



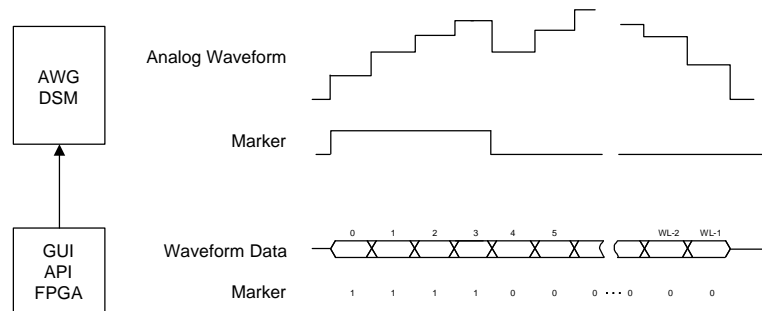
Application Notes – Module_02

Generate Analog Waveforms

This application note enumerates the procedures to generate analog waveforms in the GUI, API-based applications, and hardware-to-hardware configuration. In order to generate an analog waveform with AWG/DSM, users must:

- 1) Power up the module and feed the clock with proper power level;
- 2) Define the waveform;
- 3) Select user page and channel;
- 4) Use GUI or optional GPIO to download the waveform;
- 5) Restart the module in Free Running mode or Arm the module and provide trigger signal in burst mode.

The module hardware takes *waveform data* and *markers* as inputs from the GUI/API via the USB connection. With the GPIO option, the module hardware can also accept commands and download waveform data and markers from external hardware, such as an FPGA-based controller/data generator. Both USB and GPIO based controls and data download can coexist. Users can use the GUI to download their waveform data and, on the other hand, use their hardware FPGA to control the modules to restart/abort. This coexistence of both controls gives flexibility to developments.



In GUI and API-based applications, the *waveform data* can be *built-in waveforms*, *user-defined files*, and *user-defined stream files*. The built-in waveforms are defined by waveform codes with waveform-specific parameters. User-defined files are ASCII-based text files with data listed in chronological sequence. The user-defined stream files are binary files, which can be generated by using the datastream button. *Markers* are a part of the waveform. Each marker is 1-bit digital data with length of multiplex factor of sampling clocks.

With optional GPIO connections, users can download the waveform data from their hardware onto the modules. However, the waveform data must be ‘regulated’ due to memory segmentation and allocation in the AWG/DSM. In the GUI, there is a button to translate the waveform data to regulated data.

DEFINE WAVEFORM

For the GUI and the API, a waveform is composed of a *waveform code*, *common parameters* and *waveform-specific parameters*. The waveform code is an index to select which waveform style should be used to generate the waveform contents. The common parameters are: *Delay*, *Data Length*, and *Marker Parameters (Start, Width, Polarity, and Enable.)* The waveform-specific parameters are the parameters needed to characterize the selected waveforms.

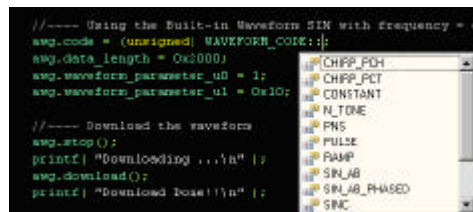
In the GUI, the Waveform Tab provides all the controls to define waveforms. Users can select the waveform style by waveform codes. Once the waveform code is specified, corresponding waveform-specific parameters are prompted in the waveform tab. Users can parameterize these parameters to define the waveform. The details of the waveform-specific parameters can be found in the. Users can use the common parameters to specify the data length and the markers' widths and positions.

In API-based applications, waveform code and the common parameters are public properties and can be set by users. For example:

```
awg.code = (unsigned) WAVEFORM_CODE::CHIRP_PCT;
awg.data_length = 0x4000;

//---- Marker
awg.marker2_enabled = true;
awg.marker2_polarity = true;
awg.marker3_enabled = true;
awg.marker3_polarity = true;
awg.marker1_start = 0x10;
awg.marker1_width = 0x100;
awg.marker2_start = 0x200;
awg.marker2_width = 0x100;
awg.marker3_start = 0x100;
awg.marker3_width = 0x100;
```

It is recommended to use the enumeration class `WAVEFORM_CODE` to specify the waveform code rather than to use a numerical value. In Microsoft Visual Studio's editor, users can see the drop down list of all available waveform codes in the `WAVEFORM_CODES` class as:



In API-based applications, once a waveform code is specified, the waveform can be parameterized by properties called *generic waveform parameters*: waveform_parameters_u0~7 and waveform_parameters_d0~7 are generic parameters for the API waveform generator. The detailed correspondence of generic waveform parameters and waveform-specific parameters can be found in API example.zip ([link](#)), where there is a file called AWG_Waveform_Explorer.dat. For example, for waveform code = 50:

```
Code= 50
  Style: Pulse
  Parameter number: 5
    parameter[0]:      Tini (samples) (Int) (u)      waveform_parameter_u0
    parameter[1]:      Tr (samples) (Int) (u)        waveform_parameter_u1
    parameter[2]:      Tw (samples) (Int) (u)        waveform_parameter_u2
    parameter[3]:      Tf (samples) (Int) (u)        waveform_parameter_u3
    parameter[4]:      Amp (full scale) (Dec) (d)    waveform_parameter_d0
```

For waveform code 50, the style name is Pulse and there are 5 waveform-specific parameters for the Pulse waveform. The first parameter is *Tini* in units of *samples*. It is an integer (*Int*) and the data type in the API is *unsigned (u)*. To set the parameter, simply assign the property in the codes:

```
waveform_parameter_u0 = 0;
```

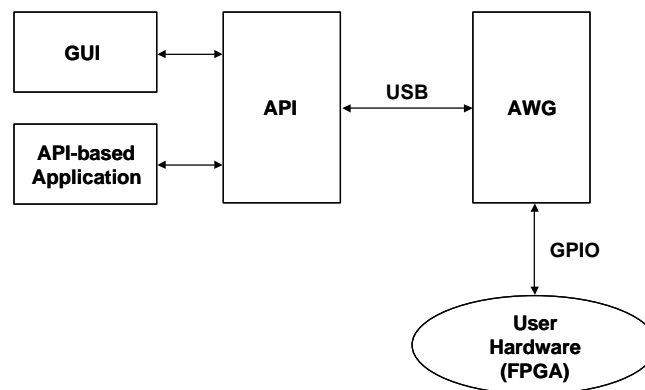
The second occurrence of an unsigned parameter, *Tr*, is designated as *waveform_parameter_u1*; the third occurrence of an unsigned parameter, *Tw*, is *waveform_parameter_u2*, and the fourth occurrence of an unsigned parameter, *Tf*, is *waveform_parameter_u3*. The last parameter *Amp* is defined relative to full scale, where 1 is for amplitude equal to full scale. The *Amp* is a decimal parameter and with a *double* data type in the API, thus being designated as *waveform_parameter_d0*.

DOWNLOAD WAVEFORM

The common parameters, waveform code, and waveform-specific parameters inform the API what kind of waveform formula should be used to generate the waveform data. However, the AWG/DSM still contains no waveform data until the download is performed. In the download process, the API calculates the waveform data based on the parameters entered and sends the waveform data to the AWG/DSM. Defining a waveform only specifies what kind of waveform data to generate, and download is the real action to calculate the waveform data and to send them to the AWG/DSM.

There are three ways to download the waveform data to the AWG's memory.

1. In the GUI click the download button.
2. In API-based applications, use the *download* method.
3. With the GPIO option, use the GPIO download commands and sequence to transfer the data to the AWG. This is purely done in the hardware and there is no need for a USB connection between the PC and AWG.



In user applications, after parameterizing the waveform, users can use the *download()* method to download the waveform data to the AWG. Before calling the *download* method, the applications must:

- 1) Select the user page.
- 2) Select the channel if using a multi-channel module, such as AWG272 or AWG472.

In GUI and API-based applications, the waveform data is downloaded via USB. Without a PC host or USB connection, the waveform data can be downloaded by user hardware via a 24-bit GPIO bus. GPIO is an option to allow users' hardware to control and communicate with the AWG. One of the beneficial features of the GPIO is to download high-speed data to the AWG without a USB connection to a PC host. For details, please reference the GPIO application notes.