Application Note

AEROFLEX
A passion for performance.

Converting user-defined IQ data files for use with the IFR 3410 Series Digital RF Signal Generator

IQCreator is a powerful software tool that provides all the necessary features required for the quick and easy generation of complex waveform files. For cases where proprietary baseband signals need to be produced, alternative software simulation tools such as MATLAB can be used to create the required waveform file.

This application note explains how ***IQCreator*** can easily be used to quickly package and download these user-defined waveforms to the 3410 Series ARB.
A discussion on data formats, waveform scaling and waveform spectral re-growth issues during continuous play of time limited waveforms is also included.

Overview

As worldwide development continues to grow in second and third generation wireless communication systems, users demand products with digital modulation capabilities. When fitted with the dual-channel ARB, the IFR 3410 Series is capable of producing high quality digitally modulated carriers with RF modulation bandwidths of up to 25 MHz.

Providing excellent modulation accuracy and linearity, the 3410 Series ARB is capable of generating 2G, 2.5G, 3G, WLAN and PMR digital radio signals.

IQCreator is a Windows™ based software tool that enables a user to create a modulation scheme from which an ARB waveform can be generated. Waveforms can be created that conform to the TDMA, CDMA and WLAN standards regarding timing, spectral distribution and amplitude probability distribution. The resultant waveform can then be downloaded to the 3410 Series ARB.

In addition, ***IQCreator*** includes a utility that allows user-defined waveforms, created using software simulation tools such as MATLAB, to be converted and packaged into a form that can be downloaded into the 3410 Series ARB.

This application note describes how such user-defined waveforms can be packaged and loaded into the 3410 Series ARB. This note discusses:

· The required format of the base band waveform.

· The constraints that are placed on the waveform.

· How a waveform can be packaged and downloaded to the 3410 Series ARB.

· The issues related to wrap around when a time limited waveform is continuously played.

Waveform generation

IQCreator provides all the necessary features required for the quick and easy generation of complex waveform files that are used in R&D and production test environments. Alternatively, for those cases where proprietary baseband signals need to be produced, software simulation tools such as MATLAB can be used to create the required waveform file.

When producing waveforms by this route it is essential that certain design criteria be adhered to if the created waveform is to be successfully generated using the 3410 Series ARB. Some of these are general signal processing concepts, while others are constraints that are placed upon the waveform designer by the 3410 Series ARB hardware. The following sections address some of the more important issues to consider when generating a waveform.

Waveform sampling rate

The following factors govern the choice of sample rate.

1. The bandwidth of the signal to be generated

The bandwidth of the signal will limit the minimum sample frequency that is allowed according to the Nyquist rate. In simplistic terms the Nyquist theorem states that a band-limited signal must be sampled by at least twice the band-limited frequency of the signal. In other words, if we have a signal that is limited to 1 MHz, then this signal must be sampled at more than 2 MHz if it is to be uniquely determined by its samples. In practice the use of 4 times over-sampling is recommended.

The maximum IQ (complex) bandwidth supported by the 3410 Series ARB is approximately 25 MHz - generated from a 12.5 MHz base-band bandwidth.

2. 3410 Series ARB hardware constraints

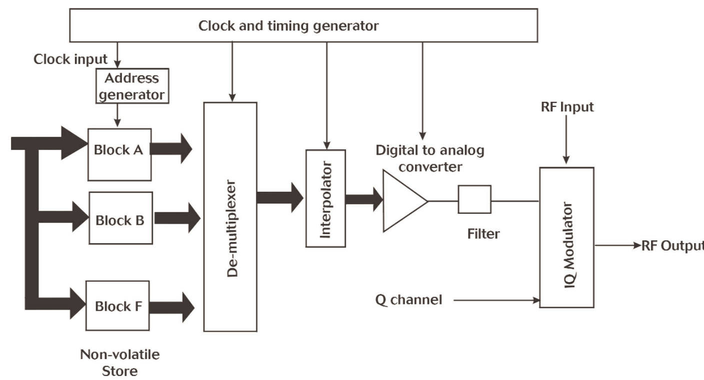

Figure 1 - 3410 Series ARB block diagram

The 3410 Series ARB is able to generate long test sequences due to its large memory size and the use of hardware interpolation techniques. A maximum of 22.5 Msamples can be stored. The memory is partitioned into 60 sectors, each of which can be further divided into 3 sub-sectors. Consequently, up to 180 waveforms can be stored at any given time. Sectors and sub-sectors can be merged together to form larger waveforms. In fact it is possible to save a single waveform that fills all the memory space. For example, the 3410 Series ARB is capable of storing a 3GPP waveform representing 153 frames. This equates to 1.53 seconds. The use of non-volatile memory ensures that it is possible to rapidly change between waveforms.

Waveforms loaded into the 3410 Series ARB can have sample rates ranging from 17 kHz to 66 MHz. Such a broad range of sample rate is possible because the design uses interpolating filters that ensure that the digital-to-analog converters (DACs) receive an optimal sample rate. Due to this design, it is essential that the following conditions are adhered to:

a) The input waveform must not be compensated for sinx/x distortion. This is taken care of in the hardware design.

b) The 3410 Series ARB design (hardware interpolation) assumes that the input waveform has been over-sampled by at least 4 times the symbol rate of the original signal and that the bandwidth of the signal is less than one sixth of the waveform sample frequency.

Note if the waveform sample rate is above 44 MHz the interpolation factor implemented in hardware will be one and the constraints given in b) do not apply.

3. Waveform length

The maximum length of the waveform is limited by the amount of memory in the instrument. The maximum size of an IQ waveform that can be stored in the 3410 Series ARB is 23592950 samples. The minimum length of a waveform is 50 samples.

Obviously, the length of the waveform in time is determined by the sample rate. For example, 10 ms of a 3GPP waveform that has been sampled at 4*3.84 Msamples per second will require 153600 samples. In this instance, there is enough memory to store 1.53 seconds of such a waveform.

There is also a limit to the number of different waveforms that may be stored. The maximum number of different waveforms is 180. In this instance the maximum length of any single file is 131072 samples and the sample rate of each file must be no greater than 22 MHz.

Formatting and downloading a waveform

Before a user-defined waveform can be generated by the 3410 Series ARB, it has to be formatted and downloaded to the instrument. ***IQCreator*** includes a utility that converts and packages a file into a form that can be downloaded into the 3410 Series ARB.

For user generated waveforms, three file formats are supported:

1. 16-bit and 32-bit signed integer binary files that are in Little Endian format with samples stored as I then Q then I then Q etc. (referred to as IQIQIQ from this point on).
2. 32-bit IEEE floating point notation, again with samples stored as IQIQIQ.
3. ASCII format.

16-bit integer format

The 16-bit signed integer format refers to a 16 bit twos complement representation, with a range of -32768 to 32767. As the 3410 Series ARB uses 14 bit DACs, the waveform values must be scaled so that the peak values do not exceed $\pm 2^{13}$-1. ***IQCreator*** assumes that the waveform has been scaled to lie within the 14-bit range. It is important that the sign information contained in bits 14 and 15 is not removed. In other words the 16 bit twos complement representation must be retained as the front panel software reads the stored data as 16-bit signed integer values. The waveform must be stored as IQIQIQ samples, with I as the first sample. In addition, the binary file must be in Little Endian format. This is the default for Intel based machines. Assuming that a waveform has been generated using Matlab, and that the IQ data are stored as complex vectors (i+ jQ), then the Matlab function, saveiq, given below may be used to save the samples in a format that can be readily downloaded into the 3410 Series ARB using ***IQCreator***. Details on how to download waveforms are given later.

```
function [error, msg] = saveiq(file_name, iq_data,
req_peak);
% This functions saves the i and q data as 16-bit
% integers in the order I Q I Q. The data is scaled so
% the peak vector is equal to req_peak. If the %required peak
value is greater than 2^13-1 or ≤ 0 an
% error will occur (function will return -1). Note that
% iq_data is assumed to be complex
error = 0;
msg = [];
if ((req_peak > 2 ^ 13-1) | (req_peak <= 0))
msg = sprintf('Peak value is not valid');
error = -1;
else
peak = max(max(abs(real(iq_data))),
max(abs(imag(iq_data))));
scale = req_peak/peak;
int_iq_data = zeros(1,2*length(iq_data));
int_iq_data(1:2:end) = round(real(iq_data)*scale);
int_iq_data(2:2:end) = round(imag(iq_data)*scale);
[fid msg] = fopen(file_name,'w');
if (fid ~= -1)
fwrite(fid, int_iq_data, 'int16');
fclose(fid);
else
error = -1;
end
end
```

Figure 2 - MATLAB saveiq function

The function above shows that the IQ data is stored as signed 16-bit integers that have been scaled to a peak value selected by the user. This particular function also ensures that the maximum peak value is less than or equal to $\pm 2^{13}$-1. For the 3410 Series ARB it is best to use a peak value of 8191. This ensures that the full range of the DACs is utilized.

32-bit signed integer format

The 32-bit integer format takes the same form as the 16-bit signed integer format, except that the data is stored in 32 bits twos complement format. The data must still be scaled to have peak values that are less than or equal to $\pm 2^{13}$-1.

IEEE 32-bit float

The float format uses the IEEE (Institute of Electrical and Electronics Engineers) single precision format. This format is represented by 4 bytes, consisting of a sign bit, an 8-bit excess-127 binary exponent and a 23-bit mantissa. The mantissa represents a number between 1.0 and 2.0. Since the high-order bit of the mantissa is always 1, it is not stored in the number. This representation gives a range of approximately 3.4E-38 to 3.4E+38. ***IQCreator*** requires the data to be scaled such that the peak values are less than or equal to 1.0. Again the samples must be stored as IQIQIQ, with I being the first sample.

ASCII format

The ASCII format is float delimited by commas, spaces or tabs, line feeds or carriage returns. The IQ values must lie within the range ±1. Each line must have no greater than 300 characters. Values are read and scaled so that the peaks are limited to 14-bits and assumed to occur as IQIQ, with I being the first sample.

Preparing a waveform for download

Before you can download your waveform to the 3410 Series ARB, the waveform must be 'packaged' into a form that is understood by the instrument. Once this process is complete, the packaged waveform can be downloaded and stored on the 3410 Series ARB.

To package your waveform, start the ***IQCreator*** application and select the menu Generate ARB File! This will then provide you with the dialog box shown in Figure 3.

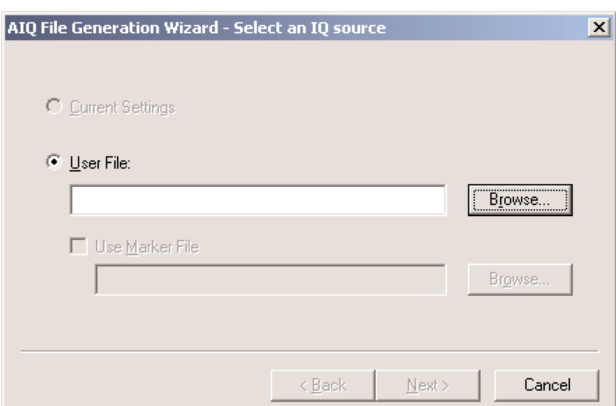

Figure 3 - Waveform selection dialog box

Select your waveform by clicking the Browse button. Once you have selected your waveform, you will need to set the data format and sampling rate for the waveform. To do this click the Next> button. The dialog box shown in Figure 4 is produced.

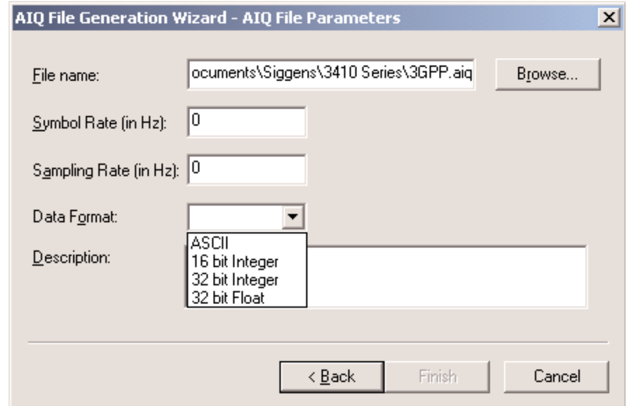

Figure 4 - Waveform parameter dialog box

Select the data format from the available list and specify the sampling rate in Hz. The symbol rate and description fields are optional. Finally, specify a filename for the packaged waveform. Once you are happy with the selections, click the Finish button.

Now that the waveform is packaged you are ready to continue to the next step of downloading and playing the file.

Downloading a waveform

Downloading a waveform to the 3410 Series ARB is performed via GPIB. Having connected the instrument to a GPIB capable computer, from 12345 select the menu 34XX, Configure Remote. This will provide you with the dialog box shown in Figure 5.

Figure 5 - GPIB Address dialog box

Having specified the GPIB address of the 3410 Series, downloading the file is a very simple and straightforward operation. Again from ***IQCreator*** select the menu 34XX, Download AIQ file. This will then provide you with the dialog box (Figure 6) prompting you to select a file for downloading.

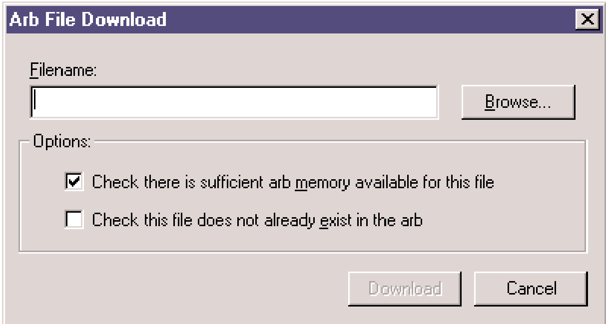

Figure 6 - AIQ File selection dialog box

Once you have selected a file and performed the download, the 3410 Series user-interface can then be used to select the waveform for generation. Please refer to the 3410 Series operating manual on how to do this and to configure other parameters such as carrier frequency and RF level.

Waveform scaling

It was mentioned earlier that the 3410 Series ARB uses 14-bit DACs. This means that the waveform must be scaled such that the maximum peak value does not exceed $\pm 2^{13}$-1.

Note: The 3410 Series ARB uses the RMS value of the waveform file to automatically set the level. By reducing the RMS level, the amplifier gains are automatically increased. The gain settings may be over-ridden using the RMS offset facility that is available on the front panel of the instrument.

Waveform continuity

The previous sections have given a brief description on how one may generate and download a waveform. It has not discussed any constraints that are placed upon a waveform that is being continuously repeated by an arbitrary signal generator. To prevent periodic spectral re-growth it is important to ensure that there is no discontinuity between the start and end of a waveform. This is not a trivial matter and is very dependent upon the type of waveform that is being generated. This section gives some guidelines that can help to ensure that the waveform you generate is cyclic. The simplest example to consider is the generation of a single sideband carrier. In this case we would generate a cosine waveform for the I vector and a sine waveform at the same frequency for the Q vector.

Let's assume that we want to generate a side-band at 10 kHz and that we intend to sample this at 45 MHz. To ensure that the waveform is repeatable we must have an integer number of complete cycles of the sinusoids. If we consider one period, this is equivalent to 0.0001 seconds or 4500 samples. In this case sample 4500 will occur at t = 0.0001-1/Fs seconds. The following sample (i.e. sample 0) will then be equal to the sample at t = 0.0001 which is what we need to ensure no discontinuity.

This is all very well for signals that have not been filtered. If a channel filter has been used, we have to ensure that the filter delay has been taken into account and that the basic waveform is cyclic. In general this can be achieved by using a circular filter. The best way to appreciate this is to look at some examples. First consider the waveform in Figure 7 which is a filtered I channel of a QPSK signal.

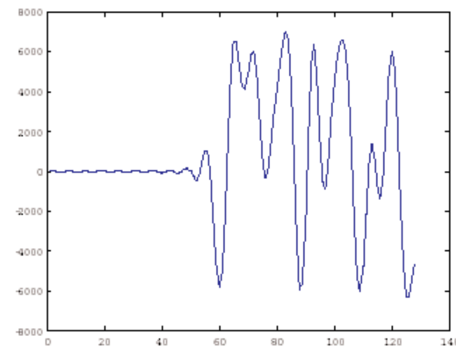

Figure 7 - Filtered I channel of a QPSK signal

From this figure we can see that if the above signal was continuously repeated there would be a discontinuity at the transition between the end and start of the waveform, as shown in Figure 8.

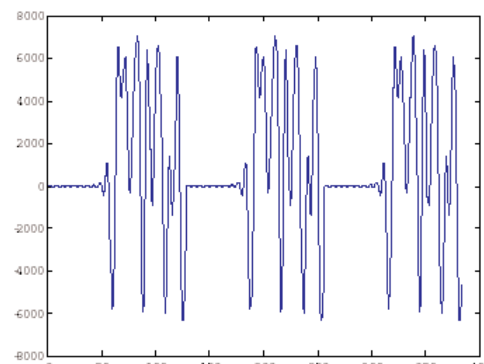

Figure 8 - Wrapped version of original QPSK signal showing discontinuities

The main reason for this is that the delay due to filtering has not been taken into account. One way to remedy this situation is to circularly filter the I and Q data. This can be achieved by repeating the input data to the filter for the duration of the filter delay and storing the full delayed version of the signal.

In other words, consider the plots in Figure 9. Figure 9a is the original input to the filter. Extend this by filter delay samples by appending the start samples to the end. This is shown in Figure 9b. Then filter this extended signal, the result is shown in Figure 9c. Once this has been done take the delayed version and save this as the cyclic waveform (Figure 9d). The repeated version of this signal is shown in Figure 10.

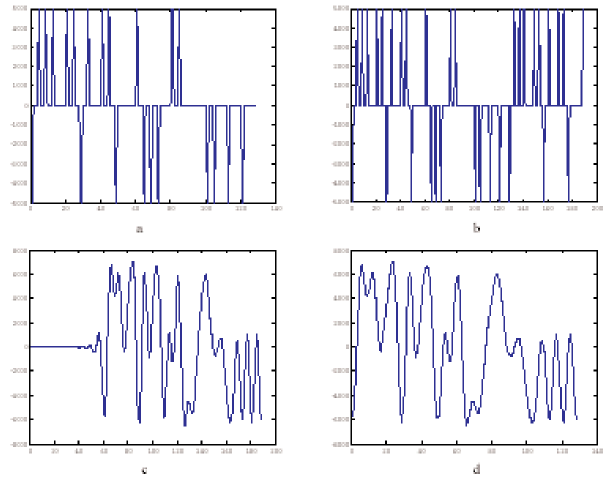

Figure 9 - Filtering the signal to make the waveform cyclic

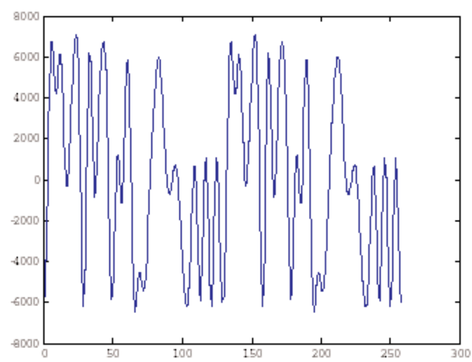

Figure 10 - Cyclic waveform after filtering

A Matlab function that performs a cyclic extension is given below.

```
function y = cyclic_filter(filter_delay, h, x)
% This function will take an input signal, extend it by
% filter_delay samples. Then filter the extended input
% using the coefficients in h. The output y is the
% filtered input with the delay removed. In general
% the resulting waveform will be cyclic.
[m,n] = size(x);
if m > n
x = x';
end
x_ext = [x,x(1:filter_delay)];
x_filt = filter(h,1,x_ext);
y = x_filt((filter_delay:end));
```

Figure 11 - MATLAB cyclic extension function

Obviously, not all waveforms can be made to be cyclic as shown in the above example. The topic is complex and each type of waveform will have to be analyzed individually to ensure that it is cyclic. For example, D8PSK waveforms can be cyclic in the sense that there is no spectral discontinuity, but there may be a symbol discontinuity. That is, a symbol transition from the end of the waveform to the start of the waveform may not be a valid one. In such cases, the number of symbols in the waveform will have to be considered, or the bit sequence will have to be chosen so that the transitions from the end to the start are valid.

It is beyond the scope of this application note to go into further details on the subject. Suffice to say that in general cyclic filtering will be adequate. However, instances will occur that require additional techniques and precautions.

CHINA
Tel: [+86] (10) 6467 2716
Fax: [+86] (10) 6467 2821

EUROPE
Tel: [+44] (0) 1438 742200
Fax: [+44] (0) 1438 727601

FRANCE
Tel: [+33] 1 60 79 96 00
Fax: [+33] 1 60 77 69 22

HONG KONG
Tel: [+852] 2832 7988
Fax: [+852] 2834 5364

SCANDINAVIA
Tel: [+45] 9614 0045
Fax: [+45] 9614 0047

SPAIN
Tel: [+34] (91) 640 11 34
Fax: [+34] (91) 640 06 40

UNITED KINGDOM
Tel: [+44] (0) 1438 742200
Toll Free: [+44] (0800) 282 388 (UK only)
Fax: [+44] (0) 1438 727601

USA
Tel: [+1] (316) 522 4981
Toll Free: [+1] (800) 835 2352 (US only)
Fax: [+1] (316) 522 1360

AEROFLEX
A passion for performance.

www.aeroflex.com

info-test@aeroflex.com

Our passion for performance is defined by three attributes represented by these three icons: solution-minded, performance-driven and customer-focused.

Part No. 46891/920, Issue 1, 08/03