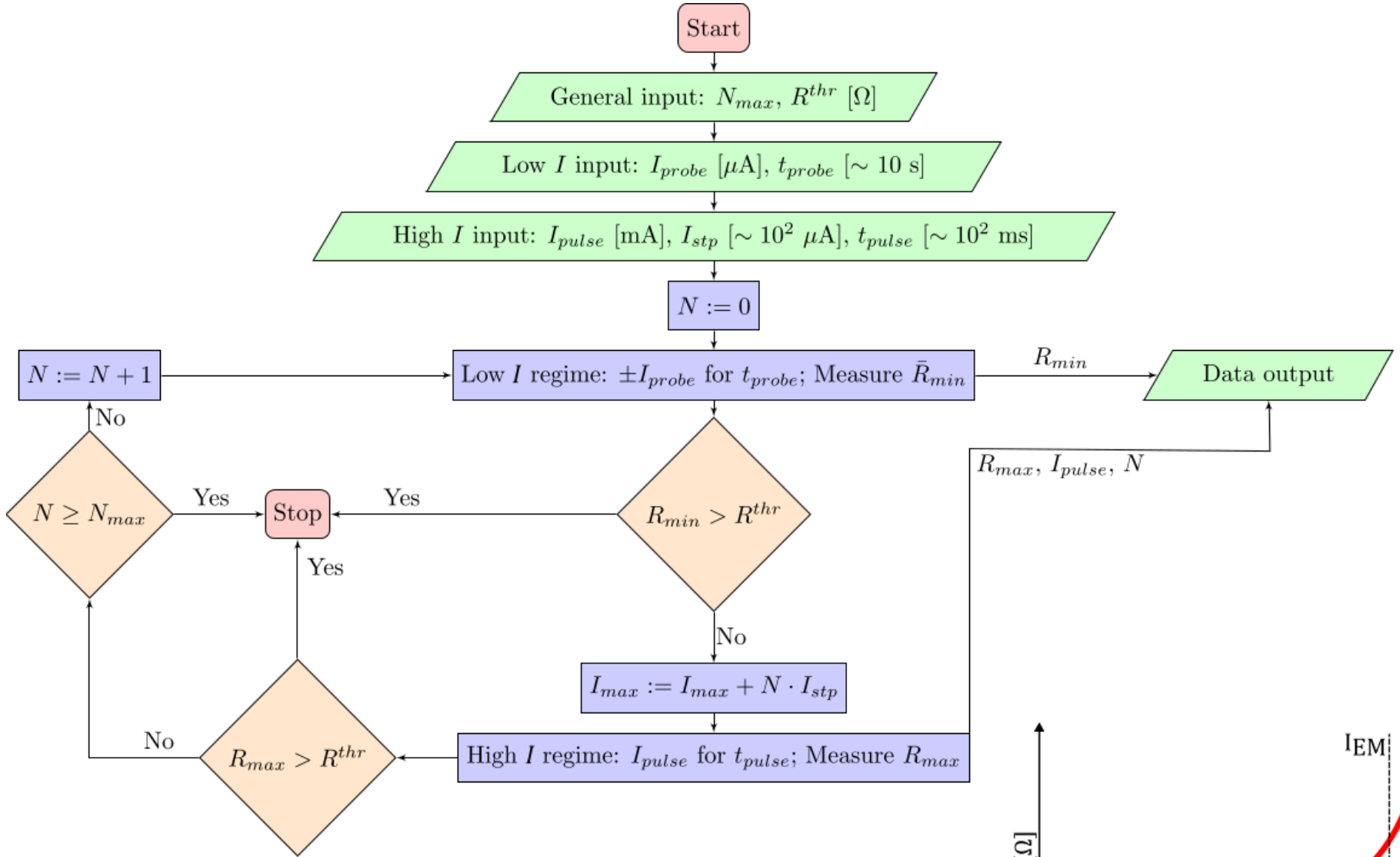
EPNM-ELEX

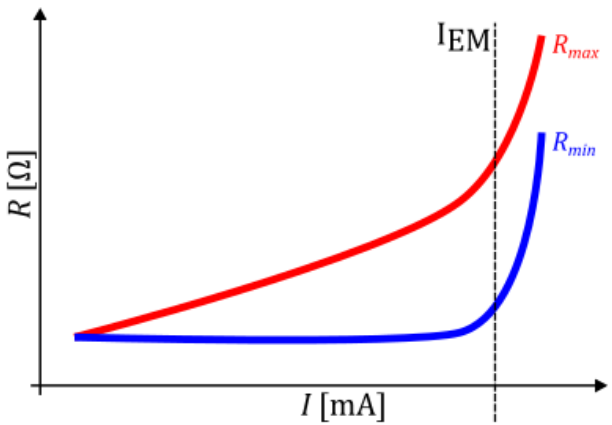
User guide

Attribution and Contact

This project is a Python-based graphical user interface (GUI) designed for controlling and automating the pulsed electromigration protocol used at **EPNM (Experimental Physics of Nanostructured Materials)**, **ULiège**. The software interfaces with various source-measure units and nanovoltmeters, providing a streamlined way to execute the pusled electromigration protocol under different configurations. Created by Stefan Marinković (smarinkovic@uliege.be; marinkovicstefan@outlook.com). See https://doi.org/10.1103/PhysRevApplied.19.054009 for a description of the protocol. Special thanks to the EPNM team and Daniel Stoffels for contribution of work and facilities.

Protocol


Start
General input: N_{max}, R^{thr} [Ω]
Low I input: I_{probe} [μA], t_{probe} [~ 10 s]
High I input: I_{pulse} [mA], I_{stp} [~ 10^2 μA], t_{pulse} [~ 10^2 ms]
$N := 0$
$N := N + 1$
Low I regime: $\pm I_{probe}$ for t_{probe}; Measure $\bar{R}_{min}$
R_{min}
Data output
No
$N \geq N_{max}$
Yes
Stop
Yes
$R_{min} > R^{thr}$
Yes
R_{max}, I_{pulse}, N
No
$I_{max} := I_{max} + N \cdot I_{stp}$
No
$R_{max} > R^{thr}$
High I regime: I_{pulse} for t_{pulse}; Measure R_{max}


I_{EM}
R_{max}
R_{min}
R [Ω]
I [mA]

Program Notes

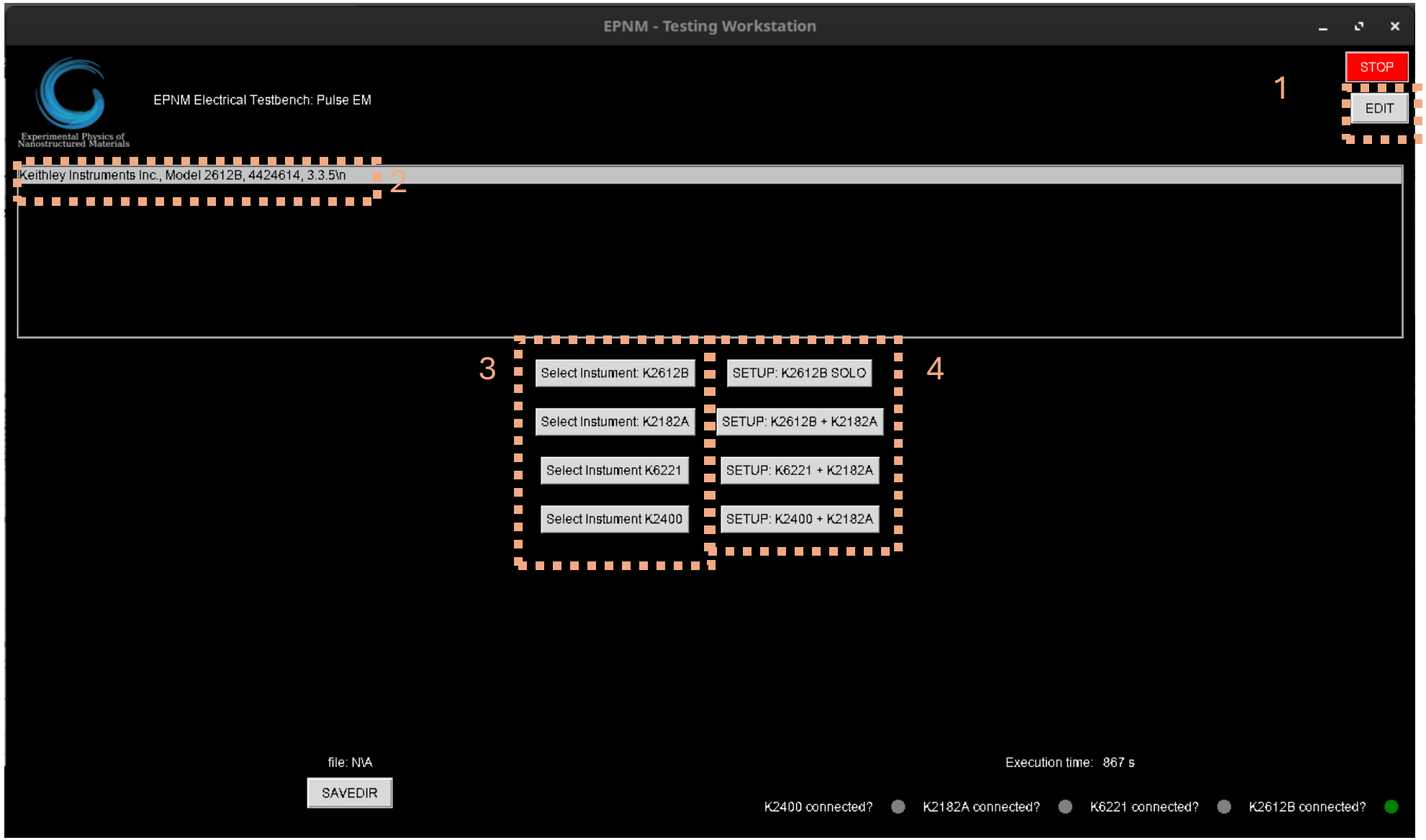
- Repo: https://github.com/stef-ma/epnm-elex
- Note: Setting up gpib controls via python can be painful, feel free to contact for help.
  The program has quite a few rough edges and may be at times difficult, I'll try to make it usable and am available to push updates to correct common errors.
- This document is meant to help you along while the README.md has the basic usage already explained.

1 – The STOP button safely shuts down the instruments and exits the program at any point, if the software is not hung up. The START button looks for Instruments and allows one to configure the program.

2 – The grey indicators turn green when the given instrument connects. The execution timer counts elapsed seconds since the start of the program. If it gets stuck, the program is stuck.

3 – the SAVEDIR button allows you to set the data saving directory. You only need the directory, as the software will create separate folders for R(t) and Pulse measurements and save data with timestamps for names in csv.

Click "START" to begin.



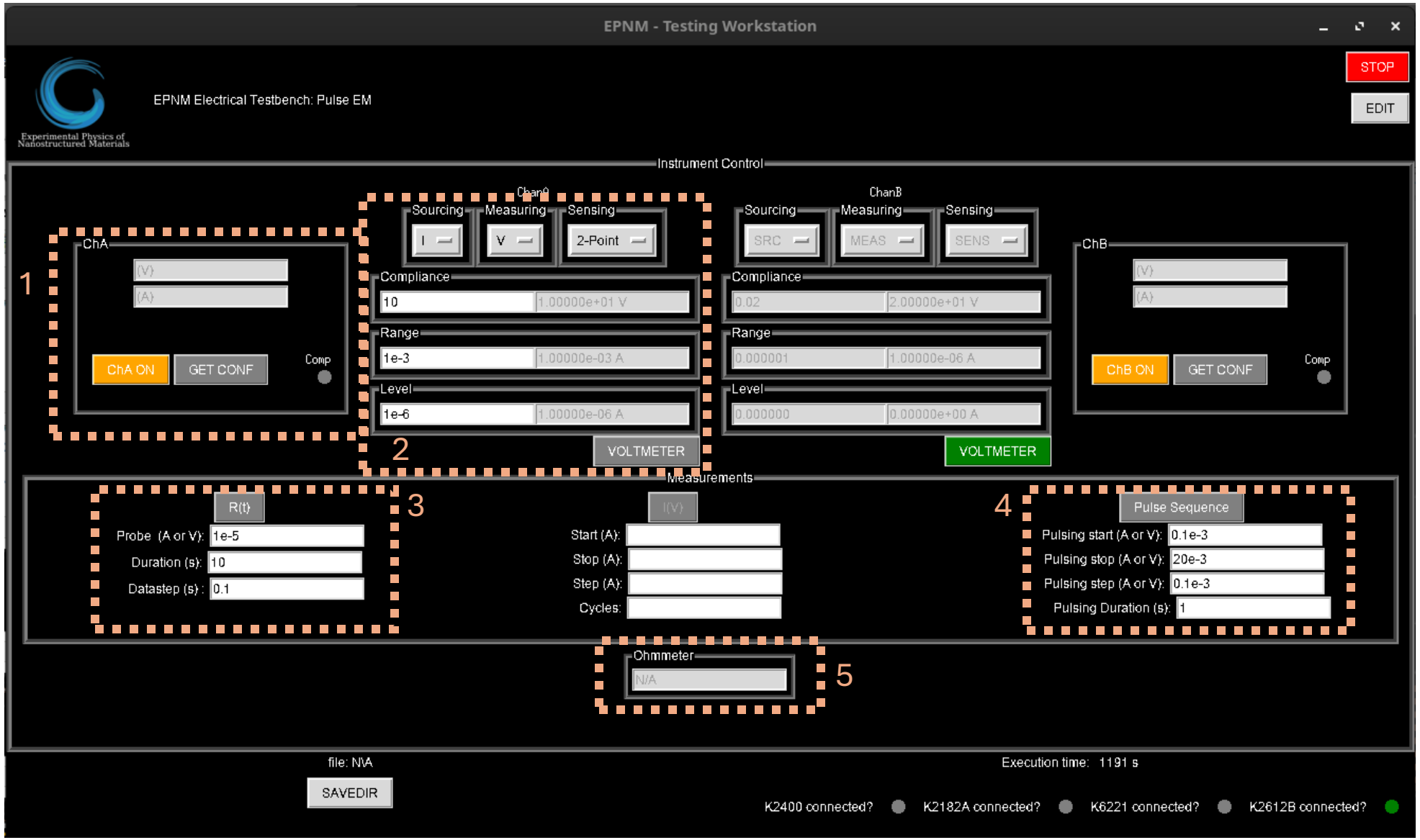
1 – The EDIT button should always disconnect all instruments and bring the operator back to this screen, allowing you to change configurations or reconnect the instrument.

2 – Detected instruments appear in the list here. The user can select them here to connect via the buttons in 3.

3 – The Select Instrument buttons instantiate the given instrument control classes. No intelligent checks are performed to make sure the correct instrument is connected with the right button. This is the users responsibility.

4- After all needed instruments are connected, the SETUP buttons instantiate the control panel and measurement setups for the given instrument combination

The single instrument K2612B example will be shown after. The bottom two SETUPs need more development and testing.



K2612B SOLO Example. This panel looks different depending on the used instruments. This SMU has two channels that can be configured as sources or voltmeters. We will configure channel A to source current, and measure voltage and channel B to be the voltmeter.

1 – V and I readout, ON/OFF button and GET CONF button. The GET CONF button reads the settings of the given channel and updates the greyed out panels in 2. The same can be seen on the far right for channel B.

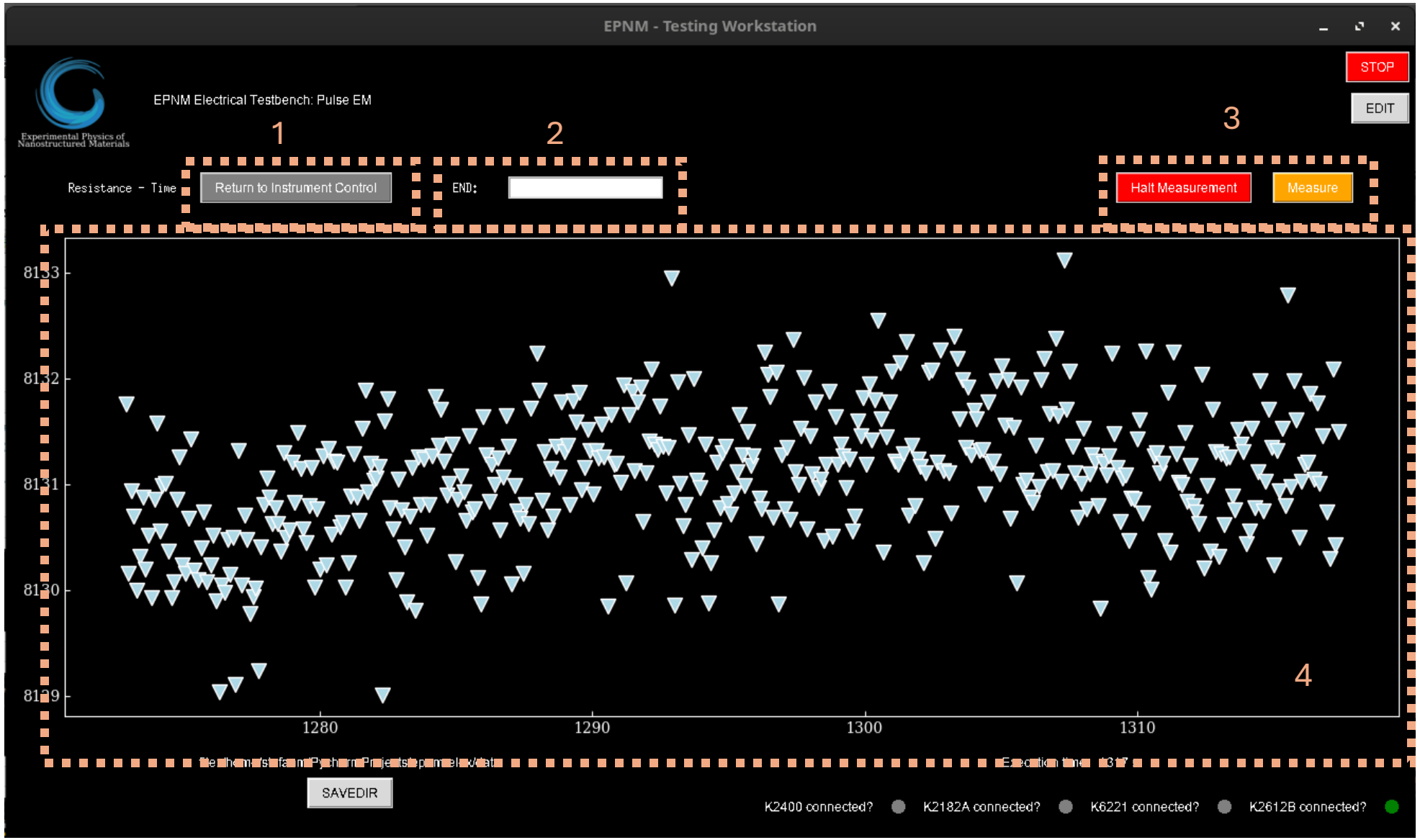
2 – Channel configuration. Sourcing can be set to I or V mode, same for measurement mode. Sensing should in theory be able to be configured between 2-point and 4-point for each channel, but all implementations currently operate on 2-point, so you can either ignore it, or set it to 2-point. The VOLTMETER button can be clicked to make the channel the voltmeter (sources 0A and reads voltage, as seen on the right). Compliance, Range and (Source) level can be configured in the input panels as usual for such instruments.

3 – Parameters for R(t) measurements. Must be setup for individual R(t) measurements and for Pulse Sequence measruements (see algorithm slide). During pulsing, R(t) is performed twice between each pulse (once positive and once negative), so be mindful of the duration you set for it here. 5-10s is what we used before.

4 – Parameters for Pulse Sequence measurements.

5 – The Ohmmeter updates automatically if both channels are turned on and at least one is a voltmeter.

NOTE: Clicking on R(t) in 3. or Pulse Sequence in 4. takes you to the Measurement panel. The I(V) section in the center is not implemented. You can ignore it.



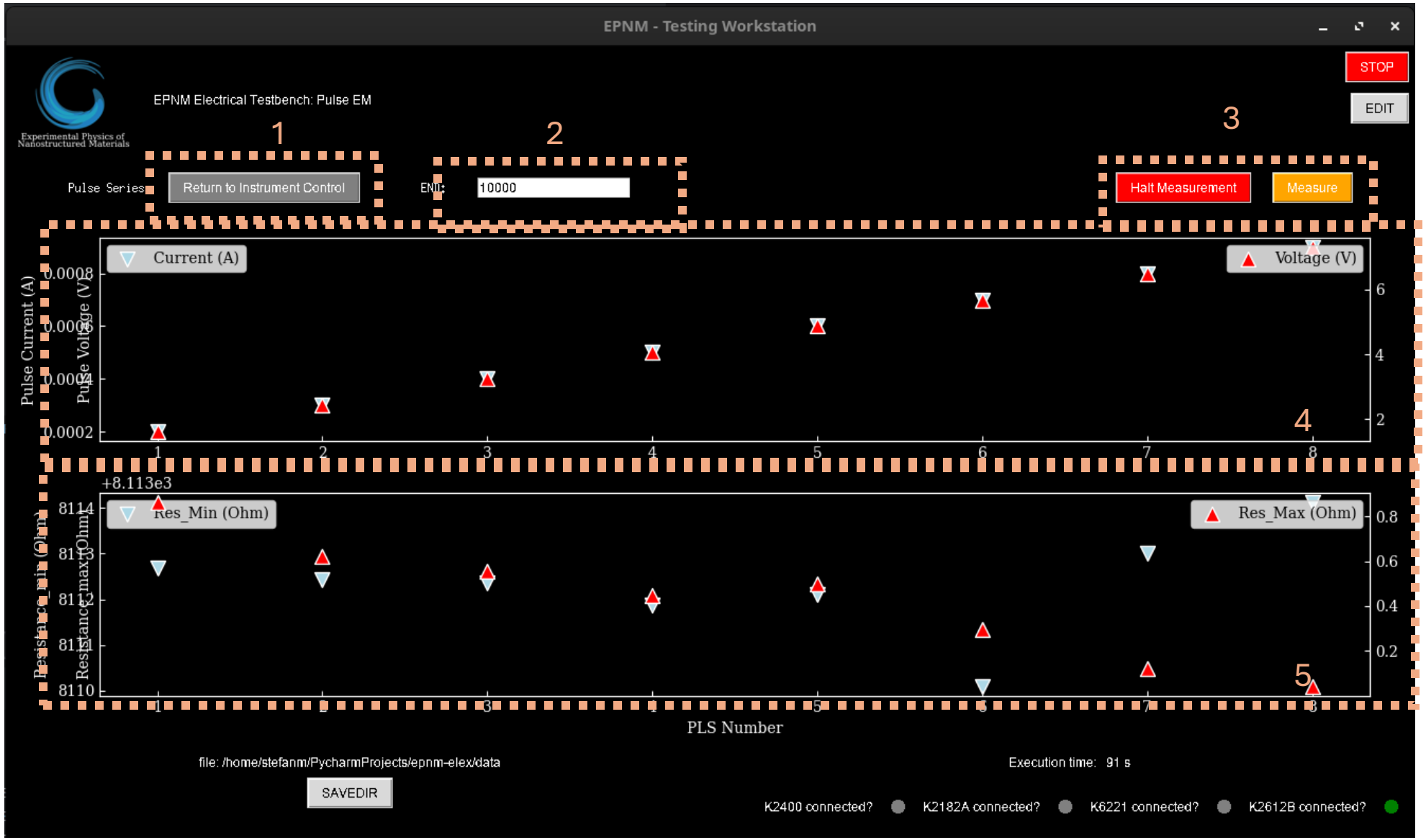
R(t) measurement panel

1 – Returns you to the configuration panel.

2 – END input takes a maximum resistance value. If it is reached or surpassed the outputs are turned off and the measurement ends. **NOTE: All the input panels are read live as you type. This can lead to some unexpected behavior as you start typing, e.g. If you want to put 1000, it will read the first digit "1" immediately and test against it on the next GUI cycle, which might force a halt before you finish typing. Ideally, you will fill this out before starting the measurement, but if you need to change it after the start copy paste the complete number into it to avoid this problem.** This behavior is to be patched out at some point.

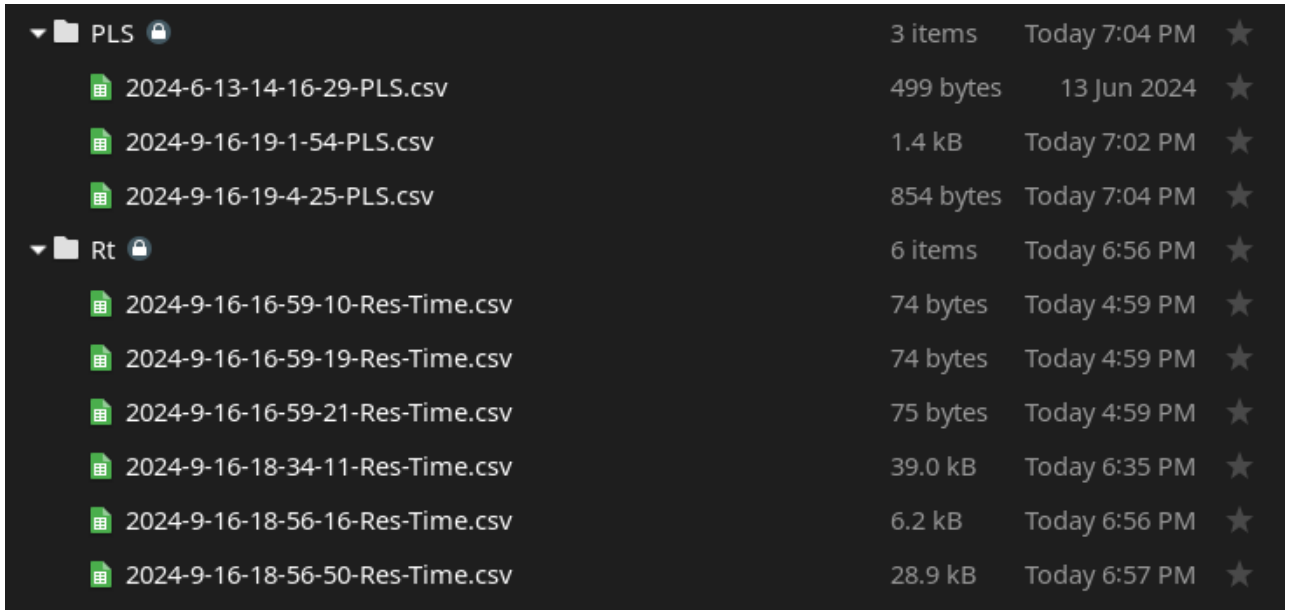
3 – "Halt Measurement" performs a safe stop of the measruement in progress. "Measure" starts the measurement with the given configuration.

4 – The readings are plotted here in terms of resistance on the ordinate and time on the abscise. There are some graphical bugs so the ax labels are not shown in the current version.



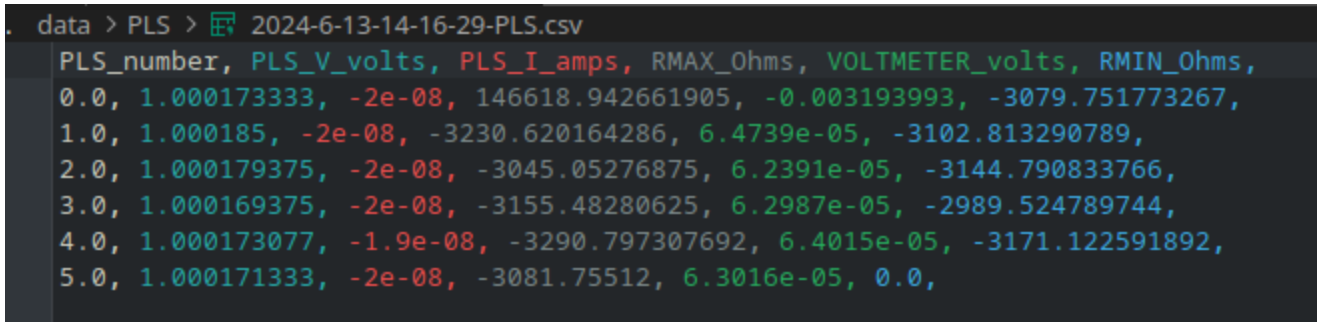
1,2,3 – Same as previous slide
4 – Pulse current (left ordinate) and Pulse Voltage (right ordinate) plots against pulse number (abscisse)
5 – R_min (left) and R_max (right) plots against Pulse number. R_max is based on the data from 4. while R_min is an average of the R(t) that is done between the pulses.

NOTE: The pannels update after the R(t) of an individual pulse is done, so there is quite some downtime. The R(t)'s are not plotted on screen in Pulse Sequence mode, and the R(t) data is not saved to avoid data bloat.



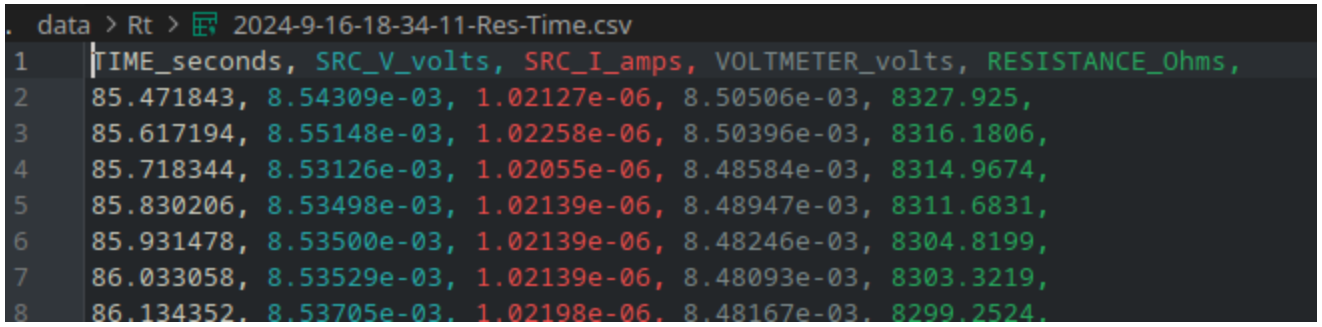
SAVEDIR

The software saves data automatically using a timestamp as the filename. Pulse sequence data is saved in PLS and individual R(t)'s are saved in Rt.



PLS Data

Includes Pulse number, Source voltage, Source Current, Voltmeter voltage R_max (calculated from voltmeter voltage and source current), and the averaged R_min. R(t) data is discarded.



Rt Data

Includes time, source voltage, source current, voltmeter voltage and resistance (from voltmeter voltage and source current)
NOTE: Time is execution time. For measurement time subtract the first value.
To be patched.