How to DIY a DAC Instrument

by Using XEM3001 and AD5676R

This article explains how to make a DAC instrument by using XEM3001 and AD5676R board. We hope that readers can build a DAC instrument by following this article, and then can use this DAC instrument by using the .bit file and python module shared in our GitHub.

The general specifications of this instrument are:

Output Range: 0 ~ 5.0V

Output Resolution: 16 bits (0.076mV)

Relative Error: ± LSB 3 bits at 16 bits (0.61mV) Maximum

Total Unadjustable Error: ±0.14% FSR (Full Scale Range) Maximum

Offset Error: ± 1.5mV Maximum

Gain Error: ± 0.06% FSR(Full Scale Range) Maximum

Random Noise: 1.0mV (peak)

Clock: 50MHz

Response Time: 1080 ns (27x2 clocks. From receiving the command to output the voltage).

Command Input Speed (via USB interface): 1140 commands / sec (average 0.88 ms per command).

1. Hardware

1.1 Hardware Needed

(1) FPGA board: Opal Kelly XEM3001v2.

(2) DAC board: Analog Device AD5676R evaluation board.

(3) Wires to connect two boards.

(4) Power Supply: 3.3V and 6.0V.

(5) Elements to make low-pass filters for voltage outputs.

(6) Box to fix all the hardware.

1.2 Hardware Integration

Fig. 1. shows our almost finished DAC instrument.

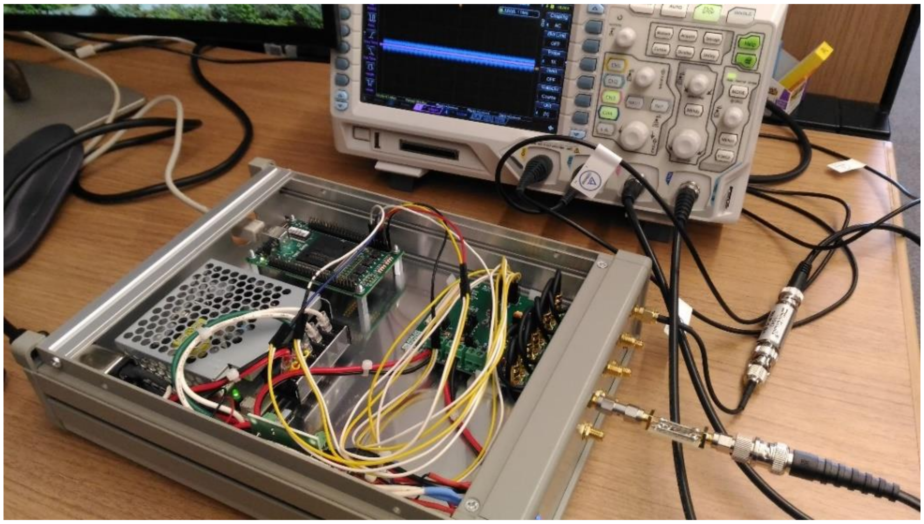

Fig. 1. Our DAC instrument in test

The internal connections and settings can be found in Fig. 2, Table 1, and Table2. The circuit diagram of the low-pass filter can be found in Fig. 3. Fig. 4. and Fig. 5 are the frequency response graphs of the low-pass filter. Please refer to the appendix file: ***PartsLists.xlsx*** to get the detail about the parts.

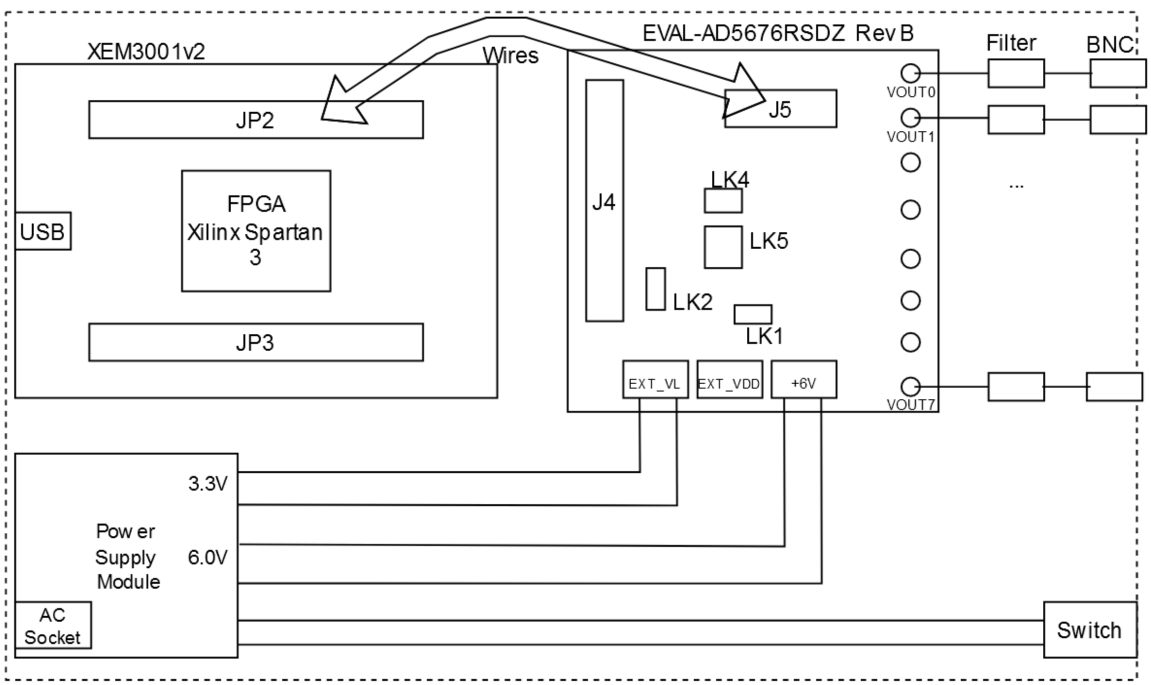

Fig. 2. The schematic of our DAC instrument.

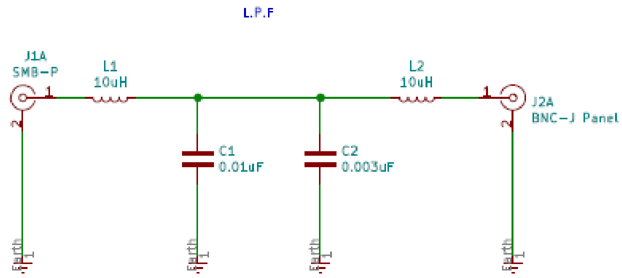

Fig. 3. The Low-Pass Filter

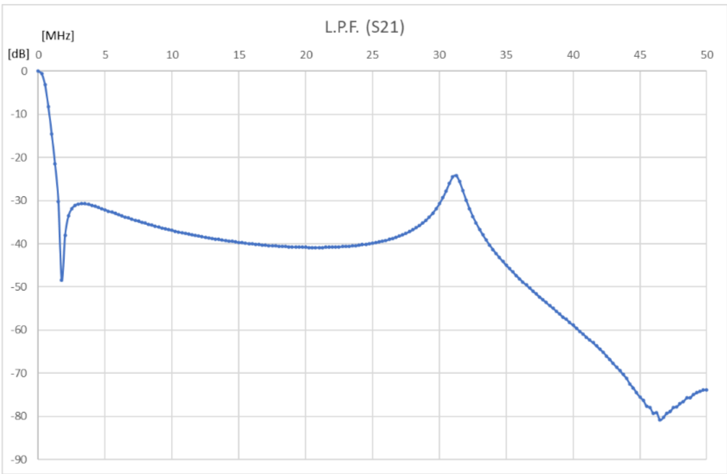

Fig. 4. The Frequency Response of the Low-Pass Filter(0~50MHz)

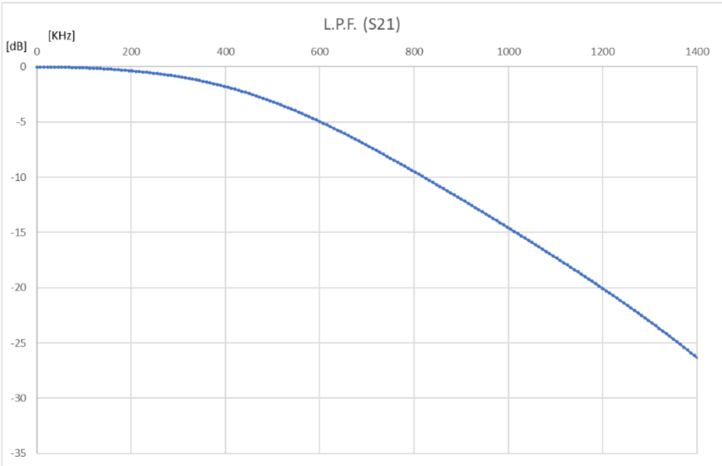

Fig. 5. The Frequency Response of the Low-Pass Filter(0~1.4MHz)

Table 1. The wire connection between XEM3001v2 and AD5676R boards.

XEM3001v2(JP2)		AD5676R(J5)
1	->	GND
5	->	SCLK
6	->	DIN
7	->	RESET
8	->	SYNC
11	->	LDAC

Table 2. Link settings on AD5676R board

Link No.	Setting	Explanation
LK1	A	For internal voltage INT_VCC
LK2	B	For external digital supply EXT_VLOGIC
LK3	NA	
LK4	A	For power up to zero
LK5	B	For a gain of 0V to 2 x Vref

2. Software

2.1 Software Needed

(1) FrontPanel API

The FrontPanel package can be obtained from Opal Kelly when you purchase their product (eg. XEM3001v2). FrontPanel API is included in the package. It will be installed by default when you install the package.

(2) XEM3001_AD5676R_DAC.py and ad5676r_dac.bit

XEM3001_AD5676R_DAC.py is the python module developed by us for using the DAC instrument. It has example codes in its comment lines to tell users how to use it.

ad5676r_dac.bit is the compiled FPGA logic for the DAC instrument. It is needed by XEM3001_AD5676R_DAC.py to initiate FPGA. Just put the .bit file in the same directory (folder) with XEM3001_AD5676R_DAC.py.

2.2 Software Installation and Settings

(1) FrontPanel API

On Windows PC, FrontPanel package is installed in "C:\Program Files\Opal Kelly\FrontPanelUSB\" by default. Let's call this directory as < FP_root > .

FrontPanel API libraries will be installed in "<FP_root>\API\lib\x64".

FrontPanel API Python module will be installed in "<FP_root>\API\Python\3.7\x64".

To make above libraries and module found and used from any directory, following environment variables need to be added:

- Add "<FP_root>\API\lib\x64" and "<FP_root>\API\Python\3.7\x64" to **PATH**.
- Add "<FP_root>\API\Python\3.7\x64" to **PYTHONPATH**

(2) XEM3001_AD5676R_DAC.py and ad5676r_dac.bit

Put these 2 files in the same directory with your Python codes.

Then you can begin to use the DAC instrument, by controlling the channels and voltages from your codes (refer to the example codes in XEM3001_AD5676R_DAC.py).

3, Test

3.1 Functions

The functions are tested that the users can control DAC output by output channel and output voltage as designed.

3.2 Output Noise

Fig. 6 and Fig. 7 are the spectrum graph of the AC part of the output with a mean voltage set at 3.5V, without and with the LPF(low-pass filter) respectively. Spectrum range: 0~500MHz.

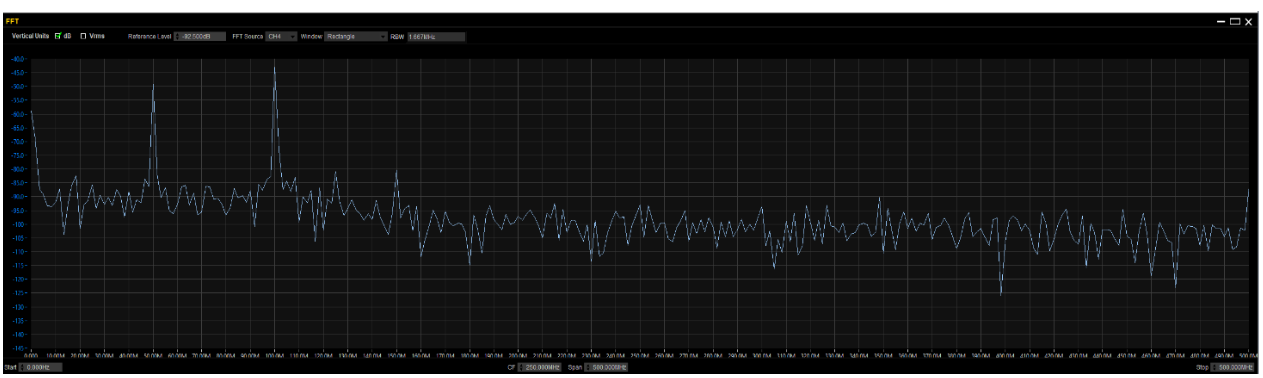

Fig. 6. The Spectrum of the AC Part in the Output Without Using LPF. (It is noisy in 50MHz and 100MHz. We believe it is relative to the main clock used by the boards.)

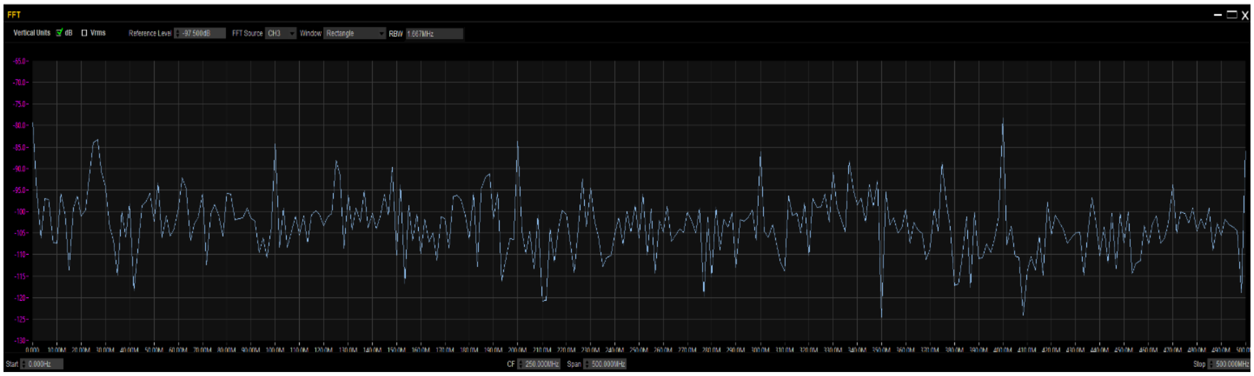

Fig. 7. The Spectrum of the AC Part in the Output Filtered by LPF. The noise in 50MHz and 100MHz are removed.

Fig. 8 and Fig. 9 are the spectrum graph of the AC part of the output with a mean voltage set at 3.5V with the low-pass filter, spectrum range 0~50MHz and 0~1.4MHz respectively.

Fig. 8. The Spectrum (0 ~ 50 MHz) of the AC Part in the Output (with LPF).

Fig. 9. The Spectrum (0 ~ 1.4 MHz) of the AC Part in the Output (with LPF).

4, Found Problems and Solutions

A Python version compatibility problem has been found.

By now, Opal Kelly only offers FrontPanel API for Python 3.7. Our XEM3001_AD5676R_DAC.py module is based this API. It is fully compatible to Python 3.7.

If you use Python 3.8 or later version, you might get a Python error when running “import ok”, complaining that “ok libraries can’t be found”. It is because Python has changed the method to locate libraries from version 3.8.

The simplest solution for this problem is to copy all the files under "<FP_root>\API\lib\x64" to the directory "<FP_root>\API\Python\3.7\x64". If it still does not work, you can try to copy all the files under these 2 directories to your working directory (where you run your own codes with XEM3001_AD5676R_DAC.py and ad5676r_dac.bit files).

5, Appendix

PartsLists.xlsx