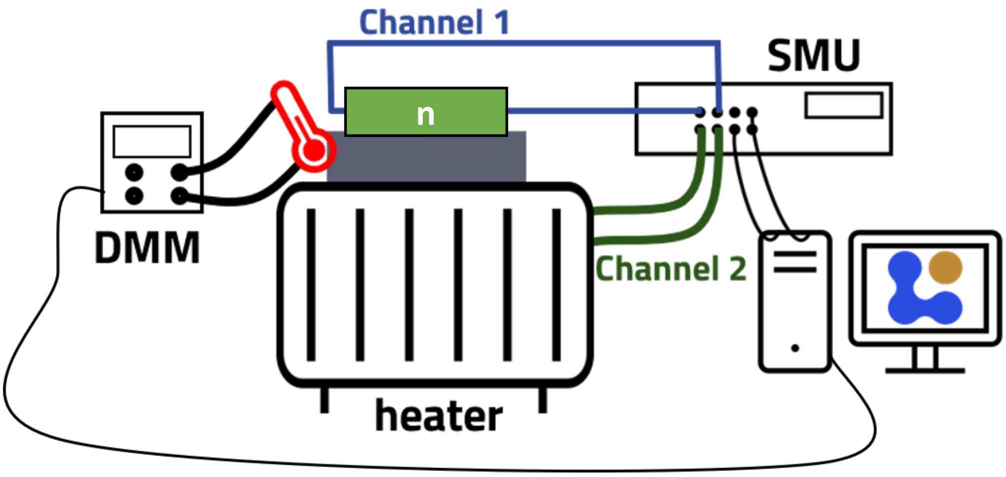
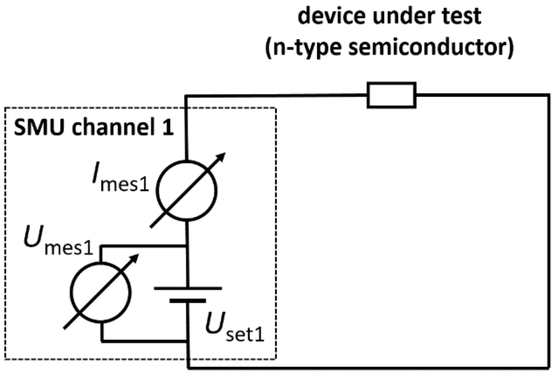
Controlling experiments and recording FAIR data with NOMAD CAMELS

Alexander D. Fuchs, Heiko B. Weber, Michael Krieger



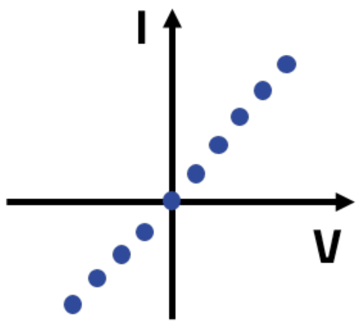
Exercise 1 - Simple I-V characteristics

At first, we will only consider a simplified setup consisting of only the n-type semiconductor and the SMU.
Our goal is to measure the current-voltage (I-V) characteristics of the semiconductor using a source-measure unit (SMU).



Start CAMELS and enter basic metadata

1. Start NOMAD CAMELS. During the first start, you will be asked to define the data directory, into which the measurement data will be stored.
2. Enter the following metadata:
   - user name (add a new user and enter your name)
   - add new sample with the following sample information:
   `name`: `ntype1`,
   `sample_id`: `1337`
   `description`: `Virtual experiment ntype semiconductor`
   - give this `session` the name `Demo`



Install and configure required instruments

3. Install the `demo_source_measure_unit` instrument driver from CAMELS' instrument manager.
   The `demo_source_measure_unit` is a 2-channel source-measure unit connected to the experiment simulator *SandboxForCAMELS*.
   In the simulator, different devices can be connected to **Channel 1** of the `demo_source_measure_unit`.
4. In the instrument manager, navigate to the tab `Configure Instruments`, and create an instance of `demo_source_measure_unit` and give it the name `SMU`.
   Change the `experiment` to `semiconductor_resistor_on_heater`.
   You can keep all other settings as they are.

Define the measurement protocol

5. Create a new measurement protocol.
   In the protocol sequence → add a step `Simple Sweep`.
   Configure the sweep channel to be `SMU_setV1` (voltage of channel 1 of the SMU), the `Start` voltage as `-1`, the `Stop` voltage as `1`, the number of data `points` `51`.

   Read the following channels: `SMU_mesI1` (current reading of SMU channel 1), `SMU_mesV1` (voltage reading of SMU channel 1) and `SMU_setV1` (voltage value set for SMU channel 1).
6. On the simple sweep configuration page (**RIGHT side**), define a plot of type `X-Y plot` with the voltage `SMU_mesV1` plotted on the x-axis, the current `SMU_mesI1` on the y-axis.
   Hint: right clicking into the input boxes displays a context menu, from which you may choose the channel value. Enter meaningful title, x-label and y-label; use only letters and white spaces here.

Let's start the experiment

7. Close the protocol editor and start the measurement protocol.
8. Inspect the measurement data file. In order to view the .h5 file (HDF5 format), you may use the h5web viewer available at https://myhdf5.hdfgroup.org/.

We now want to upload and evaluate the data in our ELN (Electronic Lab Notebook)
You can continue with Exercise 2 on the next page.

Exercise 2 - Automatic upload of measurement data into ELNs

NOMAD CAMELS can directly upload the measurement data into your Electronic Lab Notebook (ELN). Currently, two open-source ELN applications are supported: **NOMAD (Oasis)** and **eLabFTW**. This exercise shows the workflow with NOMAD, it is similar for eLabFTW, but for that you need to install an extension to NOMAD CAMELS.

ATTENTION: The "oasis" version of NOMAD is a beta version that is **NOT** intended to be used for saving your files in production! Only use it for testing purposes!

Some preparation

1. Open NOMAD in your web browser via the link https://nomad-lab.eu/prod/v1/oasis/gui/about/information and login. If you do not have a NOMAD user account, you can create one by clicking on `Login / Register` and then on `Register` at the bottom of the login form.
2. After login, create a new data space for your measurement data as follows: select `Uploads` from the top `Publish` menu. Then click `Create a new upload`. Give the upload the name `CAMELS Demo` by clicking onto the pencil symbol next to `unnamed upload`. Don't forget to click `Save`.
3. `OPTIONAL:` You can add "Basic ELN" entries for the sample and the two instruments (SMU & DMM) we are using.

Login to NOMAD

4. Back in the main window of CAMELS, select `NOMAD user` from the dropdown list in the user metadata section and click the button `NOMAD login`.
5. In the login window that is showing up, make sure that the "NOMAD Oasis" option is selected in the drop down menu. Enter the URL from above and make sure the authentication type is `token`. Click the button `get the token!`. You will be redirected to a NOMAD webpage in your web browser. On this webpage, click onto the clipboard icon in the paragraph `App token`. This will generate and copy to clipboard a new app token that is valid only for today. Close the browser tab and go back to the login window of CAMELS.
   Paste the app token from the clipboard into the textbox `Authentication token`. Click `OK`.
   CAMELS is now connected to your user space in NOMAD. Your user data (e.g. name) is automatically fetched from NOMAD and displayed on CAMELS' main window.

Connect the ELN entries with CAMELS

6. `OPTIONAL:` Under `Sample` select `use NOMAD sample` and then click `select NOMAD sample`

7. `OPTIONAL:` Open the **`Instrument Manager`**, go the SMU and DMM and connect them to the NOMAD ELN entries.

Upload the measurement data to NOMAD

8. In the main window of CAMELS, select **`auto upload`** from the dropdown list next to `NOMAD Upload:`.
   A second dropdown list will show up, which lists all your data spaces (uploads) on the NOMAD server, including the one ("CAMELS Demo") we have just created in step 2; select it from the dropdown list.

Let's go and see

9. Start the measurement protocol of the previous exercise and watch CAMELS run the measurement.
10. After the measurement is finished, go to NOMAD in your web browser and reload the upload **`CAMELS Demo`**. You will find the measurement data file in the list of processed data.
11. Inspect the data by clicking the arrow (→) next to the entry. The interactive h5web viewer will show up within NOMAD.
    You will see that the data entry references the other ELN entries you connected them to in CAMELS.

Evaluate Data in a Juypter Notebook

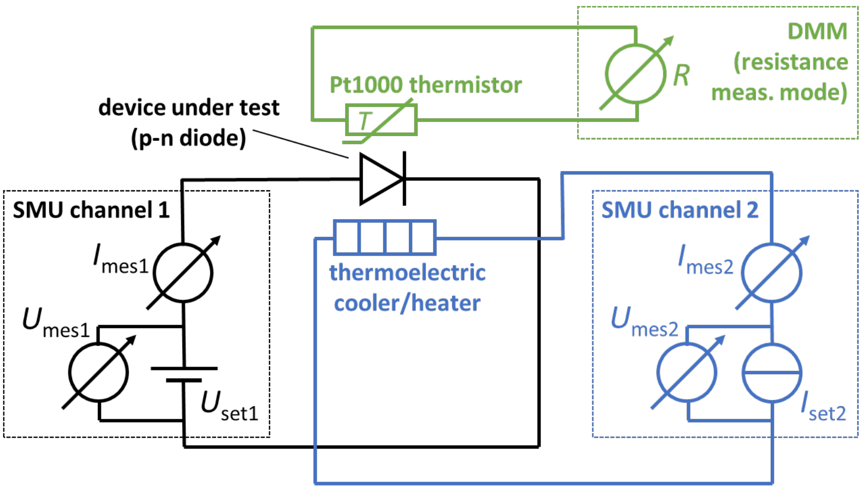
1. Go to "Analyze" in the top left menu bar and select "NOMAD Remote Tools Hub".
2. Click on "launch" for the Jupyter Notebook. Then wait briefly and click "OPEN".
3. Navigate to your upload with the menu on the left
4. Download the Jupyter Notebook from the Tutorial Website. To do this you can right-click on the link and select "save link as".
5. Drag-and-drop the file into the file section on the left side of the NOMAD Jupyter Hub
6. Run the new Notebook. Make sure that the filename matches the .h5 file you uploaded

You can continue with Exercise 3 on the next page.
This is a more complicated measurement setup to measure the IV-curve at different temperatures.

Exercise 3 (Temperature-dependent measurement)

Measure the temperature-dependent I-V characteristics of the n-type semiconductor. CAMELS provides a virtual experiment with an n-type semiconductor mounted on top of a thermoelectric cooler/heater. The temperature is measured by a PT1000 thermistor mounted side-by-side with the sample. The thermoelectric cooler/heater is connected to **Channel 2** of the SMU used in exercise 1, the Pt1000 resistance is measured by a digital multimeter (DMM) in resistance measurement mode.

*Reminder: The semiconductor is connected to **Channel 1** of the SMU.*



<u>Install and configure required instruments</u>

1. Open the instrument manager and install the `demo_digital_multimeter`, and the `PID` instrument drivers. Go to the configure instruments tab and create an instance of `demo_digital_multimeter` with the name `DMM`.
2. In the instrument manager, add an instance of the `PID` controller.
   As <u>input channel</u> for the `PID`, configure `DMM_resistance`,
   as <u>output channel</u> `SMU_setI2`.
   Enter `pt1000` as conversion function for reading.
   In the table of PID parameters, enter `kp` = `0.5`, `ki` = `0.5`, `kd` = `0.1`

<u>Define the temperature sweep</u>

3. Edit the measurement protocol of exercise 1 as follows:
4. Add a **variable** called `points` with the value `6`. This will be the number of temperature setpoints between 300K and 400K.
5. Add a `For Loop` with name `Temperature_sweep` for the temperature sweep from 300 K to 400 K with the following parameters:
   `Start` = `300`, `Stop` = `400`, `# points` = `points` (again you can right click to add the variable)

6. In the loop, add a step **Set Channels**.
   Check the channels **PID_pid_on** and **PID_setpoint** and add the following values: **PID_pid_on** = 1, **PID_setpoint** = Temperature_sweep_Value (you may select this from the context menu, right click and **Insert Variable**). This activates the PID controller and sets the temperature setpoint to the current value of the For Loop.
7. After **Set Channels** in the sequence, insert a step **PID wait for stable**. The sequence will pause here, in order to wait for the temperature to be stable at the setpoint.
8. Move the **Simple Sweep** step into the loop as the last step (drag-and-drop). The I-V measurement will be executed for each temperature defined in the For Loop.
9. Click onto the step **Simple Sweep** and read additionally **PID_current_value**, **PID_setpoint** and **DMM_resistance** in the **Read-Channels** list. This ensures that we store the temperature data along with the I-V data.
10. Add a **Read Channels** step below the simple sweep:
    Set the checkmark of **read/save variables** on the top right to true.
    Read the **PID_current_value**.

Stop the PID controller at the end of the measurement

11. Add a **Set Channels** step below the For loop (not inside!).
    Set the channel **PID_pid_on** to 0 in order to switch off the PID controller at the end of the measurement protocol.

Now we want to get some evaluation on the fly

12. Select the **Simple Sweep** in the protocol. Click on **Define Plots/Fits** and select your plot. Click on the y-axis you want to fit and check the box **fit?** to enable fitting to the data.
13. Now select **Predefined function** and choose **Linear** as the fit function. Check the boxes **guess initial parameters** and **display fit values**, which will show you the fit's result in the plot. Click **OK.**
14. Select **Define Plots/Fits** in the left part of the window. These plots are always updated when Read Channels is executed.
15. Add a new X-Y plot here. As the x-axis enter **PID_current_value**, which represents the temperature. As y-axis enter **1 / Linear_SMU_mesI1_v_SMU_mesV1_Simple_Sweep_slope**, which is 1 / the fitted slope of the IV curve, i.e. the measured resistance.

Let's start the experiment

16. Close the protocol editor by clicking **OK** and run the measurement protocol.
17. Inspect the measurement data file in NOMAD and evaluate it with a Jupyter Notebook

Exercise 4 - Evaluation of data inside CAMELS

In this exercise, you will take a closer look to the evaluation possibilities within CAMELS. The simulated semiconductor resistor has a doping of which the activation energy can be evaluated from temperature dependent IV-measurements.

The resistance of a semiconductor can be given as

$$R = R_0 \cdot \exp\left(\frac{E_a}{k_B T}\right)$$

where E_a is the activation energy of the semiconductor's dopands. To evaluate this directly in CAMELS we do the following:

<u>Add another plot</u>

1. For the general plots of the protocol, we want to add the Arrhenius plot for the resistance, i.e. $\ln(R) = \ln(R_0) + \frac{E_a}{k_B T}$. The x-axis will then be `1 / PID_current_value * const.e / const.k` where const.e and const.k are the elementary charge and the Boltzmann constant. This gives for the x-axis $\frac{1}{k_B T}$ in 1/eV.
2. The y-axis follows then as `log(1/Linear_SMU_mesI1_v_SMU_mesV1_Simple_Sweep_slope)` which is $\ln(R)$.
3. Next, define the fit. You can either use a linear function, where the slope is then E_a in eV and the intercept is $\ln(R_0)$, or you define a custom function. The custom function enables you to name the fit values directly. You can enter `log(R0) + Ea * x` as the function. Now since R_0 is in the logarithm, the fit is not as simple anymore and it is helpful to set more meaningful starting values. Uncheck `guess initial parameters` and enter as fit parameter names R0 and Ea. As initial value, you may want to give R0 a higher value, e.g. `50`.
4. Run the protocol and see whether you get a reasonable value for R_0 and E_a.

Further information

Installation of NOMAD CAMELS

The full installation documentation is available at:
https://fau-lap.github.io/NOMAD-CAMELS/doc/installation/installation.html

For Windows computers, you can download our installer, which automatically installs NOMAD CAMELS within a few minutes. The installer is available from:
https://app.lap.nat.fau.de/nomad-camels/NOMAD-CAMELS_installer.exe

Contact and information

More information? Visit our documentation! https://fau-lap.github.io/NOMAD-CAMELS/

Questions?

- Join the discussions: https://github.com/FAU-LAP/NOMAD-CAMELS/discussions
- Or contact us directly: lap-nomad-camels@fau.de

Found a bug? Please report it! https://github.com/FAU-LAP/NOMAD-CAMELS/issues

NOMAD CAMELS is being developed at the Friedrich-Alexander-Universität Erlangen-Nürnberg within the FAIRmat NFDI consortium.

Chair of Applied Physics at FAU: https://www.lap.physik.nat.fau.eu
FAIRmat: https://www.fairmat-nfdi.eu/fairmat/

Learn more about the NOMAD software family: https://nomad-lab.eu/nomad-lab/