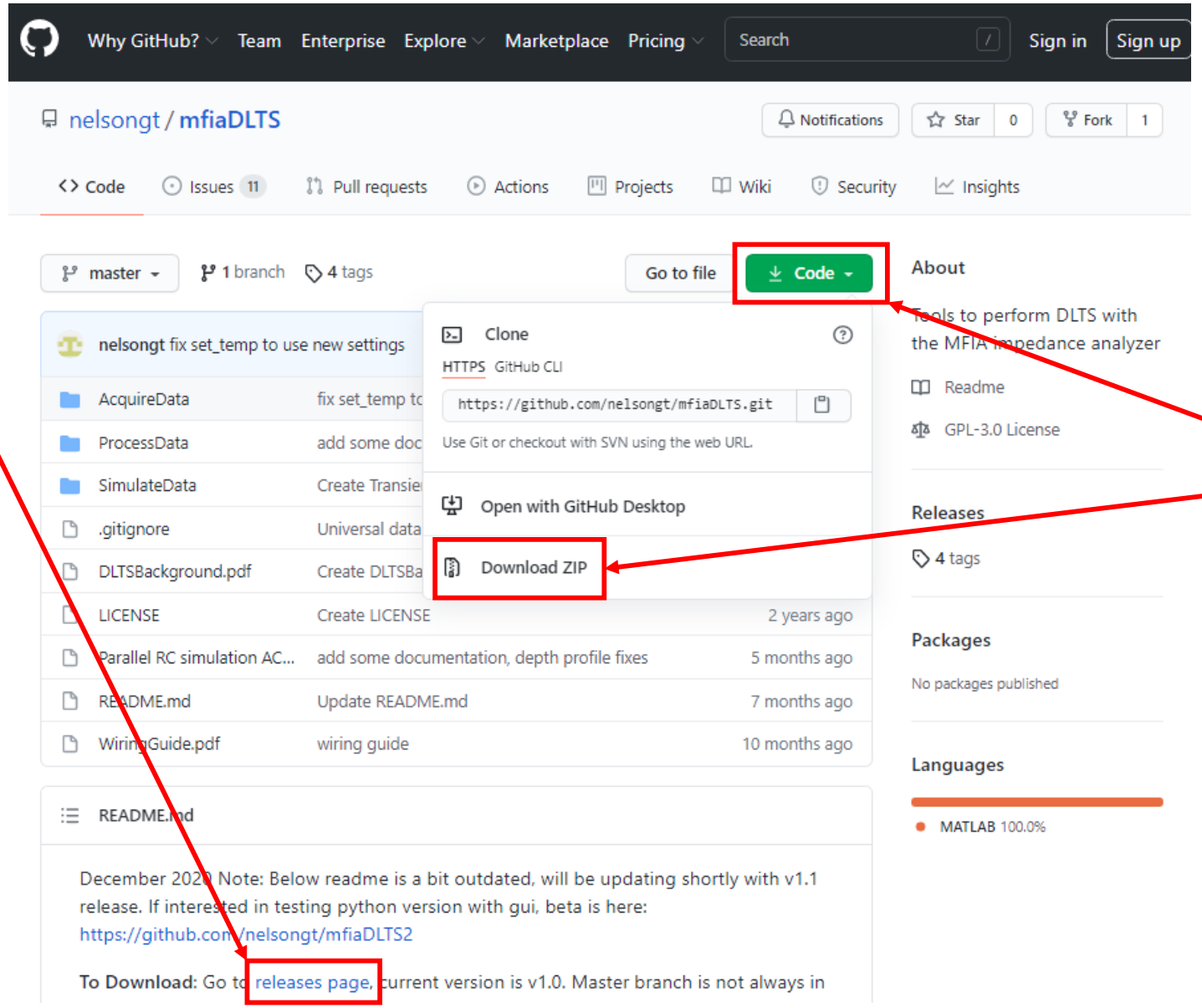
mfiaDLTS Website

<u>To download stable version</u>
Missing features but undergoes testing

1. Click “Releases Page”
2. Click “Assets”
3. Click “Source code (zip)”



<u>To download development version</u>
latest features but some functions may be broken due to bugs

1. Click “Code”
2. Click “Download ZIP”

Hardware Requirements

Requirements

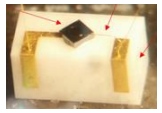
-Device to perform DLTS on

-Cryostat with electrical leads for device connection

-Lakeshore temperature controller to control cryostat temperature

-Zurich Instruments MFIA

-PC with MATLAB to run this code



LN2

Pump

Model 33X w/GPIB

CC Image: https://commons.wikimedia.org/wiki/File:Macbook_Air.jpg

Software Requirements

Software Dependencies

General:

MATLAB (with statistics and instrument control toolboxes)

ProcessData:

~~[optional,not included] ezyfit: http://www.fast.u-psud.fr/ezyfit/~~

DLTS peak fitting, delayed to future version

AcquireData:

[required, not included] LabOne: https://www.zhinst.com/downloads

[required, not included] LabOne Matlab Driver (place in AcquireData folder): https://www.zhinst.com/downloads

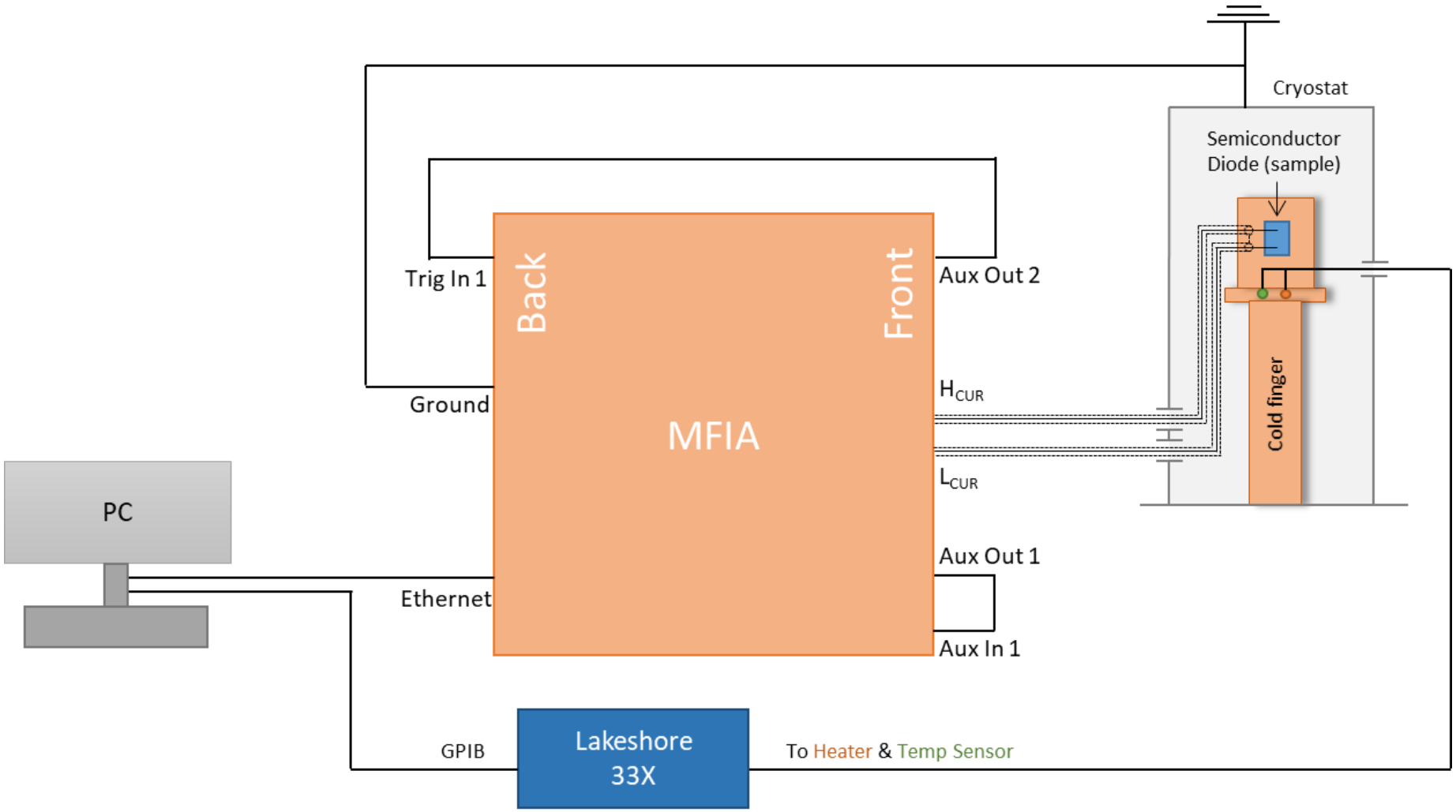
LabOne software from zhinst.com

[required, included] lakshore driver: https://www.mathworks.com/matlabcentral/fileexchange/48366-lakeshore

[required, included] cprintf: https://www.mathworks.com/matlabcentral/fileexchange/24093-cprintf-display-formatted-colored-text-in-the-command-window

Included MATLAB 3rd party code (Here for attribution purposes)

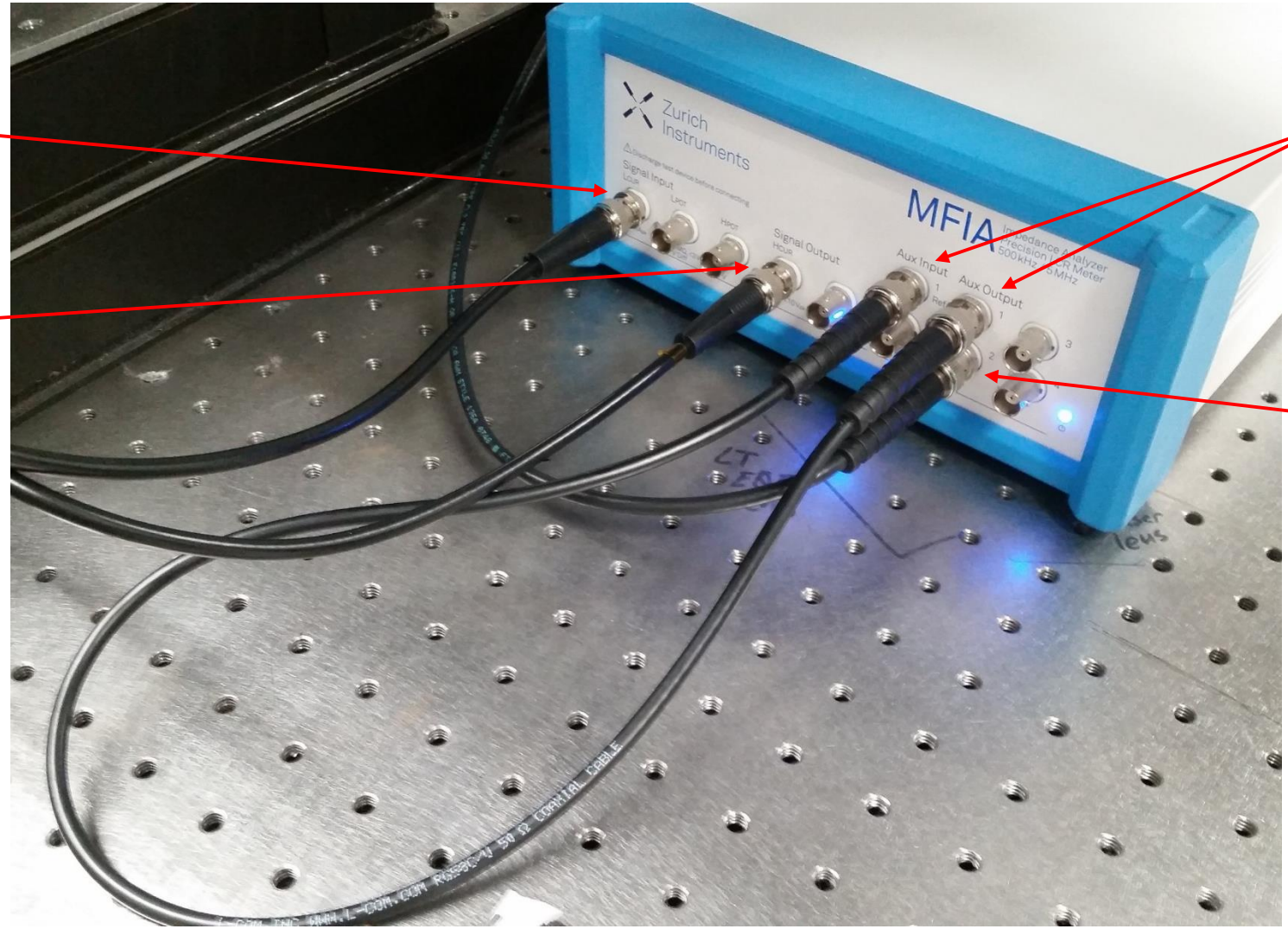
Wiring Block Diagram



Wiring Photos

L_{CUR} to sample
(2 point probe)

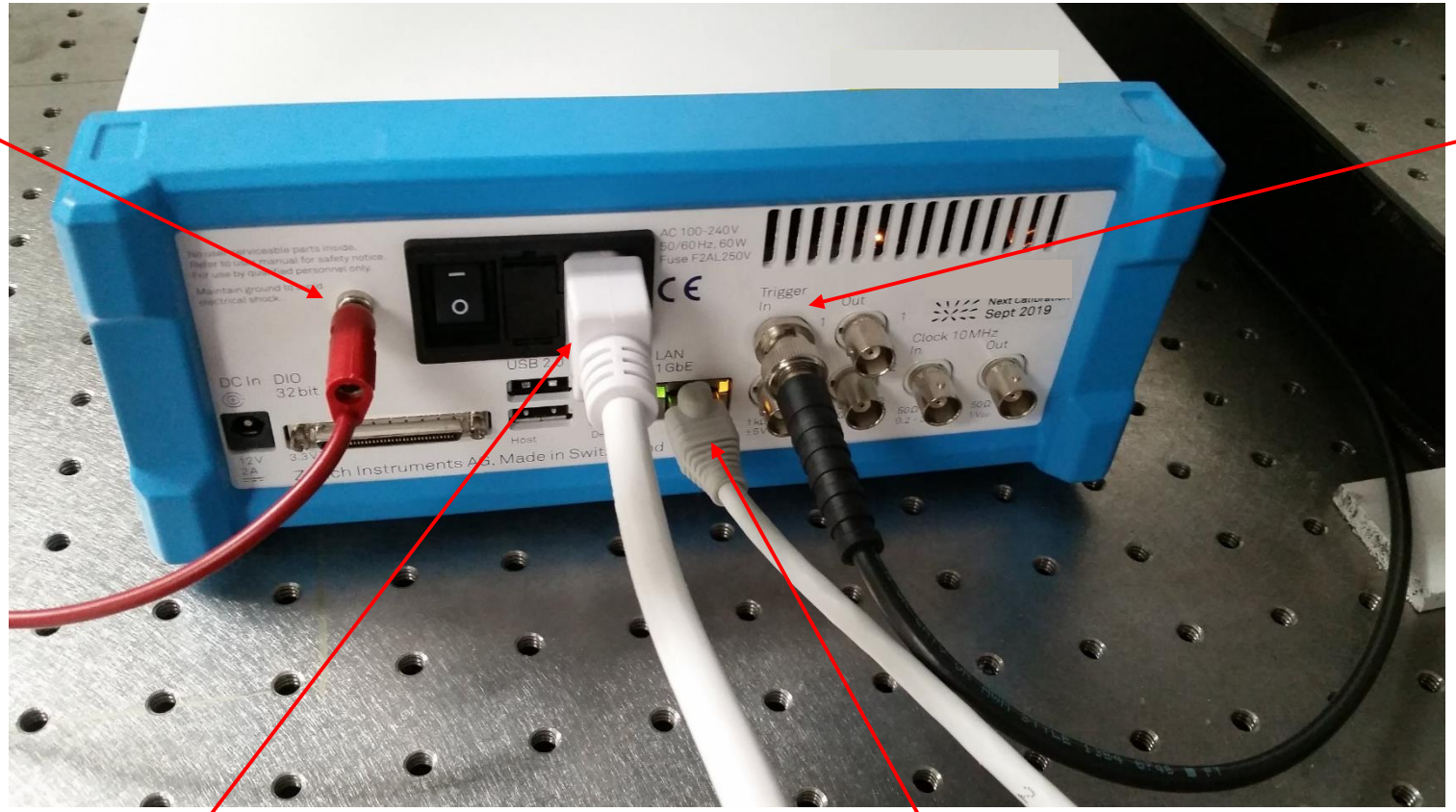
H_{CUR} to sample
(2 point probe)



Aux Out 1 to Aux In 1

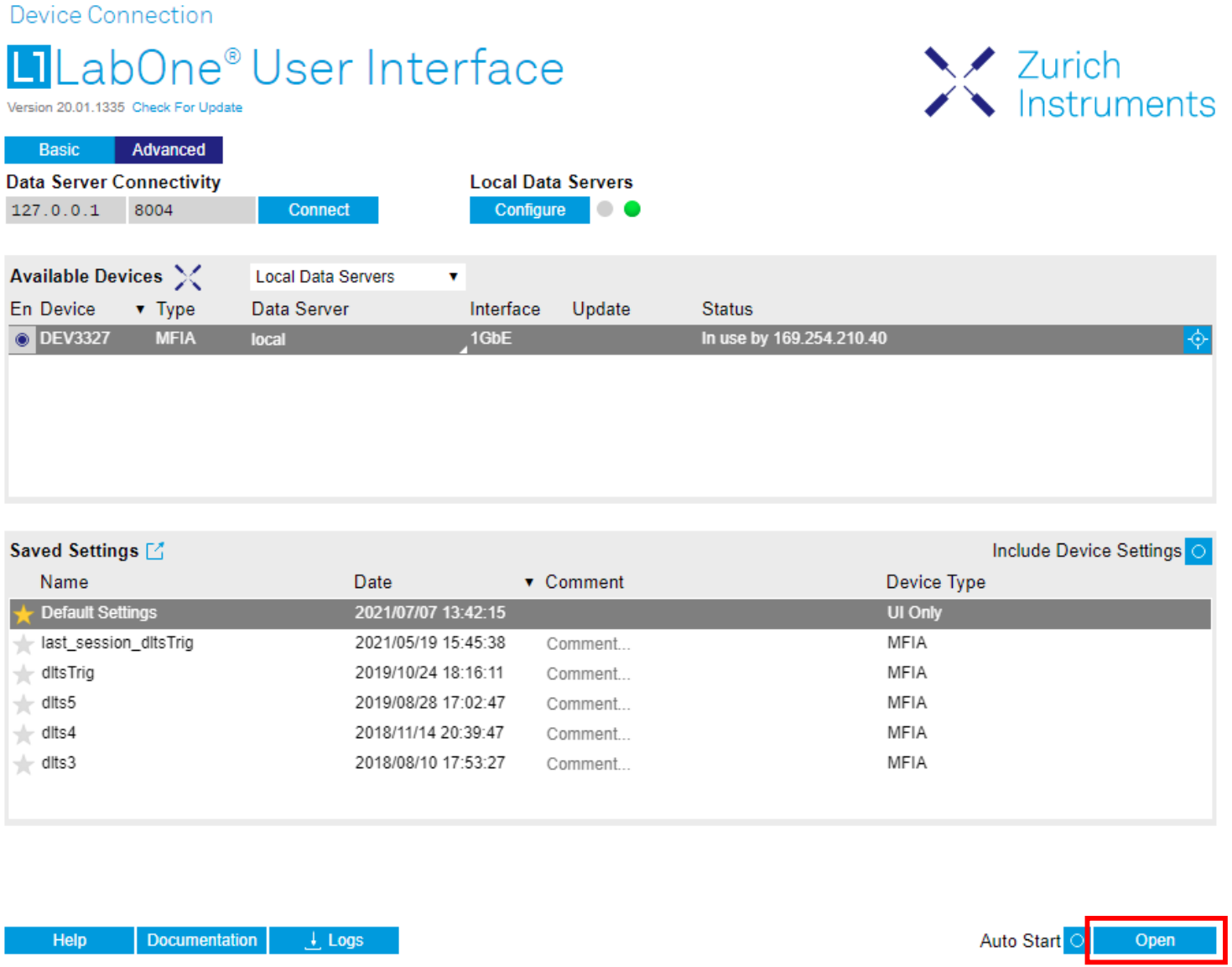
Aux Out 2 to Trig In 1
(on back)

Wiring Photos



Start LabOne local data server

1.

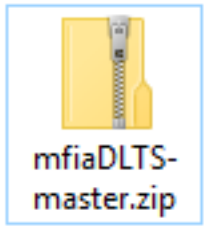


2.

3.

4.

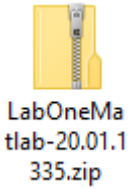
mfiaDLTS Code Organization



- AcquireData → Runs experiment, saves transients data into files
- Data → Data files go here
- ProcessData → Converts transient data into DLTS spectra
- SimulateData → Simulates transient data from DLTS energy, cross-section, density
- .gitignore
- DLTSBackground.pdf → Background on DLTS theory
- LICENSE
- Parallel RC simulation AC vs DC.pdf → Explains current reading in MFIA lock-in tab
- README.md → Readme file
- WiringGuide.pdf

Setup for AcquireData

- AcquireData
- Data
- ProcessData
- SimulateData
- .gitignore
- DLTSBackground.pdf
- LICENSE
- Parallel RC simulation AC vs DC.pdf
- README.md
- WiringGuide.pdf

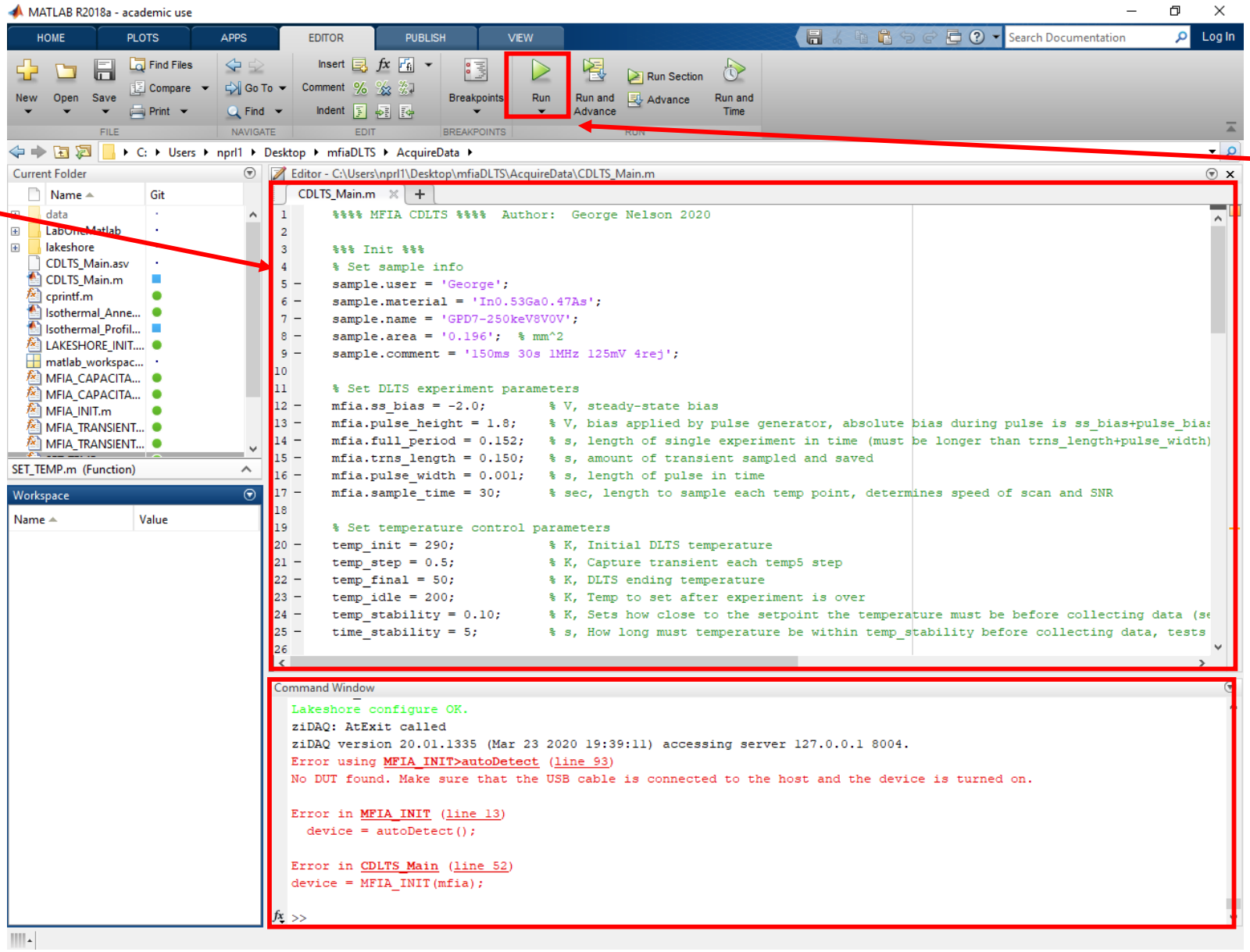
- LabOneMatlab → Extract 

here and name folder “LabOneMatlab”
- lakeshore
- CDLTS_Main.asv
- CDLTS_Main.m → Open this, script for running conventional DTLS experiment
- cprintf.m
- Isothermal_Anneal_Main.m → Automated anneal script
- Isothermal_Profile_Main.m → DLTS vs depth profiling
- LAKESHORE_INIT.m
- matlab_workspace.mat
- MFIA_CAPACITANCE_DAQ.m
- MFIA_CAPACITANCE_POLL.m
- MFIA_INIT.m
- MFIA_TRANSIENT_AVERAGER_DAQ.m
- MFIA_TRANSIENT_AVERAGER_POLL.m
- SET_TEMP.m
- TRANSIENT_FILE.m
- YSPEC_FILE.m
- YSpec_Main.m → Admittance spectroscopy experiment

Viewing CDLTS_Main.m

Script editor shows CDLTS_Main code

We will change experiment parameters here



Run button executes code

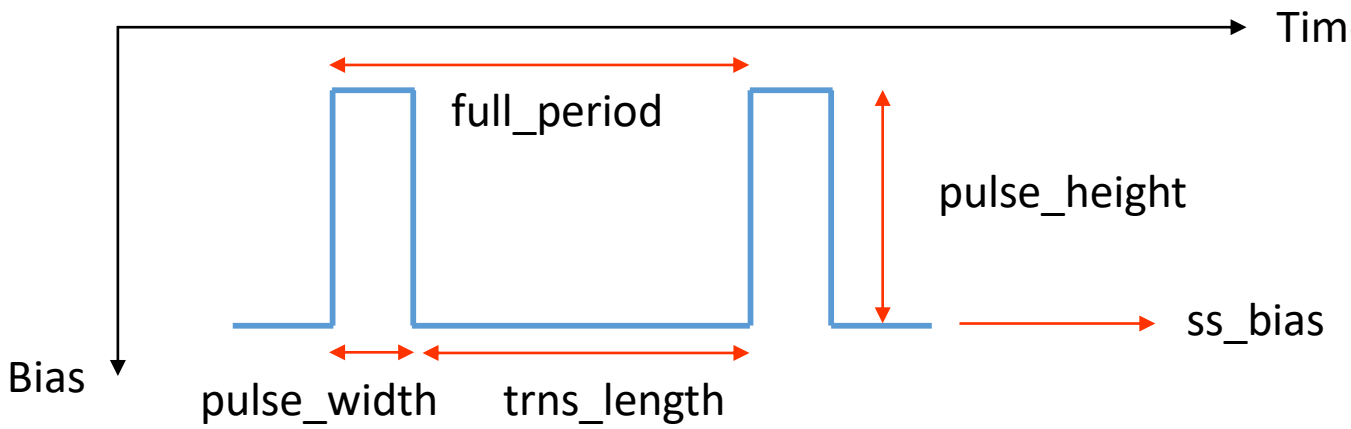
Command window acts as experiment progress log

CDLTS_Main User defined parameters – Sample and DLTS

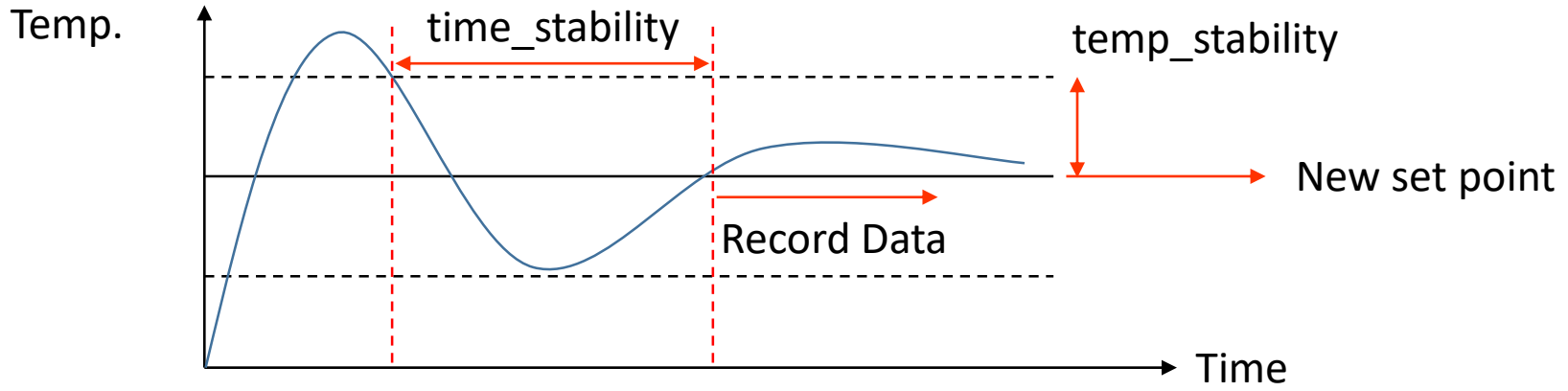
```
%%% Init %%%
% Set sample info
sample.user = 'George';
sample.material = 'In0.53Ga0.47As';
sample.name = 'GPD7-250keV8V0V';
sample.area = '0.196';  % mm^2
sample.comment = '150ms 30s 1MHz 125mV 4rej';
```

Mostly arbitrary, used to describe sample in data files

```
% Set DLTS experiment parameters
mfia.ss_bias = -2.0;          % V, steady-state bias
mfia.pulse_height = 1.8;      % V, bias applied by pulse generator, absolute bias during pulse is ss_bias+pulse_bias
mfia.full_period = 0.152;     % s, length of single experiment in time (must be longer than trns_length+pulse_width)
mfia.trns_length = 0.150;     % s, amount of transient sampled and saved
mfia.pulse_width = 0.001;     % s, length of pulse in time
mfia.sample_time = 30;        % sec, length to sample each temp point, determines speed of scan and SNR
```



CDLTS user defined parameters - Temperature




```
% Set temperature control parameters
temp_init = 290;              % K, Initial DLTS temperature
temp_step = 0.5;              % K, Capture transient each temp. step
temp_final = 50;              % K, DLTS ending temperature
temp_idle = 200;              % K, Temp to set after experiment is over
temp_stability = 0.10;        % K, Sets how close to the setpoint the temperature must be before collecting data
time_stability = 5;           % s, How long must temperature be within temp_stability before collecting data, tes

% Configure Lakeshore Parameters
temp.control = 'B';           % Control sensor (closest to heater), A or B
temp.sample = 'B';            % Measure sensor (closest to sample), A or B
temp.heatpower = 3;           % Heater power range, sets heater to high (3), medium (2), low (1)
```

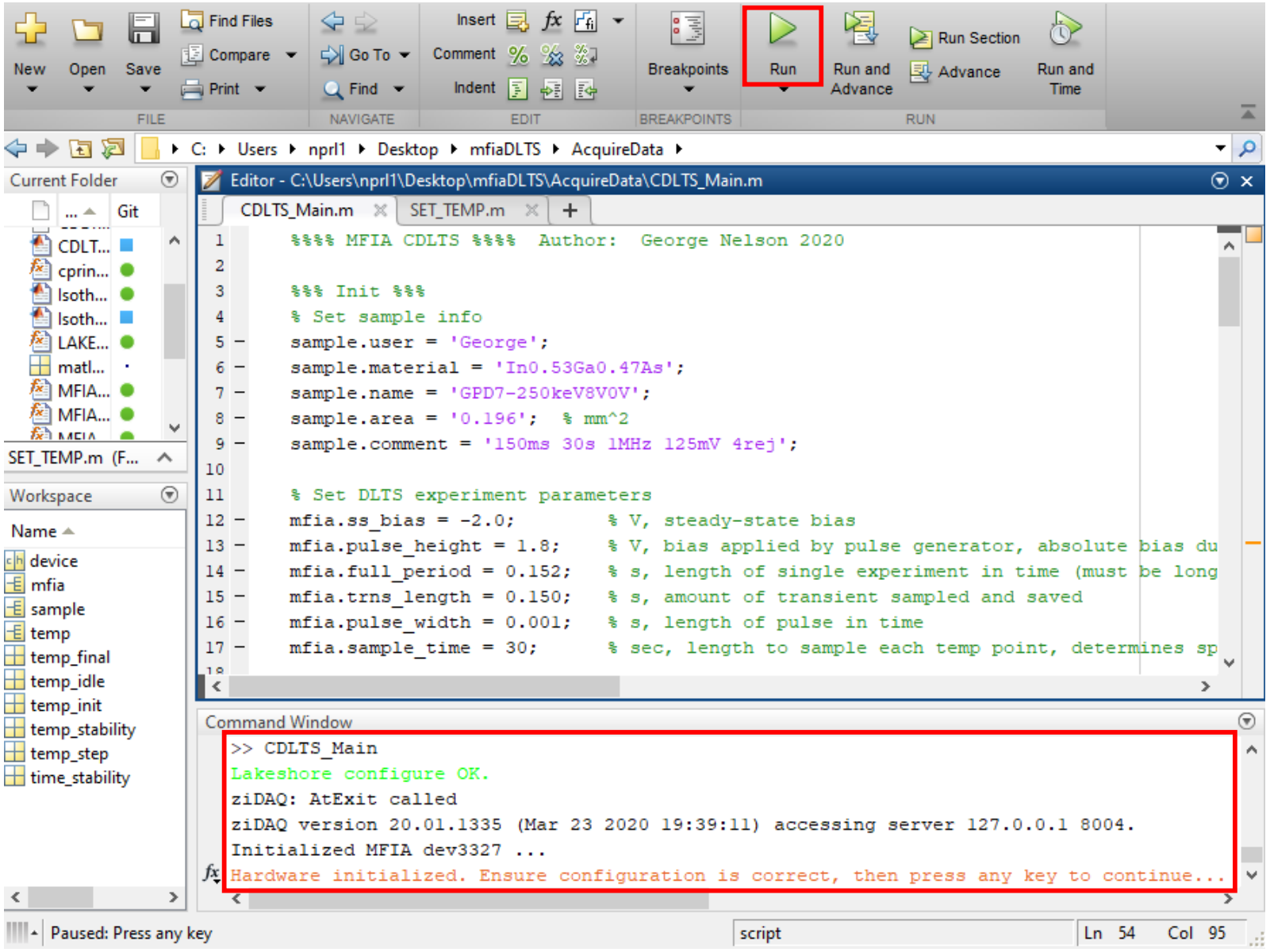
- Only these Lakeshore settings are handled by the code so far
- If you want to change eg. PID parameters you will have to use the Lakeshore front panel manually

CDLTS_Main User defined parameters – Lock-in settings

```
% Configure MFIA Parameters - Advanced Users Only
mfia.time_constant = 2.4e-6; % us, lock in time constant, GN suggests 2.4e-6
mfia.ac_freq = 1.0e6;        % Hz, lock in AC frequency, GN suggests 1MHz
mfia.ac_ampl = 0.125;        % V, lock in AC amplitude, GN suggests ~100 mV for good SNR
mfia.sample_rate = 107143;   % Hz, sampling rate Hz, for CDLTS use 53571 or 107143 or 214286
```

Controls SNR and file sizes, just keep defaults if unsure

Starting CDLTS_Main

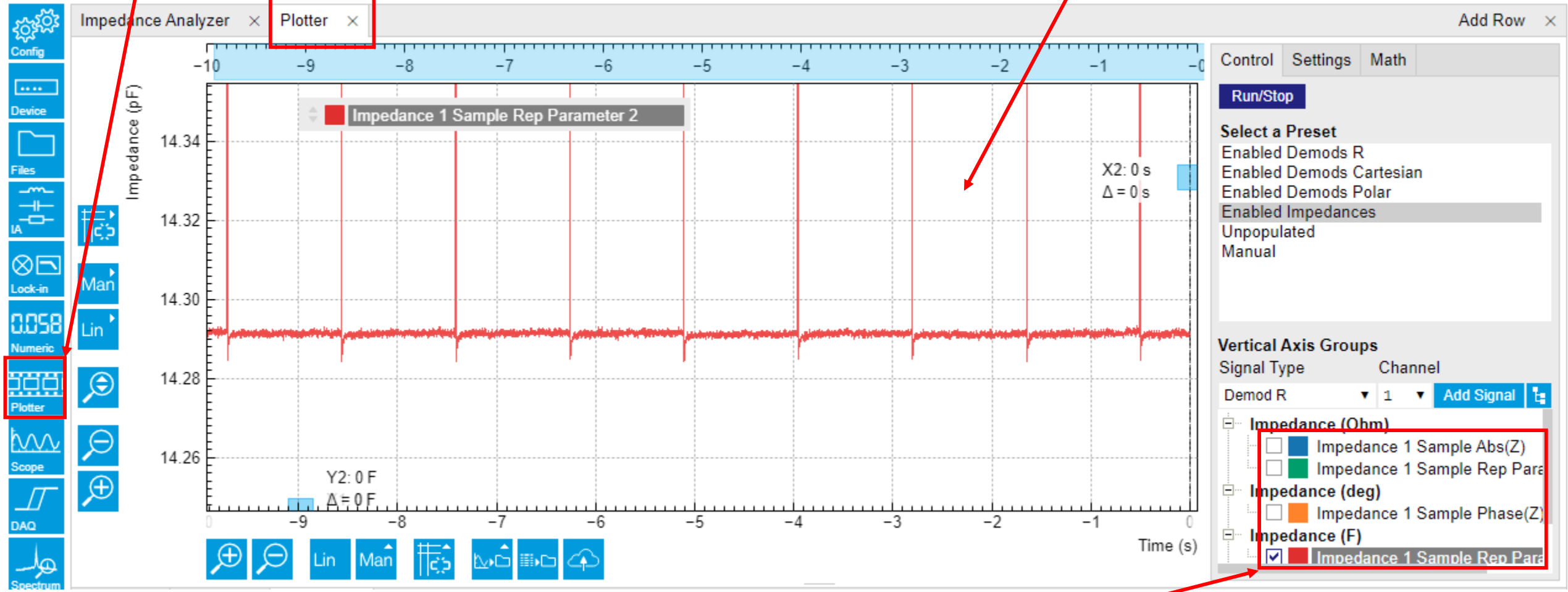


- After setting parameters, hit run
- Then, observe log window for hardware initialization
- If success, program will pause

Check configuration in LabOne Web UI

1. Plotter button opens plotter tab

3. Observe capacitance vs time data, ensure it looks correct



2. Check only parameter with Farad (F) units

Resume CDLTS_Main – typical operation log

```
Hardware initialized. Ensure configuration is correct, then press any key to continue...
Waiting for set point (300.00)...
Wait for time stability: 5 s left.
Wait for time stability: 4 s left.
Wait for time stability: 3 s left.
Wait for time stability: 2 s left.
Wait for time stability: 1 s left.
Wait for time stability: 0 s left.
Temperature has stabilized!
Capturing transient...
Acquired 7 of total 180 transients: 3.9% (elapsed time 2.116 s)
Acquired 20 of total 180 transients: 11.1% (elapsed time 4.249 s)
Acquired 35 of total 180 transients: 19.4% (elapsed time 6.389 s)
Acquired 48 of total 180 transients: 26.7% (elapsed time 8.529 s)
Acquired 63 of total 180 transients: 35.0% (elapsed time 10.680 s)
Acquired 76 of total 180 transients: 43.3% (elapsed time 12.819 s)
Acquired 91 of total 180 transients: 50.6% (elapsed time 14.962 s)
Acquired 105 of total 180 transients: 58.3% (elapsed time 17.104 s)
Acquired 119 of total 180 transients: 66.1% (elapsed time 19.248 s)
Acquired 133 of total 180 transients: 73.9% (elapsed time 21.394 s)
Acquired 148 of total 180 transients: 82.2% (elapsed time 23.552 s)
Acquired 161 of total 180 transients: 89.4% (elapsed time 25.704 s)
Acquired 176 of total 180 transients: 97.8% (elapsed time 27.862 s)
Acquired 180 of total 180 transients: 100.0% (elapsed time 30.004 s)
Done.
Finished transient for this temperature.
Warning: 0.5% data loss detected.
Saving transient...
Waiting for set point (299.50)...
Current Temp: 300.00.  Set point: 2.995000e+02.  Delta: 0.50.
Current Temp: 300.00.  Set point: 2.995000e+02.  Delta: 0.50.
```

Temperature stabilization

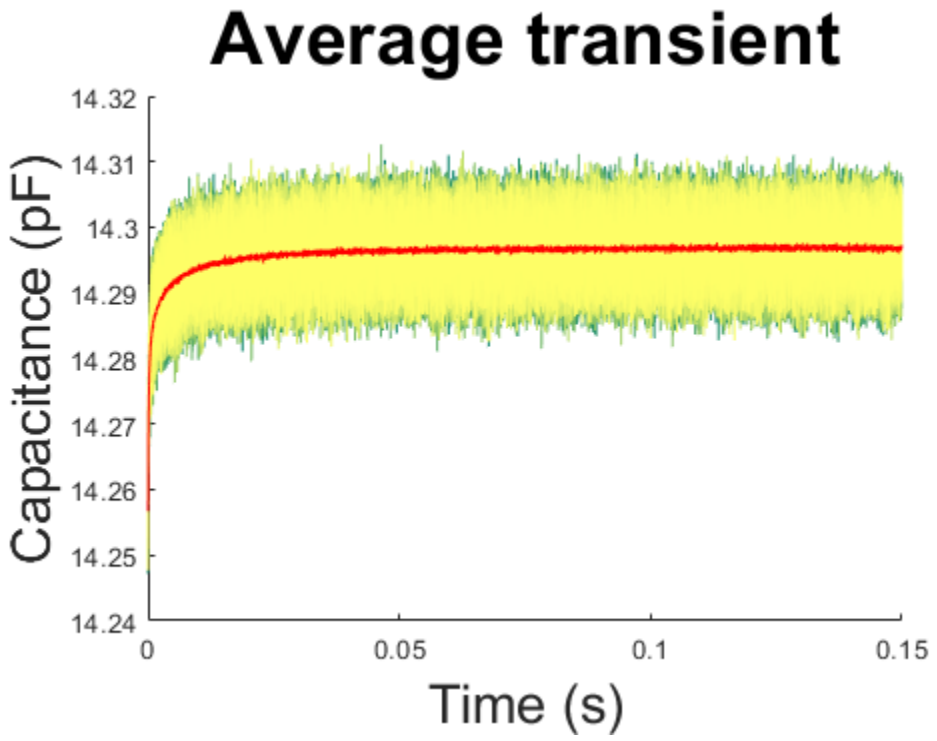
Transient collection

Transient data plot and save file →

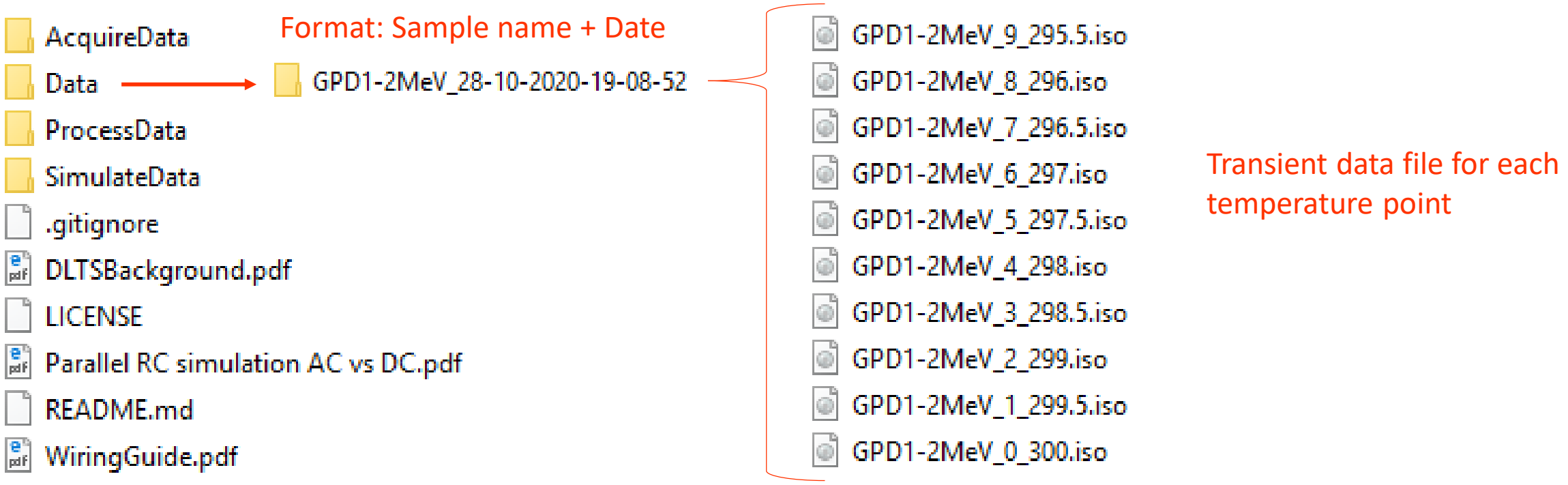
Next temp. step

Yellow and Green -> Single transient data

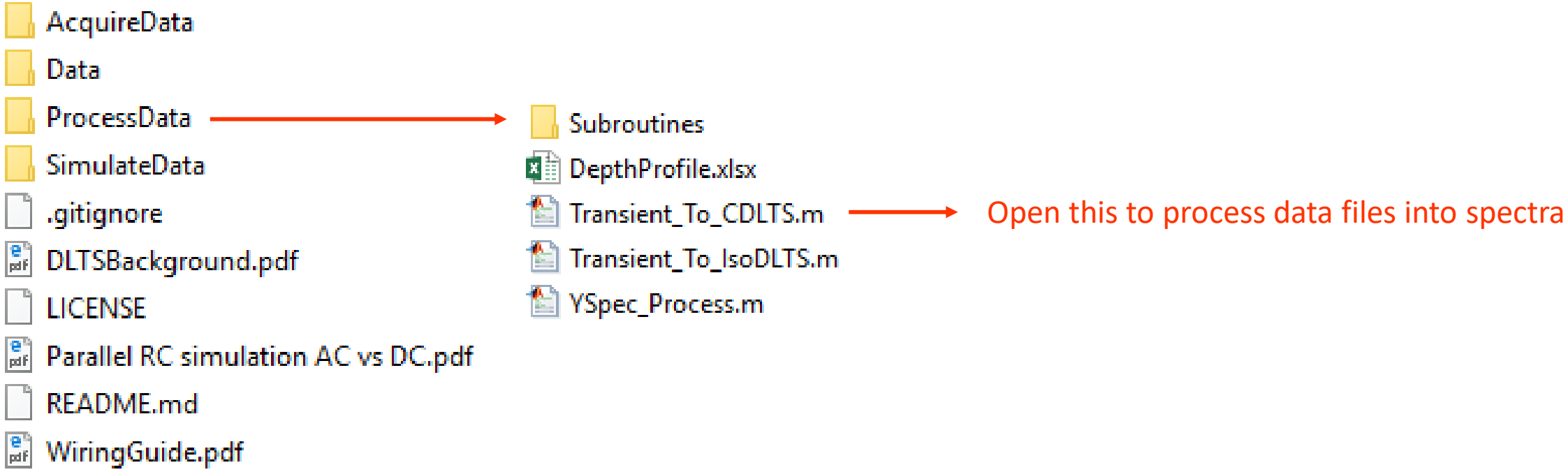
Red -> Averaged transient data



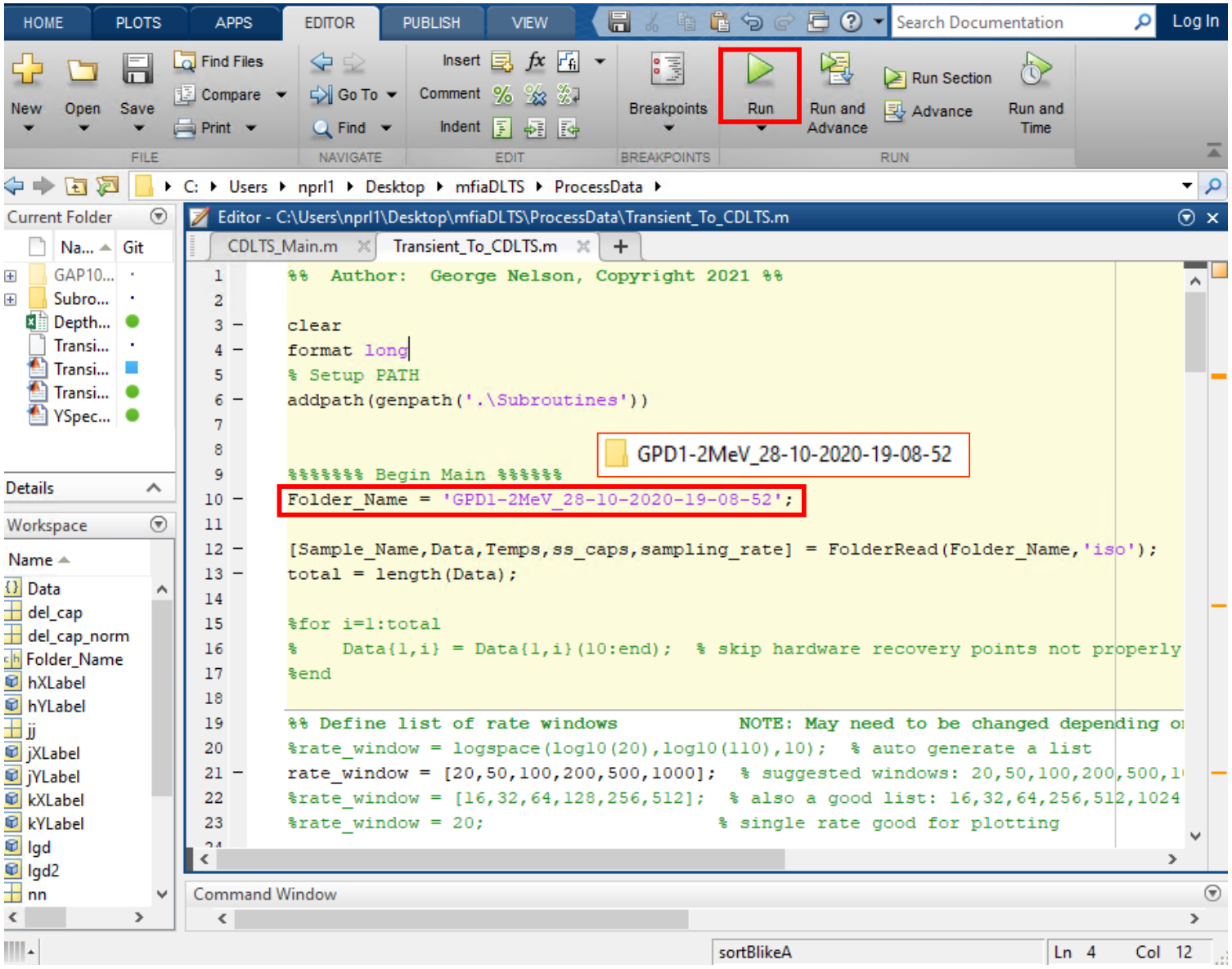
Where Data Goes



After transients are collected, Process Data



Transient_To_CDLTS – Mandatory Steps



All that is needed is:

- Change “Folder_Name” to saved data folder
- Hit run

Transient_To_CDLTS – Advanced Settings

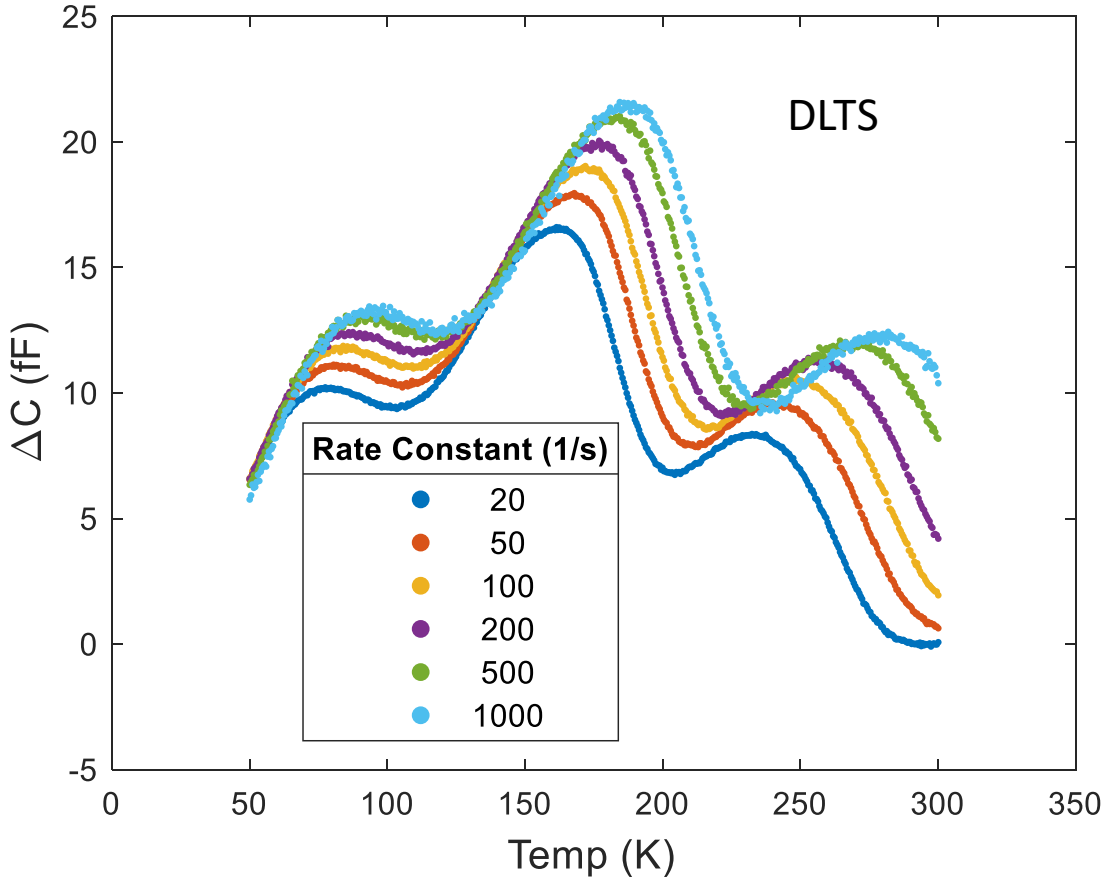
```
%% Define list of rate windows            NOTE: May need to be changed depending on weighting function
%rate_window = logspace(log10(20),log10(110),10);  % auto generate a list
rate_window = [20,50,100,200,500,1000];  % suggested windows: 20,50,100,200,500,1000,2000,5000
%rate_window = [16,32,64,128,256,512];  % also a good list: 16,32,64,256,512,1024
%rate_window = 20;                      % single rate good for plotting
```

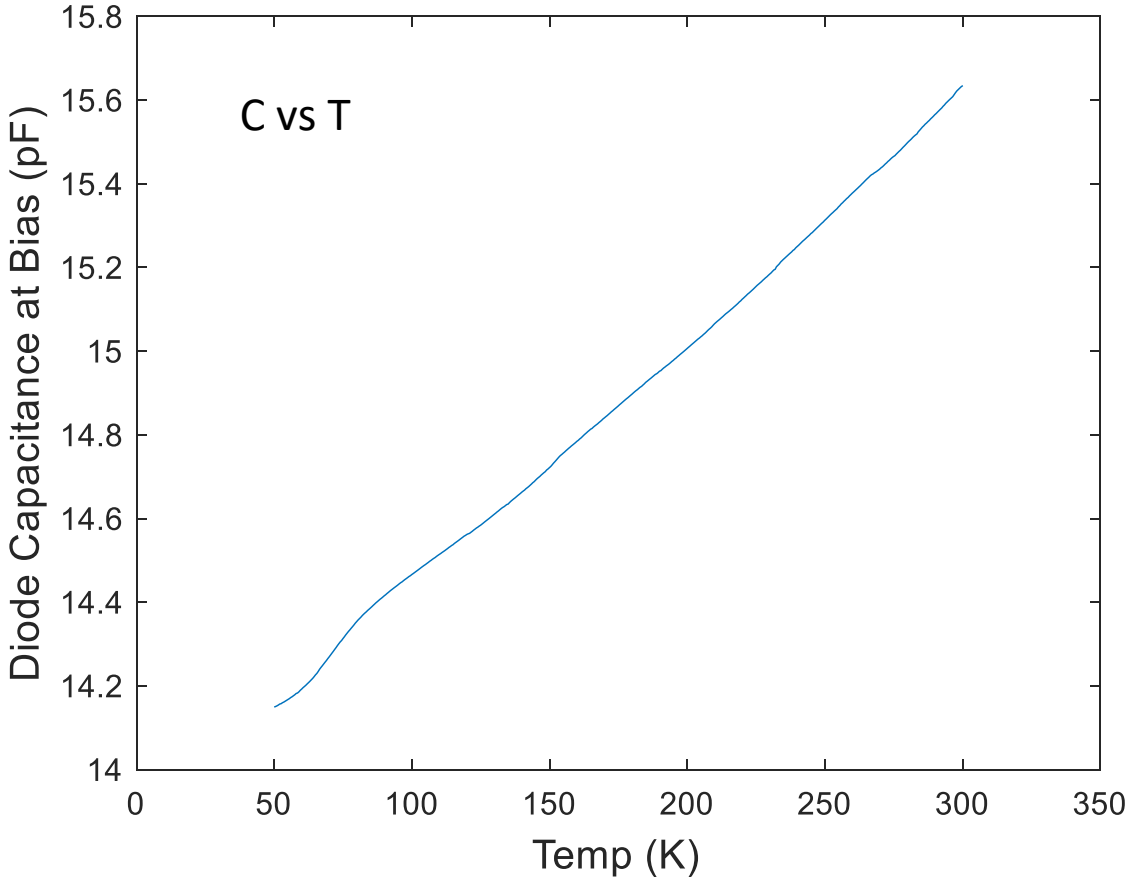
- Here you may set the rate window constants (only uncomment one line)

```
%% List of weighting functions, one must be used and only one
%[del_cap,del_cap_norm] = weightboxcar(Data,rate_window,sampling_rate,ss_caps,total);    %TODO: Find proper gating and timing
%[del_cap,del_cap_norm] = weightlockin(Data,rate_window,sampling_rate,ss_caps,total);    % Most trusted
%[del_cap,del_cap_norm] = weightexp(Data,rate_window,sampling_rate,ss_caps,total);       % Good for SNR but aliasing at high frequency
[del_cap,del_cap_norm] = weightexpaa(Data,rate_window,sampling_rate,ss_caps,total);     % Best SNR, slowest. Default
%[del_cap,del_cap_norm] = weightsine(Data,rate_window,sampling_rate,ss_caps,total);      % Alternative, decent SNR
%[del_cap,del_cap_norm] = weightcosine(Data,rate_window,sampling_rate,ss_caps,total);    % Supposed to be good for resolution
```

- Advanced users can set the weighting function (only uncomment one line)
  - You may also implement your own
- Exponential weighting function with interpolation (weightexpaa) is default and has best SNR

Results of Transient_To_CDLTS





Spectra and CvsT data output in data subfolder in .dat file after running Transient_to_CDLTS

-No Arrhenius plotting supported at this date, will have to find peaks and plot manually