

How to Create Direct Synthesis Signals Using MATLAB and Arbitrary Waveform Generators

Every engineer's job involves a lot of problem-solving, and one of today's most challenging design problems is the creation of stimulus signals that can really exercise a new motherboard prototype or silicon turn. Emerging RF technologies in the wireless field demand complex high-frequency modulated signals. Serial buses need signals with jitter, spread spectrum clocking, and other time-variant effects. It is a situation whose solution calls for a benchtop full of pulse, function, modulation and RF generators or... a source capable of delivering all these diverse signal types individually and simultaneously. This is the premise of direct synthesis signal generation.

What Is Direct Synthesis, and What Can It Do For Me?

The term “direct synthesis” denotes a means of producing waveforms, and it has two implications.

The first is architectural. Direct synthesis reads waveform data directly from a memory and offers almost unlimited flexibility in the wave shapes it can deliver. Its sampling-based architecture essentially reverses what a digitizing oscilloscope does. Whereas the oscilloscope acquires sample points from analog waveforms, a direct synthesis signal source – known as an arbitrary waveform generator, or AWG – reconstructs analog waveforms from stored sample points. These samples can define essentially any waveform, from sine waves to serial digital pulses.

The second implication is application-focused. Many traditional signal-generating tools produce one type of waveform – say, an RF sine wave – and rely on external modulators to process the signal for the real needs of the Unit Under Test (UUT). In contrast, an AWG uses direct synthesis to originate a signal that innately includes any required modulation characteristics or transmission path effects such as jitter, noise, ISI, and so forth.

The AWG is gaining favor as a stimulus source for a variety of test and measurement applications. What are its advantages? Consider these:

- Because the stored signal information consists of digital samples, it can be modified in hundreds of ways: filtered, convolved, time-shifted, modulated, and more.
- The output frequency can be changed—without altering the waveform content—simply by changing the frequency of the sample clock.
- Aberrant characteristics such as transients or dropouts can be added anywhere in the output waveform stream. This is useful for providing “real-world” stresses during prototype validation, for example.
- The AWG’s direct synthesis techniques work in conjunction with design (modelling) tools used to simulate component and system performance.
- Available software editing tools dramatically simplify the process of developing and modifying waveforms for use in AWGs.

An emerging generation of AWGs offers both the bandwidth and the resolution to meet the need for the high frequencies common in telecommunications, radar, serial computing devices, and even consumer goods. Moreover, these new AWGs provide the simultaneous digital and analog signals that such devices often require during testing.

Figure 1 depicts a simplified AWG direct synthesis architecture, showing the steps that the waveform data takes as it evolves from binary words in the memory to a finished, usable output waveform. In this simple view, a single path symbolizes a differential pair. This is considered a single channel.

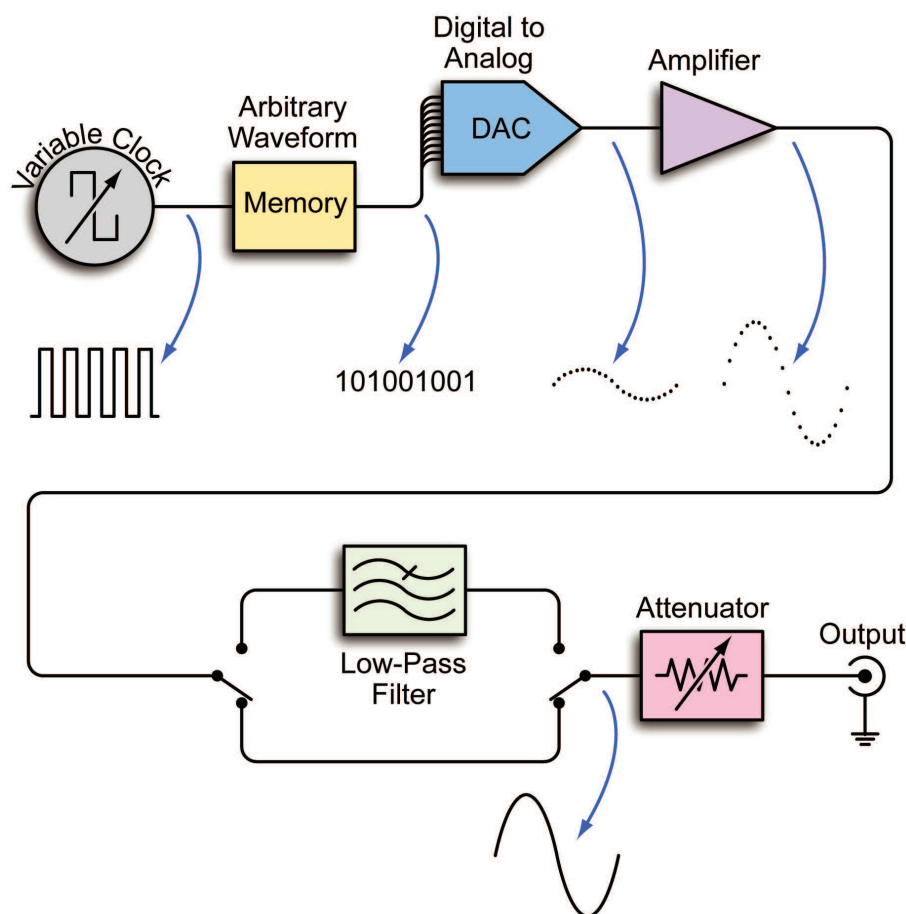


Figure 1. Simplified AWG direct synthesis architecture.

Creating Waveforms, from Equation to Output

Every AWG can generate waveforms—send electronic signals from its outputs—but many do not include on-board features intended to create waveforms. With the advancing need to design ever more complex signals, software-based development tools have become the preferred means of creating waveforms. The more comprehensive software tools offer the ability to archive and re-use code, to compute almost any wave shape from math functions, and to model the resulting waveform using DSP.

There are several kinds of waveform creation tools that span the continuum from the simplest to the most advanced. We will highlight several applications in this paper but concentrate mostly on MATLAB software for detailed discussions.

Standard Built-in Waveform Creation Tool

It is common practice among AWGs to include a set of basic functions stored permanently within the instrument for recall into the waveform memory. These waveforms are called functions because they are derived directly from simple math functions such as sine equations. Other shapes include triangles, ramps, and square waves. The onboard functions are a convenient way to perform “signal present” checks on both the AWG and devices attached to it, and they may suffice for other simple tests. The AWG5000 and 7000 Series have an on-board waveform creation tool that allows users to build standard function type waveforms such as those mentioned above.

How to Create Direct Synthesis Signals Using MATLAB and Arbitrary Waveform Generators

Application Note

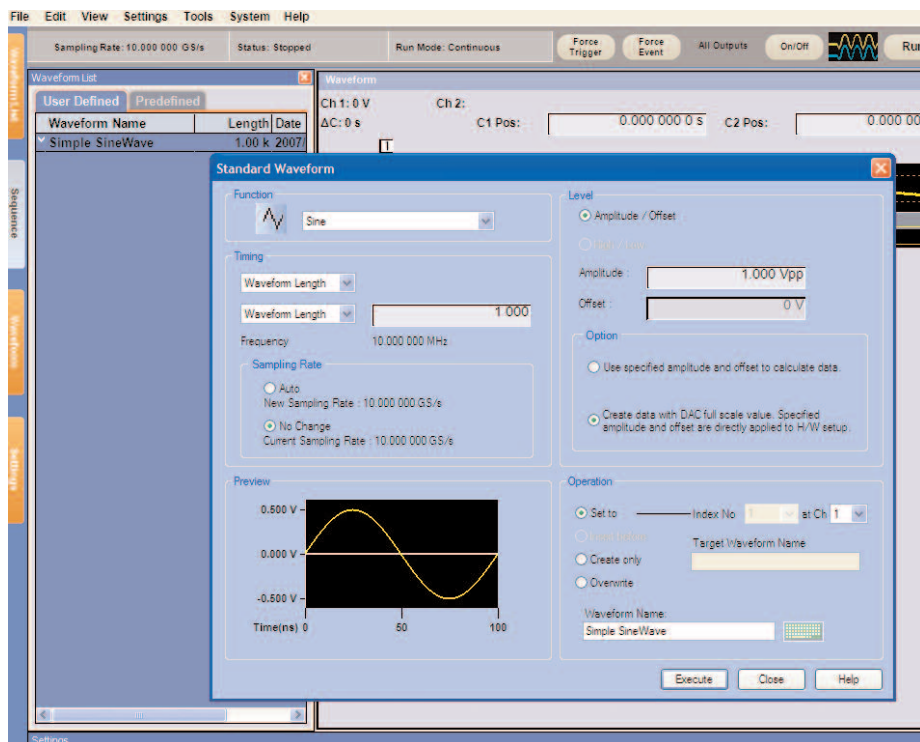


Figure 2. AWG standard waveform tool used to create a sine wave.

ArbExpress® Tools

ArbExpress consists of a basic suite of waveform creation and editing tools designed to run on the Windows® XP computing platform. Because both the AWG5000 and AWG7000 Series run on integrated WindowsXP computers, ArbExpress equation-based and graphical tools can be used to create waveforms within the instrument itself. Figure 2 depicts an ArbExpress Equation Editor window. Here it is shown creating a sine wave. The equation window (upper left) contains the actual equation and other details.

Specifically, the equation is as follows:

```
#Here w = 2*pi*10^4*t
#The Freq of the equation F= 1/((points/cycle) * clock)
#In this equation Freq is 1/1000*100ns = 10KHz i.e. 10^4
#View the waveform with the settings Points: 1K,SR,10MS/s
Range(0us, 100us)
sin(w)
```

Comment lines are highlighted here in blue for emphasis. Note that most of the text consists of comments; the operative function is simply “sin(w).”

A Graphical Editor window can be selected after the basic waveform is complete. This provides drawing tools that make it possible to modify the simplistic sine shape with intentional distortion or aberrations.

ArbExpress-created waveforms cannot be loaded directly into the memory of the AWG5000 or AWG7000 Series instruments. However, the data, when compiled, can be stored as a .TXT file or AWG-specific .wfm format in the hard drive of the instrument's PC host, and then imported without ever leaving the platform. ArbExpress has proven to be a good general-purpose waveform creation tool for a wide variety of basic applications. It is a free application available as a download from www.tektronix.com.

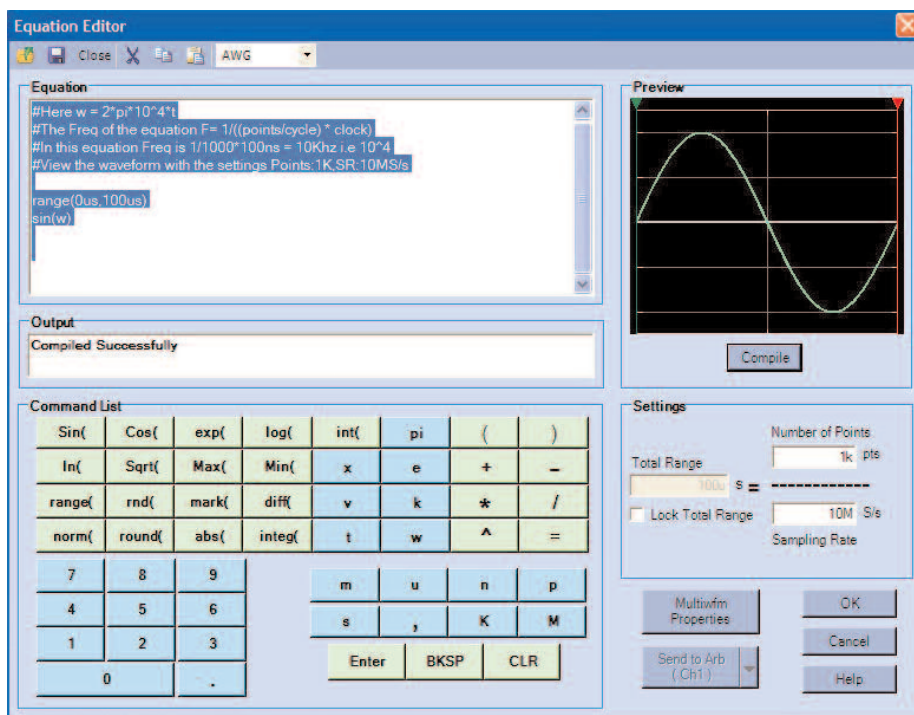


Figure 3. ArbExpress Equation Editor developing a sine wave.

SerialXpress® Advanced Jitter Generation Tool

SerialXpress® is a powerful yet easy to use software package that allows users to create complex synthesized serial data waveforms for the AWG5000 and 7000 Series. This software enables high speed serial data designers to develop waveforms with specific amounts of Random, Periodic, DCD jitter software components. Channel properties such as Pre-emphasis, ISI, noise and equalization. It also allows for creation of Spread Spectrum enabled signals. With the combination of SerialXpress and an AWG designers can stress, characterize and compliance test serial data receivers and components. SerialXpress is available for a free trial on www.tektronix.com.

MATLAB® Development Environment

This software toolset from Mathworks is an industry-leading choice among engineers who need comprehensive math, analysis, and DSP functions to support their design work. MATLAB is a high-level mathematics-based language and

interactive environment designed to carry out compute-intensive tasks faster than general-purpose programming languages such as C, C++, and others. MATLAB includes tools that can calculate and preview complex waveforms that carry, for example, modulation and jitter or distortion. Figure 3 depicts an example of such a waveform.

In many enterprises, MATLAB (or one of several similar commercially available alternatives) resides on engineering servers where it can be accessed by one or many users. In this arrangement, the integrated XP computer that hosts the AWG5000 Series or AWG7000 Series instrument becomes a node on the network, with the ability to run MATLAB on the server and save files there or locally within the instrument platform.

MATLAB's Instrument Control Toolbox facilitates communication with instruments (including signal generators) directly from MATLAB. The toolbox supports GPIB, VISA, TCP/IP, and UDP communication protocols, among others.

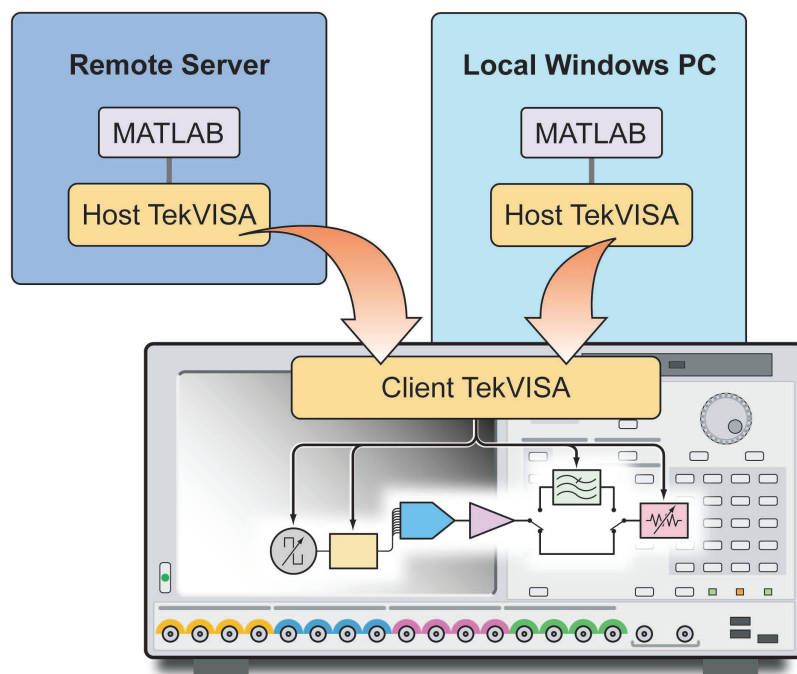


Figure 4. The TekVISA interface brings MATLAB code from the programming/computing platform to the AWG's functional elements.

Importantly, the toolbox includes features that make it possible to design graphical user interfaces (GUIs). A GUI enables both veteran and novice users to efficiently create files that can configure instrument settings, write data to instruments, etc.

A great deal of additional information about MATLAB can be found on the Mathworks website. While MATLAB is not the only solution of its kind, it is in very widespread use for engineering applications and so will be used as the reference toolset for the balance of this application note.

TekVISA Connectivity Software

MATLAB's high-level constructs simplify programming and encourage good comprehensible code development. Waveforms created with external languages such as MATLAB are ported into the signal generator's waveform memory. Technically this can be done with the AWG's integrated import features, which support ASCII comma delimited real integer formats, dedicated .wfm a, .pat files, or custom-written

drivers. But there is a ready-made and proven conduit that enables direct transfer of waveforms from MATLAB to the AWG's memory, shown in Figure 4.

This interface is known as TekVISA, the Tektronix implementation of the industry-standard VISA (Virtual Instrument Software Architecture) communication protocol. TekVISA is specifically designed to exchange external data with instruments.

With the aid of TekVISA, MATLAB can communicate with the AWG5000 Series and AWG7000 Series instruments via TCP/IP (Ethernet LAN), GPIB, or virtual GPIB. The latter protocol is designed to be used when the AWG's local PC is the source of the MATLAB code and waveform data.

In operation, TekVISA consists of two complementary elements, one residing in the computing platform (whether local or remote) and the other within the instrument hardware. All current Tektronix AWG models include the TekVISA software interface, both client and host, as standard equipment.

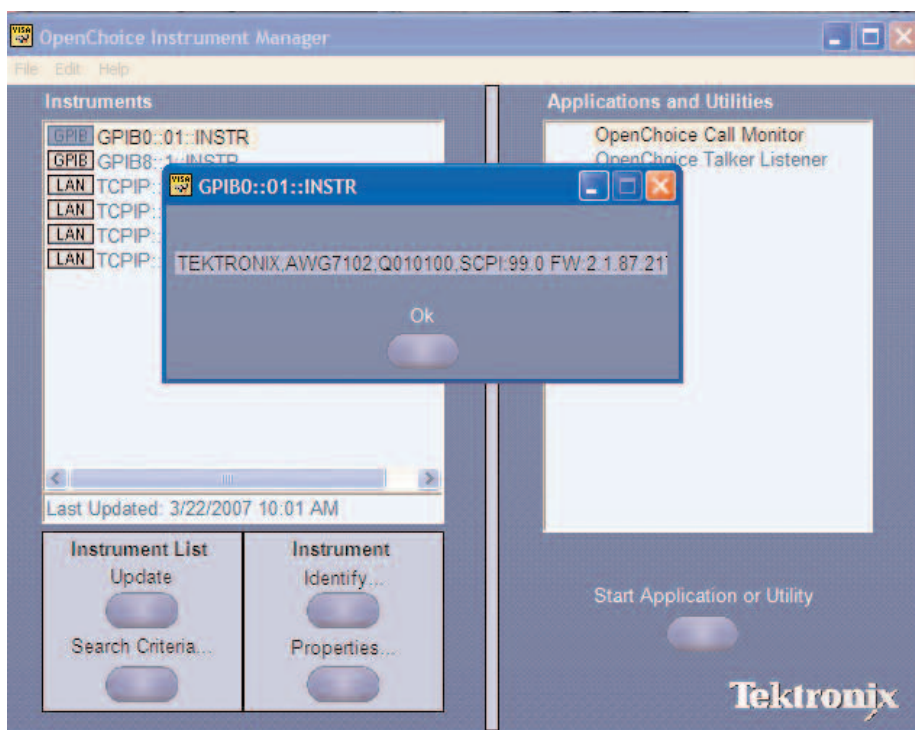


Figure 5. The VISA Instrument Manager summarizes the available connected instruments.

To communicate with a TEKVisa-enabled AWG as a listener, the AWG must first have its VXI-11 server started. This is available on the windows desktop toolbar of the AWG5000 and AWG7000 instruments. Once the server is started, the instrument connection can be verified by running the Visa Instrument Manager from the remote server PC. When the instrument search is completed, the instruments available for communication appear on the list in the Instrument Manager screen (Figure 5).

Once this communication has been established, using the TekVISA interface and Tektronix AWG MEX MATLAB drivers (available from www.tektronix.com) are straightforward and easy to use, with a small, optimized instruction set. There are two commands for session management and seven commands for instrument control and data transfer as summarized in Table 1.

Note: MEXcompiles and links source files using C or C++ into a shared library, called a MEX-file, that is executable from within MATLAB. Tektronix provides these AWG MEX files to simplify waveform transfer from within the MATLAB environment.

The NewSession command essentially opens up communication with the AWG, connecting the computing platform (external or local) to the instrument identified by the address string. The “Initialize and connect” segment in the example file below depicts this procedure. The CloseSession command ends the communication with the AWG.

Session Management Control and Data Transfer	
NewSession	Read
CloseSession	Write
	LoadWfm
	TransferWfm
	TransferWfmReal
	TransferWfmInteger
	Query

Table 1. MEX MATLAB command set for arbitrary waveform generators.

The NewSession and CloseSession commands require device addresses to be on file in the TekVISA Instrument Manager. The procedure for entering these addresses is beyond the scope of this discussion, but available TekVISA documentation explains the requirements and conventions.

The TransferWfmInteger and TransferWfmReal commands are intended for use with the AWG5000 Series and AWG7000 Series instruments. These instructions send waveform data in integer or real formats to the connected device’s waveform list pending a subsequent Load command to move the data into the waveform memory. Note, the simple TransferWfm command is not recommended for the AWG5000 and AWG7000 Series instruments.

Creating a Waveform with MATLAB

Earlier we saw the code involved in generating a basic function waveform with ArbExpress: a two-line “program” that simply stated the range and called the function. With MATLAB and other tools of its kind, much more complex waveforms can be designed. Realistically, waveforms burdened with modulation, distortion, transients and jitter are what stress testing is all about. While programming such signals will likely require more than two lines of code, the process need not be intimidating.

To prove this point, let’s look at a sample file that creates a waveform, installs it in an AWG7000 Series instrument, and commands the AWG to run. Comment lines are highlighted in green for emphasis. The “%” sign that precedes these same lines is a MATLAB convention.

Consult MATLAB documentation for specific commands and their qualifiers. Now to proceed with the program step-by-step:

1. Specify the waveform characteristics

%This sample shows how to connect to AWG5000 and AWG7000 instruments creating a linear sweep waveform

```
%////////////////////////////////////
```

```
%// Parameters you can change
```

```
%////////////////////////////////////
```

```
clock = 10e+9; %// AWG clock
```

```
fc = 1.25e+9; %// Center frequency
```

```
pd = 4e-6; %// sweep period
```

```
fs = -4.5e6; %// starting frequency
```

```
fe = 4.5e+6; %// ending frequency
```

```
len = pd * clock; %Waveform length
```

```
t = (0:len-1)/clock; %Sample interval
```

The action items in this segment set the instrument clock frequency and the center frequency as well as the sweep frequencies. The desired waveform is an RF chirp that sweeps about a center frequency of 1.25 GHz. Values are expressed in conventional engineering notation, where “e+9,” for example, signifies a multiplier of 109. Thus the center frequency *fc* will be 1.25 x 109, or 1.25 GHz. The length of the sweep will be the product of the period (4e-6) and the clock (10e+9).

2. Generate the waveforms

```
%Create a sample pulse waveform with I & Q
```

```
i = cos(2*pi*fs*t + 2*pi*(fe-fs)*((t.^2)/2)/pd);
```

```
q = sin(2*pi*fs*t + 2*pi*(fe-fs)*((t.^2)/2)/pd);
```

```
%Create IF waveform
```

```
Waveform = i .* cos(2*pi*fc*t) - q .* sin(2*pi*fc*t);
```

```
%Plot Waveform
```

```
plot(Waveform)
```

This is the math-intensive step. The terms set up in Step 1 are entered into equations that define the in-phase (I) component using a cosine equation and the quadrature (Q) component using a sine equation. Next, an array named “Waveform” is calculated based on the combination of IQ terms. And lastly, the array known as Waveform is plotted. This is the actual IF waveform.

3. Initialize and connect

```
%Initialize the driver and connect to the instrument using Visa calls
```

```
%Use this for LAN, with your IP address
```

```
%[sess,msg]=NewSession('TCPIP::xxx.xxx.xxx.xxx::INSTR','LAN');
```

```
%Use this for GPIB, with your GPIB address
```

```
[sess,msg]=NewSession('GPIB0::01::INSTR','GPIB');
```

```
%Reset the instrument.
```

```
Write(sess,'*RST');
```

In this step, we make the first contact with the instrument. The TekVISA interface is compatible with both TCP/IP and GPIB protocols. This example segment opens a new communication session with an AWG at a particular GPIB address. Note that an equivalent line for an instrument on a LAN has been included, although disabled by a comment symbol. The “Write” command in the last line resets the instrument. Unlike segments 1 and 2 above, which are application-specific, this is a good general-purpose code segment suitable for almost every MATLAB session with the AWG.

4. Create synchronization markers

```
%Create marker1 data with single pulse at beginning.
```

```
Marker1 = int32([1 zeros(1 , len-1) ]);
```

```
%Create marker2 data with alternating 1s and zeros.
```

```
Marker2 = int32(mod((1:len) * 1, 2));
```

Both the AWG5000 Series and the AWG7000 Series provide digital marker outputs for each main analog output. Often it is useful to have a discrete digital reference pulse to synchronize external instruments such as oscilloscopes or counters. The marker outputs are ideal for this purpose. This code segment creates a single pulse at the start of the waveform in Marker1, while Marker2 is set up as a clock consisting of alternating ones and zeros (101010101...) through the whole waveform record.

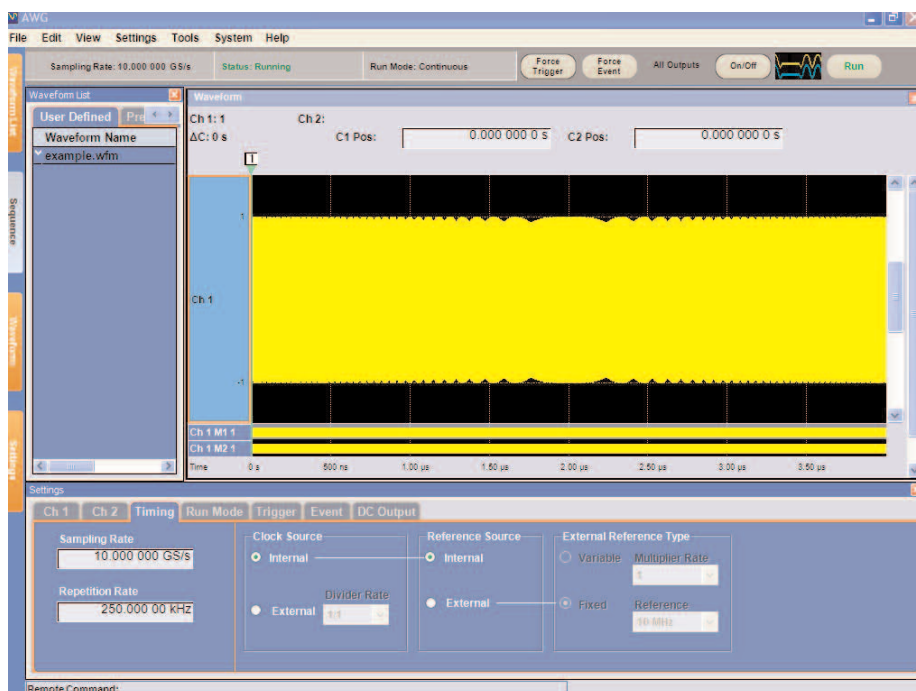


Figure 6. The chirp waveform as viewed on the AWG7000 Series screen.

5. Send the waveform to the instrument

```
TransferWfmReal(sess, 'example.wfm', Waveform, Marker1,
Marker2, len);
```

```
%Load the waveform to ch1
```

```
LoadWfm(sess, 'example.wfm', '1');
```

This step is actually two steps in one, as implied by the comment lines. A “TransferWfmReal” in this syntax causes the waveform data to be transmitted with the filename “example.wfm” to the AWG’s waveform list located in the AWG system setup. The “Load” command shifts the same data into the actual waveform memory of the Channel that will be used. (See Figure 1). The execution of this pair of commands readies the AWG to generate the waveform that was created in MATLAB. However, the actual electrical properties of the output signal are not yet defined.

6. Set the channel’s output parameters

```
%Set channel properties
```

```
Write(sess, 'SOUR1:VOLT:AMPL 0.2');
```

```
Write(sess, 'SOUR1:MARK1:VOLT:HIGH 0.5'); %Marker1
```

```
Write(sess, 'SOUR1:MARK2:VOLT:LOW 0.1'); %Marker2
```

```
%Write(sess, 'SOUR1:FREQ 10e+9');
```

```
Write(sess, ['SOUR1:FREQ ' num2str(clock)]);
```

This group of instructions sets the output parameters and is almost self-explanatory. The first Write command sets the amplitude of the output signal. The final Write command invokes the instrument’s sample clock rate, which was defined as 10 GS/s in the first segment of this program.

7. Enable the channel and RUN

```
%Switch the channel on
```

```
Write(sess, 'OUTPut1 1');
```

```
%Run
```

```
Write(sess, 'AWGC:RUN')
```

Here the comments tell the whole story. The AWG channel will not deliver a signal until the channel output is enabled; (switched on). This function can be thought of as a simple switch between the final attenuator stage and the output connector (actually a differential pair treated as one entity). The OUTPut1 command closes this switch. Each AWG channel (up to four) must be enabled separately.

Now the instrument can be instructed to RUN, with the result that the MATLAB-created waveform will appear at the output connector. Note that a CloseSession command is not given; this is not required in the circumstances shown.

Figure 6 is a screen image taken from an AWG7000 Series arbitrary waveform generator running the waveform created in the preceding steps.

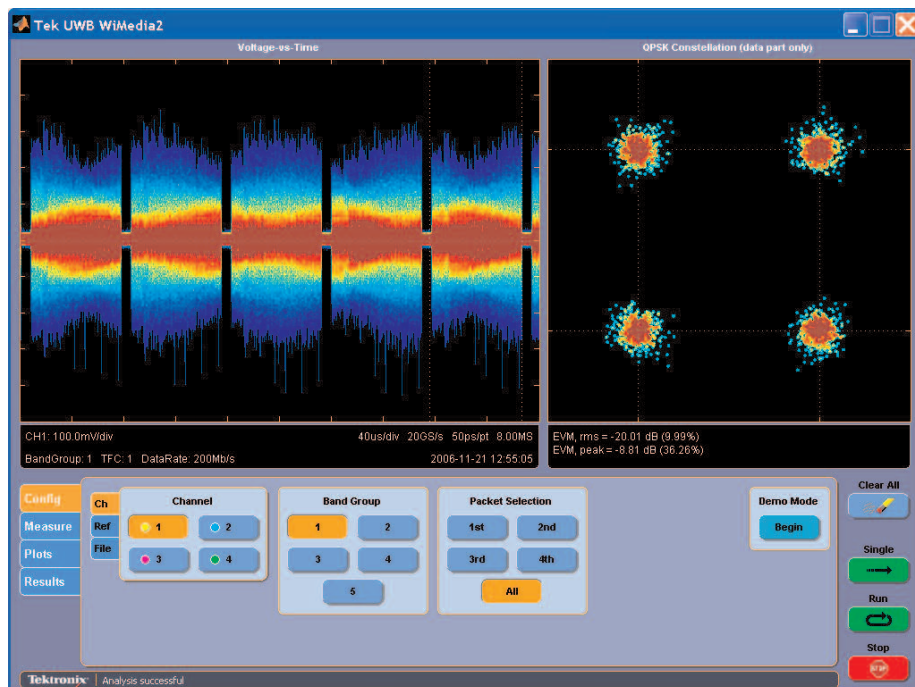


Figure 7. A GUI developed in MATLAB.

Building A Graphical User Interface For MATLAB Programming

The foregoing section explains how text-based MATLAB commands can create a waveform and control the AWG. For one-time measurements or experiments, this method is sufficient. But for day-in, day-out testing that relies on varying measurement parameters within a fixed procedure, a graphical user interface is much faster and easier to use. A well-designed GUI presents only the controls and ranges needed to get the job done. It can be narrowly tailored for a specific application, or tabs and buttons can be set up to allow the user to select from a group of related applications.

MATLAB's TMScope™ application, part of the MATLAB Instrument Control Toolbox, is a simple graphical interface with a good general-purpose feature set: search for hardware, connect, configure, write, and read.

Figure 7 depicts an application-specific GUI for the Tektronix UWB WiMedia measurement tool for oscilloscopes. This GUI was developed entirely within MATLAB and is presented here for the purpose of illustrating what is possible with a GUI. The underlying instrument commands are similar to those in the foregoing MATLAB procedure, but access to these functions is faster and more foolproof than hand-coding the instructions. Tabs help the user structure his or her tests, while buttons simplify functions like channel activation and single/continuous execution. Color highlights indicate the selected buttons, minimizing the potential for misinterpretation.

From RF to Serial and Beyond

By now it should be clear that the AWG architecture and direct synthesis technology offer the versatility to support almost any imaginable signal generation need. New applications for the AWG toolset are emerging constantly.

Compliance measurements for serial bus standards such as PCI Express and Serial ATA, particularly with respect to serial receiver tests, are a discipline that will be transformed by direct synthesis. Consider the jitter tolerance tests that make up a large part of the compliance requirements for almost every serial standard. Until the advent of new-generation AWGs, jitter tests required a test system and fixturing arrangement consisting of: a data generator to create the pattern; a noise generator to be used as a random modulation source; a sine or function generator to be used as a deterministic source; a mixer; a PC running a special BIST (built-in self test) setup routine; a power combiner; and a frame error counter to monitor the output of the device under test. This complex configuration is designed solely to accomplish an uninterrupted and variable stream of serial pattern data with controllable jitter.

Now the AWG can deliver such a stream directly from its waveform memory. In this application, as elsewhere, a software-driven approach is proving much more dynamic than its hardware-based predecessors.

The desired jitter is imposed at the time the pattern waveform is created with MATLAB or similar tools. At the same time, the data can be further modified with spread spectrum clock characteristics, pre- or de-emphasis, and other standardized serial data attributes – all without additional hardware. The AWG solution doesn't need to rely on any external modulators or modifiers. It adds up to a simpler test setup that is much more controllable and easy to use.

A host of other applications will benefit from direct synthesis: radar; software-defined radio; wireless Ultra-Wideband RF, and more. The AWG is the only platform that's equally at home on both the serial designer's desk and the RF component evaluation bench.

Conclusion

Direct synthesis in the form of the modern arbitrary waveform generator is an enabling technology for efficient new techniques used in wireless communications measurements, serial compliance tests, and more. Comprehensive math and analysis software packages such as Mathworks' MATLAB bring a new level of efficiency to the creation of waveforms and functional programs for today's AWGs. With the help of such a math toolset, it is possible to create almost any cyclical waveform shape and then purposely distort it to simulate real-world conditions. The pairing of a powerful AWG platform such as the Tektronix AWG5000 or AWG7000 Series with MATLAB is a combination to speed the development of complex stimulus waveforms.

Contact Tektronix:

ASEAN / Australasia (65) 6356 3900
Austria +41 52 675 3777
Balkan, Israel, South Africa and other ISE Countries +41 52 675 3777
Belgium 07 81 60166
Brazil & South America (11) 40669400
Canada 1 (800) 661-5625
Central East Europe, Ukraine and the Baltics +41 52 675 3777
Central Europe & Greece +41 52 675 3777
Denmark +45 80 88 1401
Finland +41 52 675 3777
France +33 (0) 1 69 86 81 81
Germany +49 (221) 94 77 400
Hong Kong (852) 2585-6688
India (91) 80-22275577
Italy +39 (02) 25086 1
Japan 81 (3) 6714-3010
Luxembourg +44 (0) 1344 392400
Mexico, Central America & Caribbean 52 (55) 5424700
Middle East, Asia and North Africa +41 52 675 3777
The Netherlands 090 02 021797
Norway 800 16098
People's Republic of China 86 (10) 6235 1230
Poland +41 52 675 3777
Portugal 80 08 12370
Republic of Korea 82 (2) 6917-5000
Russia & CIS +7 (495) 7484900
South Africa +27 11 206 8360
Spain (+34) 901 988 054
Sweden 020 08 80371
Switzerland +41 52 675 3777
Taiwan 886 (2) 2722-9622
United Kingdom & Eire +44 (0) 1344 392400
USA 1 (800) 426-2200
For other areas contact Tektronix, Inc. at: 1 (503) 627-7111
Updated 17 October 2007

For Further Information

Tektronix maintains a comprehensive, constantly expanding collection of application notes, technical briefs and other resources to help engineers working on the cutting edge of technology. Please visit www.tektronix.com



Copyright © 2008, Tektronix. All rights reserved. Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specification and price change privileges reserved. TEKTRONIX and TEK are registered trademarks of Tektronix, Inc. All other trade names referenced are the service marks, trademarks or registered trademarks of their respective companies.
1/08 FL/WOW 76W-20718-1

Tektronix®
Enabling Innovation

