AWG Manual

Home

Welcome to the Euvis AWG Manual. Before doing anything else, please see the First Powerup Checklist section to ensure that you have a working AWG board.

This manual includes instructions for the hardware and software setup to use the AWG module with the provided graphical user interface. The user can design his own waveforms or use any of the build-in waveforms with user-configurable parameters in the GUI. Advanced functions, such as using the AWG as part of an embedded system, require the user to design his own application programming interface, guidelines for which are available in a separate manual.

Please use the menu on the left to navigate through the manual. If you have problems, please feel free to contact us. You can get our contact information by clicking on "Contact Us" in the menu to the left.

Quick Start

- First Powerup Checklist
- Simple Waveform Output
- Loading A Waveform
- Loading A Custom Waveform
- Firmware Update

Getting Started -

Thank you for choosing the Euvis Arbitrary Waveform Generator. To use all the functions of the AWG, you must install the software and drivers prior to connecting the AWG to your computer.

Software and Drivers

Before connecting the AWG module to your computer, you first must have installed the requisite software on your computer running Windows XP:

- both Microsoft .NET Framework 2.0 and Microsoft .NET Framework 2.0 SDK, and
- the Euvis AWG application and related drivers

Please follow the step-by-step instructions to install the software and drivers before connecting your computer to the AWG. The drivers are specialized to work with the AWG, so they must be installed correctly before connecting the AWG and the computer.

Hardware Connections

Next, connect the AWG module to a clock source through the CKIP SMA connector and to the power supply through the pluggable header. Connect a USB cable from the AWG to the computer. Also set up a small fan to blow air across the AWG module.

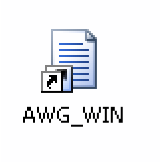
Your clock and power connections should be set up according to instructions detailing the power supply and clock source requirements. Set the clock source frequency to 2 GHz with 3dBm power. If your module includes an optional on-board clock module, then you do not need to connect an external clock source.

Power-Up

Before opening the graphical user interface (GUI) when you first power up, you must wait until you see the demo waveform output. If you open the GUI before you see the output, you will not be able to control the AWG. The demo waveform is burned into the ROM on the module so you will not be able to change it by yourself. If you would like another demo waveform when you first turn on the AWG, you can contact us to request the change.

Now that the demo waveform is running, this is a good time to check that the 1.8V voltage has not dropped too low due to resistance along the power wires between the module and your power source. The current varies with clock frequency, and therefore the voltage drop along your wires will increase at higher clock frequencies. At a clock rate of 2 GHz, the voltage measured at the power plug screw corresponding to 1.8V should be around 1.75V; if yours measures too low, please increase your 1.8V power source voltage slightly to compensate for the resistance of your wires. If your AWG module included an on-board power module, then you do not have to check the voltage.

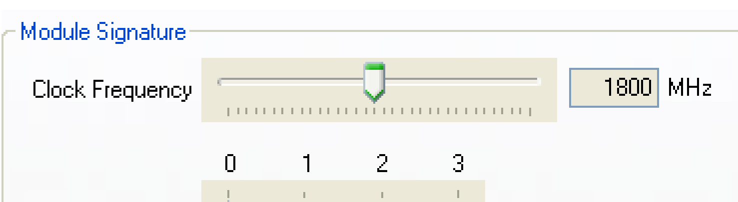
Launch the **AWG_WIN** application by double clicking the icon on the desktop. This program will allow you to download waveforms to the AWG module and will facilitate the configuration of waveform parameters and operation settings. When you open the program, the startup demo waveform will stop.



Changing the Clock Frequency (Signature File)

The first thing that you should do after you open the GUI application is to set the clock frequency. When operating the AWG at different clock frequencies, the settings may need to be changed for optimal performance, especially at higher clock speeds, when the timing and thresholds of data samples are less forgiving and need to be adjusted for different clock frequencies. These settings can be saved in the signature file. We have already provided you with a signature file in the main AWG directory. The signature file contains preset signatures for frequencies from 500 MHz to 2.5 GHz, and up to 4 GHz for the AWG452. Therefore, you most likely will not need to adjust much when you change frequencies. The only thing you might need to do is to click on the "ATE" checkbox to fine-tune the AWG.

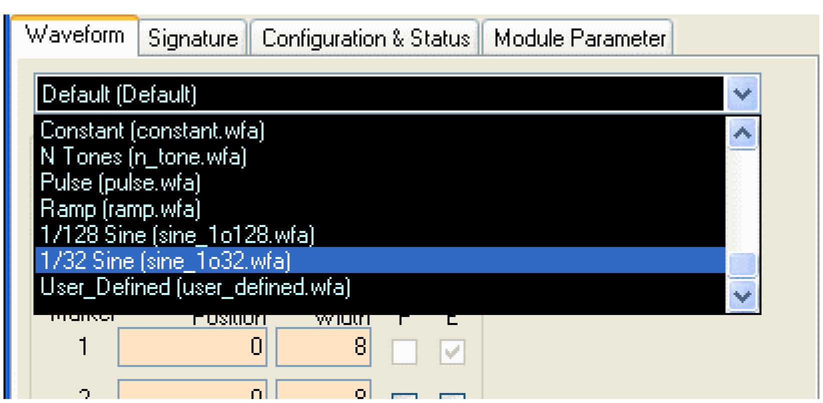
To change the clock frequency in the GUI, click on the Signature tab and slide the frequency bar to match the frequency of your input clock.



More details on the signature can be found on the signature page.

Waveforms

The AWG comes with four built-in waveforms: absolute sine wave, relative sine wave, ramp, and N-tones. To access these waveforms, click on the Waveform tab. In the Waveform panel, use the pull down menu to select the waveform named "1/128 Sine (sine_1o128.wfa)". This is a Sine wave at 1/128ths of the clock frequency. Optionally, you may adjust the parameters for this waveform. The computer calculates the newly parameterized data, although data on the AWG does not yet change.



Press the **Download** button (at the lower right corner of the waveform panel). The computer then downloads the parameterized data to the AWG memory so that the firmware generates the actual waveforms. You must press the Download button each time you finish changing the parameters, unless you have enabled Auto-Download in the Options menu.

Press the Restart button (in the right-click pop-up menu) as needed to restart the waveform.

For a brief description of other waveforms available in the graphical user interface, please see the Waveforms page.

Abort and Restart

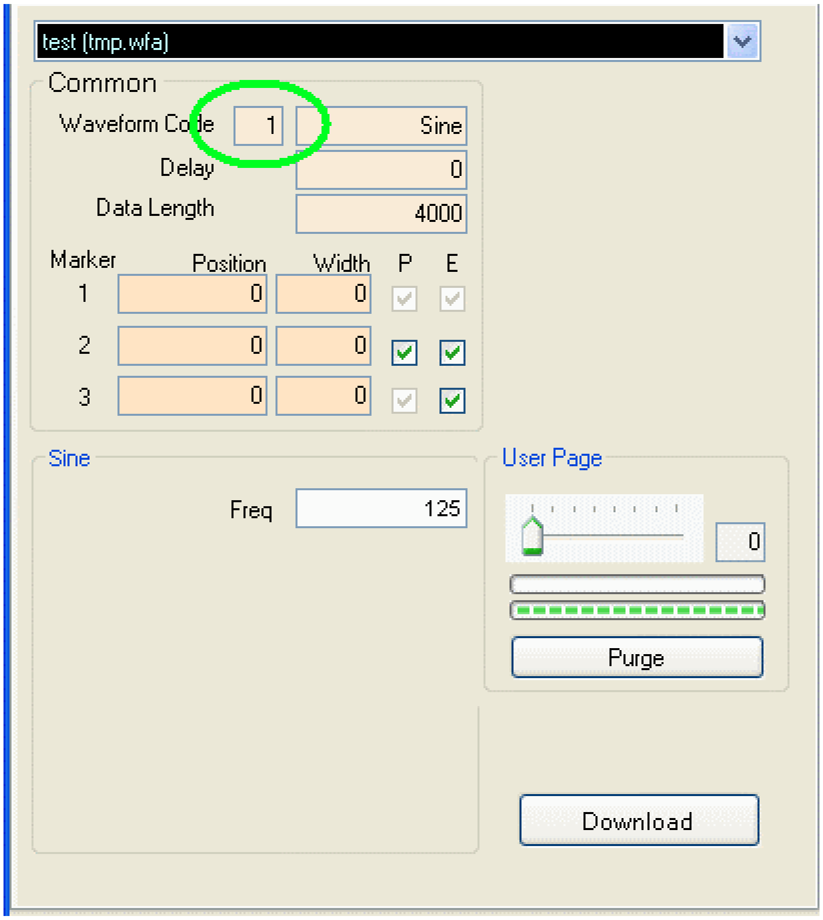
Clicking on the "Abort" button will stop output from the AWG.

Clicking on the "Restart" button will cause the AWG to start output. After downloading data to the AWG, click the Restart button.

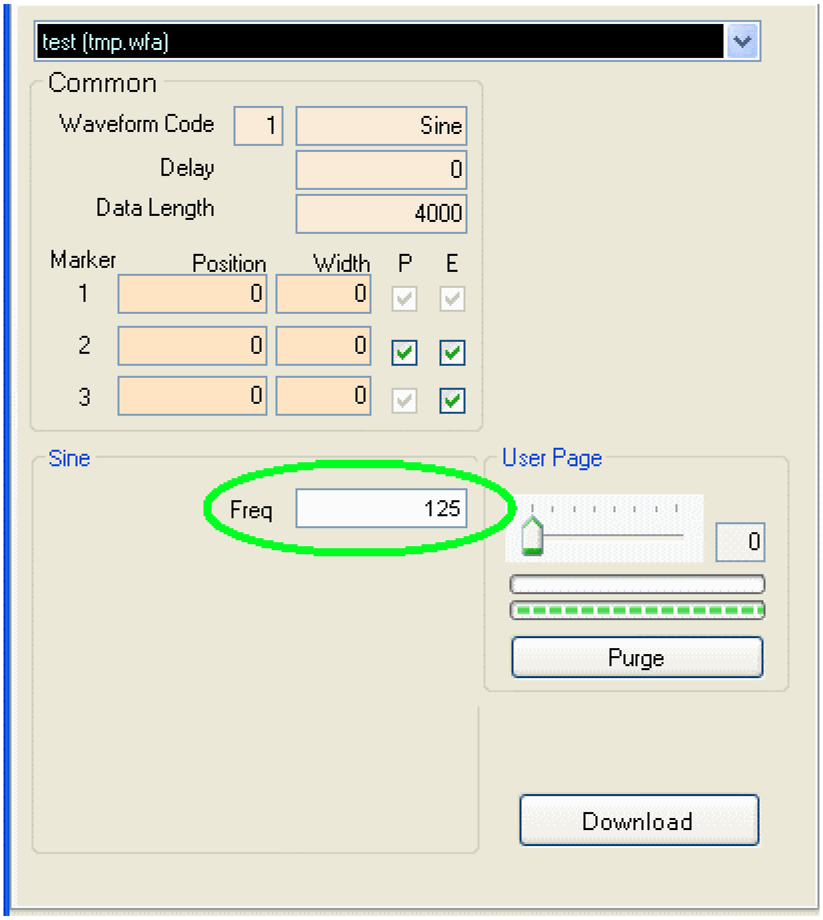
Example 1 : Absolute Sine Wave

This example will show you how to create an absolute frequency sine wave. Sine waves are a good way to test the RF performance of the AWG. You should have already set up all the hardware, software, clock, and power connections, and you should have launched the AWG_WIN.exe application, as described briefly in the Getting Started page.

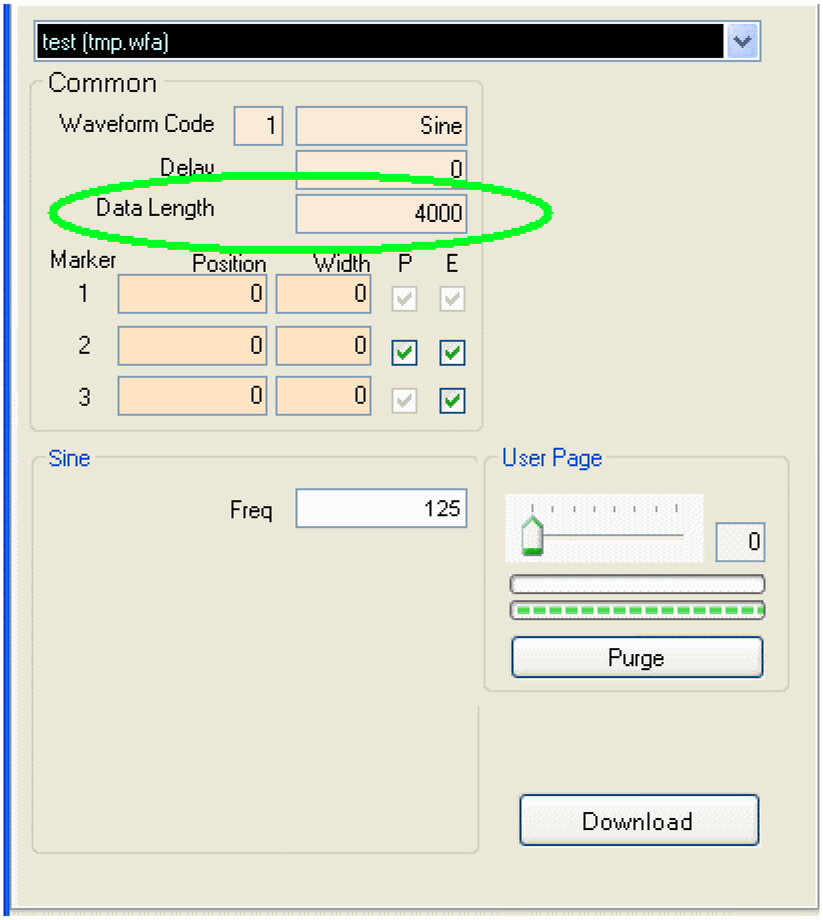
Click on the Waveform tab. In that panel, make sure that the Waveform Code is set to "1". This is the code for absolute frequency sine waves.



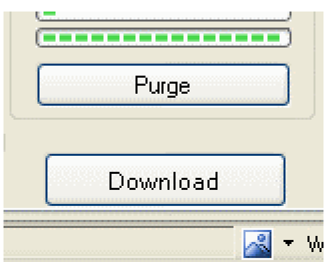
You should see that in the Waveform Type box it will say "Sine". On the bottom in the Freq box, enter the frequency that you would like the Sine wave to have. The frequency is specified in hertz so if you would like a 250 MHz sine wave you would enter "250000000".



You can also change the number of points that the waveform will have per cycle. To do this, change the Data Length to the number of points you want. Please note that the Data Length is specified in hexadecimal, and it should be a multiple of 20 (hex). For more information on Data Length, please see the Data Length page.



Press the Download button (at the lower right corner of the waveform panel). You must press the Download button each time you finish changing the parameters.

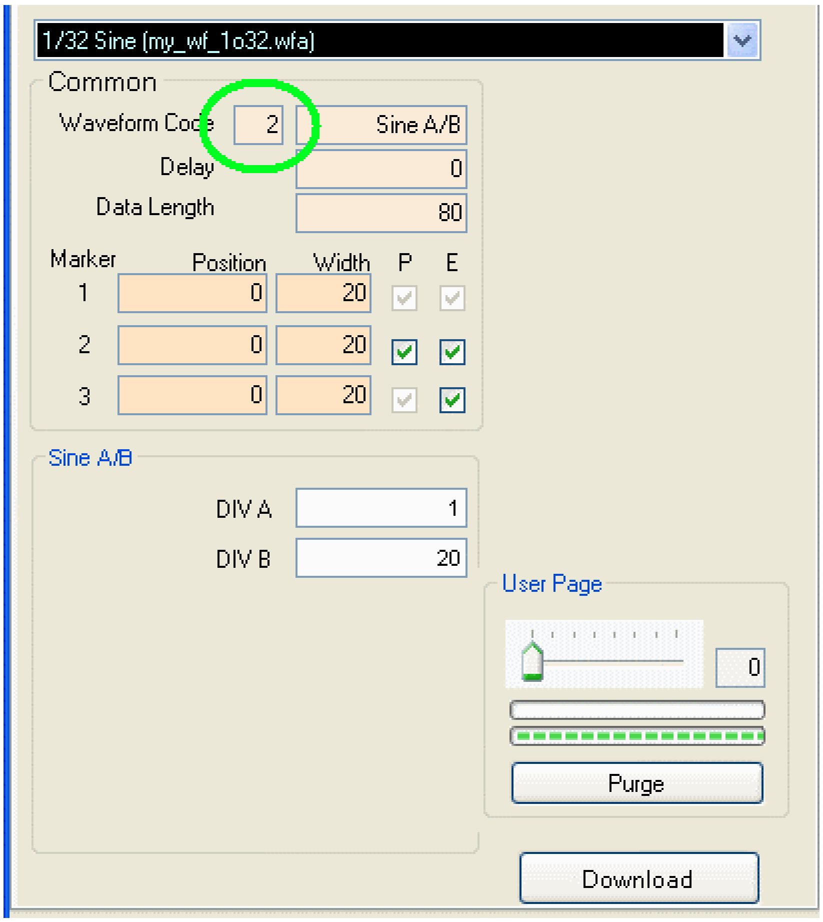


Press the Restart button as needed to restart the waveform.

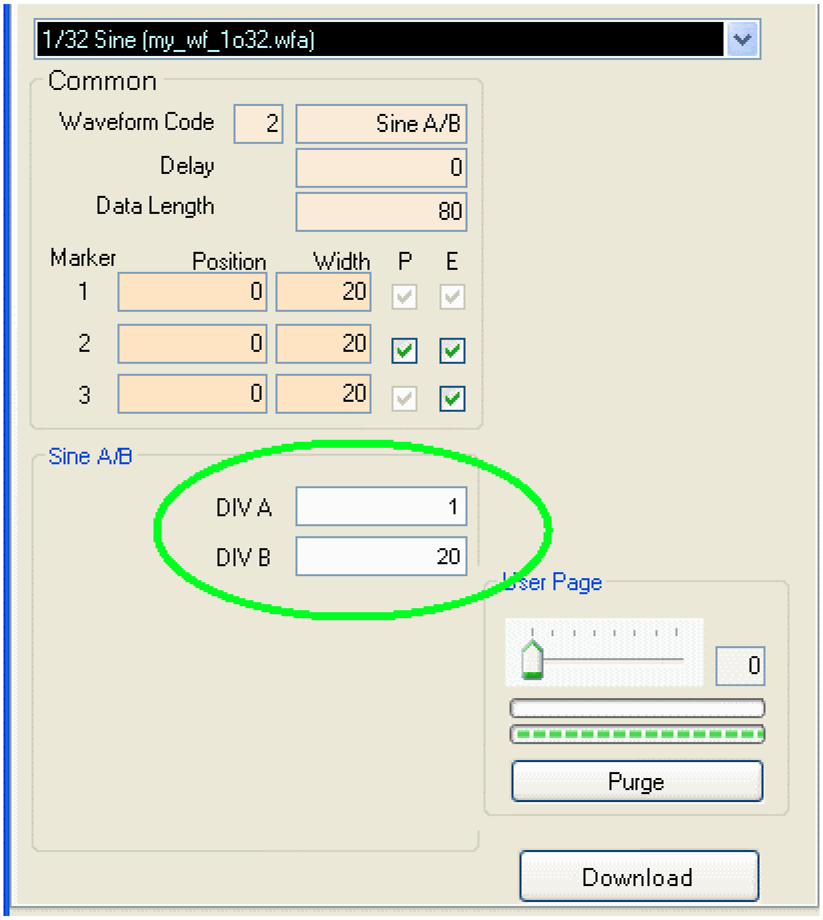
Example 2 : Relative Sine Wave

This example will show you how to create a relative frequency sine wave. The relative sine wave is different from the absolute sine wave in that you use a frequency factor to mulitply with the input clock to get the output sine wave. The output frequency will always be the same factor of the input frequency.

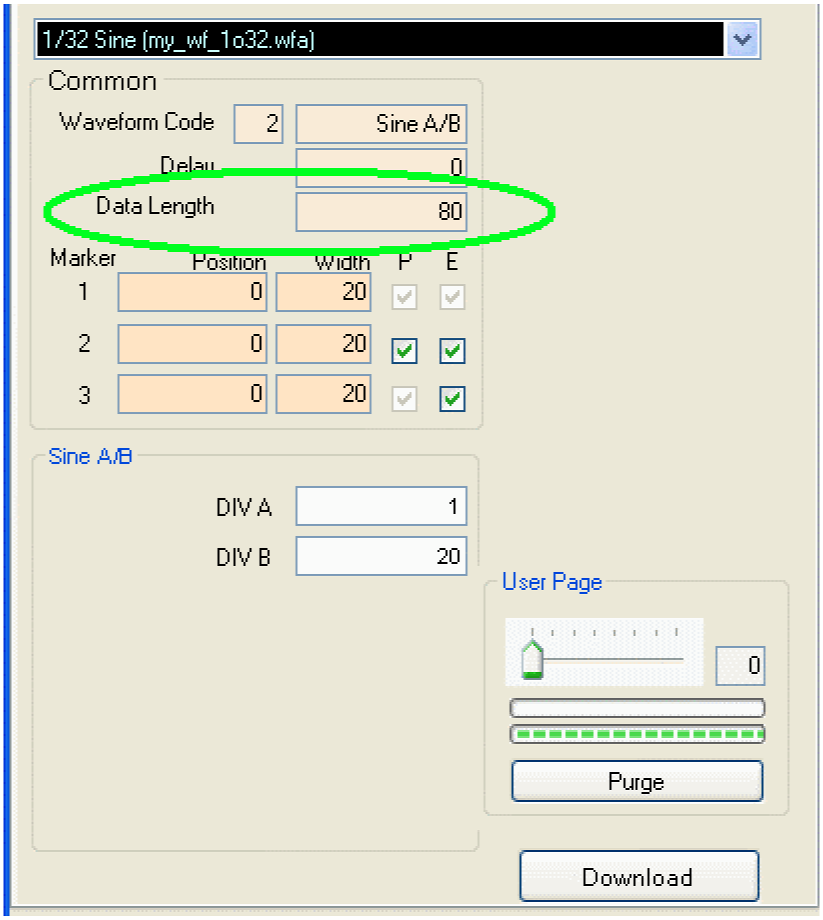
Click on the Waveform tab. In that panel, make sure that the Waveform Code is set to "2". This is the code for relatve frequency sine waves.



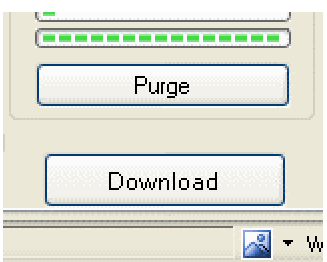
You should see that in the Waveform Type box it will say "Sine A/B". On the bottom you will see two boxes: DIV A and DIV B. DIV A is the numerator of the factor that you will multiply to the input clock and DIV B is the denominator of the factor. They should both be hexadecimal integers. So if wanted to have a factor of 3/4 then you would have DIV A be equal to 3 and DIV B equal to 4. If your input frequency were 2 GHz, then the output frequency of the sine wave would be 1.5 GHz.



You can also change the number of points that the waveform will have per cycle. To do this, change the Data Length to the number of points you want. Please note that the Data Length is specified in hexadecimal, and it should be a multiple of 20 (hex). For more information on Data Length, please see the Data Length page.



Press the Download button (at the lower right corner of the waveform panel). You must press the Download button each time you finish changing the parameters.

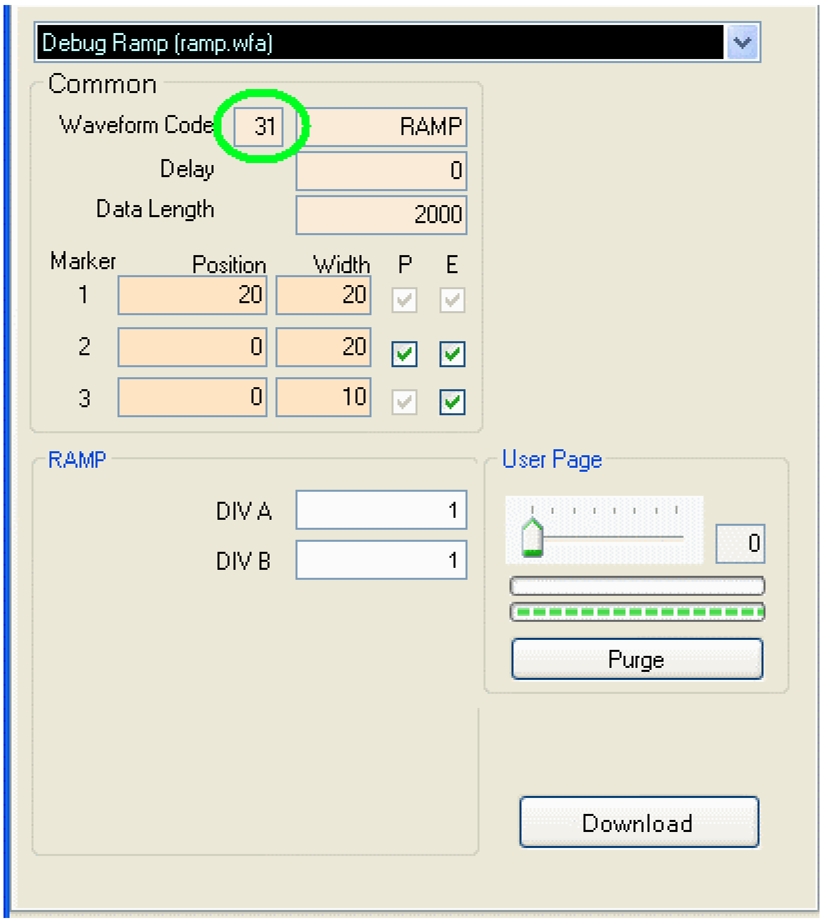


Press the Restart button as needed to restart the waveform.

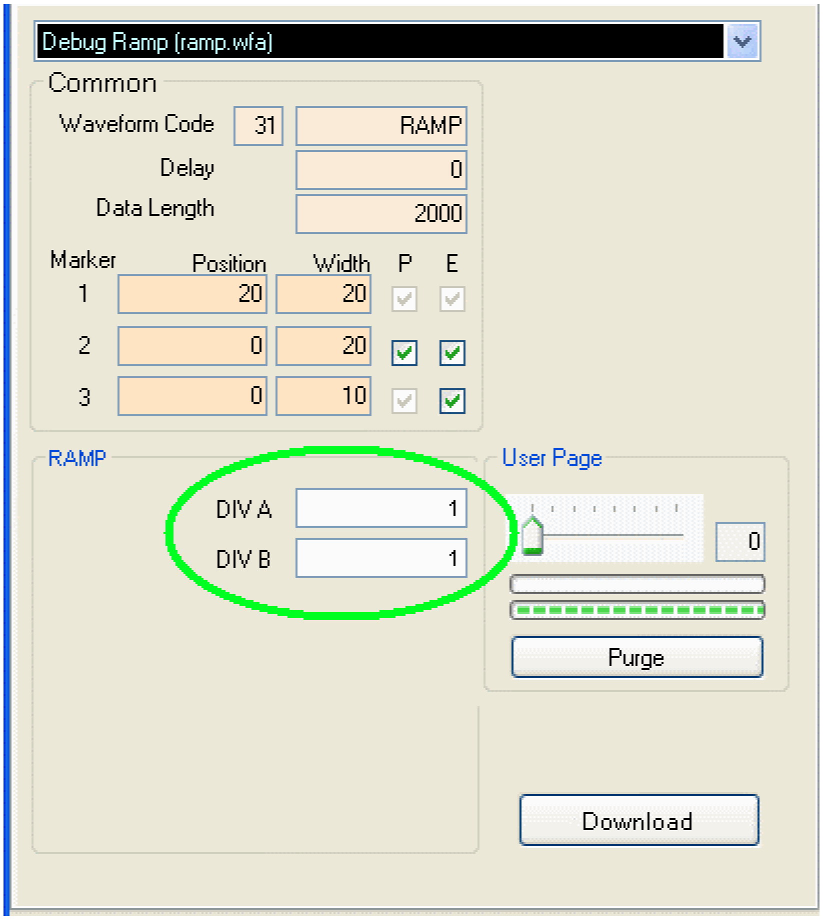
Example 3 : Ramp

This example will show you how to create a ramp waveform. Ramps are a good way to test the normal functioning of the AWG.

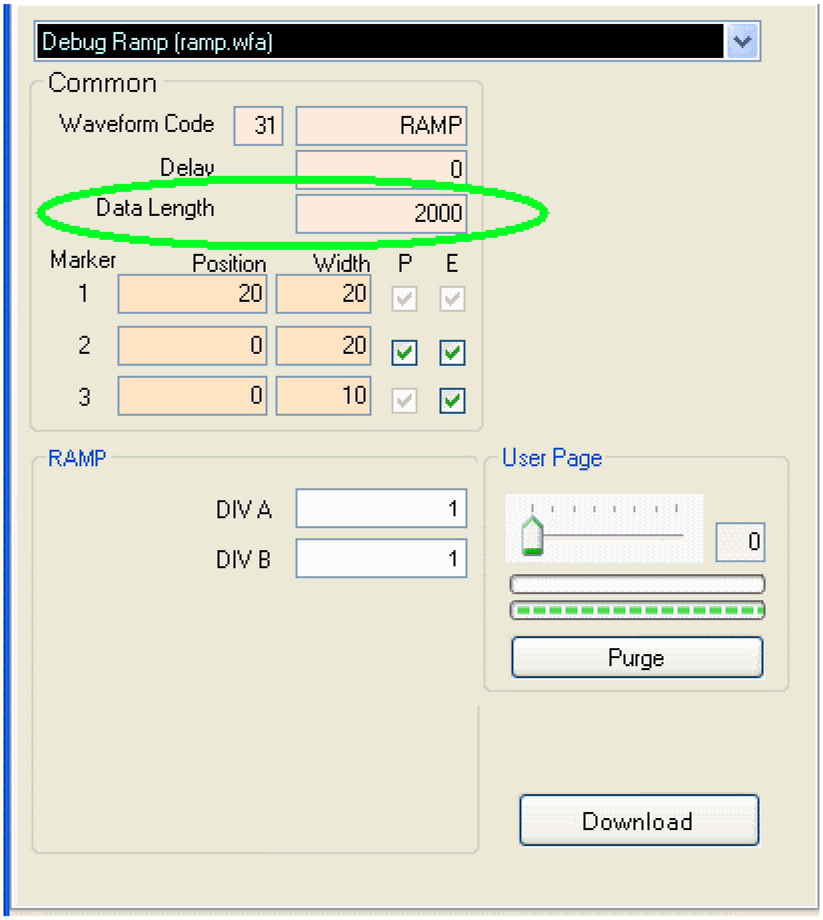
Click on the Waveform tab. In that panel, make sure that the Waveform Code is set to "31". This is the code for a ramp waveform.



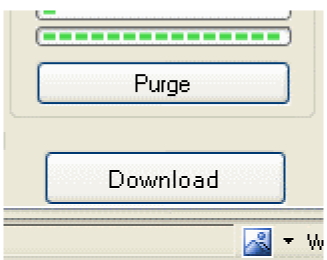
You should see that in the Waveform Type box it will say "RAMP". On the bottom you will see two boxes: DIV A and DIV B. The AWG produces a ramp by multiplying the data point index by the A/B slope factor and the amplitude resolution of the DAC, which is 4096 for the AWG252 and AWG452 and which is 2048 for the AWG801. If you wanted a ramp with a high slope then you would have A/B greater than 1. If you wanted a ramp with a low slope then you would have A/B less than 1.



You can also change the number of points that the waveform will have per cycle. To do this, change the Data Length to the number of points you want. Please note that the Data Length is specified in hexadecimal, and it should be a multiple of 20 (hex). For more information on Data Length, please see the Data Length page.



Press the Download button (at the lower right corner of the waveform panel). You must press the Download button each time you finish changing the parameters.

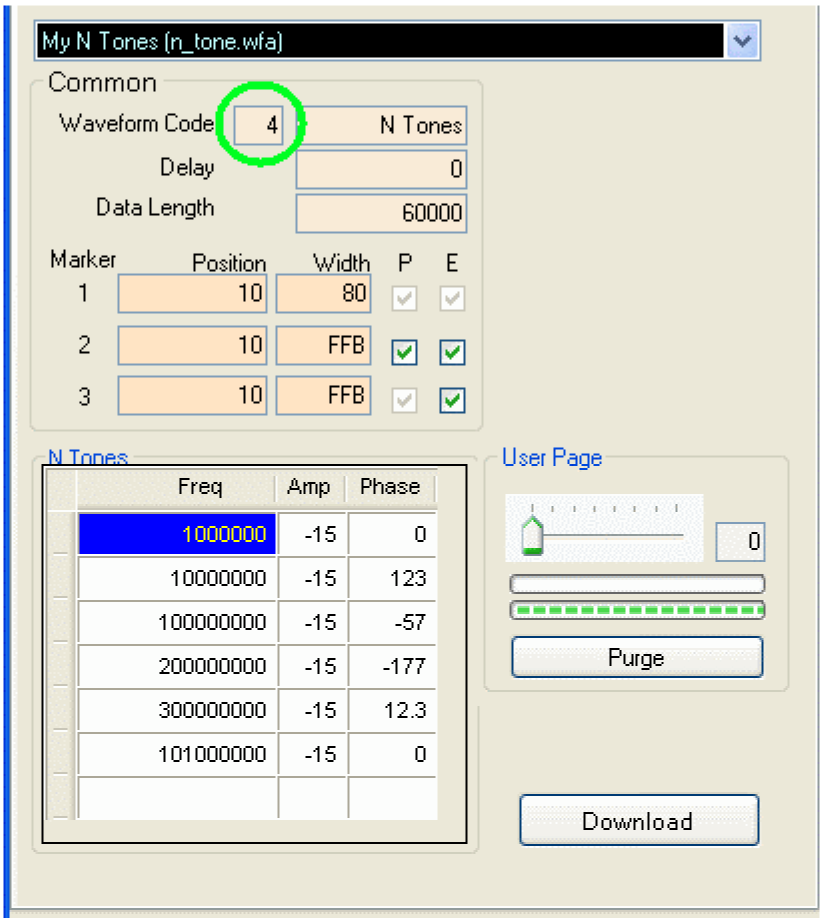


Press the Restart button as needed to restart the waveform.

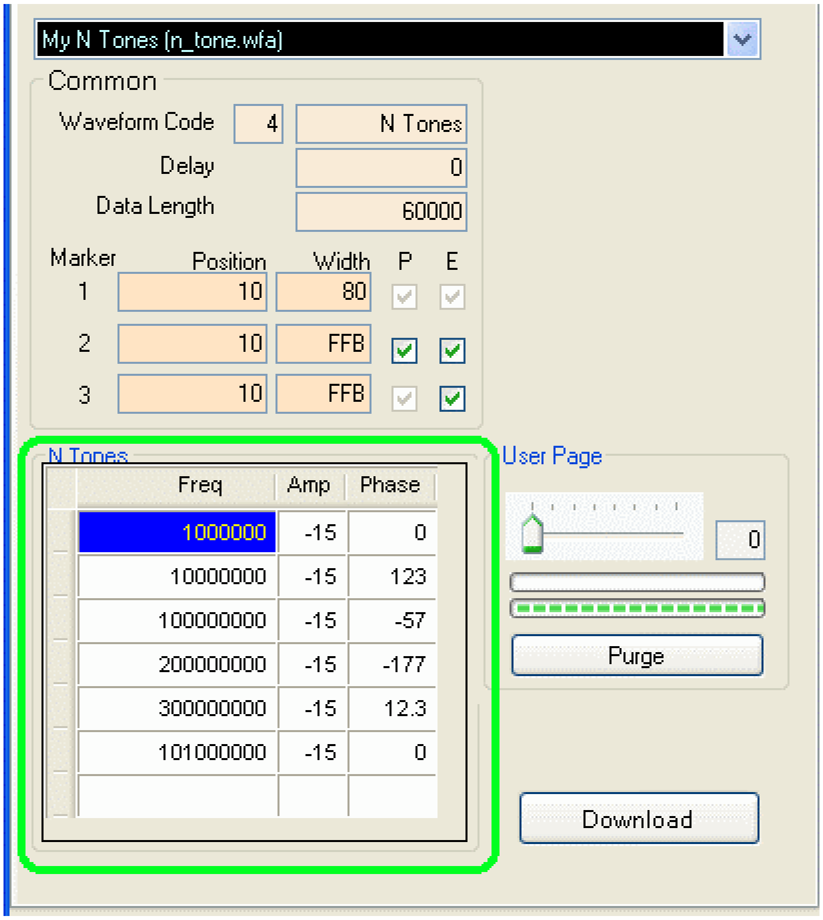
Example 4 : N Tones

This example will show you how to create a N-tone waveform which is just several sine waves added together.

Click on the Waveform tab. In that panel, make sure that the Waveform Code is set to "4". This is the code for N tones.



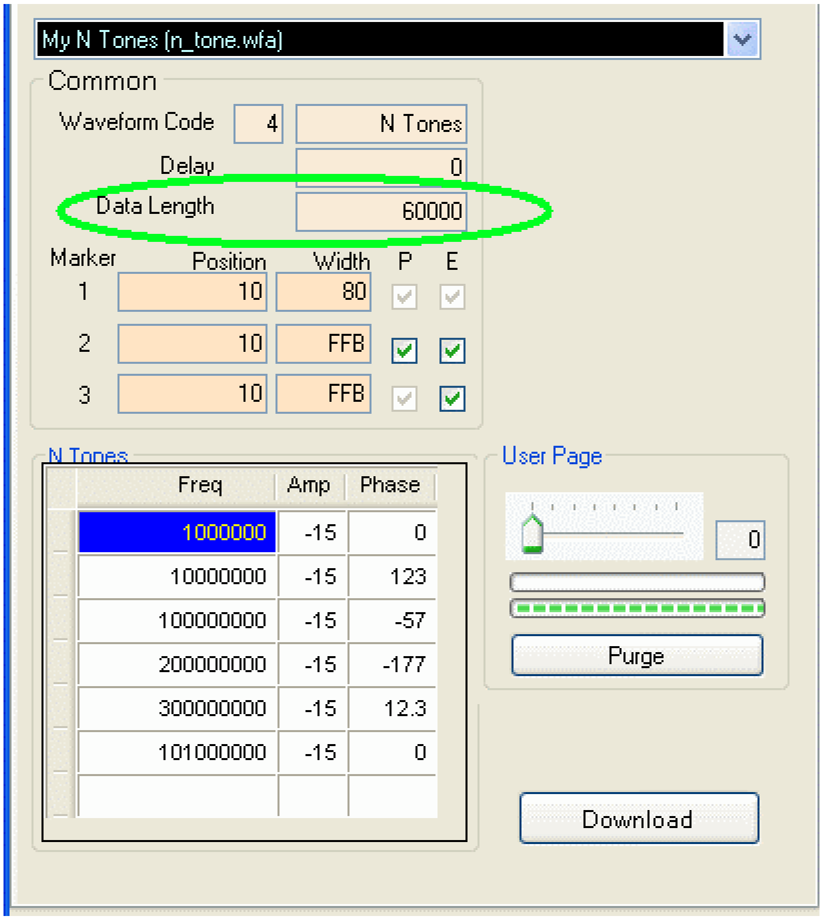
You should see that in the Waveform Type box it will say "N Tones". On the bottom, you will see a table with 3 columns. The first column is for the frequency (Hz) of the sine waves, the second is for the amplitude (dBc) and the last column is for the phase (degrees) of the sine waves.



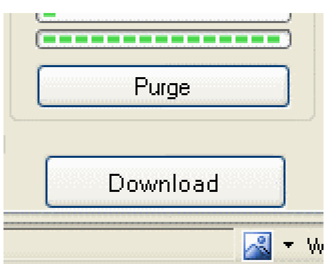
Below is an example of the parameters for a six-tone waveform. There are six sine waves added together:
100 MHz with amplitude of -15 dBc and phase of 223 degrees,
567 MHz with amplitude of -16 dBc and phase of 0,
333 MHz with amplitude of -16 dBc and phase of 45 degrees,
120 MHz with amplitude of -14 dBc and phase of 123 degrees,
1 MHz with amplitude of -14 dBc and phase of 77 degrees, and
1.23 MHz with amplitude of -14 dBc and phase of 234 degrees.

Freq	Amp	Phase
100000000	-15	223
567000000	-16	0
333000000	-16	45
120000000	-14	123
1000000	-14	77
1230000	-14	234

You can also change the number of points that the sine wave will have per cycle. To do this, change the Data Length to the number of points you want. Please note that the Data Length is specified in hexadecimal, and it should be a multiple of 20 (hex). For more information on Data Length, please see the Data Length page.



Press the Download button (at the lower right corner of the waveform panel) and wait for the data to finish downloading to the AWG module. You must press the Download button each time you finish changing the parameters.

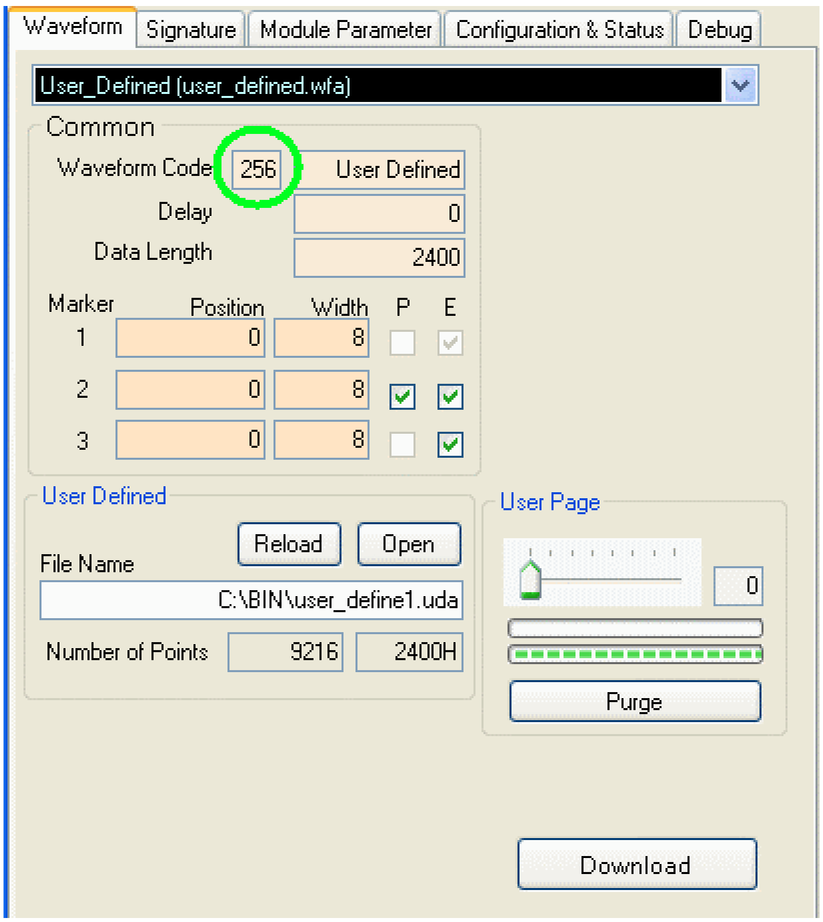


Press the Restart button as needed to restart the waveform.

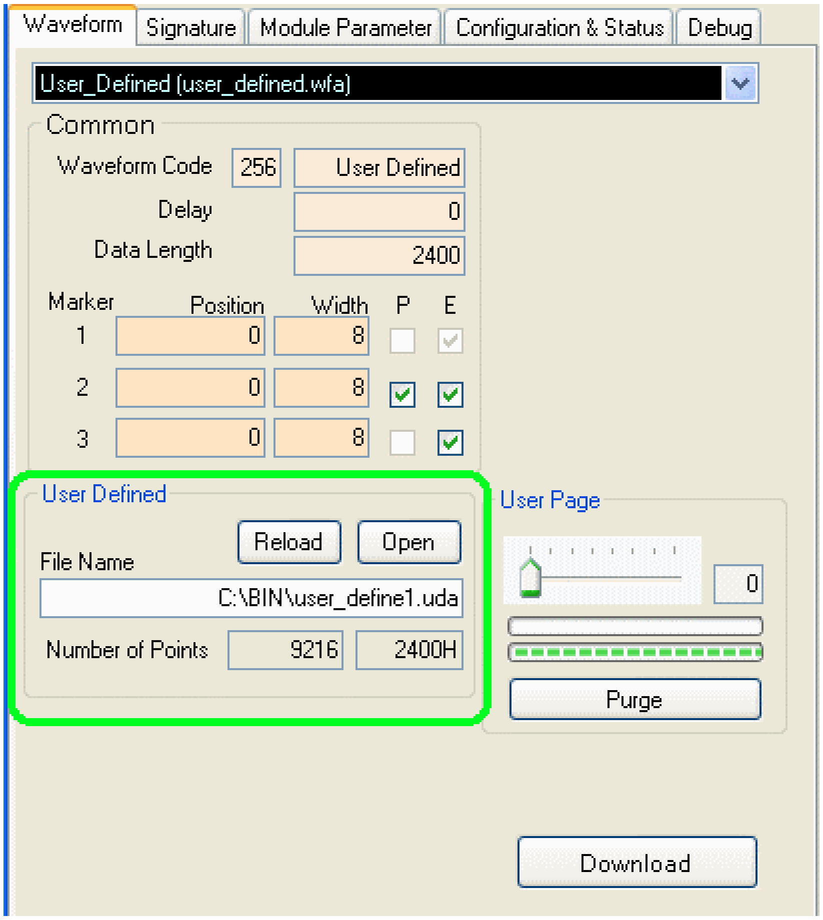
Example 5 : User Defined

This example will show you how to create an user-defined waveform which is a custom waveform that you can create using an external text file.

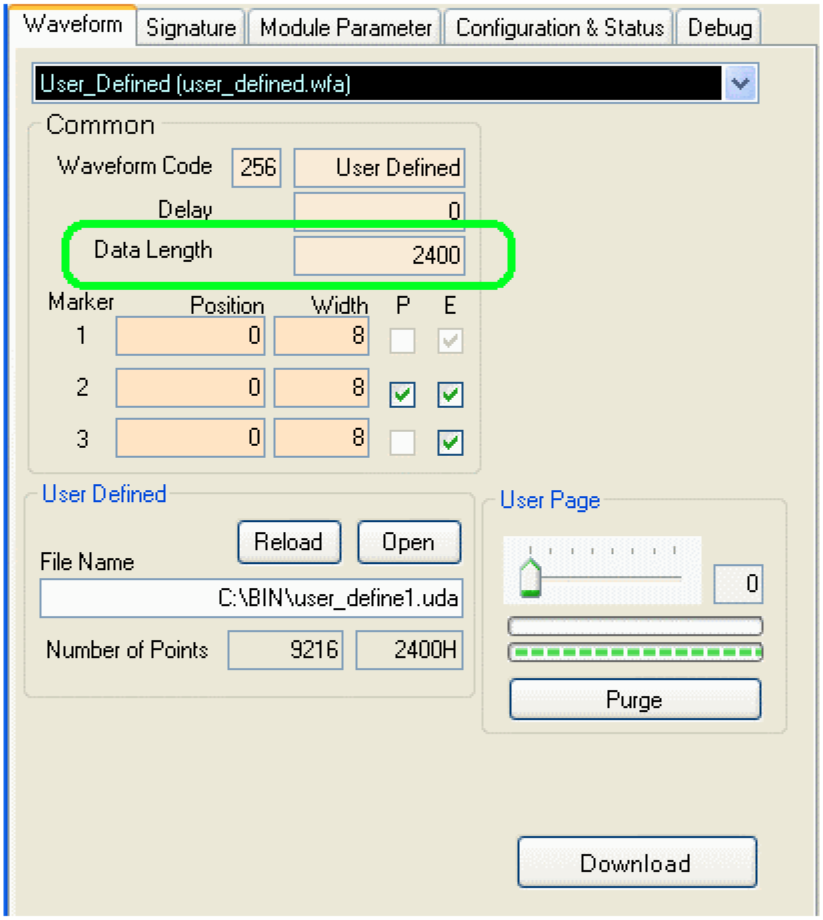
Click on the Waveform tab. In that panel, make sure that the Waveform Code is set to "256". This is the code for user-defined waveforms.



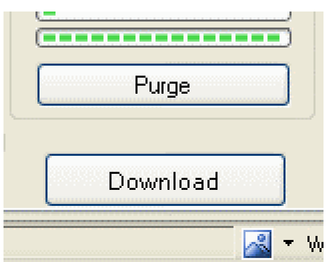
You should see that in the Waveform Type box it will say "User Defined". On the bottom you will see the box change and you will be able to open up a custom ".uda" file to load. For more information on the custom user defined files, please see the User Defined Waveforms page.



You can also change the number of points that the sine wave will have per cycle. To do this, change the Data Length to the number of points you want. Please make sure this number is at least as long as the number of points in your user-defined waveform. Also, please note that the Data Length is specified in hexadecimal, and it should be a multiple of 20 (hex) for the AWG452 or a multiple of 10 (hex) for the AWG252. For more information on Data Length, please see the Data Length page.



Press the Download button (at the lower right corner of the waveform panel). You must press the Download button each time you finish changing the parameters.

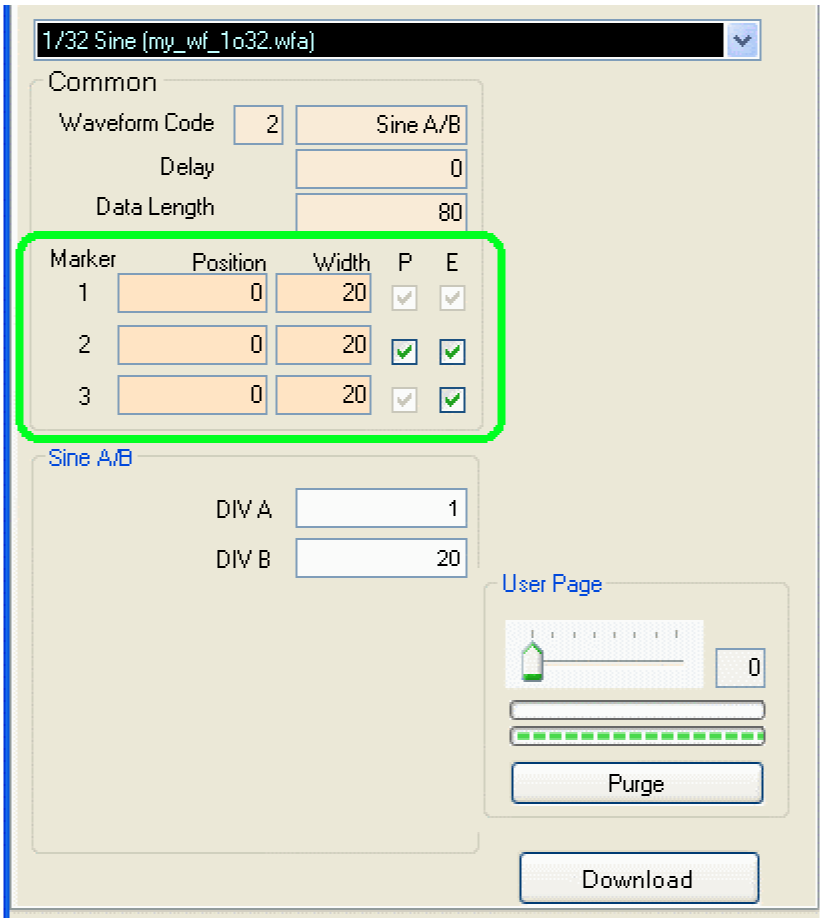


Press the Restart button as needed to restart the waveform.

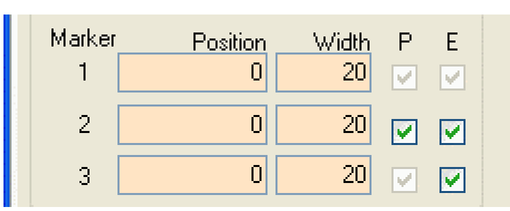
Markers

Markers can help you determine if the waveform that is produced is correct. We have provided three markers, each with different characteristics. From the GUI, you can specify when and for how long each marker will appear, but if you define your own waveforms, you can have any of the markers appear numerous times.

The marker settings are in the Waveform panel, just below the Data Length box.



Position specifies the data point where you would like to start the marker. Width specifies the number of data points you would like the marker to be active.
The "P" polarity checkbox is only available for Marker 2 and allows you to select active low marker signals (when unchecked).
The "E" enable checkbox is only available for Marker 2 and Marker 3 and is used for turning on those marker outputs.



Marker 1 is 1.8 V TTL (rail-to-rail) logic level while Marker 2 and Marker 3 are both 3.3 V LVTTL (rail-to-rail) logic level. Marker 1 has less delay than either Marker 2 or Marker 3. For details about the three available markers, please see the Parameters page of the Operation Details section.

Background Information -

General Information -

The AWG modules generate arbitrary CW waveforms with sampling rates up to 2 Gsps for the AWG252, 4 Gsps for the AWG452, and 8 Gsps for the AWG801. The on-board SRAMs provide 4M 12-bit data memory for the AWG252, 8Mx12-bit data memory for the AWG452, and 8Mx11-bit memory for the AWG801. The AWG can be controlled by a PC or can work alone with pre-stored waveforms. The sole RF input is the single-ended clock source **CKIP**, which can be operated above 2.0 GHz with minimum power of 0 dBm. The RF outputs of the module consist of a pair of differential analog outputs, **OUTP** and **OUTN**, which have 50 ohm back termination. The module accepts a high-speed trigger and generates programmable synchronization output and marker signals.

Key Features:

- 12-bit DAC with 10-bit linearity
- Clock rate up to 2.0 GHz for the AWG252 and 4 GHz for the AWG452 and AWG801
- 4M x 12-bit memory depth for the AWG252,
  8M x 12-bit memory depth for the AWG452, and
  8M x 11-bit memory depth for the AWG801,
  all with multi-page configuration
- Up to 2 millisecond waveform at 2 GHz clock
- Programmable cyclic length and marker signal
- API software trigger and Hi-speed hardware trigger
- USB 2.0 compliant interface (other interfaces available upon request)
- 8 W power consumption for the AWG252,
  15 W power consumption for the AWG452, and
  30 W power consumption for the AWG801
- Various built-in waveforms, including multi-tone CW and continuous chirping
- Companion API and software drivers for easy system development
- Multi-board Synchronization
- Dynamic User Paging

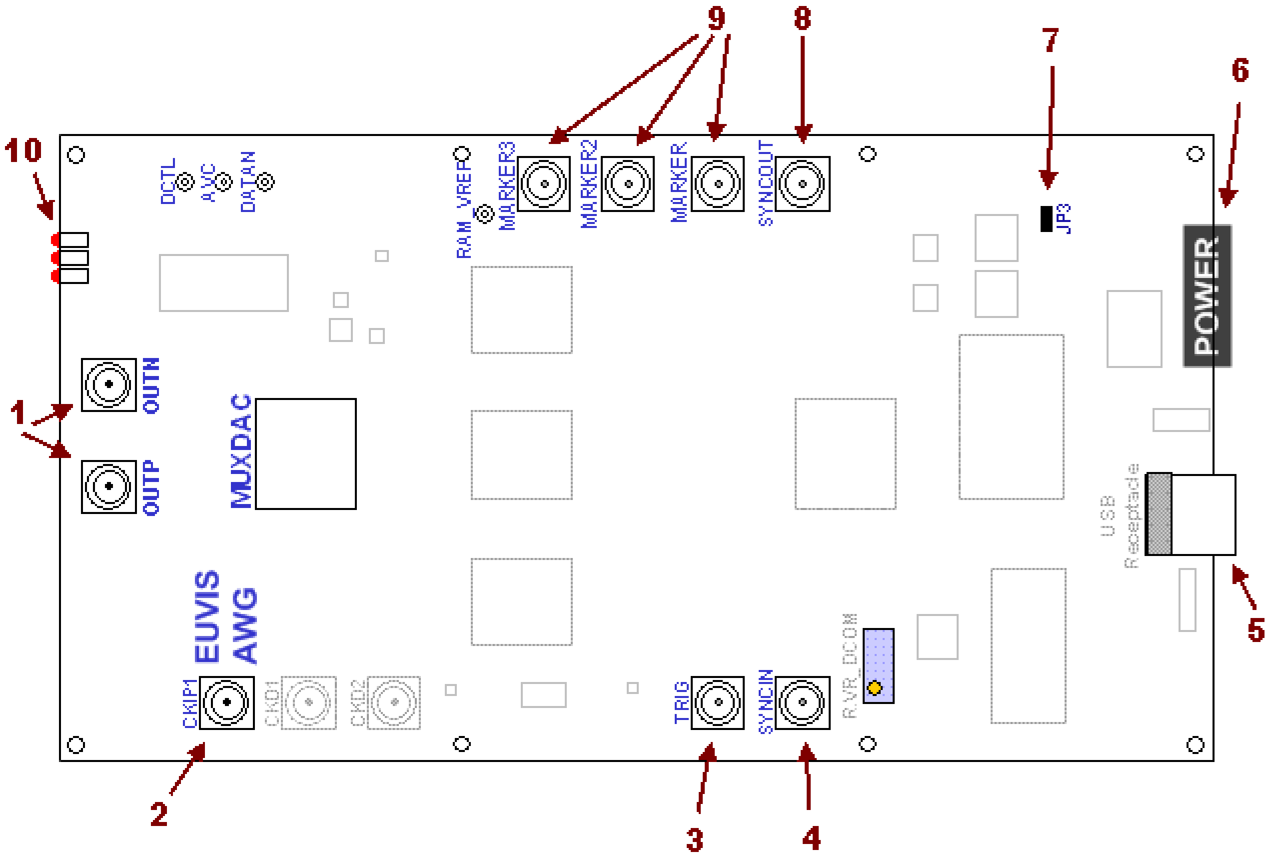
Electrical Specifications -

Parameter	Symbol	Min	Typical	Max	Unit
Operating Temperature	T_O		25		C°
AWG252 Clock Frequency	f_{CK}	0.5		> 2	GHz
AWG452 Clock Frequency	f_{CK}	1		> 4	GHz
Clock Input Power	P_{CK}	0	3	10	dBm
Output Level	V_{out}	-635		0	mV
Output Power	P_{out}	-4		0	dBm
Output Phase Noise	N_f			-130	dBc/Hz
Clock Port Return Loss	RL_{CK}		15		dB
Output Port Return Loss	RL_{RF}		15		dB
AWG252 Power Supply	-5V	-4.5	-5	-5.5	V
	I_{-5}		915		mA
	+1.8V	+1.7	+1.8	+1.9	V
	$I_{+1.8}$		1600*		mA
	+3.3V	+3	+3.3	+3.5	V
	$I_{+3.3}$		195		mA
	+5V	+4.5	+5	+5.5	V
	I_{+5}		205		mA
AWG452 Power Supply	-5V	-4.5	-5	-5.5	V
	I_{-5}		1.2		A
	+1.8V	+1.7	+1.8	+1.9	V
	$I_{+1.8}$		2.9*		A
	+3.3V	+3	+3.3	+3.5	V
	$I_{+3.3}$		405		mA
	+5V	+4.5	+5	+5.5	V
	I_{+5}		130		mA
AWG801 Power Supply	-5V	-4.5	-5	-5.5	V
	I_{-5}		3.9		A
	+1.8V	+1.7	+1.8	+1.9	V
	$I_{+1.8}$		3.9*		A

	+3.3V	+3	+3.3	+3.5	V
	$I_{+3.3}$		850		mA
	+5V	+4.5	+5	+5.5	V
	I_{+5}		160		mA

* Current for the 1.8V supply varies with clock frequency. At 2 GHz clock frequency, the 1.8V supply current is near 1.6A for the AWG252. At 4 GHz clock frequency, the 1.8V supply current is near 2.9A for the AWG452 and 3.9A for the AWG801.

Board Layout -



The AWG252 board dimensions are 9.5 inches by 5 inches.
The AWG452 board dimensions are 9 inches by 5.25 inches.

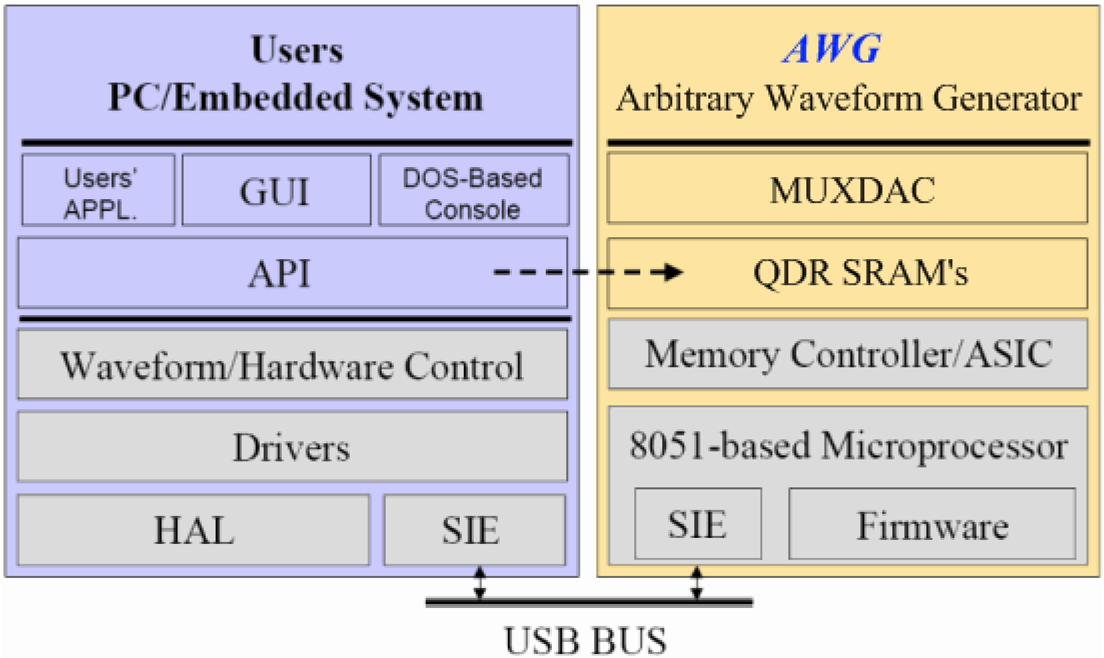
The module is manufactured from 4-layer sandwiched RO4003/FR4/RO4003 with a total thickness of 54 mils. The RF I/O's are standard SMA connectors and the USB port is a type-B USB receptacle. Two heat sinks are applied to the MuxDAC and the ASIC memory controller.

	Description
1	**OUTP / OUTN (SMA)** SMA connectors for output waveforms. They should be connected to an oscilloscope and a spectrum analyzer. If only using one output, please terminate the unused output with a 50 ohm termination.

2	**CKIP (SMA)** SMA connector for input clock. Please connect to a clock source with minimum power of 0 dBm.
3	**TRIG (SMA)** Optionally connect a trigger signal source to the AWG board. For more information on synchronization, please go to the Synchronization section.
4	**SYNCIN (SMA)** SMA connector for SYNCIN signal input. Part of the synchronization process. For more information on synchronization, please go to the Synchronization section.
5	**USB receptacle** Please use a type-B USB cable to connect the AWG to your computer any time after you have installed the Software and Drivers.
6	**Power Header** Please use the provided plug to connect to the power header. The plug and header can only be connected in one orientation but you MUST make sure that the order of the wires and the power supplies are correct on the plug so that you do not burn out the board. For more information on setting up the plug, please go to the Power Plug section.
7	**ROM Jumper (JP3)** For Generic Installation of firmware, you will need to remove this shunt/jumper temporarily before powering up the AWG board. Just before writing to the ROM, you will need to place the jumper back. For normal Firmware Updating, you can leave the jumper in place.
8	**SYNCOUT (SMA)** SMA connector for SYNCOUT signal output. Part of the synchronization process. For more information on synchronization, please go to the Synchronization section.
9	**MARKERS (SMAs)** SMA connectors for Marker signal outputs. Up to three different markers can be output every time a waveform is output. For details, please see the discussion of how to set the Markers in the waveform panel of the GUI.
10	**LEDs** Six indicator LED's help identify the AWG status.

Operation Theory -

The conceptual architecture is shown in the following:



Hardware — The AWG252 consists of a Euvis MD652D high-speed MUXDAC, three QDR SRAMs (Cypress CY7C1313BV18), an ASIC memory controller, and an enhanced 8051 microcontroller (Cypress CY7C68013A) with external RAM and EEPROM. The key front-end component MD652D features >3GHz clock rate, 12-bit amplitude and 13-bit phase resolution. It takes 48 single-ended amplitude word data as inputs. The differential analog outputs are 50 ohm terminated. The waveform data are stored in the QDR SRAMs, which provide 4M x 12-bit memory depth.

The AWG452 replaces the MUXDAC and SRAMs of the AWG252 with the Euvis MD681S high-speed MUXDAC and six QDR SRAMs (Cypress CY7C1313BV18). The MD681S features a >4GHz clock rate and 12-bit amplitude resolution. It receives 96 single-ended amplitude word data as inputs, which are stored in the QDR SRAMs, which provide 8M x 12-bit memory depth.

The AWG801 replaces the MUXDAC and SRAMs of the AWG252 with the Euvis MD662 high-speed MUXDAC plus Euvis MX4411 high-speed Muxes and six QDR SRAMs (Cypress CY7C1315BV18). The MD662 features a >4GHz clock rate and 12-bit amplitude resolution. It receives 88 differential amplitude word data as inputs, which are 4:1-muxed from data stored in the QDR SRAMs, which provide 8M x 11-bit memory depth.

The ASIC memory controller performs reading/writing controls and data transfers. The microcontroller has an integrated USB 2.0 transceiver, a series interface engine (SIE) and an enhanced 8051 microprocessor, which provides a user friendly interface for the host PC or existing systems and general-purpose controls.

Software — The companion API (Application Programming Interface) performs all the hardware controls and handles the data transfers on the user end. The gray-color blocks represent user-invisible kernel layers in the operations. All of the user operations/commands are executed virtually onto the QDR SRAMs and the MUXDAC on the AWG even though physically the bulk of the instructions are transferred via the USB bus and executed by the kernel layers. This virtual connection between the API and RAM provides a clean and simple interface for users to develop their own application software without the trivial knowledge of the low-level drivers, the USB interface, the firmware, and the control hardware.

The API consists of a set of callable routines in Microsoft Visual C++ library. Users can develop their own application software to operate the AWG in their own manner or modify the existing system to adopt the AWG into their end products. The API consists of three groups of functions:

1. Waveform generation: The built-in waveform generator takes users' parameter inputs, computes the digital codes, and downloads the codes onto the RAM's accordingly. The built-in waveforms include monotonic sine waves, multi-tone sine waves, triangle waves, square waves, and high-speed linear chirping CW FM waveforms that can have frequency updates at each clock cycle.
2. Waveform control: The waveform control function downloads the digital data onto the RAM in the writing phase, controls the AWG, and provides the RAM's cyclic addresses in the reading phase. It also controls the cyclic depth.
3. Import/Export waveforms: The API provides routines to import user-defined waveform data in various formats, such as x-y ASCII format and MATLAB compatible ASCII format. On the other hand, all the built-in waveforms can be exported to files for analysis.

In addition to the API, a console-based control program and a Windows-based graphical control program provide users convenient ways to control and operate the AWG.

Setup -

First Powerup Checklist -

We have provided you a list of things you need to check for when you first use the AWG hardware and software. Going through this list will ensure proper operation of the AWG in the future. After completing this list, please see the AWG Manual for instructions to use the software.

1. Power Connection

The power adapter is rated for 100V-240V AC input. Please plug it into an electrical outlet. Insert the other end into the header at the bottom of the AWG board. The plug has flanges so that there is only one orientation in which the plug and the header can connect.

2. Initial Powerup Waveform

You will now need to check if the initial powerup waveform is correct. Connect your clock to the CKIP SMA near the upper left corner of the board, and set the frequency to 4 GHz with power +3dBm. Then connect your oscilloscope to the OUTP SMA and your spectrum analyzer to the OUTN SMA at the upper edge of the board. You should see a sine wave at 125 MHz, or 1/8th of your clock frequency.

If you can see the waveforms correctly then you can go on to install the AWG software. If you can see a waveform but it does not look like the one above then try unplugging and reconnecting the power supply.

If after a few tries the waveform still does not show up correctly then you will need to contact us for troubleshooting. Likewise, if you do not see any waveform at all then you will also need to contact us.

3. Software Installation

Please download the Setup Files and follow the instructions detailed in the Software Setup section of the manual. Please be sure to also check the Software Requirements section. After installation, navigate to the AWG program folder, which by default is "C:\Program Files\Euvis\AWG\" and check to see if there is a file called "AWG_452_SGN_0XX.dat" where XX is your AWG series number. This is the signature file that will allow you to switch clock frequencies without needing to adjust internal AWG settings for proper operation every time. If you do not have this signature file in the program folder, then you will need to contact us.

Software Requirements -

- Windows XP
- Microsoft .NET Framework 2.0
- Microsoft .NET Framework 2.0 SDK
- Euvis AWG Application

» If you do not have .NET Framework 2.0, you may download it here. The installer is approximately 23MB for the x86 version and will take about 4 minutes on a 768 kbps connection.

» If you do not have .NET Framework 2.0 SDK, you may download it here. The installer is about 354 MB and will take about 1 hour and 3 minutes on a 768 kbps connection.

Software Setup -

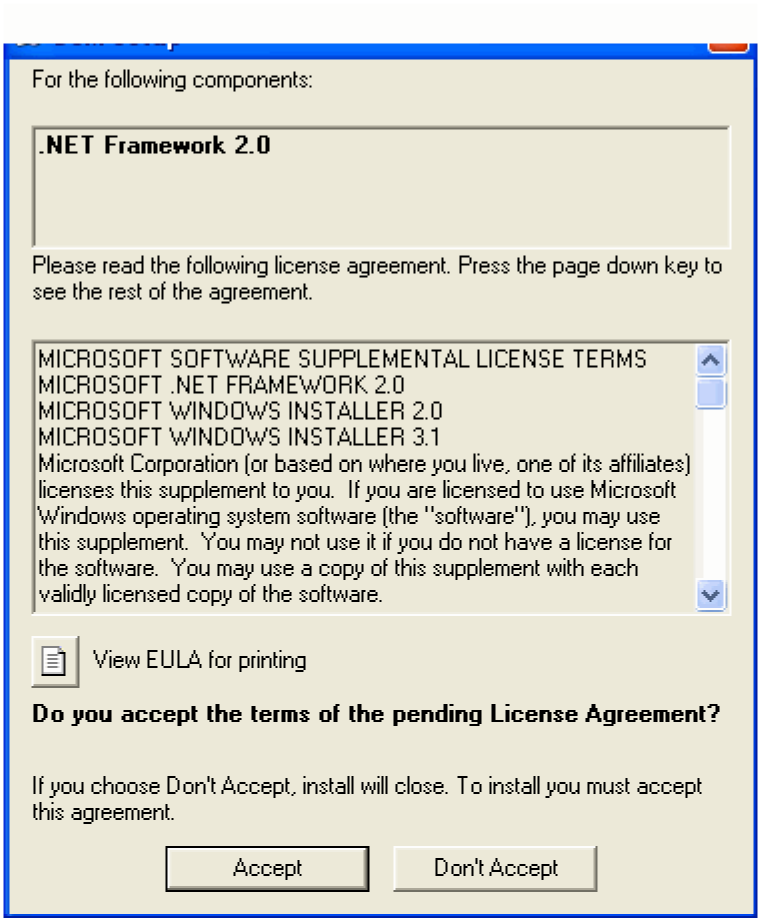
» For now, **do not connect the USB cable** yet. You will plug it in later in the software installation procedure.

» Ensure that you have the two Euvis AWG installation files **setup.exe** and **setup.msi** on your computer.

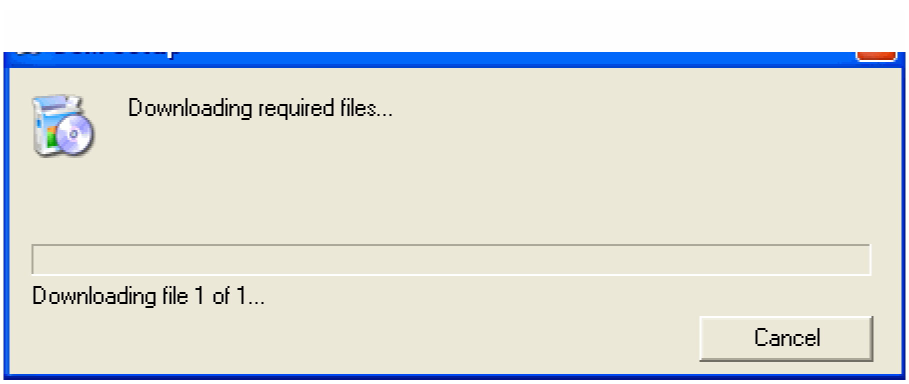
» Go to the directory where you downloaded the installation files.

» Double click the **setup.exe** file. It is the icon with the box and CDROM without the monitor.

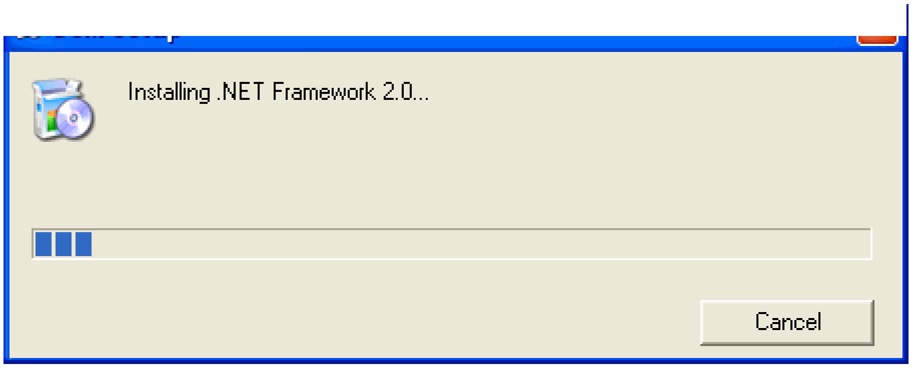
» If you do not have .NET Framework 2.0, the setup program will prompt you to install it. If you do have .NET Framework 2.0, please skip the next few instructions. If not, please click **Accept**.



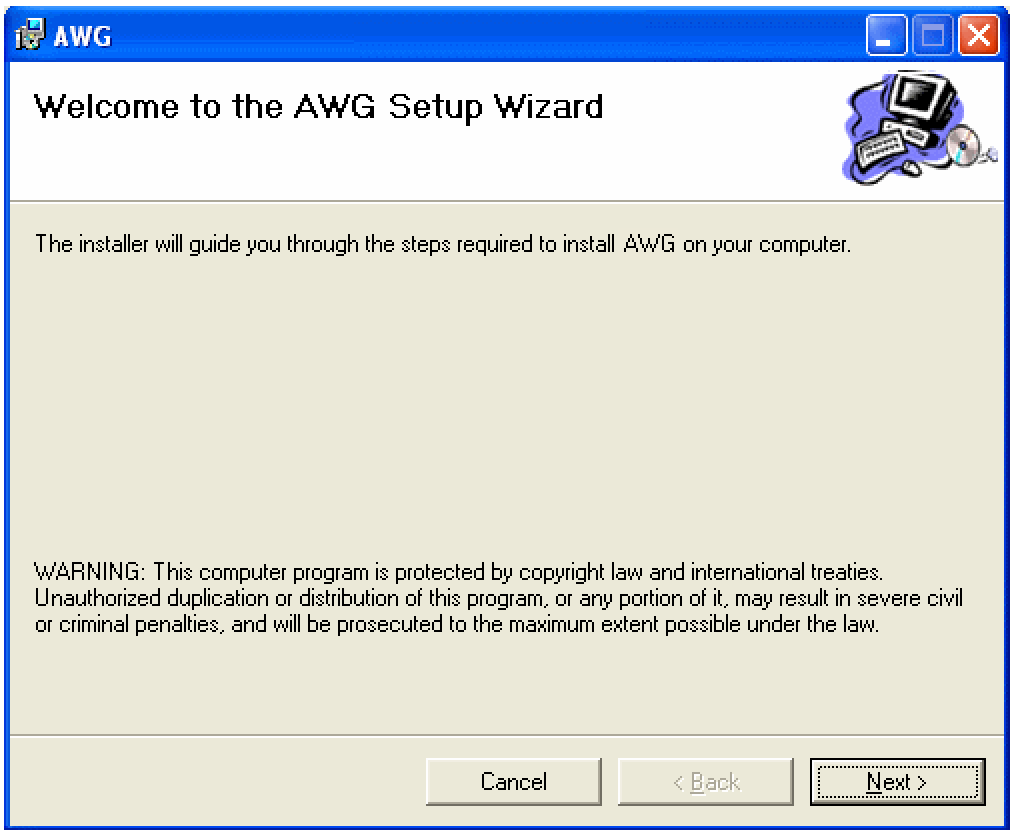
» The installer will download the .NET Framework 2.0 from the Microsoft website. The file is approximately 23MB for the x86 version. On a 768 kbps connection it will take approximately 4 minutes to complete. If you do not have the software and the installer does not download it for you, you may manually download it at the following website: Microsoft .NET Framework 2.0 Redistributable.



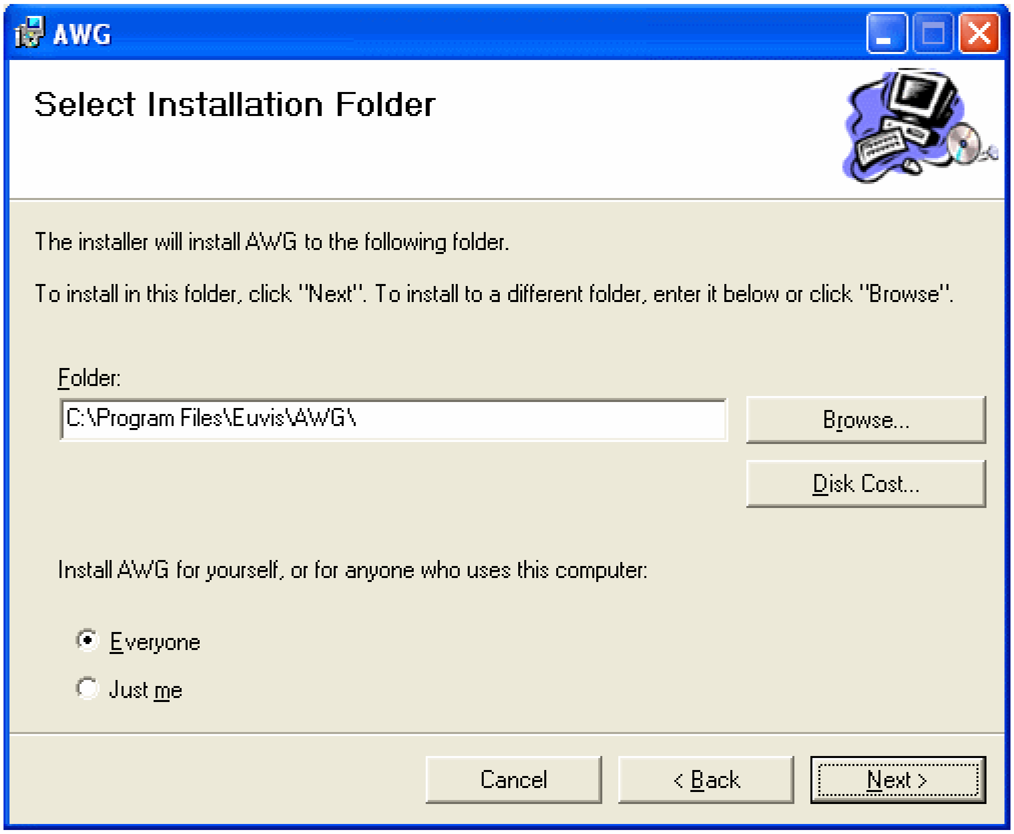
» After downloading, the installer will install the .NET Framework 2.0.



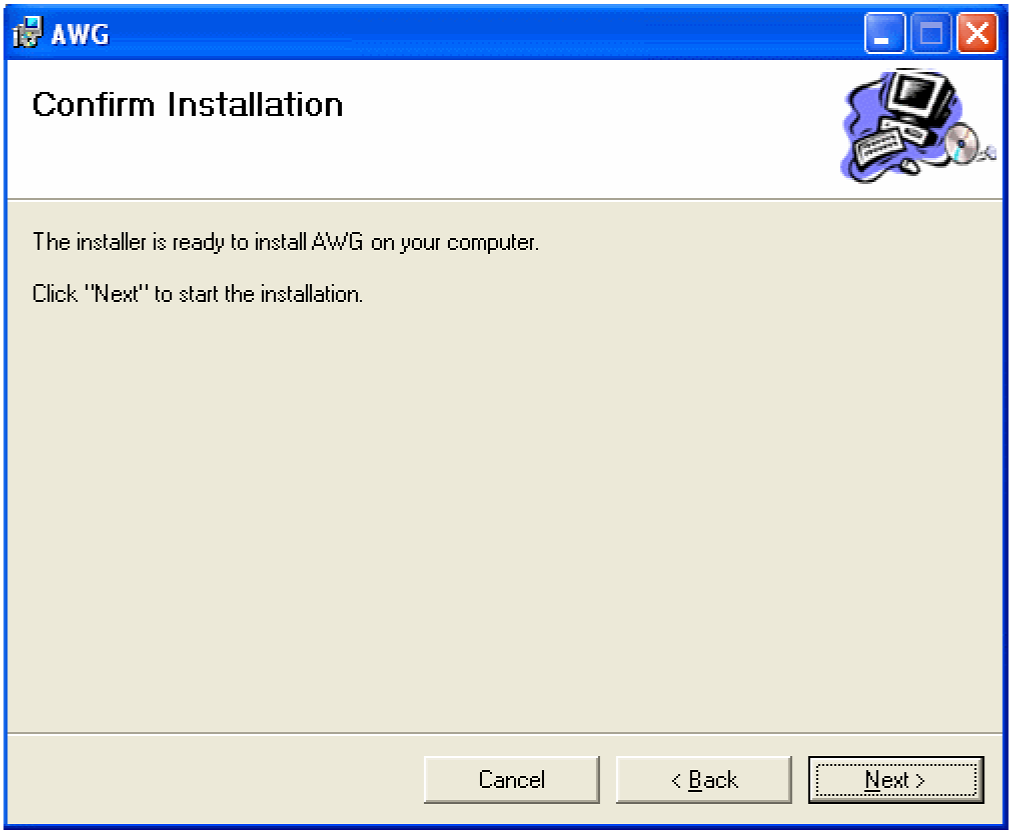
» Following the .NET installation, the AWG Setup Wizard will appear. Please click on **Next**.



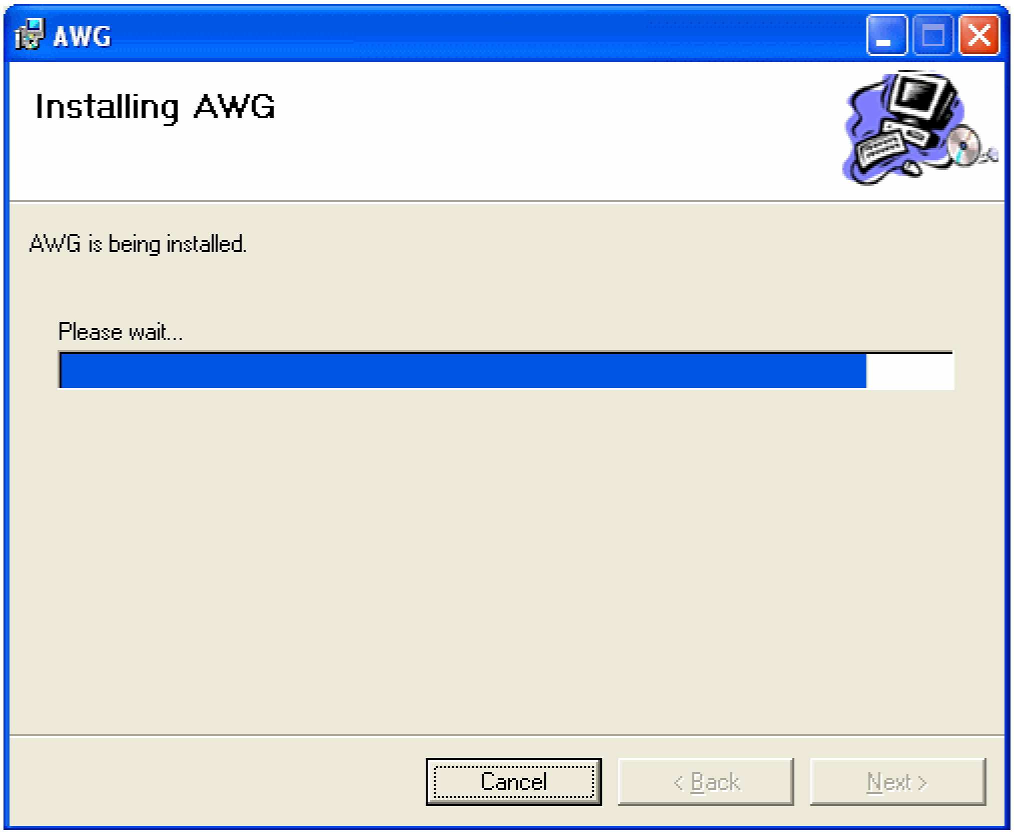
» You may choose another location to install the files or you can keep the default location. You can also choose to install for all users or for just the user you are currently logged in as. After you are satisfied, click on **Next**.



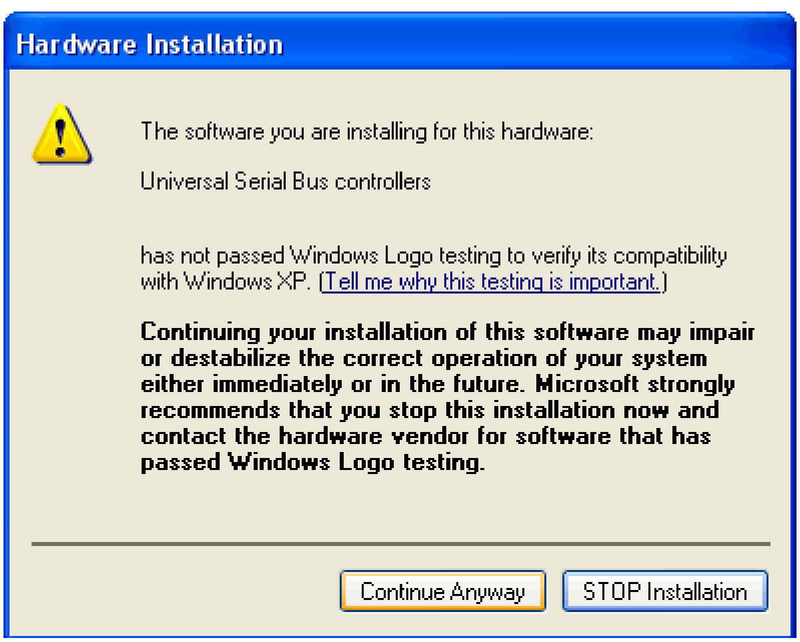
» Click **Next** on the "Confirm Installation" screen.



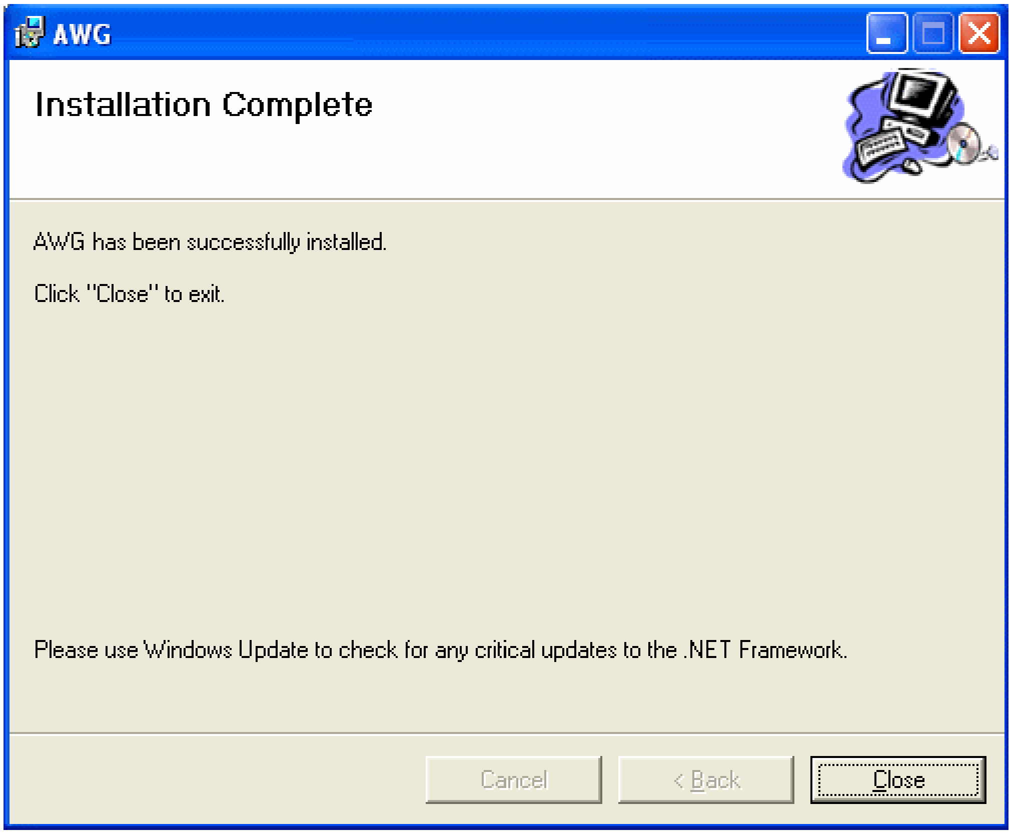
» The installer will copy all necessary files into your system.



» When setup attempts to install the USB controller drivers, Windows will give you a warning. Please click on **Continue Anyway**.



» After installation is complete, please click on **Close**.



» At this point, if you do not have Microsoft .NET Framework 2.0 SDK you will need to install it. This software is required, and our program will NOT run if you do not have this. You may download the SDK software at the following website: Microsoft .NET Framework 2.0 SDK. The installer is about 354 MB and will take about 1 hour and 3 minutes on a 768 kbps connection.

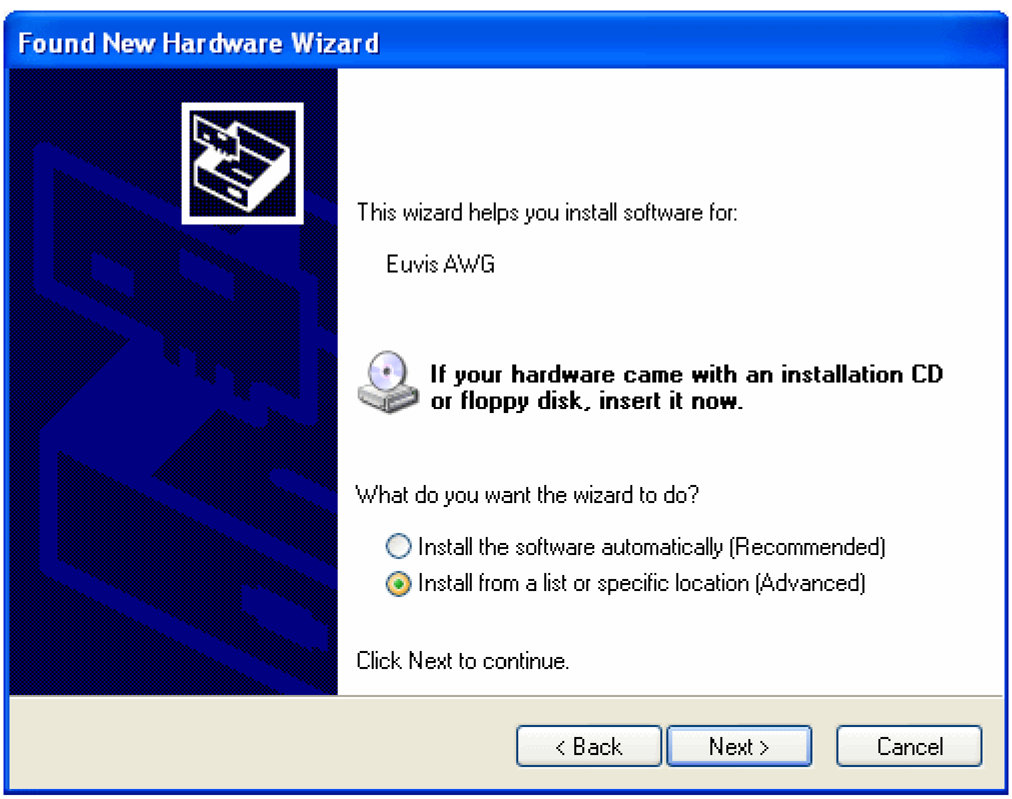
» After you have downloaded and installed the SDK software you will have completed the initial software and driver installation.

» Now, please **connect a USB A to B Cable** from the AWG board to the computer for the first time.

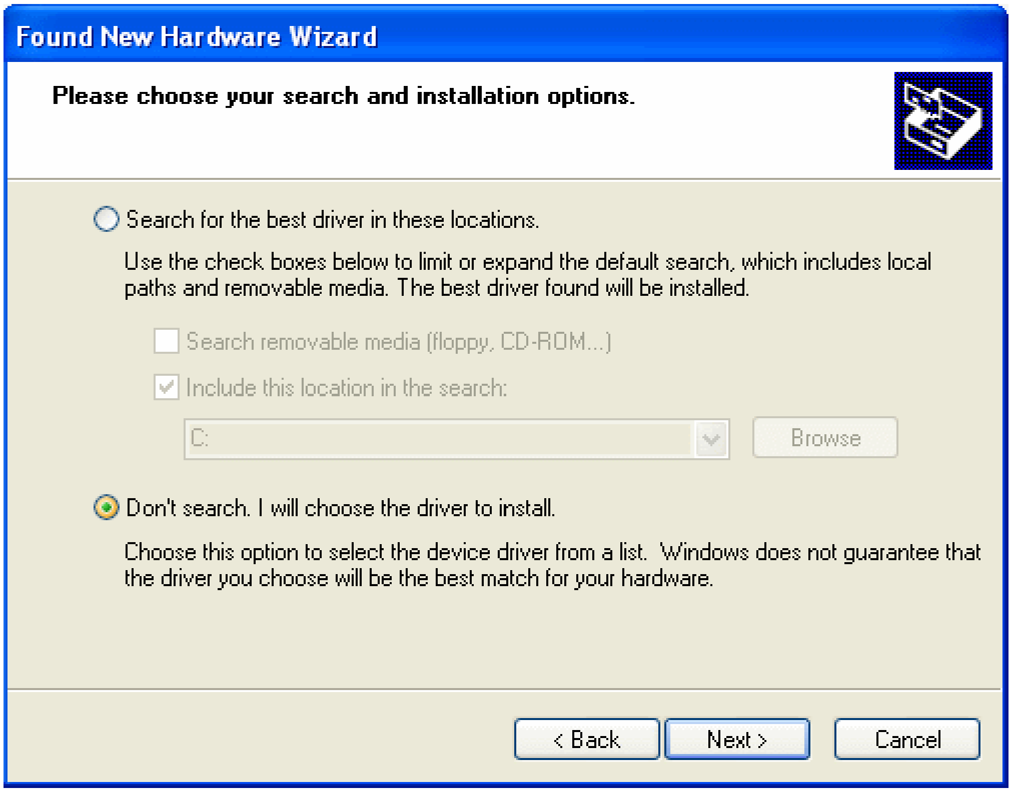
» Windows will detect the AWG within a few seconds.

» A window will pop up asking you to find the drivers for the USB controller device. Select **Yes, this time only** and then click on **Next**.

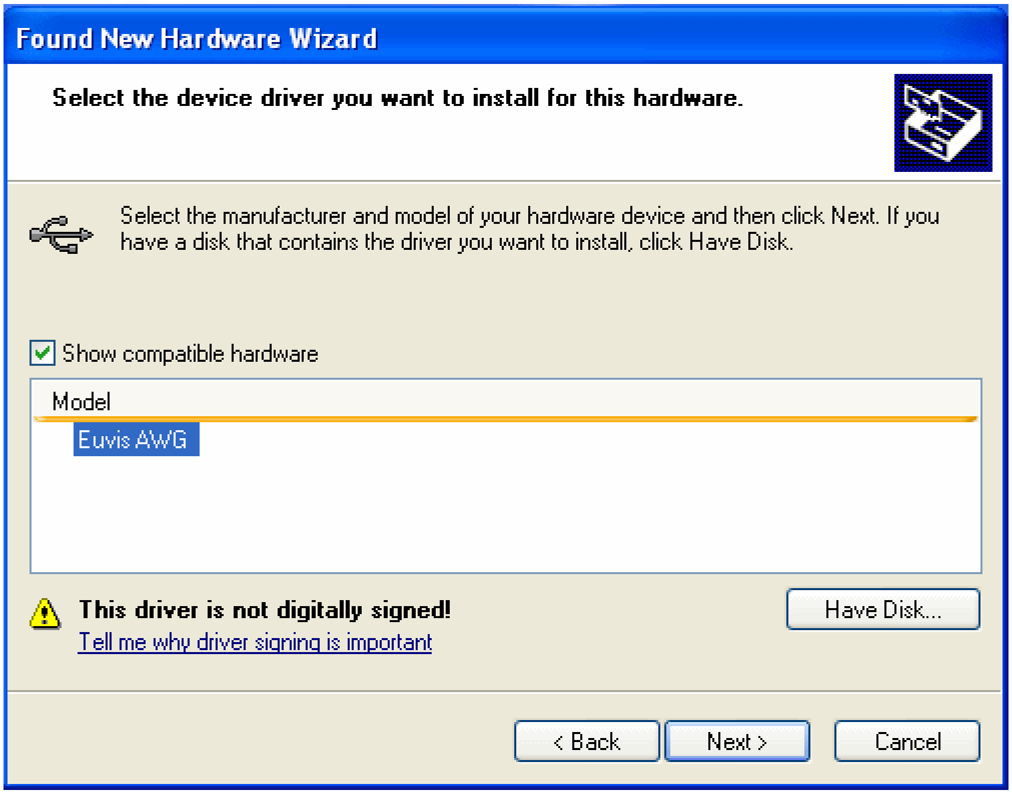
» Select **Install from a list or specific location (Advanced)** and then click **Next**.



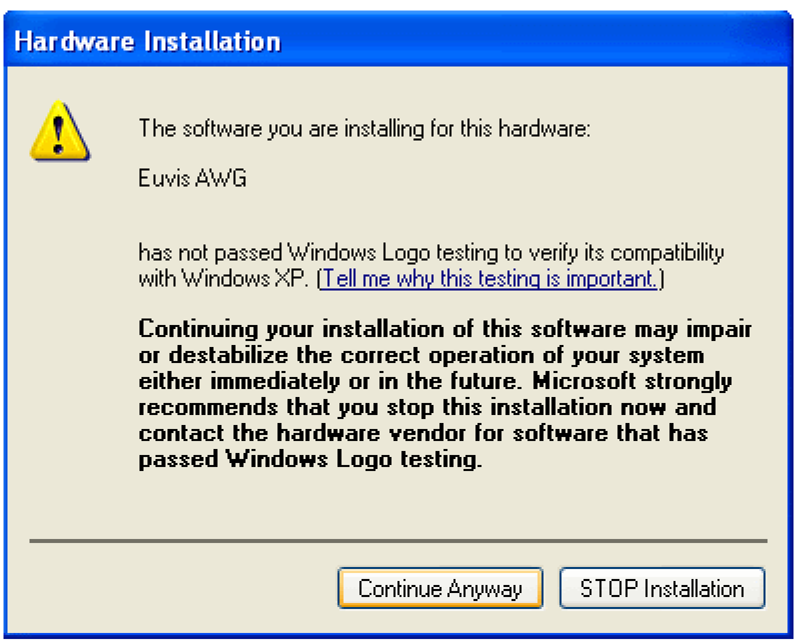
» Select **Don't search. I will choose the driver to install** and then click **Next**.



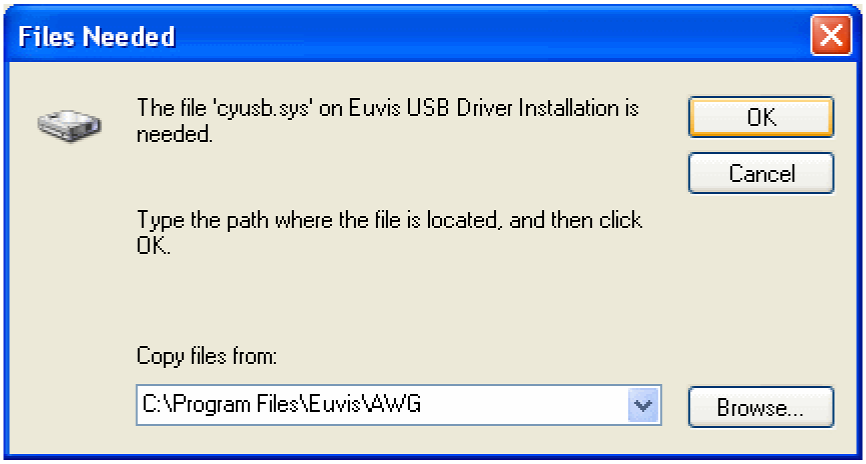
» Under Model, select **Euvis AWG** and then click **Next**.



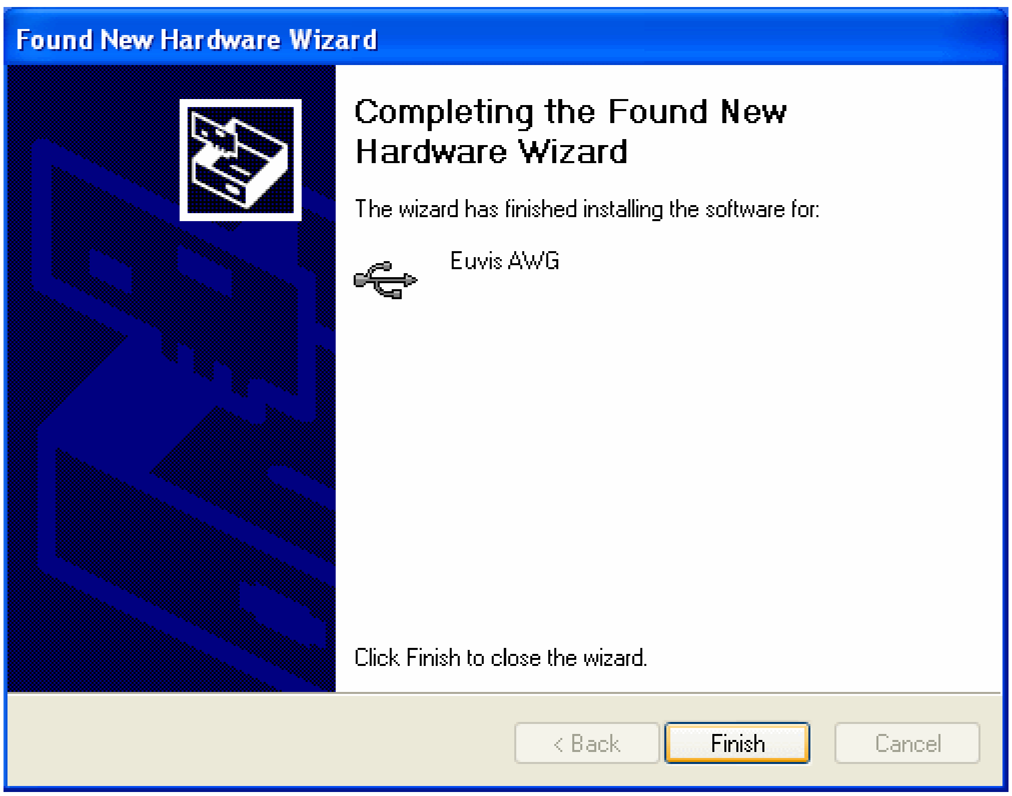
» A window will pop up giving you a warning about the driver. Please click **Continue Anyway**.



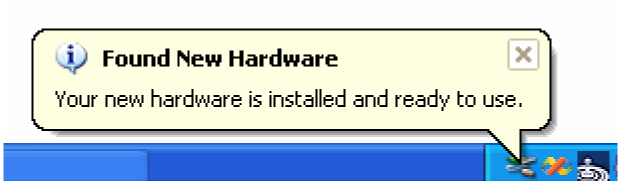
» After a few seconds, the installer will ask you for the drivers for the onboard USB controller. In the "Copy files from" box enter **C:\Program Files\Euvis\Driver**.



» Click **Finish**.



» Windows will inform you that the device is ready for use.



» You have now completed software setup.

The first time you run the AWG_WIN.exe application, you will get the following message:

Simply click OK. The missing file awg_win0.ini will then be generated, and the GUI application will launch. You will not see this message again.

» If you ever reinstall the software or install a newer version, please uninstall the old software first, and then follow the software setup from the beginning.

Hardware Requirements-

- AWG Module
- PC with USB 2.0 port
- USB A to B Cable
- Clock source capable of 3 dBm signal
- Oscilloscope
- Spectrum analyzer
- standard US Electrical Outlet, or
- Power supplies capable of 12V output with at least 4A current
- Two 16 to 18 AWG wires
- At least 3 SMA coaxial cables for clock source, oscilloscope and spectrum analyzer
- More SMA coaxial cables will be needed if you intend to use synchronization

Hardware Setup-

Outputs

» Connect your oscilloscope and spectrum analyzer to the **OUTP** and **OUTN** SMA connectors. They are 50-ohm back terminated, so you can connect one or both, in either order.

Clock Source

» Set your clock to the desired frequency with 3 dBm power. Connect your clock to the **CKIP** SMA connector at the upper left corner of the board.

Trigger (optional)

» For single-board free running operation, you do not need a trigger signal. For multi-module operation, or for operation in a triggered mode, you may want to use a trigger source so that the master module will generate a SYNCO signal periodically. Set your trigger source, such as a function generator, to a 1 kHz square wave with 1 Vpp amplitude and 0.5 V DC offset (for a 50 ohm load; set the amplitude higher if your setting is for high impedance). Connect your trigger source to the **TRIG** SMA connector on the left side of the board.

Fan

» Set up a small fan to blow air across the module. This is necessary to keep the board cool, which prevents overheating and allows the module to operate consistently.

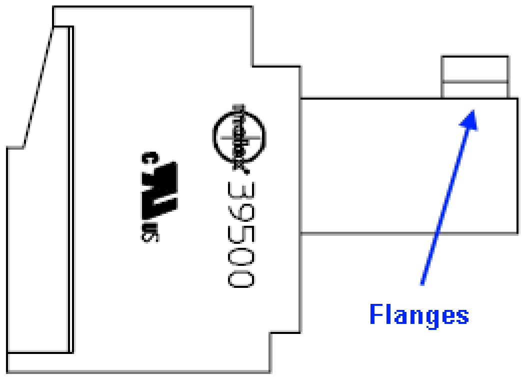
Power Plug Setup

For your convenience, we have provided a plug and header configuration for power supply connections to the AWG board. Also included is power adapter already connected to the plug. But, if you wish to use your own power supply, or if you disconnect the plug from the wires, it is essential that you connect wires from the plug to the power supplies in the correct order.

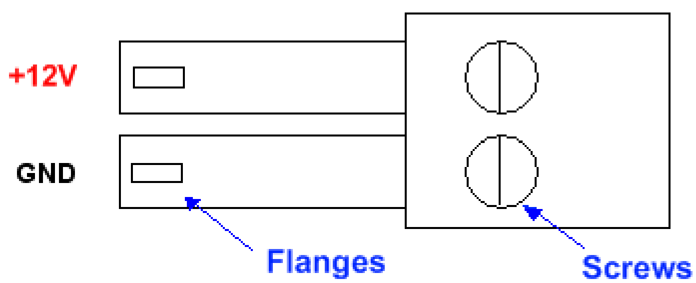
The power plug is rated for a maximum of 300V and 8A. The plug manufacturer suggests wire sizes that range from 16 to 24 AWG but we recommend you use either 16 or 18 AWG wires to reduce significant voltage drops across the wires. The strip length for the wires should be at least 5mm (0.2 inches) for the end that will go into the power connector.

We suggest that you either color code the wires or label them so that you do not plug them into the wrong power supplies. We cannot emphasize enough the importance of correct power supply connections. Any wrong connection can potentially damage the board permanently.

The plug has flanges so that there is only one orientation in which the plug and the header can connect.



With the screws and flanges facing upward and the flanges on the left side, connect the wires as indicated. Please strip at least 5mm (0.2 inches) of the wires and insert them into the square holes on the opposite side of the flanges (on the right side in the image below).



After inserting each wire, use a 5/64" slotted screwdriver to fasten the wire in place. Repeat this procedure for the other wire. The +12V wire from the power adapter has white dashes to differentiate it from the other wire.

Power Supply

The power adapter is rated for 100V-240V AC input. Please plug it into an electrical outlet. Insert the other end into the header at the bottom of the AWG board. The plug has flanges so that there is only one orientation in which the plug and the header can connect.

» If you have not already done so, please insert the power plug into the power header near the lower right corner of the board. You should have already set up the plug so that the wires are connected to the power supply in the correct order, and you should already have a fan blowing air across the board.

If you are using your own power supply, then after you have connected the power and turned it on, you should use a multi-meter to check the voltages at the connector just in case there is a significant voltage drop across any of the wires. You can check the voltage at the connector by probing the screw which you used to fasten the wires. The ground should be near 0 V. The +12 V should all be close to the stated value. If any reading at the connector is not correct, adjust your power supply until the reading is correct.

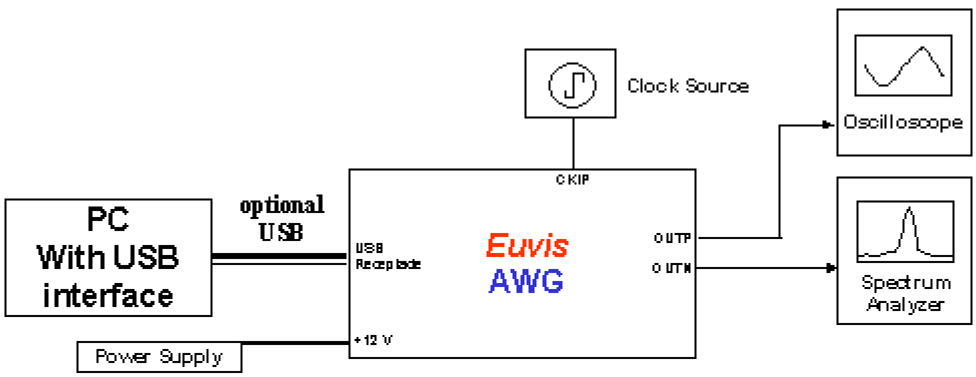
NOTE: When unplugging the power plug from the header, pull the actual plug and not the wires to avoid disconnecting the wires from the plug.

USB port

» Connect the AWG board to your PC with a USB cable only if you have already installed the software and drivers.

» If you have not installed all the software and drivers, please **do not** connect to the USB port. You can still operate the board in standalone mode.

» Schematically, the connections should look like the diagram below:



Power On

» **Turn on** the power supply. You may turn on the power supply in any order.

» If you have done everything correctly, then you should see the built-in waveform output on your oscilloscope and your spectrum analyzer after a few seconds.

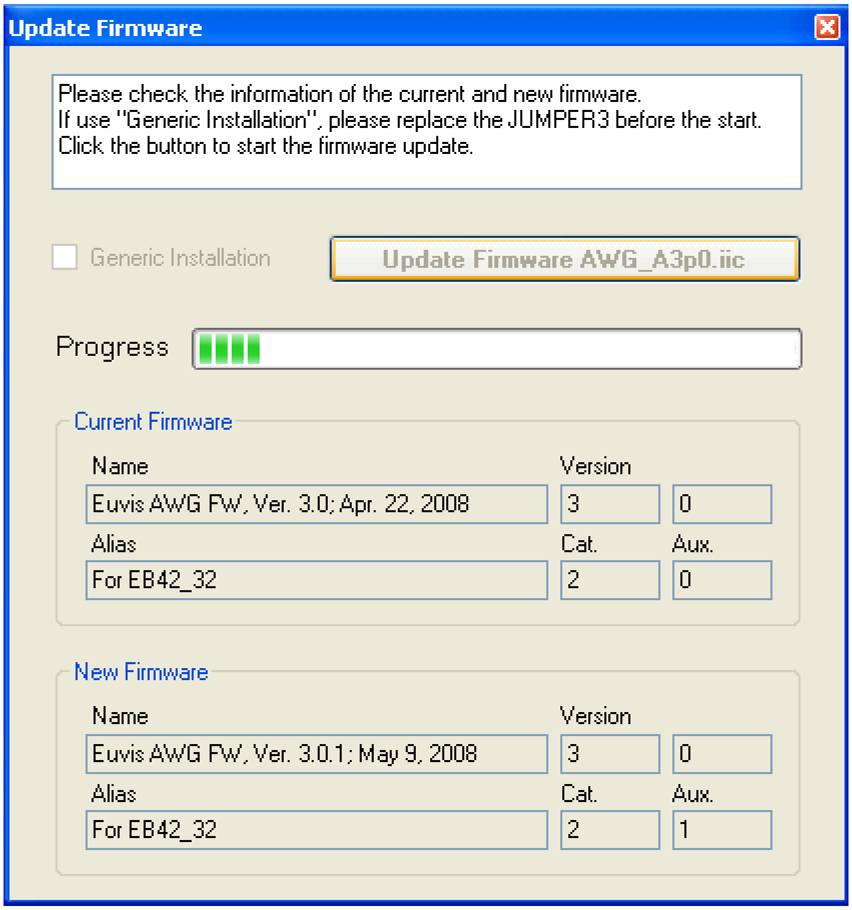
The built-in waveform is pre-stored in the firmware and is automatically loaded at startup. Even with a PC connected to the AWG, the built-in waveform loads and runs before the PC downloads a new waveform to the board, although you must press the Restart button in the AWG_WIN.exe application once you have started the program.

The built-in waveform is a 256-point cyclic linear chirping waveform sweeping from 1/256 to 1/16 of the clock frequency with frequency update at 8 clock cycles.

» The built-in waveform can be customized upon request.

Firmware Update -

All new firmware will be accompanied by a GUI application so that the functions will match. To update the firmware, please follow the instructions below.



Place the new .exe GUI application and the new .iic firmware files in the program directory you were already using.

For normal firmware updating, power up the module with a jumper on JP3 and click on PHY --> Firmware Update to bring up the above dialog box. The dialog box will display the current version of the firmware on the device and the new firmware that you will write to the ROM. Confirm that the name of the new firmware identifies that to which you want to update, and then simply click on the **Update Firmware** button to begin the process. It will take approximately 30 to 60 seconds to complete.

To register the changes to the application, power down the module and exit the GUI. Power up the board, and wait a few seconds for the demo waveform to show up.

After the demo waveform has appeared, re-launch the GUI, or click on PHY --> USB Reconnect. To verify that you have successfully updated the firmware, click on PHY --> Firmware Info, and it should show the new firmware information.

If there is a problem, we recommend that you perform a Generic Installation. To do this, with the module off, remove the jumper (shunt) from JP3 on the board. Power up the device and launch the GUI application. Go to PHY --> Firmware Update to open the dialog box. Place the jumper back on to JP3. Confirm that the name of the new firmware identifies that to which you want to update, and then simply click on the **Update Firmware** button to begin writing to the ROM. To register the changes to the application, exit the application, power down the board, and then power up the board. Wait a few seconds for the demo waveform to appear, and then launch the GUI application. When you open the program again and click on PHY --> Firmware Info, it should show the changed firmware information.

Software Update

Before installing a new version of the GUI software, please first ensure that your waveform files (.wfa, .xml, .uda) are saved in a directory different from your program directory.

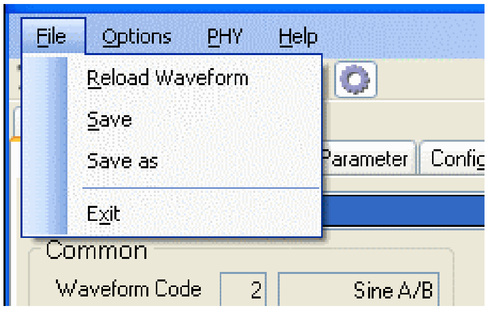
Then, uninstall the existing GUI application from the Add/Remove Programs control panel in Windows. The uninstallation process will remove the application, including waveform files in the program directory. After the GUI application has been uninstalled you may proceed with the Software Setup to install the new GUI application.

Application Window -

Menus -

At the top of the AWG application window is the menu bar, from which you can set several options. Below the menu bar is a toolbar with several frequently used functions.

File :



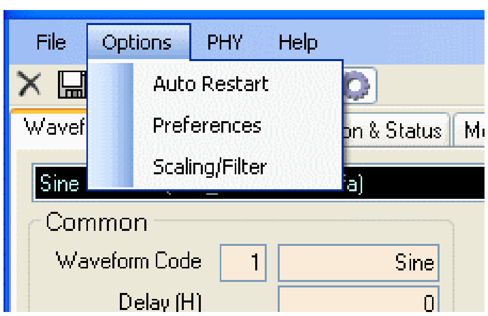
Save

Saves all parameters into current waveform file. The current waveform is the one listed in the drop-down box in the Waveform panel. For detailed instructions on saving waveforms please see the Save Waveform page.

Save As

Saves all parameters into a new waveform file. When you click on "Save As", a new dialog box will open listing all the parameters. For detailed instructions on saving waveforms please see the Save Waveform page.

Options

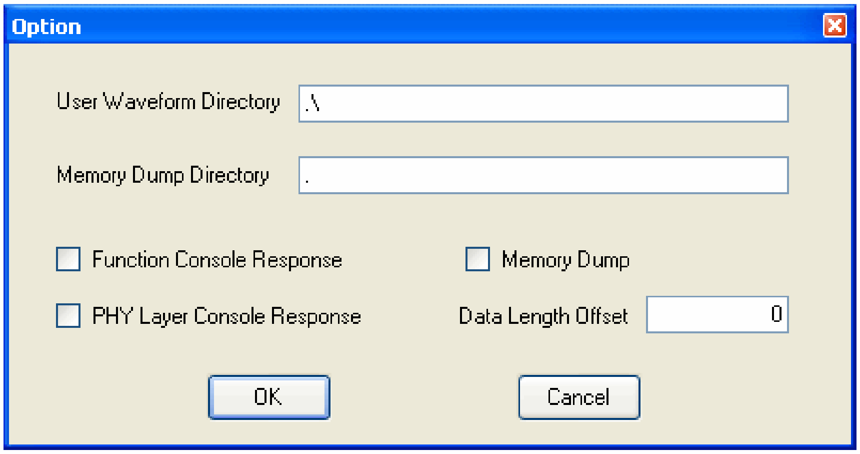


Auto Restart

Automatically restarts AWG memory when you load a new waveform.

Preferences

Brings up the Preferences window:



Waveform Directory is the location where you want the waveform files to be saved to. The ".\" represents the root directory of the program, which you specified during installation. The default is C:\Program Files\Euvis\AWG

Memory Dump Directory is the location where you want the memory dump to be saved to.

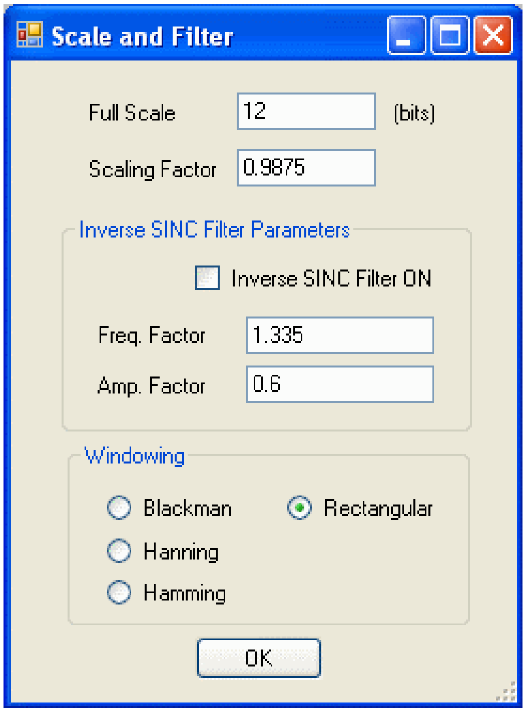
Function Console Response enables or disables the Function console. The Function console displays all user commands that are sent to the board.

PHY Console Response enables or disables the PHY Layer Console. The PHY Layer Console, at the lower right corner of the AWG window, displays all information that the application receives from the AWG board.

Memory Dump enables or disables memory dumping. When memory dumping is enabled, the application will dump all data sent to the board in a .dat file located in the aforementioned Memory Dump Directory.

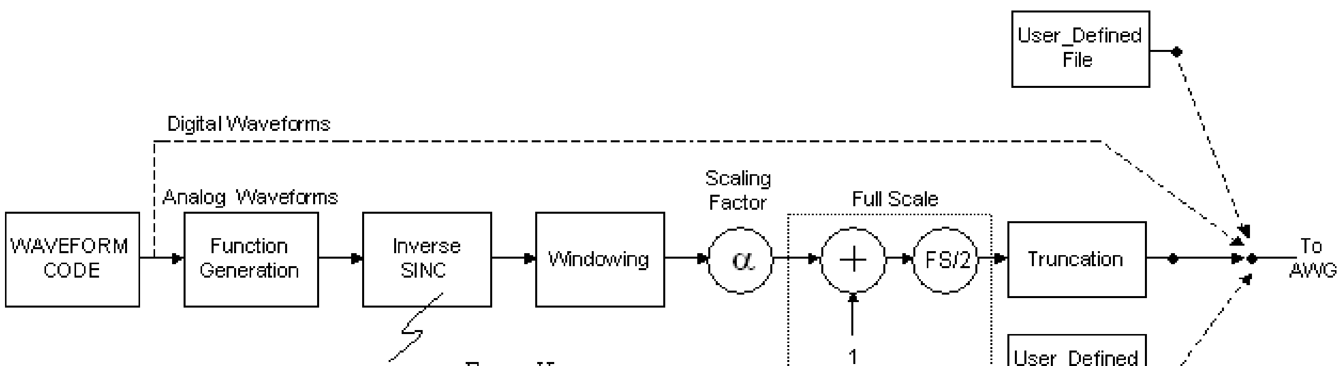
Data Length Offset specifies how many memory addresses to leave empty at the end of the waveform for more stable output. At around 2 GHz clock frequency, set this to "C30".

Scaling/Filtering



Due to windowing effects from using time-limited waveform data, the frequency spectrum of analog waveforms may appear to have been convolved with a Sinc, and this filter is designed to mitigate the effects of that modulation.

This feature allows you to modify the inverse filter frequency factor, the subsequent window applied, and the full-scale amplitude applied to the analog waveform. Digital waveforms, such as ramp, pulse, and user-defined waveforms, are not affected by these settings. Analog waveforms, which are calculated internally with double precision, go through this inverse sinc filter, windowing, a linear scaling factor, and then are translated to full scale amplitude.



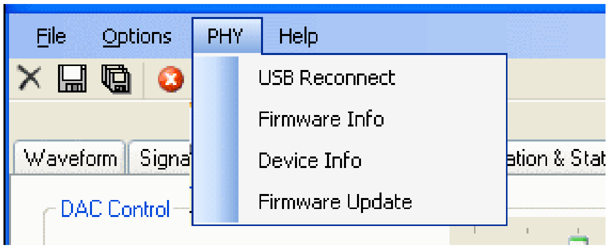
$$H(K) = G \cdot Sinc^{-1}\left(\frac{F \cdot \pi \cdot K}{M}\right)$$

$$M = 1024, F = 1.335, and \quad G = 0.6$$

We recommend that you leave the Scaling Factor unchanged at 0.9875, and only use this option to turn on or off the filter. If desired, you can change the type of windowing applied to your analog waveform. Please remember to re-download the waveform data if you change any of these settings.

For an example using the windowing filter, please see Filter in the Waveforms section.

PHY

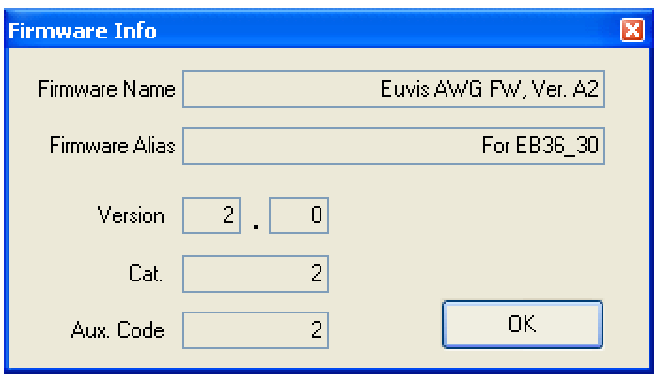


USB Reconnect

If you accidentally unplug the board while the program is still running, you can use this to reconnect without having to exit the program.

Firmware Info

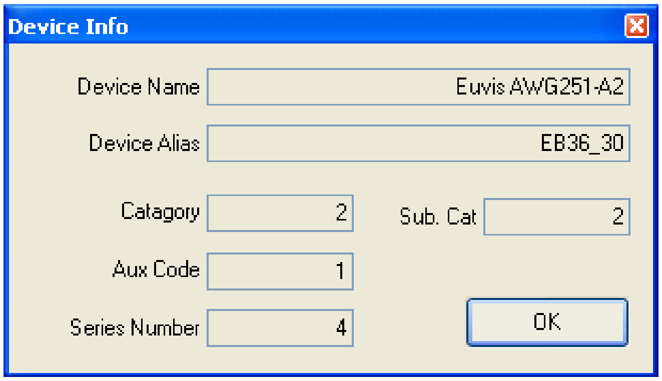
Provides information about the firmware:



The most important items here are Firmware Name, Firmware Alias and Version number. If you contact us for support, we will need to know what these are.

Device Info

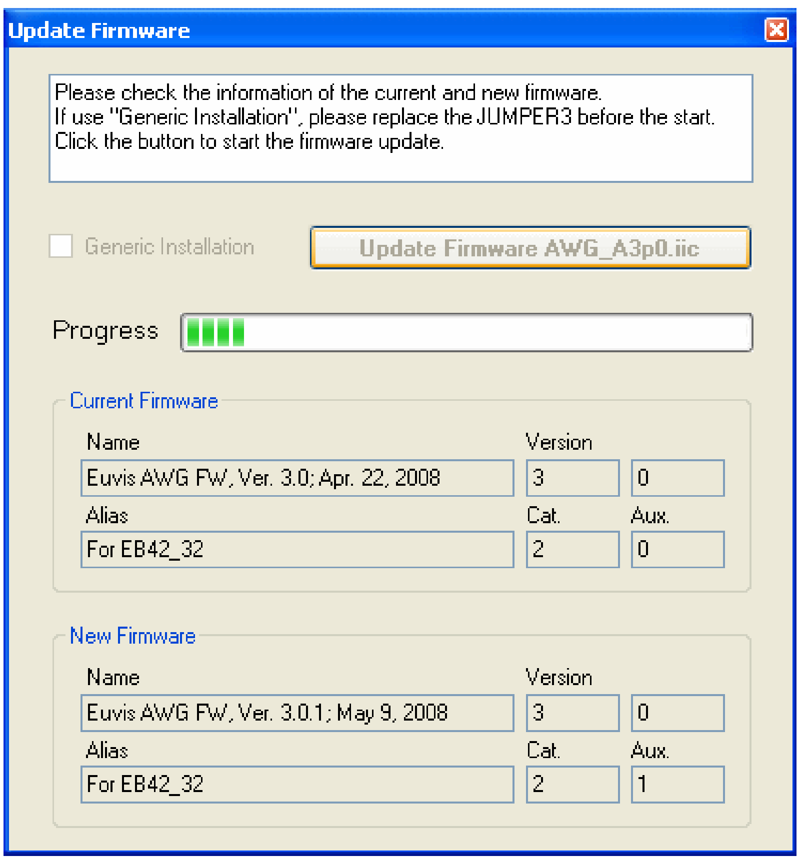
Provides information about the AWG hardware:



The most important items here are Device Name, Device Alias and Series Number. If you contact us for support, we will need to know what these are.

Firmware Update

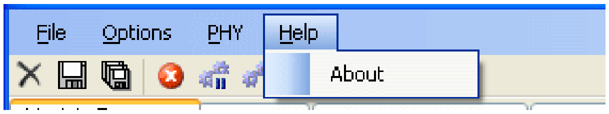
Advanced feature to update the firmware by flashing the ROM:



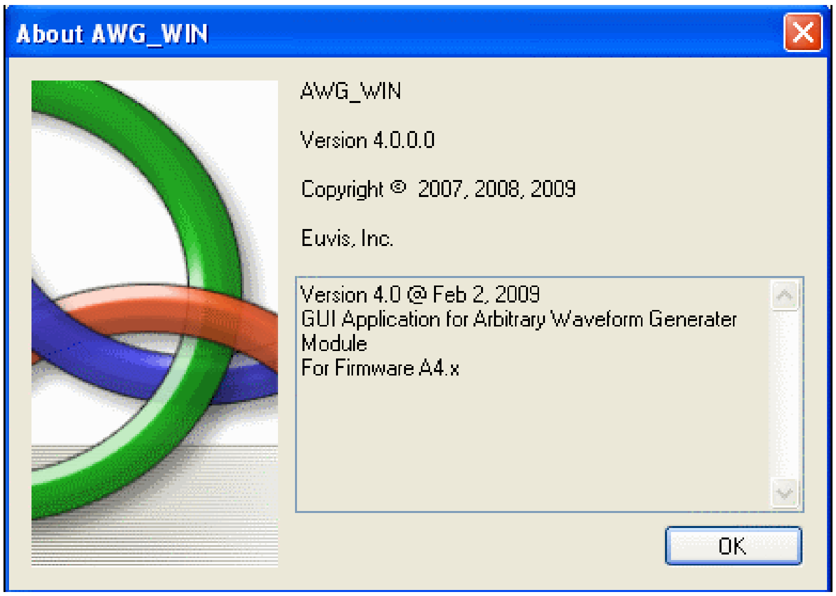
For normal flashing, power up the AWG board with a jumper on JP3 and click on PHY --> Firmware Update to open the above dialog box. The dialog box will show you the current version of the firmware on the device and the new firmware that you will flash on to the ROM. Simply click on **Update Firmware** to begin the process. It will take approximately 30 to 60 seconds to complete flashing. To register the changes to the application, close the application, power down the board, and then power up the board. When you open the program again and click on PHY --> Firmware Info, it will show the new firmware information.

If there is a major problem with a new firmware, we will recommend to you to do a Generic Installation. To do this, with the AWG power supply turned off, remove the jumper from JP3 on the board. Power up the device and open the AWG application. Go to PHY --> Firmware Update to open the dialog box. Place the jumper back on to JP3. Now click on **Update Firmware** to begin flashing. To register the changes to the application, close the application, power down the board, and then power up the board. When you open the program again and click on PHY --> Firmware Info, it will show the changed firmware information.

Help



About



This provides information about the version of software you are running.

Toolbar

The toolbar is located below the menu bar. Some commonly used functions are placed here. These functions are:
Delete Waveform,
Save Waveform,
Save Waveform As ...,
Abort,

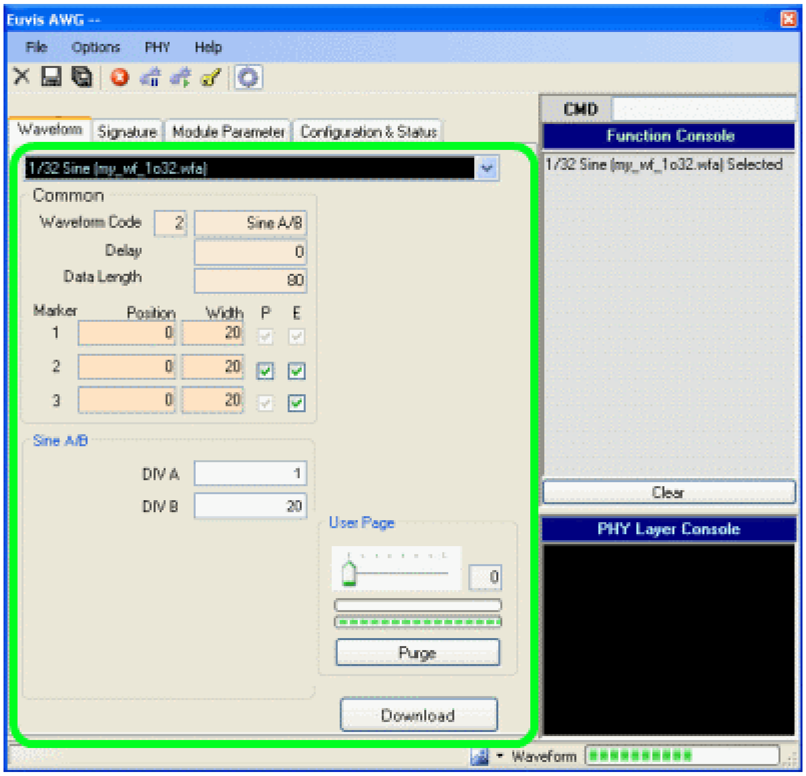
Arm&Ready,
Restart,
Trigger,
Endless Loop, and
Alternate timing.

Alternate Timing

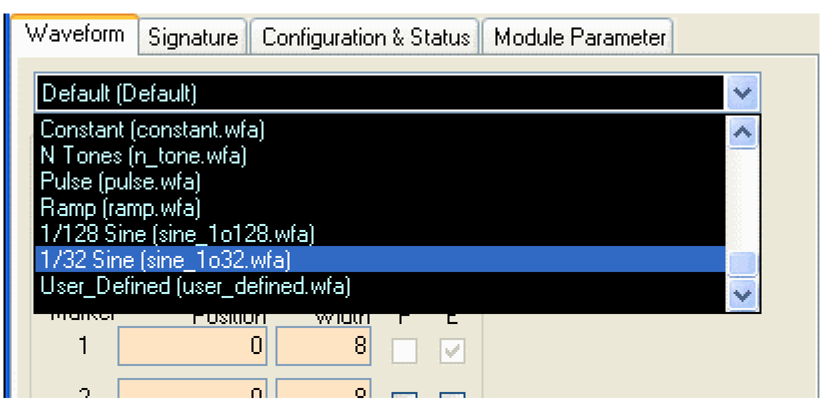
The DAC clock is reset whenever you download a new waveform to the AWG module. Sometimes the clock is reset so that the timing is not matched with the sample timing used when recording the signature settings. In this event, click the A button in the toolbar for Alternate timing. The alternate timing button will stay on until you click it again to turn it off.

Waveform Panel -

The Waveform panel is accessed by the 1st tab. In this panel you can load various prestored waveforms, adjust parameters of those waveforms, and select the memory address page.

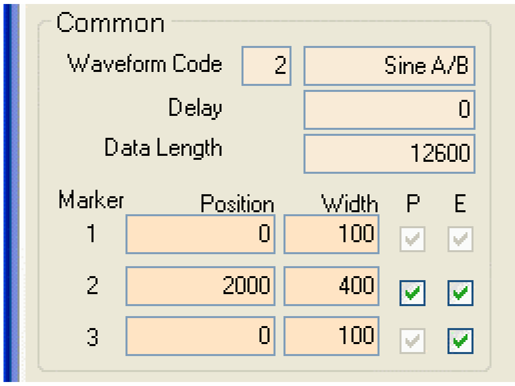


Waveform Selector



Click on the drop-down list to select a waveform. The software comes with several pre-defined waveforms. You can make your own waveforms to put on this list. To learn how to do this, please go to Save Waveform in the Waveform Files section.

Waveform Parameters



Waveform Code

Specifies the style of waveform to be implemented. Each style has corresponding specific parameters to adjust the waveform. These parameters may be entered in a special section below Common parameters only after you have entered a waveform code. For more details about this, please go to Waveforms in the Waveform Files section.

Delay

Specifies waveform delay in terms of memory addresses. This is defined in hexadecimal code. For example, if delay were set at "5", the first 5 memory addresses would have the first data value of the waveform, and then at the 6th memory address the waveform amplitude changes would begin.

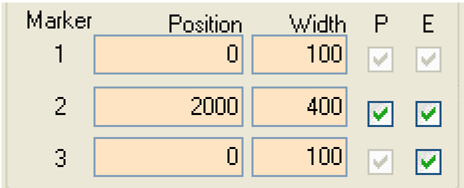
Data Length

Specifies how many memory addresses to use for the waveform. This is defined in hexadecimal code, and the Data Length must be a multiple of 10 (hex) for the AWG252 and 20 (hex) for the AWG452. The minimum data length is frequency dependent, with higher clock frequencies requiring longer data lengths, but generally you should use a data length of at least "400" (hex, corresponding to decimal 1024). However, for proper output, you must make sure that the Data Length is equal to or greater than the number of values of your waveform. For a detailed discussion on this parameter, please see the Data Length page in the Operation Details section.

NOTE: If you wish to use dynamic paging, which offers up to 15 user pages, there are some considerations to make with your Data Length. The Data Length per uPage memory partition is "80000" (which is decimal 524288) in the AWG452. If you use many uPages in your waveform, there will be fewer user pages available for you to use.

Marker

The AWG module offers 3 marker signals. Markers are useful for identifying waveforms, since the markers are generated simultaneously with the waveforms. These parameters specify where and how long to output the markers.

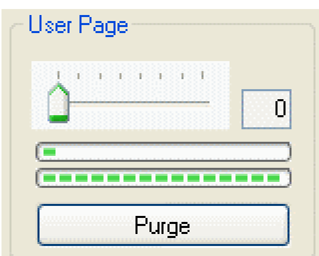


The Position box specifies the data point where you would like to start the marker. The Width specifies how many data points you would like the marker to be active. The "P" checkbox is only available for Marker 2 and specifies whether you would like active low markers. The "E" checkbox is only available for Marker 2 and Marker 3 and is used for enabling or disabling those markers. The different characteristics of the three markers are detailed in the Parameters page of the Operation Details section.

The Marker is useful as a diagnostic tool. Markers are ONLY output when there is waveform generation. If you have marker enabled but you have no Marker signal then it means that there is no waveform output.

Paging

The AWG module features memory address paging. You can set which user page to use here in this panel (for details, see the description of Multiple waveforms with paging).



Each user page can load a separate waveform, but the more pages you have, the fewer memory addresses you have available for each waveform -- meaning the maximum data length of each waveform is shorter. Or, the longer the data length of each of your waveforms, the fewer the number of user pages you will have available. The total number of memory addresses is fixed.

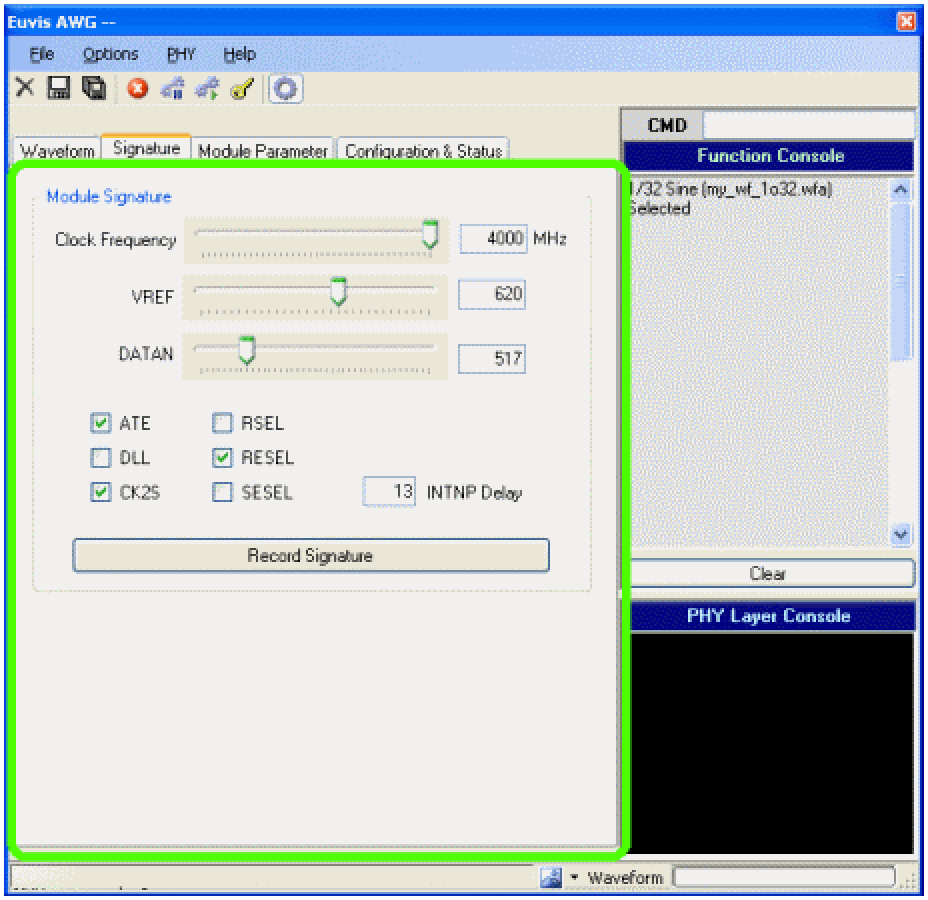
Download Waveform

Click this button, located near the lower edge of the waveform panel, to send current waveform parameters to the board. You will have to use this button whenever you change any waveform parameters.

After a waveform finishes downloading, you should press the Restart button.

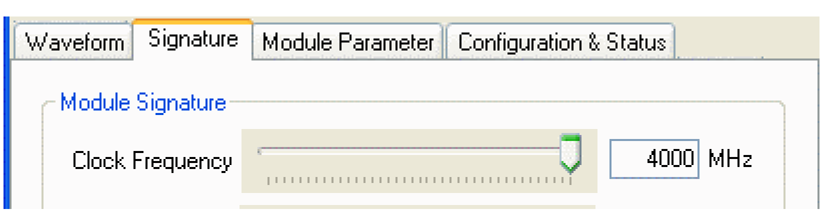
Signature Panel -

The Signature Panel is accessed by clicking the 2nd tab. In this panel you can adjust and save settings to optimize the waveform output at various clock frequencies.



When operating the AWG at different clock frequencies, the settings may need to be changed for optimal performance, especially at higher clock speeds, when the timing and thresholds of data samples are less forgiving and need to be adjusted for different clock frequencies. We have already provided you with a signature file in the main AWG directory. The signature file contains preset signatures for frequencies from 500 MHz to 2.5 GHz for the AWG252 or from 1 GHz to 4 GHz for the AWG452, assuming that the CKD1 and CKD2 SMA's are left open (not 50 ohm terminated or connected to anything else). When using the AWG, you might need to fine tune the preset signatures, and you can record your changes.

Clock Frequency



Slide the Clock Frequency bar to match the frequency of your clock source. If settings had previously been stored for this frequency (or a nearby frequency) in the signature file, the settings will be recalled.

It is very important that you enter the input clock correctly, matching your actual clock source, because data for the output waveforms are calculated based on this. Ensure that the clock frequency is right and that the units are also correct.

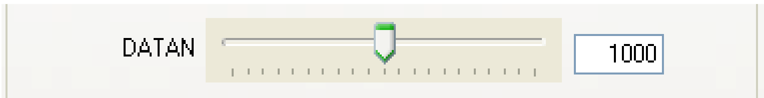
Whenever you change the clock frequency, you should always re-download the waveform data to the module.

For the AWG252, it is recommended to set the clock frequency within the range of 500 MHz to 2.5GHz.
For the AWG452, it is recommended to set the clock frequency within the range of 1 GHZ to 4 GHz.

VREF

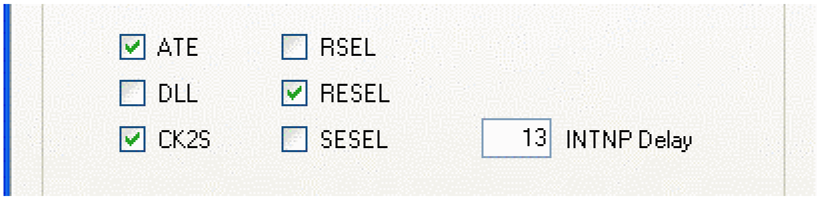
Sets the reference voltage, which is a threshold for binary data. Typically the lower end works better, but setting VREF too low may result in data loss.

DATAN



Sets an average voltage for the data. This control does not affect anything significantly, although adjusting it may provide slight performance improvements at higher clock frequencies.

ATE



Toggles between using the falling edge and using the rising edge to sample the address. ATE is an internal ASIC parameter and is used to control stability of the waveform.

Often one setting will be superior to the other for several waveforms at a given frequency, although a few waveforms may require the opposite setting. If you find that your waveform is not stable, try switching this option on or off.

If you change this value, you must save the setting by pressing the Record Signature button before you can restart the waveform again, or else the change in setting will be lost.

DLL

Stands for Delay Lock Loop. DLL is an internal Cypress CPU parameter and is also used to control stability of the waveform. If you find that your waveform is not stable try switching this option on and off.

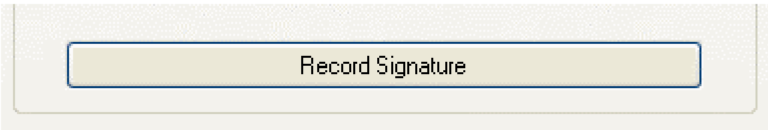
If you change this value, you must save the setting by pressing the Record Signature button before you can restart the waveform again, or else the change in setting will be lost.

INTNP Delay

Sets the timing before the end of a waveform for the module to prepare the next waveform. The correct values ensure seamless continuity from one waveform to the next, and have already been saved in the signature file provided to you. Decreasing this number may delay the output of subsequent waveforms. If you change the clock frequency and notice that this INTNP Delay number has changed, you must re-download the waveform data to the module.

If you change this value, you must save the setting by pressing the Record Signature button before you can restart the waveform again, or else the change in setting will be lost, and you must re-download the waveform data to the module for any change in the INTNP Delay setting to take effect.

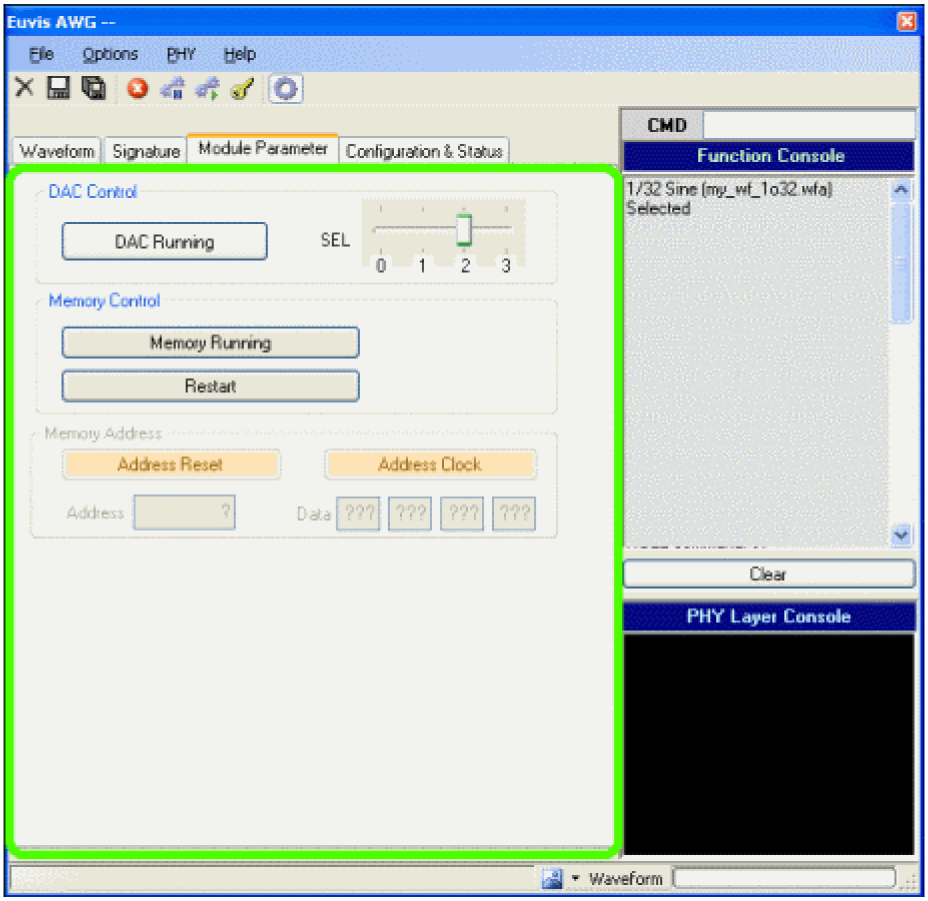
Record Signature



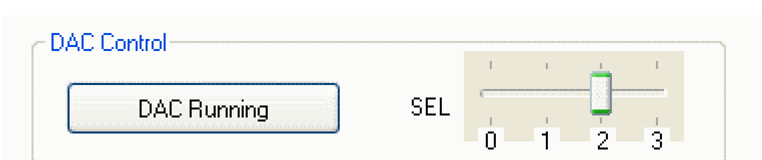
When you have adjusted the settings optimally for your current clock frequency, click on the Record Signature button to save these settings. These settings will overwrite any previously saved settings for your selected frequency. The signature file must be in your working directory so that, in the future, these settings will be recalled whenever you slide the clock frequency bar to this frequency (or to a higher frequency, up to the next recorded signature).

Module Parameter Panel -

The module parameter panel is accessed by the third tab below the tool bar and menu bar, near the top of the application window. From this panel you can set the MUXDAC sampling select window, set whether to stop or start the DAC and memory, and see the actual memory data.



DAC Control

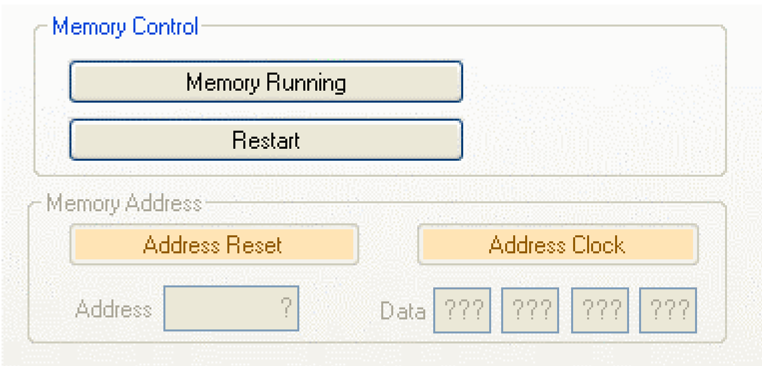


DAC Running/Reset

Stops or resets the DAC. While the DAC is stopped, the waveform is no longer output.

SEL

Selects which of four possible times to sample the data. The optimal SEL setting changes for different clock frequencies, and at least one SEL setting allows data to be sampled near the most open part of the eye between signal switching. The SEL setting for the currently set frequency can be saved in the signature panel.

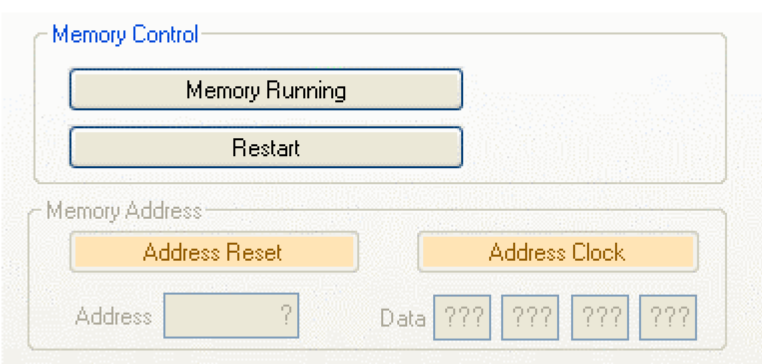


Memory RUN/STOP

Stops the waveform at the current memory address. When memory is stopped, you can go through each memory address manually and see the waveform data stored in each address using the **Address Reset** and **Address Clock** buttons.

Memory RUN/STOP is also useful if you wanted to change the input clock frequency in the middle of a waveform. You click the **Memory STOP** button to temporarily stop the waveform, change the input clock frequency, and then click on **Memory RUN** button (same button) and the waveform will start again at the new frequency. Be sure also to adjust the value in the **Clock Frequency** bar in the Signature panel to ensure that the output will match the values desired.

Memory Address



Address Reset

When memory is stopped, you can click this button to view the waveform data for each of four memory banks at the first memory address.

Address Clock

Increments memory location by 1 so you can view output data values of sequential memory addresses. Only enabled when memory is stopped.

Address

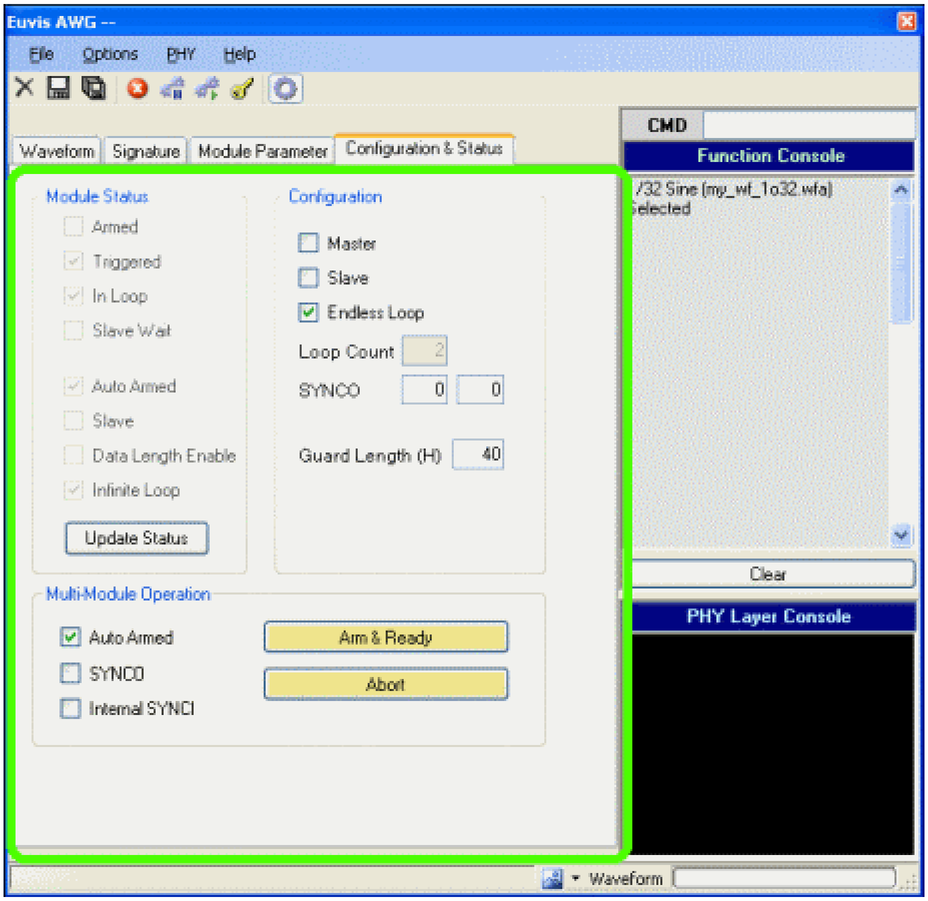
Displays the current memory address. Not available unless memory is stopped by clicking on **Memory STOP**. You may type in a memory address in this box to view the data at that address.

Data

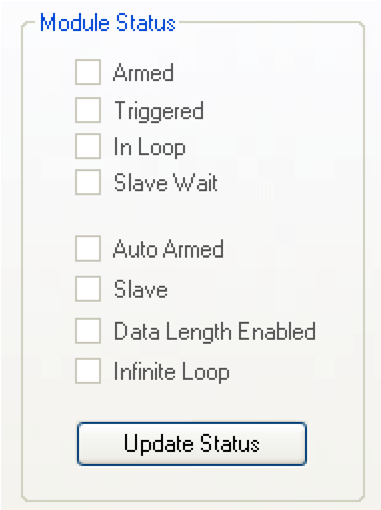
Displays the waveform data word in hexadecimal of the current memory address for each of four memory banks.

Configuration and Status Panel -

The Configuration and Status Panel is accessed by the 3rd tab. In this panel you can check on the internal status of the AWG module, configure the module as a master or slave for multi board operation, and set the guard length to delay a waveform.



Internal States



Module Status

Displays the current status of the AWG board. The status is not automatically refreshed. In order to see the most current status of the module, you must poll its status by clicking on the **Update Status** button.

Auto Armed - displays if board has Auto Armed enabled. When Auto Armed is enabled, the board automatically enters the "Armed" state after the "Loop Done" state. For more information about the internal states please see the Internal States page.

Armed - displays if board is in the "Armed" state.

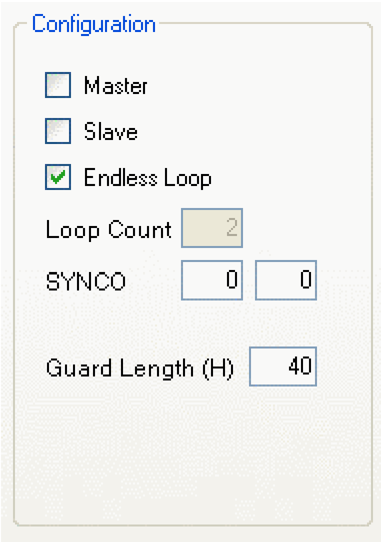
Triggered - displays if board is in the "Triggered" state.

In Loop - displays if board is in the "In Loop" state.

Slave - displays if board is in Slave mode. When this option is not indicated, the board is in either Master mode or Free-running mode.

Slave Wait - displays if module is in the "Slave Wait" state, which is equivalent to the "Armed" state of the Master or Free-running modes. In the "Armed" state of the Master or Free-running modes, the board waits for the *TRIG* signal to begin waveform output, while in the "Slave Wait" state of the Slave, the board waits for the *SYNCI* signal to begin waveform output.

Configuration



Master

When enabled, puts board into Master mode.

Slave

When enabled, puts board into Slave mode.

Endless Loop

Enables continuous waveform output. With Endless Loop, the Loop Count parameter is ignored.

Loop Count

Specifies the number of waveforms to output for each trigger signal. If you want an infinite loop or continuous output, please check the box for **Endless Loop**.

SYNCO

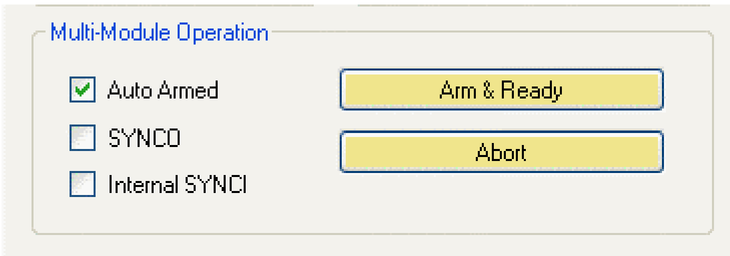
When the board is in Master mode, the board will output the SYNCO signal from the SYNCOUT SMA connector on the boards. When in Slave mode, the SYNCO option does not do anything. For more information about the *SYNCO* signal, please go to the Synchronization page.

Guard Length

Sets the delay before the first waveform. This is useful for synchronizing the master and slave waveforms under multi-board operation. The default Guard Length is 40 (hex). Since the slave waveform is slightly delayed behind the master waveform, simply enter a shorter Guard Length for the Slave, usually between 20 and 38 (hex) if the Master has a Guard Length of 40 (hex). A positive Guard Length is necessary to ensure the first waveform appears correctly. Please use a multiple of 4.

After you change the Guard Length, you must re-arm the board to see the waveform correctly.

Multi-Module Operation



Auto Armed

When enabled will automatically arm the AWG so that after the "Loop Done" state, the board will automatically go into the "Armed" state and be ready for another trigger signal. You cannot change this option if the board is in Slave mode.

If disabled, the AWG will go into the "Disarmed" state after the "Loop Done" state in which case the user will have to click on the **Arm & Ready** button to put the board into the "Armed" state. For more information about the internal states please see the Internal States page.

SYNCO

When this is checked and the board is in Master mode, the board will output the *SYNCO* signal from the SYNCOUT SMA connector on the boards. When in Slave mode, the SYNCO option does not do anything.

Internal SYNCI

When this is enabled in Master or Standalone mode, the internal *SYNCI* signal will mirror the *SYNCO*. You cannot change this option if the board is in Slave mode. For more information please go to the Synchronization page.

Arm & Ready

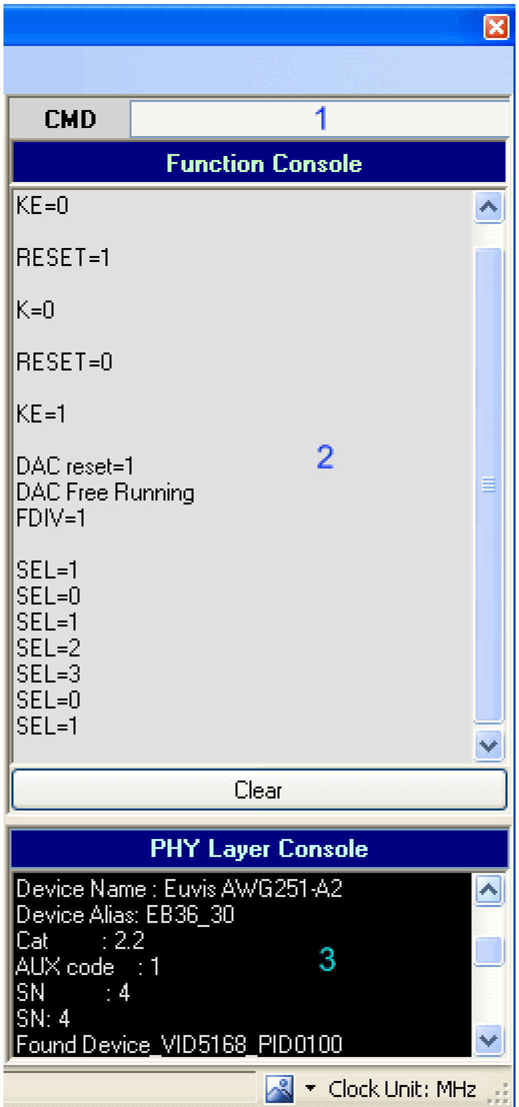
You will need to click on this button to put the board in the "Armed" state whenever you load a new waveform. Additionally, if **Auto Armed** is not checked you will have to click on this to put the board into "Armed" state manually after the board finishes outputting the waveform loops.

Abort

You may click on this button at any time to put the board into the "Disarmed" state.

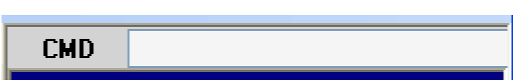
Consoles -

The consoles are located on the right side of the AWG application window. From the consoles, you can enter commands and see the last commands entered. This is useful to test your commands when preparing to write your own custom application, but it is not necessary to use the consoles if you simply use the functions in the left side of the graphical user interface.



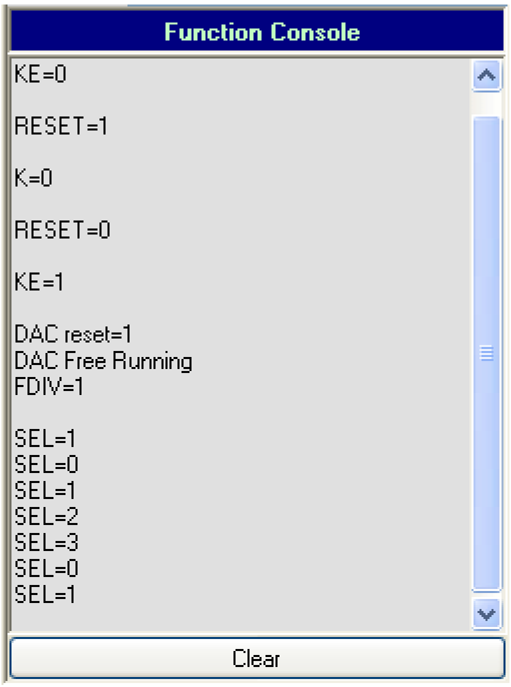
1. Function Command Line
2. Function Console
3. PHY Layer Console

Function Command Line



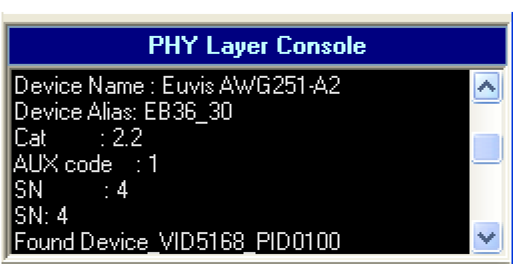
You can manually enter commands to control the AWG here. This is also useful when you are writing your own application and want to test if your sequence of commands will work.

Function Console



Displays commands that have been entered into the AWG. Only available if Function Console Response in Preferences is checked.

PHY Layer Console



Displays device and firmware information from the AWG. Only available if PHY Console Response in Preferences is checked.

Operation Details -

Parameters -

The waveforms that the module will output are defined by parameters common to all waveform styles and by parameters specific to the chosen waveform style. The following are common parameters:

- Data Length
- Delay
- Markers

Before going into detail about the parameters, a brief explanation of how the module works will help you understand the parameters better. All waveform data are stored on the SRAM bank on the board.
The AWG252 RAMs are capable of storing 4 × 1,048,576 data words, each of size 12 bits. Each value in a waveform is represented as a 12-bit data word. Since there are 12 bits available, you have a maximum amplitude resolution of 4096 (2^{12}).
The AWG452 RAMs are capable of storing 8 × 1,048,576 data words, each of size 12 bits. Each value in a waveform is represented as a 12-bit data word. Since there are 12 bits available, you have a maximum amplitude resolution of 4096 (2^{12}).
The AWG801 RAMs are capable of storing 16 × 524,288 data words, each of size 11 bits. Each value in a waveform is represented as a 11-bit data word. Since there are 11 bits available, you have a maximum amplitude resolution of 2048 (2^{11}).

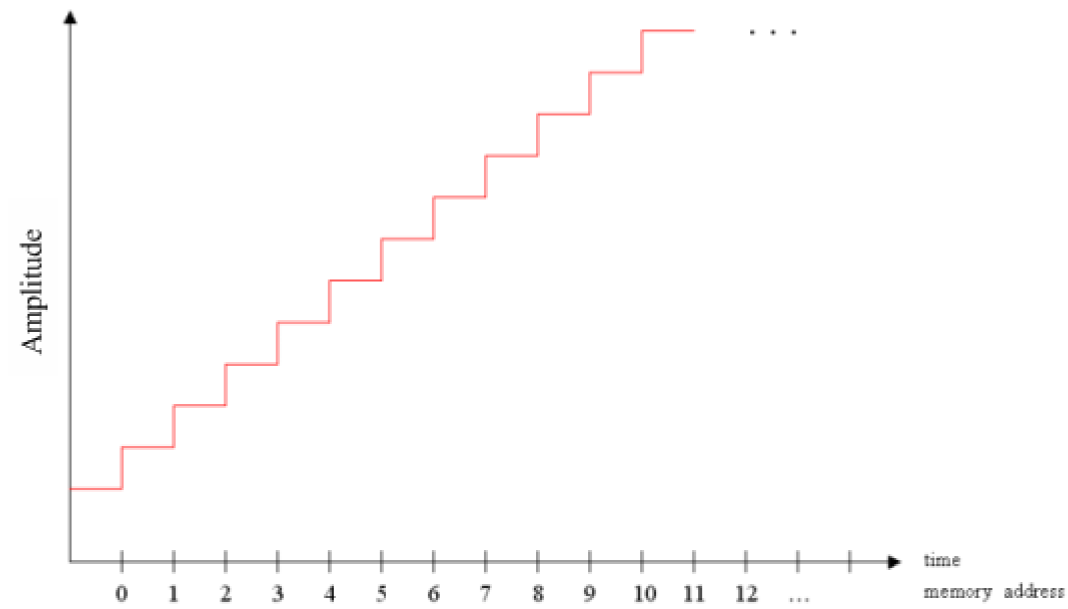
The actual chip creating the waveforms is one of the following: AWG252: the Euvis MD652D Digital-to-Analog Converter with 4:1 Multiplexers (MUXDAC). The waveform output data rate is the same as the input clock rate.
AWG452: the Euvis MD681S Digital-to-Analog Converter with 8:1 Multiplexers (MUXDAC). The waveform output data rate is the same as the input clock rate.
AWG801: the Euvis MD662H Digital-to-Analog Converter with 4:1 Multiplexers (MUXDAC). The waveform output data rate is twice the input clock rate.

The module creates waveforms by storing sequences of waveform data words onto the RAM bank (maximum of 4 × 1,048,576 words) and then "feeds" these data words into the MUXDAC chip. For the AWG252 and AWG452, a new data word is sent to the MUXDAC and the analog waveform will be updated at every clock; each set of values will also be output for one clock. For the AWG801, a new data word is sent to the MUXDAC and the analog waveform will be updated at every half clock; each set of values will also be output for one half clock.

You can change the default time per data word by changing the Oversampling in the Module Parameter Window. The MUXDAC is phase coherent, so the output waveform maintains the same phase even when output waveform frequencies change.

Data Length

The Data Length parameter should always be greater than or equal to the number of data points in your waveform. It specifies the memory addresses to make available *for the waveform*, so this is the parameter that will determine how many waveform data values will be output. You can imagine the waveform as a series of data points. Each data value is output for 1 clock until we reach the last value in the waveform cycle. Go to the Data Length section to get a detailed discussion regarding these two parameters.



If the total number of waveform data values is greater than the Data Length, then the last waveform data value that is output is not the one you specify but, instead, is the Nth value, where N equals the Data Length. For example, if your waveform had 100 data values, but your Data Length were 40_{16} (decimal 64), then the actual last output value would be only the 64th value.

Therefore, assuming there is no delay, for proper waveform output based on your desired values, the necessary condition is:

Number of Data Values ≤ Data Length

In addition, recall that there are only 4 × 1,048,576 total memory addresses available (maximum memory depth). But, some memory addresses are reserved, so:

AWG252: Number of Data Values ≤ Data Length ≤ 3,932,160
AWG452: Number of Data Values ≤ Data Length ≤ 7,864,320

Please see the Data Length section for a more detailed discussion regarding the relationship between these two parameters.

Delay

Delay specifies how many memory addresses to keep at the starting value before the waveform starts. For example if your Delay were "5" then the first 5 memory addresses would store values corresponding to the starting data value, and then at the 6th memory address, the waveform would begin. Please note that the delay is counted toward the data length. So if you set data length to 32 and had 30 data values but also set delay to 5, then the last 3 values in your waveform would not be output, since the delay and the data values together exceeded the data length.

Therefore the new Data Length relation becomes:

AWG252: Number of Data Values + Delay ≤ Data Length ≤ 3,932,160
AWG452: Number of Data Values + Delay ≤ Data Length ≤ 7,864,320

This is the guideline to use for determining proper Data Length.

Marker

The module features three markers, each with different characteristics. They are generated simultaneously with the waveform data, so the markers help you identify when a waveform is output. The marker settings are in the Waveform panel, just below the Data Length box, and allow you to choose when, and for how long, the signals go active.

Markers are useful as diagnostic tools. Markers are ONLY output when there is waveform generation. If you have a marker enabled but you detect no Marker signals, then it means that there is no waveform output.

Marker 1 features 1.8 V TTL (rail-to-rail) logic level. Marker 1 is generated as part of the waveform data, so there is no delay and, in fact, slightly precedes the corresponding waveform output, due to the delay from the latching of the data in the MUXDAC to the analog output. This latency is about 2-3 nanoseconds at the earliest sampling of the waveform data. The signature file records settings for optimal timing of samples, which may result in a waveform output latency up to 3 clocks later than the latency at the earliest sampling opportunity. For instance, if the input clock is 1 GHz, the latency changes by 1 nanosecond at each change in the SEL setting on the Signature panel.

Marker 2 and Marker 3 both feature 3.3 V LVTTL (rail-to-rail) logic levels. These higher voltage levels are compatible with most instruments, but there is a slight delay, about 6-8 nanoseconds, compared with Marker 1. Markers 2 and 3 can be disabled, and when disabled Marker 3 remains low. To disable Markers 2 or 3, uncheck the corresponding E enable box in the Waveforms panel.
Marker 2 also features optional reverse polarity if an active low marker is desired. To make Marker 2 active low, uncheck the P positive polarity box in the Waveforms panel.

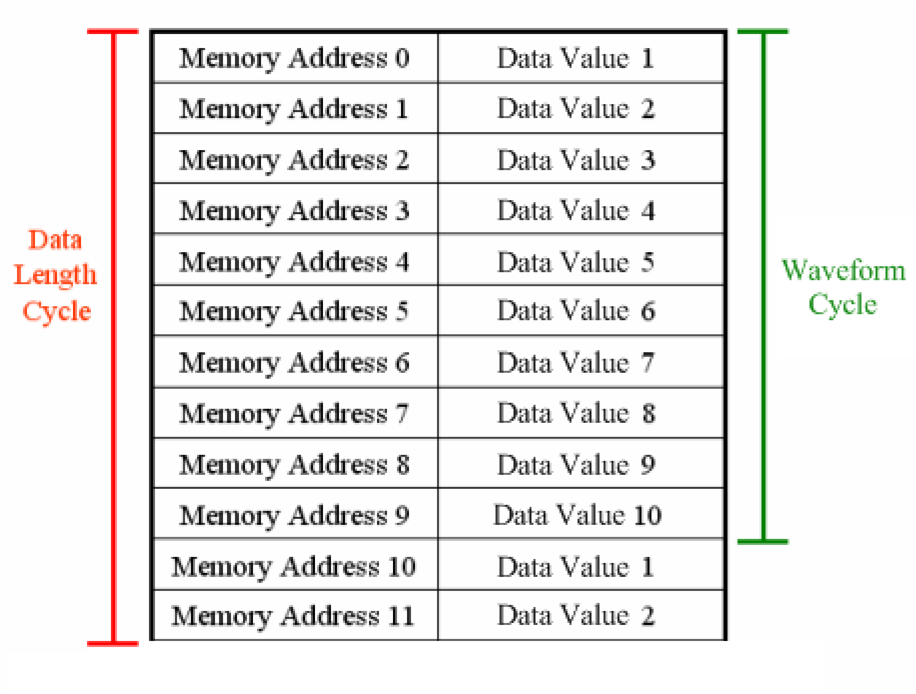
Data Length -

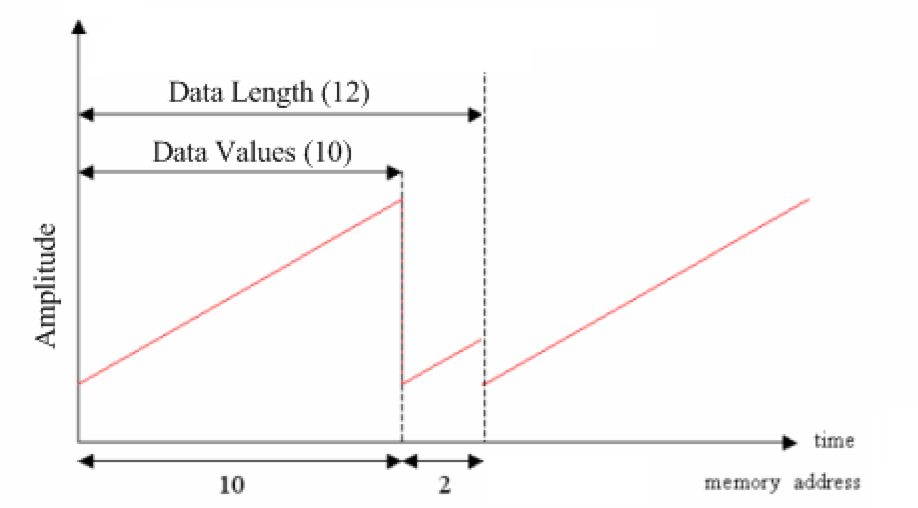
The choices of data length and the number of data values that you have will affect the output of your waveform. First, recall that the data length should be greater than or equal to the number of values. As a result, we have the following necessary condition:

AWG252: Number of Data Values + Delay $\leq$ Data Length $\leq$ 3,932,160
AWG452: Number of Data Values + Delay $\leq$ Data Length $\leq$ 7,864,320

The above must be satisfied in order to have proper waveforms. The AWG stores amplitude data as data words. There are a total of 4 × 1,048,576 memory addresses available, each able to store a 12-bit data word, in the AWG252. There are a total of 8 × 1,048,576 memory addresses available, each able to store a 12-bit data word, in the AWG452. 1/16ths of those addresses are reserved for internal use. The other 15/16ths can be used for one waveform or partitioned for use with multiple waveforms facilitated by the dynamic user paging feature. The module stores the data in special ways depending on the data length and memory depth parameters. An example will help you see exactly what is going on.

Assume that you have specified 10 data values in your waveform and you have set Data Length = 12 (satisfying the necessary condition, above). By setting those parameters, you will have made 12 memory addresses available for your waveform data (Data Length value). The module stores the 10 user specified values in the first 10 memory addresses, but since there are still two more memory addresses available for data (Data Length - number of data values), it copies the first 2 values again in the remaining memory addresses made available by the Data Length. The images below illustrate the example of a ramp waveform:



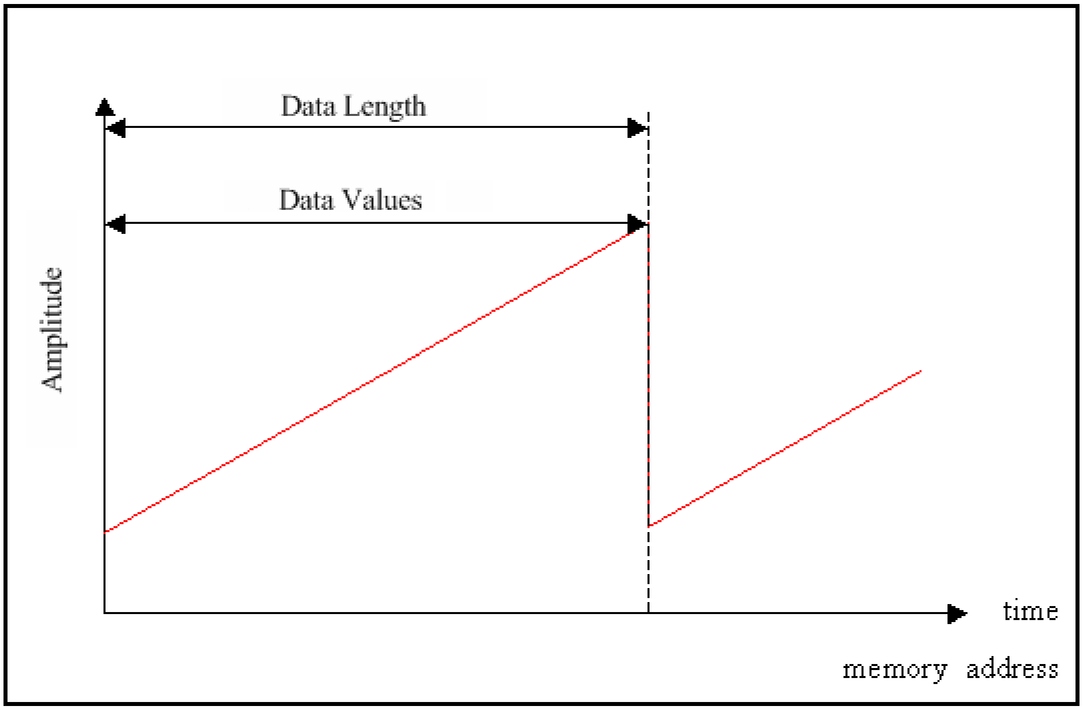


When you operate the module with these settings, the module will run through all 16 memory addresses in the Data Length cycle and then start over again.

In the discussion below, there are two cycles to be aware of: Data Length Cycle and waveform cycle.
One **Data Length cycle** goes through all of the memory addresses made available by the Data Length parameter. In the above example, one complete Data Length cycle would be 12 memory addresses.
One **waveform cycle** goes through every value in the user defined waveform. In the above example, one complete waveform cycle has 10 memory addresses, since there are 10 data values.

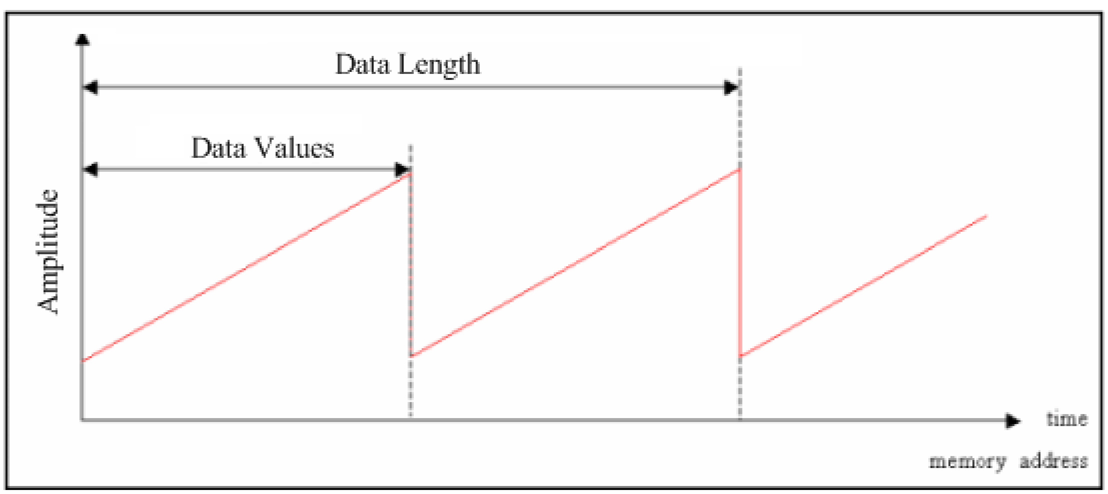
There are three different ways you can manipulate the output with the Data Length parameter.

1. Data Length = Number of Waveform Data Values



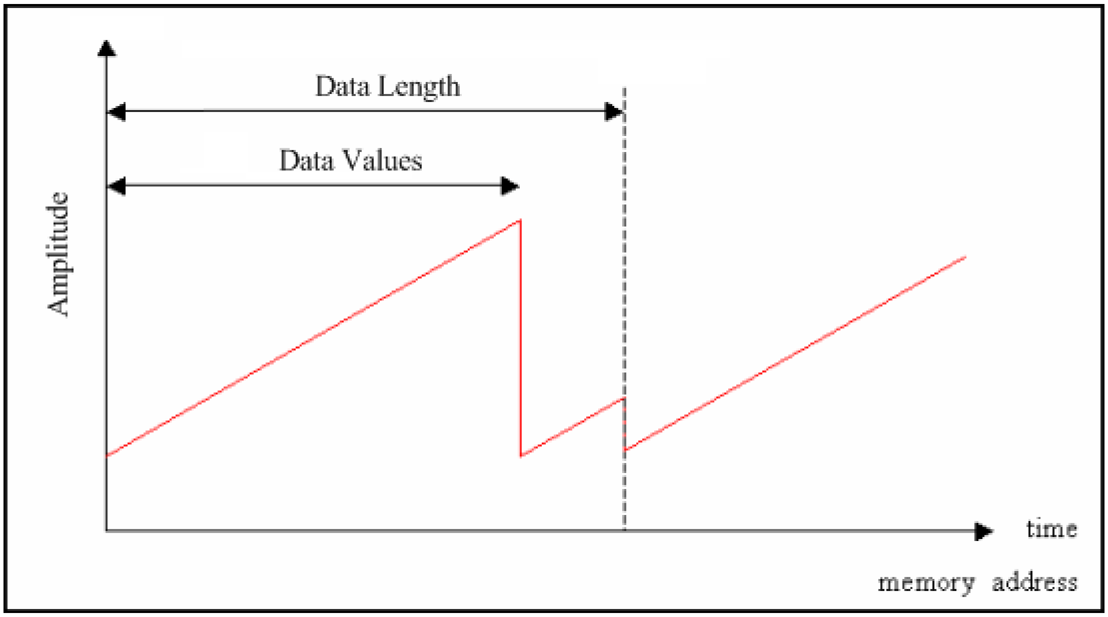
In one Data Length cycle, the module will run through all of the desired values exactly once. For example, if you had 64 data values, and Data Length = 64, then all of the 64 values would be output once during each Data Length cycle.

2. Data Length = n × Number of Waveform Data Values



In one Dat Length cycle, the module will run through all of the desired data values n times. The image above illustrates the case of n = 2. For example, if your waveform had 32 data values and Data Length = 64, then each value in the waveform would be output twice during each Data Length cycle.

3. Data Length > Number of Waveform Data Values, but Data Length is not n × Number of Waveform Data Values



In one Data Length cycle, the module will run through a whole number multiple of waveform cycles and repeat a fraction of the waveform cycle again until the start of the new Data Length cycle. For example, if you had 50 data values and Data Length = 64, then all 50 data values would be output during the first waveform cycle, but only the first 14 data values would be output during the next waveform cycle, since you then would have reached the end of the Data Length cycle.

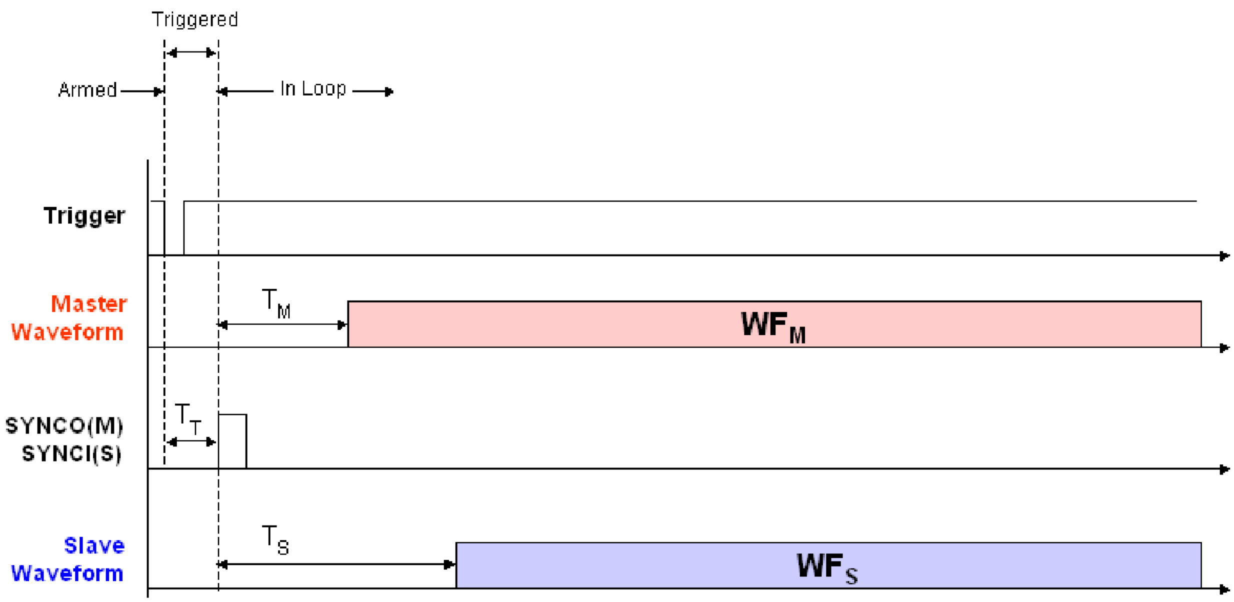
Internal States -

The module uses internal states to produce controllable outputs, and these states are also necessary for synchronization between two boards. There are five internal states that the board can be in:

- **Disarmed** - in this state, the board does not do anything.
- **Armed** - in this state, the AWG waits for a trigger signal which will commence the waveform output sequence
- **Triggered** - this is the delay state between the trigger signal and the beginning of waveform output
- **In Loop** - this is the state where the AWG outputs the waveforms. The board stays in this state until the Loop Count is complete
- **Loop Done** - this is the state right after "In Loop". This is where the board decides to either go to the "Disarmed" state or go to the "Armed" state

The module produces waveforms ONLY in the "In Loop" state and must go through several other states before it actually reaches that state. The different internal states are important not only because they are necessary for synchronization but also so that the module itself is able to output waveforms in a controllable fashion.

Below is a timeline of the different signals within the board and how they affect the internal states. Two states, "Disarmed" and "Loop Complete", are not shown.

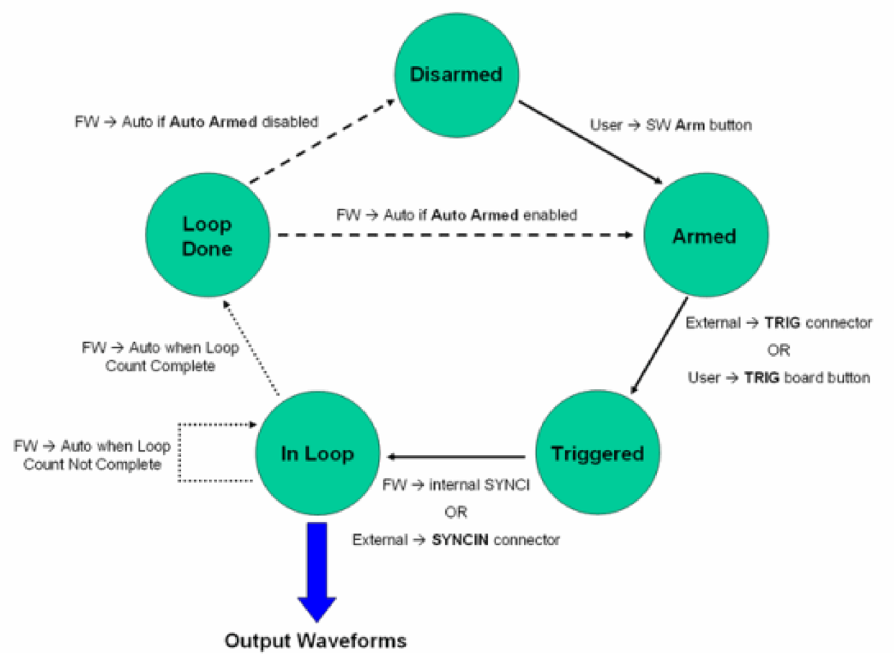


Basically, the goal is to be able to reach the "In Loop" state in a repeatable and orderly way. The signals above are the specified signals that are able to change the module's state from one state to the other. It is assumed that the module is already in the "Armed" state. The falling *TRIG* signal changes the state from "Armed" to "Triggered" and also causes the *SYNCO* signal to rise after 1-3 µs (T_T). The *SYNCO* signal is mirrored by the *SYNCI* signal (see the Synchronization page for a more detailed discussion), and when *SYNCI* signal signal rises, the module's state changes from "Triggered" to "In Loop" and begins waveform generation.

It is important to note that only the right combination of signals and states will have an effect on the AWG. For example, if the board were in the "Armed" state and you made the *SYNCI* signal rise, the board would not do anything because it was not in the "Triggered" state. But if the board were in the "Triggered" state and you made the *SYNCI* signal rise, then this would cause the board to change to the "In Loop" state and produce waveforms.

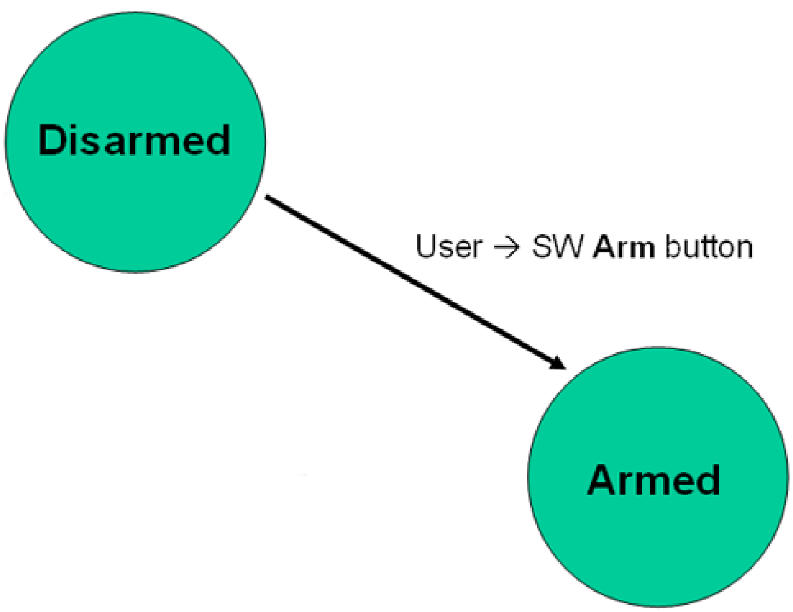
States are changed automatically through the firmware, manually by user action, or through an external input such as clock signals. For example, changing the board state from "Disarmed" to "Armed" requires that the user click on the **Arm & Ready** button in the GUI application, but changing the state from "In Loop" to "Loop Complete" is done automatically without user intervention or external input.

The diagram below shows how the board operates in regards to its different states. Waveform generation as stated previously is done in the "In Loop" state. "SW" stands for software and refers to the GUI application. "FW" stands for firmware and refers to the onboard firmware, which operates automatically.



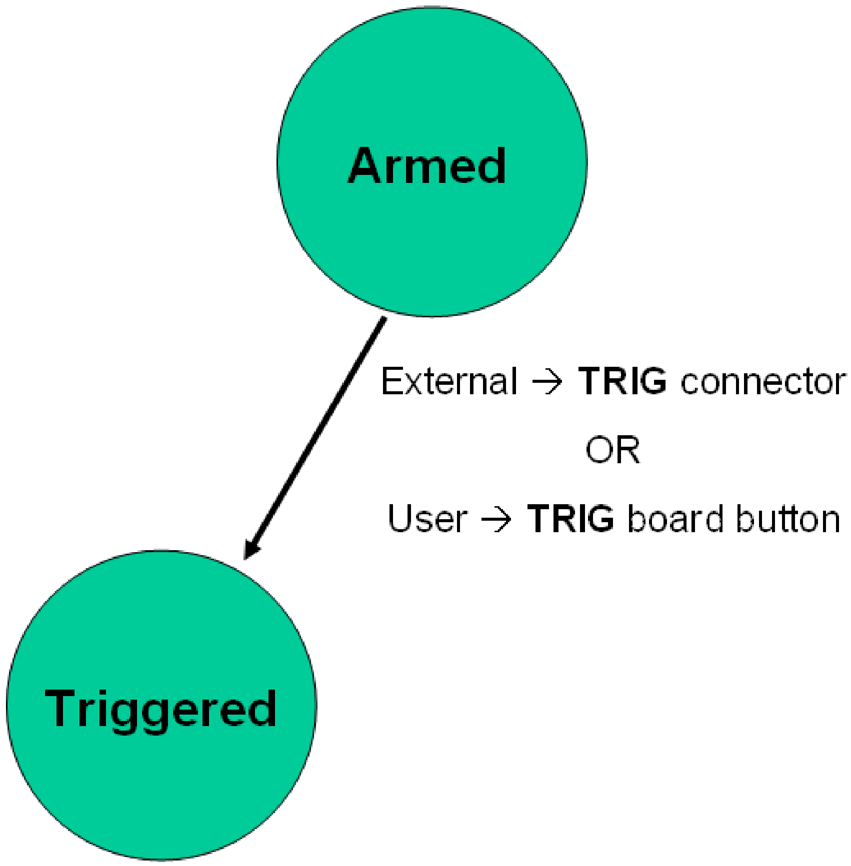
Disarmed

When the GUI application is first launched, it will put the board in the "Disarmed" state. At this point, the user can click on the **Arm** button in the GUI application to change the board to the "Armed" state. No other inputs will affect the board during the Disarmed state.



Armed

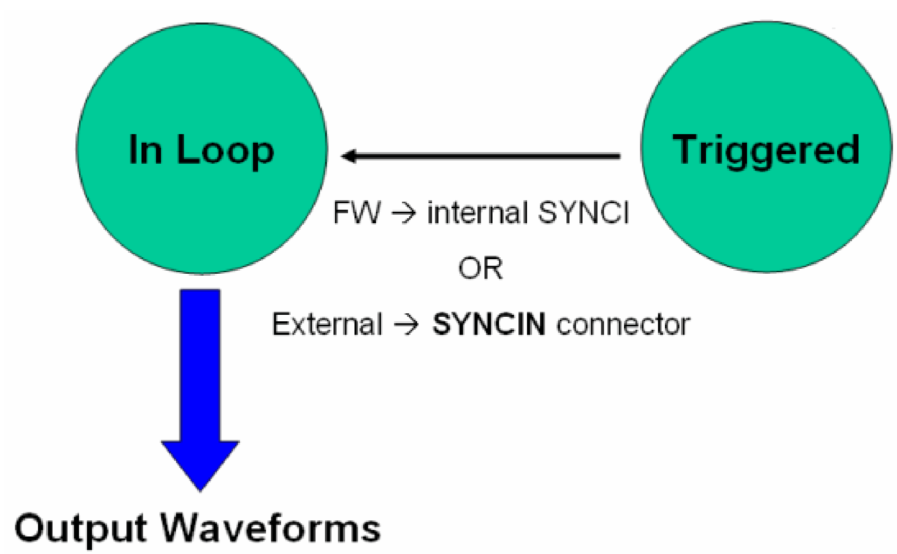
In the "Armed" state, the board is ready to be triggered. The user can either supply a trigger signal through the TRIG SMA connector or manually press the TRIG button on the physical board. Only falling signals (from logical high to low) will cause the board to change to state "Triggered."



Triggered

Once the board is in "Triggered" state, it will automatically go to the next "In Loop" state after a brief delay of 1-3 us (T_T). The falling edge Trigger signal can be provided either through software by pressing the trigger button in the GUI toolbar, or through hardware by either connecting a signal to the TRIG SMA connector on the module. The trigger signal source can be set according to the hardware setup page.

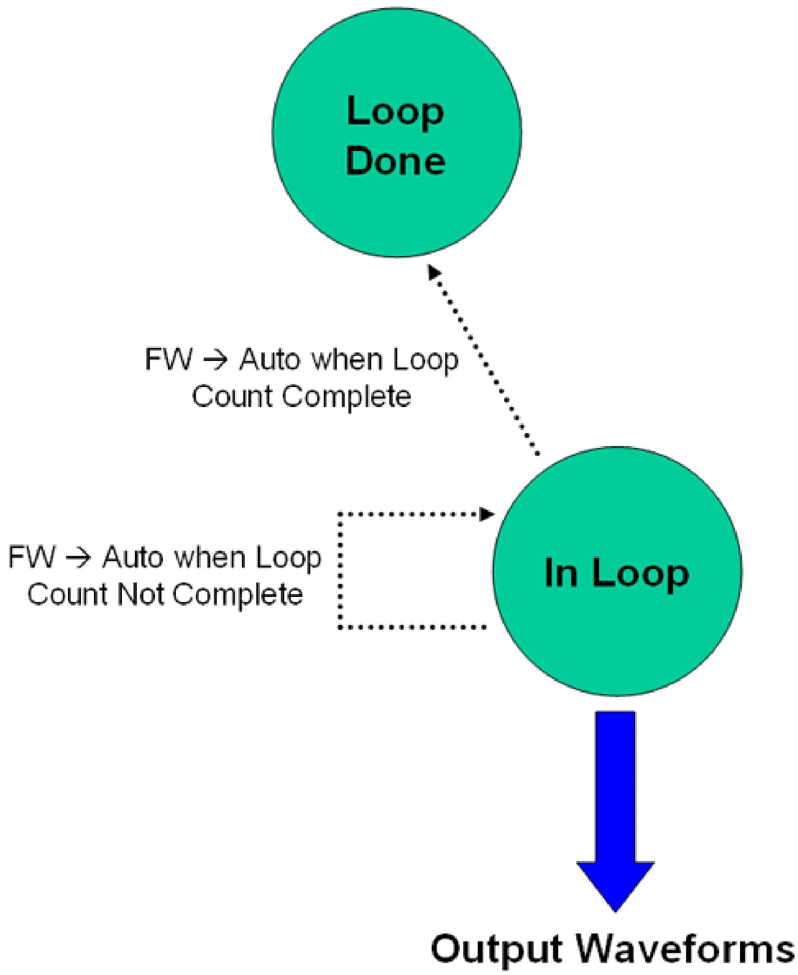
For a detailed look at the "Triggered" to "In Loop" transition, please go to the Synchronization page in the Multiple Board Operation section.



In Loop

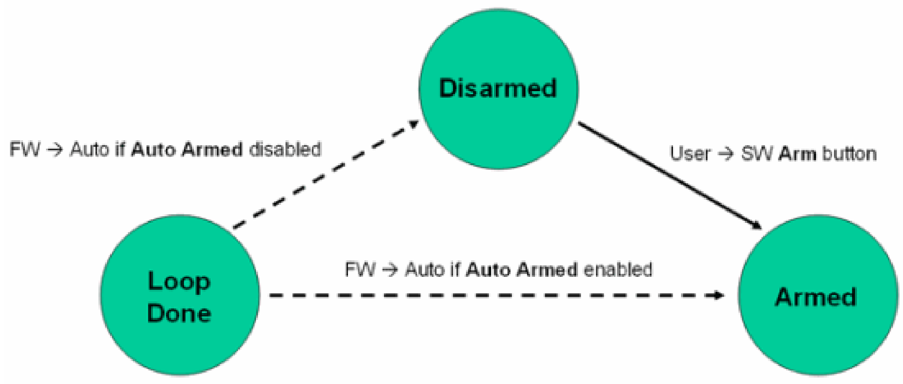
When the board is producing output waveforms, it is considered to be in the "In Loop" state. Each loop is equivalent to one waveform, which is basically running through one Memory Depth Cycle. Loop Count, in the GUI application, specifies the number of loops. If Loop Count is zero, then the board will be in an infinite loop and will output waveforms continuously until the user presses the **Abort** button in the GUI application.

NOTE: You may stop the entire sequence of events and return the board to "Disarmed" by using the **Abort** button at any time.



Loop Done

If Loop Count is finite, then after the last waveform is complete, the board will automatically enter the "Loop Done" state. As this point, the board either enters the "Disarmed" state or returns to the "Armed" state, depending on if **Auto Arm** is disabled in the AWG application. If **Auto Arm** is checked, then the board will go into the "Armed" state and will be ready for the next trigger signal. If it is unchecked, then the board will go into the "Disarmed" state and the user will have to manually arm the board again.


Disarmed
FW → Auto if Auto Armed disabled
User → SW Arm button
Loop Done
FW → Auto if Auto Armed enabled
Armed

Waveform Files -

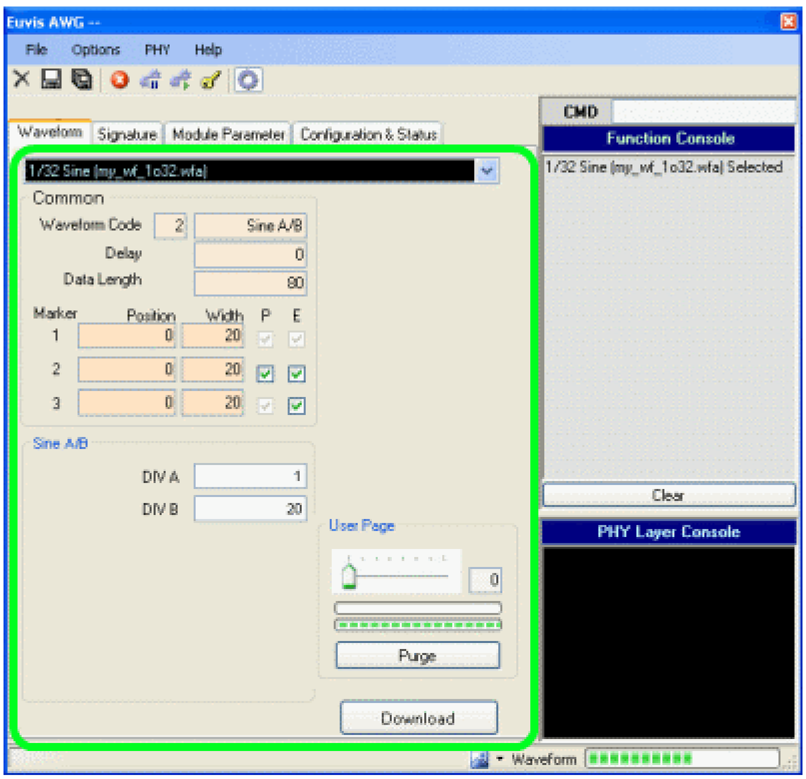
Load Waveform -

There are two ways you can load waveform files.
The first way, which is detailed below, is simply loading a saved .wfa file from a previous session with the GUI application.
The second way involves entering a waveform code corresponding to a waveform style for which you then enter specific parameters. For details on using the waveform codes, please go to the Waveforms page.

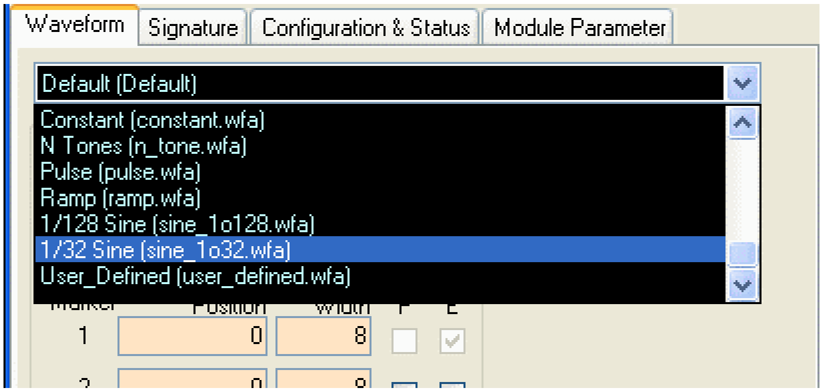
Waveforms previously saved

Waveform files saved from a previous session in the GUI application can be loaded easily. The files have the extension .wfa and are stored in your application folder or the user-defined waveform directory.

» With the program open, click on the **Waveform** tab if you are not already in the Waveform Panel.



» Click on the drop-down list and select one of the saved waveforms.



» Change any parameters if desired, and then please press the **Download** button at the bottom of the waveform panel or from the right-click pop-up menu. Wait until the waveform finishes downloading to the AWG module, and then restart the waveform (from the right-click pop-up menu).

For example, from the drop-down list select the waveform named "1/32 Sine (sine_1o32.wfa)". This is a Sine wave at 1/32nds of the clock frequency. You can adjust various parameters if desired. Then click the Download button, and after the file finishes downloading, press Restart.

Waveform codes

For details on using the waveform codes, please go to the Waveforms page.

Inverse Sinc Filter

The Inverse Sinc Filter is available from the Options menu bar. For details on applying the inverse sinc filter to analog waveforms, please see the description on the menus page.

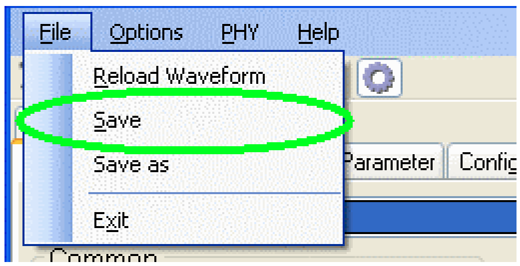
Save Waveform -

There are two ways to save a waveform: save into an existing waveform and save into a new waveform file. Waveform files are marked ".wfa" and by default are in the directory C:\Program Files\Euvis\AWG\. We strongly recommend that you change this directory by changing the "User Waveform Directory" in Preferences, so that your waveform files are preserved if you uninstall the AWG software.

Saving into Existing File

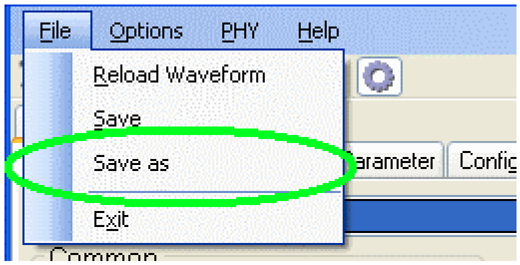
» After you have loaded an existing file, change the waveform parameters and the module parameters to the desired settings.

» When you are ready to save, click on either the "Save Waveform" button or go to the menu bar and click on File --> Save.

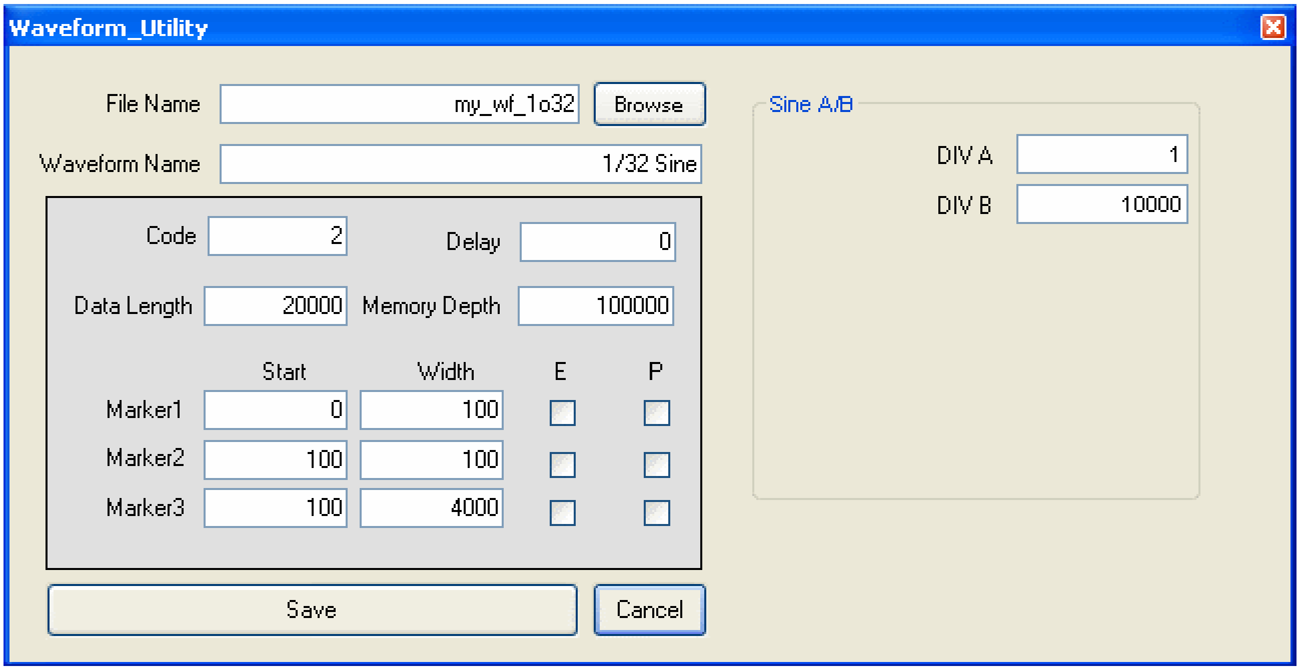


Saving into New File

» First, load any predefined waveform from the drop-down list in the Waveform panel, and simply change any parameters to the desired settings. Then either click on the "Save Waveform As" button or go the menu bar and click on File --> Save as.



» A new dialog box will appear, giving you the option to save the waveform.



File Name is the actual file name that the operating system uses to identify your file. Make sure that the file name ends with ".wfa" to distinguish it from other files.

Use the **Browse** button to choose the directory in which to save your files. The default directory is your program directory, but please note that if you uninstall the AWG application, any waveform files saved in your program directory may be erased, so please save your waveforms in a different directory.

Waveform Name is an internal program name that will show up in the drop-down list in the Waveform panel and will help you identify the waveform file.

The other options in the gray box are just the currently used parameters. If you wish to change them, you may alter the values in the boxes to the right of each parameter. Some parameters may not automatically reflect the currently used parameters and need to be entered again here before you save them.

» After you are satisfied, click on the **Save** button.

Note that if the File Name is not unique, the program will overwrite the existing file without warning you, so please be careful.

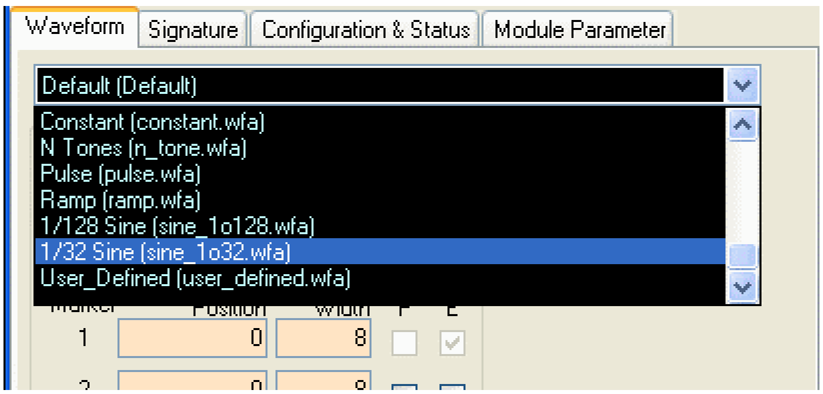
Waveform Styles -

This module has several pre-stored waveform styles that you can configure with specific parameters.

From the Waveform panel, there are two ways to recall a pre-stored waveform: you may either select one from the drop-down list or enter a waveform code.

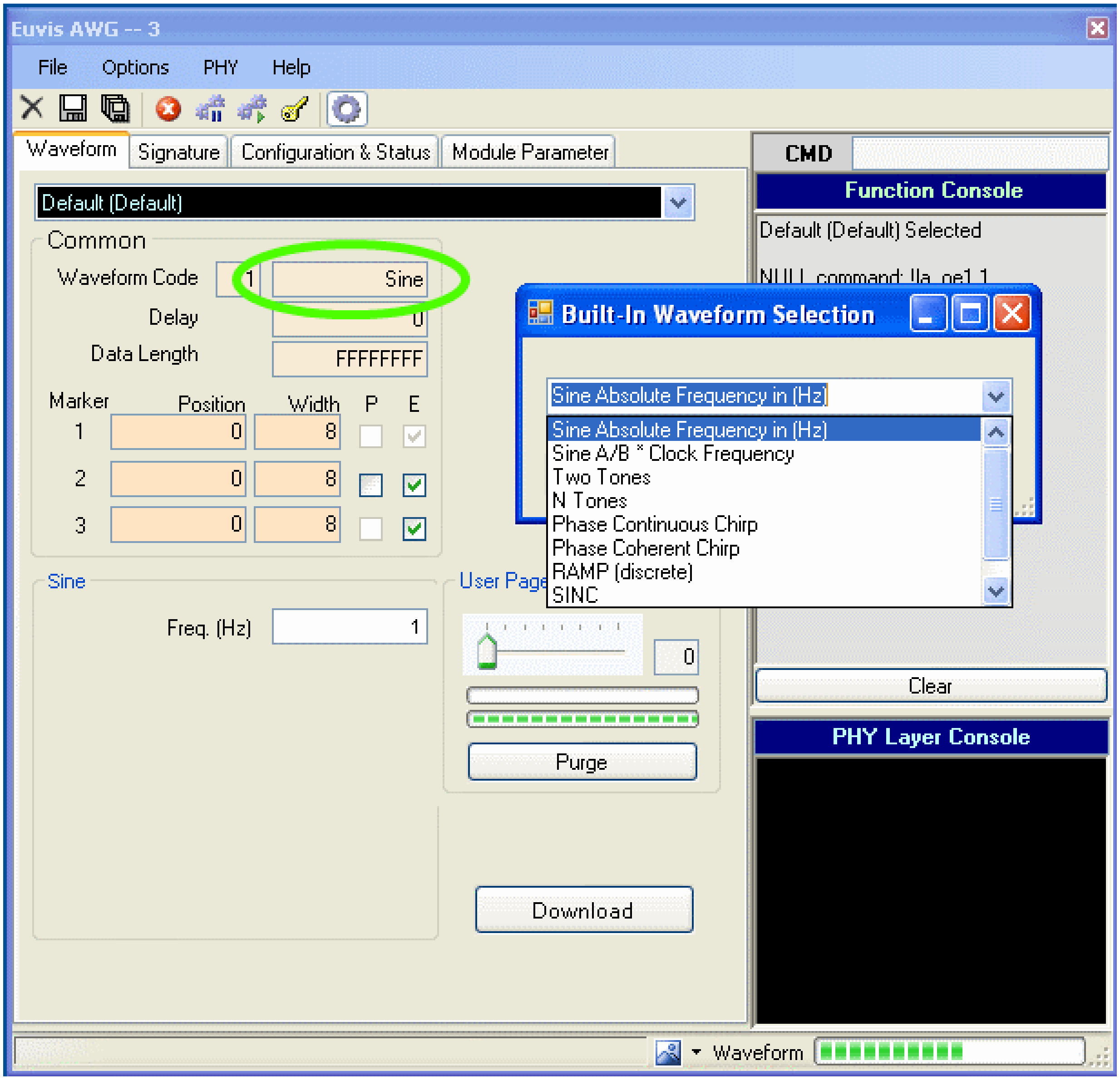
Drop-down List

The drop-down list is a convenient way to recall pre-stored waveforms.



To select a waveform, click on the Waveform tab of the GUI application. Near the upper left corner of this waveform panel is a drop-down list containing a list of pre-stored waveforms. Each pre-stored waveform has its own adjustable parameters. You can edit those parameters in the waveform panel, and you can also save the parameters for use in future sessions. Please refer to the Waveform Codes section below for information on the different waveform styles and their corresponding parameters.

Another way to access the list of pre-stored waveform styles is to double click the box to the right of the waveform code text box. A "Built-in Waveform Selection" window should appear. Click on the drop-down list to select a waveform style.



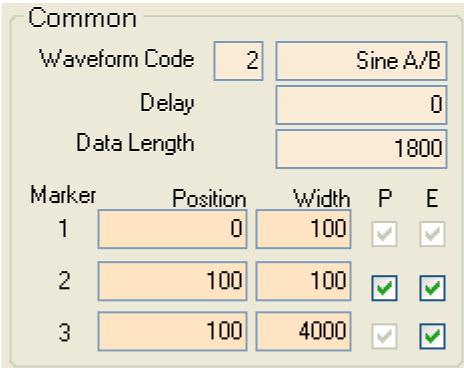
To transfer waveform data from your computer to the module, press the **Download** button (at the lower right corner of the waveform panel). The computer then downloads the parameterized data to the module memory so that the firmware generates the actual waveforms. You must press the Download button each time you finish changing the parameters.

Waveform code

Each waveform code is associated with a waveform style. Entering the waveform code is a fast way to recall the waveform style you want.

Entering Waveform Codes

The waveform code text box is located at the top of the Common parameters section in the Waveform panel. Before entering in a code, you should first choose any waveform from the drop-down list. This step initializes some parameters for the GUI. You do not need to download that waveform.



Enter a number from the table below to choose a waveform style. Alternatively, you can double-click in the waveform style box to the right of the waveform code box (where it says "Sine A/B" in the above image) and choose the waveform style from a pop-up list.

Waveform Code	**Waveform Style**	**Waveform Specific Parameters**
1	Sine	Frequency (Hz)
2	Sine A/B	A=numerator, B=denominator, (unsigned integers) A/B is a fraction of the clock frequency
3	Two Tones	A=numerator, B=denominator, (unsigned integers) A/B is a fraction of the clock frequency for tone1 C=numerator, D=denominator, (unsigned integers) C/D is a fraction of the clock frequency for tone2
4	N Tones	Frequency, Amplitude (dBc) relative to full scale, Phase (degrees)
21	Chirp (Phase Coherent)	Fstart=Start Frequency, Fstop=Stop Frequency T1=Delay until Fstart, T2=Chirp time, T3=Delay after Fstop
22	Chirp (Phase Continuous)	Fstart=Start Frequency, Fstop=Stop Frequency T1=Delay until Fstart, T2=Chirp time, T3=Delay after Fstop
31	Ramp	A=numerator, B=denominator, (unsigned integers)
50	Pulse	Tini=Start time, Tr=Rise time, T=Pulse Width, Tf=Fall time, Amp=Amplitude [hex]
101	Constant	Value [hex] (unsigned integers) absolute value
256	User Defined	.uda user defined waveform file

Waveform Code

is the index that the operating system uses to identify the waveform style.

Waveform Style

will headline the waveform-specific parameters and will help you identify the waveform. It appears to the right of the waveform code. If you double-click in this box, you can select the waveform style from a pop-up list instead of entering a waveform code.

Waveform Specific Parameters

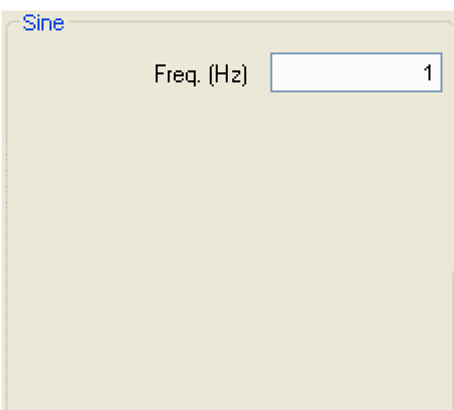
After you have entered a valid waveform code, style-specific parameters will appear below the common parameters section. Here you may change the specific parameters to your desired settings. The next section will explain the different parameters in detail.

Waveform Specific Parameters

The AWG module has various waveform styles available, each with its own waveform parameters. This section describes the pre-stored waveforms and their associated parameters.

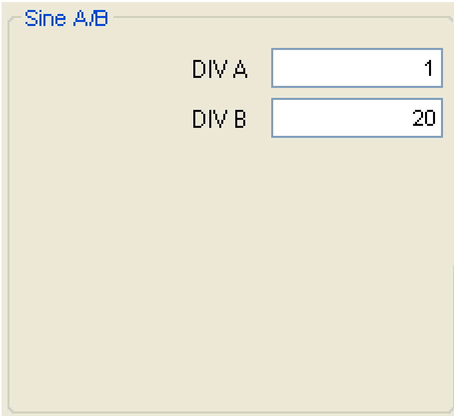
Sine

The **Freq** frequency is the only specific parameter for the sine waveform style. Enter the desired numeric frequency in Hertz (Hz). For example, to output a 1 MHz sine wave, enter 1000000. Note that this is an absolute frequency based on the input clock you specify in the Signature panel. If you set this frequency first and then change the input clock, then the output sine wave will not be at the same frequency as the one you entered.



Sine A/B

This sine wave frequency is the input clock frequency scaled by the ratio A/B. The numerator, A, and the denominator, B, must be nonzero unsigned hexadecimal integers.



Two Tones and N Tones

The multi-tone waveform style can be configured to have a different amplitude and phase for each frequency tone. Two closely spaced tones be can useful for testing the ability of an active circuit to reject 3rd harmonic interference.

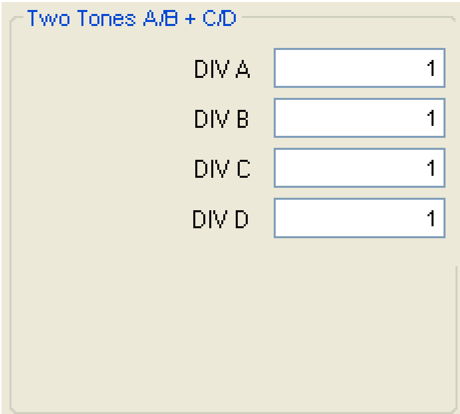
Each row in the table corresponds to one tone. For each desired tone, enter in one row:

- the frequency in Hertz (Hz)
- the power amplitude in dB, relative to full scale
- the phase in degrees

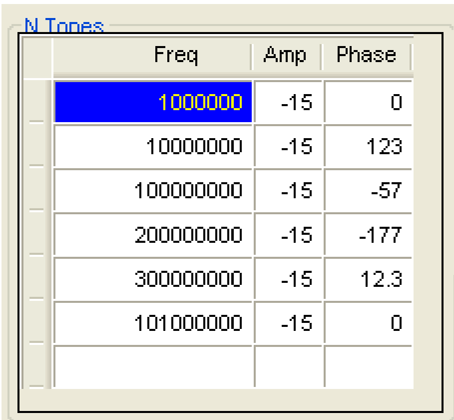
Please be aware that when the amplitudes of each waveform (each tone) add constructively, the instantaneous amplitude of the envelope waveform (the n tones) may exceed the full scale amplitude. When this occurs, the overflow data will be truncated.

To illustrate, consider the case of two tones, each with power amplitude set to -3 dB. Their peak voltages would be 1/sqrt(2),or about 0.7 of the full scale. When added constructively, the peak voltage is 1.4 times the full scale. When the instantaneous voltage exceeds the full scale, the data wraps around back to 0. Other ways to think of it are that the overflow data is ignored or that the data is *modulo* full scale. The remedy in this two-tone case is to set the power amplitude of each tone lower than -6 dB, relative to full scale.

The example My 2 Tones has two tones at 567 and 568 MHz with equal amplitude and different phase.



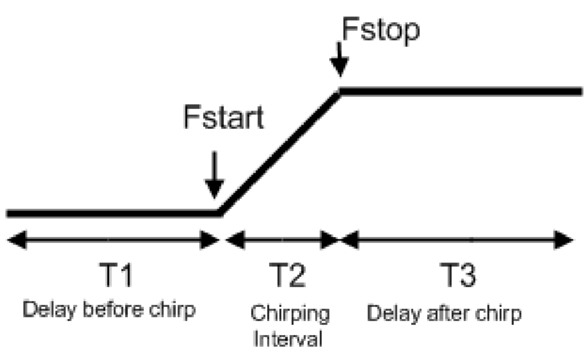
The example My N Tones has 6 tones between 1 MHz and 100 MHz, with various amplitudes and phases, and uses raised cosine (Hanning) windowing.



Freq	Amp	Phase
1000000	-15	0
10000000	-15	123
100000000	-15	-57
200000000	-15	-177
300000000	-15	12.3
101000000	-15	0

Chirp

The chirp waveform can be easily produced using user defined parameters. The module has phase coherent chirp and phase continuous chirp as pre-stored waveforms. There are five parameters that define this waveform style: Fstart, Fstop, T1, T2, and T3.

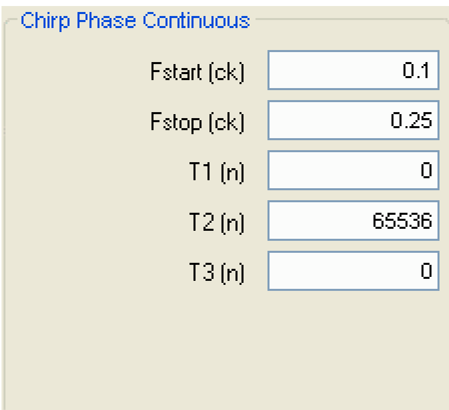


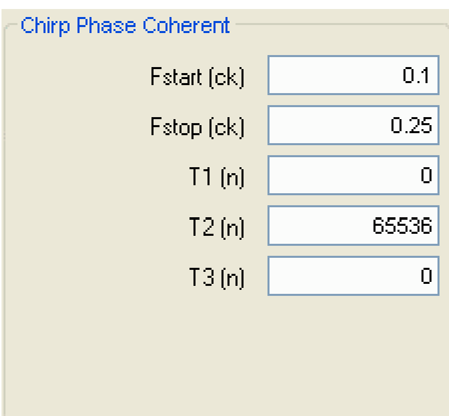
You must define the start and stop frequencies of the chirp. Respectively, they are the Fstart and Fstop parameters. These two values will determine the frequency boundaries of your chirp. To chose a start frequency you must define your desired frequency as a ratio of your input clock. The stop frequency is also defined as a ratio of the input clock.

In addition to defining the start and stop frequencies of this waveform, you can define 3 time interval parameters. "T1" set the delay time before the chirp, "T2" sets the duration of the chirp, and "T3" sets the delay time after the chirp.

These 5 parameters are all defined relative to the input clock "ck".The "T" parameters are the actual number of discrete data values for each interval, corresponding to one clock cycle per data value. For instance, if your input clock frequency is 1 GHz, then each data value would correspond to 1 ns.

As an example, we will use the default values for the chirp waveform styles below:

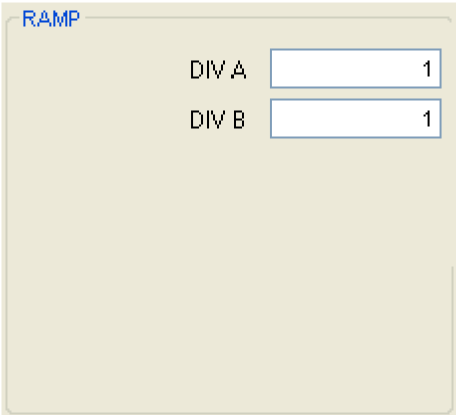




If these chirp parameters were used with a 4 GHz clock input, then the start frequency woud be 400 MHz (0.1*4000MHz), and the stop frequency would be 1GHz (0.25*4000MHz). As for their time interval parameters, you can see that the default values have no delay before and after the chirp. However, the chirp duration is set as 65536 data points. This will correspond to about 16.4 microseconds ((1/4e9)*65536).

Ramp

The module produces a ramp by multiplying the data point index by the A/B slope factor and the amplitude resolution of the DAC, which is 4096 for the AWG252 and AWG452 and which is 2048 for the AWG801. If you want a ramp with a high slope then you should set the ratio A/B greater than 1. If you want a ramp with a low slope then you should set the ratio A/B less than 1.

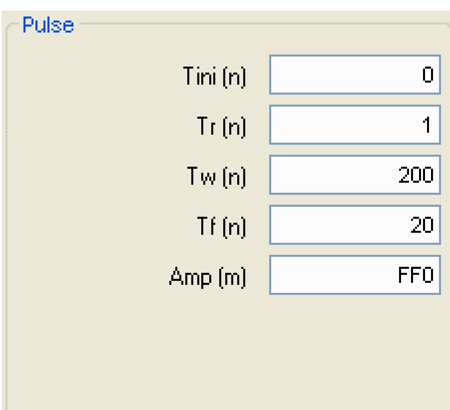


Pulse

The pulse is a time-domain trapezoid defined by 5 parameters: the start time (Tini), rise time (Tr), pulse width (Tw), fall time (Tf), and amplitude (Amp). These parameters are relative to the input clock and are represented in hexadecimal unsigned integers.

The T parameters are defined as discrete data points "n" relative to the AWG. Each clock cycle corresponds to one data point. For instance, if your clock is at 1 GHz and you want a pulse width that spans 20 nanoseconds, then you should enter 14 in the "Tw" field.

The "Amp" parameter is defined as the DAC amplitude. The AWG252 and AWG452 have 12-bit amplitude resolution, corresponding to a full scale hexadecimal value of FFF, but you should not exceed FD0; for the AWG801, full-scale is 7FF, and you should not exceed 7E0. Remember to define these parameters in hexadecimal.

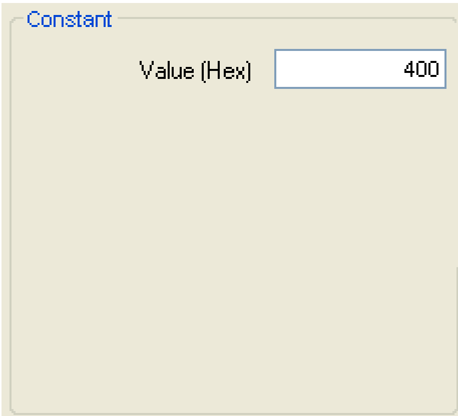


Constant

The AWG produces a constant value waveform. The Value parameter should be a hexadecimal integer in the range from 0 to about 98% of full-scale.

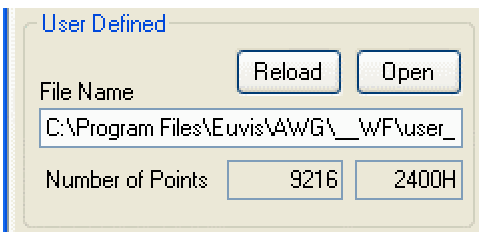
For the AWG252 and AWG452, full-scale is FFF, and you should not exceed FD0; for the AWG801, full-scale is 7FF, and you should not exceed 7E0.

For example, if you were to enter an amplitude of 400, then the output should be about 1/4th of the full scale amplitude for the AWG252 or AWG452, or 1/2 of the full scale amplitude for the AWG801.



User Defined

The user defined waveform style is a custom waveform loaded from a (.uda) user generated waveform file. Please ensure that the Data Length is at least as long as your waveform data.

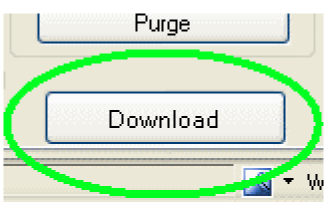


If your .uda file is located in a different directory, you should specify the location by clicking the Open button in the User Defined parameters section.

Download the waveform

Check that the Data Length is long enough for your waveform. Please see the Data Length page for details.

Finally, click the "Download" button to download the waveform to the board,



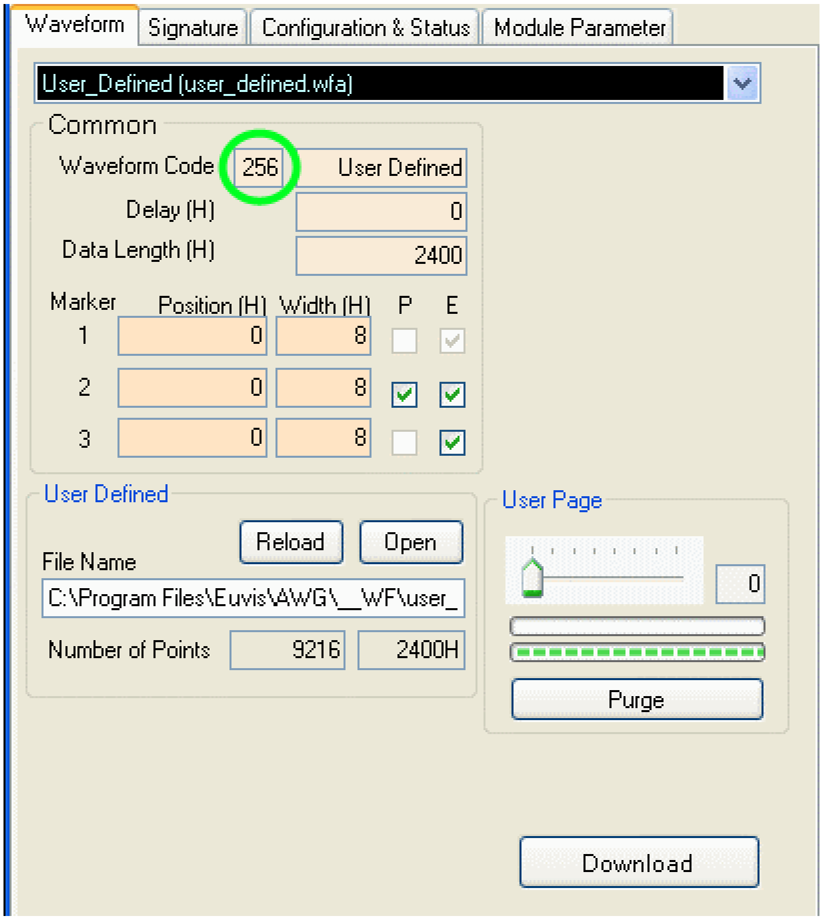
and press "Restart" in the toolbar or in the right-click menu to output the new waveform.

Saving your waveform

You may save the waveform with all of the current parameters by clicking the Save As button. A new dialog box will appear, where you can edit the names and parameters before you save the waveform. For more details on saving process go to the Save Waveform page.

User Defined Waveforms -

You can create your own waveforms by loading a custom user defined waveform into the GUI application. To load the file, go to the Waveforms panel and choose the User_Defined (user_defined.wfa) waveform in the drop-down list. Or, select any predefined waveform, and then enter "256" into the Waveform Code box.



Then, in the User Defined parameters section, click on Open and load the .uda file that you have created. The Data Length will automatically adjust.

User Defined Waveform Files

To create a user defined waveform file you will just need to fill a text file with the waveform information and save the file with a ".uda" extension.

Click on the file below to see what a user defined waveform file looks like:

User Defined Example

For the "#type" there are only two options. If you want to use the GUI to control the markers then you would use "#type=1". By using this type, you can only control the markers with the three marker boxes in the

Waveform tab. The other option is to use "#type=5" like the example. In this case, you can specify the markers using the second column of the .uda file.

The "#hex=1" signifies that all data are entered as hexadecimal. This is the only option that you should use.

The first column of the .uda user defined file determines the amplitude of the output.
For the AWG252 and AWG452, the MUXDAC has 12 bit amplitude resolution, so you should specify the amplitude within the range from 0 to FFF (hexadecimal).
For the AWG801, the MUXDAC uses 11 bit amplitude resolution, so you should specify the ammplitude within the range from 0 to 7FF (hexadecimal).

The second column is optional and only relevant when you use "#type=5". This column specifies the three marker outputs. You can think of the three markers as three bits. The LSB is for Marker 1, the second LSB is for Marker 2 and the third LSB is for Marker 3. To set any of the markers active you would just set that bit to "1". Therefore when you want to set Marker 1 active, you would set the second column to "1". When you want Marker 3 active you would set the second column to "4". If you wanted all three markers to be active then you would set the second column to "7". The table below shows the different options for the markers:

Setting	**Marker 3**	**Marker 2**	**Marker 1**
0	Off	Off	Off
1	Off	Off	Active
2	Off	Active	Off
3	Off	Active	Active
4	Active	Off	Off
5	Active	Off	Active
6	Active	Active	Off
7	Active	Active	Active

Unlike with the marker settings in the GUI, you can have any of the markers go active multiple times during each waveform. In addition, the marker is active for every four data points, and you must have the marker active on the first of the four data points. If the marker is set to active on the second, third or fourth data points, but not the first, then the marker will not be active. Since Marker 2 has a polarity select, you must still use the "P" checkbox in the Waveform panel to select reverse polarity for that marker. For details about the three available markers, please see the Parameters page of the Operation Details section.

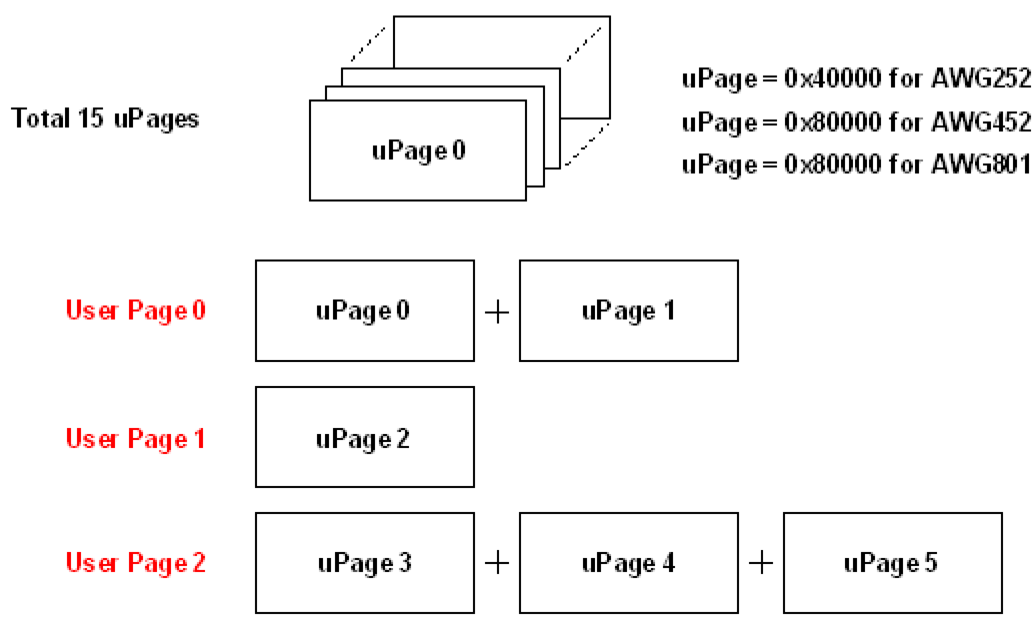
Multiple Waveforms with Dynamic User Paging -

This module features memory address paging. The GUI application can demonstrate what can be done with this user page option. In it, you can choose which one of up to 15 possible user pages to use, and you can quickly switch between various waveforms stored on those user pages.

A waveform is stored in a User Page. Each user page consists of several memory partitions called uPages. The API automatically manages the uPages according to the data length of waveforms. Therefore, the uPages are virtually invisible to you. To download a waveform to the module, you need to select a user page, optionally modifying the waveform parameters if built-in waveforms are used, and download the waveform data. After downloading the waveform data from the GUI, the user page information is automatically updated under the waveform tab. In the API, the user page information, such as how many user pages are used and how many upages remain, can be derived via API properties.

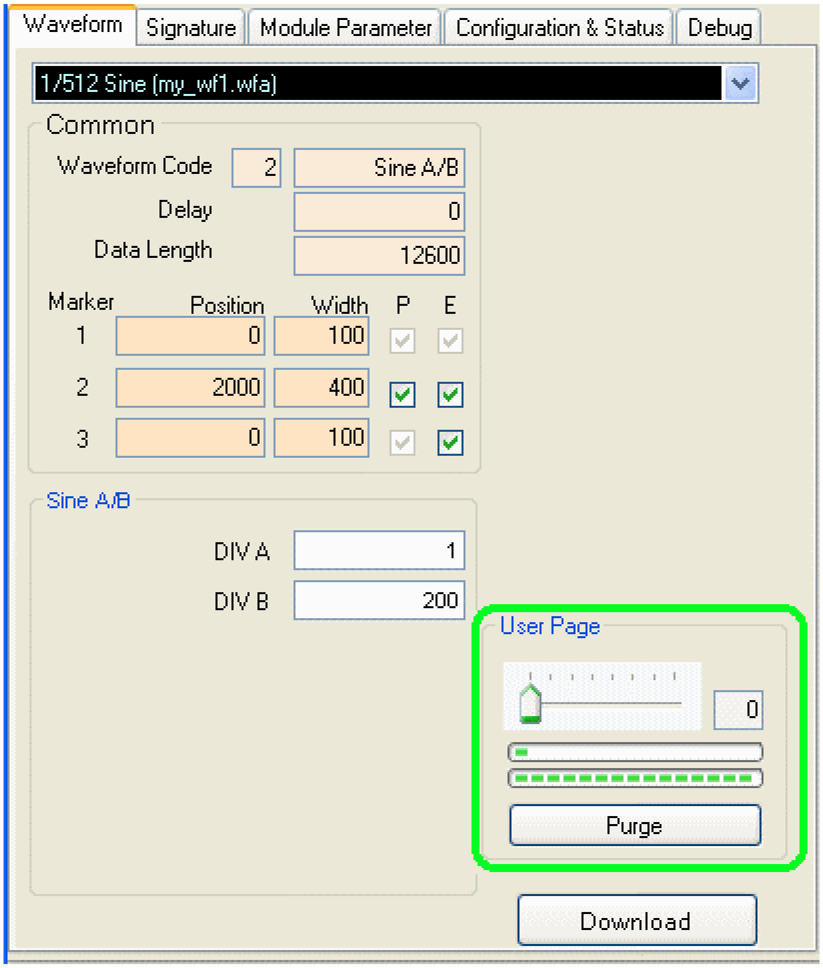
The total memory in the module is divided into several equal-length page-wise segments. These segments are called uPages. This partitioning enhances the efficiency of memory usage and makes it possible to swap waveforms on the fly without any interruption or latency. Typically, the modules have 15 uPages with lengths of 0x40000 for the AWG 252 and 0x80000 for the AWG 452 and AWG 801.

If a waveform is shorter than the length of the memory partition, the remainder of that uPage memory partition is simply unused. If a waveform requires more than one such memory partition, then the total number of available user pages will be reduced.



To choose the user page

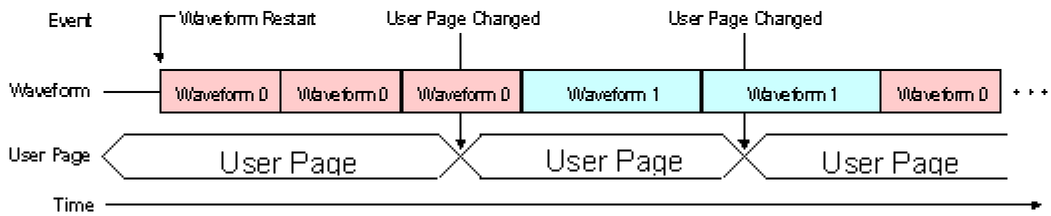
Go to the Waveform panel. In the lower right area of that panel, you'll see the User Page section, which has a slider where you can choose the different user pages.



Each user page can load a separate waveform, but the more user pages you have, the fewer memory addresses you have available for each waveform -- meaning the maximum data length of each waveform is shorter.

Dynamic Paging

Once you have downloaded waveforms onto the user pages, the waveforms can be selected and generated dynamically without restarting the module. The newly selected waveform will follow the previous one without latency. The new waveform starts right after the end of the preceding one. The user page selection can happen any time. As long as the user page is selected (altered) before the current waveform ends, the newly selected waveform will be generated right after the end of the current waveform. Otherwise, the subsequent waveform remains the same as specified in the current user page.

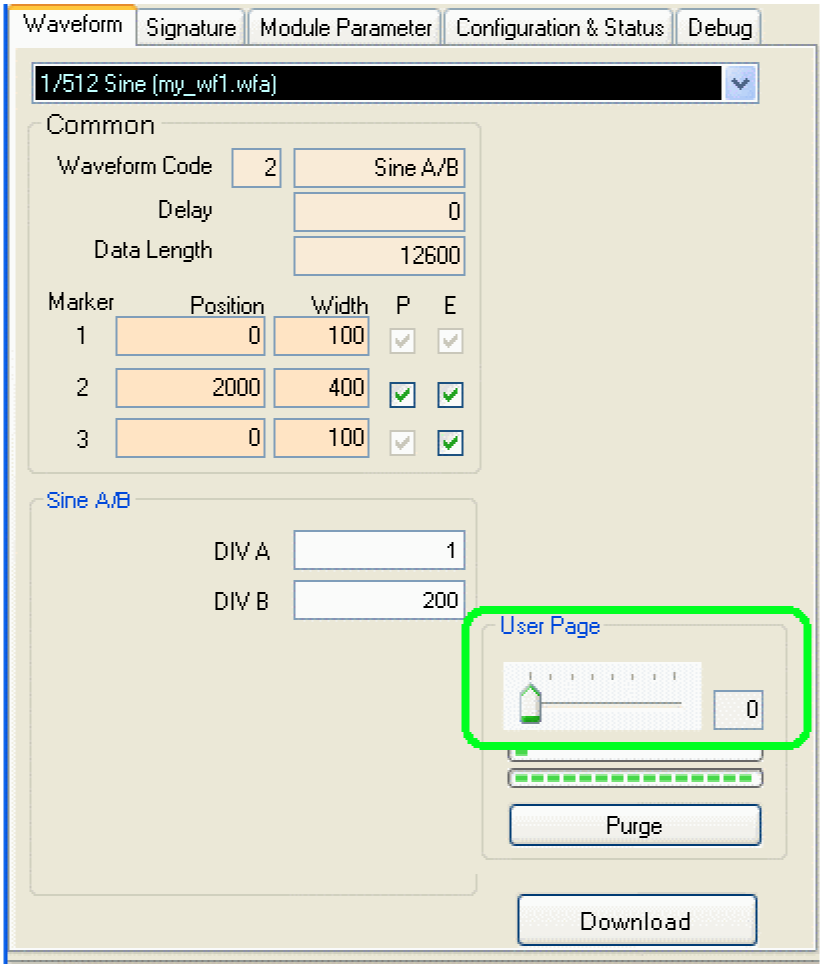


The above figure shows an example of how the waveforms change dynamically according to the user page selections, which can be made via the GUI or the API. Two different waveform, waveform 0 and waveform 1, are stored in the user page 0 and 1, respectively, using download operations. The AWG waveform generation mode in the example is free run continuous mode. The user page 0 is selected at the beginning. Once the AWG restarts, waveform 0 is generated repeatedly. In the third waveform generation, the user page is changed to user page 1 by the user. Waveform 0 will continue to its end, and the following waveform generated is waveform 1 according to the new user page selection. In the fifth waveform, the user page is changed again back to user page 0. The sixth waveform will be waveform 0 accordingly. Dynamic paging gives the ability to generate compound waveforms as combinations of basic waveforms.

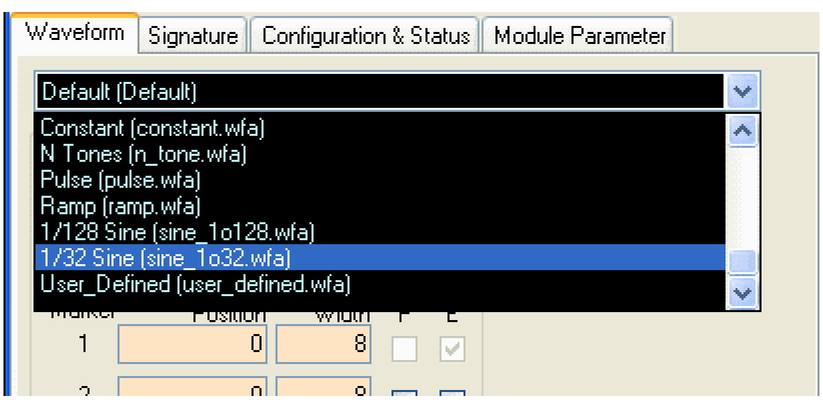
Example

For this example, we will limit each waveform data length to one uPage length to take advantage of all fifteen user pages so that we can have 15 different waveforms.

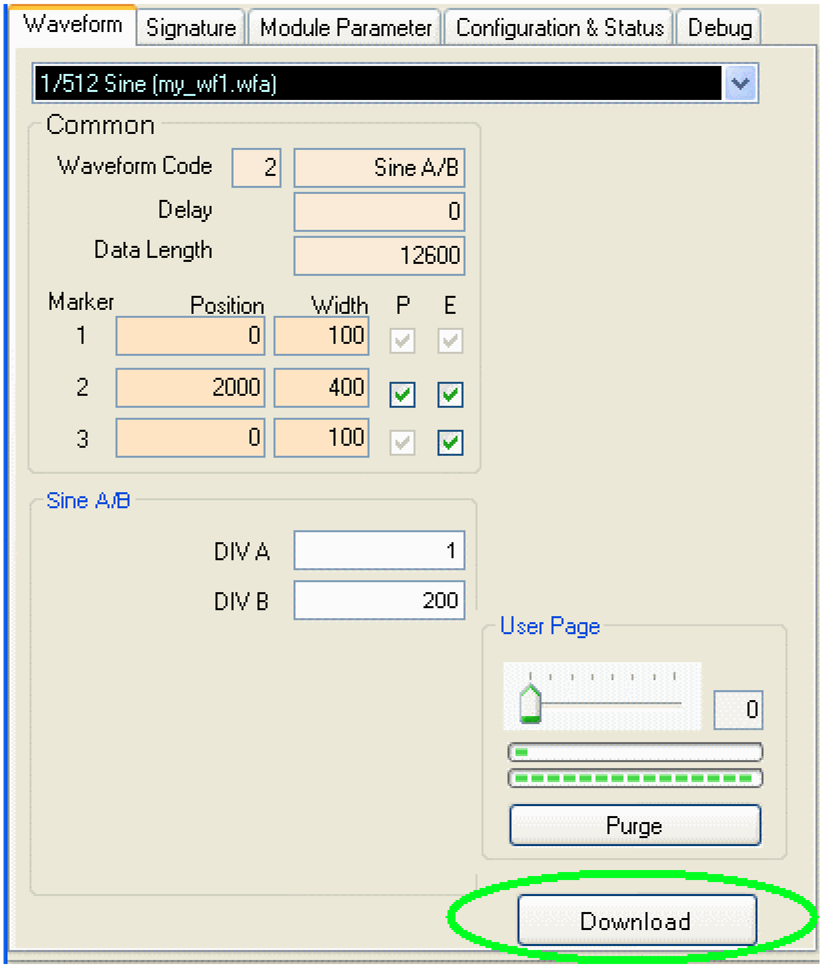
» Slide the slider to go to User Page 0.



» Either load a pre-stored waveform or create a waveform by adjusting the waveform parameters of an existing waveform. For this example we will load a previously saved waveform.



» Click on the **Download Waveform** button. After the waveform file finishes downloading to the module, press the **Restart** button (right-click the mouse, and then choose Restart) and you should see the waveform in your oscilloscope or spectrum analyzer if you them connected to the module outputs.



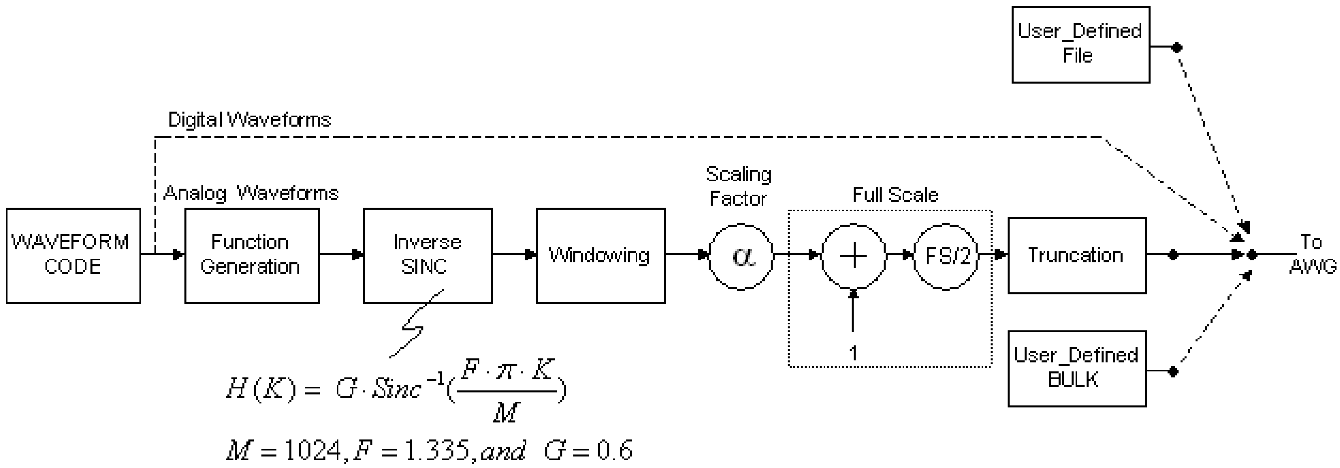
» Now slide the slider bar to the next tick. The corresponding page number increments by 1. Repeat the above two steps, downloading different waveforms for each page. After you have done this, you will have up to 15 different waveforms on the module. You can easily switch between waveforms by using the slider. Once you move the slider, the output waveforms will change immediately.

The GUI is just a demonstration of the paging option. To take full advantage of its potential, you would have to write your own application with our API so that you can switch around the different waveforms for a unique waveform. The minimum switching time is under 10 microseconds.

Scale and Filter -

Due to windowing effects from using time-limited waveform data, the frequency spectrum of analog waveforms may appear to have been convolved with a Sinc, and this filter is designed to mitigate the effects of that modulation.

This feature allows you to modify the inverse filter frequency factor, the subsequent window applied, and the full-scale amplitude applied to the analog waveform. Digital waveforms, such as ramp, pulse, and user-defined waveforms, are not affected by these settings. Analog waveforms, which are calculated internally with double precision, go through this inverse sinc filter, windowing, a linear scaling factor, and then are translated to full scale amplitude.



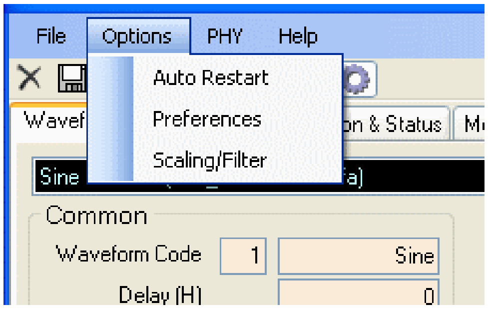
We recommend that you leave the Scaling Factor unchanged at 0.9875, and only use this option to turn on or off the filter. If desired, you can change the type of windowing applied to your analog waveform. Please remember to re-download the waveform data if you change any of these settings.

Example using Scale and Filter option

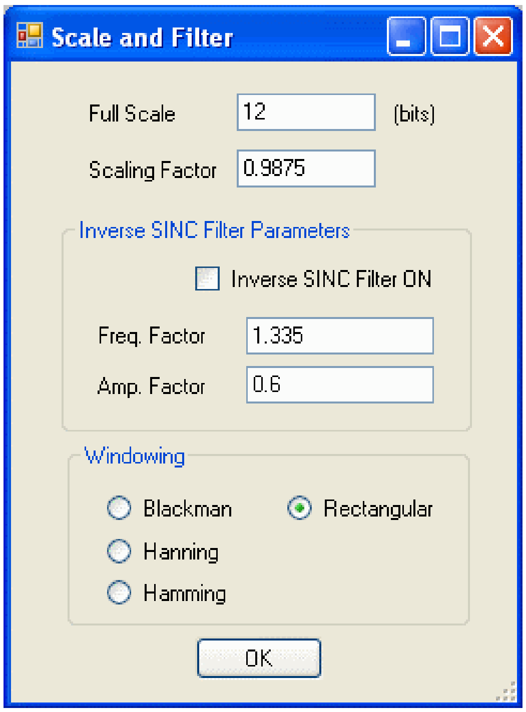
To illustrate the Scale and Filter feature, we will apply the filter to a waveform that is not exactly periodic -- Sine at 123,456,789 Hz. The reason we consider this waveform non-periodic is because this frequency is not a regular fraction of our clock frequency, so the actual waveform amplitude data does not repeat. Because of this, the time-domain waveform is not quite periodic, and this results in a frequency spectrum that includes other frequencies. Therefore, it is useful to apply a time-domain windowing filter to such waveforms.

Load the pre-stored Sine 123 MHz (sine_123456789.wfa) waveform file. Or, simply load a Sine waveform at 123,456,789 Hz.

Go to the menu bar and click on Options --> Scaling/Filter.



The Scale and Filter box will pop up.



Click on the box for Inverse SINC Filter ON and the Windowing button for "Rectangular", and then click OK. Download the waveform, and then press Restart (right-click pop-up menu). You will see a Sine wave near 123 MHz.

Now change the windowing to a different type, such as Hamming, and repeat the download and restart procedure. You will see the frequency spectrum of the sine has been filtered to reduce the spurs.

Also try this with the Inverse SINC Filter turned off. For a narrowband waveform like sine, you will not see any difference, except that the amplitude will no longer scaled by the Amp. Factor set in that section. For a wider-band waveform, the inverse sinc filter will help balance the amplitude across the frequencies.

Waveform Generation Modes -

There are three modes in which you can have the module generate waveforms -- Free Run/Continuous, Triggered Free Run, and Triggered Burst modes.

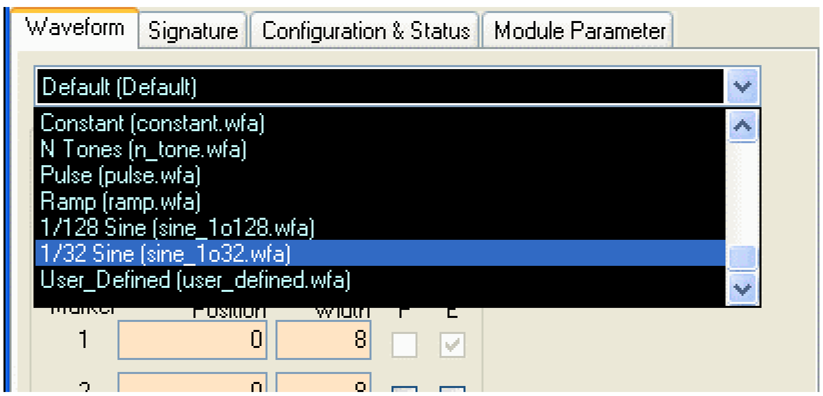
- Free Run/Continuous
- Triggered Free Run
- Triggered Burst

Free Run / Continuous Mode -

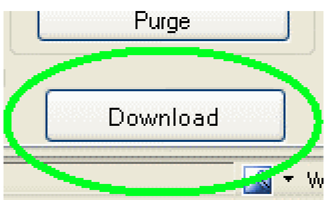
In Free Run/Continuous mode, the module starts waveform generation by a *Restart* command from the GUI or API-based applications. Once the waveform starts, the module repeats the waveform continuously. There is no latency between two consecutive waveforms. The following waveform starts right after the end of the preceding waveform. The waveform generation can be aborted by an *Abort* command from the GUI or API-based applications.

» With the program open, click on the **Waveform** tab if you are not already in the Waveform Panel.

» Click on the drop-down list and select one of the saved waveforms, such as "1/128 Sine (my_wfa2.wfa)", which is a Sine wave at 1/128ths of the clock frequency. The files have the extension .wfa and are stored in your application folder or the user-defined waveform directory. You can adjust various parameters if desired.



Click on **Download Waveform** near the bottom of the window or click on File --> Load Waveform. Wait several seconds for the file to finish downloading.



Then press Restart (from the right-click pop-up menu or from the 6th button in the toolbar).

If you are using the dynamic user paging feature and have downloaded more than one waveform, you can simply switch to another page and Restart to see the other waveform(s) without re-downloading the data.

Triggered Free Run Mode -

In Triggered Free Run mode, the operation manner is similar to that in Free Run/Continuous mode except for the start of waveform generation.

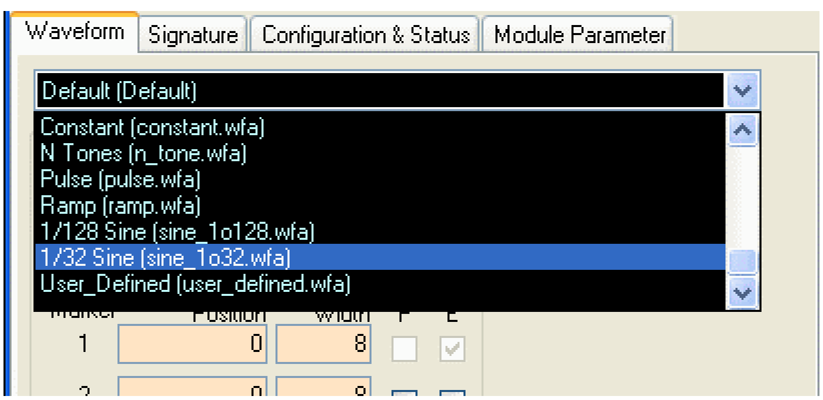
In this mode, waveform generation is initiated by a trigger signal. In order to accept the upcoming trigger signals, the module must be *armed* prior the acceptance of the trigger signals. Trigger signals received before the module is armed will be ignored. An *Arm* command from the GUI or API-based applications can be used to arm the module. Once the module is *armed*, it waits for the trigger signal.

The waveform generation starts after the falling edge of the trigger signal. The trigger signal can be applied via the TRIGGER SMA connector or provided by a command Trigger via the GUI or API-based applications. Due to the asynchronous timing between the upcoming trigger signal and the module clocking, there will be some uncertain delay/latency between the trigger and the waveform generation. However, the waveform generation is synchronized with respect to the module clock.

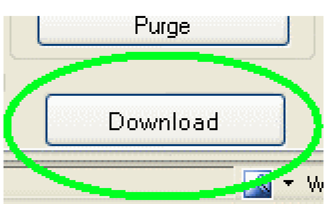
After waveform generation has begun, the module proceeds as if in Free Run/Continuous mode. It will repeat the waveform continuously. There is no latency between two consecutive waveforms. The following waveform starts right after the end of the preceding waveform. The waveform generation can be aborted by an *Abort* command from the GUI or API-based applications.

» With the program open, click on the **Waveform** tab if you are not already in the Waveform Panel.

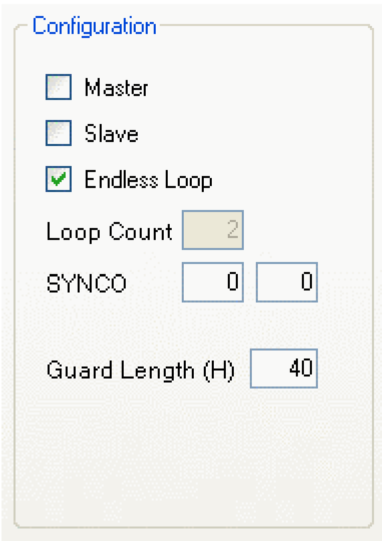
» Click on the drop-down list and select one of the saved waveforms, such as "1/128 Sine (my_wfa2.wfa)", which is a Sine wave at 1/128ths of the clock frequency. The files have the extension .wfa and are stored in your application folder or the user-defined waveform directory. You can adjust various parameters if desired.



Click on **Download Waveform** near the bottom of the window or click on File --> Load Waveform. Wait several seconds for the file to finish downloading.



» Click on the **Configuration and Status** tab. In the Configuration section, check the box for Master, and ensure that Endless Loop is also checked.



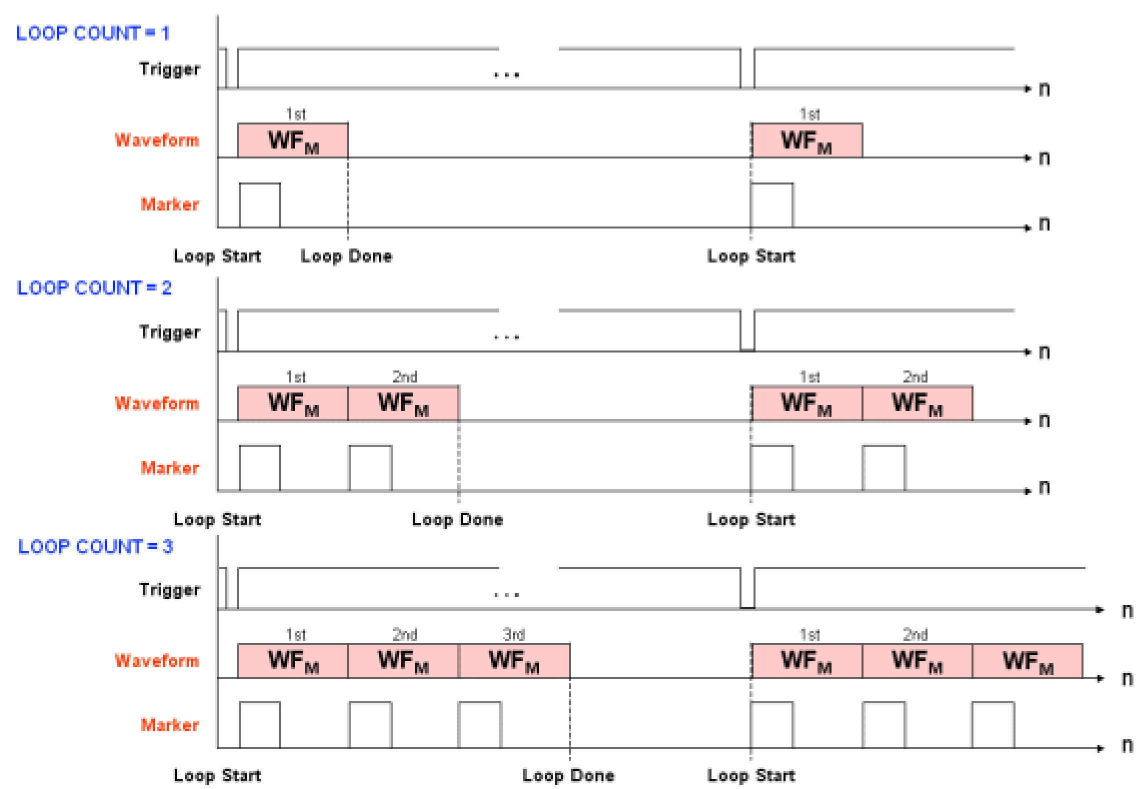
Then press the *Arm&Ready* button in the toolbar, which is the 5th button from the left in the area between the menu bar and the panel tabs.

Finally, apply a *trigger* signal, either via the TRIG SMA on the module or by pressing the Trigger button, which is the 7th button in the toolbar.

Triggered Burst Mode -

In Triggered Burst mode, the module starts waveform generation when it is armed and receives the trigger signal, as in Triggered Free Run mode. However, instead of repeating continuously, the waveform starts, repeats, and stops after a finite number of repetitions. The number of the repetitions can be specified by the property *Loop Count* via the GUI or API-based applications. The *Loop Count* property can be set from 1 to 255. Similar to the Triggered Free Run Mode, trigger signals received before the waveform stops or before the module is armed will be ignored. Once the waveform stops, the module will arm itself automatically and await the next trigger signal.

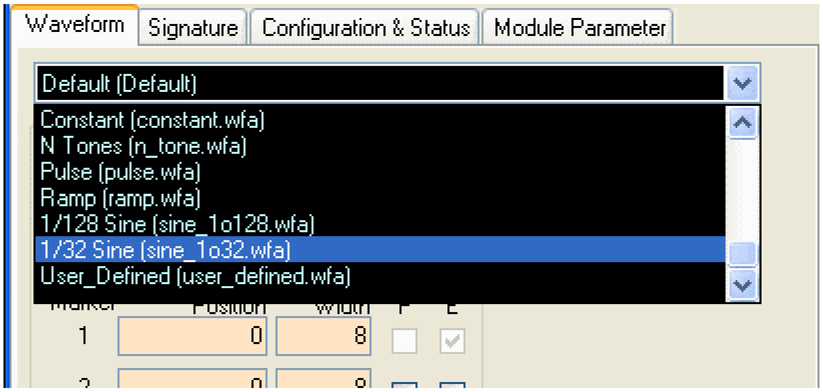
The following figure shows waveform generation for different *Loop Counts*: 1, 2, and 3.



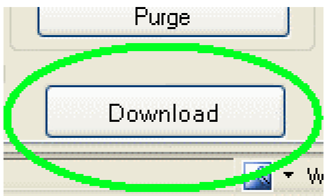
Until *Loop Done* is reached at the end of completing the number of loops in *Loop Count*, the module repeats the waveform continuously. There is no latency between consecutive waveforms within a burst. The following waveform starts right after the end of the preceding waveform. After the module receives the next *trigger* signal, it repeats this process. The waveform generation can be aborted by an Abort command from the GUI or API-based applications.

» With the program open, click on the **Waveform** tab if you are not already in the Waveform Panel.

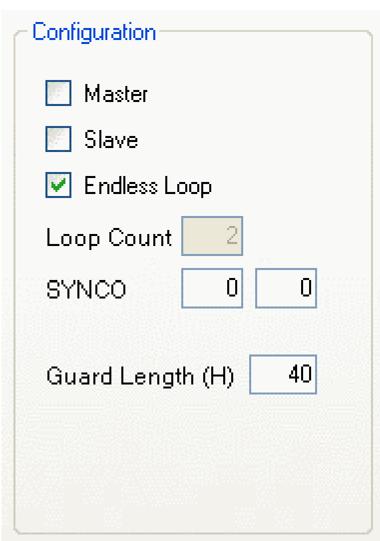
» Click on the drop-down list and select one of the saved waveforms, such as "1/128 Sine (my_wfa2.wfa)", which is a Sine wave at 1/128ths of the clock frequency. The files have the extension .wfa and are stored in your application folder or the user-defined waveform directory. You can adjust various parameters if desired.



Click on **Download Waveform** near the bottom of the window or click on File --> Load Waveform. Wait several seconds for the file to finish downloading.



» Click on the **Configuration and Status** tab. In the Configuration section, check the box for Master, and disable the checkbox for Endless Loop. Enter a value for *Loop Count*.



Then press the *Arm&Ready* button in the toolbar, which is the 5th button from the left in the area between the menu bar and the panel tabs.

Finally, apply a *trigger* signal, either via the TRIG SMA on the module or by pressing the Trigger button, which is the 7th button in the toolbar.

Single Board Operation -

Free Run Mode -

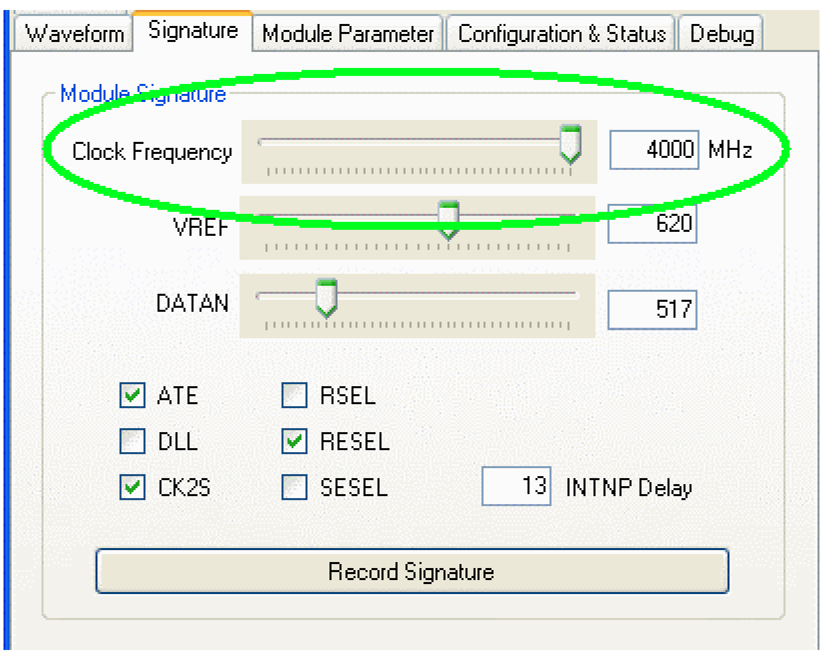
In Free Run mode, the module starts waveform generation by a Restart command from the GUI or from API-based applications. Once the waveform starts, the module repeats the waveform continuously. There is no latency between two consecutive waveforms. The subsequent waveform starts right after the end of the preceding waveform. The waveform generation can be aborted by an Abort command from the GUI or API-based applications will output a continuous waveform without pause. From the GUI, you can use your computer to choose the waveform or make other parameter changes in free run mode.

Free Run Mode Operation Example:

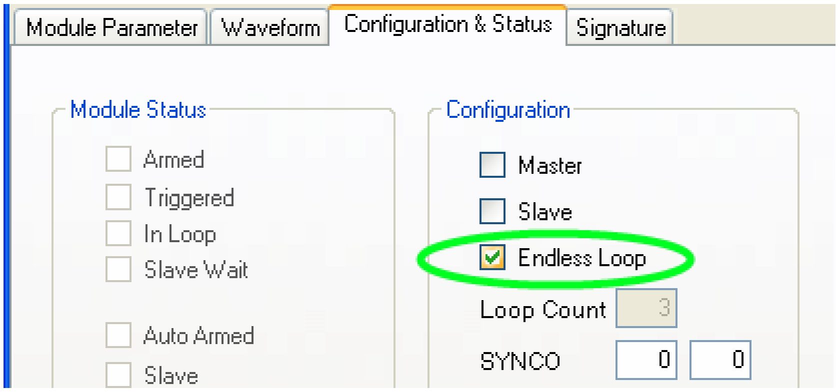
» Power up the module by plugging in the power supply. Remember that you should install them as specified in the Hardware Setup page. Connect the clock source, set to 4.0 GHz frequency and 3dBm power, to CKIP on the module. Soon after the module powers up, you should see the pre-stored waveform displayed in the oscilloscope with the corresponding spectrum in the analyzer. The standalone waveform can be customized upon request.

» Open the AWG application. You should have already set up the software and drivers and have connected the module to a USB port on the computer. Upon loading the AWG application, the module will stop producing the pre-stored waveform.

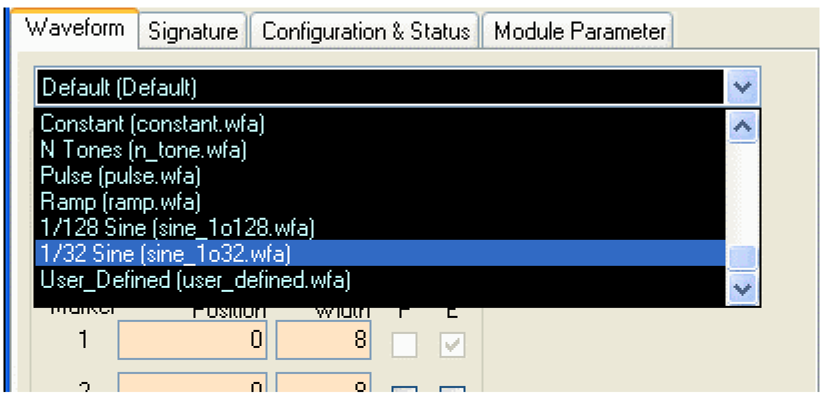
» Adjust the clock in the Signature panel. For our example, we have set the clock to 4.0 GHz since our clock input is at 4.0 GHz.



» Click on the "Configuration & Status" tab to go to that panel. Make sure that the configuration is **Endless Loop**.



» Click on the "Waveform" tab to go to that panel. Click on the drop-down list and click on the "1/32 Sine (sine_1o32.wfa)" waveform.



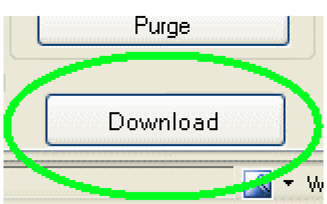
» The waveform, a Sine wave at 1/32nds of the clock frequency, should load, and you should already be able to see the output in the oscilloscope and the spectrum analyzer. Right-click the mouse and select "Refresh" if necessary to refresh the waveform.

Note that DIV B is "20," which is the hexadecimal representation for decimal 32.

» We are going to change the waveform to a Sine wave at 1/128ths of the clock frequency.

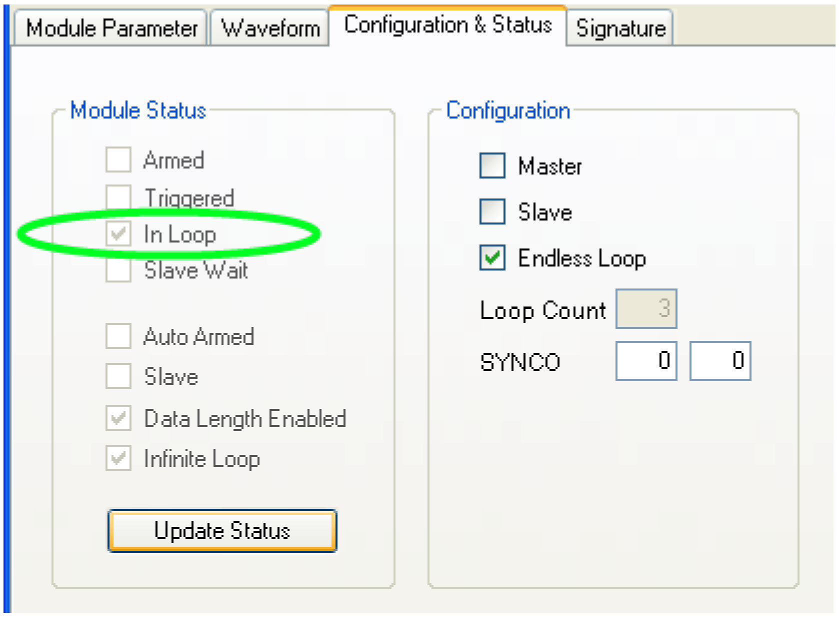
» Change DIV B to "80", which is the hexadecimal equivalent of decimal 128.

» Now click on **Download Waveform**.



» The outputs on the oscilloscope and analyzer will change. In the oscilloscope, you will see that the Sine wave frequency has decreased to 1/4th of the previous frequency. In the spectrum analyzer, the frequency should have shifted to 1/4th of the previous frequency.

» If you go to the "Configuration & Status" panel and click on **Update Status**, the application will show that it is in the "In Loop" state. This is correct since the board only produces output waveforms in the "In Loop" state as discussed on the Internal States page.



Master Mode -

The module can be configured as a Master module to activate other modules configured as Slave modules. By default, the module is in Master mode. To configure the module in Slave mode, the Master property can be turned off via the GUI or the API-based applications.

In Master mode, a few more options are available to you. You now have the ability to control the number of waveforms that are output (see Burst Mode) as well when to start ouputting waveforms (see Triggered Free Run Mode).

In Master mode, you can also synchronize the module with one or more Slave modules in multi-board operation.

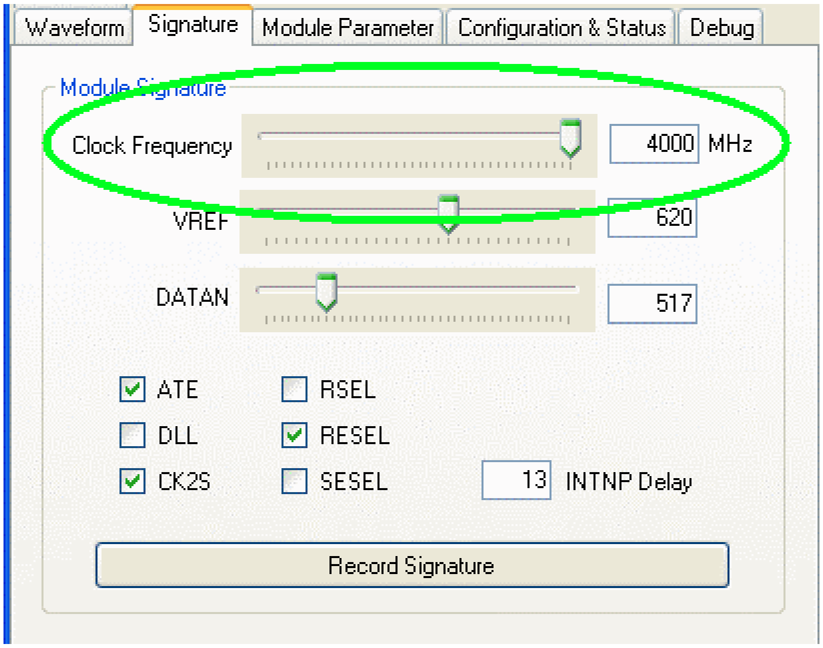
Besides the usual CKIP, OUTP and OUTN connections, you can now connect to the input TRIG connector and the board can output signals from the SYNCOUT connector. As a master, after receiving the trigger signal, the module generates a *SYNCO* signal, which can be used to activate any slave modules.

Master Mode Operation Example:

» Power up the AWG board by turning on the power supplies. Remember that you should install them in the right order as specified in the Hardware Setup page. Connect the clock source, set to 4.0 GHz frequency and 3dBm power, to CKIP on the AWG module. When the module powers up, you should see the demo waveform displayed in the oscilloscope with the corresponding spectrum in the analyzer.

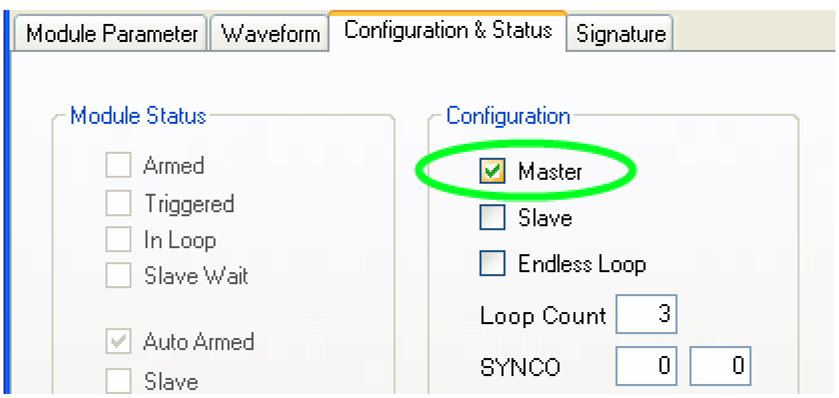
» Open the AWG application. Upon loading, the board will stop producing the demo waveform.

» Adjust the clock in the Signature panel. For our example, we have set the clock to 4.0 GHz since our clock input is at 4.0 GHz.



» Click on the "Configuration & Status" tab to go to that window. Change the configuration so that only **Master** is checked. Enter a value for **Loop Count**. Here we used 3, instructing the module to produce 3 sets of the waveform after a trigger signal. After the 3rd waveform, the board will not output anything while it awaits another trigger signal.

For an explanation of these parameters, please see the Internal States section of the Configuration & Status page.

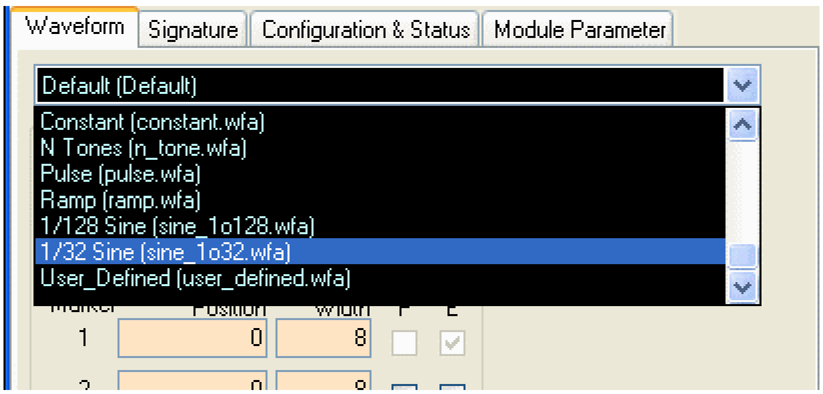


Auto Armed - since this is enabled, after outputting the waveforms, the AWG board will automatically go back to the "Armed" state and will wait for the next trigger signal. If this were disabled, then the board would go into the "Disarmed" state following a waveform output, and the user would need to arm the board manually by clicking on the **Arm & Ready** button. For an explanation of the Armed state, please see the Armed section of the Internal States page.

SYNCO - when the Master box is checked, the SYNCO box will also be checked, and the board will output the *SYNCO* signal which is normally used to control a Slave board, but the user can ultimately decide for what it will be used. The value in the first box to the right of SYNCO is T_{SYNC1}, which is the delay time between the falling signal of the trigger and the falling signal of the *SYNCO*. The value in the second box to the right of SYNCO is T_{SYNC2}, which is the amount of time for *SYNCO* to stay low before going back to high. You can change the T_{SYNC2} value to "FF" to see it clearly in an oscilloscope.

Waveform output is controlled by the *SYNCI* signal, which mirrors the *SYNCO* signal. If the module were in slave mode, it would only commence waveform output if it received a falling signal from the SYNCIN connector on the board.

» Click on the "Waveform" tab to go to that window. Click on the drop-down list and click on the "1/128 Sine (my_wf2.wfa)" waveform. Note that the number of periods of sine in this waveform will depend on the data length.



» Press the Download button at the bottom of the waveform panel. Wait a few seconds for the waveform data to finish downloading to the board. Then press the Restart button.

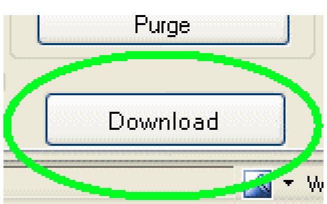
The waveform, a Sine wave at 1/128ths of the clock frequency, should begin to be output, and you should already be able to see the output in the oscilloscope and the spectrum analyzer.

Note that DIV B is "80," which is the hexadecimal representation for decimal 128.

» We are going to change the waveform to a Sine wave at 1/32nds of the clock frequency.

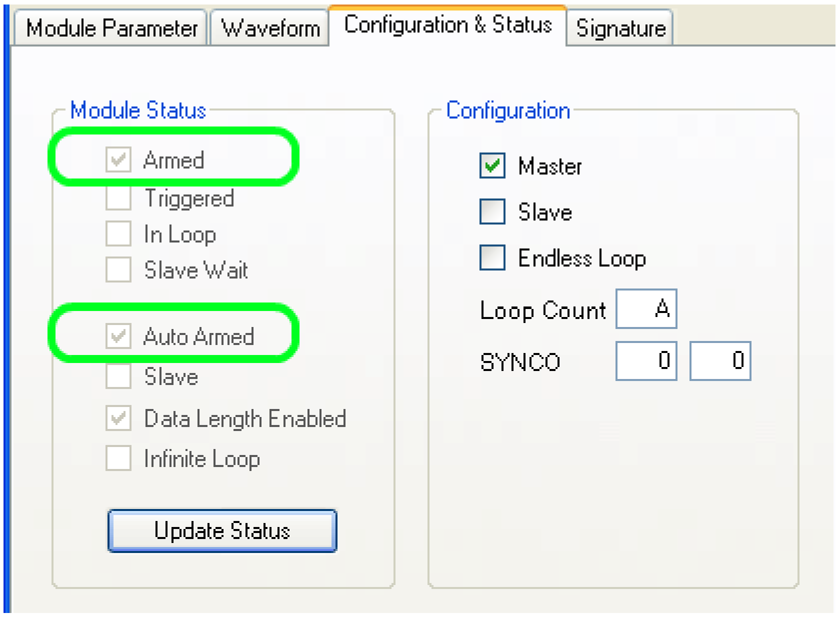
» Change DIV B to "20", which is the hexadecimal equivalent of decimal 32.

» Now click on **Download Waveform**.

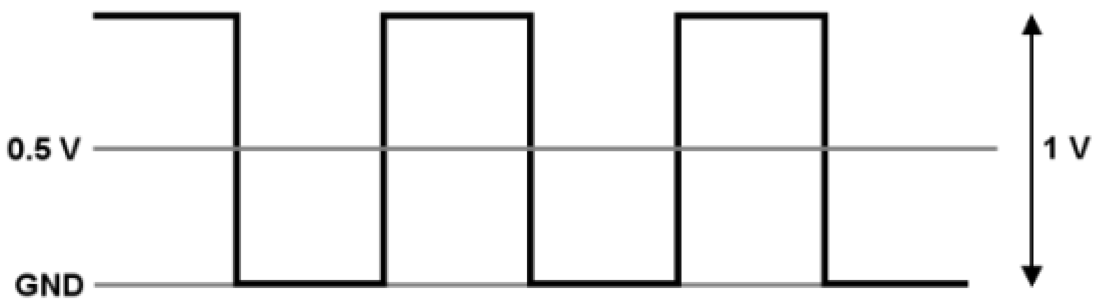


» You will notice from the oscilloscope and analyzer displays that the outputs have not started yet. This is because the board has not been triggered yet. If you go to the "Configuration & Status" panel and click on

Update Status you will notice that the **Auto Armed** and **Armed** boxes are checked. This means that the board is in the Armed state and is awaiting a trigger signal to begin waveform output.



» At this point you must either provide a trigger signal or press the TRIG button on the AWG board. The trigger signal we used in our lab was a 1 kHz 1 V peak-to-peak square wave with a DC offset of 0.5 V (assuming a 50 ohm load). You should use a similar signal for your trigger. The frequency of the trigger, of course, is up to you.

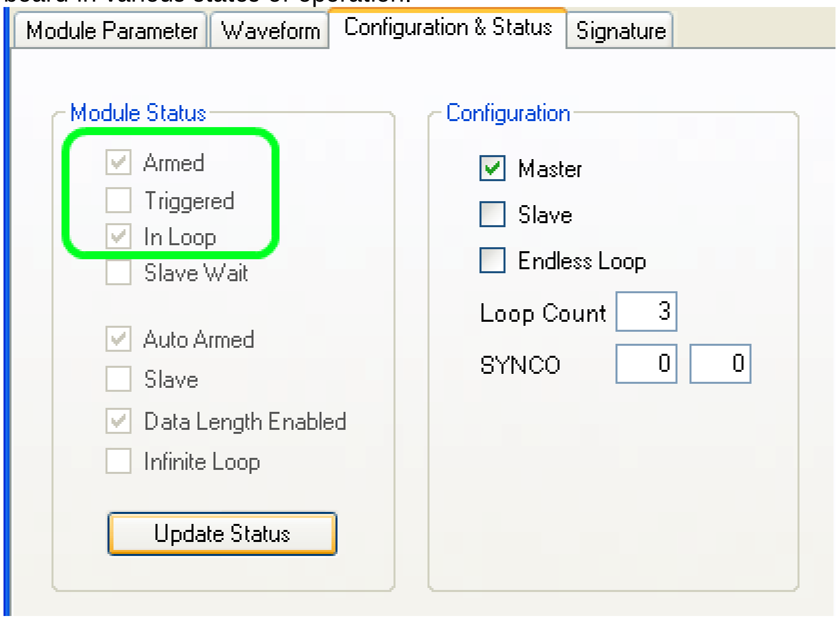


» Once the board receives the trigger signal, the outputs on the oscilloscope and analyzer will change. In the oscilloscope, you will see that the Sine wave frequency has slowed to one quarter of the previous frequency. In the spectrum analyzer, the frequency should have shifted to one quarter of the previous frequency. Also, the waveform will appear in a burst of 3, where one waveform consists of several periods of a sine wave. The length of this waveform depends on the data length.

» The *MARKER* signal is output every time a waveform is output. In our current setup, we have set *Loop Count to 3* waveforms, so there should be 3 signal pulses for the MARKER signal.

» If you connect SYNCOUT to an oscilloscope you can view its signal. The *SYNCO* signal (the signal that is output from the SYNCOUT connector) is the signal that the board normally sends out to Slave boards in multi-board operation. To see the *SYNCO* signal more clearly, make the second box to the right of SYNCO "FF" and then click on the **Restart** button.

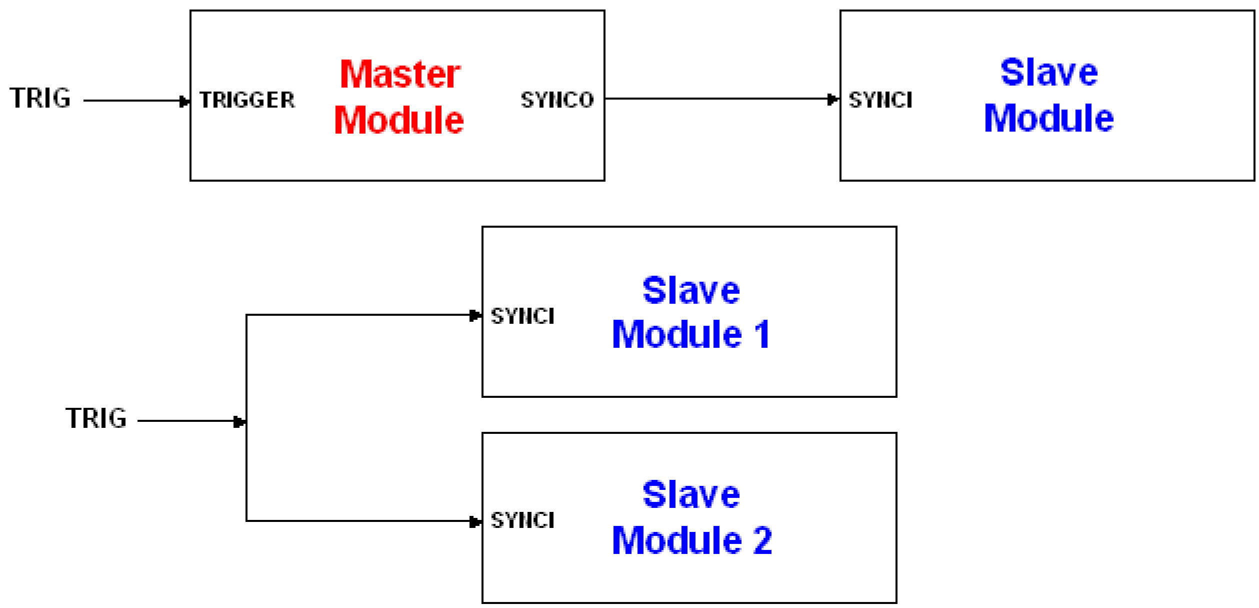
» If you go to the "Configuration & Status" window and click on **Update Status** repeatedly, you will notice the board in various states of operation.



Multiple Board Operation -

The module can be configured as a Master module to activate other modules configured as Slave modules. By default, the module is in Master mode. Connecting two or more boards in a Master-Slave configuration facilitates synchronization between the output waveforms.

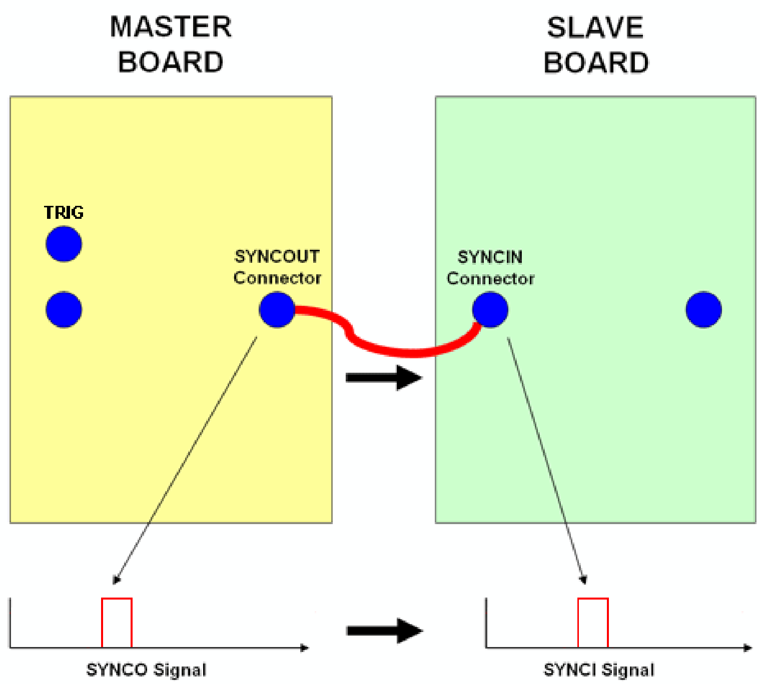
Synchronization
Configurations
Operation Example



Synchronization -

Synchronization between two boards is done through the SYNCOUT and SYNCIN connectors. We call the signal sent through the SYNCOUT connector *SYNCO* and the signal sent through the SYNCIN connector *SYNCI*, so we do not confuse the connectors and the actual signals. For multiple board operation, there must always be one Master board. The Master board will send out the *SYNCO* signal through the SYNCOUT connector and the Slave boards will receive that signal through their SYNCIN connectors. Inside the Slave boards, the signal from SYNCIN is called *SYNCI*. The Slave *SYNCI* signals will always match the Master *SYNCO* signal.

When an armed master module receives a trigger signal, it generates a *SYNCO* signal to start the waveforms of itself and of the slave module. The *SYNCO* delay T_T is about 1~3 µs and the waveform delays, T_M and T_S, are typically 128 ns and can be adjusted via the GUI or API-based applications. In applications, the slave waveform delay, T_S, can be adjusted to compensate for the cable delay between interconnection of modules. To minimize delays between the trigger signal and waveform generation, the modules can be configured in slave mode and receive the trigger signal via their *SYNCI*'s as shown in the following figure.

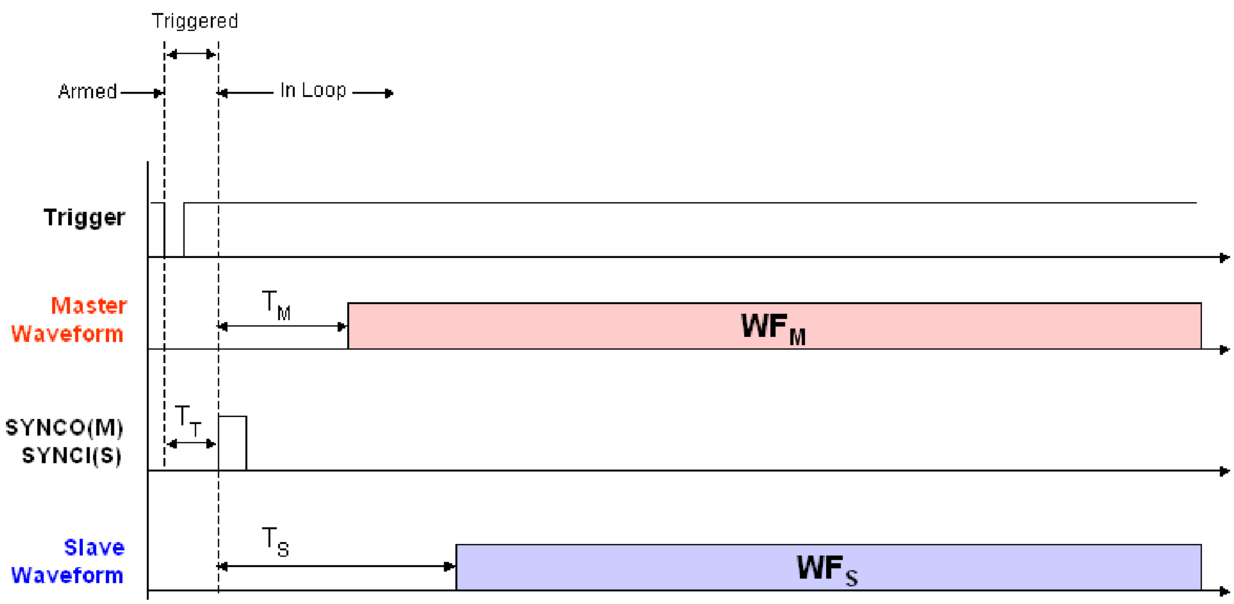


It is important to note that waveform generation starts ONLY based on the *SYNCI* signal (and only when module is in the "Triggered" state), even in the Master board or when in Standalone operation. The user may then wonder where does the Master board or the Standalone board receive its *SYNCI* signal? The *SYNCI* signal in the Master board, or when in standalone operation, mirrors the internally generated *SYNCO* signal.

Due to gating delays, there will be a slight difference between the SYNCI signals on each board. You can compensate for this difference by adjusting the Guard Length.

You may split the *SYNCO* signal to more than one slave board, but there can only be one Master board, and only the Master board requires a trigger signal.

The diagram below illustrates the signal timeline of a Master and Slave board. The board is assumed to already have been in the "Armed" state, where it remains until the *TRIG* signal falls and the board enters the "Triggered" state. Then, after the SYNCI/SYNCO signal rises, the board is in the "In Loop" state until it finishes running through the Loop Count parameter.



The trigger signal at the top can represent either the input at the TRIG SMA connector or a software trigger generated by pressing the trigger button in the GUI toolbar. You must have a trigger to generate the Master board's *SYNCO* signals. Before the rising edge of the *SYNCI* signal, the board is considered to be in the “Triggered” state.

The *SYNCI* signal mirrors the *SYNCO* signal so both signals are always the same. The Slave board does not have a *SYNCO* signal. Since the *SYNCO* and the *SYNCI* signals are the same, for convenience we will use "SYNC" when referring to either of the signals.

T_T

The delay between the falling edge of the *TRIG* signal and rising edge of the *SYNCI*/*SYNCO* signals is T_T. It is typically 1-3 ?s.

When the *SYNCI* signal rises from logical level low to high, the board changes state from "Triggered" to "In Loop". Waveform output does not start immediately but is instead delayed for a brief time, T_M for the Master board and T_S for the Slave board(s).

T_M

The delay between the rising edge of the *SYNCI*/*SYNCO* signals and the beginning of waveform output on the Master board is T_M. It is typically 128 ns with the default Guard Length.

T_S

The delay between the rising edge of the *SYNCI*/*SYNCO* signals and the beginning of waveform output on the Slave board(s) is T_S. It is typically around 32 ns longer than T_M, but can easily be shortened to match the Slave board waveform output with the Master board waveform output by shortening the Guard Length.

After the SYNC signals go back to low, the board will return to "Triggered" state when the waveform loop is completed if **Auto Arm** is enabled.

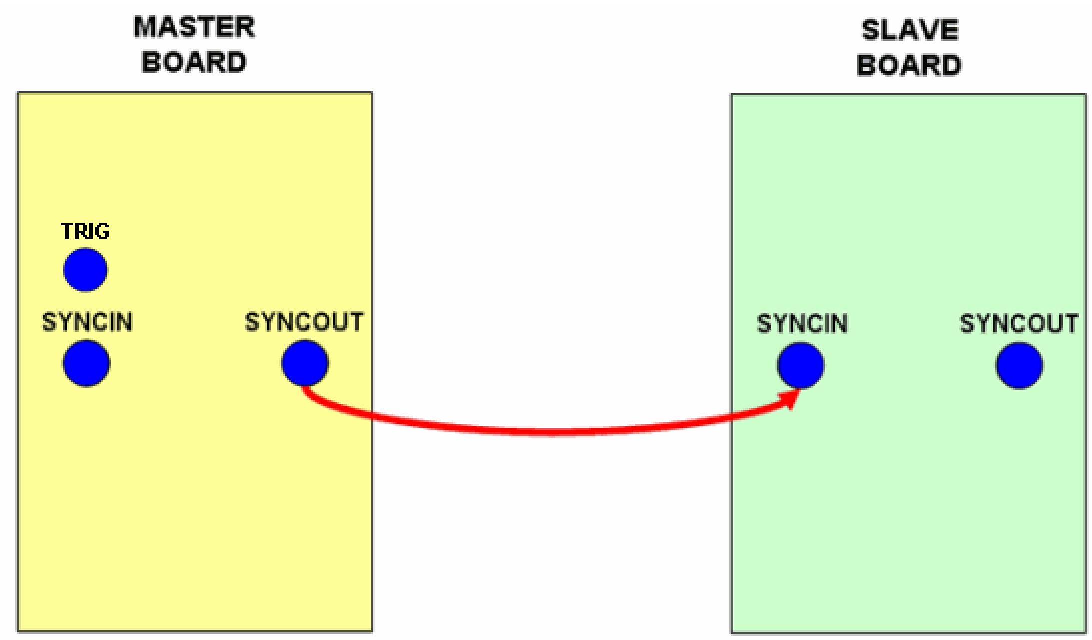
Configurations -

There are a few ways to setup master and slave boards. Remember to set up a trigger signal for the Master board.

Single Master, Single Slave

Short Connection between Master and Slave

For a single master, single slave configuration, simply connect the SYNCOUT on the master to the SYNCIN on the slave with a SMA cable.



If the cable is fairly short, the SYNCO and SYNCI signals will be the same. When the user sends a trigger signal, the boards will begin outputting waveforms.

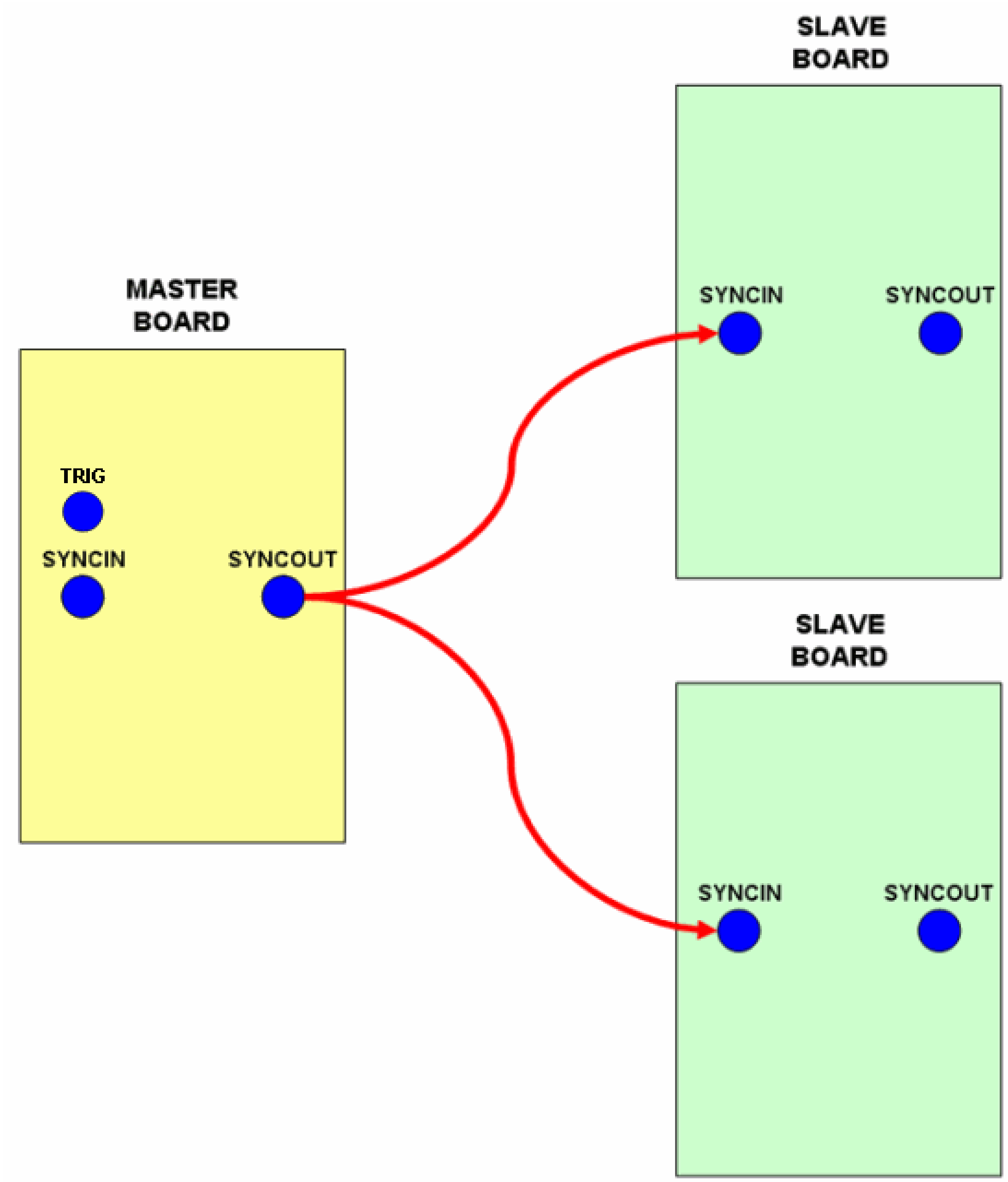
Long Connection between Master and Slave

If the cable connecting master and slave is very long and lag time is a concern, you can adjust the Guard Length.

This way, the lag times will be the same and the boards will have a synchronized output.

Single Master, Multiple Slaves

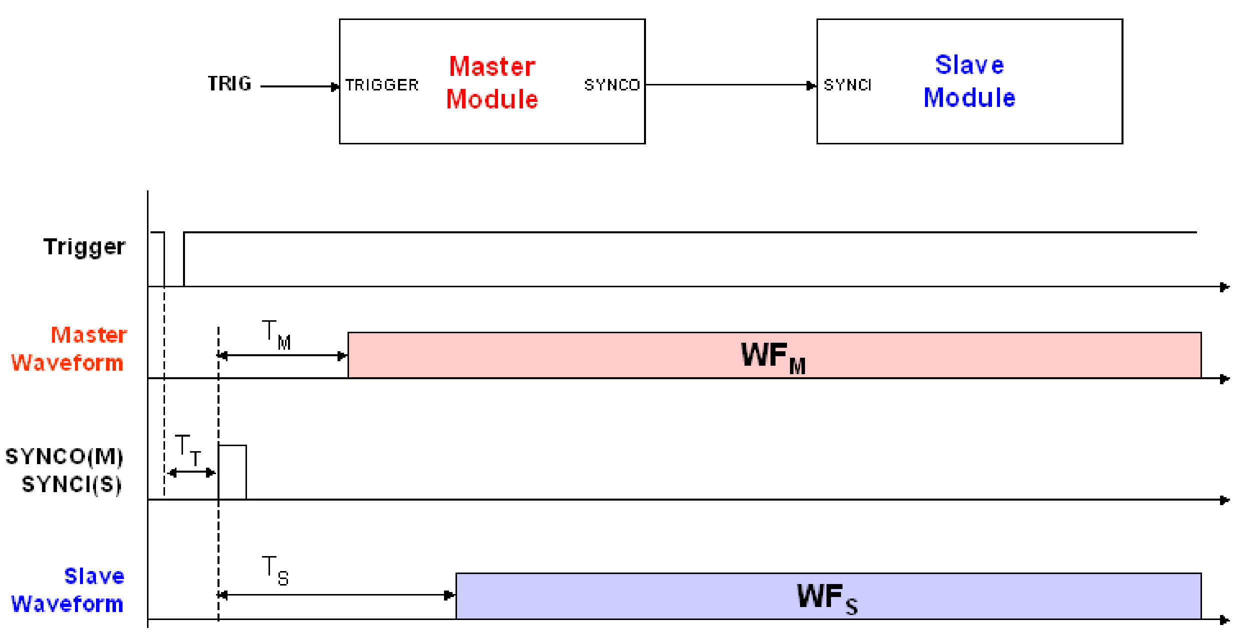
For multiple slaves, we split the SYNCOUT signal from the master board as demonstrated below:


SLAVE
BOARD
SYNCIN
SYNCOUT
MASTER
BOARD
TRIG
SYNCIN
SYNCOUT
SLAVE
BOARD
SYNCIN
SYNCOUT

Operation Examples -

The module can be configured as a Master module to activate other modules configured as Slave modules. By default, the module is in Master mode. To configure the module in Slave mode, the Master property can be turned off via the GUI or the API-based applications.

As a master, after receiving the trigger signal, the module generates a *SYNCO* signal, which can be used to activate the slave modules. The configuration is shown in the following figure.



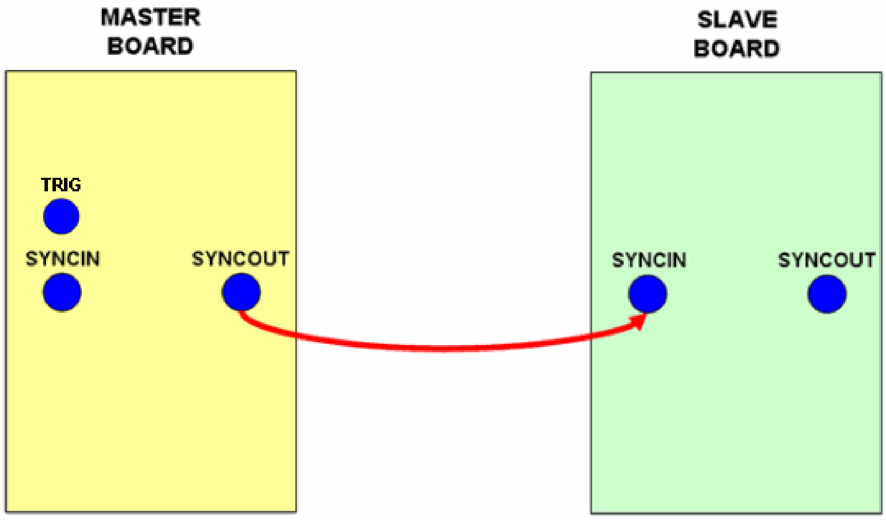
Master with one slave

We will now present an example of setting up one Master board to control one Slave board.

» Plug the power supplies into each board, ensuring that the power source is capable of twice the current requirements as for one board.

» Connect the necessary SMA cables to the Master and Slave boards. Note that you must connect the SYNCOUT on the Master Board to SYNCIN in the slave board. The input clock power should be increased by 6dB and then split via a 6dB power divider and input to both CKIP inputs on the Master and Slave boards.

If you are using a trigger signal, connect it to the TRIG connector on the Master board. The Slave board will ignore any TRIG signal, so it is not necessary to split the TRIG signal. If you are not using both of the OUTP and OUTN outputs on both boards, you may terminate the unused ones with 50 ohm terminal resistors.



» Now connect both boards to the computer using two USB A-to-B cables.

» Power up the AWG boards by plugging in the power supplies. Remember that you should set up your power connectors in the right order as specified in the Hardware Setup page.

» You will need to open two instances of the AWG application. When you open the first AWG application, a dialog box will pop up allowing you to choose the board number.

Each board is identified by a series number. We will make one our Master board and the our Slave board. In this case, we select #4 and click **OK**.

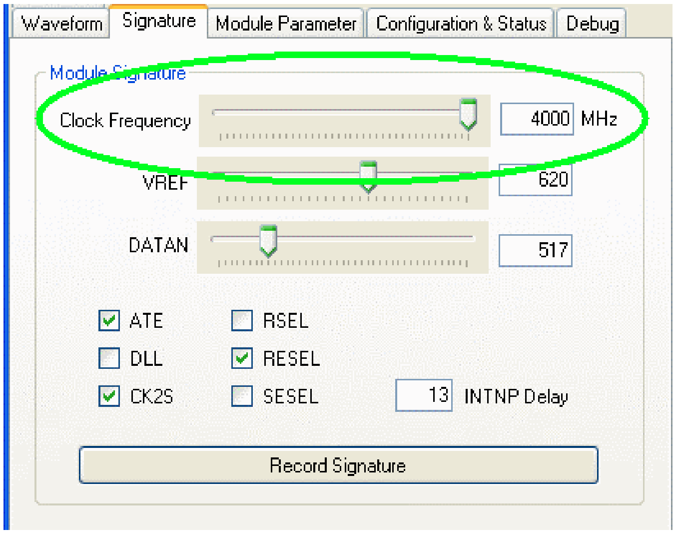
» The AWG application will now open. If you look at the title bar, you will notice that it has the board number so that you can identify the boards easily.

» Now open up a second AWG program. Again the dialog box will appear.

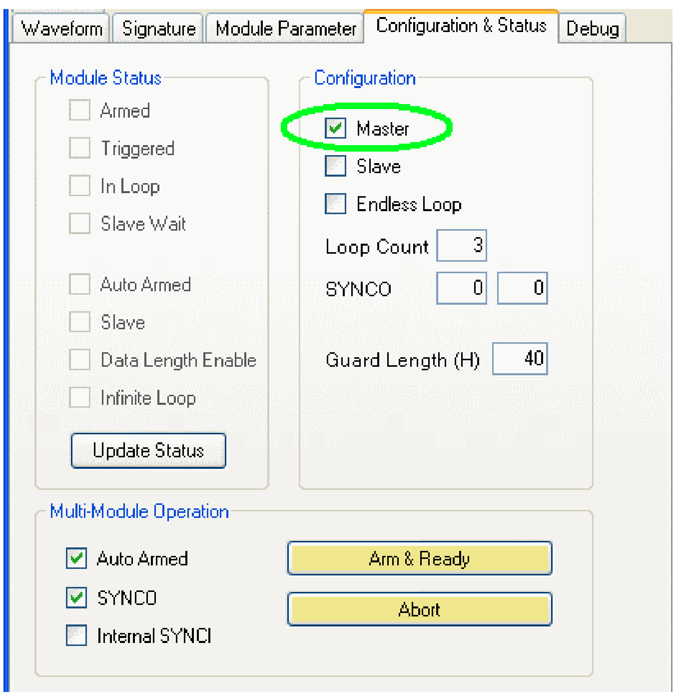
Choose the series number of the second board and then click on **OK**.

» Once again you will notice that the title bar will have the board's number.

» Go back to the Master board application. Adjust the **Clock Frequency** so that it matches the input clock frequency. Our clock is at 4.0 GHz, so we will set it to "4000" with the units in "MHz". Repeat the same procedure with the Slave board application.

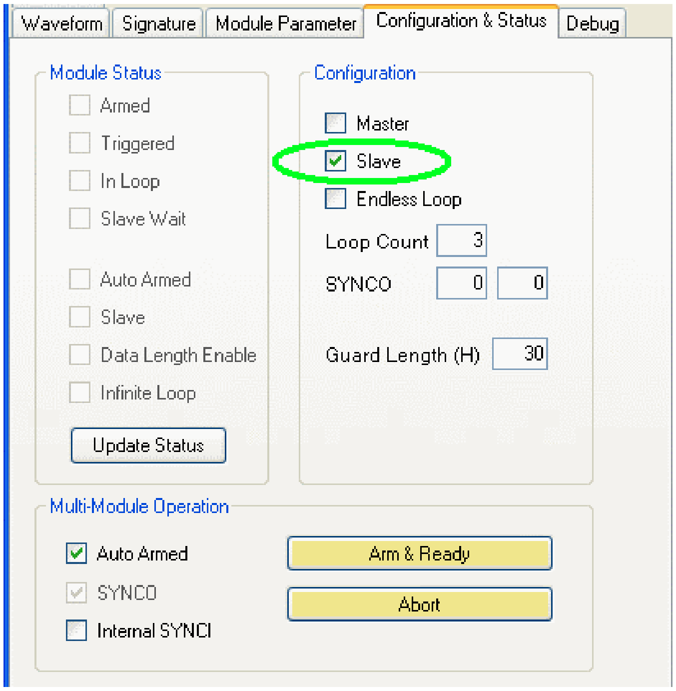


» Go back to the Master application and click on the "Configuration and Status" tab. Change the configuration so that only **Master** is checked. Leave the Guard Length unchanged at 40 (hex).



» In the Master application, make sure that **Auto Armed** and **SYNCO** are checked. Also, you will notice that **Loop Count** is set to "3". For an explanation of these parameters, please see the Internal States section in the Module Parameters page.

» Now go to the Slave application and click on the "Configuration and Status" tab. Change the configuration so that only **Slave** is checked. Change the Guard Length to 30 (hex). You may need to adjust this Guard Length after you see the waveforms.

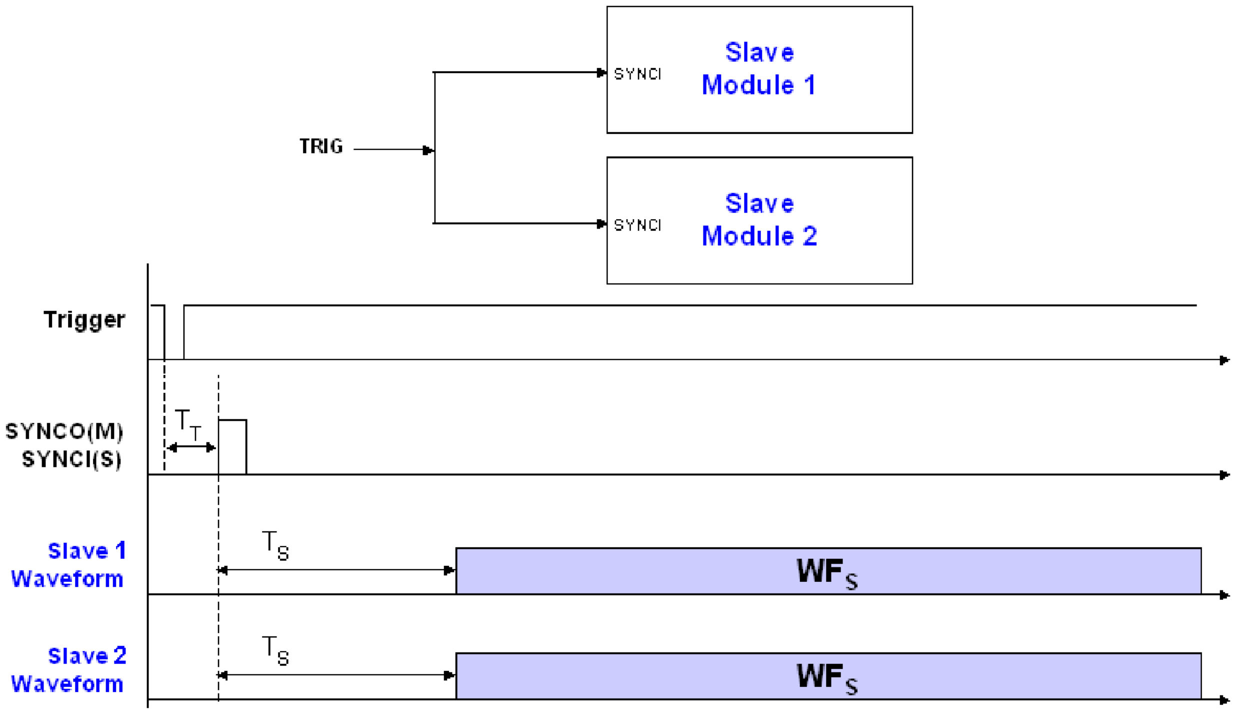


» In the Slave application, make sure that **Auto Armed** is checked. Although the **SYNCOUT** checkbox is available, it has no effect on the board even if you change any of the values or enable it. Slave boards never output a *SYNCO* signal. Also, you will notice that **Loop Count** is set to "3".

The Slave board does not have to mirror the output of the Master board. The Master board merely synchronizes the Slave board so that they begin outputting waveforms at the same time. The two boards can output two different waveforms at the same time and even different numbers of waveforms so you can change **Loop Count** to whatever value you desire as long as it is LESS than or equal to the loop count of the Master board.

Two slaves synchronized

We will now present an example of setting up two Slave boards. They will be synchronized because they have the same T_S delay. The *SYNCI* signal should come from an external trigger source (which can be a Master board or just a positive pulse from a function generator).



» Plug the power supplies into each board, ensuring that the power source is capable of twice the current requirements as for one board.

» Connect the necessary SMA cables to the two Slave boards. Note that you must split your trigger source to connect to the SYNCIN connectors on both slave board. Since *SYNCI* is triggered by the rising edge, this trigger signal is the opposite of the falling edge trigger signal used to trigger a Master board. The input clock power should be increased by 6dB and then split via a 6dB power divider and input to both CKIP inputs on the Slave boards.

If you are not using both of the OUTP and OUTN outputs on both boards, you may terminate the unused ones with 50 ohm terminal resistors.

» Now connect both boards to the computer using two USB A-to-B cables.

» Power up the AWG boards by plugging in the power supplies. Remember that you should set up your power connectors in the right order as specified in the Hardware Setup page.

» You will need to open two instances of the AWG application. When you open the first AWG application, a dialog box will pop up allowing you to choose the board number.

Each board is identified by a series number. In this case, we select #4 and click **OK**.

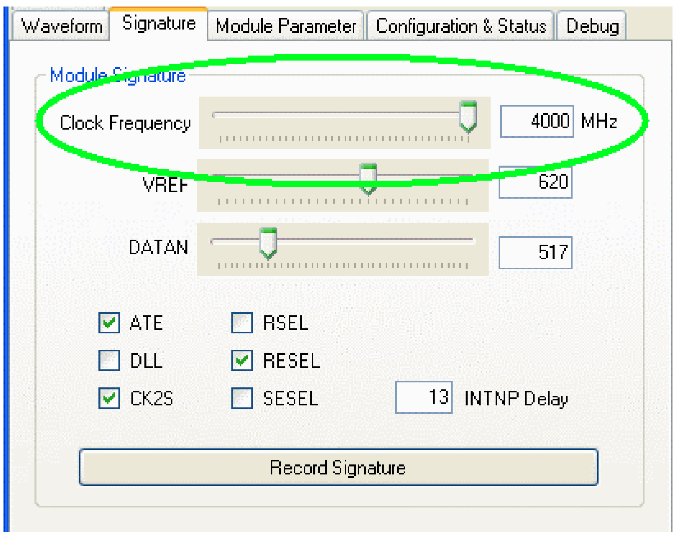
» The AWG application will now open. If you look at the title bar, you will notice that it has the board number so that you can identify the boards easily.

» Now open up a second AWG program. Again the dialog box will appear.

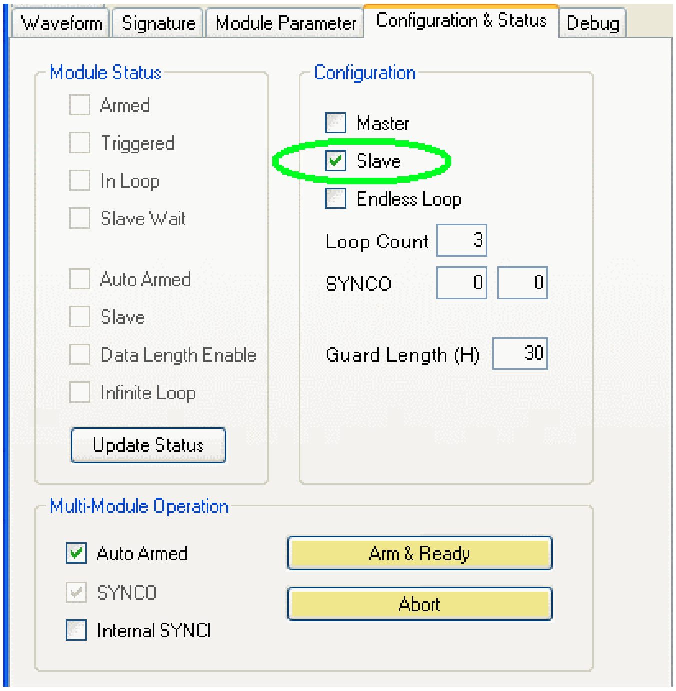
Choose the series number of the second board and then click on **OK**.

» Once again you will notice that the title bar will have the board's number.

» In the signature panel of each application window, adjust the **Clock Frequency** so that it matches the input clock frequency. Our clock is at 4.0 GHz, so we will set it to "4000" with the units in "MHz". Repeat the same procedure in the other application window.



» In the Configuration & Status panel of each application window, change the configuration so that only **Slave** is checked. You can leave the Guard Length unchanged. But, if you change it, please ensure that it is changed to the same value in the other application window.



» In each application, make sure that **Auto Armed** is checked. Although the **SYNCOUT** checkbox is available, it has no effect on the board even if you change any of the values or enable it. Slave boards never output a *SYNCO* signal. Also, you will notice that **Loop Count** is set to "3".

The two Slave boards doe not have to match each other. This process merely synchronizes the Slave boards so that they begin outputting waveforms at the same time. The two boards can output two different waveforms at the same time and even different numbers of waveforms, so you can change **Loop Count** to whatever value you desire.

Application Programming Interface -

A separate API manual will describe how you can design your own application programming interface (API) to use instead of the AWG_WIN.exe graphical user interface application.

Troubleshooting -

No LED's are lit

This means there is no power to the board. Check your power connections. For details, please consult the power supply requirements. After turning power on, the +5V LED should be lit. If the USB is also connected, then the LED below the +5V LED will also be lit. These two LED's should remain lit as long as the module is connected to power and to the USB port on your computer. Within a few seconds of powering up the board, and a third LED will light or blink to indicate the data writing phase, after which the default waveform should begin.

Waveform or spectrum has extra repeated data

If the waveform or its spectrum appears to be correct but has parts of the data repeated, perhaps the Data Length does not match the actual number of data points in your waveform. See the Data Length page for a description of how that parameter can affect the waveform.

Waveform or spectrum has some errors

If the waveform or its spectrum appears to be nearly correct but has errors, make sure the clock frequency of your clock source matches that specified in the computer application. Then, adjust the VREF, DATAN, ATE, and DLL settings in the Signature panel. and or the SEL setting in the Module Parameter panel.

If this doesn't fix all the errors, perhaps data was lost under an inferior signature setting, and you may need to reload the waveform data.

Also, you may try setting your clock source to 1 GHz frequency and 3dBm power and repeating the above steps, in case your clock source settings were beyond the range of the board capabilities.

Try to abort and restart the waveform by pressing the Abort button near the bottom of the Module Parameter page and then pressing the Restart button.

If the problem persists, troubleshoot further by setting the waveform to a ramp, downloading it, restarting, and adjusting the signature settings. If any memory bits are consistently bad, they will be apparent in this linear function.

Waveform did not update after I changed parameters

Press the Download button (at the lower right corner of the waveform panel) to download the parameterized waveform data to the AWG memory to allow the firmware to generate the actual waveforms. You must press this button each time you finish changing the parameters. Remember to wait for the waveform to finish downloading before you press the Restart button.

Current changes when clock frequency changes

This is perfectly normal. The memory requires more power at higher clock frequencies and for various waveforms, so you can observe that the 1.8V supply current will increase dramatically as you increase the clock frequency.

Current changes a few seconds after startup

This is perfectly normal, and you may not even notice it. The 1.8 V power supply current will be much lower when you first power on the AWG, but then the current will increase after the module achieves regular operation within a few seconds. The current will also increase significantly with clock frequency.

The hardware setup page lists the typical operational voltage and current.

Memory is underpowered

Check the voltage at the power plug (header) for the 1.8V supply. If the wires are too long from your power supply, the resistance will cause a voltage drop, and the voltage at the board will be too low. In this case, please shorten your wires. If this is not practicable, then you may need to raise your power supply voltage slightly to compensate for the voltage drop.

Module ignores trigger and continues to output waveform

You may have pressed the Arm & Ready button twice while the module was waiting for the trigger. This causes the module to generate a hardware trigger. The waveform output becomes endless because the processor is fully occupied and ignores the loop count or trigger requirements.

Simply press the Abort button and then Arm the board again (only once).

Waveform appears shaky

Shorten the SYNCO width, which is the second box after SYNCO on the Module Parameter panel. Try setting it to 0. Or, try using a waveform with a longer data length. If SYNCO is too wide, or if the data is too short, the module may not have enough time to get the waveform data (for the next loop) ready at the next trigger, resulting in a possible delay.

The Module Status does not change

You must press the Update Status button to see the current status of the module. The status monitor does not automatically update.

Waveform is truncated

When your waveform data length exceeds the total memory depth, the data will be truncated.

Waveform amplitude is truncated in multi-tone (n-tone) waveform

When the amplitudes are too high the multi-tone waveform (Waveform code 4), the instantaneous amplitude of the envelope waveform may at times exceed the full scale amplitude. If this occurs, the overflow data will be truncated and may appear to have wrapped around to a different amplitude.

Decrease the amplitudes in the second column of the table in which you specified the N Tones. Please see the Waveforms page for an example.

There are fewer than 15 user pages available.

One or more of your waveforms is using up more than one uPage memory partition. You can purge the memory, and then use data lengths shorter than 0x40000 for the AWG252 or 0x80000 for the AWG452, and re-download the data to each page. Please see paging for details on dynamic paging.

The inverse sinc filter has no effect on my waveform

You are probably using a digital waveform (pulse, ramp, or user-defined). Try to use an analog waveform, like a sine function or multi-tone waveform. Open the filter dialog box from the Options menu bar. Ensure that the filter status is "on." You can also try changing the windowing type. Remember to download the data again.

Windows Device Manager says Cypress Generic USB device -- EEPROM missing

Ensure that there is a jumper at JP3 and that the firmware is installed on the EEPROM if you have changed the firmware. Try turning the power off and on again.

The module firmware needs to be updated

Occasionally we update the firmware to improve the functioning of the board. If these updates are available, they can be downloaded from our website. To install the new firmware, please follow the instructions here.

What if I need to contact you for help?

In the event you need to contact us for help, please provide the software and firmware versions, as well as the model and series number. The software versions are available in the About page of the help menu in the AWG application. The model and series numbers are marked on the board itself. This information is also available in the Startup Notes provided with the module. The AWG software records log files in your program directory. If the program is not running properly, copying and saving these .log files before exiting the program may provide helpful information. Our contact information is located on the Contact Us page also found on the menu column on the left.

Contact Us -

Mailing Address:

Euvis, Inc.
685 Cochran St.
Suite 160
Simi Valley, CA 93065

Phone and Fax:

Tel: 805-583-9888
Fax: 805-583-9889

On the Web:

Email: info@euvis.com
Website: http://www.euvis.com