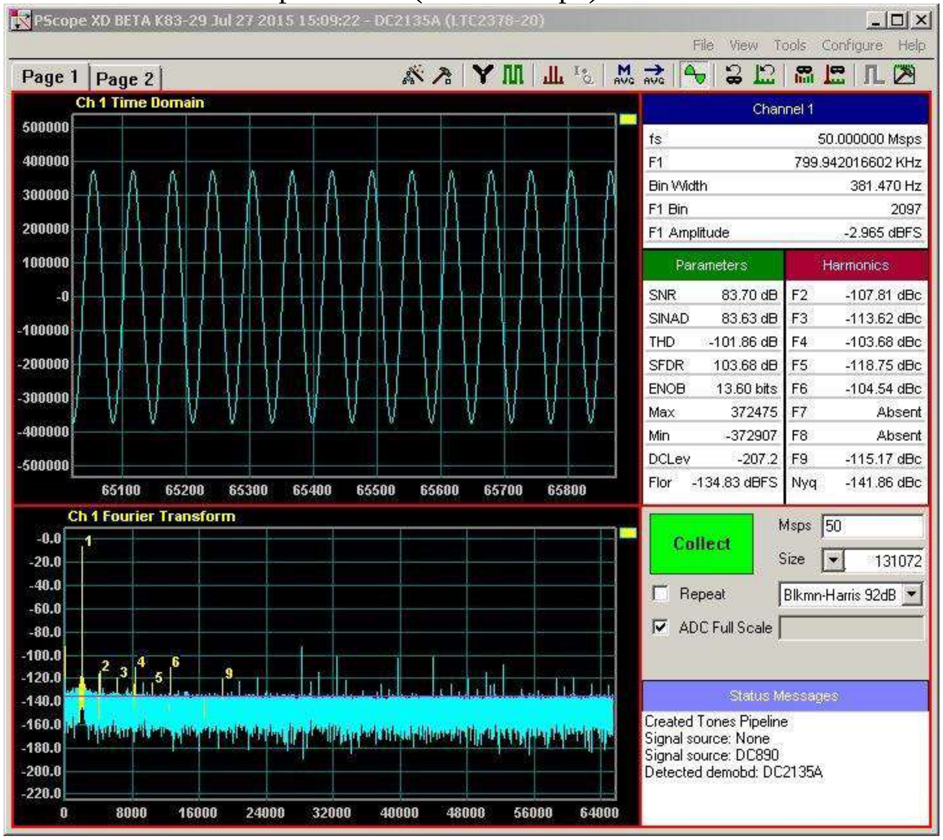
LinearLabTools Python Install Instructions

LinearLabTools Step-by-step installation for Python users. Updated August, 2020

Currently LinearLabTools supports Demo Circuits for legacy LTC parts only. For other Analog Devices parts similar functionality can be achieved in ACE if it supports the part.

Demo-board Setup

- Follow the instructions in the Demo Manual or Evaluation Kit Manual for the demo-board.
    - This will include downloading software such as PScope or LTDACGen depending on the part.
        - If the instructions mention QuikEval or some other software, it is unlikely that it is supported by LinearLabTools.
    - It will also include proper hardware setup including clock and power connections.
- Once the Hardware is setup properly, the software will be able to collect or generate data as shown in the example below (from PScope).



Channel 1	
fs	50.000000 Msps
F1	799.942016602 KHz
Bin Width	381.470 Hz
F1 Bin	2097
F1 Amplitude	-2.965 dBFS

Parameters		Harmonics	
SNR	83.70 dB	F2	-107.81 dBc
SINAD	83.63 dB	F3	-113.62 dBc
THD	-101.86 dB	F4	-103.68 dBc
SFDR	103.68 dB	F5	-118.75 dBc
ENOB	13.60 bits	F6	-104.54 dBc
Max	372475	F7	Absent
Min	-372907	F8	Absent
DCLev	-207.2	F9	-115.17 dBc
Flor	-134.83 dBFS	Nyq	-141.86 dBc

Msps 50
Size 131072
□ Repeat
Blkmn-Harris 92dB
☑ ADC Full Scale

Status Messages

```
Created Tones Pipeline
Signal source: None
Signal source: DC890
Detected demobd: DC2135A
```

- **NOTE:** If you are having errors in PScope due to hardware issues, you will get similar errors in Linear Lab Tools.
- Once the hardware is working properly and communicating with the host computer, quit the software before proceeding to avoid communication conflicts with LinearLabTools programs.

Installing Python

- LinearLabTools still requires Python 2.7. We recommend the Anaconda distribution.
  - If you don't use anaconda you will have to install some packages such as NumPy and MatPlotLib
- For 32-bit systems, download the 32-bit installer. For 64-bit systems, the 32-bit or the 64-bit
  installer can be used.
- It can be tricky to get to Python 2.7, here is how I did it.
  - Install Anaconda
  - Start typing "anaconda" in the search bar and click "Anaconda Prompt (Anaconda 3)"
  - In the console window enter the following commands:
    - conda create --name py2 python=2.7
    - conda activate py2
    - conda install spyder

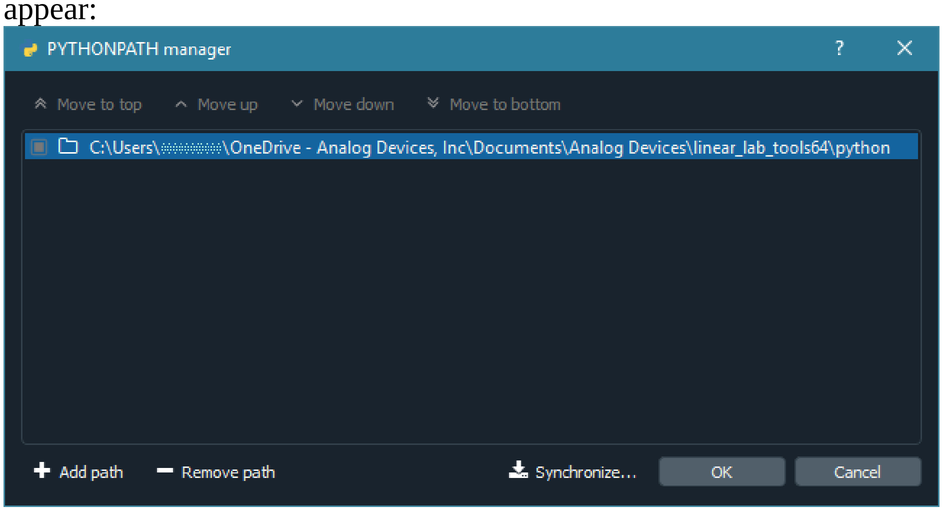
Installing LinearLabTools

- Note that both 32-bit and 64-bit installers are provided.
  - The 32-bit installer is only for 32-bit systems.
  - The 64-bit installer includes everything needed to use LinearLabTools with 64-bit Python, as well as 32-bit Python on 64-bit computers.
- Run the installer and follow the directions.
- Pay attention to where it gets installed, usually that is under Analog Devices in your Documents folder.

Preparing The Environment

- The Anaconda distribution includes a few Integrated Development Environments (IDEs) and editors. We will be using the Spyder IDE.
- Run Spyder, which will be in the Anaconda program group.
  - If you installed Python 2.7 using conda as described above, you will have 2 Spyder installs, you want "Spyder (py2)"
- It’s a big program, this takes a bit of time.

- Once Spyder is open, click Tools → PYTHONPATH Manager. The following dialog will appear:

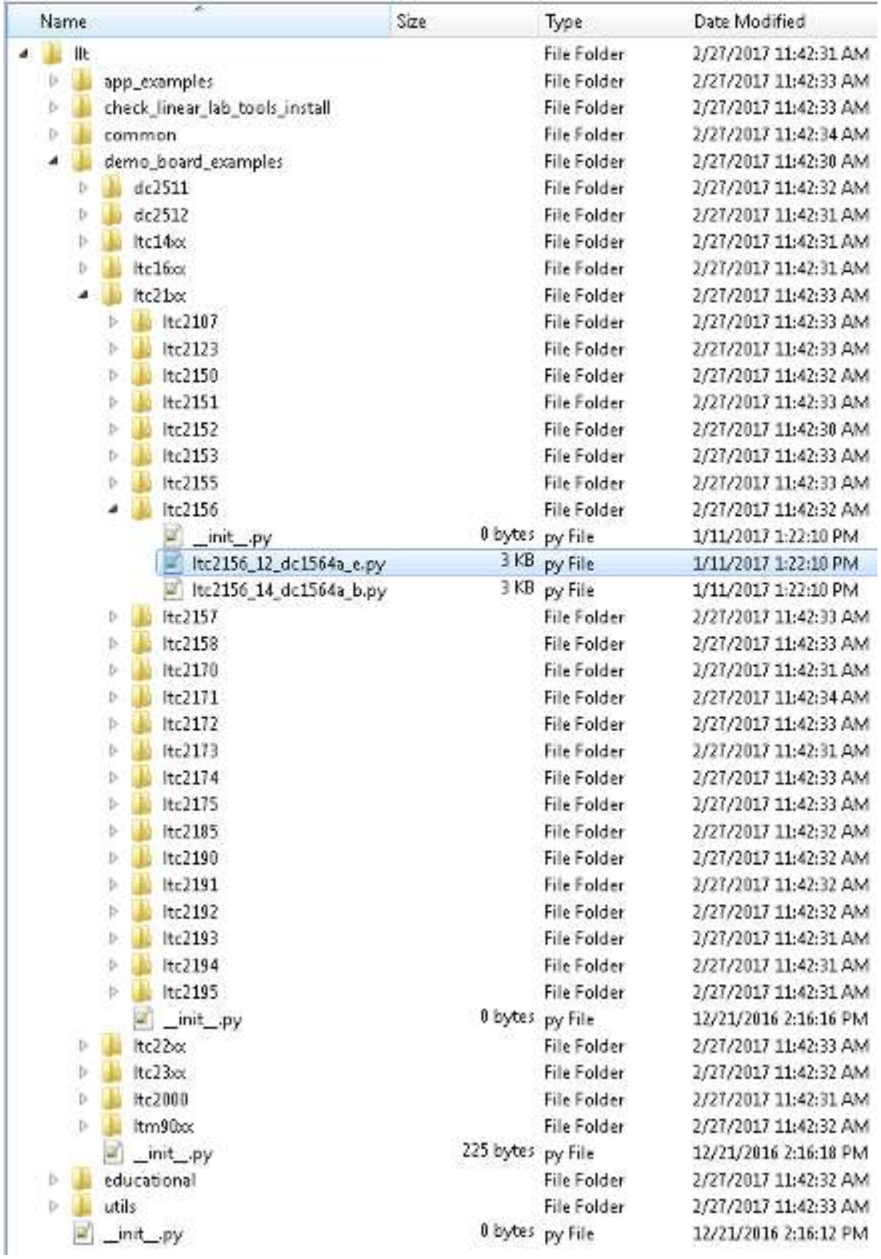


C:\Users\[redacted]\OneDrive - Analog Devices, Inc\Documents\Analog Devices\linear_lab_tools64\python

- Click “Add path”, and navigate to the folder where you installed LinearLabTools, and select the
python subdirectory. Click Close.

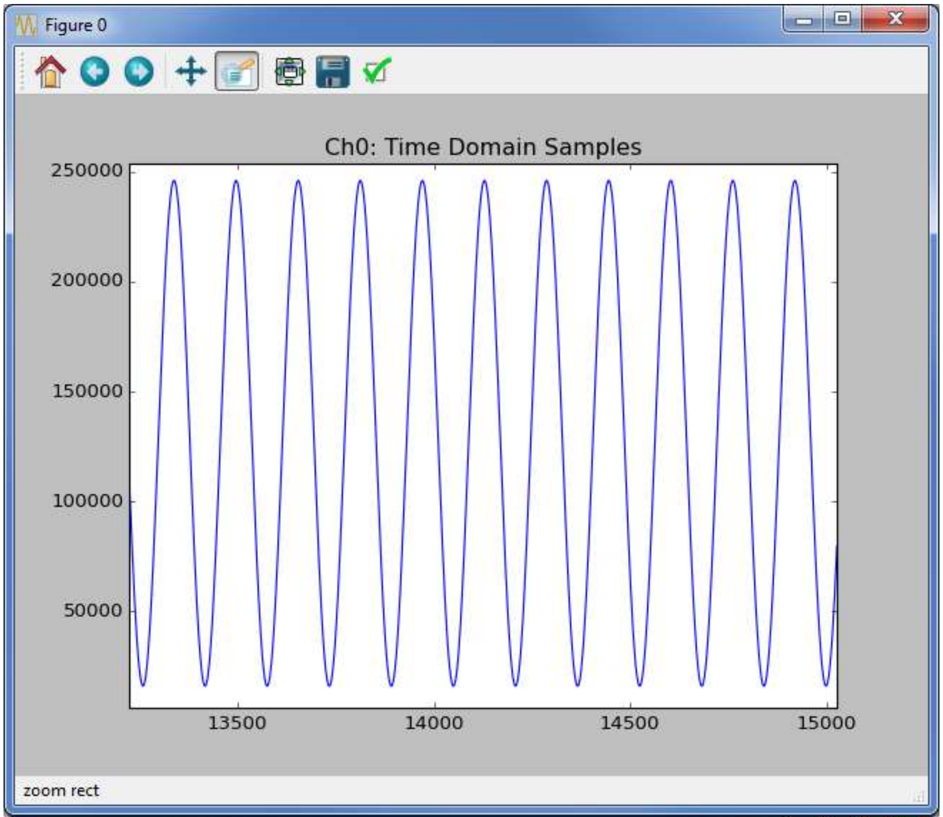
Communicating with the Hardware

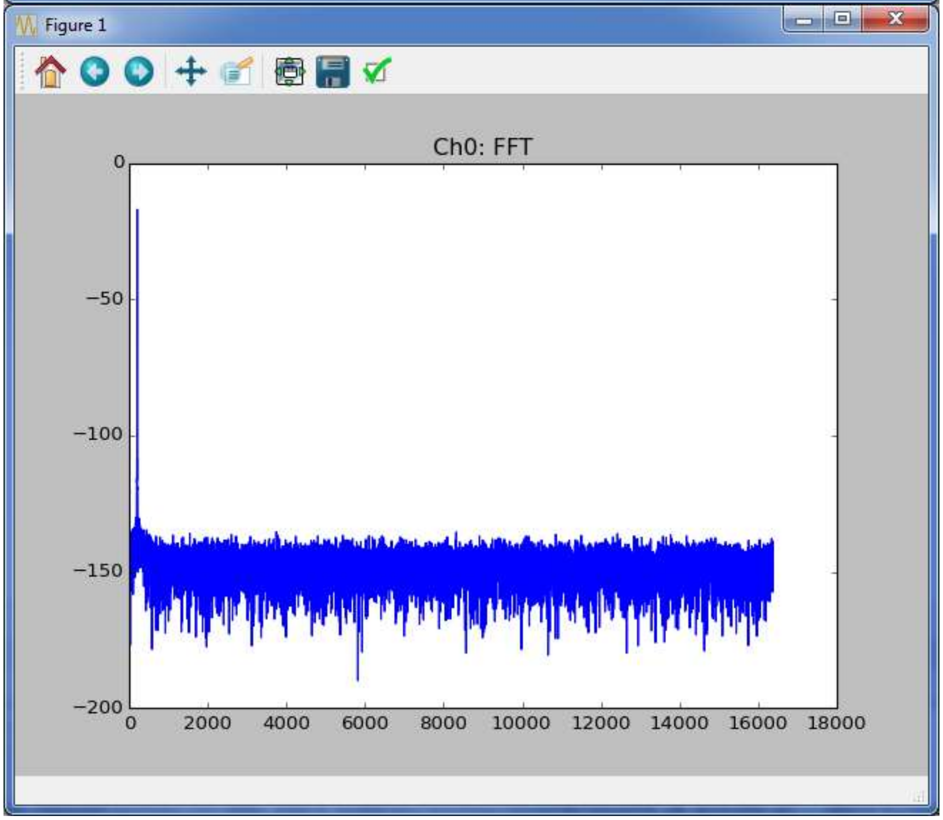
- The figure below shows the organization of LinearLabTools:



Name	Size	Type	Date Modified
llt		File Folder	2/27/2017 11:42:31 AM
app_examples		File Folder	2/27/2017 11:42:33 AM
check_linear_lab_tools_install		File Folder	2/27/2017 11:42:33 AM
common		File Folder	2/27/2017 11:42:34 AM
demo_board_examples		File Folder	2/27/2017 11:42:30 AM
dc2511		File Folder	2/27/2017 11:42:32 AM
dc2512		File Folder	2/27/2017 11:42:31 AM
ltc14xx		File Folder	2/27/2017 11:42:31 AM
ltc16xx		File Folder	2/27/2017 11:42:31 AM
ltc21xx		File Folder	2/27/2017 11:42:33 AM
ltc2107		File Folder	2/27/2017 11:42:33 AM
ltc2123		File Folder	2/27/2017 11:42:33 AM
ltc2150		File Folder	2/27/2017 11:42:32 AM
ltc2151		File Folder	2/27/2017 11:42:33 AM
ltc2152		File Folder	2/27/2017 11:42:30 AM
ltc2153		File Folder	2/27/2017 11:42:33 AM
ltc2155		File Folder	2/27/2017 11:42:33 AM
ltc2156		File Folder	2/27/2017 11:42:32 AM
__init__.py	0 bytes	py File	1/11/2017 1:22:10 PM
ltc2156_12_dc1564a_e.py	3 KB	py File	1/11/2017 1:22:10 PM
ltc2156_14_dc1564a_b.py	3 KB	py File	1/11/2017 1:22:10 PM
ltc2157		File Folder	2/27/2017 11:42:33 AM
ltc2158		File Folder	2/27/2017 11:42:33 AM
ltc2170		File Folder	2/27/2017 11:42:31 AM
ltc2171		File Folder	2/27/2017 11:42:34 AM
ltc2172		File Folder	2/27/2017 11:42:33 AM
ltc2173		File Folder	2/27/2017 11:42:31 AM
ltc2174		File Folder	2/27/2017 11:42:33 AM
ltc2175		File Folder	2/27/2017 11:42:33 AM
ltc2185		File Folder	2/27/2017 11:42:32 AM
ltc2190		File Folder	2/27/2017 11:42:32 AM
ltc2191		File Folder	2/27/2017 11:42:32 AM
ltc2192		File Folder	2/27/2017 11:42:32 AM
ltc2193		File Folder	2/27/2017 11:42:31 AM
ltc2194		File Folder	2/27/2017 11:42:31 AM
ltc2195		File Folder	2/27/2017 11:42:31 AM
__init__.py	0 bytes	py File	12/21/2016 2:16:16 PM
ltc22xx		File Folder	2/27/2017 11:42:33 AM
ltc23xx		File Folder	2/27/2017 11:42:32 AM
ltc2000		File Folder	2/27/2017 11:42:31 AM
ltm90xx		File Folder	2/27/2017 11:42:32 AM
__init__.py	225 bytes	py File	12/21/2016 2:16:18 PM
educational		File Folder	2/27/2017 11:42:32 AM
utils		File Folder	2/27/2017 11:42:33 AM
__init__.py	0 bytes	py File	12/21/2016 2:16:12 PM

- To run the example Python script for your demo-board, open the desired demo-board script and hit run:
  - e.g. Open llt → demo_board_examples → ltc23xx → ltc2378 → ltc2378_20_dc2135a And run the script.
- The script will go through the basic operations of capturing data from the board, then display time and frequency domain plots.
- Exact operations may vary from board to board. You should see plots similar to those below:





- When run as above, each demo-board example makes a time domain plot and a frequency domain plot for each channel and writes the data to a text file.

- You can also call the function directly passing it several parameters and returning the data for each channel.
    - For example:

      Calling collect function from Python

      ```
      from llt.demo_board_examples.ltc23xx.ltc2378.ltc2378_20_dc2135a
      import ltc2378_20_dc2135a
      data = ltc2378_20_dc2135a(num_samples=16*1024, spi_registers=[],
      is_verbose=false,
      do_plot=true, do_write_to_file=false);
      ```

- Most functions have a signature similar to the one above. See the code for additional information.
- Many parts do not have SPI configuration, for these pass [] for the SPI registers.
- For other parts, look at the code for an example of correct SPI register format.
- For parts with multiple channels replace data with something like ch0, ch1, … chn for the function output.
- At this point, data from the demo board is stored in an array.
    - You can extend the functionality of the script as required for your evaluation.
        - Incorporate other test hardware such as signal generators, etc.
    - You can also call the function from your existing Python code. (Just remember to add the imports.)