IQtools – a flexible waveform generation tool  
for Keysight Arbitrary Waveform Generators

June 16, 2021

# Overview

IQtools is a collection of MATLAB example applications for creating I/Q, IF/RF, serial data, multi-tone, radar pulse and many other types of waveforms on the Agilent/Keysight M8199A, M8194A, M8196A, M8195A, M8190A, 81180A, M933xA, 81150A and 81160A arbitrary waveform generators (AWG) as well as several Keysight Signal Generators. MATLAB is a software environment and programming language that is very well suited for calculating arbitrary waveforms, measurement and analysis routines, and instrument applications. MATLAB is available directly from Keysight as an instrument option (option [N6171A](http://www.agilent.com/find/n6171a)-M03) with many instruments, including the M8190A and M933xA, and also from MathWorks.

# Purpose

IQtools was created to demonstrate the waveform generation capabilities of Keysight arbitrary waveform generators, along with the value of using these instruments together with MATLAB software. IQtools is provided at no charge, it is available as source code, except for a few routines that are delivered in encrypted form (\*.p files) because they contain specific Keysight IP. Alternatively, IQtools is also offered as a compiled executable, that does not require a MATLAB license.

The source code version of IQtools requires MATLAB software to operate. You can request a free trial of MATLAB software from your Keysight representative or directly at [www.mathworks.com/keysight/trial](http://www.mathworks.com/keysight/trial). The source code version will allow you to extend or customize the functionality. The executable version of course does not allow that.

# Requirements

These example applications require the MATLAB N6171A-M03 package to operate (or the equivalent of these options which include MATLAB, Instrument Control Toolbox, Signal Processing Toolbox, Communications System Toolbox, and RF toolbox). These example applications were tested with **MATLAB Version 9.7.0 (R2019b) and Version 9.8.0 (R2020a)** and might not work on older versions of MATLAB.

Some of the analysis functionality works in conjunction with the **PathWave VSA software (89600)**. Version 21.0 or higher is required. In this document, the PathWave 89600 VSA software will simply be referred to as “VSA” software.

Installation

The MATLAB examples files are distributed in the form of a compressed .ZIP archive. Please choose an empty directory and unzip the files into this directory. Make sure that the directory structure of the unzipped files remains intact. It is recommended to use a subdirectory of your MATLAB working directory (e.g. c:\users\<username>\Documents\MATLAB\iqtools).

Set the MATLAB search path to include the directory that you have chosen. You might want to add the following line to your “startup.m” script: addpath(‘c:\users\<username>\Documents\MATLAB\iqtools’);

Once you have done that, you can always start IQTools by simply typing iqtools on the command line.

# Creating and downloading waveforms

Currently, the following types of waveform types can be created with these example applications:

1. **Multi-tone** (including single and dual-tone) signals with the ability to specify sampling rate, number of tones, tone frequency range and phase relationship as well as one or more notches with adjustable depth. The amplitude flatness can be automatically corrected using a spectrum analyzer. This tool can also be used to generate **band-limited Gaussian noise** (with notches).
2. **Multi-Carrier Modulated Signals** (including single carrier) with the ability to specify the sampling rate, number of symbols, oversampling factor, modulation type, pulse-shaping filter, number of carriers and carrier spacing. Amplitude and phase correction can be performed from within this tool in conjunction with the VSA software. This will significantly improve the EVM performance.
3. **Pulsed frequency sweeps** with adjustable rise/fall time PW, PRI and pulse shape, frequency span and offset
4. **Frequency switching** between two or more frequencies with adjustable tone duration and phase continuous switching
5. **OFDM** signal generation with custom modulation schemes
6. **Serial data** with adjustable data rate, various formats, transition times, pulse shape filter, sinusoidal and (pseudo-)random jitter, sinusoidal or pseudo-random noise.
7. **Function Generator**, which can generate various types of pulse shapes
8. **Load a waveform from a file.** With this function a MATLAB (.MAT), Binary (.BIN) or ASCII (.CSV) file containing sample and marker values can be downloaded to the AWG.
9. Setting up **Sequences** of waveforms
10. A number of demo examples for specific applications (e.g. CATV, Radar)

The waveforms can be created, displayed and downloaded to the AWG either as command typed in on the **command line** prompt or using a **graphical user interface** where parameters can be entered. To launch the graphical user interface, type

**iqtools**

on the MATLAB command line (or select them in the MATLAB file browser and press F9). Alternatively, the individual tools can be launched directly using the commands iqconfig, iqtone, iqmod, iqpulse, iqfsk, iqpulsegen, iqloadfile and iserial.

Each of these graphical user interfaces has a number of parameter fields as well as a "**Visualize in MATLAB**" and a "**Download**" button. Pressing the "**Visualize in MATLAB**" button shows the generated waveform in MATLAB plots (there is no hardware required). Some of the utilities also have an additional button called “**Visualize in VSA**”. This button starts the VSA software, sets it up with the appropriate parameters to show the calculated waveform. In order to use this function, no hardware connection is required.

The "**Download**" button loads the waveform into the AWG. In order for the download functionality to work properly, the connection to the AWG and optionally a few other parameters have to be configured. This can be done using the “Configuration” button in iqmain (or alternatively launching iqconfig).

Using the “**Ok**” button in Configuration Window, the parameters are stored in a file called “arbConfig.mat” in the current folder and they **affect all following download operations**. In addition to the AWG configuration, Configuration can set up the communication parameters for a spectrum analyzer, real-time and sampling oscilloscope and a power meter. The connection to these additional instruments is required for frequency/phase response correction routines as described in another section of this document.

# Entering numerical values

All of the numeric input fields in all of the tools will accept number is fixed point or engineering notation. Units are always seconds or Hertz and unit indicators are NOT allowed. E.g.:

* 12e9 - to indicate 12 GHz
* 6.2e-6 - to indicate 6.2 Microseconds
* 3 MHz - NOT allowed (must be specified as 3e6)
* 1ps - NOT allowed (must be specified as 1e-12)

Some fields which are marked with (\*) in the graphical user interface also accept lists of values (i.e. MATLAB vectors). Lists can simply be specified by multiple values separated by spaces or commas. E.g.:

* 100e6, 200e6, 500e6 - to indicate 100, 200 and 500 MHz
* 100e6 \* [1 2 5] - same as above
* 1e-6 3.5e-6 1e-4 - to indicate 1 us, 3.5 us and 100 us

In addition to numbers and lists of numbers, the fields also accept formulas using MATLAB syntax. E.g.:

* 1e9-5e3 - to indicate 5 kHz less than 1 GHz (or 999.5 MHz)
* 100e6:10e6:200e6 - a list of values from 100 MHz to 200 MHz in steps of 10 MHz
* linspace(100e6,200e6,11) - different way of describing the same as above: 11 values  
   evenly spaced between 100 and 200 MHz
* 100e6 \* rand(1,10) - 10 random frequencies between 0 and 100 MHz  
   (“rand” is a built-in MATLAB function)

In addition to that, all of the input fields will evaluate variable names that are defined in the MATLAB workspace and formulas using MATLAB workspace variables. E.g.

* Fs - can be used e.g. in the sample rate field if the variable “Fs”  
   is defined in the MATLAB workspace
* 1e6\*f - can be used to specify a list of frequencies assuming that “f”  
   contains the list of frequencies in MHz
* fc-5e3 - center frequency minus 5 kHz – assuming that the variable “fc”  
   is defined in the MATLAB workspace

# Programmatic interface

Instead of using the graphical user interfaces of the scripts, they can also be called from within a MATLAB script to generate, display and download waveforms. This can be useful to generate more complex waveforms that e.g. consist of multiple signals added together. To generate a starting point for a MATLAB script that calculates and downloads a waveform, you can click on “File” 🡪 “Generate MATLAB code” on any of the IQTools utilities. An example might look like this:

% automatically generated code from IQTools

%

fs = 1.2e+10;

tone = [1e+09 1.5e+09 2e+09];

magnitude = [0 0 0];

iqdata = iqtone('sampleRate', 1.2e+10, 'numSamples', 0, ...

'tone', tone, 'phase', 'Random', ...

'magnitude', magnitude, 'correction', 0);

iqdownload(iqdata, fs, 'channelMapping', '[1 0; 0 1]', ...

'segmentNumber', 1);

The input parameters of iqtone, iqmod, iqpulse, iqfsk, iqpulsegen and iserial are specified as parameter/value pairs. All parameters have default values if they are not specified. Please look at the headers of the individual script for available arguments and ranges.

The scripts that should be used for your own programs are:

iqtone - single- and multitone

iqnoise - generate noise signal

iqmod - digital Modulations

iqpulse - pulsed RF signals incl. modulation on pulse (chirps, FMCW, Barker, etc.)

iqfsk - frequency hopping signals

iqpulsegen - function generator (pulses, ramps, etc.)

iserial - serial data signals (NRZ, PAMx)

# Waveform display

The **iqplot** scripts can be used to display waveforms in both time and frequency domains. You can use iqplot to display real and I/Q data that has been generated by any of the waveform generation functions listed above or by your own MATLAB functions. Iqplot is called as follows:

iqplot(data, fs [, options])

The first parameter (data) is the desired signal represented as a vector of real or complex values in the range [-1…+1] and the second parameter (fs) is the sampling rate in Hz.

Optionally, you can specify further parameters that will influence how signals are being displayed:

* 'figure', N – the figure number to be used for plots (default is 1)
* 'no\_timedomain' – the time domain plot will not be shown
* 'nospectrum' – the spectrum plot will not be shown
* 'smallspectrum' – only the “interesting” part of the spectrum will be shown (i.e. the frequency range where the signal is above -90 dBm)
* 'spectrogram' – shows a spectrogram of the signal
* 'constellation' – shows the waveform as a constellation diagram
* 'no\_CCDF' – no CCDF plot will be shown

# Waveform download

The **iqdownload** script is used to download waveforms into the AWG and start waveform generation. You can call iqdownload with a vector or array of real or complex data that has been generated by one of the waveform generation functions listed above or your own MATLAB functions. Iqdownload is called is follows:

iqdownload(data, fs [, option, value])

The first parameter (data) is the desired signal represented as a vector or array of real or complex values in the range [-1…+1] and the second parameter (fs) is the sampling rate in Hz.

By default, the real part of the data vector will be downloaded to channel 1 of the AWG and the imaginary part will be downloaded to channel 2. If you want the mapping to be different, please refer to the “channelMapping” parameter, that is described below.

Please DO NOT call the download routines for individual AWGs directly, since the interface might change over time. (e.g. iqdownload\_M8190A, iqdownload\_81180A, etc.)

**iqdownload** accepts additional parameters as name/value pairs. These are explained in the header section of iqdownload.m. Some of them are explained in the following sections:

### ‘channelMapping’

Using the ‘channelMapping’ parameter with iqdownload allows you to specify to which channel(s) the real and imaginary parts of a given column of your data will be downloaded. The channelmapping parameter must be a 2-dimensional array with 2\*N columns (representing I and Q) of the N columns in your data array and M rows (representing the M channels of the AWG or synchronized AWG system). The individual values of the array must be either 0 or 1, to indicate whether the corresponding signal will be downloaded to a channel. E.g. a channelMapping of [1 0; 0 0; 0 1; 1 0] indicates that the real part of the waveform will be loaded into channels 1 and 4, while the imaginary part will be loaded into channel 3. Note, that for DUC mode (if supported by the AWG), both real and imaginary part must be downloaded to a channel. A valid ‘channelMapping’ Parameter would for example be: [1 1; 0 0], which means real and imaginary part are loaded into channel 1 of the AWG.

### ‘marker’

The ‘marker’ parameter allows you to specify the signal that will be generated out of the sample marker and sync marker signals. In general, the ‘marker’ parameter must be a vector with the same number of elements as there are samples in the waveform. The values of the marker array are treated as integers where the lower four bits represent the four markers (sample1, sync1, sample2, sync2). A zero means that all markers are 0, a value of 15 (=binary 1111) means that all four markers are on. A value of 5 (=binary 0101) means that the two sample markers are on at the corresponding sample time). Note that the markers in the M8190A cannot be arbitrarily turned on and off at any sample location, but will have a certain minimum pulse width (See datasheet for more details). An example MATLAB code that uses markers is shown below:

%

% automatically generated code from IQTools

%

fs = 1.2e+10;

iqdata = iqmod('sampleRate', fs, 'numSymbols', 3000, ...

'modType', 'QAM16', 'oversampling', 12, ...

'filterType', 'Square Root Raised Cosine', 'filterNsym', 80, ...

'filterBeta', 0.35, 'carrierOffset', 1e+09, ...

'magnitude', [0], 'quadErr', 0, 'correction', 0);

% generate a marker vector with the same number of elements as

% the analog waveform with a pulse on the sample markers every

% 192 samples and a sync marker pulse at the beginning of the waveform

n = length(iqdata);

pw = 192;

cnt = floor(n / pw);

marker = repmat([5 \* ones(1, pw/2) zeros(1, pw/2)], 1, cnt);

marker(1) = 10;

iqdownload(iqdata, fs, 'channelMapping', '[1 0; 0 1]', ...

'segmentNumber', 1, 'marker', marker);

# Generating RF / IF waveforms

All of the IQtools waveform generation utilities (except iserial) will generate I/Q baseband or direct IF/RF waveforms. In case of baseband I/Q waveforms, the “I” signal and “Q” signals are intended to be connected to an I/Q modulator, such as the wideband I/Q inputs of a Vector PSG (E8267D) or an optical modulator. They can also be connected directly to an oscilloscope for analysis of the I/Q waveform using the VSA software.

However, the tools can also be used to generate an IF/RF signal directly. In order to generate an RF signal, make sure that:

For **Multi-Tone**, both start and stop frequency are positive

For **Digital Modulation signals**, the carrier offset is positive and larger than ½ of the bandwidth of the modulated signal

For **Radar chirps**, frequency offset is positive and larger than ½ the frequency span.

For **Frequency Switching**, all frequencies in the list must be positive

The simulated spectrum that is shown when the "Display" button is pressed should only show positive frequency components.

**iserial** is an exception and always generates a real-valued signal.

# Using the M8190A in Digital Up-conversion Mode

The IQTools utilities (except iserial) can also be used to generate baseband signals for the M8190A operating in “Digital Up-conversion” (DUC) mode. In order to use DUC mode, use the configuration window and select one of the DUC mode (x3, x12, x24 or x48) in the “Instrument Model” popup menu. Optionally, you can also select the carrier frequency in the same window. After you click OK, open any of the IQTools utilities and define the baseband waveform.

Notice that the “Download” popup menu now shows “RF to channel 1” and “RF to channel 2” as the possible selections.

# Working with two M8190A modules simultaneously

IQTools supports a 4-channel setup that consists of two M8190A modules with an optional M8192A synchronization module.

### Working WITHOUT the M8192A SYNC module

It is possible to synchronize two M8190A modules with the help of an oscilloscope down to approx. 1 ps skew between each pair of channels. This is described in the chapter “M8190A specific functions: 4-channel synchronization” below.

### Working WITH the M8192A SYNC module

When working with the M8192A SYNC module, you have to make sure that:

* you have a M8190A firmware instance running for each of the two modules
* you have an M8192A firmware/SFP running
* you have both M8190A VISA addresses as well as the M8192A VISA address configured in the IQTools configuration window.

With these prerequisites, the “Download To” button in each of the utilities allows you to select to which of the four channels your real and imaginary part of the waveform will be downloaded. It is possible to load the same component to multiple channels.

# Using the Multi-tone utility

The multi-tone utility can be used to generate single tone, two tone, multi tone signal with equidistant or non-equidistant spacing as well as band-limited noise. “Notches” with variable width and depth can be included with multi-tone or noise signals.

The algorithm in the multi-tone utility always calculates complex-valued signals (I & Q). If you need a single (non-I/Q) signal, simply choose your tone frequencies to be all positive (or all negative). The desired signal will show up on both channels of the AWG. E.g. if you generate a 100-tone signal from 20 MHz to 2 GHz, you will see that signal on channel 1 of the AWG as well as channel 2.

If you use an AWG in conjunction with an I/Q modulator (e.g. M8190A connected to a Vector PSG), your range of tones can be from – FS/2 to + FS/2. The I/Q modulator will move the frequency range to its carrier frequency.

In the following sections, some special cases are described that can be covered by the multi-tone utility.

### Noise

To generate a noise waveform in a certain frequency band, set the number of tones to zero. Choose a large number of samples (1000000 is a good starting point). The larger your number of samples, the more “random” is your noise. On the other hand, a larger number of samples takes longer to calculate and download the waveform. The “Start” and “Stop” frequency fields can be used to limit the bandwidth of the noise. The “Notch” feature can be used to generate one or more spectral “gaps” in the noise signal with adjustable width and depth.

### Equidistant and Non-equidistant tones

If you just select the number of tones and a start and stop frequency, the tones will be distributed equidistantly between the start and stop frequency. The algorithm will put a tone on the start and stop frequency itself. You should take this into account when selecting the number of tones. E.g. if you want equidistant tones from 1 GHz to 2 GHz with 10 MHz spacing, you should choose 101 tones (not 100).   
For non-equidistant tones, set the number of tones to “1” and enter the list of tone frequencies that you would like to generate in the “Stop frequency” field. You can use MATLAB expressions such as:  
 100e6 \* [-3 -1 4 5 7.5 9]  
This will generate tones at -300, -100, 400, 500, 750 and 900 MHz.

### Tones with different amplitudes

You can use the “Notch” feature to generate tones at different relative amplitudes. You just have to define a “Notch” for each tone (or range of tones) that you want at a different level.

Example: You set the “number of tones” to 1 and the Stop frequency to:  
 100e6 \* [-3 -1 4 5 7.5 9]  
to generate tones at-300, -100, 400, 500, 750 and 900 MHz. Without anything else, they will all have the same relative amplitude. Now check the “Notch” checkbox, and enter   
 100e6 \* [-3 -1 4 5 7.5 9]  
in the “Notch frequency” field. Change the “Notch span” to 1e6 (can be any number > 0 because you just want to hit that specific tone) and enter  
 -5 +5 -10 +10 0 0  
in the “Notch depth” field. This will change the relative amplitude of the tones.

This works both with individual tones as well as with tone ranges. To see an example with tone ranges at different levels, click on “Preset” 🡪 “Multi-tone with multiple Notches” in the Multi-tone utility.

Click “Visualize in MATLAB” or “Download” to see the result in MATLAB resp. in hardware.

# Multi-tone amplitude correction

In order to improve amplitude flatness for multi-tone signals, the “iqtone” utility comes with an amplitude correction function. Amplitude correction is performed by measuring a multi-tone signal with a spectrum analyzer and feeding back the correction curve into the waveform calculation routine. This amplitude correction can later on also be used by the other waveform creation utilities to improve the flatness. The amplitude flatness correction can be used to compensate non-flatness on the output of the AWG or to perform correction after I/Q up-conversion.

To perform an amplitude correction:

1. Use the “Configure Instrument Connection” tool to set up the communication parameters for the spectrum analyzer (visa is recommended). Choose whether you want to use the **Zero Span** or **List sweep** mode. The LIST SWEEP commands are only available on MXA (N9020A) and PXA (N9030A). The LIST SWEEP measurement is a little faster, but less accurate. If in doubt, use the “Zero Span” mode. Don’t forget to click “Ok”.
2. Use “Multi-tone” to generate a multi-tone signal that spans the entire frequency range that need to be corrected later on. A good starting point is the Preset “101 tones, +/- 1 GHz, asymm.” (can be found in the menu bar) if you are generating I/Q data. For a direct IF/RF signal (not using I/Q), choose a range of frequencies that are all positive. A larger number of tones will improve the accuracy, but increase the measurement time. Around 100 – 200 tones produces reasonable results. For demo purposes, 30 or 40 tones are good enough. To get optimal results, the “Phase” should be set to “Random” and when using I/Q up-conversion, the tones should be selected such that the images don’t fall on top of generated tones. (i.e. don’t use symmetrical intervals around 0 Hz)
3. You might want to click “Download” once to see how the un-corrected waveform looks like. The multi-tone calibration works either with direct AWG output or with I/Q output going into the wideband modulation inputs of a PSG and measuring the PSG output. (Make sure the “Apply Correction” checkbox is off).
4. Set the center frequency (“Fc – for calibration only”) in the Multi-Tone GUI to zero if you are analyzing the direct AWG output. If you have the I/Q signals connected to a PSG and you are correcting the flatness of the final RF output, enter the Carrier frequency that is set in the PSG. If you are up-converting using a mixer, enter the difference between the LO and IF frequency (LO – IF) if you are looking that the upper sideband of the mixer. For the lower sideband, enter -1 \* (LO + IF)
5. Make sure the “Apply correction” checkbox is OFF to start a new correction measurement.  
   Now press the “Calibrate” button. Depending on the selected measurement mode, the spectrum analyzer will perform a measurement and the correction curve is written to the file “ampCorr.mat” in the local directory. The measured frequency response is also displayed as a MATLAB plot. This plot is just for your information and can be closed if it no longer needed.
6. To further improve the calibration, you can run it again, taking the previous correction factors into account. Experience shows that after running the calibration about 3 times, no further improvement can be achieved. The red graph indicates the measured deviation for each tone from the average tone power. It should get close to the zero line as you perform repeated calibrations.
7. The saved correction factors can be used by the other utilities (iqmod, iqpulse, iqfsk) as well. Just turn on the “Apply correction” in the respective utility. Note: When performing the amplitude correction, make sure that the start and end frequencies span at least the same range of frequencies that you will use later on in other utilities.

# Using the “Radar pulses / Chirps” utility

This utility generates pulsed IQ or RF waveforms with standard or custom modulations on pulse as well as FMCW. The following modulation formats are supported:

* **None** – generates pulses with constant frequency (i.e. no modulation on pulse)
* **Increasing, Decreasing** – generates a linear FM chirps
* **V-shape, inverted V** – generates up/down resp. down/up FM chirps
* **FMCW** – generates gap-free FM chirps (see details below)
* **Barker-XX, Frank-XX** – generates pulses with various types of Barker and Frank codes
* **User defined** – generates pulses with custom FM and PM modulations (see details below)
* **Custom IQ** – generates pulses with custom IQ modulation (see details below)
* **Custom Phase Code** – generates pulses with custom phase code (see details below)

## Single and multiple different pulses

To define a single pulse, enter scalar values in each of the fields that are marked with “(\*)”. To generate multiple different pulses in sequence, enter a list of values separated by space of comma in those fields. Example: To generate a series of three pulses with different pulse widths, enter:  
 1e-6, 2e-6, 3e-6  
in the “Pulse width” field. It is possible to vary multiple parameters at the same time. If you also enter three different Frequency Offsets or Amplitude values, then each of the three pulses will be at a different center frequency, resp. have different amplitude.

## “FMCW” modulation

When FMCW is selected as a modulation scheme, the parameters “Rise time”, “Fall time”, “Pulse Width”, “Repeat Interval” and “Pulse shape” have a different meaning than for all the other modulation schemes. While these parameters usually affect the *envelope* of the pulse, in FMCW they affect the *frequency* of the pulse as follows:

* “Rise time” is the duration of the frequency up ramp
* “Fall time” is the duration of the frequency down ramp
* “Pulse width” is the duration for which the frequency is kept at the highest value
* “Repeat Interval” is the overall duration of a complete cycle. You can enter a value of “0” to indicate the shortest possible repeat interval (i.e. rise time + pulse width + fall time)
* “Pulse shape” indicates the kind of FM: “Trapezoidal” creates a linear FM, “Raised Cosine” generates a cosine-shaped FM. The other pulse shape options are invalid in FMCW mode.

## “User defined” modulation

This modulation type can be used to define non-linear FM and PM modulations. When selecting user-defined modulation on pulse, you can specify the desired frequency and phase modulation through a mathematical formula. The formula is entered as a MATLAB expression and can either use a built-in function or a user-defined function. The argument “x”, which is passed to the FM and PM function is a vector of numbers linearly increasing from 0 to 1. This vector represents the pulse width. The function must return a vector of the same length as “x”. In case of the FM formula, the return values should be in the range of [-1 … +1]. They are scaled to the frequency span and shifted by the frequency offset. The result is used as the frequency modulation of the pulse. The return values from the PM formula must be in degrees. They are used as the phase modulation of the pulse. If either FM or PM is not required, a value of “0” can be specified as a formula. Examples:

* if the FM formula is:  
   sin(2\*pi\*x)  
  the (non-linear) frequency modulation of the pulse will have the shape of a sine wave with a deviation that equals the specified frequency span and is centered at the specified frequency offset
* To define a more complex formula, you can enter  
   myfunc(x)  
  as the FM formula and create the MATLAB function myfunc.m e.g. as follows:  
   function y = myfunc(x)  
   y = (2\*x - 1) .^ 3;

## “Custom IQ” modulation

With Custom IQ modulation on pulse, it is possible to define an arbitrary IQ waveform to be used as a pulse modulation. The custom IQ modulation is specified through a MATLAB expression that returns a vector of complex values that represents the IQ signal which is sampled at “sampleRate”. The variable “sampleRate” is available in the MATLAB workspace when the function is evaluated. If the resulting vector is too short to fill the entire pulse width, it is repeated. If it is too long, it will be truncated. Example:

* If you enter the following as a custom IQ waveform formula:  
   iqtone('sampleRate', sampleRate, 'tone', [100e6 200e6 300e6])  
  you’ll get a multi-tone signal with 100, 200 and 300 MHz as a pulse modulation. Note, that the variable “sampleRate” is passed to the IQtools function “iqtone” in this example
* To define a more complex formula, you can enter  
   myfunc(sampleRate)  
  as a custom IQ formula and create a MATLAB function e.g. in myfunc.m as follows:  
   function y = myfunc(sampleRate)  
   y = iqtone('sampleRate', sampleRate, 'tone', [100e6 200e6]);

Of course, you are not limited to IQtools built-in functions for the modulation waveform.

## “Custom Phase Code” modulation

For custom phase code modulation, a vector of phases in degrees can be specified. These phase steps are distributed across the pulse in equidistant intervals. The vector must be enclosed in square brackets. Alternatively, a function with no arguments can be specified. The return value of this function must be a vector of phases in degrees. Example:

* To generate a Barker-7 code, one can specify the phase code: 90\*[1 1 1 -1 -1 1 -1]

# Using the “Digital Modulation” utility

This utility allows you to generate digitally modulated waveforms with selectable modulation schemes, pulse shaping filters and impairments – either as I/Q baseband or as RF waveforms. The utility supports integer and fractional oversampling, single or multi-carrier waveforms user-definable relative magnitudes. It also supports amplitude and phase correction in conjunction with the VSA software (see next chapter).

If you change the symbol rate, the utility will pick an oversampling factor and sampling rate such that the sample rate is as high as possible (but within the limits of the selected AWG) and with integer oversampling. If you want to use a particular sample rate, you can type it in. In that case, the oversampling factor will be re-calculated – and might be fractional.

If you set the “Carrier offset” field to zero, the utility will generate an I/Q baseband signal. If carrier offset is non-zero, the waveform will be shifted to the specified frequency.

# Using the “Serial Data” utility

The serial data utility allows you to generate NRZ or multi-level (PAMx) signals with adjustable transition times, pre-emphasis, pulse shape filter with or without additional distortions. Like all the other waveform generation tools, the waveform can be pre-distorted to compensate the frequency and phase response of the AWG and channel. Some comments on the individual fields:

* **Symbol Rate** – specify the baud rate in units of symbols/second
* **Sample Rate** – should be left on “Auto” when possible. In this case, IQtools will choose an appropriate sample rate that works with the required number of symbols and baudrate.  
  If the “Auto” checkbox is unchecked, a particular sample rate can be selected. Note, that in this case, the software must adjust the number of symbols in order to meet the AWG granularity requirements.
* **# of samples** – the number of samples that will be used for the serial data waveform. This field is read-only. It is calculated based on the symbol rate, sample rate and number of symbols as follows: #samples = #symbols / symbolRate \* sampleRate
* **Type of data** – selects the data content and encoding scheme:
  + **Random** – uses the MATLAB “rand()” function to generate the data pattern of the specified length. Format can be selected to be NRZ or PAM4
  + **Clock** – 1-0-1-0-… pattern
  + **MLT-3** – MLT-3 coded random data
  + **PAMn** – PAM signal with n levels and random data content. The voltages for each PAM-level can be individually controlled. By default, they are equally spaced
  + **PRBS2^n**-1 – PRBS pattern of length 2^n-1. The polynomials are hardcoded. They can be modified in iserial.m. Note, that PRBS2^23-1 and PRBS2^31-1 are only supported on the M8195A. Adding distortions on these long PRBS sequences is not yet implemented.
  + **Pattern from file** – see detailed explanation below
  + **User defined** – see detailed explanation below
* **# of symbols / shift** – the number of symbols in the pattern. When the Type of data is changed to a pre-defined pattern, this field is populated with the default length (e.g. for PRBS2^15-1, this field is populated with 2^15-1). But it can be overwritten by the user to be longer or shorter. In this case, the data pattern will repeated the necessary number of times and truncated to the exact length. As mentioned above, the number of symbols is also adjusted by IQtools if the sample rate “Auto” checkbox is turned off
* **User defined data** – see detailed explanation below
* **Pre/Post-Cursors (lin.)** – pre and post cursor taps for pre-emphasis. The number of tap values is not limited. The sum of all tap values should be 1. If delay is not relevant, all coefficients can be entered in the post-cursor field and the pre-cursor field can be left empty
* **Pulse shaping filter** – there are two fundamentally different algorithms implemented for converting a data pattern to a waveform:
  + **Transition time** – the waveform is constructed from sections of “straight lines” and “transitions”. The transition sections take the shape of a raised cosine wave. The transition time parameter determines the duration of the “transition” sections
  + **Raised Cosine, Root Raised Cosine, Gaussian** – In this case, the waveform is constructed by a series of dirac pulses and then filtered with the selected pulse shaping filter. For RC and RRC filters, the filter length (in number of symbols) and the roll-off are adjustable
* **Jitter frequency and shape** – jitter is mathematically added to the waveform. Note, that for sinusoidal jitter, the jitter frequency may be rounded such that an integer number of periods fits into the waveform
* **RJ** – pseudo-random jitter is added to the waveform. Since the waveform length is finite, the jitter pattern will repeat along with the waveform. So, it is not really random.
* **SSC freq / depth** – allows SSC at a certain frequency and depth to be added to the signal. Note that the SSC frequency may be rounded such that an integer number of periods fits into the waveform
* **Noise frequency** – adds sinusoidal level noise at a given frequency. If the frequency value is set to 0, pseudo-random noise is added to the waveform. As with jitter, the noise is not truly random, but repeats with the waveform
* **Noise** – determines the amount of noise relative to the signal amplitude
* **Duty cycle** – allows duty cycle adjustment for both NRZ and PAM4 waveforms
* **Normalized amplitude** – default = 1. Sets the amplitude of the output waveform in “normalized DAC values”. This is relevant when a sequence of waveform segments with different amplitude needs to be generated

### Pattern from file

Allows symbol values to be imported from a .PTRN, .CSV or .TXT file. The .PTRN format is compatible with Keysight BERT data forms. For .CSV and .TXT formats, the data can be in rows or columns, separated by spaces or newline. For NRZ signals, values must be ‘0’ and ‘1’. For PAM<n> signals, values must be between 0 and n-1. Fractional values are possible to define non-equidistant level spacing.

### User defined data

You can either fill in a list of values separated by spaces or comma (e.g. 1 0 0 1 1 1 0 1 0 0 1) or a MATLAB expression that evaluates to a vector of values (e.g. to generate a vector of 256 random 0’s and 1’s, you can put in the expression: randi([0 1], 256, 1) ). In order to make PAM4 signals easier to read it can be written as 1/3\*[0 1 2 3 …], instead of 0, 1/3, 2/3, 1 …

In addition to the “nominal” values (0 & 1 for NRZ; 0, 1/3, 2/3, 1 for PAM4), you can also use any fractional value between 0 and 1 to represent intermediate voltage levels. E.g. 0, 0, 0, 0.8, 0, 0, 0 can be used to generate an isolated “1” that does not quite reach the correct voltage level. To simplify generating PRBS sequences with errors in them, the “User defined data” field behaves as follows:

Whenever you change the selection in the “Type of Data” pull-down menu, the corresponding data pattern is copied to the “User defined data” field. (E.g. when you select PRBS2^7-1, PAM4 as the “Type of data”, the “User defined data” field changes to “1/3 \* [1 2 2 3 2 2 3 1 2 3 3 2 3 0 1 3 2 3 3 1 3 0 3 3 2 0 0 1 1 2……]”). If you change the selection to “User defined data” afterwards, that same data pattern will be generated. Now you have the possibility to change individual symbols by changing one of the numbers in the square brackets to another number in the range 0…3 or even force the signal to go through the middle of an eye by changing one of the numbers to a fractional value between 0 and 3.

# Amplitude and Phase corrections for Digital modulation waveforms

When generating a digitally modulated signal with the “Digital Modulation” utility, you can improve the EVM by performing an amplitude and phase calibration in conjunction with the VSA software. The VSA software has to be installed on the same PC that runs the MATLAB scripts. The connection to the oscilloscope that captures the signal has to be established before using the calibration function in the MATLAB script. The calibration routine uses the Equalizer that is built into the VSA software to determine the channel frequency response. After generating an (un-corrected) signal, the MATLAB script launches the VSA software, turns on the equalizer and uses the frequency response of the equalizer to calculate a pre-distorted waveform. Unlike the flatness correction using multi-tone, this method corrects magnitude and phase of the signal.

Please follow these steps to generate a pre-distorted signal:

1. Set the desired parameters in the “Digital Modulation” tool and press download to generate a digitally modulated signal. Make sure that the “Apply Correction” checkbox is turned OFF.  
   Do **NOT** start the VSA software manually – the script will do that.
2. Press the “Calibrate (VSA)” button. This will start an instance of the VSA software and set up the parameters to demodulate the signal that has been configured.
3. When you see the dialog “VSA measurement running. Please press OK when Equalizer has stabilized”, you should first check the Input range in VSA and then observe the Equalizer stabilizing. If it does not converge, you might have to modify the Equalizer parameters.
4. Once the equalizer is stabilized, press “Ok” to continue in the calibration process. The MATLAB script will read the current equalizer frequency response, display it as a MATLAB plot, download the pre-distorted waveform and turn the equalizer in VSA off.
5. Optionally, you can press the “Calibrate” button again (it will now be labeled “Re-Calibrate”) to further improve the EVM performance.
6. If you make changes to your iqmod parameters, please un-check the “Correction” checkbox, press “Download” and then “Calibrate (VSA)”.
7. The connection between the MATLAB script and the VSA software remains intact until you either close VSA or exit MATLAB. So, for consecutive calibration runs, the VSA software will not be launched again, but the already running instance will be re-used.

# In-system calibration

In-system calibration is a method to characterize the magnitude and phase response of the AWG, and if desired, the device under test including any cables and adapters between them. This information can then be used to pre-distort the AWG signal. As a result, the signal at the reference plane can be calibrated to have a flat response, both in magnitude and phase.

It is assumed that the oscilloscope which is used as the wide-band analyzer has a perfectly flat response (in mag and phase). Both real-time scopes as well as sampling scopes can be configured to show such a flat response.

### Measurement method

The in-system calibration algorithm in IQtools uses the AWG to generate a wide bandwidth multi-tone signal. The maximum frequency and number of tones can be defined by the user. This multi-tone signal passes through the AWG, device under test, cables, adapters, etc. and is then captured by a scope (real-time or sampling scope). The captured signal is uploaded from the scope and by comparing the FFT’s of the transmitted and captured signal, the in-system calibration algorithm can determine the magnitude and phase response of the overall system.

The multi-tone method has been chosen over a step-response method because

1. With a step-response method, there is very little energy at high frequencies and hence lots of averaging needs to be done to reduce the effect of noise. The multi-tone method distributes the energy more evenly across the frequency band of interest
2. The multi-tone signal is more like a typical digital modulation signal which is often used in real applications
3. Some transmission systems are not DC coupled and for those it is difficult or impossible to make a step response measurement

### Running the in-system calibration

### Troubleshooting in-system calibration errors

Before running the in-system calibration, it is a good idea to download a benign signal (e.g. a sine wave) into each of the AWG channels that will be calibrated and check if the signal can be captured on the scope. This is a good time to double check which AWG channel is connected to which scope channel.

The in-system calibration algorithm is quite complex and if the physical setup does not match the configured setup, you’ll get a lot of error messages and it is hard to find out what went wrong.

##### Phase outliers

the "phase outlier" error message pops up when the algorithm cannot reliably determine the phase of the signal. There are several reasons why this can occur:

* One or more of the AWG signals are not properly connected to the scope or the mapping of AWG channels to scope channels in the in-system calibration window does not match the physical setup.  An easy way to test that is to output a simple signal (e.g. a sine wave) and look at it on the scope
* You have selected too many tones.  Since the AWG distributes the available energy across all tones, there is very little energy per each individual tone. This can cause the high frequency tones to be below the noise floor. Try to reduce the number of tones in this case. Somewhere around 200 - 300 tones is generally a good choice. Note, that blindly increasing the number of tones will not necessarily increase measurement accuracy: With a larger number of tones, there will be a finer resolution of the frequency axis, but since the energy per tone is lower, the accuracy of the magnitude and phase for each tone gets worse. Eventually, the high frequency tones will be buried in the noise
* The “max frequency” is too high.  If you are trying to run the in-system calibration up to a frequency where there is too much loss, the high frequency tones will be down in the noise and hence the phase can no longer be determined.  Try to lower the "Max frequency" to a value where you know that the attenuation will be well below the SNR of the system

# Creating sequences

For some of the AWG models (currently M8190A and 81180A), IQtools also support the sequencer that is built into these AWGs. To set up a sequence, you must first define and download the individual **waveform segments** that will later be combined into a sequence. Then you can define the sequence itself. The sequence is a table that describes which waveform segment to use, how often it will be looped and under which condition the next segment will follow.

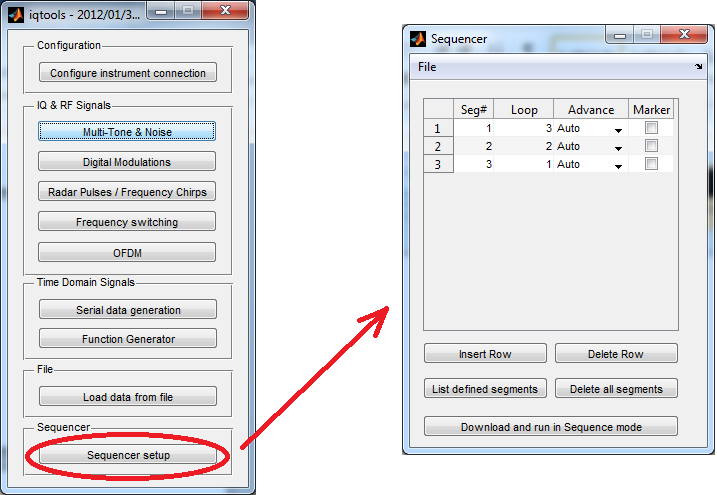
To define the individual waveform segments, you use the “Segment Number” field that is available in each of the individual waveform tools (Multi-Tone, Digital Modulation, Radar Pulse, etc.)



The default segment number is 1, meaning that the waveform will be downloaded to that segment. (Waveform segments are numbered starting from 1 up to the AWG model-specific limit).

In order to set up a number of waveform segments, you choose one of the waveform creation tools (multi-tone, digital modulations, radar chirps, etc.), define the desired parameters, set segment number to “1” and press “Download”. Now you can change any of the parameters (or use another waveform tool), set segment number to “2” and press “Download” and so on until you have defined all the individual segments that you would like to use in your sequence.

Next, you open the “Sequencer Setup” tool from the IQtools main window and edit the sequence table:



In each row of the sequence table, you can specify the waveform segment that you would like to be generated, how many times this waveform segment will be looped, under which condition the sequencer will advance to the next segment and finally whether the marker outputs that you have defined for this segment will be output.

You can insert and delete rows from the table with the respective buttons and also display all the available segments (M8190A only) or delete all of them. Note that when you delete all segments, you have to go back to the waveform tools and download the segments again.

Once you press the “Download and run in Sequence mode” button, the sequence table will be downloaded and the AWG starts to generate the programmed sequence. In the example above, segment#1 will be played three times, followed by segment 2 twice and segment 3 once, then back to segment#1 three times and so on. After the last segment has been generated, the sequence will automatically loop back to the beginning of the table.

The “Advance Modes” offers the following choices:

* “Auto” – the waveform segment will be looped for the programmed number of times. Then, the sequence will automatically proceed to the next table entry
* “Conditional” – the segment will be looped until an external signal is applied to the “Event” input (or SCPI command is sent to the instrument to simulate such an event). A segment will always be completed - independent on when the Event signal is asserted
* “Repeat” – the waveform segment will be looped for the programmed number of times and then the output is paused as that last sample value. The sequence proceeds once the “Event” input is asserted.
* “Stepped” – similar to “Repeat”, except that processing is paused after each loop.

The sequencer allows you to do much more complicated setups using SCPI programming, but the IQtools utility currently only supports those “simple” sequences. There are some more sophisticated sequencer examples in the “M8190A specific examples” section. Please click on the individual examples and take a look at the MATLAB source code for these examples to see how to program the sequencer.

# M8190A-specific utilities: 4-channel synchronization

This utility allows you to demonstrate the synchronization of two M8190A modules (= 4 channels) either with or without an M8192A synchronization module. It is generally possible to synchronize more than two M8190A modules (up to 6 with the M8192A sync module), but the current implementation of IQTools only supports two.

In order to run four M8190A channels fully synchronous, follow these steps:

1. Start the M1890A firmware for the first and second module (you need one instance of the firmware for each module)
2. If you are using the M8192A module, launch the Soft Front Panel for the M8192A module (this includes the firmware). Find the VISA address of the M8192A (it is shown under Help->About)
3. Find out and note the IP address of the oscilloscope and add the oscilloscope to the Keysight IO Connection Expert. If LAN(TCPIP) does not work reliably, try to use USB to connect to the scope. Any DSO, DSA or MSO scope will work.
4. In the IQTools GUI click on "Configure Instrument"
   1. Select M8190A\_12bit or M8190A\_14bit mode. Digital upconversion mode are currently not supported in IQTools (although synchronization works in the same way)
   2. Enter the VISA addresses for both AWG modules (just copy them from the firmware window – you might want to click the “Test connection” button to be sure that connection can be established
   3. Enter the VISA address of the scope (copy it from the IO Control, Connection Expert) – you might want to click the “Test connection” button to be sure that connection can be established
   4. Enter the VISA address of the M8192A if you are using a SYNC module. It is important that your hardware configuration and cabling matches the configuration that you set up in IQTools. If you are NOT using an M8192A module, make sure you uncheck the “Use M8192A” checkbox
5. From the main screen of the iqtools MATLAB GUI select > specific functions > 4-channel sync
6. Make the cable connection described in the Connection Diagram
7. Set the “AWG#2 Clk Source” to “ext sample clock” for best accuracy. (Make sure you have the Sample clocks connected according to the connection diagram). If you are using the M8192A SYNC module, the sample clock connection is part of the blue cable connections.
8. Make sure “Analog Outputs Ch1 used for de-skew … “ is selected
9. Click on „Automated De-skew". This will synchronize the two first channels on both modules.  
   If an error message pops up, look at the scope – it should display the rising edge of a square wave on channel 1 and 2. If one (or both) of the channels show no signal, double check the connection diagram. Also, make sure you using the correct output (DAC out vs. DC out)
10. The modules are now synchronized – you should see all 4 scope channels perfectly aligned on the oscilloscope screen.
11. Choose one of the waveforms in the pull-down menu and click the “Start” or “Stop” buttons to see how different waveform can be loaded without changing the skew between the modules

### For customer specific waveform the script has to be modified:

1. Open the file multi\_channel\_sync.m for editing in MATLAB
2. Go to line 202 and decide which item of the sample list you would like to customize.
3. The waveformID is linked to the point in the drop down menu of the GUI. This gives you an idea which case needs to be modified.
4. The waveform vectors *testSegment1* or *testSegment2* have to be complex due to the IQ philosophy of the scripts.
5. testSegment1 is then downloaded to AWG#1 (primary)
6. testSegment2 is then downloaded to AWG#2 (secondary)
7. Once the modifications are done, save the .m-file and restart the GUI for 4-channel-sync.

Select the modified waveform in the drop down menu and play it by pressing the buttons “start” and “stop”

# Next Steps

1. Learn more about the Keysight M8190A & M8195A arbitrary waveform generators (used in this document) at [www.keysight.com/find/m8190a](http://www.keysight.com/find/m8190a) resp. [www.keysight.com/find/M8195A](http://www.keysight.com/find/M8195A).
2. Learn more about MATLAB software and ordering it directly from Keysight with the M8190A, other arbitrary waveform generators, and other instruments at: [www.keysight.com/find/matlab](http://www.keysight.com/find/matlab)
3. Information on all Keysight signal and waveform generators can be found at [www.keysight.com](http://www.keysight.com)
4. Additional information on using MATLAB with Keysight instruments is available at [www.mathworks.com/keysight](http://www.mathworks.com/keysight)