



# GC Studio

SLWU025 – February 2006

## *User's Manual*

## **IMPORTANT NOTICE**

THE PRODUCTS ORDERED HEREUNDER ARE EXPERIMENTAL, DEVELOPMENTAL OR PROTOTYPE PRODUCTS. THE PRODUCT SPECIFICATIONS ARE PRELIMINARY & SUBJECT TO CHANGE WITHOUT NOTICE. TI MAKES NO WARRANTY, EITHER EXPRESSED, IMPLIED OR STATUTORY, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A SPECIFIC PURPOSE, AS TO THESE PRODUCTS AND TI DOES NOT REPRESENT THAT A FINAL PRODUCTION VERSION WILL BE SOLD BY TI.

## Contents

Introduction.....	4
Background.....	4
Scope.....	4
Release Notes (v 2.0) .....	4
Release Notes (v 1.03) .....	5
The Main IDE .....	6
Layout (Numbered Diagram).....	6
The Diagram Browser and Property Grid.....	7
Concept .....	7
Usage.....	8
Standard Symbols .....	8
Property Types.....	8
Projects.....	11
Concept .....	11
Contents/File Format .....	11
Creating a new project .....	11
Opening an already created project.....	13
Editing Properties.....	13
Displaying Data .....	13
Graphs.....	14
Adding Data to a Graph .....	14
Graphing Functions.....	15
Formatting Options.....	15
Zoom/Formatting Controls .....	16
Metrics .....	16
Adding Metrics .....	16
Evaluation Board Setup .....	17
Scripting Language .....	18
Invoking the Script Interpreter.....	18

# Introduction

## ***Background***

Because the GC line of TI parts are complex devices and must be programmed using a microprocessor interface, an Evaluation Module (EVM) has been developed to demonstrate and test their capabilities. The GC Studio program controls that evaluation board and provides an intuitive user interface for programming the GC parts.

A DSP device can be thought of as a Multiple Input, Multiple Output Transfer Function (MIMO Transfer Function). It expects a stream of stimulus data and then produces a stream of response data. The primary use of the GC101 EVM is to program the Device Under Test (DUT), provide a stream of stimulus information, and capture the response information for processing by the computer. The GC Studio software controls the GC101 EVM and provides a stimulus image as well as retrieving and processing the response image.

## ***Scope***

This document discusses the setup and use of the GC Studio software as well as the basic functioning of the GC101 EVM. A useful supplement to this manual is the “SCR GC101 Language Reference” that defines and explains the scripting language used.

Specific plug-ins are explained in documents that are grouped as “Studio Reference” documents. These documents are included with releases of GC Studio.

## ***Release Notes (v 2.0)***

Version 2.0 of GC Studio incorporates a number of significant changes, not the least of which is support for the GC5016. In addition to this new plugin, this version of GC Studio also includes the following major new features.

- The old VCR controls have been replaced with compiler style build, load, and capture buttons. This allows more flexibility when using external sources or sinks (ADCs, DACs, etc)
- Each of the plugins now allows the user to view each of the four sync fields, in order to eliminate the need to look at the debug files for lining up data. Additionally, there is now a menu entry under Tools that will allow the user to export the device programming file, eliminating the need to mess with the debug settings to retrieve the device programming.
- All of you who have computers without parallel ports will appreciate the presence of a connect/disconnect button. By default, the system will try to connect to the parallel port on startup. However, if it fails, it will just go into offline mode. This preference can be edited from the **Tools|Options** screen
- Full motherboard controls are now available that will allow people to configure the system any way they would like. Additionally, changing either of the memories from a source to a sink or vice versa is now a matter of just clicking "Source" or "Capture" on the appropriate page in the EVM Parameters screen.
- A "Pattern Generator / Logic Analyzer" plugin has been added that allows the ports to be setup manually, rather than inside the plugin, so that the GC101 can be

- used to source data out the output port, read it in the input port, and perform processing.
- There is now a stream cropping tool under **Tools|Stream Cropping Tool** that can be used to evaluate only the important sections of TDD data. - This tool can also be used to only process parts of the output stream that are useful.
  - The General Error Vector Magnitude measurement has been fixed - the delay was off, producing a measurement of ~50%EVM from certain cases of the GC1115. We all know the 1115 is far better than those numbers (9%EVM). Now we can prove it.

### ***Release Notes (v 1.03)***

I am very excited about this new release (1.03) of GC Studio. In addition to a variety of new DUTs that are supported by the system, we have made numerous enhancements. Some of the general features that you will notice include the following.

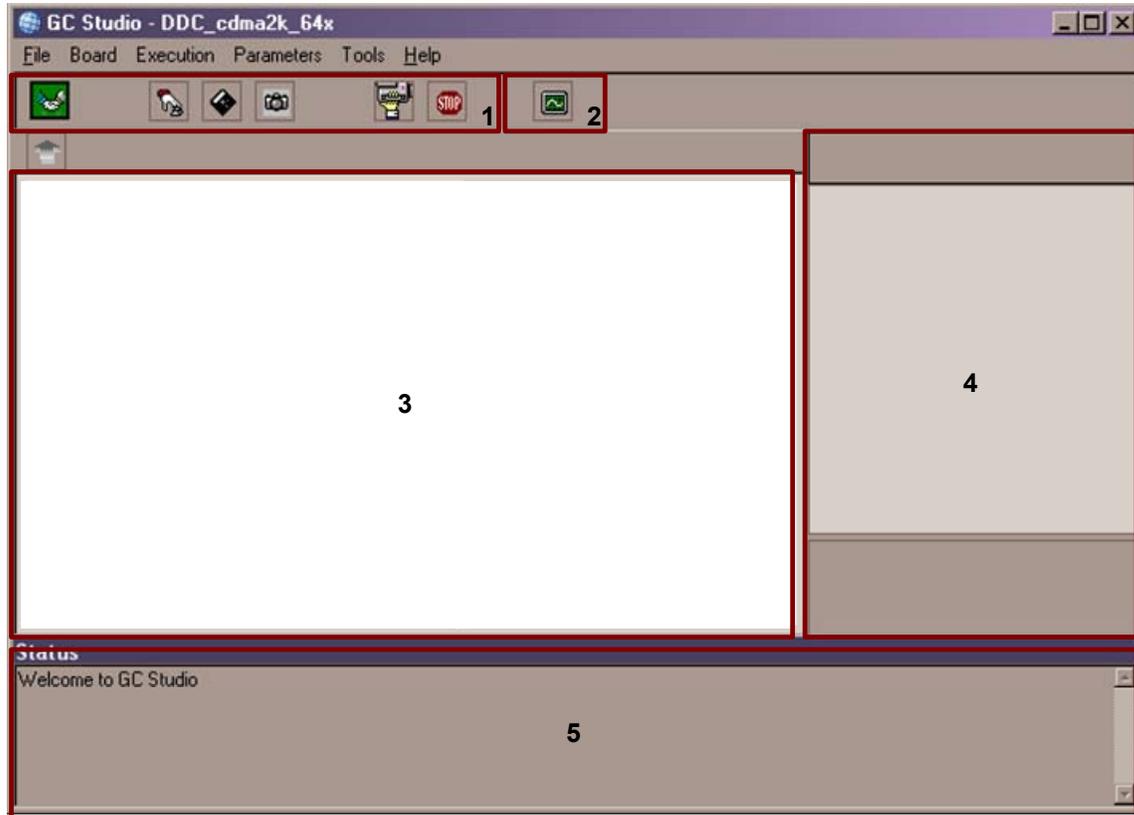
- GC5018 / AFE8406 Support
- GC1115 Support
- Performance Metrics that allow the efficient evaluation of output data
- A new graphing system that is easier to use and works orders of magnitude faster
- Enhanced stability throughout the entire system
- Increased flexibility in the parameter entry and modification structures
- A simplified interface that presents a single main screen, rather than a multiple-document interface

A number of people have worked very hard in creating this release, and I hope that it will meet with the field success that it deserves. Because this is an “experimental development” product, we are sure that users will encounter bugs. However, we have worked hard to make sure that the number of bugs is minimized and we hope that our customers will enjoy using this product. Please let me know what feedback, both positive and negative, you have with this product – [krbreton@ti.com](mailto:krbreton@ti.com)

*-Kyle R. Breton*

# The Main IDE

## Layout (Numbered Diagram)



### 1. Control Toolbar

These replace the “VCR controls” from the previous version. A description of these six buttons, in left-to-right order, follows below.

- Connect/Disconnect: Allows the user connect to or disconnect from the EVM board. This feature is particularly useful for those users with laptops who may not have a parallel port or don't have the Warp Nine drivers installed. If one disconnects from the EVM, then he or she will still be able to create, load, and save projects but won't be able to run them.
- Build: “Compiles” the current project, by creating the command stream (from the block diagram) that will eventually get sent down to the DUT, configuring the format of the stimulus and response memory streams, and creating the stimulus image. Building a project also entails loading it (described next).
- Load: If the current project built without any errors (an example of error would be if the command stream could not be formed due to an ambiguity in the block diagram), then this button programs the EVM, loads the input memory, and programs the DUT.
- Capture Once: Run through the main device loop a single time.

- Capture Continuously: This will setup the chip and run memory through it continuously, so that the user can see the effects of any parameter changes in almost real time.
- Stop: Discontinue the main device loop, and then reset the loop so that the next time build or load is pressed, the loop will start over.

## 2. Create New Graph



Sinewave.ico

This creates a new graph screen that can be used to display various output information. Any stimulus or response port can be plotted, and various measurements can be taken. Please see the section entitled “Displaying Data” for more information.

## 3. Diagram Browser

The diagram browser helps to visualize the setup of the part. It works with the property window to display the different parameters for the parts of the chip.

## 4. Property Window

The property window displays the properties for the currently selected block in the diagram browser. Properties may be edited by the user, and after the main loop has been run, the properties will update.

## 5. Status Window

When the control loop or other steps have status information to report, it appears in this window. The window automatically scrolls to the most recent status message, and also provides a scrollback history.

## ***The Diagram Browser and Property Grid***

The diagram browser is designed to help the user visualize the status of the part. While the diagram itself and the properties that can be changed depend entirely on the plug-in that the current project works with, the usage of the diagram browser and the types of properties that can be changed remains constant, to provide a common user interface.

### **Concept**

For wide flexibility in assessing very complex devices, access to thousands of parameters may be needed. The GC Studio “Diagram Browser” provides access to these parameters in an intuitive fashion. The GC Studio displays, in graphical form, a functional and interactive block diagram of the DUT. The right side of the diagram browser window is the list of the parameters that may be changed for the currently selected block. This allows the user to see a picture very similar to all of the information about the DUT presented in the datasheet. Thus, the user can step through both the datasheet and the interactive diagram at the same time for a clearer understanding of the part.

## Usage

A single-click on any block in the diagram browser will highlight that block by overlaying a shade of gray. The properties that pertain to that block will then appear in the property window to the right.

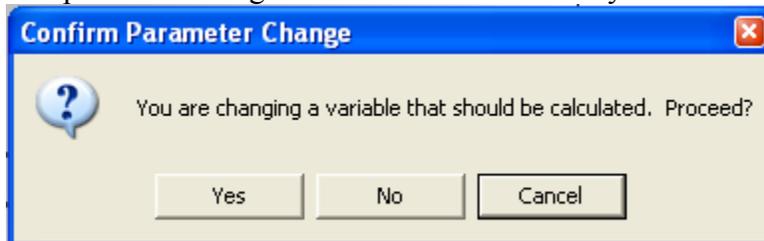
A double-click on a block in the diagram browser will “zoom” into that block if there is more detail to display.



In order to “zoom” back out of that block, press the up arrow in the top left corner of the diagram browser.

A single-click on any of the properties that is currently visible will display a description of that property in the box at the bottom of the property grid.

If a property is changed that is considered to be a “calculated” property, then the system will present a dialog box to determine whether you wish to override the calculated value.



Selecting Yes will change the value of the parameter.

Selecting Cancel will leave the parameter in the same state it was in previously.

Selecting No will change the value of the parameter back to default and will set the parameter as not modified, so that it will be overwritten by the calculation step in the main loop.

## Standard Symbols



This diagram object represents a port. Ports are points where data can either be loaded from a file, or read back out to a file. A green port represents a port that will be loaded into, or retrieved from the EVM. A red port represents a port that cannot be loaded into or retrieved from the EVM because of a configuration setting.

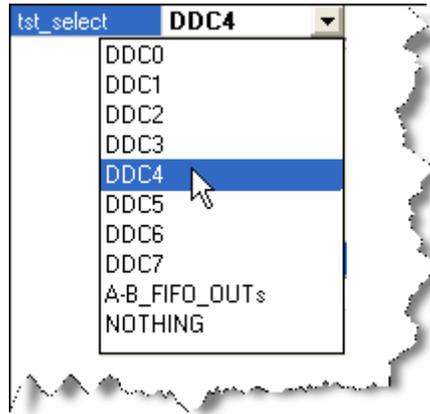
Red blocks represent blocks that have been disabled. Green blocks represent blocks that can be disabled, but are currently enabled. Channels are an excellent example of this property. The channels on DUCs and DDCs will turn green or red, depending on whether the channel is enabled or not.

## Property Types

There are a number of different properties that can be set for any of the different blocks in a diagram. Most of these are self explanatory and will only be mentioned here. The rest require some amount of explanation.

- **Bit Field** – These parameters show up as integers, and the bit-width should be specified in the description at the bottom of the property grid. If a number is entered that is larger than the bit width allows, it is automatically truncated and the actual value will appear in the property grid, to avoid confusion regarding the actual programmed value.

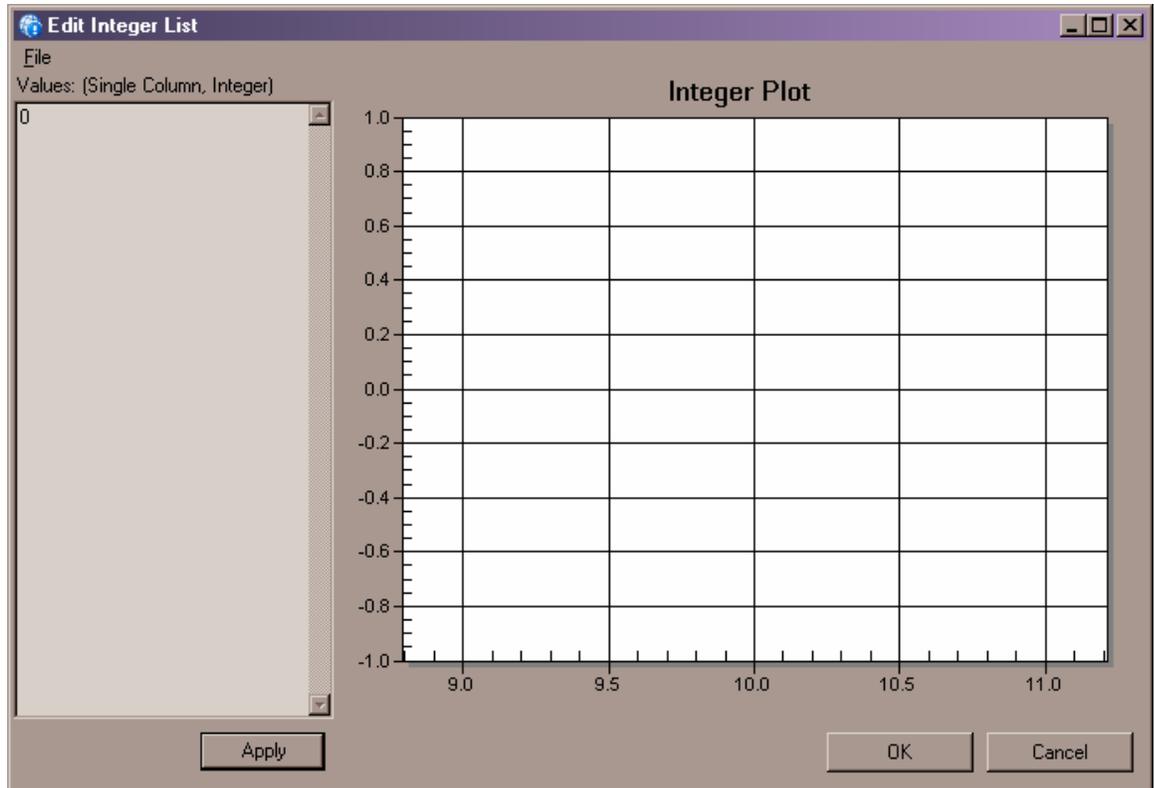
- Boolean – These parameters allow you to select either “True” or “False” from a drop-down menu.
- Decimal – These parameters allow a decimal number, including a decimal point to be input.
- Enumeration – These parameters allow you to select a single value from a list of pre-defined options.



- Integer List – This parameter allows you to define a list of integers, which may be used as taps for FIR filters, RAM locations, or other values. The easiest way to edit these filters is by clicking the ellipsis at the side of the property.

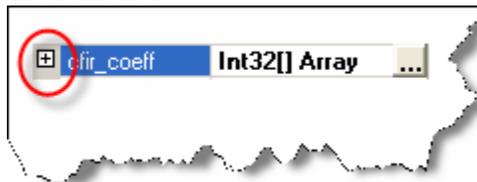


The following screen will then appear:



Values can be entered in single column format into the text box on the left hand side. They can also be copied-and-pasted from another application (i.e. excel) or tab-delimited filter tap files can be imported via the **File|Load** menu selection. Clicking the “Apply” button will cause the integers to be plotted on the graph on the right hand side. This can be useful for plotting filter taps. Clicking “OK” will apply the integer list to the property.

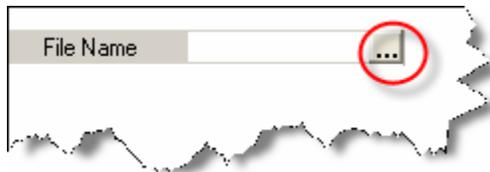
Alternately, integer list properties can be edited in place by clicking on the “+” symbol to the left of the list.



While the length of the integer list cannot be changed in this method, individual values can be modified.

cfir_coeff	Int32[] Array
[0]	1
[1]	2
[2]	3
[3]	4
[4]	5
[5]	6

- Input and Output Files Specify a location for a file to save to or read from. While the path to the file can be manually entered, it is suggested to click the ellipsis and browse for the file using the standard windows file browser dialog.



## Projects

### **Concept**

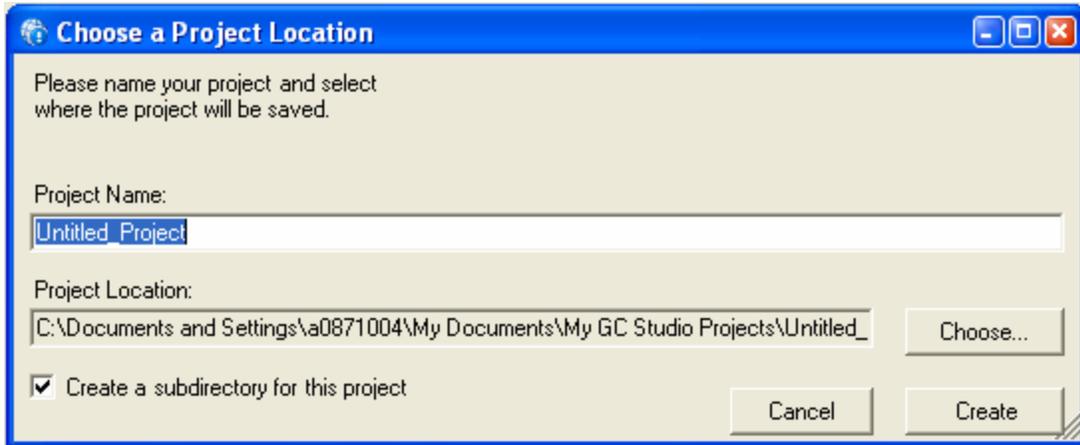
In general, a DSP chip can be thought of as a black box that is given one or more stimuli and produces one or more responses. The GC Studio software project contains information about where to get the stimulus information, how to program the DUT, and what response information should be shown. A project is specific to a plug-in. All of the settings in the project are stored, along with the name of the chip that the project is designed to control. When a new project is opened, the correct plug-in will be automatically loaded.

### **Contents/File Format**

The project file is stored in an XML format and is intended to be edited only by the program. This format provides a minimal readability level, but tools to modify the project can be easily created if necessary. The project file is broken into multiple major portions. The <evmparams> tag encases all of the properties that are specific to the setup of the Motherboard. A <block> tag encases all of the settings that pertain to the specific chip that is being programmed. Finally, a <graphlist> tag contains information on all of the graph output screens that should be shown when the project is opened.

### **Creating a new project**

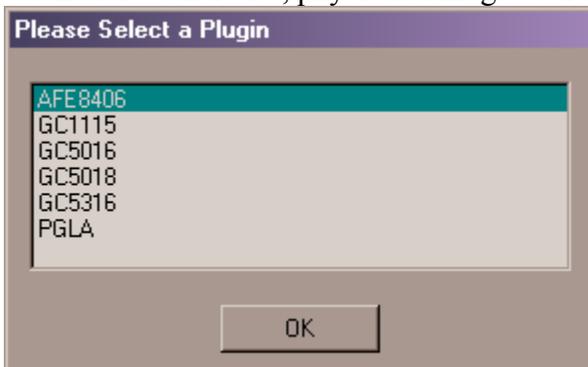
In order to create a new project, choose “New Project” from the “File” Menu. The first screen will ask you to decide where you would like to place the project.



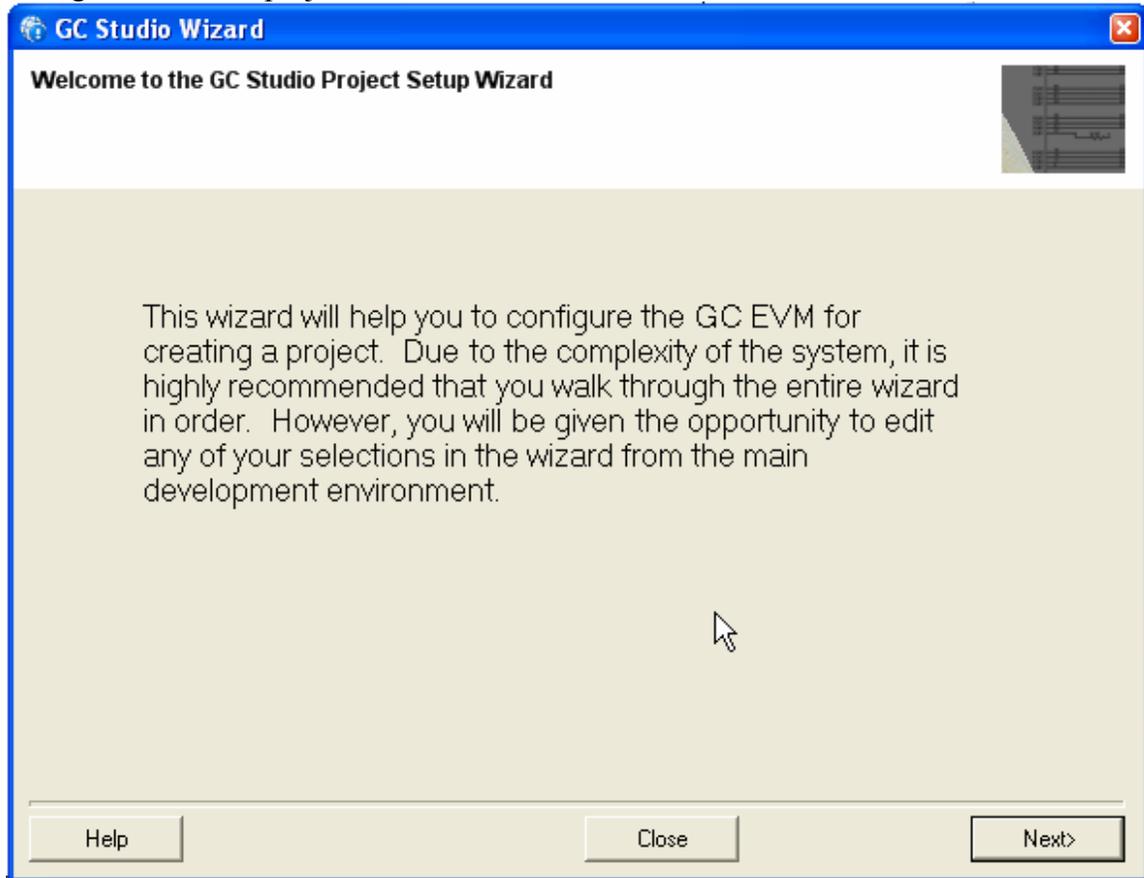
Enter the name of your new project in the “Project Name:” textbox. If the “Create a subdirectory for this project” box is selected, then a subdirectory will be created with the project name. While not absolutely necessary, for portability reasons, it is suggested that project names not have spaces in them.

Selecting “Choose...” will bring up a folder browser dialog where the user can select a folder to place the new project in.

After entering a name and choosing a location, click the “Create” button to continue. If the user has multiple plug-ins installed, the system will now ask which plug-in should be used. Be sure to select the plug-in that corresponds with the card that is inserted into the GC EVM. Otherwise, physical damage could occur to the board.



Clicking OK will create the project and open the setup wizard, where the user can configure the initial project.



Click "Next" on the wizard screens to move through the wizard.

After the wizard is complete, you will be in the main window, where you can browse through the diagram of the chip to further specify settings.

### ***Opening an already created project***

Previously created projects can be opened by selecting "Open Project..." from the "File" menu. The appropriate plug-in will automatically be loaded, and the user will automatically be positioned on the main window. A project description dialog automatically appears whenever a project is loaded from disk.

### ***Editing Properties***

By selecting blocks within the main diagram and editing the properties in the right hand property grid, the user can configure the chip in the manner they wish to see.

### **Displaying Data**

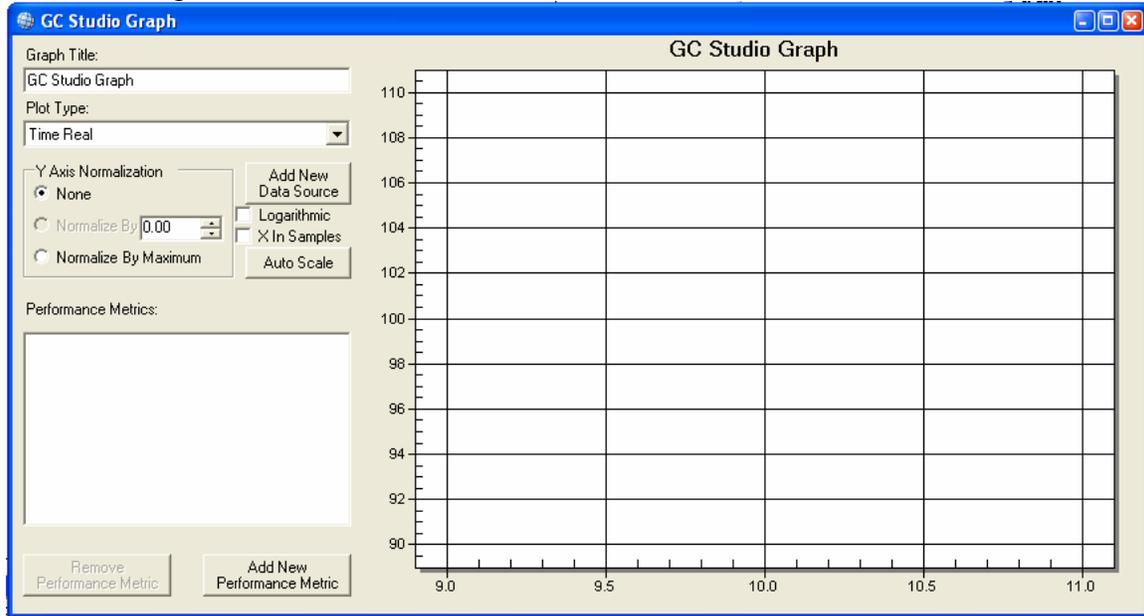
The GC Studio program is designed to allow for convenient viewing of the data the stimulus and response data in a project. For this purpose, the Graph screen is provided, which can plot sources of data, and can display various metrics regarding that data. In

the event that further analysis of the data is required, any output port can be tied to a file by selecting it and filling in a file name. Data will be output to this file in two column decimal integer format. The first column is the real part of each sample, and the second column is the imaginary part of each sample.

Graph windows can only be created while a project is open. In order to create a new graph window, select the “New Graph” icon from the toolbar:



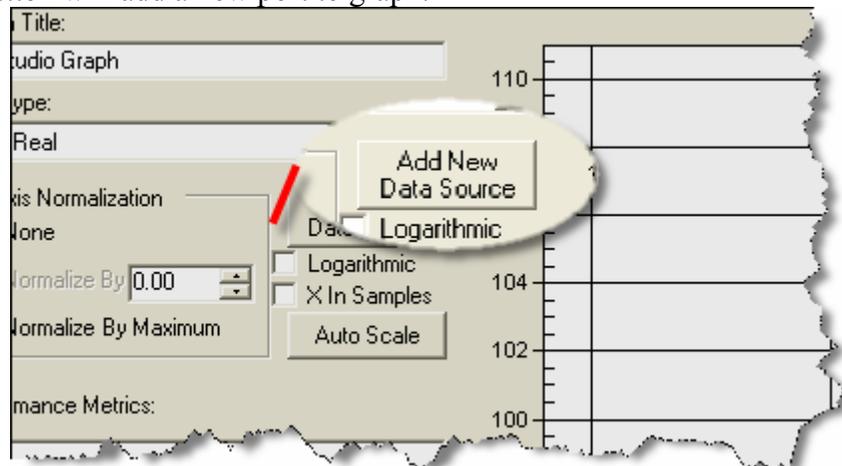
The main output window looks like so:



## Graphs

### Adding Data to a Graph

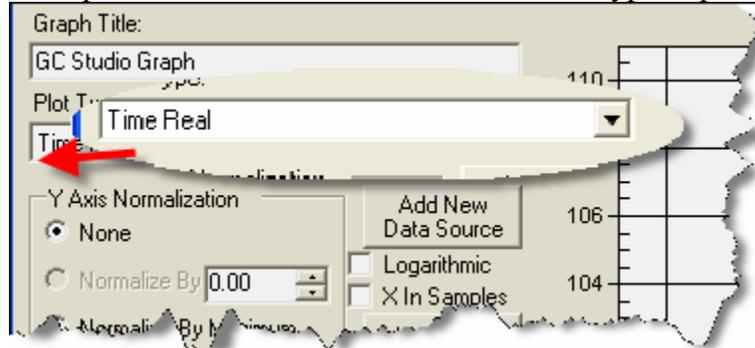
Any input or output port can be added to the graph. When the data for that port is updated, the graph will automatically redraw that data. Clicking the “Add New Data Source” button will add a new port to graph.



Select the appropriate data source from the resulting list. Standard Windows multiple selection using the Shift and Ctrl keys works in this list.

## Graphing Functions

The “Plot Type” drop-down box allows the user to select the type of plot to be shown.



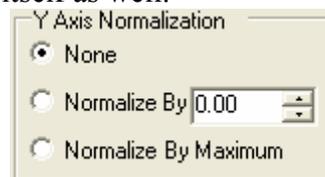
The available options are:

- Time Real: Presents a plot of real portion of the signal, in time domain.
- Time Imaginary: Presents a plot of the imaginary portion of the signal in time domain.
- Time Polar: Presents a plot with the X axis being the real part of the signal and the Y axis being the imaginary part of the signal.
- Spectral Magnitude: Presents a plot of the magnitude of the DFT of the signal.
- Spectral Phase: Presents a plot of the phase of the DFT of the signal.
- CCDF: Presents a plot of the cumulative complementary distribution function, which shows the number of samples at each power. Please note that the Y axis of this function is presented in logarithmic form, without the “Logarithmic” checkbox being pushed. Pushing the “Logarithmic” checkbox results in logically useless data.

## Formatting Options

In addition to choosing the type of plot that is being presented, the user may also choose different methods for representing the data itself.

The Y Axis Normalization is most often used in conjunction with the “Logarithmic” check box, but may be used by itself as well.



A normalization setting of “None” presents the data exactly as it is stored in memory. A normalization setting of “Normalize By” will divide all samples by the value in the decimal box. Please note that a setting of 0.00 in the decimal box is the same as setting the normalization to “None”

A normalization setting of “Normalize By Maximum” will divide each sample by the maximum sample in that data stream. Therefore, a “Normalize By Maximum” setting, in combination with the “Logarithmic” check box provides a “Corrected” dB Scale.

The other options provided for formatting data are in the form of check boxes.



The “Logarithmic” setting graphs  $20 \cdot \log_{10}(\text{data})$  instead of the original data. The “X In Samples” button normalizes the X Axis so that, instead of being set to frequency, or time, the X Axis is the sample count.

## Zoom/Formatting Controls

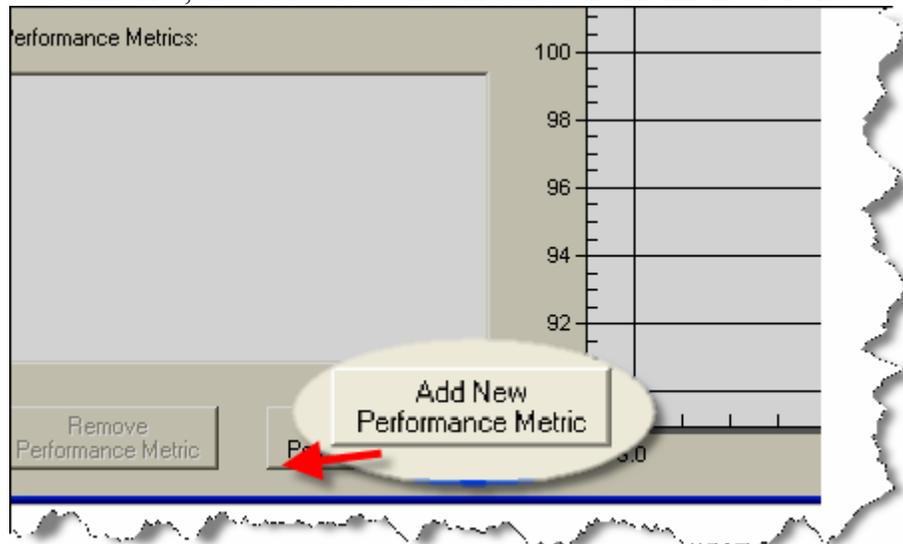
In order to zoom in on any region of the graph, simply draw a box around that region of the graph, using the mouse and the left-hand button. The graph will be re-drawn showing only the selected image. Additionally, scrollbars will appear on the bottom and right hand sides of the graph, allowing the user to pan through the zoomed in image. The graph will remain at the selected level of zoom until the user selects the “Auto Scale” button to the left of the graph, at which point the graph will zoom out to show the entire signal. Additionally, when new data comes in, the graph will automatically resize to show all of the data.

## Metrics

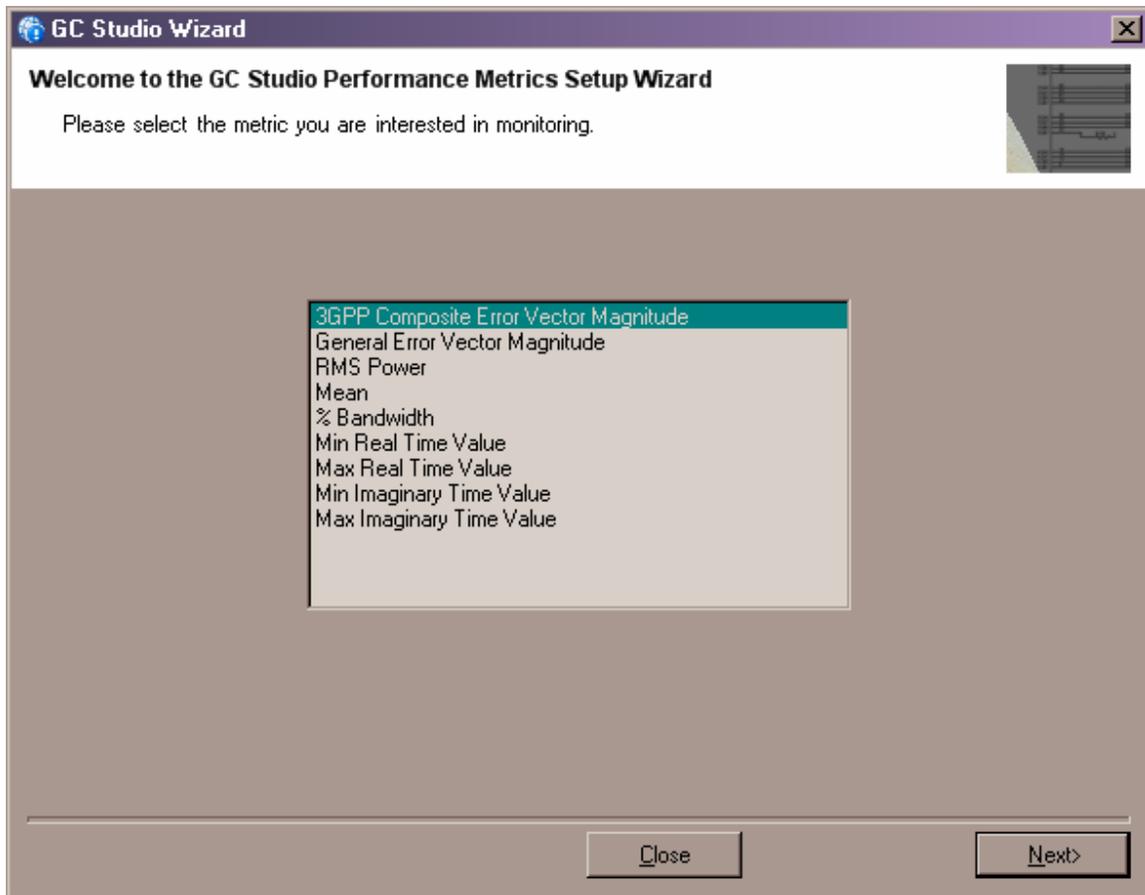
Metrics are provided to allow basic analysis of the data retrieved from or input into the DUT. All metrics will update whenever the data that they depend on is changed.

## Adding Metrics

In order to add metrics, the user selects the “Add New Performance Metric” button.



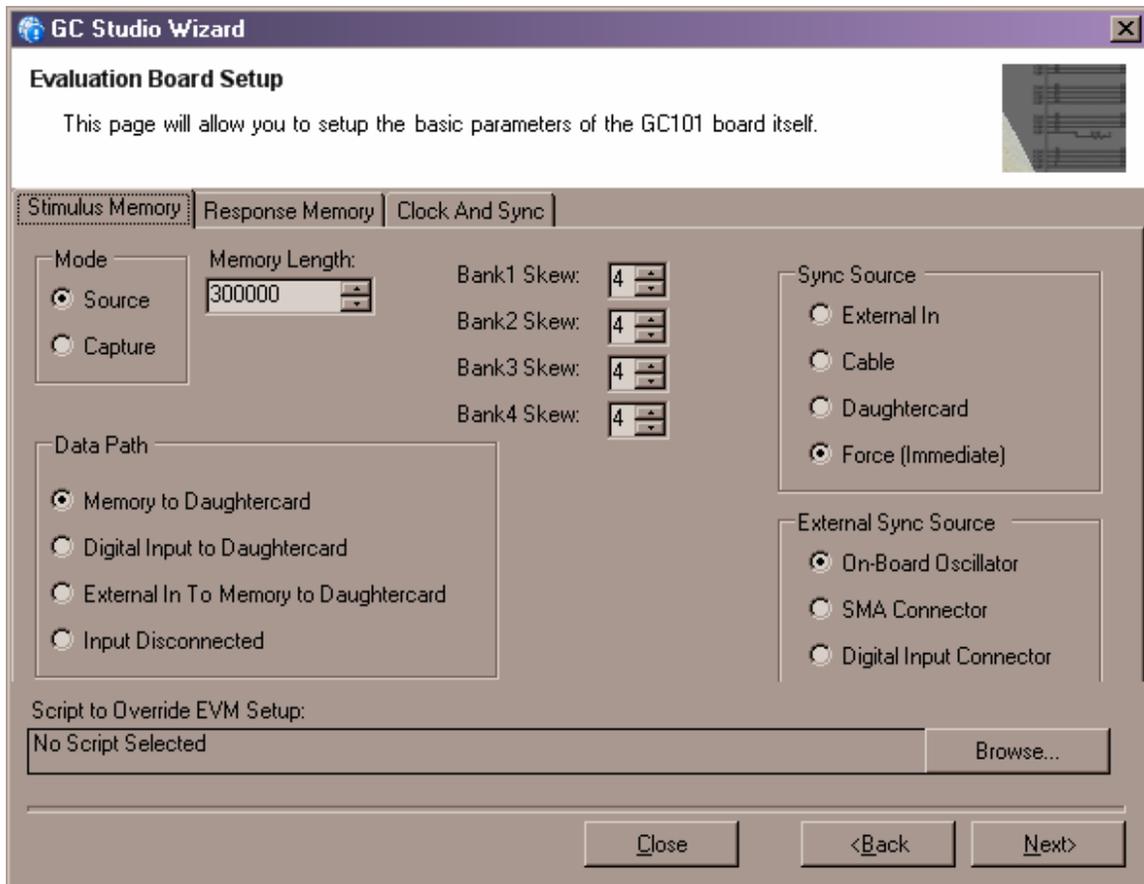
A wizard will present the available metrics, and then allow the user to select data sources and parameters to compute the metrics.



Select the requested metric, and click “Next>” The wizard will walk the user through all of the necessary sources of input. The list of available metrics within GC Studio are enumerated in the above screen-capture.

## Evaluation Board Setup

While creating a new project, the second wizard screen will ask about setting up the EVM. While the default settings should suffice for a majority of projects, these settings are explained in detail here.



Note that there are three tabs for stimulus, response, and clock and sync configuration.

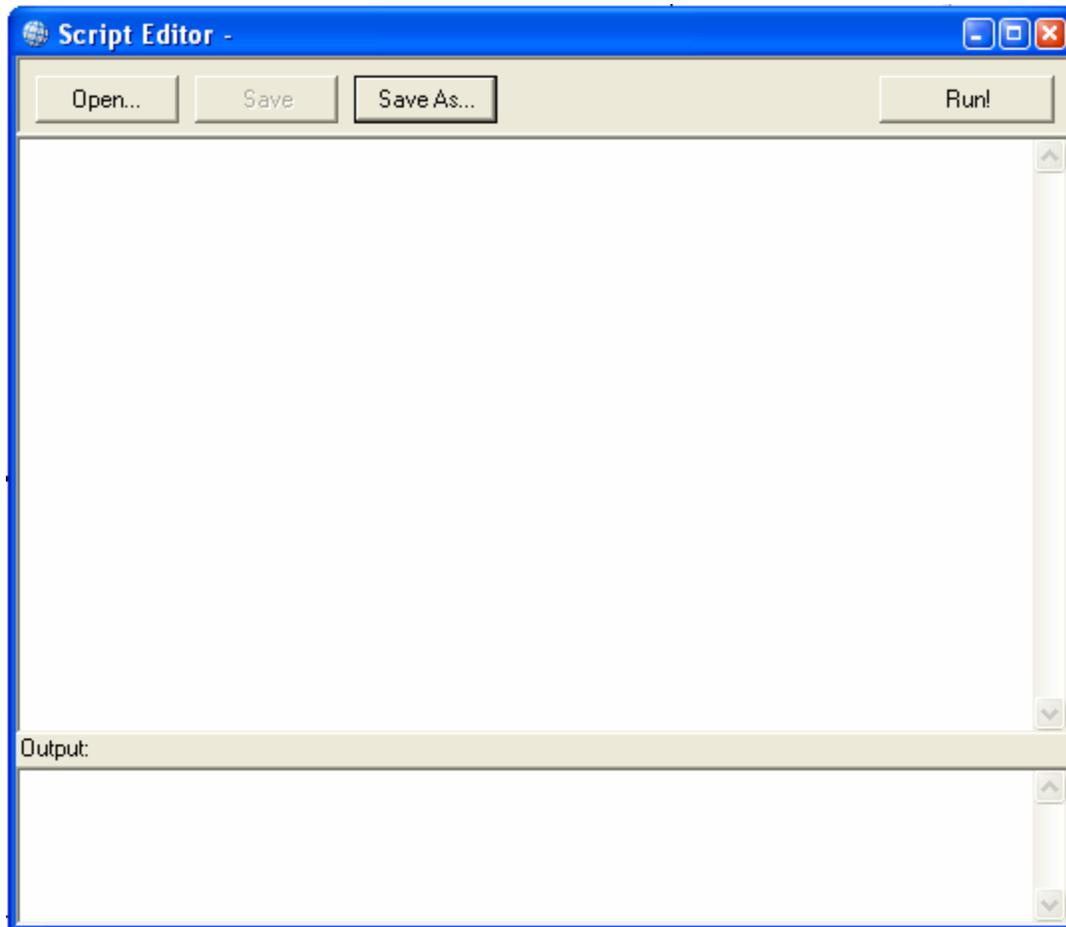
## Scripting Language

The GC Studio program provides an integrated means of executing SCR GC101 scripts. These scripts contain general purpose scripting commands and commands that setup and interact with the GC101 EVM and the DUT. For information on the layout and syntax of these scripts, the SCR GC101 Language Reference is available.

Use of the scripting language for projects that do not require the GC Studio interface is through the program "GCScripter.exe" supplied with the GC Studio program.

### *Invoking the Script Interpreter*

The script editor can be accessed by selecting "Script Editor" from the "Tools" menu.



The script editor has three main parts. The top toolbar allows the user to open or save the current script. The center portion is where the selected script is displayed and can be edited. The bottom portion of the screen displays the output from the script.

To run a script, use the “Run!” button in the upper right hand corner. When this button is clicked, the script is run through the interpreter and the results appear in the output section. If no filename has been specified, a “Save As...” dialog box will appear.