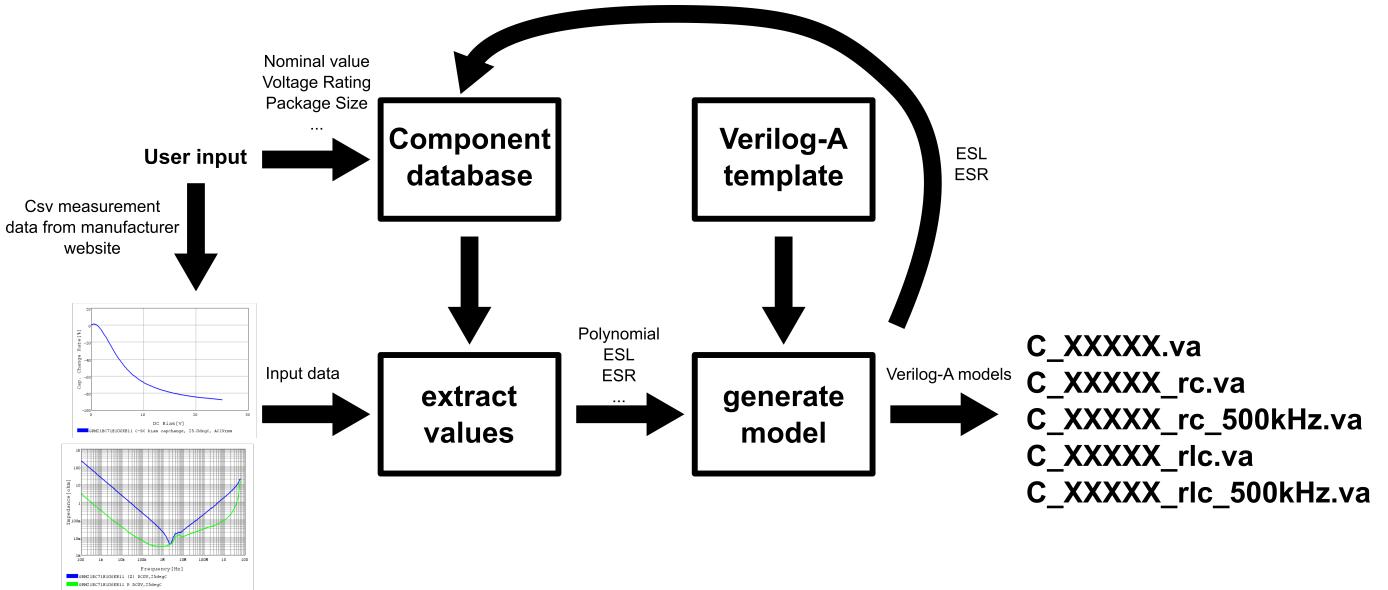


User guide for generate_capacitor python script.

Introduction



This user guide is for the generate_capacitor.py script. The script is responsible for generating various verilog-A models for discrete capacitors based on measurement data provided by the manufacturers website. The script requires the user to download the measurement data, update a component database excel file and then run the script. This user guide is written to help the PhD students at Leibniz University of Hannover to get started with Python, download the generate_capacitor.py script and run the script to extract a new capacitor.

Getting started

To get started using the python script to generate verilog-A models for your discrete capacitors you need to install Python and the required Python-packages used in the script. These packages handle things like: reading/writing to excel files, reading csv data, plotting the results etc. We also need an editor from which we can open, read, edit and run the script.

Download generate_capacitor python script:

I've created a github repository for the generate_capacitor python script:

[GitHub - muheicdesign/generate_capacitor](https://github.com/muheicdesign/generate_capacitor)

Go to the github:

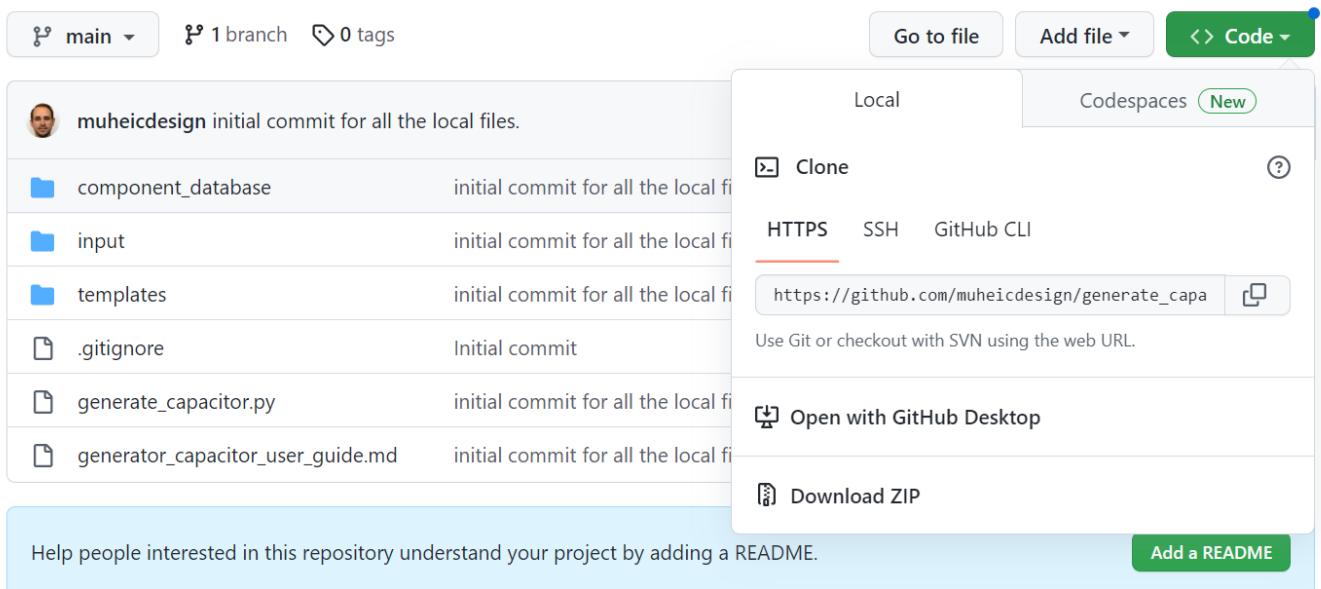
1. Press `< > Code`

2. Choose **Local**

3. **Download ZIP**

or if you use git (you should) then you can clone the repository how your normally would - then you can also easily get any updates to the script.

Save and extract the zip file anywhere you have your other scripts or files.



Install VSCode (or other code editor of your preference)

We need an editor for the scripts where we can; read, edit, compile and run our scripts from. There are many good IDEs out there. I prefer VSCode.

If you don't have a preference download VSCode here: [Download Visual Studio Code - Mac, Linux, Windows](#)

Install VSCode with default setups and open it and go through its "getting started" tutorial. It will ask you to select color schemes etc. When you've set everything as you want it - close the editor again. We'll open it later through Anaconda

Installing Anaconda

While it is perfectly possible to install Python yourself and installing all the required packages one by one it is much easier to use a dedicated software for it: **Anaconda**

Anaconda is a "distribution software" that creates a virtual environment on your computer with a version of Python (default: newest) and where you can easily install new packages and open an **code editor** to edit and run your scripts.

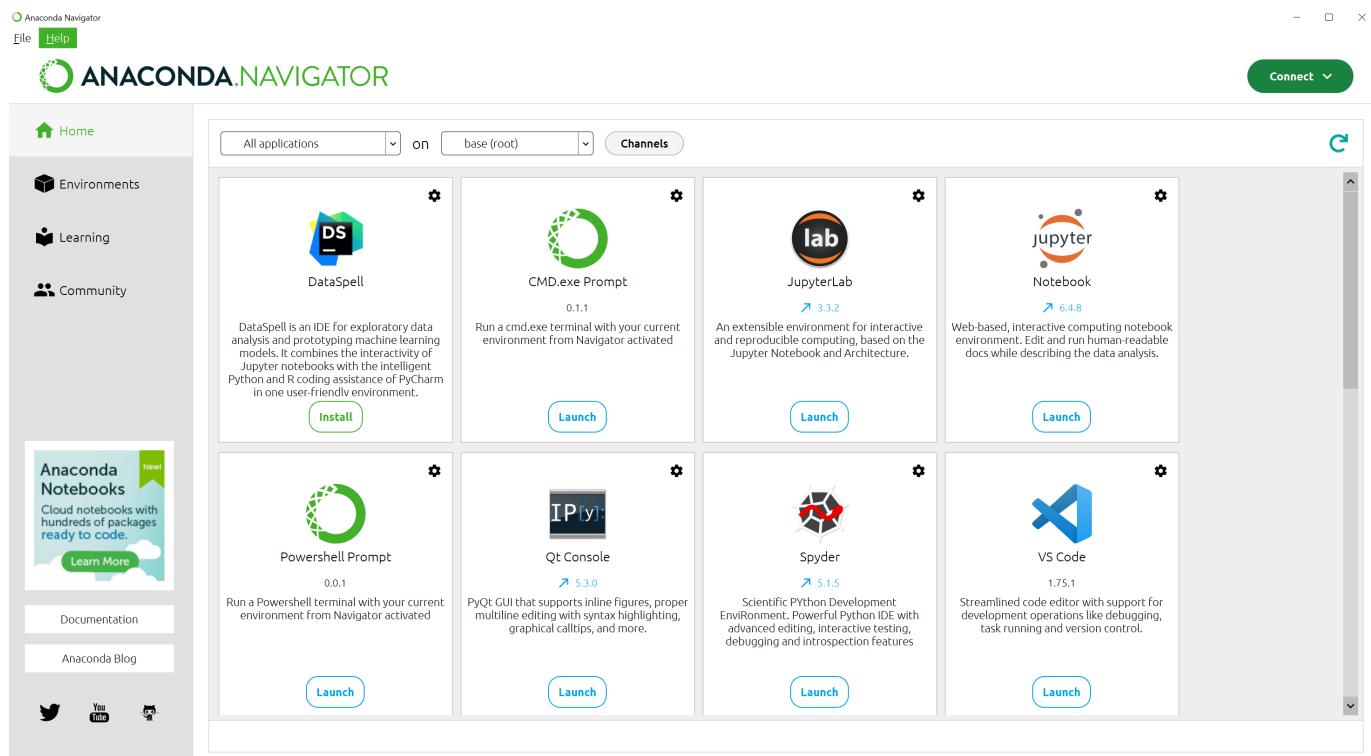
Download anaconda here: [Anaconda | Anaconda Distribution](#)

Install Anaconda on your PC with default settings.

Open Anaconda and create new conda environment

Open Anaconda on your computer. If it needs to update - then update immediately.

After opening it the "home" tab should look something like this:



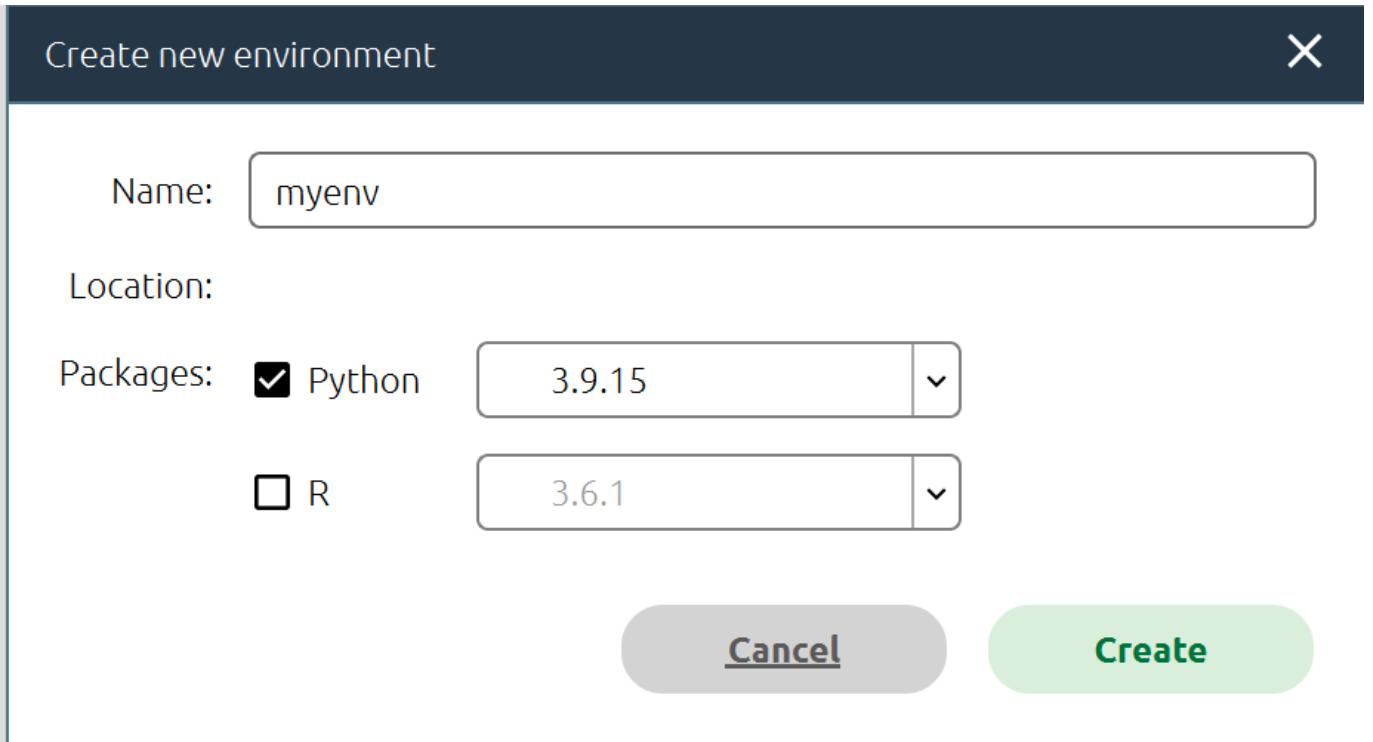
Ensure that "VS Code" is there on the home tab. If it is not - then make sure that you've installed it correctly on your computer.

Now we need to create a new "Environment". The default environment is called "**base (root)**". This has the newest version of Python installed and the most common packages. In general we don't want to mess around too much with base (root). Instead we create a new environment:

1. Go to **Environments**
2. Click **Create**
3. Name your environment

4. Ensure that Python is checked (We don't need R).

1. Just leave the Python version to whatever default it chooses. There are newer versions but they might not be fully compatible with all our packages.



After pressing **Create** the environment will be created. This might take a few minutes:

Search Environments

Installed

Name	T	Description	Version
ca-certificates	<input checked="" type="checkbox"/>	Certificates for use with other packages.	2023.01.10
certifi	<input checked="" type="checkbox"/>	Python package for providing mozilla's ca bundle.	2022.12.7
openssl	<input checked="" type="checkbox"/>	Openssl is an open-source implementation of the ssl and tls protocols	1.1.1t
pip	<input checked="" type="checkbox"/>	Pypy recommended tool for installing python packages	22.3.1
python	<input checked="" type="checkbox"/>	General purpose programming language	3.9.16
setuptools	<input checked="" type="checkbox"/>	Download, build, install, upgrade, and uninstall python packages	65.6.3
sqlite	<input checked="" type="checkbox"/>	Implements a self-contained, zero-configuration, sql database engine	3.40.1
tzdata	<input checked="" type="checkbox"/>	The time zone database (called tz, tzdb or zoneinfo)	2022g
vc	<input checked="" type="checkbox"/>	A meta-package to impose mutual exclusivity among software built with different vc versions	14.2
vs2015_runtime	<input checked="" type="checkbox"/>	Msvc runtimes associated with cl.exe version 19.27.29111 (vs 2019 update 5)	14.27.290...
wheel	<input checked="" type="checkbox"/>	A built-package format for python.	0.38.4
wincerctstore	<input checked="" type="checkbox"/>	Python module to extract ca and crl certs from windows' cert store (ctypes based).	0.2

12 packages available

Make sure that your new environment is chosen indicated by the green play arrow. This is your current environment.

Installing dependencies for running *generate_capacitor.py*

Note that **Installed** is selected in the drop down menu. This filters all the Python packages that has already been installed in the environment. From scratch our environment is pretty empty. Let's add the required packages to run the generate_capacitor script:

In the drop down menu **installed** choose **Not Installed** instead.

Then in the search bar to the right search and check off these packages:

- numpy
- matplotlib
- jinja2
- openpyxl

Name	Description	Version
jinja2	A very fast and expressive template engine.	3.1.2
matplotlib	Publication quality figures in python	3.6.2
numpy	Array processing for numbers, strings, records, and objects.	1.9.3
openpyxl	A python library to read/write excel 2010 xlsx/xlsm files	3.0.9

Note that the packages have now just been "selected" for installing. To install them press **Apply**

The installer will pop up with a window that shows all the co-dependent packages that these packages need. Press **Apply** again:

Install Packages



63 packages will be installed

	Name	Unlink	Link	Channel	Action	▼
1	*zstd	-	1.5.2	pkgs/main	Install	^
2	*zlib	-	1.2.13	pkgs/main	Install	
3	*xz	-	5.2.10	pkgs/main	Install	
4	*tornado	-	6.2	pkgs/main	Install	
5	*toml	-	0.10.2	pkgs/main	Install	
6	*tk	-	8.6.12	pkgs/main	Install	
7	*six	-	1.16.0	pkgs/main	Install	
8	*sip	-	6.6.2	pkgs/main	Install	
9	*qtwebkit	-	5.212	pkgs/main	Install	▼

* indicates the package is a dependency of a selected packages

[Cancel](#)

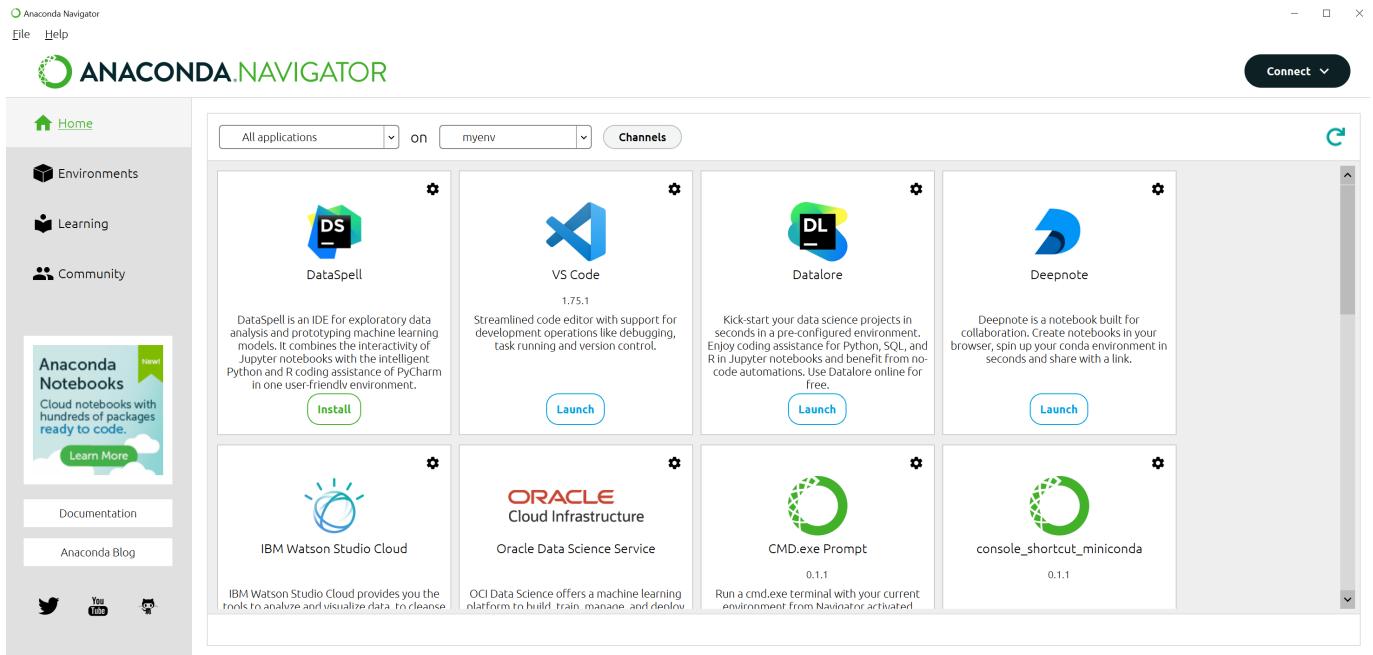
[Apply](#)

The installation of all the packages might take a minute or two.

When the installation has finished - everything is installed and we're ready to go!

Choosing Python code editor

Navigate to the Home tab in the Anaconda Navigator

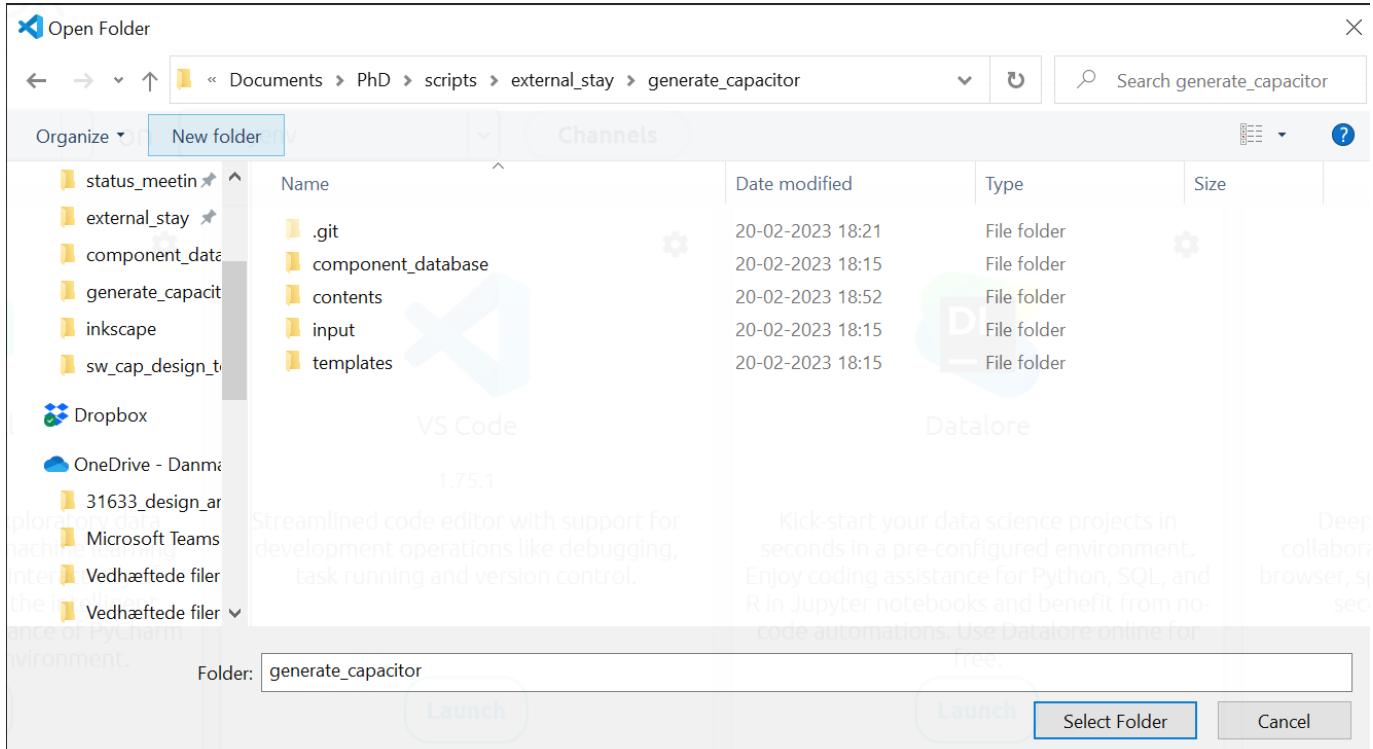


Ensure again that your new anaconda environment is selected at the top!

Press the **VS Code** or your preferred editor (you might have to scroll down to find it!)

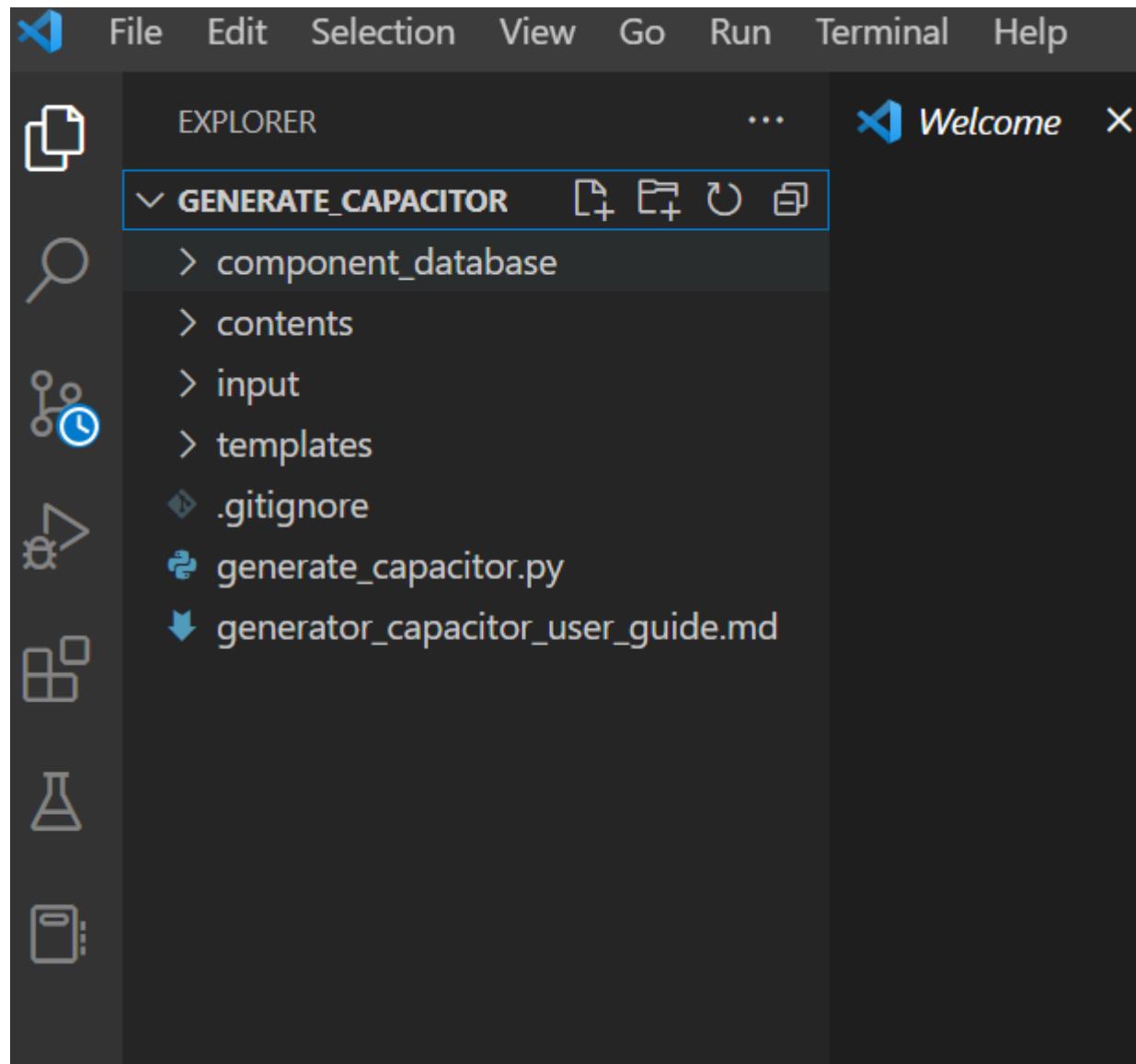
Run default generate_capacitor script

When you've opened VS Code go to File -> Open Folder --> select the "generate_capacitor" folder that you downloaded from GitHub



Press "Select Folder"

Your **Explorer** window in VS Code, showing you the file hierarchy, should look something like this:



Click on "generate_capacitor.py" this is the main script responsible for generating the Verilog-A files. Scroll to the bottom of the script until you see:

```
if __name__ == "__main__":
```

```
File Edit Selection View Go Run Terminal Help
EXPLORER generate_capacitor.py ...
GENERATE_CAPACITOR > component_database > contents > input > templates &lt;.gitignore &lt; generate_capacitor.py &lt; generator_capacitor_user_guide...
if __name__ == "__main__":
    #####
    ##### EXAMPLE 1: #####
    #####
    #### Generate verilog-A files for a GRM21BC71E106KE11 - murata capacitor.
    #### Before this script is run the following has been done by the user:
    # 1: The user has found the component on the vendor webpage and downloaded the
    # the csv data for: ac impedance measurement, ac_esr measurement and the
    # capacitance DC bias voltage degradation
    #
    # 2: The user has inserted ATLEAST the capacitor name and nominal capacitance
    # value in the component_database.xlsx excel file. The ESR and ESL values
    # should not be inserted, since they are extracted from this script.
    #
    # 3: The user has ensured that the csv data is saved under the correct names
    # and in the correct path as specified by the user guide.
    #
    # 4: Remember to close the component_database.xlsx file before running the
    # the script. Otherwise the script cannot open and edit the file.

    ##### USER INPUTS #####
    ### choose manufacture vendor:
    vendor = "murata"

    # part number:
    capacitor = "GRM21BC71E106KE11"

    # Frequency for f_esr extraction:
    f_esr = 500           #in kHz

    ### show plots or not (for visual inspection of polynomial fit and impedance extraction)
    SHOW_PLOT = True

    #Run main generator function:
    generate_capacitor(capacitor, vendor, f_esr)

    #####
    ##### EXAMPLE 2: #####
    #####
    #### Generate verilog-A files for all available in component_database.
    # if errors exist in database or missing csv data, the function should
    # report a WarningInfo and skip these.

    f_esr = 500           #in kHz
    #make_all(f_esr)
```

You should be able to run the script by pressing:

Run -> Run Without Debugging

or you can simply press **ctrl + F5**

Two figures should pop up and in the Terminal window some output prints should appear:

```

generate_capacitor.py
-----
319 if __name__ == "__main__":
320 ##### EXAMPLE 1 #####
321 ##### EXAMPLE 1 #####
322 ##### EXAMPLE 1 #####
323 ##### EXAMPLE 1 #####
324
325     ## Generate verilog-A files for a GRM21BC71E106KE11 - murata capacitor.
326     ## Before this script is run the following has been done by the user:
327     # 1: The user has found the component on the vendor webpage and downloaded the
328     #     the csv data for: ac impedance measurement, ac_esr measurement and the
329     #     capacitance DC bias voltage degradation
330
331     # 2: The user has inserted ATLEAST the capacitor name and nominal capacitance
332     #     value in the component_database.xlsx excel file. The ESR and ESL values
333     #     should not be inserted, since they are extracted from this script.
334
335     # 3: The user has ensured that the csv data is saved under the correct names
336     #     and in the correct path as specified by the user guide.
337
338     # 4: Remember to close the component_database.xlsx file before running the
339     #     the script. Otherwise the script cannot open and edit the file.
340
341 ##### USER INPUTS #####
342
343     ## choose manufacture vendor:
344     vendor = "murata"
345
346     # part number:
347     capacitor = "GRM21BC71E106KE11"
348
349     # Frequency for f_esr extraction:
350     f_esr = 500             #in kHz
351
352     ## show plots or not (for visual inspection of polynomial fit and impedance extraction)
353     SHOW_PLOT = True
354
355     ##Run main generator function:
356     generate_capacitor(capacitor, vendor, f_esr)
357
358
359 ##### EXAMPLE 2 #####
360 ##### EXAMPLE 2 #####
361 ##### EXAMPLE 2 #####

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell <https://aka.ms/powershell>
PS C:\Users\mmuhu\Documents\PHD\scripts\external_stay\generate_capacitor> conda activate myenv
PS C:\Users\mmuhu\Documents\PHD\scripts\external_stay\generate_capacitor & 'C:\Users\mmuhu\Anaconda3\envs\xternal_stay\generate_capacitor\generate_capacitor.py'

#####
EXTRACTING VALUES FOR: GRM21BC71E106KE11
#####
ESR is: 3.0776 mOhm @500.0000kHz
ESR_min is: 3.0072 mOhm
ESL is: 0.297 nH
#####
Script finished as expected!
#####

The script has now generated verilog-A models for the capacitor **GRM21BC71E106KE11** from **murata** and evaluated the ESR at 500kHz (as indicated by Figure 2).

To stop the script click the red square on top or press **shift+F5**.

You can also just close the two Figures and the script will finish automatically.

Congrats! That is it - you are ready to add more capacitors to the component_database excel file and contribute to a big library of discrete capacitors for all!

Example #1 - find new capacitor and extract Verilog-A model:

To add a new capacitor to the component_database and extract it there are 3 steps:

1. Find desired capacitor and insert entry manually in component_database excel file
2. Download measurement csv data from manufacturer website and saved at correct file location.
3. Run generate_capacitor.py with anaconda environment and with the entry in the user inputs in the script.

Find desired capacitor and add entry to component_database

In this example we will be adding a murata capacitor: # **GRM155R60E106ME16**

The datasheet link on muratas website is: <https://www.murata.com/en-eu/products/productdetail?partno=GRM155R60E106ME16%23>

Here we can see that the relevant data for the database is:

Capacitance: 10uF

Rated voltage: 2.5Vdc

Size: 0402

Go to the component_database excel file and open it:

component_database/component_database.xlsx

1	A	B	C	D	E	F	G	H
	name	value	voltage_rating	ESR	ESL	size	description	datasheet_link
2	GRM155C61E475ME15	4,70E-06	25	0,0045841	1,807E-10 "0402"	Murata Capacitor, 25V rating, size: 0402		
3	GRM155R61H105KE05	1,00E-06	50	0,0106066	2,1E-10 "0402"	Murata Capacitor, 50V rating, size: 0402		
4	GRM155R61J224KE01	2,20E-07	50	0,0198874	2,298E-10 "0402"	Murata Capacitor, 50V rating, size: 0402		
5	GRM188R61E106KA73	1,00E-05	25	0,0034079	2,466E-10 "0603"	Murata Capacitor, 25V rating, size: 0603		
6	GRT188R61H225KE13	2,20E-06	50	0,0053132	3,041E-10 "0603"	Murata Capacitor, 50V rating, size: 0603		
7	GRM21BC71E106KE11	1,00E-05	25	0,0030072	2,967E-10 "0805"	Murata Capacitor, 25V rating, size: 0805		
8	GRM21BR61H106ME43	1,00E-05	50	0,0028834	3,054E-10 "0805"	Murata Capacitor, 50V rating, size: 0805		
9	GC331CD7LP683KX19	6,80E-08	450	0,0125545	6,966E-10 "1206"	Murata Capacitor, 450V rating, size: 1206		
10	GR321BD7LP223KW01	2,20E-08	450	0,0158652	4,168E-10 "0805"	Murata Capacitor, 450V rating, size: 0805		
11	GC332DD7LP154KX18	1,50E-07	450	0,0069525	5,188E-10 "1210"	Murata Capacitor, 450V rating, size: 1210		
12	GC343DD7LP334KX18	3,30E-07	450	0,0051124	5,426E-10 "1812"	Murata Capacitor, 450V rating, size: 1812		
13	GRM32ER71K475KE14	4,70E-06	80	0,0023757	3,247E-10 "1210"	Murata Capacitor, 80V rating, size: 1210		
14	GRM32DC72A475ME01	4,70E-06	100	0,0027382	3,523E-10 "1210"	Murata Capacitor, 100V rating, size: 1210		
15	GRM32EC72A106ME05	1,00E-05	100	0,0018263	3,406E-10 "1210"	Murata Capacitor, 100V rating, size: 1210		
16	GRM32DR72D224KW01	2,20E-07	200	0,0096566	5,308E-10 "1210"	Murata Capacitor, 200V rating, size: 1210		
17	GRM32DR72E224KW01	2,20E-05	250	0,0096566	5,308E-10 "1210"	Murata Capacitor, 250V rating, size: 1210		
18	GRM155R71C223KA01	2,20E-08	16	0,0486246	2,701E-10 "0402"	Murata Capacitor, 16V rating, size: 0402		
19								

It already has a bunch of entries. Add the new capacitor name to the bottom of the list:

	A	B	C	D	E	F	G	H
1	name	value	voltage_rating	ESR	ESL	size	description	datasheet_link
2	GRM155C61E475ME15	4,70E-06	25	0,0045841	1,807E-10 "0402"		Murata Capacitor, 25V rating. size: 0402	
3	GRM155R61H105KE05	1,00E-06	50	0,0106066	2,1E-10 "0402"		Murata Capacitor, 50V rating. size: 0402	
4	GRM155R61J224KE01	2,20E-07	50	0,0198874	2,298E-10 "0402"		Murata Capacitor, 50V rating. size: 0402	
5	GRM188R61E106KA73	1,00E-05	25	0,0034079	2,466E-10 "0603"		Murata Capacitor, 25V rating. size: 0603	
6	GRT188R61H225KE13	2,20E-06	50	0,0053132	3,041E-10 "0603"		Murata Capacitor, 50V rating. size: 0603	
7	GRM21BC71E106KE11	1,00E-05	25	0,0030072	2,967E-10 "0805"		Murata Capacitor, 25V rating. size: 0805	
8	GRM21BR61H106ME43	1,00E-05	50	0,0028834	3,054E-10 "0805"		Murata Capacitor, 50V rating. size: 0805	
9	GC331CD7LP683KX19	6,80E-08	450	0,0125545	6,966E-10 "1206"		Murata Capacitor, 450V rating. size: 1206	
10	GR321BD7LP223KW01	2,20E-08	450	0,0158652	4,168E-10 "0805"		Murata Capacitor, 450V rating. size: 0805	
11	GC332DD7LP154KX18	1,50E-07	450	0,0069525	5,188E-10 "1210"		Murata Capacitor, 450V rating. size: 1210	
12	GC343DD7LP334KX18	3,30E-07	450	0,0051124	5,426E-10 "1812"		Murata Capacitor, 450V rating. size: 1812	
13	GRM32ER71K475KE14	4,70E-06	80	0,0023757	3,247E-10 "1210"		Murata Capacitor, 80V rating. size: 1210	
14	GRM32DC72A475ME01	4,70E-06	100	0,0027382	3,523E-10 "1210"		Murata Capacitor, 100V rating. size: 1210	
15	GRM32EC72A106ME05	1,00E-05	100	0,0018263	3,406E-10 "1210"		Murata Capacitor, 100V rating. size: 1210	
16	GRM32DR72D224KW01	2,20E-07	200	0,0096566	5,308E-10 "1210"		Murata Capacitor, 200V rating. size: 1210	
17	GRM32DR72E224KW01	2,20E-05	250	0,0096566	5,308E-10 "1210"		Murata Capacitor, 250V rating. size: 1210	
18	GRM155R71C223KA01	2,20E-08	16	0,0486246	2,701E-10 "0402"		Murata Capacitor, 16V rating. size: 0402	
19	GRM155R60E106ME16	1,00E-05	2,5		"0402"		Murata Capacitor, 2,5V rating. size: 0402	

Do **not** add anything to the ESR and ESL column. These values are extracted from the script and automatically inserted whenever you run generate_capacitor.py.

Save and close the component_database.xlsx file

Download measurement csv data from manufacturer website and saved at correct file location:

Now we need to download the csv data from the Murata webpage and save it in the correct folder.

First let's create the folder for our capacitor. Navigate to input -> murata and create a folder with the name of the capacitor:

File Home Share View

generate_capacitor < input > murata

Search murata

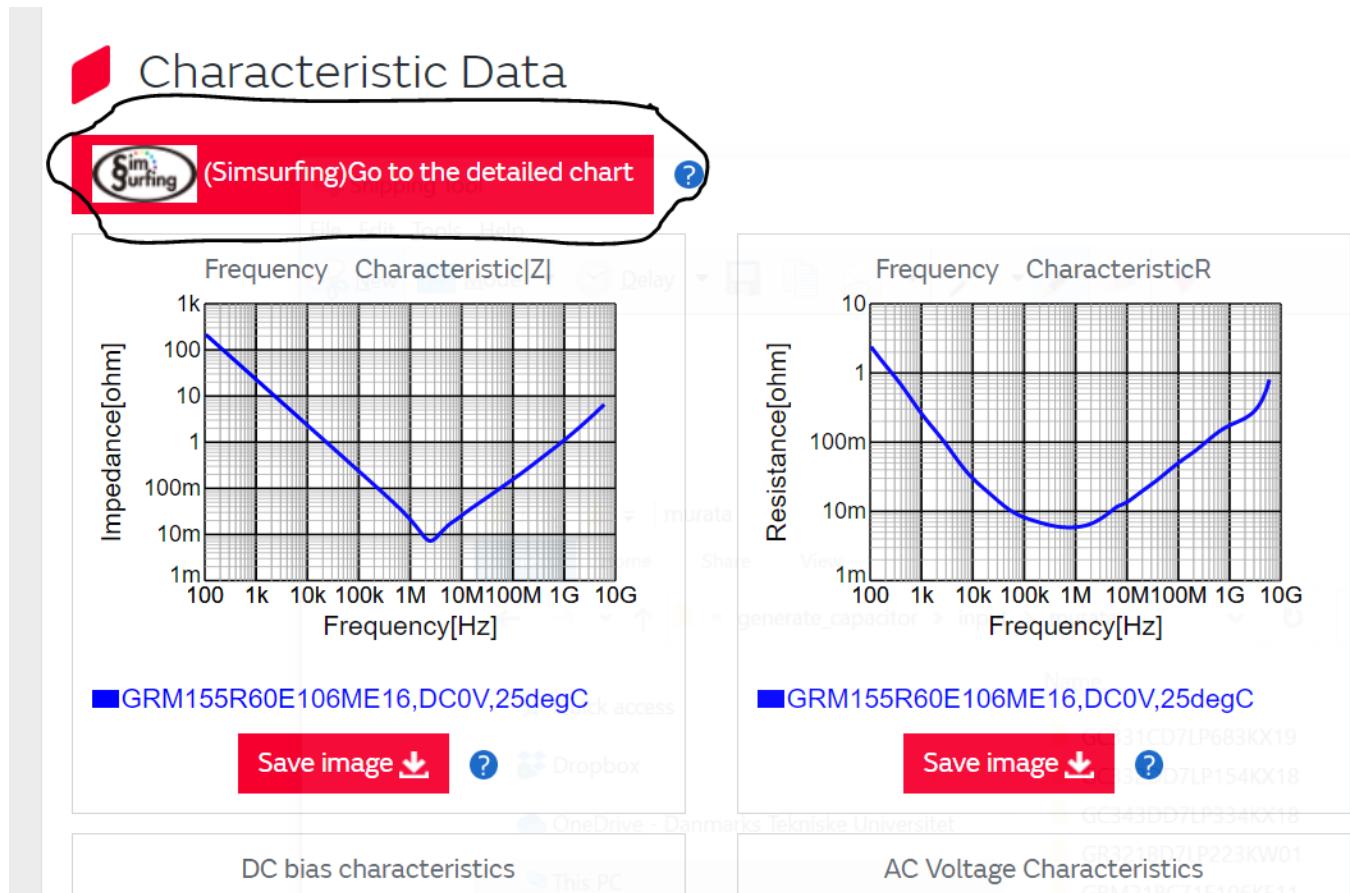
Name	Date modified	Type	Size
GC331CD7LP683KX19	20-02-2023 18:15	File folder	
GC332DD7LP154KX18	20-02-2023 18:15	File folder	
GC343DD7LP334KX18	20-02-2023 18:15	File folder	
GR321BD7LP223KW01	20-02-2023 18:15	File folder	
GRM21BC71E106KE11	20-02-2023 18:15	File folder	
GRM21BR61H106ME43	20-02-2023 18:15	File folder	
GRM32DC72A475ME01	20-02-2023 18:15	File folder	
GRM32DR72D224KW01	20-02-2023 18:15	File folder	
GRM32DR72E224KW01	20-02-2023 18:15	File folder	
GRM32EC72A106ME05	20-02-2023 18:15	File folder	
GRM32ER71K475KE14	20-02-2023 18:15	File folder	
GRM155C61E475ME15	20-02-2023 18:15	File folder	
GRM155R61H105KE05	20-02-2023 18:15	File folder	
GRM155R61J224KE01	20-02-2023 18:15	File folder	
GRM155R71C223KA01	20-02-2023 18:15	File folder	
GRM188R61E106KA73	20-02-2023 18:15	File folder	
GRT188R61H225KE13	20-02-2023 18:15	File folder	
GRM155R60E106ME16	20-02-2023 19:30	File folder	

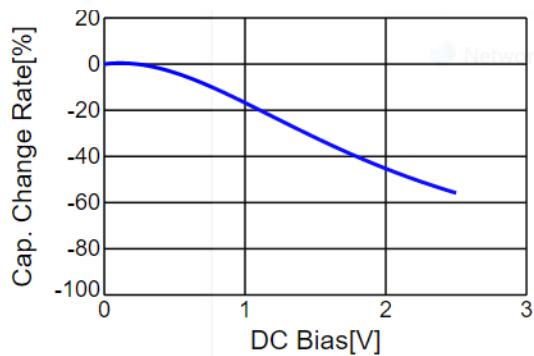
This is where we will add our csv data.

To download the .csv data go to the murata capacitor datasheet again:

<https://www.murata.com/en-eu/products/productdetail?partno=GRM155R60E106ME16%23>

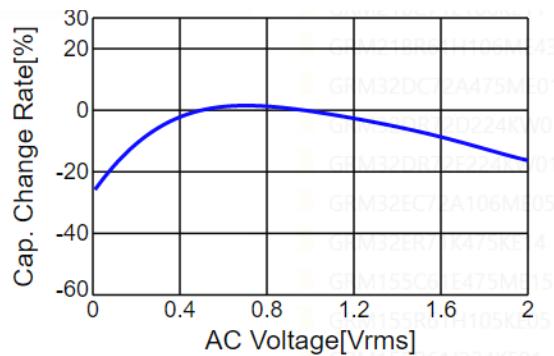
Scroll down until you see the "Simsurfing" characteristic data:





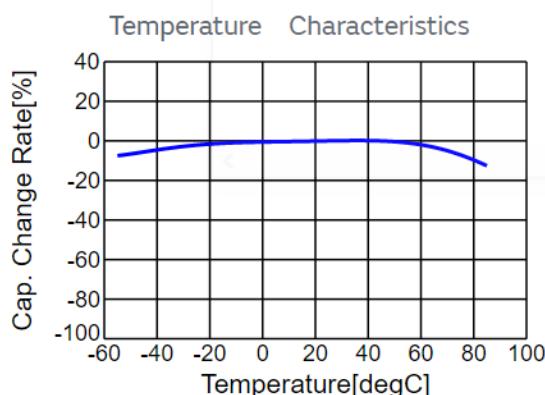
GRM155R60E106ME16, capchange,
25.0degC, AC0.5Vrms

[Save image](#) [?](#)



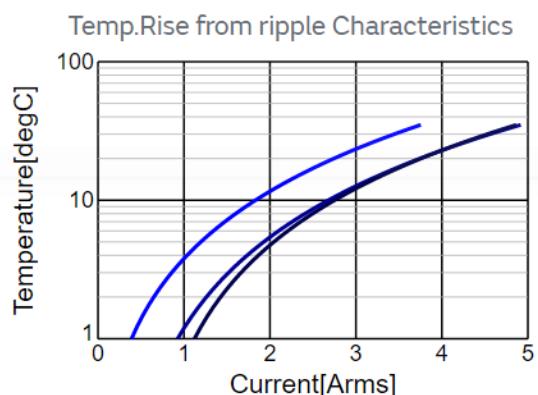
GRM155R60E106ME16, capchange, DC0V,
25.0degC

[Save image](#) [?](#)



GRM155R60E106ME16, capchange,
DC0.0V, AC0.2Vrms

[Save image](#) [?](#)



GRM155R60E106ME16

80kHz 400kHz 800kHz

[Save image](#) [?](#)

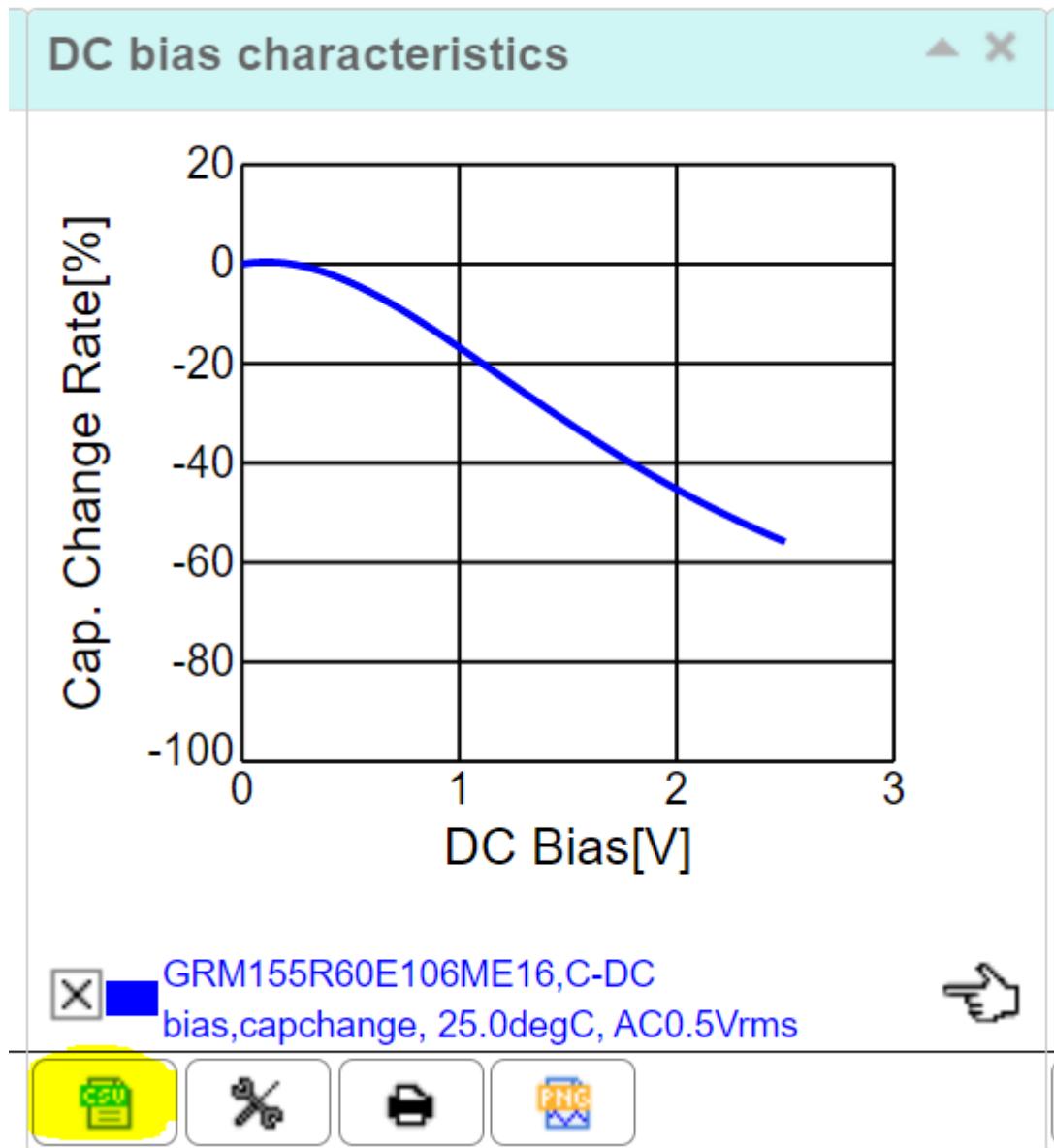
Note: Not all capacitors have characteristic data in the website and sometimes the DC bias characteristics are not measured. If that is the case - too bad. We cannot generate a proper model for it.

Click the red (**SimSurfing**)[Go to detailed chart](#) button. This opens a new tab. Press "Agree" on the License Agreement.

From here we can download the csv data for the measurement results. We need to download three individual files:

1. DC bias characteristics
2. AC impedance frequency characteristic
3. ESR frequency characteristic

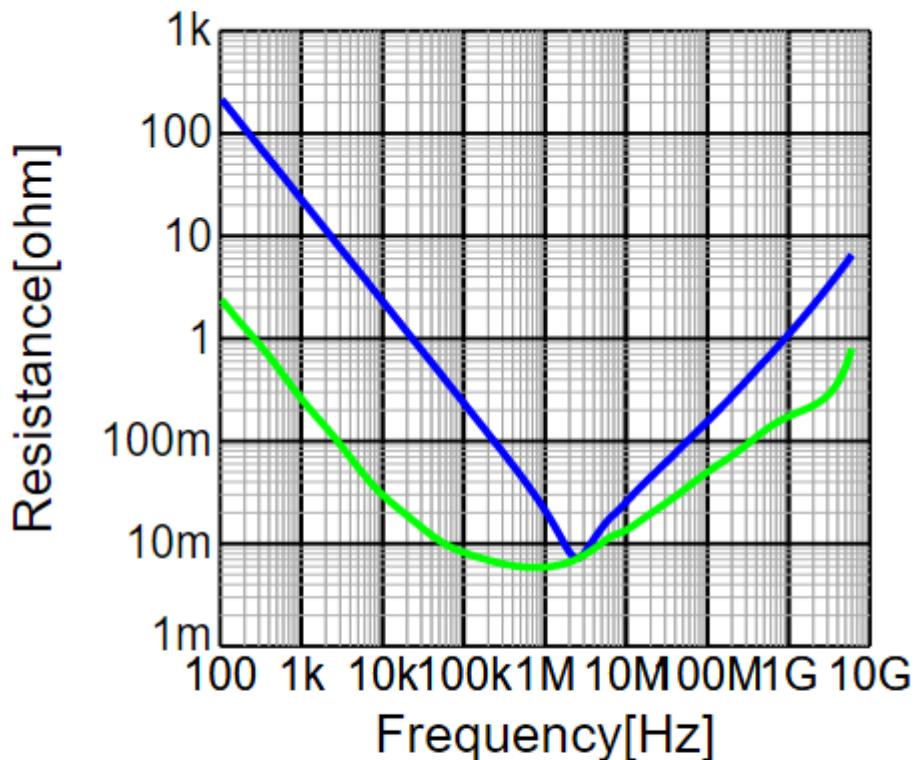
First download the DC bias characteristic by pressing the small "csv" icon underneath the graph:



save the file in the folder you created earlier and name it: GRM155R60E106ME16_dc_bias

Now for the AC impedance and the ESR frequency characteristics these are both located on the same figure:

Frequency Characteristic



GRM155R60E106ME16,|Z|,DC0V,25degC

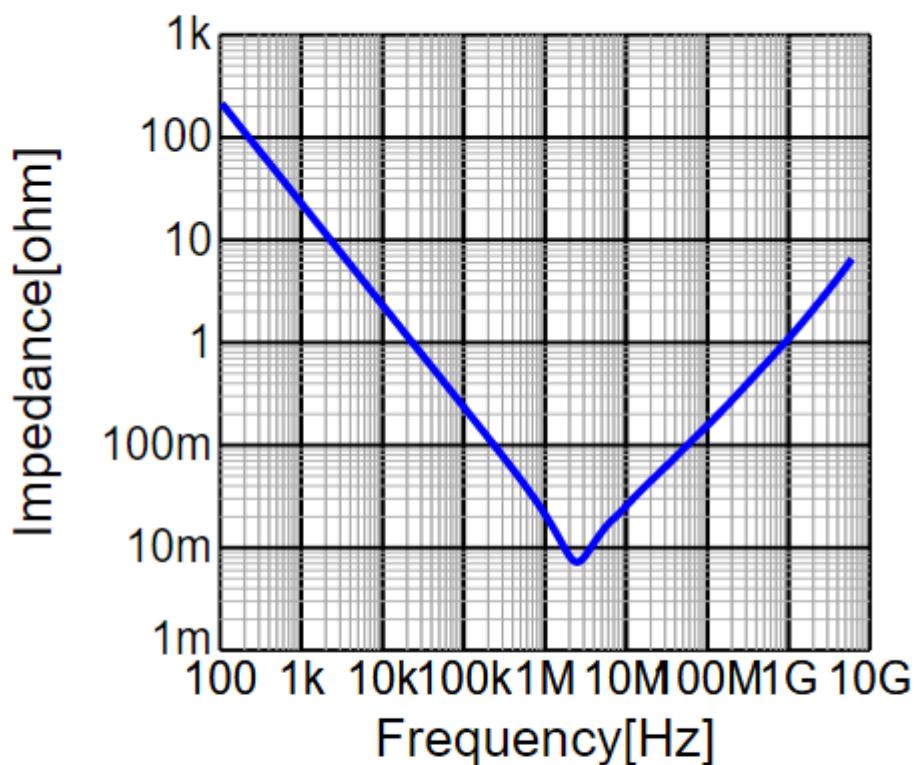
GRM155R60E106ME16,R,DC0V,25degC



We want to download the two graphs in two separate files. The webpage is a bit annoying though, since pressing the close button [X] removes the graph completely.

I usually open the website in two tabs and close the ESR in one of them and the AC impedance in the other:

Frequency Characteristic

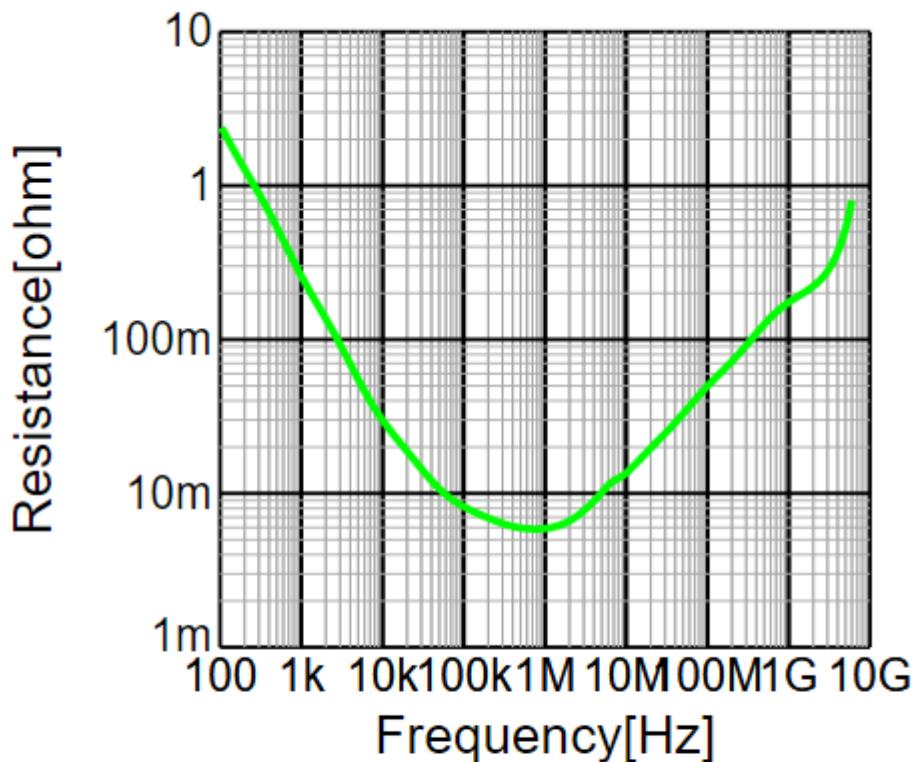


GRM155R60E106ME16,|Z|,DC0V,25degC



Frequency Characteristic

▲ X



GRM155R60E106ME16,R,DC0V,25degC 



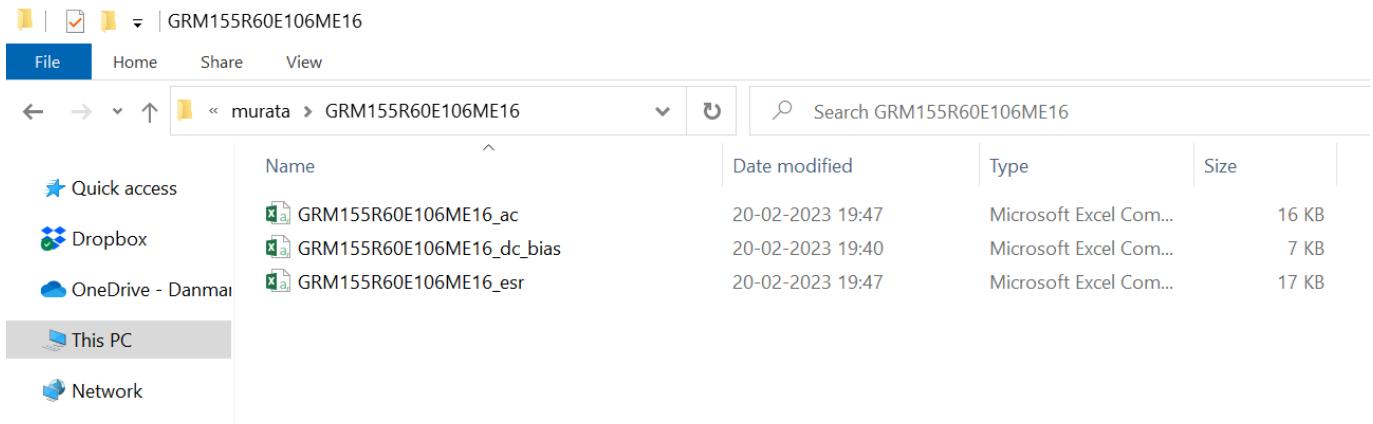
Save the csv data for both graphs to the folder you created earlier. Ensure that they have the correct filenames:

For AC impedance characteristics: GRM155R60E106ME16_ac

For ESR frequency characteristics: GRM155R60E106ME16_esr

This should ALWAYS be ensured when saving a new file. Since otherwise the generate_capacitor.py script cannot find the correct files.

Now your folder should look like this:



and we're ready to proceed!

Run `generate_capacitor.py` with anaconda environment and with the entry in the user inputs in the script

Now we are ready to generate the verilog-a files.

Ensure that you have closed the component_basebase excel file first!

Open anaconda, choose your python environment and open VSCode (as explained in the user guide above)

At the bottom of the `generate_capacitor.py` script there is room for user inputs:

```

✓ if __name__ == "__main__":
    #####
    ##### EXAMPLE 1: #####
    #####
    ### Generate verilog-A files for a GRM21BC71E106KE11 - murata capacitor.
    ### Before this script is run the following has been done by the user:
    # 1:   The user has found the component on the vendor webpage and downloaded the
    #       the csv data for: ac impedance measurement, ac_esr measurement and the
    #       capacitance DC bias voltage degradation
    #
    # 2:   The user has inserted ATLEAST the capacitor name and nominal capacitance
    #       value in the component_database.xlsx excel file. The ESR and ESL values
    #       should not be inserted, since they are extracted from this script.
    #
    # 3:   The user has ensured that the csv data is saved under the correct names
    #       and in the correct path as specified by the user guide.
    #
    # 4:   Remember to close the component_database.xlsx file before running the
    #       the script. Otherwise the script cannot open and edit the file.

    ##### USER INPUTS #####
    ### choose manufacture vendor:
    vendor = "murata"

    # part number:
    capacitor = "GRM155R60E106ME16"

    # Frequency for f_esr extraction:
    f_esr = 500           #in kHz

    ### show plots or not (for visual inspection of polynomial fit and impedance extraction)
    SHOW_PLOT = True

    #Run main generator function:
    generate_capacitor(capacitor, vendor, f_esr)

```

change the capacitor variable to our new desired capacitor (as seen above) and run the script (run -> Run Without Debugging) or by pressing **ctrl+F5**:

```

319 if __name__ == "__main__":
320
321 ##### EXAMPLE 1: #####
322 ##### EXAMPLE 1: #####
323 ##### EXAMPLE 1: #####
324
325     ## Generate verilog-A files for a GRM21BC71E106KE11 - murata capacitor.
326     ### Before this script is run the following has been done by the user:
327     # 1: The user has found the component on the vendor webpage and downloaded the
328     #     the csv data for: ac impedance measurement, ac_esr measurement and the
329     #     capacitance DC bias voltage degradation
330     #
331     # 2: The user has inserted ATLEAST the capacitor name and nominal capacitance
332     #     value in the component_database.xlsx excel file. The ESR and ESL values
333     #     should not be inserted, since they are extracted from this script.
334     #
335     # 3: The user has ensured that the csv data is saved under the correct names
336     #     and in the correct path as specified by the user guide.
337     #
338     # 4: Remember to close the component_database.xlsx file before running the
339     #     the script. Otherwise the script cannot open and edit the file.
340
341 ##### USER INPUTS #####
342
343     ### choose manufacture vendor:
344     vendor = "murata"
345
346     # part number:
347     capacitor = "GRM155R60E106ME16"
348
349
350     # Frequency for f_esr extraction:
351     f_esr = 500      #in kHz
352
353     ### show plots or not (for visual inspection of polynomial fit and impedance
354     SHOW_PLOT = True
355
356     #Run main generator function:
357     generate_capacitor(capacitor, vendor, f_esr)
358
359
360 ##### EXAMPLE 2: #####
361 ##### EXAMPLE 2: #####

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

ESL is: 0.297 mH

Script finished as expected!

[C:\Users\mhhe\Documents\PHD\scripts\external_stay> c:\Users\mhhe\Documents\PHD\scripts\external_stay>cd ..\..\debugpy\launcher" "55847" >>> "c:\Users\mhhe\Documents\PHD\scripts\external_stay>##### EXTRACTING VALUES FOR: GRM155R60E106ME16 #####

[WinError 183] Cannot create a file when that file already exists: 'output'

ESR is: 5.9681 mOhm @ 500.000kHz

ESR_min is: 5.8607 mOhm

ESL is: 0.182 mH

Script finished as expected!

The script should finish, write the extracted values to the component_database excel file and show figures with the polynomial fit for the capacitor degradation value and two points where ESL and ESR are calculated.

Boom! That's it! You've generated you're first verilog-A model for your capacitor.

To view the verilog-A model navigate to: output -> GRM155R60E106ME16

GRM155R60E106ME16				
File	Home	Share	View	
◀ ▶ ⌂ ⌃ ⌄	output > GRM155R60E106ME16	▼	⟳	🔍 Search GRM155R60E106ME16
Quick access	Name	Date modified	Type	Size
Dropbox	veriloga	20-02-2023 19:53	VA File	2 KB
OneDrive - Danmark	veriloga_rc	20-02-2023 19:53	VA File	2 KB
This PC	veriloga_rc_500khz	20-02-2023 19:53	VA File	2 KB
Network	veriloga_rlc	20-02-2023 19:53	VA File	2 KB
	veriloga_rlc_500khz	20-02-2023 19:53	VA File	2 KB

here are all the veriloga models. Then you just need to copy these to cadence, create a new view with a symbol and copy the veriloga file into the view.

Example #2 - extract Verilog-A models for all entries in database:

To quickly generate **ALL** the capacitors which are already inserted in the component_database excel file (and which have the download measurement .csv data), I've created a function in the generate_capacitor.py script that is called make_all().

Scroll down to the bottom of generate_capacitor.py and comment out the "generate_capacitor(capacitor, vendor, f_esr)" line from Example 1.

Also uncomment the "make_all(f_esr)" line under Example 2:

```

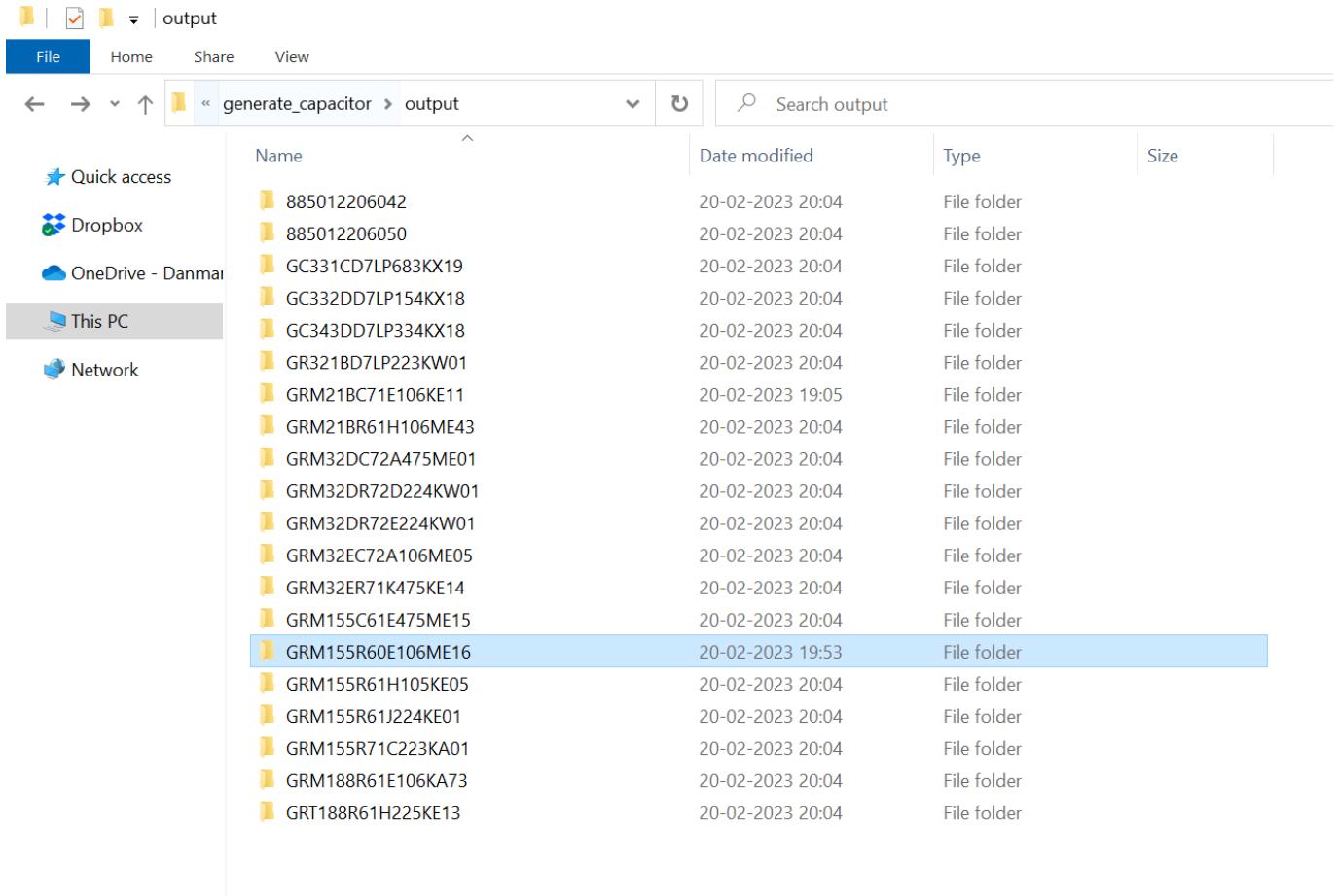
319 if __name__ == "__main__":
320
321 ##### EXAMPLE 1: #####
322 ##### EXAMPLE 1: #####
323 ##### EXAMPLE 1: #####
324
325     ### Generate verilog-A files for a GRM21BC71E106KE11 - murata capacitor.
326     ### Before this script is run the following has been done by the user:
327     # 1:    The user has found the component on the vendor webpage and downloaded the
328     #        the csv data for: ac impedance measurement, ac_esr measurement and the
329     #        capacitance DC bias voltage degradation
330     #
331     # 2:    The user has inserted ATLEAST the capacitor name and nominal capacitance
332     #        value in the component_database.xlsx excel file. The ESR and ESL values
333     #        should not be inserted, since they are extracted from this script.
334     #
335     # 3:    The user has ensured that the csv data is saved under the correct names
336     #        and in the correct path as specified by the user guide.
337     #
338     # 4:    Remember to close the component_database.xlsx file before running the
339     #        the script. Otherwise the script cannot open and edit the file.
340
341 ##### USER INPUTS #####
342
343     ### choose manufacture vendor:
344     vendor = "murata"
345
346     # part number:
347     capacitor = "GRM155R60E106ME16"
348
349
350     # Frequency for f_esr extraction:
351     f_esr = 500          #in kHz
352
353     ### show plots or not (for visual inspection of polynomial fit and impedance extraction)
354     SHOW_PLOT = True
355
356     #Run main generator function:
357     #generate_capacitor(capacitor, vendor, f_esr)
358
359
360 ##### EXAMPLE 2: #####
361 ##### EXAMPLE 2: #####
362 ##### EXAMPLE 2: #####
363
364     ### Generate verilog-A files for all available in component_database.
365     #    if errors exist in database or missing csv data, the function should
366     #    report a WarningInfo and skip these.
367
368     f_esr = 500          #in kHz
369     make_all(f_esr)

```

Rerun the script by: Run -> Run Without Debugging or by pressing **ctrl+F5**.

The `make_all` function simply iterates through the `component_database` excel file and creates a verilog-a model for each entry that also has the csv data. The script takes a minute or so to run.

After running the script go to the output folder and check all the new models:



Name	Date modified	Type	Size
885012206042	20-02-2023 20:04	File folder	
885012206050	20-02-2023 20:04	File folder	
GC331CD7LP683KX19	20-02-2023 20:04	File folder	
GC332DD7LP154KX18	20-02-2023 20:04	File folder	
GC343DD7LP334KX18	20-02-2023 20:04	File folder	
GR321BD7LP223KW01	20-02-2023 20:04	File folder	
GRM21BC71E106KE11	20-02-2023 19:05	File folder	
GRM21BR61H106ME43	20-02-2023 20:04	File folder	
GRM32DC72A475ME01	20-02-2023 20:04	File folder	
GRM32DR72D224KW01	20-02-2023 20:04	File folder	
GRM32DR72E224KW01	20-02-2023 20:04	File folder	
GRM32EC72A106ME05	20-02-2023 20:04	File folder	
GRM32ER71K475KE14	20-02-2023 20:04	File folder	
GRM155C61E475ME15	20-02-2023 20:04	File folder	
GRM155R60E106ME16	20-02-2023 19:53	File folder	
GRM155R61H105KE05	20-02-2023 20:04	File folder	
GRM155R61J224KE01	20-02-2023 20:04	File folder	
GRM155R71C223KA01	20-02-2023 20:04	File folder	
GRM188R61E106KA73	20-02-2023 20:04	File folder	
GRT188R61H225KE13	20-02-2023 20:04	File folder	

and that's it!

Known limitations

Wurth not having csv data to download from website.

Wurth does show interactable figures of their measurements but does not have the option to download the data as csv data.

The best workaround, that I've found for this is to:

1. Download the PNG pictures for dc_bias, ac characteristics and esr characteristics
2. use: [WebPlotDigitizer - Extract data from plots, images, and maps](#) to extract the data based on the figures and save it as a csv file.

The process is a bit slower, but it is possible to get the data! Maybe if someone pokes Wurth they could just share the csv data on their website.