Signadyne AWG

Monday, August 9, 2021 11:38 PM

Updated 9/1/21

TODO:
- ~~Figure out how to change temporal duration of the modulation~~
- ~~generate triggered two channel output.~~
- If necessary, integrate into CsPy/instr server.

HSDIO 2 ch 7 – AWG trig

Set up to run python scripts

Run the Essentials installer to install all of the Signadyne software and drivers. Note that the virtualknob hardware will not function because it needs a license, which we neither have nor can obtain, as Signadyne no longer exists. Keysight, which acquired Signadyne, no longer has the old Signadyne software and can't help with this.

The scripts provided here run as expected on Rb lab's PXI controller. The comments + plus relevant documentation in the Signadyne manual should be sufficient for making sense of what is going on.

☆ Useful points when creating an arbitrary waveform:
- The waveform points represent how much to change a default parameter value: amplitude, frequency, or phase. If the default amplitude on a particular channel is 0 V, and my amplitude ramp is configured to have deviationGain = 2, the peak value of my waveform will be 2 V. The points in the waveform array itself should always range from -1, 1, and then be scaled with deviationGain.
- The duration of the waveform is set by the prescale argument in the AWGqueueWaveform function. This changes the effective sampling. See manual. I find it helpful to use a lambda function to determine the prescale factor. Note that because the prescale factor must be an integer, you will get different sampling rate than desired if you do not carefully choose your waveform pts for your desired duration.
- At least for amplitude ramps, the value does not return to the default value at the end of the ramp, so the desired endpt should be set with the ramp.

Performance notes
The jitter of my waveform is ~ 100 ns with the onboard clock at 100 MHz. Try using external clock for better performance.

Installed the signadyne software and rebooted the pxi crate.

☆ Python

WORKAROUND: ISSUE: Two channel outputs with external trigger do not obey the imposed startDelay in the AWGqueueWaveform. Soln: a delay can be artificially added in the waveform pts created, e.g. by Changing the mean of a Gaussian distribution.

SOLVED: ISSUE: the exact same waveform is being played back for each channel. I offset one of the Gaussian waves, but this doesn't show up. More convincingly, I made that waveform a linear ramp. It doesn't show up. Soln: the waveform queue waveformNumber should refer to the number of the waveform in RAM to playback.

Tried:
- startDelay = 0 in one channel, delay between 10 and 10000 in another
- StartDelay < 0 in one channel: the channel is apparently never played out

SOLVED: ISSUE: waveform generated with prescale > 0 looks distorted: my Gaussian looks like a triangle ramp. Soln: at least in the case of using an array to define the waveform, more points seems to fix the problem. I used 100 pts.

SOLVED: ISSUE: waveform not being played on external trigger; I get a constant . Only getting a waveform with modulation when using autotrigger, which overlaps the start/end of each waveform.

Soln: Use AWGstart after all of the card/channel/waveform setup. Trigger is set to EXTTRIG_CYCLE, and nCycles=0.

SOLVED: ISSUE: waveform output with Xiaoyu's awg_ramp.py is a 20 MHz triangle wave, regardless of the frequency and modulation settings. Soln: need to have wavetype set to sinusoid in the channelWaveShape.

Tried:

- In code: changing cycle number, deviationGain, channel frequency
- Scope: trigger with holdoff, checked BW, impedance setting

SOLVED ISSUE: python import signadyne line fails, can't find dll. Soln: need to install Python 3.6.8 to get the dll to load. Signadyne import succeeds and Xiaoyu's code runs without error.

When I run Xiaoyu's awg_ramp.py, the signadyne import fails. It thinks there is an dll missing; it is not. It is in Signadyne/shared. Here, I explicitly added that to the path in the signadyne.py module

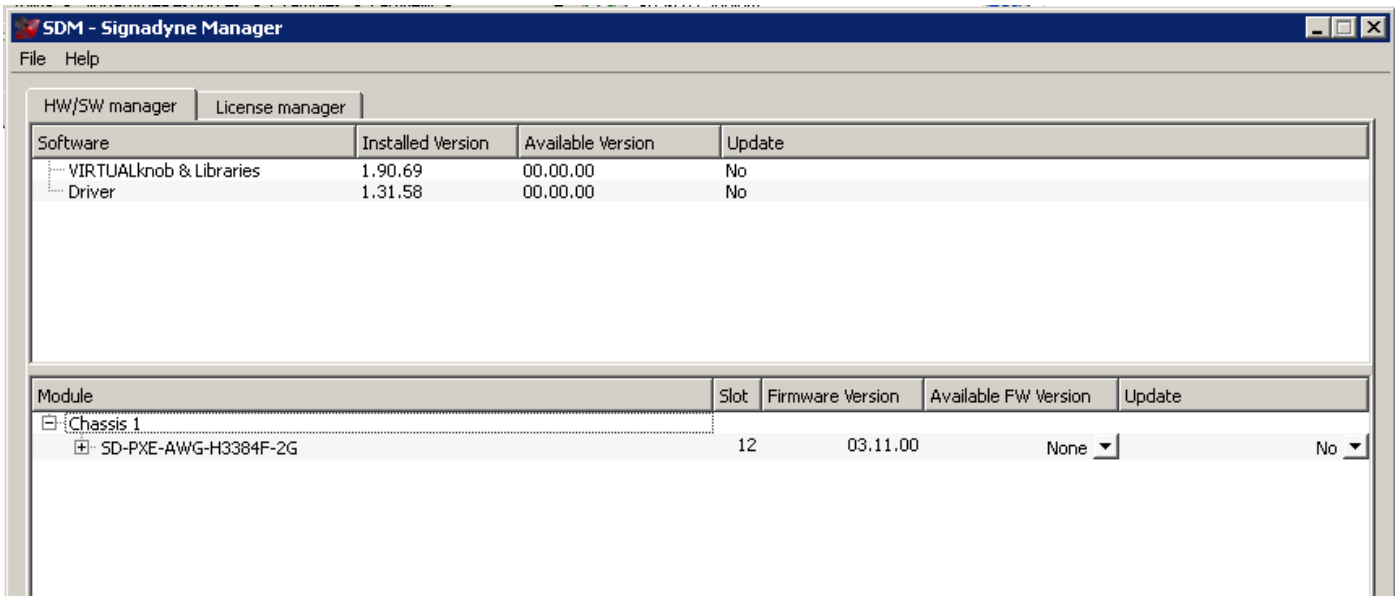
Tried:

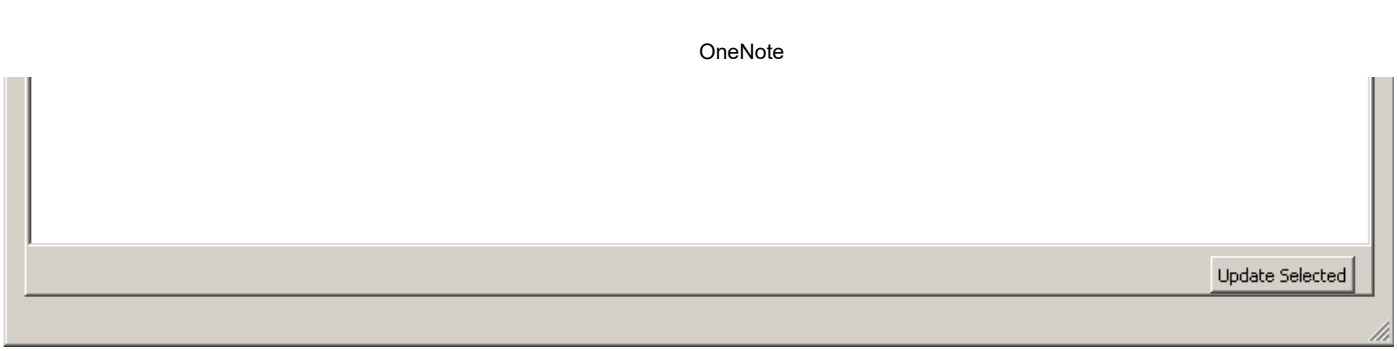
- Running python 3.8 in a pipenv environment.
- running the command line as admin first
- Inserting/appending the dll path in the signadyne.py
- Adding the dll path in signadyne.py with os.add_dll_directory()
- Explicitly calling the dll by its full path

```
(AWG-zdnYF9VH) C:\Users\RbPXI\LabSoftware\AWG>python awg_ramp.py
path: ['C:/Program Files/Signadyne/Libraries/Python', 'C:\\Users\\RbPXI\\LabSoft
ware\\AWG', 'C:\\Users\\RbPXI\\.virtualenvs\\AWG-zdnYF9VH\\Scripts\\python38.zip
', 'c:\\python38-32\\DLLs', 'c:\\python38-32\\lib', 'c:\\python38-32', 'C:\\User
s\\RbPXI\\.virtualenvs\\AWG-zdnYF9VH', 'C:\\Users\\RbPXI\\.virtualenvs\\AWG-zdnY
F9VH\\lib\\site-packages', 'C:/Program Files/Signadyne/shared']
Traceback (most recent call last):
  File "awg_ramp.py", line 5, in <module>
    import signadyne_debug
  File "C:/Program Files/Signadyne/Libraries/Python\signadyne_debug.py", line 15
8, in <module>
    class SD_Object :
  File "C:/Program Files/Signadyne/Libraries/Python\signadyne_debug.py", line 15
9, in SD_Object
    __signadyne_dll = cdll.LoadLibrary("SD_DLL_009600" if os.name == 'nt' else "
libSD_DLL.so")
  File "c:\python38-32\lib\ctypes\__init__.py", line 451, in LoadLibrary
    return self._dlltype(name)
  File "c:\python38-32\lib\ctypes\__init__.py", line 373, in __init__
    self._handle = _dlopen(self._name, mode)
FileNotFoundError: Could not find module 'SD_DLL_009600' (or one of its dependen
cies). Try using the full path with constructor syntax.
```

NIMAX only identifies devices in slots 1-5 and 16-18, which are all marked superscript H on the chassis. for hybrid (both express and non-express compatible).

The AWG, in slot 12, does show up in the Signadyne Manager:





☆ LabVIEW

All of the LabVIEW VIs have broken subVI links. Just right click each to relink.

LabVIEW aou_awg example runs without error! Use part number SD-PXE-AWG-H3384F-2G from the manager identification.

TODO: read how to create the sort of waveform file that the Signadyne VIs expect.