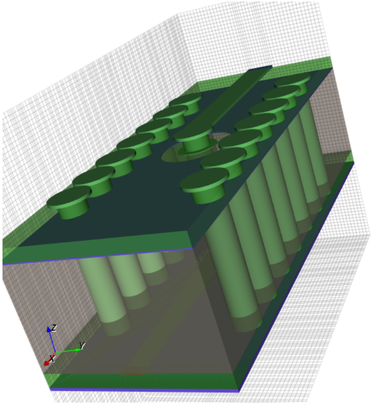
Microwave Design with KiCad and OpenEMS

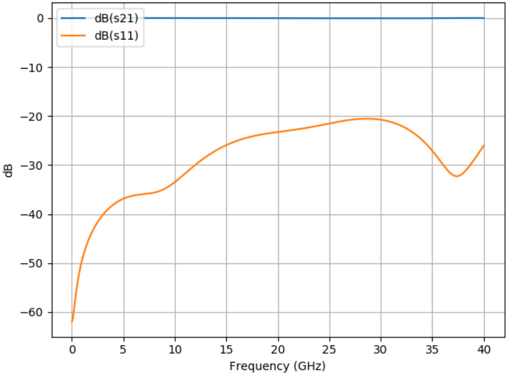
Darrell Harmon

April 26 2019

OpenEMS

- Finite Difference Time Domain (FDTD) 3D electromagnetic field solver
- GPLv3 license
- Matlab/Octave or Python scripts describe geometry, ports, etc
- 3D viewer (CSXCAD) for resulting model





pyOpenEMS

- written in Python 3, fairly small and simple
- originally a wrapper for OpenEMS which generated an Octave script and ran it
- OpenEMS is adding Python support in the next release
- generates a KiCad module for most PCB compatible geometry (planar, vias)
- has some helper functions and classes for common geometry used in a PCB (via, plane with hole, etc)
- being simplified to be a very thin wrapper only requiring minor changes to standard OpenEMS scripts
- primarily a tool for use at Harmon Instruments

Low Pass Filter (examples/LPF_DGS_bowtie_7)

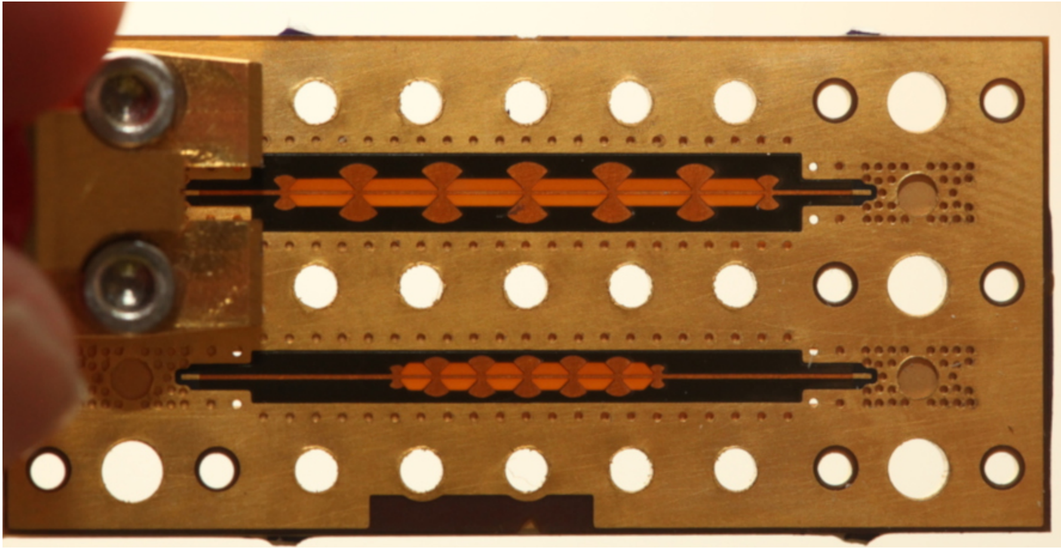
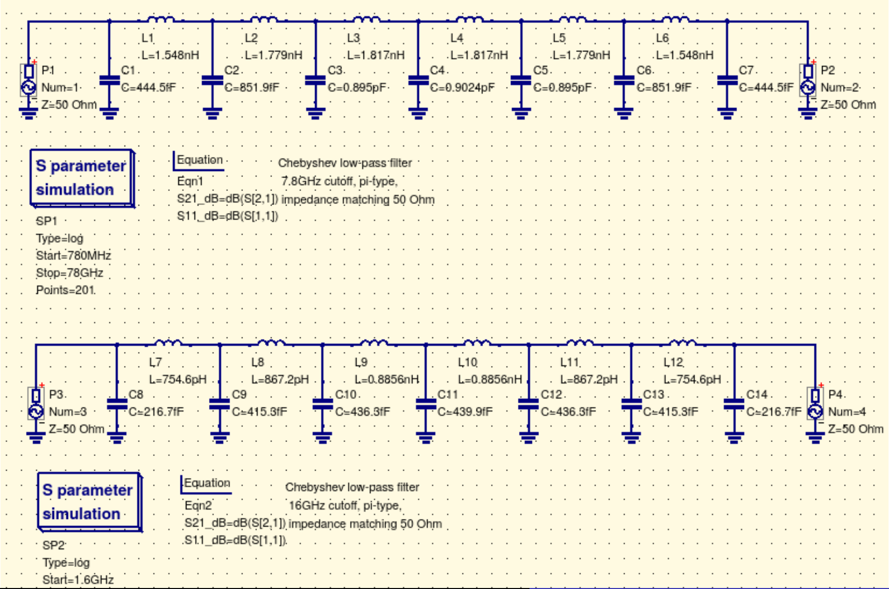


Figure 1: photo of filters

Lumped filter prototype - QUCS



Distributed element filter

- No parts
- circuit board traces and vias
- can be implemented on a PCB, in metal, etc

Material

- OSHPark flex https://docs.oshpark.com/services/flex/
- 102 µm (4 mil) Polyimide (Panasonic Felios)
- ε_r = 3.2, relative permittivity or dielectric constant
- Df = 0.002, dissipation factor as in a capacitor
- Copper is 35 µm each side
- Mask is about 25 um thick, Er = 3.3, Df = 0.020 (guess, mask not specified)
- Mask is retained over the circuit to avoid losses associated with ENIG plating
- moisture absorption causes dielectric properties to shift

```
metal = openems.Metal(em, 'pec') # lossless metal
sub = openems.Dielectric(em, 'polyimide', eps_r=3.2,
                         tand=0.002, fc=fc)
mask = openems.Dielectric(em, 'soldermask', eps_r=3.3,
                          tand=0.020, fc=fc)
```

Capacitors

- Pairs of radial stubs commonly referred to as a bowtie or butterfly
- enables better low pass performance than rectangles used in a stepped impedance filter
- The choice of the thin substrate reduces the required area for the stubs

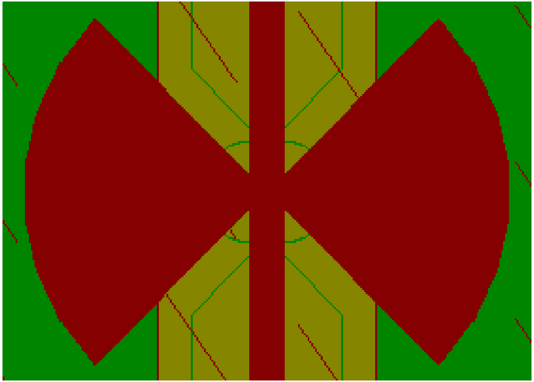


Figure 2: bowtie stub

Inductors

A high impedance line can be used to approximate an inductor. On OSHPark flex, the highest achievable impedance is about 60 ohms with the design rule minimum trace over a ground plane. Adding the ground cutout increases the impedance to around 200 ohms.

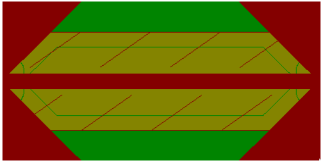
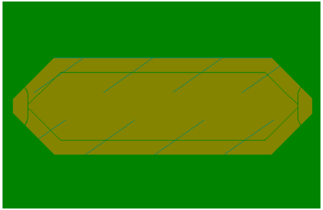
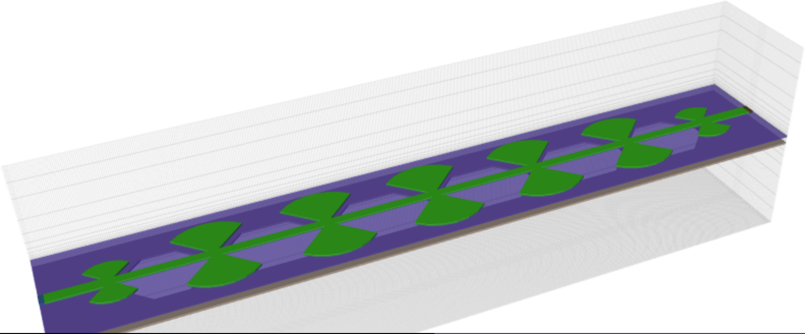


Figure 3: inductor top layer

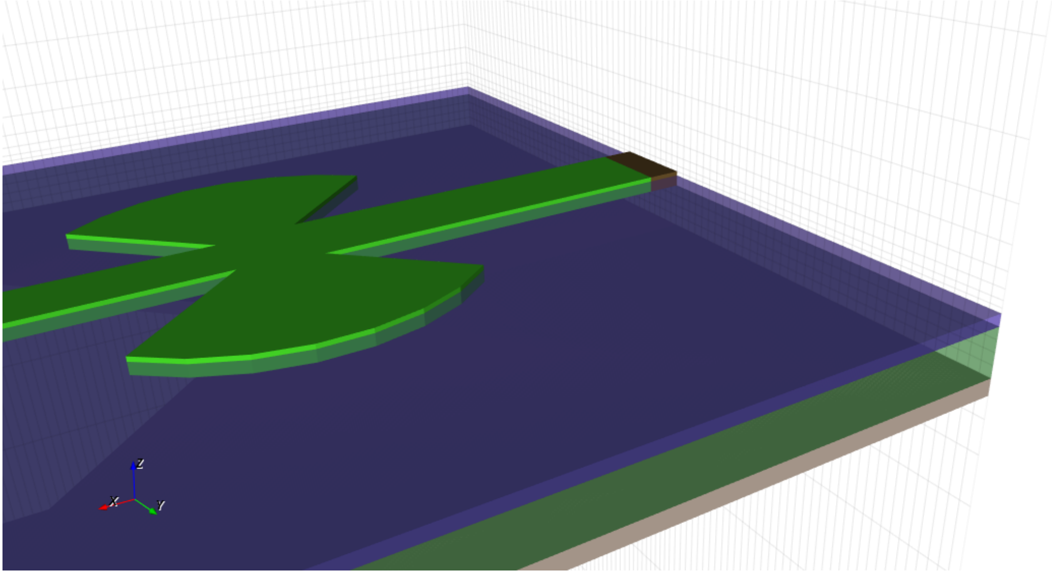


Simulation Box

- Boundaries default to PEC (perfect electrical conductor), lossless metal
- Other available boundaries include PMC (useful for symmetry), PML (absorptive), etc
- Unused space is vacuum $\varepsilon_r = 1.0$, close enough to air $\varepsilon_r \approx 1.0006$
- Best to break a problem up into pieces - don't model an entire board



Ports

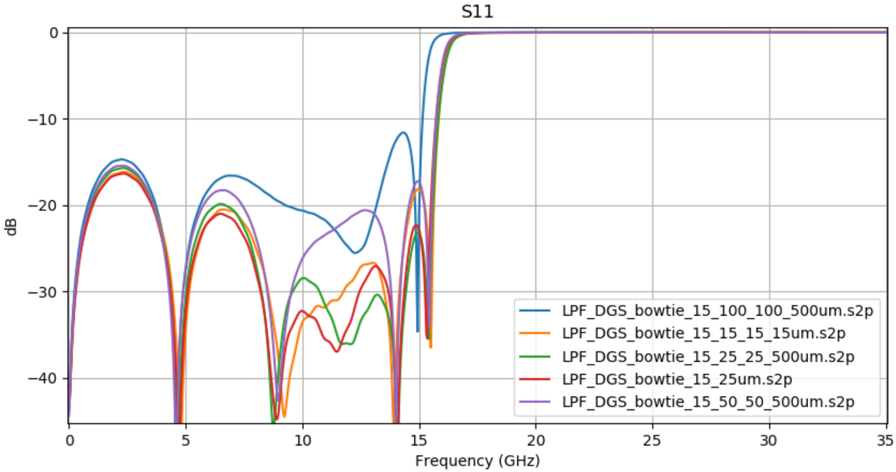


Mesh

- The mesh does not have to be uniform
- Smaller cells decrease the minimum time step. Halving the cell size in x, y and z can result in a 16x increase in run time.
- Times on an i7-3820 PC, 8 threads, x, y, z are maximum cell size in μm

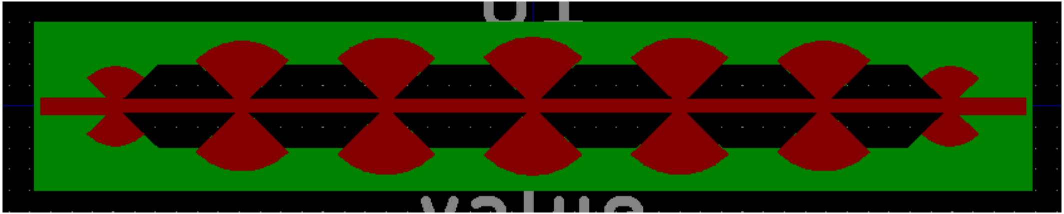
x	y	z	cells	run time MM:SS
15	15	15	25.6 M	144:48
25	25	25	5.7 M	14:29
25	25	500	1.1 M	03:16
50	50	500	293 k	00:38
100	100	500	95 k	00:21

Mesh



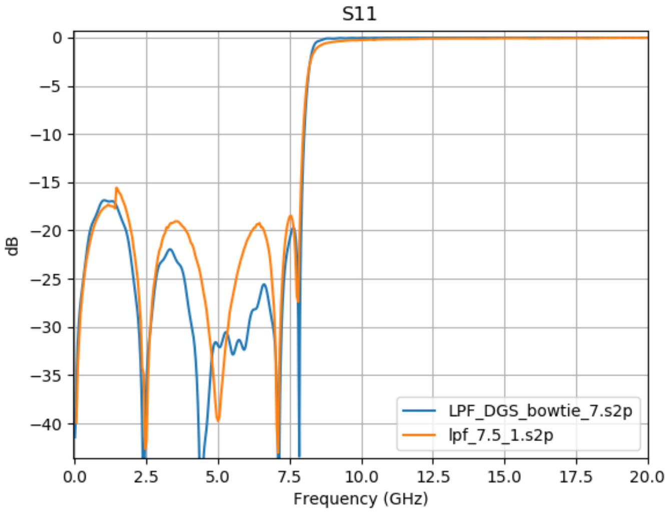
S11
0
−10
−20
−30
−40
dB
0
5
10
15
20
25
30
35
Frequency (GHz)
LPF_DGS_bowtie_15_100_100_500um.s2p
LPF_DGS_bowtie_15_15_15_15um.s2p
LPF_DGS_bowtie_15_25_25_500um.s2p
LPF_DGS_bowtie_15_25um.s2p
LPF_DGS_bowtie_15_50_50_500um.s2p

KiCad Footprint

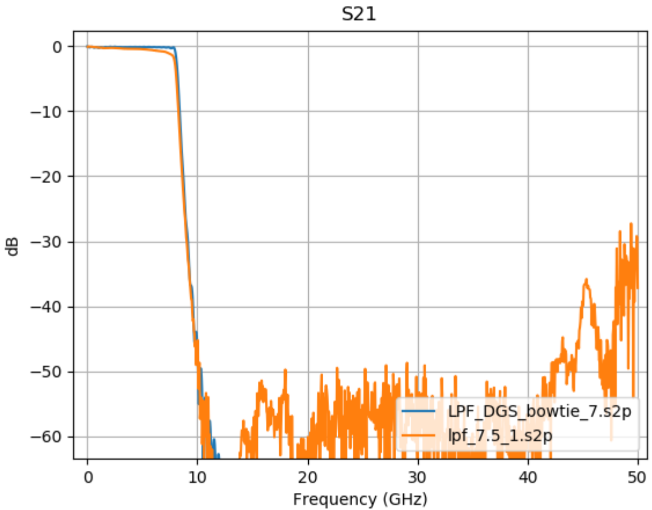


- Ground plane is composed of two overlapping custom shape pads
- Top copper is a polygon and two normal pads, net tie
- Requires adding a polygon pour keepout on the bottom layer

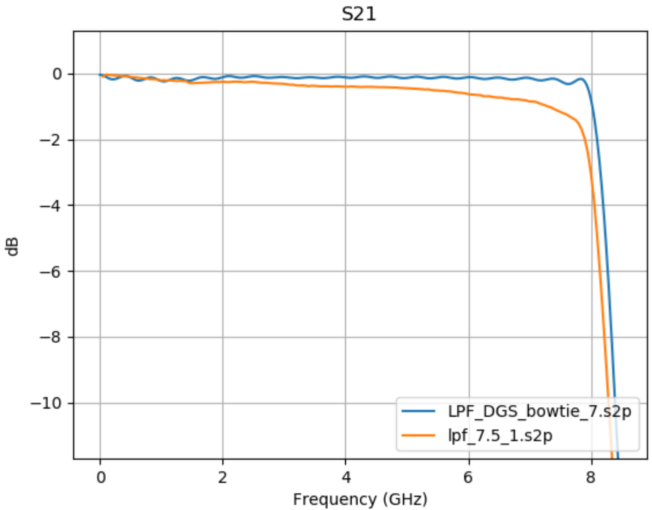
Results - 7.5 GHz S11 (reflection)



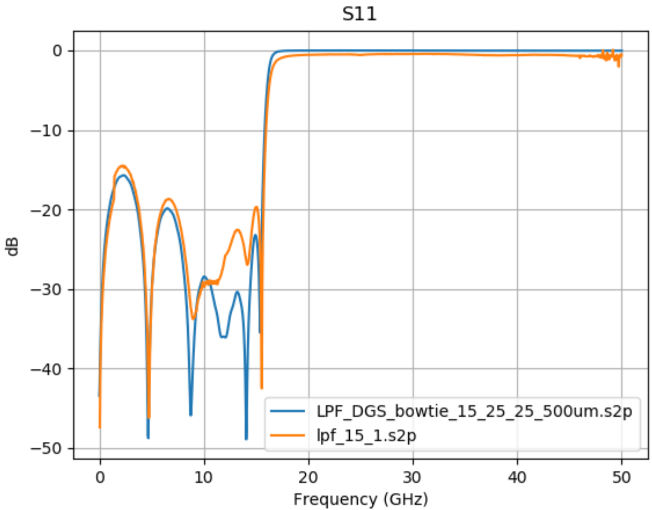
Results - 7.5 GHz S21 (transmission)



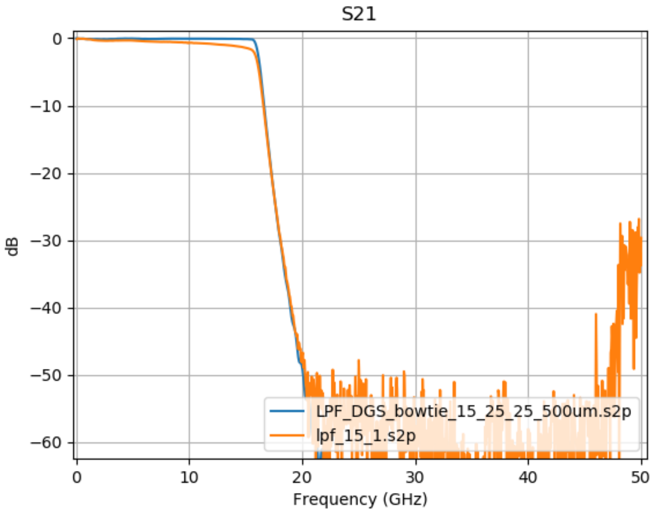
Results - 7.5 GHz S21 (transmission)



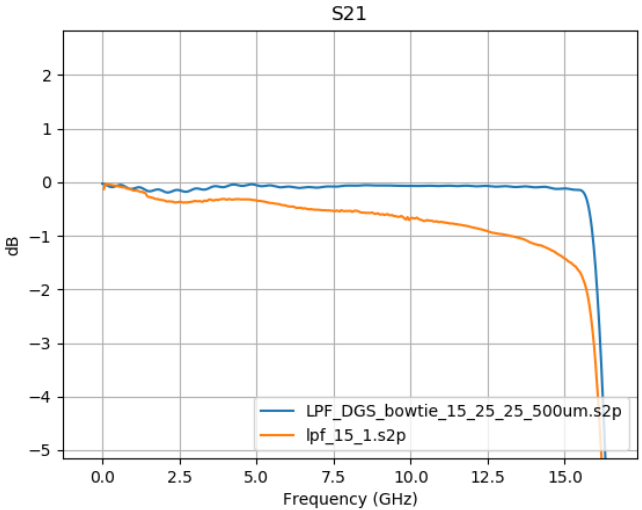
Results - 15 GHz S11 (reflection)



Results - 15 GHz S21 (transmission)

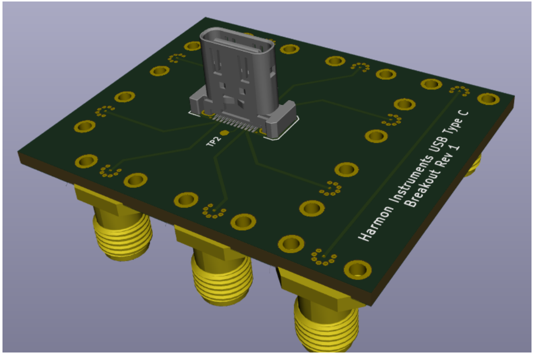


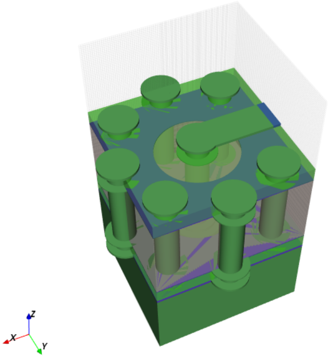
Results - 15 GHz S21 (transmission)



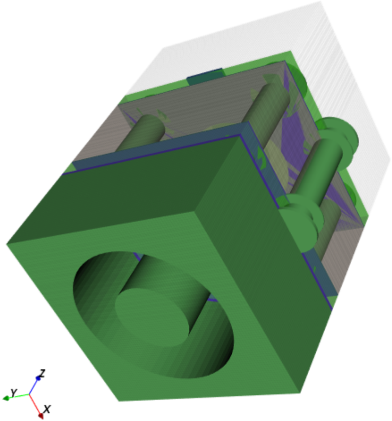
Connector

- examples/vert_connector_ms_oshpark.py
- 2.92 mm compression mount

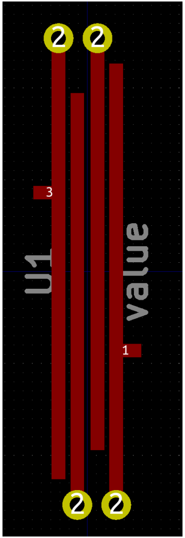


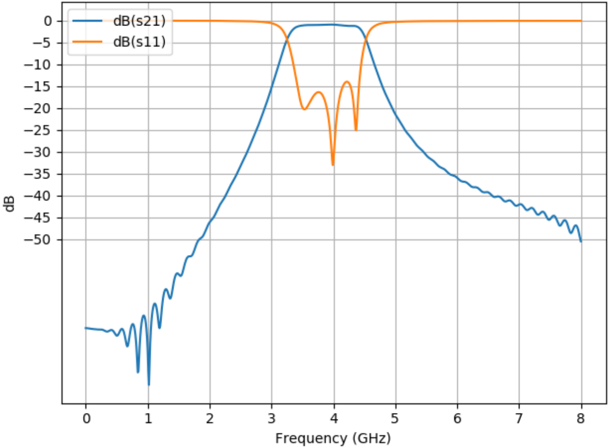


Connector - continued



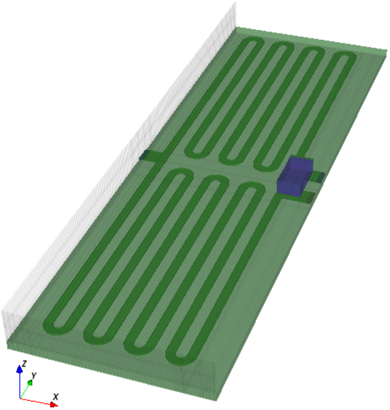
Interdigital Bandpass

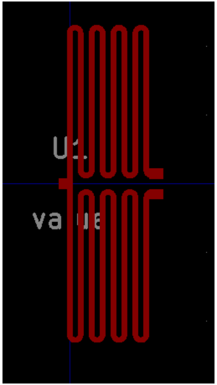


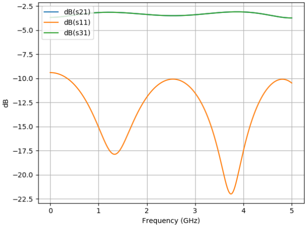


Wilkinson Splitter

- examples/wilkinson_1.py







Future work

- Documentation
- move many models from private repository to examples

Bend example

- Live demo if time allows

Links

Software

- pyopenems https://github.com/dlharmon/pyopenems
- OpenEMS https://openems.de
- ATLC http://atlc.sourceforge.net
- QUCS http://qucs.sourceforge.net

Darrell Harmon

- Blog https://harmoninstruments.com/blog/
- Twitter: harmoninst
- https://harmoninstruments.com