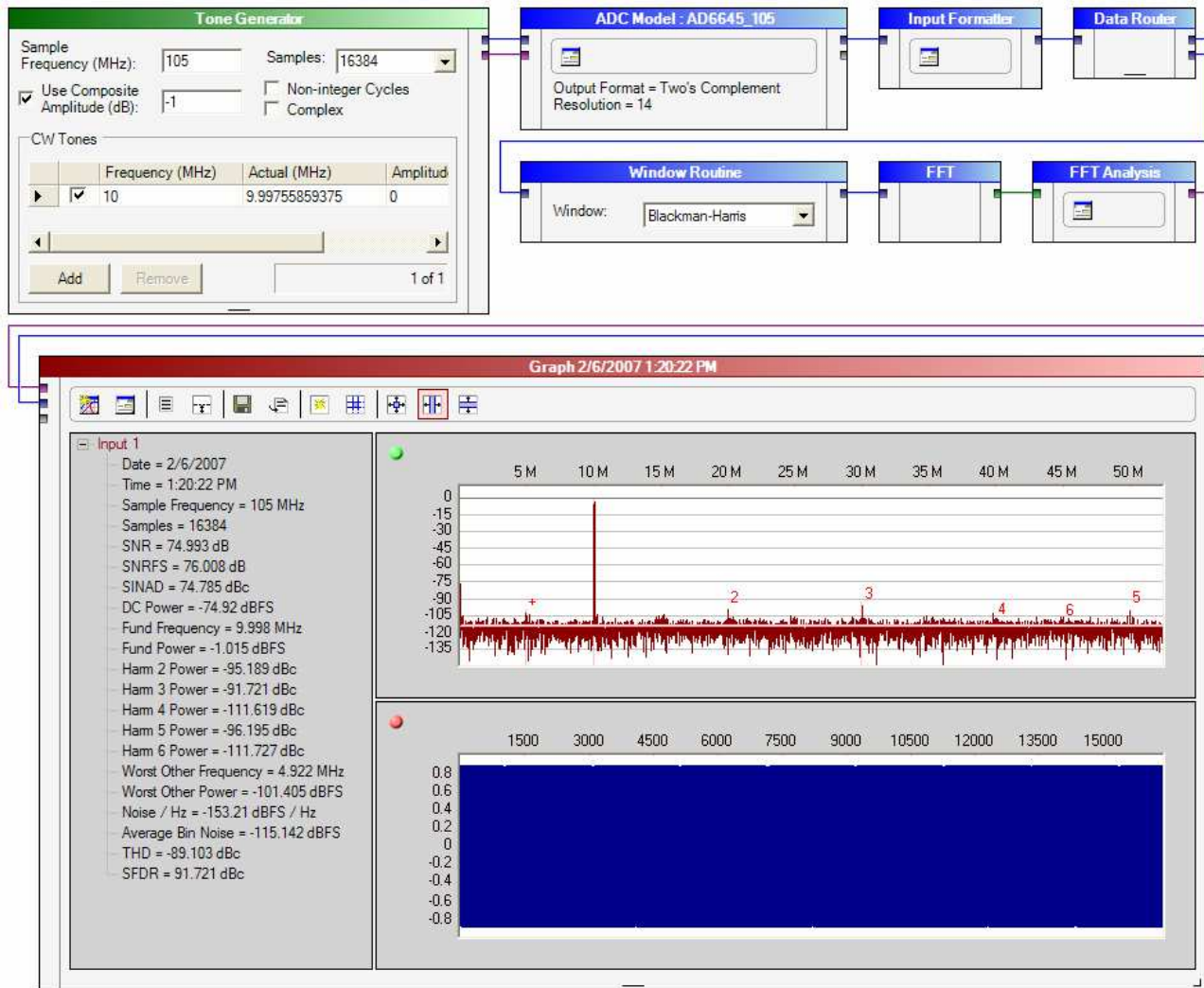


Figure 29. Updated Canvas

## MAKING LAYOUT CHANGES

Most layout changes require that you select an object first. To select a component or wire, click the mouse somewhere on the object. If you are trying to select a component, be sure to click somewhere on the body (the header at the top always works). Hold down the Ctrl key to select or deselect more than one object.

Alternatively, you can draw a box on the canvas to select objects. Just press the mouse somewhere on the blank canvas, drag, and release around the objects you wish to select.

While objects are in a selected state, they can be deleted from the canvas. Click **Edit**, and select **Delete**, or just press the **Delete** key to remove the selected items.

While components are in a selected state, they can be moved to a new location. Press the left mouse button to click, hold, and drag one of the selected components to an empty location. Any connected wires automatically adjust accordingly.

Some components, such as the graph, are resizable. Just click, hold, and drag on the resize handle at the bottom of the component.

You can cut, copy, or paste components and wires to new locations or other Canvas windows. To cut or copy, select the desired components. If any selected components are connected, the software copies the connecting wire as well. The appropriate **Edit** commands in the menu or use the standard Windows shortcut keys. To execute the cut, copy, or paste commands, use Ctrl + X, Ctrl + C, or Ctrl + V.

You can undo (or redo) parameter changes and layout changes. VisualAnalog maintains an undo stack and a redo stack of up to five layers of user actions.

## USING THE MENU BAR

The menu bar provides access to a variety of file and execution options using a standard menu format as shown in Figure 30.



Figure 30. Menu Bar

## File

**New**—Opens a blank canvas used for building a new component diagram.

**Open**—Loads an existing canvas using a standard file browser.

**Close**—Shuts down the current canvas. If you have changed the canvas since the last save, the program asks if you wish to save before closing.

**Save**—Saves the current canvas under the existing canvas file name. If the canvas does not already have a name, the software prompts for one before the save.

**Save As**—Saves the current template under a new name.

**Recent Files**—Displays a list of the last five canvas files opened or saved.

**Exit**—Exits VisualAnalog.

## Edit

**Undo**—Reverses up to the last five actions, including deletion, parameter changes, component placement, and connectivity.

**Redo**—Performs the last undone action again.

**Cut**—Copies the selected objects to the clipboard and then deletes them from the canvas.

**Copy**—Copies the selected objects to the clipboard.

**Paste**—Places the items from the clipboard on the canvas.

**Select All**—Selects all items on the canvas.

**Delete**—Removes all currently selected items from the canvas.

## View

**Components**—Displays the **Components Tool** form if it is not visible.

## Canvas

**Update**—Runs the currently selected canvases by executing the component flow on each canvas.

**Continuous Update**—Causes the selected canvases to run continuously. When started, the **Continuous Update** menu item changes to **Stop Update**. Selecting this stops all processing. You can also automatically stop continuous update by using the **Stop Component**. See information on the **Stop Component** for more details.

**Properties**—Displays editable properties for the currently activated canvas (the canvas focused for editing).

## Tools

**External Tools**—Displays a form with which the user can select external programs that VisualAnalog can open. Adding an executable item with this form places a new menu item underneath the **Tools** menu.

**Options**—Opens the VisualAnalog options form.

## Window

**Tile Horizontally**—Tiles the canvases in a horizontal direction.

**Tile Vertically**—Tiles the canvases in a vertical direction.

**Cascade**—Cascades the canvases.

**Canvas Selection**—Provides a list of open canvases. From this list, the user can select a canvas for editing.

## Help

**User Manual**—Opens this user manual with the associated PDF viewer.

**About VisualAnalog**—Displays the VisualAnalog version number and other information.

## USING THE TOOL BAR

The tool bar provides quick access to common features that are available on the menu bar.

**New Canvas**—Opens a blank canvas used for building a new component diagram.



Figure 31. New Canvas Button

**File Open**—Loads an existing canvas using a standard file browser.



Figure 32. File Open Button

**File Save**—Saves the current canvas under the existing canvas file name. If the canvas does not already have a name, the software prompts for one before the save.



Figure 33. File Save Button

**Update**—Runs the currently selected canvases, by executing the component flow on each canvas.



Figure 34. Update Button

# AN-905

**Continuous Update**—Runs the selected canvases continuously. When started, the **Continuous Update** menu item changes to **Stop Update**. Selecting this stops all processing. You can also automatically stop continuous update by using the **Stop Component**. See information on the **Stop Component** for more details.



Figure 35. Continuous Update and Stop Update Icons

## USING THE OPTIONS FORM

The options form contains settings that affect the behavior of VisualAnalog. You can access the VisualAnalog options form by clicking the menu **Tools** and selecting **Options**.

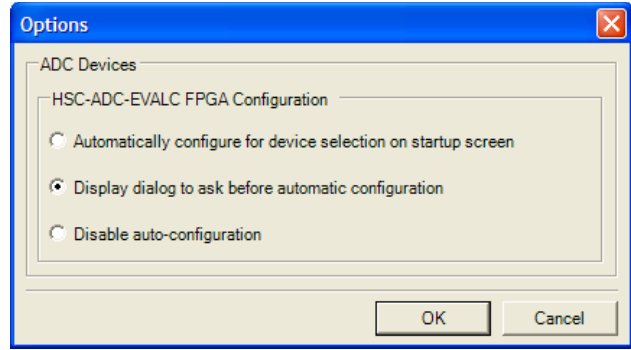


Figure 36. Options Form

### ADC Devices

**ADC Devices** contains options that affect the behavior of VisualAnalog when interfacing with ADC data capture boards.

**HSC-ADC-EVALC FPGA Configuration** allows you to adjust automatic FPGA configuration settings for the HSC-ADC-EVALC by selecting one of three options. These options only have an effect when choosing a template from the startup form that corresponds to a detected ADC device. For more information, see the Using the Start-Up Form section.

## DATA TYPES OVERVIEW

VisualAnalog components transfer information through wires. The wires themselves can carry any type of data, but most component inputs have limitations on what types of data they can accept. Most components that have an output pass a certain type of data to the next component.

You can pass any data type to the **Data Grid** component which displays an important portion of the data type. For example, when the **Data Grid** displays **Real Waveform Data**, it displays the list of samples.

### REAL WAVEFORM DATA

Real waveform data consists of an array of real samples along with the sample frequency.

### COMPLEX WAVEFORM DATA

Complex waveform data consists of an array of complex samples (I and Q) along with the sample frequency.

### REAL FFT DATA

Real FFT data consists of an array of data that is the FFT result of real waveform data, along with the sample frequency.

### COMPLEX FFT DATA

Complex FFT data consists of an array of data that is the FFT result of complex waveform data, along with the sample frequency.

### ANALYSIS DATA

Analysis data consists of analysis results along with graph data and formatting information. This type of data can behave differently depending on the component that outputs it. Normally, this data type is passed straight to a **Graph** component because that component displays all the pertinent information. If you send this data to a **Data Grid** component, it displays the analysis results portion of the data only.

### VALUE COLLECTION

Value collection consists of parameter and value pairs. Components use this data type to display information.

## NUMERIC VALUE

Numeric value contains only a floating-point number. However, some components treat this number as an integer. The **Average** and **Peak Hold** components both output a numeric value to indicate when their current sequence is finished. The **Stop** and **Graph** components both can use this value for a control input. See the Components Overview section for more information about the use of numeric value.

## TONE LIST

The **Tone Generator** outputs the tone data type. The tone list contains frequency, phase, and amplitude information about the tones generated.

Note that two numeric formats can occur within VisualAnalog when using the waveform types. These are normalized data (which the majority of processing components use) and integer data.

The **ADC Model**, **ADC Data Capture**, and the **FIFO4.x Interface** components are all output integer data. Follow these components with an **Input Formatter** component to normalize the data.

The **Pattern Saver** and **Pattern Loader** components can support either data format as needed by the application. When loading a vector file, if the format is uncertain, manually examine it to determine how to process the file. Integer format should be obvious as either text-readable integers or hexadecimal values, and normalized data format appears as text-readable floating-point numbers. Note that VisualAnalog only assumes hexadecimal format for files with a .hex extension. To input hexadecimal values, rename the extension to .hex.

Finally, the **DPG Interface** always expects integer type format. If the waveform is not already in this format, precede this component with an **Output Formatter** component.

See the Components Overview section for more specific information on the requirements of each component.



## COMPONENTS OVERVIEW

VisualAnalog provides a variety of components used to accomplish tasks. Though some of these have a fixed mode of operation, others have a range of adjustable parameters that allow for customizable operation. Some of the components require you to access a settings form to adjust parameters. To access these forms, click the **Settings** button on the component.



## BOARD INTERFACES

The following sections are a basic overview of the functionality of the components.



Figure 38. DAC Pattern Generator

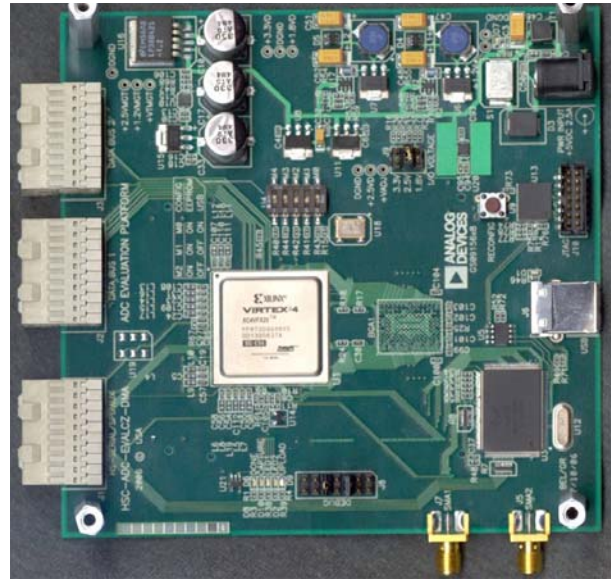


Figure 39. HSC-ADC-EVALC (ADC Data Capture Board)

## ADC DATA CAPTURE

The **ADC Data Capture** component is responsible for acquiring data from high speed ADC data capture boards. This component interfaces with a wide variety of ADCs, including those with specialty outputs such as power and other on-chip measurements.

An **Input Formatter** must always follow the **ADC Data Capture** component if you want to perform processing within the canvas. The only exception to this rule is **Logic Analysis**. The **Input Formatter** takes the resolution and alignment of the ADC into account when moving data into the environment of VisualAnalog. For more information on setting up the **Input Formatter**, see the Input Formatter section.

Most of the Analog Devices ADC evaluation boards are MSB-justified to 16 bits and, therefore, the **Alignment** box within the **Input Formatter** should be set to 16 bits. The resolution on the **Input Formatter** should be set to the native resolution of the ADC. **Number Format** should be set in accordance with the data format of the ADC.

Access **ADC Data Capture Settings** by clicking **Settings**.

Use the **General** tab to set parameters about the device you are using and the data you want to capture.

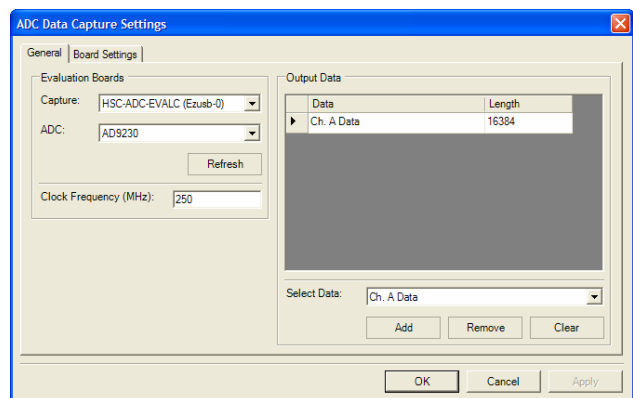
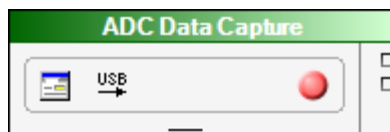


Figure 40. ADC Data Capture and Settings Form, General Tab

**Capture**—Shows the USB devices available. Use this drop-down box to select the board you want to use. Boards appear in the order in which they were connected. If no devices are found, check the cables and power supplies and click **Refresh**.

**ADC**—Displays special configurations available for the ADC that you are using. Select one or use the **Default** setting.

**Clock Frequency (MHz)** —Should be set to the sample rate of the device.

**Output Data**—Determines which outputs the component passes back to the canvas. You can choose multiple outputs that provide data from multiple core ADC outputs or pass the same output to multiple processes on the canvas much like the **Data Router** component. Modify the default options for **Output Data** with the **Add**, **Remove** or **Clear** buttons. To add a new selection, simply choose from the available selections in the **Select Data** drop-down box and click **Add**. **Remove** deletes an existing **Output Data** entry. **Clear** removes all outputs and allows a completely new set of outputs to be added.

In **Output Data**, set the length to the sample size of the capture board FIFO or the desired sample size, whichever is less. Normally, the HSC-ADC-EVALA and HSC-ADC-EVALB boards have 32-KB devices. It is possible to reconfigure them with compatible 256-KB devices. The HSC-ADC-EVALC board can support various sample sizes, depending on the current FPGA configuration.

Use the **Board Settings** tab to set board-related parameters.

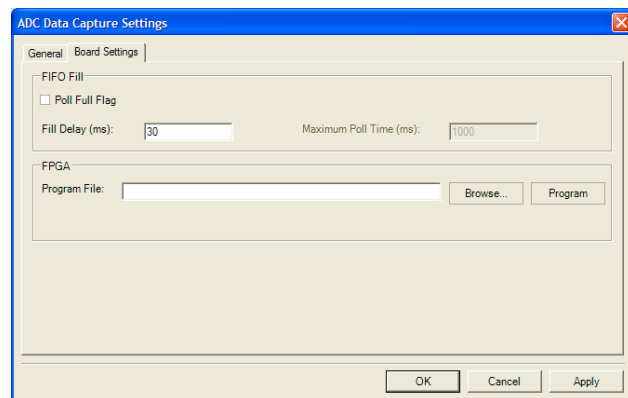


Figure 41. ADC Data Capture Settings Form, Board Settings Tab

**FIFO Fill** allows you to adjust parameters that are used while filling the on-board FIFOs with ADC data. Normally, the software sends the fill command to the FIFOs and waits a specified amount of time before reading the data. You can set the delay time by inserting the desired amount in the **Fill Delay (ms)** option.

When you have a slow clock rate, sometimes it is useful to poll the FIFOs for a full flag, which indicates when the FIFOs are full. Select the **Poll Full Flag** check box and then set the **Maximum Poll Time** so the software stops polling after a certain amount of time if it never receives a full flag. Note that this setting is not recommended for the HSC-ADC-EVALC because the FPGA configuration may not use a full flag.

**FPGA** allows you to adjust FPGA settings specifically for the HSC-ADC-EVALC board. You can program the on-board Xilinx® FPGA through the USB interface by entering a firmware file into the **Program File** text box (click **Browse** to select one on disk) and then click **Program**. You may from time to time receive an update to FPGA firmware for use with a particular ADC; manually program it using this method. FPGA firmware files have a .bin extension.

## DAC PATTERN GENERATOR INTERFACE

The **DAC Pattern Generator Interface** provides access to the physical DPG board via a standard USB interface. This tool handles all of the hardware interfacing and data formatting issues that are required to move data from a software data set (vector) to the hardware. Additional information on the DPG is available at [www.analog.com](http://www.analog.com). Note that only one DPG should be connected at any given time. When using the DPG along with other USB devices in VisualAnalog, be sure to connect the DPG first.



Figure 42. DAC Pattern Generator Interface

The **DAC Pattern Generator Interface** component requires unsigned data that matches the resolution, bit alignment, and data format of the DAC. When using VisualAnalog processed data, the **Output Formatter** is required to format the data for the DPG device in a numeric format and resolution that the DAC expects. If this is not set properly, incorrect data sent to the DAC results in erroneous performance. Consult the DAC product data sheet to ensure that the output formatter is set properly. For more information on setting up the **Output Formatter**, see the Output Formatter section.

## DAC PATTERN GENERATOR CONTROL FORM

Click **Settings** on the **DAC Pattern Generator Interface** to open the **DAC Pattern Generator Settings** form. You can use this form to adjust settings on the DPG and control playback of patterns. The form consists of four tabs: **Setup**, **Tuning**, **Debug**, and **Tx Config**.

### Setup

Use the **Setup** tab to set the DPG in the desired operating mode and to control data playback settings.

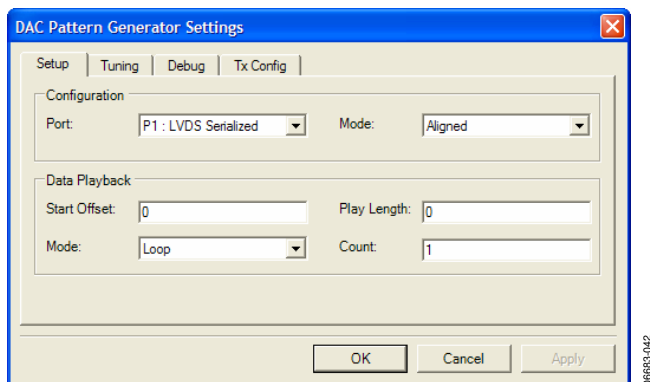


Figure 43. Setup Tab

**Configuration** allows you to set the DPG in the desired operating mode.

Table 1. DPG Operating Modes

Port	Mode
P1: LVDS Serialized	Clock aligned Clock centered
P2: LVDS Direct	SDR DDR: centered DDR: coincident
P3: LVCMOS	Single port Dual port

**Data Playback** controls playback settings and start/stop playback and becomes active when the control is fully configured and data is loaded.

Table 2. Data Playback Controls

Control	Description
Start Offset	Specifies the start location of the playback with respect to the first vector data value. Must be a multiple of 256 bits (32 bytes).
Play Length	Specifies the data length to playback. Must be a multiple of 256 bits (32 bytes).
Mode	Sets the desired playback mode as follows: Loop - File content is played and repeated until the session is stopped. Count - File content is played and repeated the number of times specified in the Count field. Once - File content is played only once on the output port.
Count	Specifies the playback count. Only active when the count mode is selected.
State Indicator	Provides general playback status information to the user.
Play/Stop Button	Starts/stops a playback session. A vector must be loaded.

### Tuning

The **Tuning** tab holds the controls for data and clock tuning.

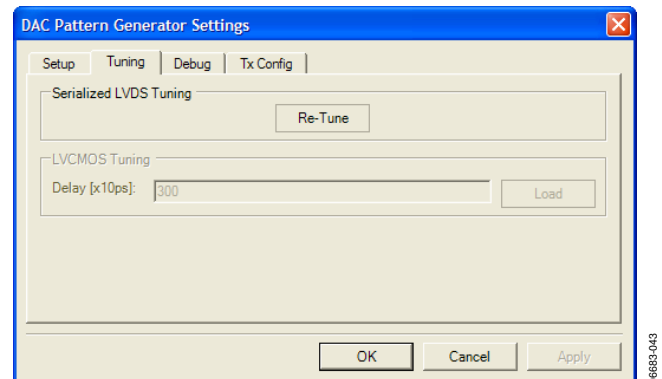


Figure 44. Tuning Tab

**Serialized LVDS Tuning** holds the controls related to tuning for the serialized LVDS port.

**Table 3. Serialized LVDS Tuning Controls**

Control	Description
Retune Button	Manual trigger to retune the clock and data bits on the serialized LVDS port. The alignment type is based on the selected mode (aligned or centered). Tuning is automatically performed when playback is started.

LVC MOS Tuning holds the controls related to tuning for the LVC MOS port.

**Table 4. LVC MOS Tuning Controls**

Control	Description
Delay	Specifies the desired delay value. The value range is from 0 to 1023 (0x3FF) in 10 ps units. This allows a variation of approximately 10 ns.
Load Button	A manual trigger to load the specified delay value for the clock and data bits on the LVC MOS port.

**Debug**

The **Debug** tab allows you to verify functionality of the DPG.

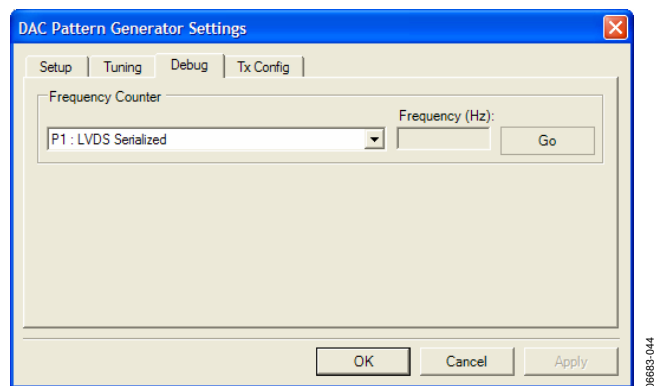


Figure 45. Debug Tab

**Frequency counter** allows readback of the frequency counter from the DPG to ensure correct operation.

**Table 5. Frequency Counter Controls**

Control	Description
Port Selection	Selects the output port being used before reading back the frequency
Frequency Indicator	Displays the frequency read back from the DPG
Go Button	Reads back the frequency counter from the DPG

**Tx Config**

The **Tx Config** tab holds controls for transmission features.

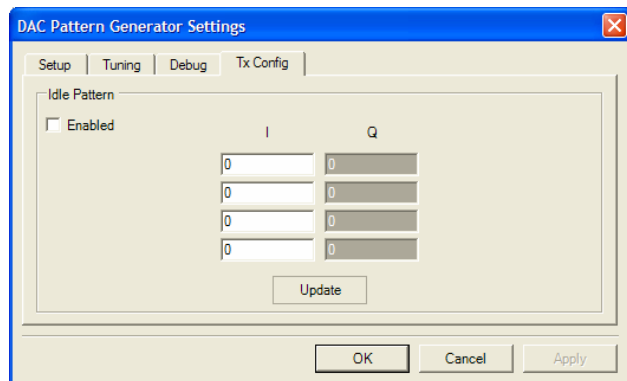


Figure 46. Tx Config Tab

**Idle Pattern** specifies the transmit pattern when there is no playback in progress. A four-sample pattern can be generated. The four-sample values in the pattern are played sequentially, then looped and played again until a vector file playback or the feature is disabled.

**Table 6. Idle Pattern Controls**

Control	Description
Enabled	Enables/disables the idle pattern generation feature. When the feature is disabled, zeros are played on the output port.
Idle Pattern Text	Specifies the pattern values. When operating in dual stream mode, four patterns must be specified for each of the two streams. The values are played sequentially from top to bottom.
Update Button	Updates the idle pattern in the DPG.

**PATTERN LIMITATIONS**

The DPG is useful for playing a large range of user vector patterns. However, there are a few basic limitations to the data.

- The sample size of the output vector must be a multiple of 16 samples.
- The minimum vector length is 640 samples.
- The available memory in the DMM sockets limits the maximum vector length. Each sample takes two bytes in memory. Complex samples count as two samples in memory.

**FIFO4.x INTERFACE**

The FIFO4.x interface component handles all interfacing between the ADC capture board and the software package, simply moving data from the hardware to the software data set (vector). This interface exists for compatibility. The preferred ADC interface is the ADC data capture component.

Additional information is available at [www.analog.com/fifo](http://www.analog.com/fifo).

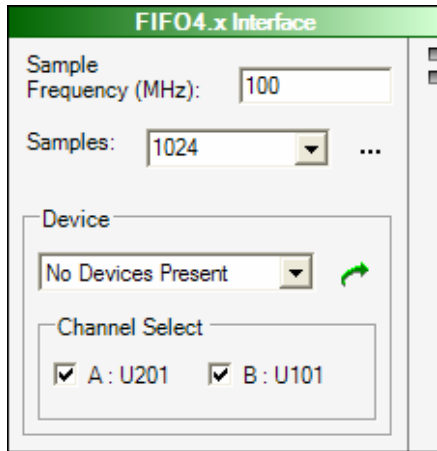


Figure 47. FIFO4.x Interface

You can use the ADC data capture board to capture data from the high speed ADCs. Interfacing with the data capture board is via USB.

FIFO configuration allows the sample rate and sample size to be set. Although sample rate is relative, sample size may be no larger than the physical memory available on the FIFOs. Normally, the HSC-ADC-EVALA and HSC-ADC-EVALB boards have 32-KB devices. It is possible to reconfigure them with compatible 256-KB devices. The HSC-ADC-EVALC board can support various sample sizes, depending on the current FPGA configuration.

In addition, you can select the channels to enable which devices are present. The interface supports both dual- and single-channel devices. Access the output from the ADC data at the output nodes on the right side of the component. The A and B channels are output separately, with the A data on the top node and B data on the bottom node.

An **Input Formatter** must always follow the **ADC Data Capture** component if you want to perform processing within the canvas. The only exception to this rule is **Logic Analysis**. The **Input Formatter** takes the resolution and alignment of the ADC into account when moving data into the environment of VisualAnalog. For more information on setting up the **Input Formatter**, see the Input Formatter section.

Most of the Analog Devices ADC evaluation boards are MSB-justified to 16 bits and, therefore, the **Alignment** check box within the **Input Formatter** should be set to 16 bits. The resolution on the **Input Formatter** should be set to the native resolution of the ADC. **Number Format** should be set in accordance with the data format of the ADC.



## COMPONENTS MODELS

In many instances, it may be desirable to use behavioral models instead of real devices, such as when test hardware is not available or if a device is in preselection phase. No matter the case, VisualAnalog supports converter models in addition to physical hardware, enabling a virtual test bench. Currently only ADC models are available using the Analog Devices ADIsimADC™ platform. This provides a seamless integration of models into the evaluation platform. It is also possible to include both models and real devices at the same time to compare predicted vs. actual performance.

### ADC MODEL

The **ADC Model** component interfaces with ADIsimADC to simulate ADC performance. The model interface provides translation from the analog domain to the digital domain. To access **ADC Model Settings** click **Settings**.

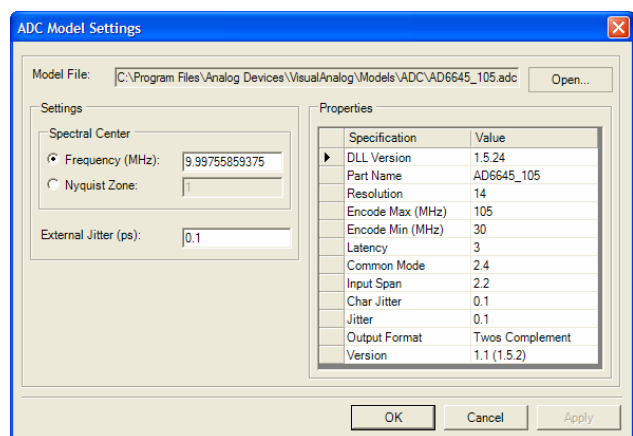
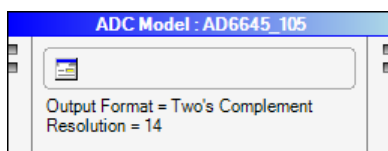


Figure 48. ADC Model and Settings Form

The model interface allows selection of the desired model, setting of the analog input range (the actual analog frequency is detected or may be overridden by selecting the Nyquist zone), and setting any external clock jitter—the default is that used during characterization by Analog Devices. Internal device jitter is automatically included and cannot be changed. More information about ADIsimADC is available at [www.analog.com/adisimadc](http://www.analog.com/adisimadc).

When using the **ADC Model** component, the two input terminals on the left represent the input waveform on the top and a tone list on the bottom. When connecting to a VisualAnalog **Tone Generator** (see the Tone Generator section), the input terminals of the ADIsimADC™ model map directly to the output terminals of the **Tone Generator**. The **ADC Model** component uses the tone list input to set the spectral center frequency. When connecting the input to other components, you can leave the lower input disconnected, and enter the spectral center manually.

An **Input Formatter** must always follow the **ADC Model Component** if you want to perform processing within the canvas. The only exception to this rule is **Logic Analysis**. The **Input Formatter** takes the resolution of the ADC into account when moving data into the environment of VisualAnalog. For more information on setting up the **Input Formatter**, see the Input Formatter section.

All Analog Devices models are LSB-justified and, therefore, the **Alignment** check box within the **Input Formatter** should be set to the native resolution of the model. The resolution on the **Input Formatter** should be set to the native resolution of the model as well. Note that the alignment setting is different when using the **Input Formatter** with the **ADC Data Capture** component. The number format should be set in accordance with the data format of the **ADC model**.

## COMPONENTS PROCESSES

These functional blocks provide basic numeric processing associated with converter testing. These blocks can be cascaded to create more complex evaluation processes.

### ARRAY MATH

The **Array Math** component performs array arithmetic on two or more inputs. Arrays may be multiplied or added and must be of the same size.

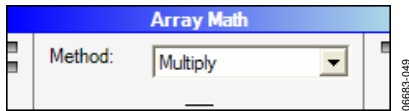


Figure 49. Array Math

### AVERAGE

The **Average** component computes the spectral average from the provided data. The indicator on the component shows how many averages have occurred out of the total number of averages. To set the total number of averages, click the **Settings** button. To reset the average series, click the **Reset** button.

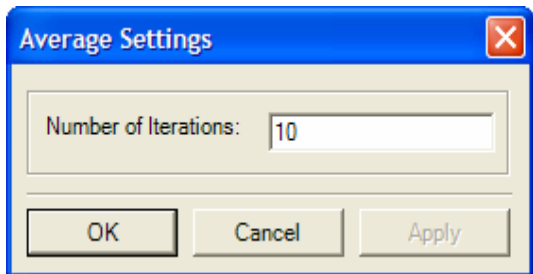
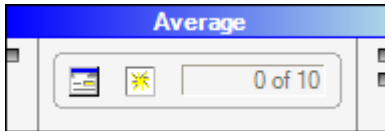


Figure 50. Average and Settings Form

The **Average** component is most efficient when running a continuous update. A running average occurs and the average continues until either the user stops the canvas or the number of runs equals the terminal count. In the second case, **Average** repeats another series of averages unless a stop block halts the canvas programmatically.

The **Average** component has two outputs. The top output is the running average FFT data. The second output is a numeric value that indicates when the average reaches its terminal count by outputting a nonzero value.

The **Stop** component (see the Stop section) can use the terminal value from the **Average** component to halt **Continuous Update**. In this case, when the **Stop** component receives a non-zero value and halts the canvas, the last update presents the final average.

Other components can use the terminal value as an indicator to perform processing only on a nonzero value. Consult the **Graph** component for more information on how to use it with this terminal value.

### BIT PROCESSOR

**Bit Processor** either flips (LSB to MSB or vice versa) or inverts the data set. When using VisualAnalog processed data, the **Output Formatter** should precede this component to format the data in a particular numeric format and resolution. In addition to the process selection, the **Bit Resolution** should be equal to the desired precision.

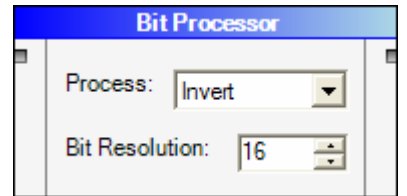


Figure 51. Bit Processor

### BIT SHIFTER

The **Bit Shifter** component is used to shift the data set bitwise up or down by the specified number of positions. This has the effect of multiplying or dividing by a power of 2. The **Bit Resolution** and **Shift** amount must also be set as desired. When using VisualAnalog processed data, the **Output Formatter** should precede this component to format the data in a particular numeric format and resolution.

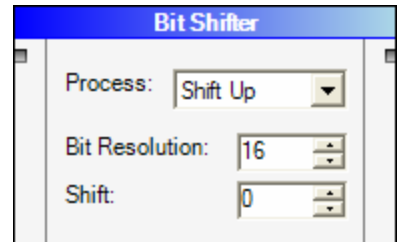


Figure 52. Bit Shifter

### COMMENT

The **Comment** component allows the user to display comments on the canvas. This is useful to document signal flow, options, or other operations. The **Comment** component serves no computational purposes. To edit the **Comment**, click the ellipses (...) and enter the desired text. You can resize the form to fit the context of your comment. It should be noted that the frame of the comment is invisible when not selected, showing only the text. To locate the frame, click the text to highlight the frame.

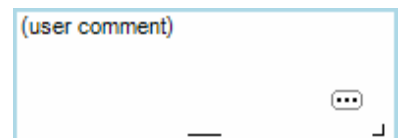


Figure 53. Comment

### COMPLEX WAVEFORM MERGER

The **Complex Waveform Merger** component merges two real waveforms into a complex interleaved waveform. The real input is on the top and the quadrature input is on the bottom.

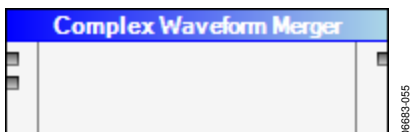


Figure 54. Complex Waveform Merger

### COMPLEX WAVEFORM SPLITTER

**Complex Waveform Splitter** splits a complex waveform into two real waveforms. The real output is on the top and the quadrature output is on the bottom.

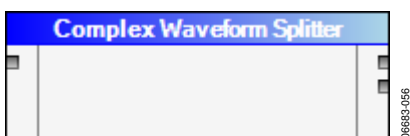


Figure 55. Complex Waveform Splitter

### DATA ROUTER

The **Data Router** component routes a single input to multiple destinations. This works for all data types. When the two output terminals are used, the component adds new outputs automatically. You can resize the component to provide room for more outputs.

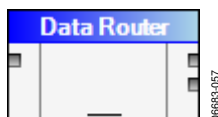


Figure 56. Data Router

### FFT

The **FFT** component translates real or complex data between the time domain and the frequency domain. This component works best with powers of two in size. It can, however, work with nonpowers of two as well.



Figure 57. FFT

### FFT ANALYSIS

The **FFT Analysis** component performs a numerical analysis on FFT input data to the user's specification, and outputs analysis data. By default, the analysis includes normal analysis for ADCs and DACs. The component can tailor to any specific application as discussed next. To configure the analysis for your application, click **Settings**.

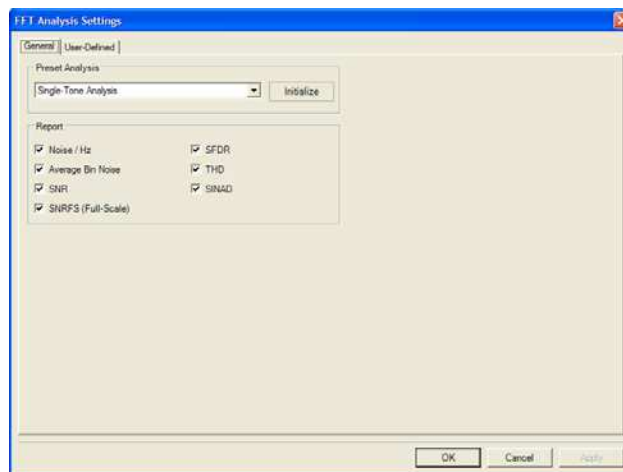
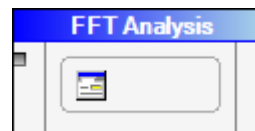


Figure 58. FFT Analysis Settings Form, General Tab

To access **FFT Analysis** settings click **Settings**.

Use the **General** tab to revert to a preset FFT analysis type. Currently the preset analysis list consists of **Single-Tone Analysis**, **Two-Tone Analysis**, and **Basic DAC Analysis**.

To use or initialize to a preset analysis, select the type in the drop-down box and click **Initialize**. Exit the form and keep the default settings or make changes that suit your needs.

In the **Report** frame, you can enable or disable commonly used calculations on FFT data.

You can use the **User-Defined** tab (see Figure 60) to adjust and customize particular calculations done on the FFT data. VisualAnalog supports almost any FFT analysis.

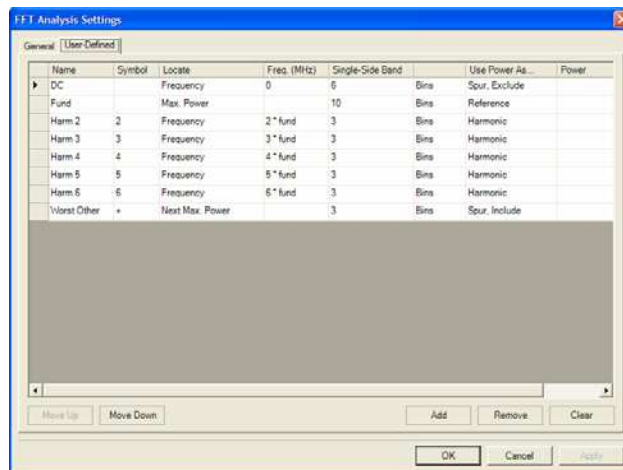


Figure 59. FFT Analysis Settings Form, User-Defined Tab