LinearLabTools Step-by-step installation for Python users

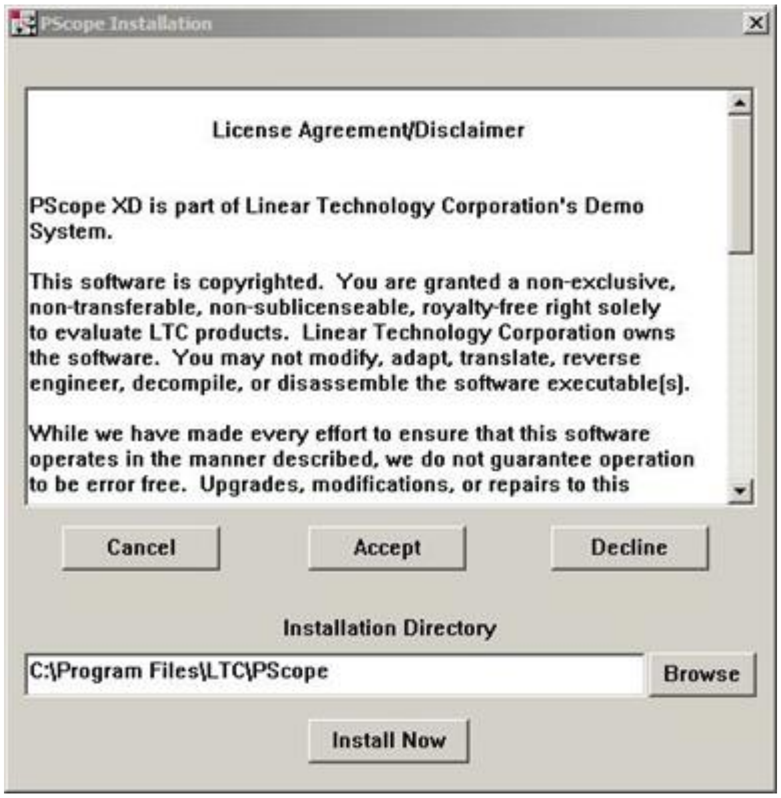
February, 2017

This document shows all steps for the installation of LinearLabTools for Matlab users, including the installation of PScope.

Step 1: Linear Tech converter evaluation software installation

Install the software required by the demo board being evaluated. This is PScope for Analog to Digital converters, or LTDACgen for high-speed DACs such as the LTC2000. This document will use PScope as an example. The procedure for LTDACgen is similar.

Run the installer and follow the directions:



License Agreement/Disclaimer

PScope XD is part of Linear Technology Corporation's Demo System.

This software is copyrighted. You are granted a non-exclusive, non-transferable, non-sublicenseable, royalty-free right solely to evaluate LTC products. Linear Technology Corporation owns the software. You may not modify, adapt, translate, reverse engineer, decompile, or disassemble the software executable(s).

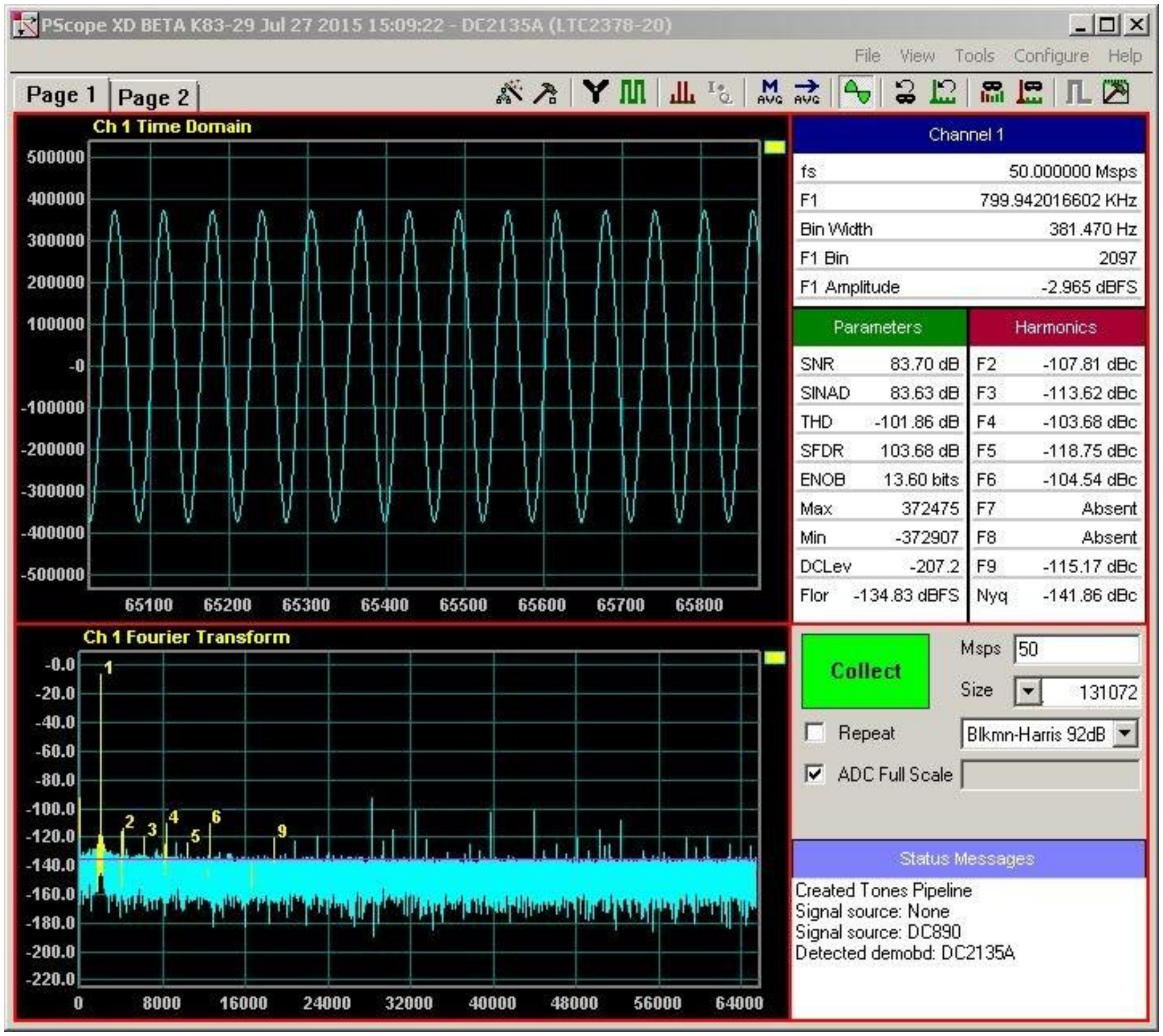
While we have made every effort to ensure that this software operates in the manner described, we do not guarantee operation to be error free. Upgrades, modifications, or repairs to this

C:\Program Files\LTC\PScope

PScope will then open. Follow the procedure in the data converter evaluation board's Demo Manual. This will include board configuration, connection of power supplies, clock signals, analog signals, and any other requirements.

IMPORTANT: Both PScope and LinearLabTools programs will show errors if the demo boards are not connected properly.

Once the hardware is set up properly, PScope should be able to collect data properly as shown below.



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

At this point the hardware is working properly and communicating with the host computer. Quit PScope before proceeding to avoid communication conflicts with LinearLabTools programs.

Step 2: Python Installation

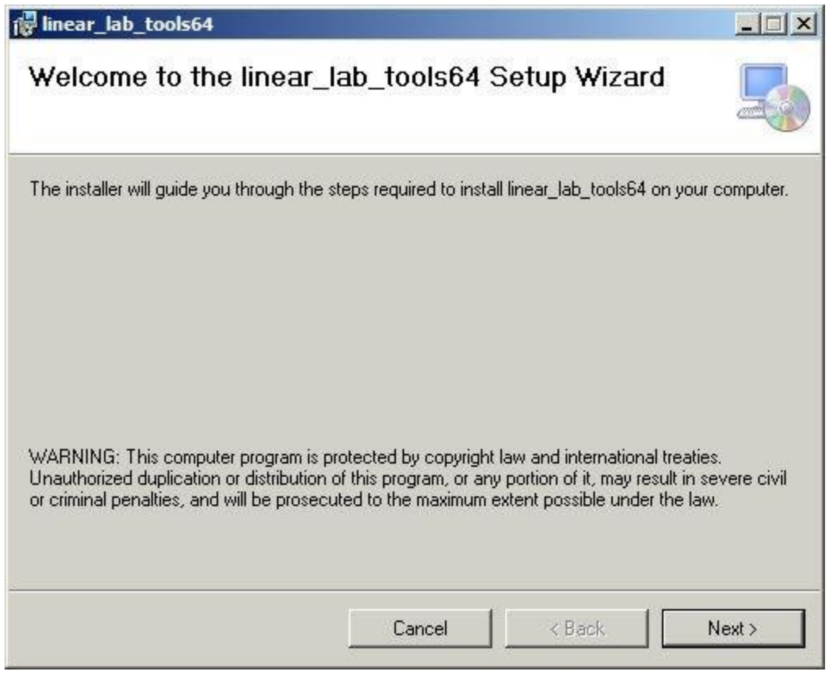
LinearLabTools requires Python 2.7. It is recommended that the Anaconda distribution from Continuum Analytics:
https://www.continuum.io/downloads
For 32-bit systems, download the 32-bit installer. For 64-bit systems, the 32-bit or the 64-bit installer can be used.

Step 3: Install LinearLabTools

Note that both 32-bit and 64-bit installers are provided. The choice depends on the Python environment. If using a 32-bit Python on a 64-bit system, use the 32-bit installer.

32-bit installer
64-bit installer

Run the installer and follow the directions.



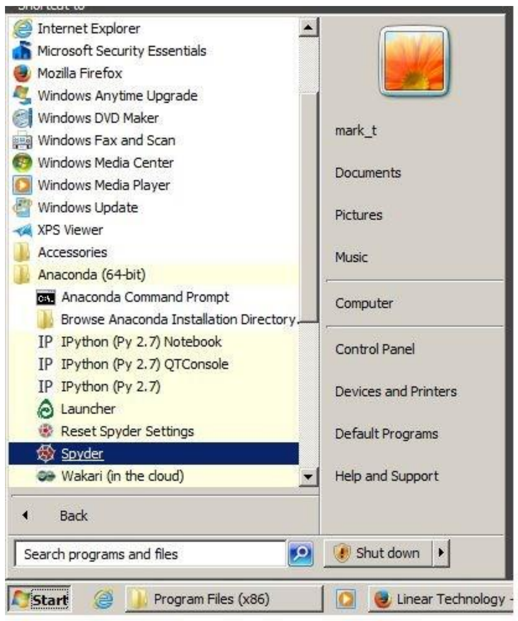
Welcome to the linear_lab_tools64 Setup Wizard

The installer will guide you through the steps required to install linear_lab_tools64 on your computer.

WARNING: This computer program is protected by copyright law and international treaties. Unauthorized duplication or distribution of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Step 4: Prepare The Environment

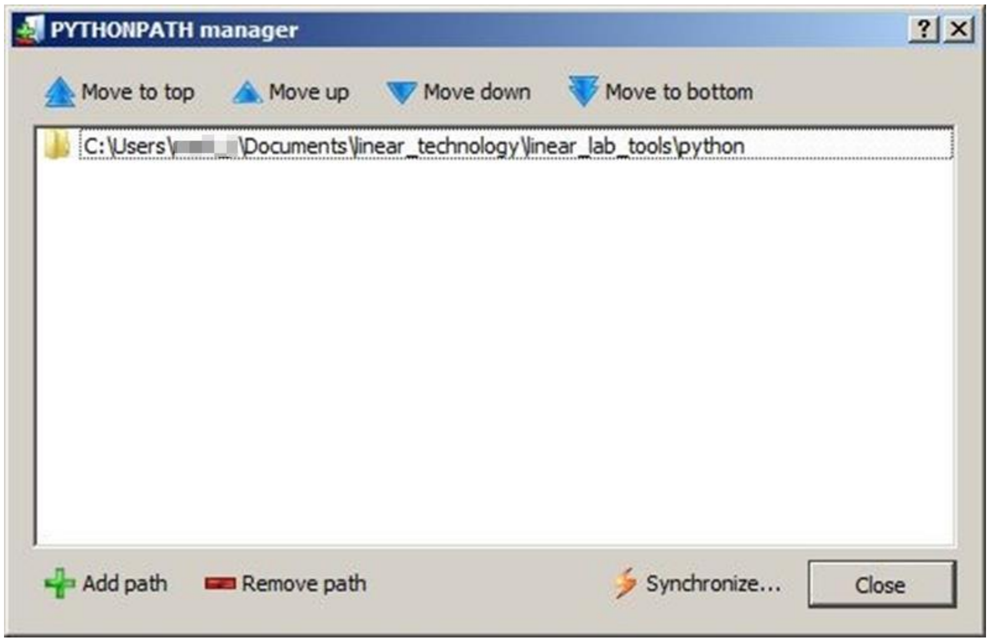
While “Anaconda” is the name of the distribution, the debugging environment is called “Spyder”. Run Spyder, which will be in the Anaconda program group.



It’s a big program, this takes a bit of time. If you see this, things are going well.

Loading external console...

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

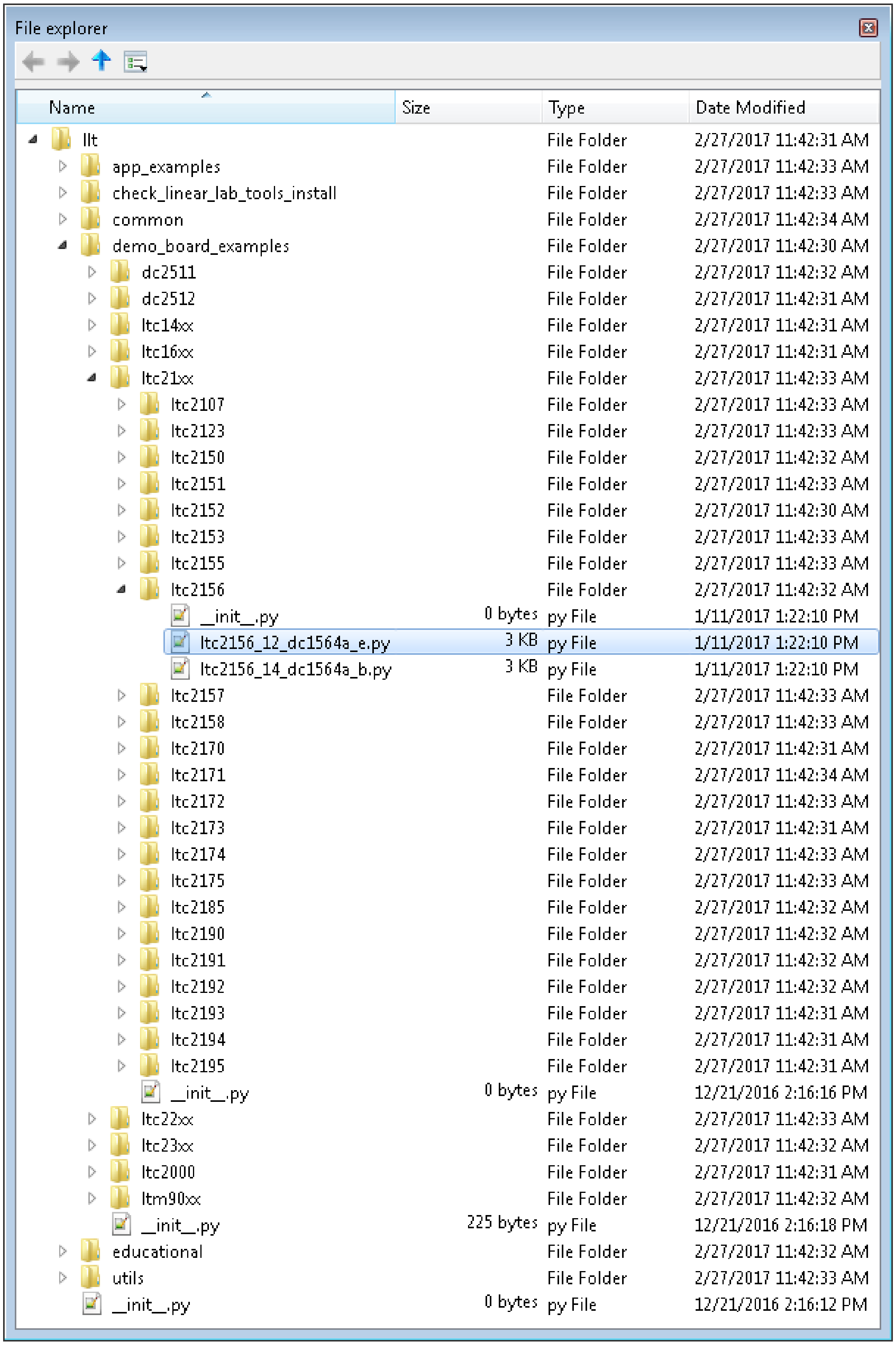


```
C:\Users\[redacted]\Documents\linear_technology\linear_lab_tools\python
```

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

Step 5: Communicating with the Hardware

The figure below shows the organization of LinearLabTools:



File explorer

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

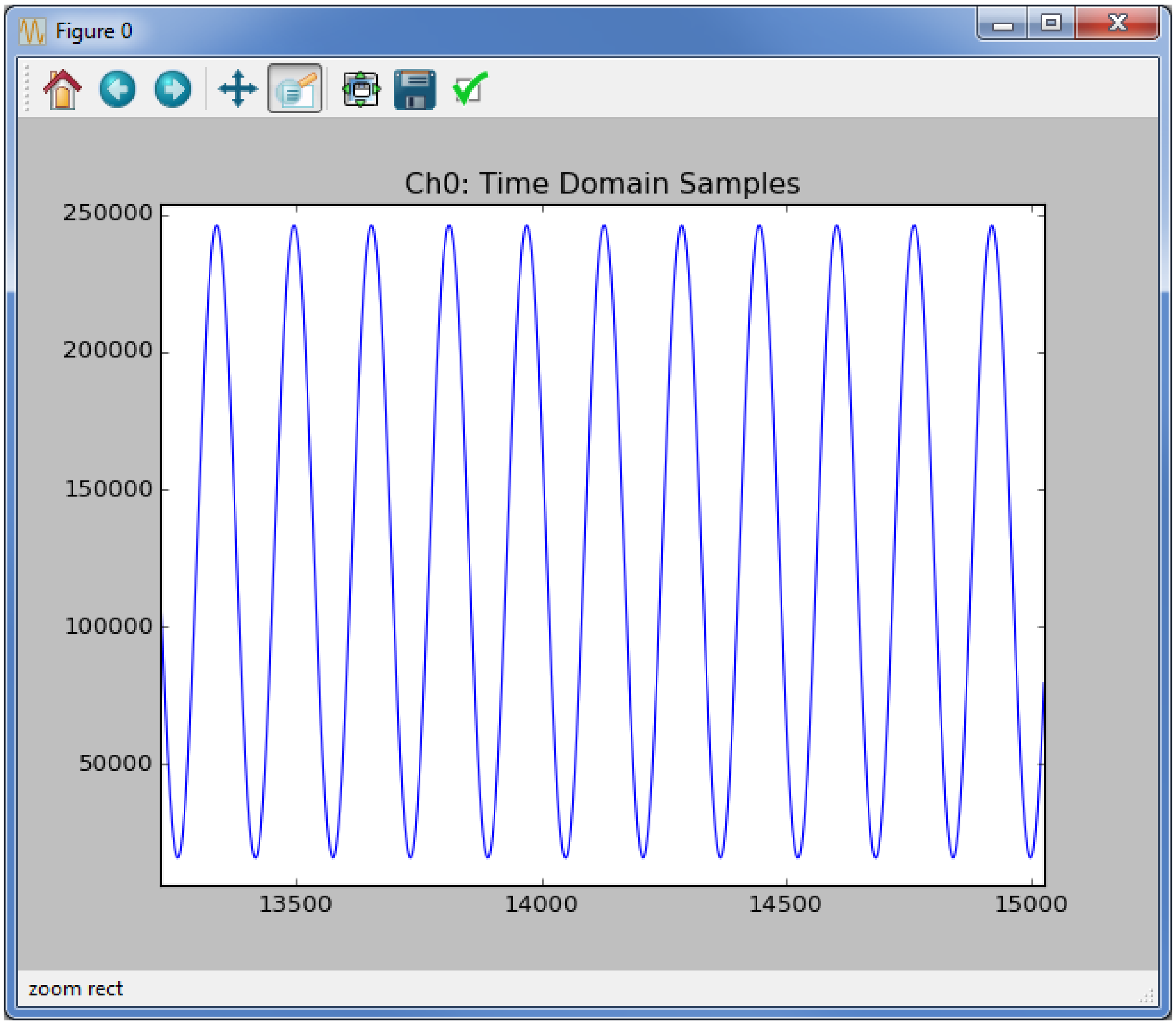
To run the example Python script for your demo board, open the desired demo board script and hit run:

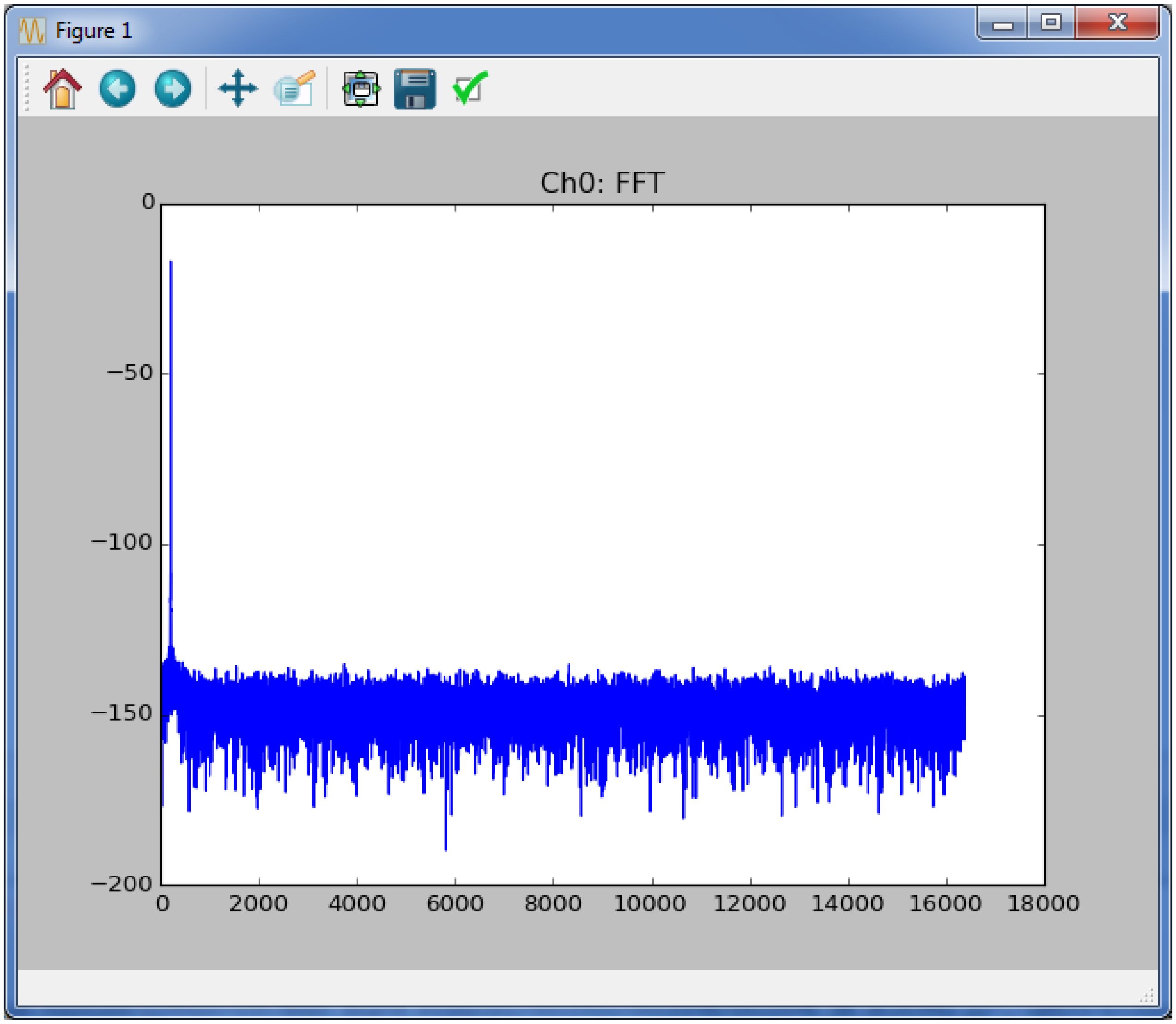
e.g.

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 with 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:

```
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 in the program. You can then extend the functionality of the program as required for your evaluation, incorporate other test hardware such as signal generators, etc.