

LW RF Studio - User Manual

Version 1.2.6 · Nov 6, 2026 · Luc Wathelet



LW RF Studio - User Manual 1
1. Introduction 5
2. Installation and first start 6
Install LW RF Studio on Windows 6
Install LW RF Studio on Linux (Ubuntu) 6
Install LW RF Studio on macOS 6
Install openEMS 6
First start 7
3. The main window: PCB Antenna Lab 8
Menus 9
Toolbar 11
Views 11
Geometry view 11
Grid, snap and moving shapes 13
The Geometry toolbar 15
Parameters from shapes 16
Units 16
4. Designing an antenna 17
Start from a template 17
Parameters 18
Shapes 18
Feeds 19
Vias 19
Holes 20
Stackup 20
Volumes: metal parts and a casing 20
Model errors 22

Undo and redo 22
5. Importing designs 22
PCB boards: Eagle, Fusion, KiCad and Altium 23
Gerber and drill files 24
MATLAB pcbStack scripts 24
6. Simulation settings 24
Fast, Standard and Accurate presets 25
Settings 25
GPU engine 27
Remote GPU (beta) 27
Making a run faster 28
7. Running a simulation 28
How a run works 28
Progress 28
When it finishes 29
The simulation folder 29
Results saved with the project 29
If the run stops with an error 29
If the program meets an error 30
Tuning to a frequency 30
8. Results: S11, impedance, VSWR and Smith chart 31
Reading values on the charts 33
Comparing with a measurement 34
Matching network 34
RF calculator 35
9. Radiation pattern 36
The 2D pattern 36
The 3D pattern 37
10. Current distribution 37
Reading the view 38
Controls 38
How it is computed 38
11. Exporting 39

The antenna as a PCB part 39
Results 39
Images 39
Reports 40
openEMS model 40
12. SmithMatch 40
Opening SmithMatch 40
The window 40
Designing a network 41
Würth design kits 41
VSWR sweep, files and printing 41
13. Command line 42
Example: a quick sweep 43
14. KiCad plugin: LW Antenna Check 44
15. Trial and license 46
Getting a license 46
Good to know 47
Licence messages 47
Legal 48
16. Tips and troubleshooting 48
A tuning workflow 48
Why simulation and measurement differ 48
Problems and solutions 49
17. Quick reference 50
Keyboard 50
Mouse 51
Expressions 51
Files 52
Contact 52

1. Introduction

LW RF Studio is RF engineering software for Windows, Linux and macOS by lw tech, a brand of Wattlet AS. It brings two tools together in one program, with one installer and one license:

- **PCB Antenna Lab**, the main window, designs printed antennas (inverted-F IFA, meandered MIFA and other planar antennas) and simulates them with openEMS, a free full-wave FDTD solver. You describe the antenna with named dimensions, run a simulation and read S11, impedance, VSWR, the Smith chart, the radiation pattern and the current on the copper, much as in MATLAB's PCB Antenna Designer.
- **SmithMatch** designs L/C matching networks on an interactive Smith chart, with the values of the Würth Elektronik design kits (section 12). PCB Antenna Lab opens it with **Tools > SmithMatch** (Ctrl+M) and can send it the simulated antenna impedance.

With PCB Antenna Lab you can:

- Start from a template, or import a MATLAB pcbStack script, a board from Eagle, Autodesk Fusion, KiCad or Altium, or its Gerber files.
- Change any dimension and see the geometry update at once; sweep one parameter with the tuning slider.
- Add metal parts and a plastic casing around the board (Volumes tab), and see their effect on the antenna.
- Draw with the mouse: a grid with snap, moving and resizing shapes, and Undo and Redo.
- Let **Simulate > Tune to frequency** find the length and the match that put the resonance and the best VSWR at your frequency.
- See the board in 2D, in 3D and as a realistic picture to scale.
- Simulate in openEMS with live progress, compare with a measured VNA trace and try an L/C matching network, or design one in SmithMatch.
- Export the antenna as a KiCad footprint, an Eagle library or an Altium footprint library.
- Run the simulations on the GPU of an NVIDIA graphics card, 10 to 15 times faster than on the CPU (section 6).
- Run them on the GPU of another PC through lwtech.org, where the LW RF Studio Agent runs them (Remote GPU, beta: section 6).

The program runs for 30 days from the first start on a PC (the trial), with every feature. After that it needs a license for that PC. Section 14 explains the trial and how to get a license.

Requirements: Windows 10 or 11 (64-bit), Linux x64 (Ubuntu 22.04 or later) or macOS 13 or later (Apple silicon or Intel) and, for the simulations, openEMS (free; on Windows the setup downloads it, on Linux and macOS you build it: see section 2). A simulation takes from about a minute (a dipole) to about four minutes (the TI DN007 IFA) on a current multi-core PC, 10 to 15 times less on the GPU of an NVIDIA graphics card, and longer for large imported boards.

2. Installation and first start

Install LW RF Studio on Windows

1. Run `LWRFStudio-<version>-setup.exe`. Setup is available in English, Norwegian and French.
2. Accept the license agreement.
3. Choose whether to install for all users or only for you, then the folder (for all users, the default is `C:\Program Files\LW RF Studio`) and, if you want, a desktop icon.

The program is self-contained: no .NET runtime is needed. Running a newer setup upgrades the installed version and keeps your projects, settings and license. LW RF Studio installs alongside an older PCB Antenna Designer; it does not replace or uninstall it.

Setup also installs the command-line tool (section 13), the sample projects (Start menu > LW RF Studio > Samples), the README, the license agreement and the third-party licenses.

Install LW RF Studio on Linux (Ubuntu)

The Linux package, LWRFStudio-<version>-linux-x64.tar.gz, runs on Ubuntu 22.04 or later (x64), and on other desktop distributions with X11 or XWayland. It is self-contained too.

- Unpack it where the program should stay, for example in ~/opt: **mkdir -p ~/opt && tar -xzf LWRFStudio-<version>-linux-x64.tar.gz -C ~/opt**
- Add it to the application menu: **~/opt/LWRFStudio-<version>-linux-x64/install.sh**. Run it as yourself, not with sudo: with sudo, the menu entry goes to the menu of root.
- To remove it: **install.sh --uninstall**, then delete the folder.

INSTALL-linux.txt in the package has the details. In a virtual machine (VMware, VirtualBox), give the machine several CPU cores: openEMS uses one thread per core.

Install LW RF Studio on macOS

There are two packages: LWRFStudio-<version>-osx-arm64.zip for Apple silicon (M1 and later) and LWRFStudio-<version>-osx-x64.zip for Intel Macs. Both need macOS 13 or later.

- Unzip the package and drag **LW RF Studio.app** to **Applications**.
- The app is not signed by Apple yet, so the first time macOS says that it cannot be opened or is damaged. Remove the download mark once, in Terminal: **xattr -dr com.apple.quarantine "/Applications/LW RF Studio.app"**, or open System Settings > Privacy & Security and click **Open Anyway**.
- Start LW RF Studio from Applications or Launchpad.

The command-line tool is inside the app: /Applications/LW RF Studio.app/Contents/MacOS/LWRFStudioCli. INSTALL-macOS.txt in the package has the details.

Install openEMS

openEMS is the solver. It is free software (GNU GPL v3) and is not part of the program. On Windows, if setup doesn't find it on the PC, it offers to download openEMS 0.37.0-38 with its GPU engine (64 MB from lwtech.org, needs internet) and unpacks it to C:\openEMS, where the program, its command line

and the KiCad plugin all find it; uninstalling removes it again. An openEMS that an earlier setup put in the program's openEMS folder keeps working. This build of the openEMS project uses an NVIDIA graphics card when there is one (section 6) and the CPU otherwise. On a PC with an NVIDIA card whose openEMS has no GPU engine, setup offers to replace it. COPYING.txt and SOURCE.txt in its folder give its licence and the address of its source code.

On Windows, without internet or to use your own copy:

1. Download openEMS with its GPU engine from lwtech.org/download, or the Windows build of the openEMS project from github.com/thliebig/openEMS-Project/releases (without GPU engine).
2. Unzip it, for example to `C:\openEMS`.

On Windows, the program finds openEMS.exe by itself in C:\openEMS, C:\opt\openEMS, C:\Program Files\openEMS, %USERPROFILE%\openEMS and on the PATH. If it is elsewhere, the first simulation asks you to select it. You can change it later with Tools > Locate openEMS.exe (Tools > Locate openEMS on Linux and macOS); the command line and the KiCad plugin then use the same openEMS. The radiation pattern also needs nf2ff (nf2ff.exe on Windows), which comes with openEMS in the same folder.

On Linux and macOS, build openEMS from its source into ~/opt/openEMS, the first place the program looks (then ~/openEMS/bin, ~/openEMS, /opt/openEMS/bin, /usr/local/bin, /opt/homebrew/bin, /usr/bin and the PATH). On a Mac, install Homebrew (brew.sh) first. Do not run the build with sudo:

```
sudo apt install git                  (Linux; on a Mac, git comes with Homebrew)
git clone --recursive https://github.com/thliebig/openEMS-Project.git
cd openEMS-Project
./scripts/install_deps.sh            (the packages the build needs)
./update_openEMS.sh ~/opt/openEMS --disable-GUI
```

The build takes 10 to 30 minutes. Then LWRFStudioCli engine (section 13) shows the openEMS the program found and the engine it uses. An openEMS built this way has its GPU engine only when the CUDA toolkit was installed for the build; on a Mac, openEMS runs on the CPU.

First start

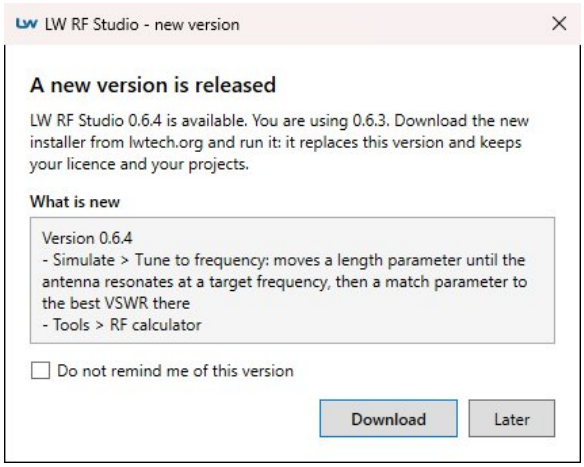
The first time, the program opens the Printed IFA template. After that it reopens the last project you opened or saved (File > Open recent lists the others), and shows the results of its last simulation if they are still in its simulation folder.

The first start on a PC also starts the 30-day trial. The title bar shows the days left, for example “LW RF Studio (trial, 30 days left) - Printed IFA”, and at every start a window shows them with a **Buy now** button (section 14). After the trial, without a license, the program opens only the license window.



The trial window at the start of the program.

At every start the program also tells lwtech.org that it runs: the PC ID, the version and whether it has a license. When a newer version is released, a window says so, with what is new. **Download** opens the download page, and **Do not remind me of this version** stops the reminder for that version. Without internet the program starts as usual.



The new-version notice, with the release notes.

The program opens the last project at start. Only when that project has a very large drawing (10,000 shapes and vias, or a million polygon corners: several times a whole 4-layer board) does it ask first: open it again, or start with an empty project. Saved simulation results do not count: a project opens at once with its results, also a whole imported board.

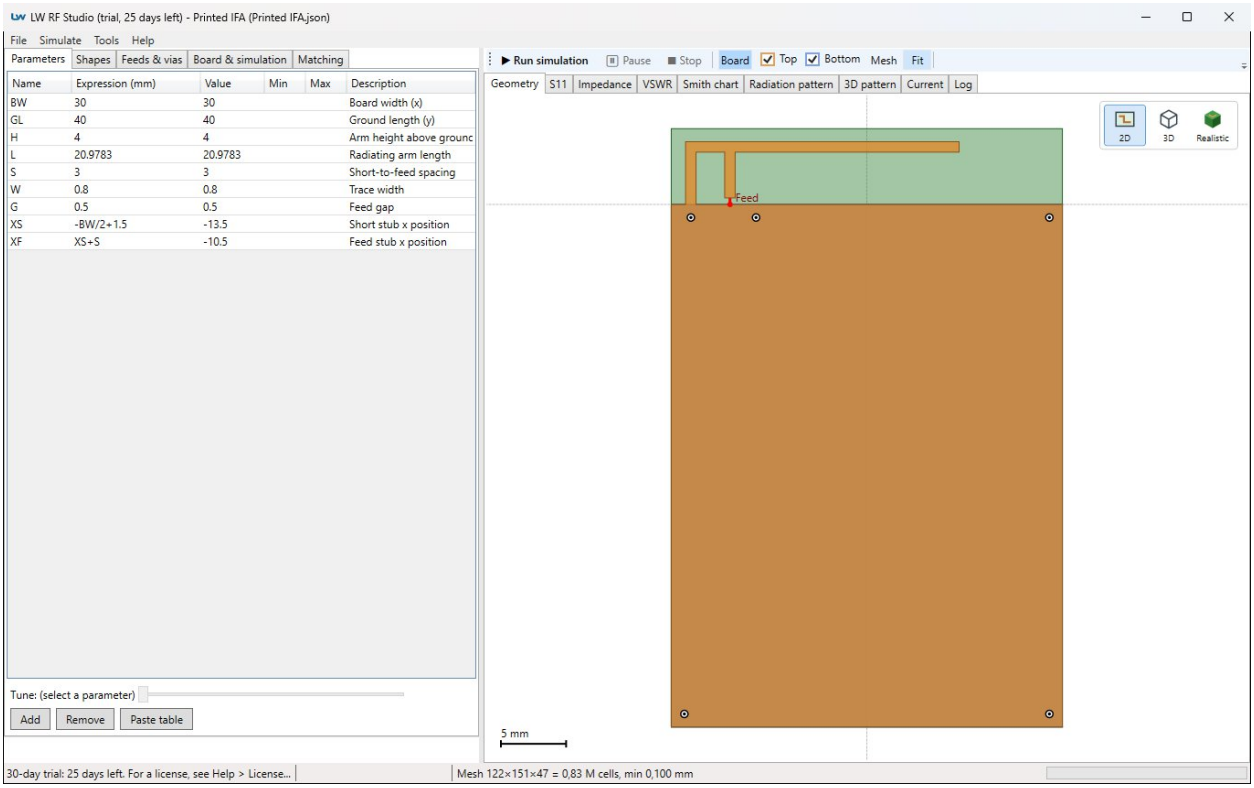
3. The main window: PCB Antenna Lab

PCB Antenna Lab is the main window of LW RF Studio. It has four areas:

- **Editors (left):** six tabs. Parameters, Shapes, Feeds & vias, Volumes, and Board & simulation describe the antenna; Matching tries a matching network on the results. Model errors are listed in red under the tabs, and the simulation will not run until they are fixed.
- **Views (right):** the geometry and the results, one tab each.
- **Toolbar** (above the views): Run, Pause and Stop, layer and mesh display, grid and snap, Fit, and a one-line summary of the last result.
- **Status bar (bottom):** the last message, the cursor position in mm, the mesh size, and the progress of a running simulation.

Drag the bar between the editors and the views to share the width between them. The tabs stay on one row: when some do not fit, a small arrow at the end of the row shows that more tabs are hidden, and a click on it brings the next one into view.

The title bar shows the program, the days left during the trial, the project and its file, and a * when there are unsaved changes, for example “LW RF Studio (trial, 12 days left) - My IFA (my_ifa.json) *”.



The main window: the editors on the left, the views on the right (the Printed IFA template).

Menus

Menu	Command	What it does
File	New from template	Starts a new project from one of the built-in antennas (section 4).
File	Open (Ctrl+O), Save (Ctrl+S), Save As	Projects are saved as `.json` files.
File	Open recent	The last ten projects you opened or saved, newest first.
File	Project properties	The name and description of the project, the unit of the lengths shown (mm, inch or mil) and the grid spacing (section 3).
File	Import PCB board (Ctrl+I)	Reads an Eagle .brd, Autodesk Fusion .fbrd, KiCad .kicad_pcb or Altium .PcbDoc board (section

Menu	Command	What it does
		5).
File	Import Gerber and drill files	Reads a set of Gerber and Excellon files (section 5).
File	Import MATLAB pcbStack script	Reads a MATLAB `.m` script that builds a `pcbStack` (section 5).
File	Export openEMS XML, KiCad footprint, Eagle library, Altium footprint library, results, radiation pattern	See section 11.
Edit	Undo (Ctrl+Z), Redo (Ctrl+Y)	Steps back and forward over the changes to the parameters, shapes, feeds, vias, stackup and settings (section 4).
Edit	Paste dimension table	Reads a dimension table copied from a datasheet into the Parameters tab (section 4).
Simulate	Run (F5), Stop	Starts or stops openEMS (section 7).
Simulate	Tune to frequency	Tunes a length and a match parameter to a target frequency (section 7).
Simulate	Load results from simulation folder	Shows the results of an earlier run (section 7).
Simulate	Load reference S11, Clear reference	Overlays a measured trace on the charts (section 8).
Tools	SmithMatch (Ctrl+M)	Opens the SmithMatch window (section 12).
Tools	RF calculator	Return loss, VSWR and $\|\Gamma\|$, a quarter-wave estimate and the scaling of a design (section 8).
Tools	Locate openEMS.exe (Locate openEMS on Linux and macOS)	Selects the solver if it is not found automatically.
Tools	Use the GPU engine when available	On (the default): openEMS runs on the GPU of an NVIDIA graphics card when it can (section 6). Off: always on the CPU.
Tools	Remote GPU (beta)	Runs the simulations on the GPU of another PC, through lwtech.org (section 6). Contact us to test this feature.
Help	User manual (F1), Licence, About LW RF Studio	Opens this manual as a PDF (installed with the program); licence status, buying a license and

Menu	Command	What it does
		activation (section 14), version and credits.

Toolbar

- ▶ Run simulation, ⏸ Pause and ■ Stop: start, pause or stop the simulation (section 7).
- **Board** and one box per metal layer: show or hide the board outline and each copper layer in the Geometry view.
- **Mesh:** draws the openEMS mesh lines over the geometry, so you can check that the fine mesh covers the antenna.
- **Volumes**: shows or hides the metal boxes and the casing of the Volumes tab in the Geometry view.
- **Grid**: shows the grid in the Geometry view. Its spacing is set in File > Project properties. The grid and the mesh exclude each other: showing one hides the other.
- **Snap**: snaps the shapes you move to the grid (see Grid, snap and moving shapes below).
- **Fit:** zooms the Geometry view to the whole board, the same as double-clicking it.
- **2D**, **3D** and **Realistic** (top right of the Geometry view): see the board flat, in 3D or as it will be made (see Geometry view below).
- **Summary:** after a simulation, the resonance frequency, the best S11, the input impedance and the −10 dB bandwidth, and the matched result when a matching network is set.

Views

Tab	Shows
Geometry	The board seen from above, updated as you edit; also in 3D and as a realistic picture.
S11, Impedance, VSWR, Smith chart	The input match of the antenna versus frequency (section 8).
Radiation pattern, 3D pattern	The far field (section 9).
Current	The current on the copper (section 10).
Log	The progress and the openEMS output of the running simulation (section 7).

Geometry view

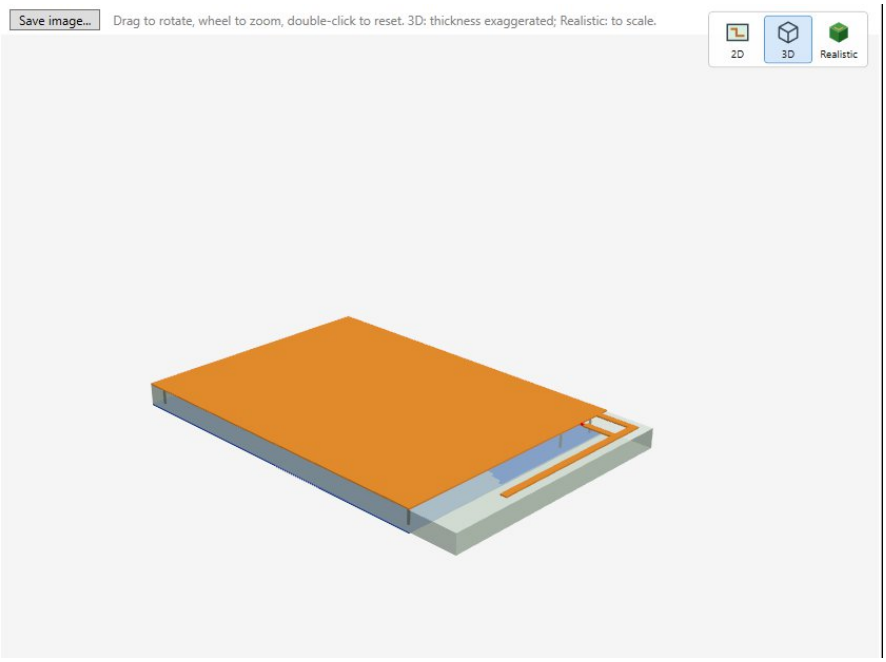
Drag with the middle mouse button (press the wheel) to move the view, turn the wheel to zoom, and double-click to fit the board. A left drag on an empty place does nothing, so a missed shape never moves the view. The status bar shows the cursor position in the project unit (mm at first; see Units below), with x to the right and y up. Where the origin lies depends on the project; imported boards put it at the bottom-left corner of the board.

Select a shape: click it. The Shapes tab opens on its row, ready to edit, and the shape is outlined in the view. The cursor becomes a hand over a shape. Shift+click adds a shape to the selection, or takes it out; in the Shapes tab, Shift+click and Ctrl+click select several rows.

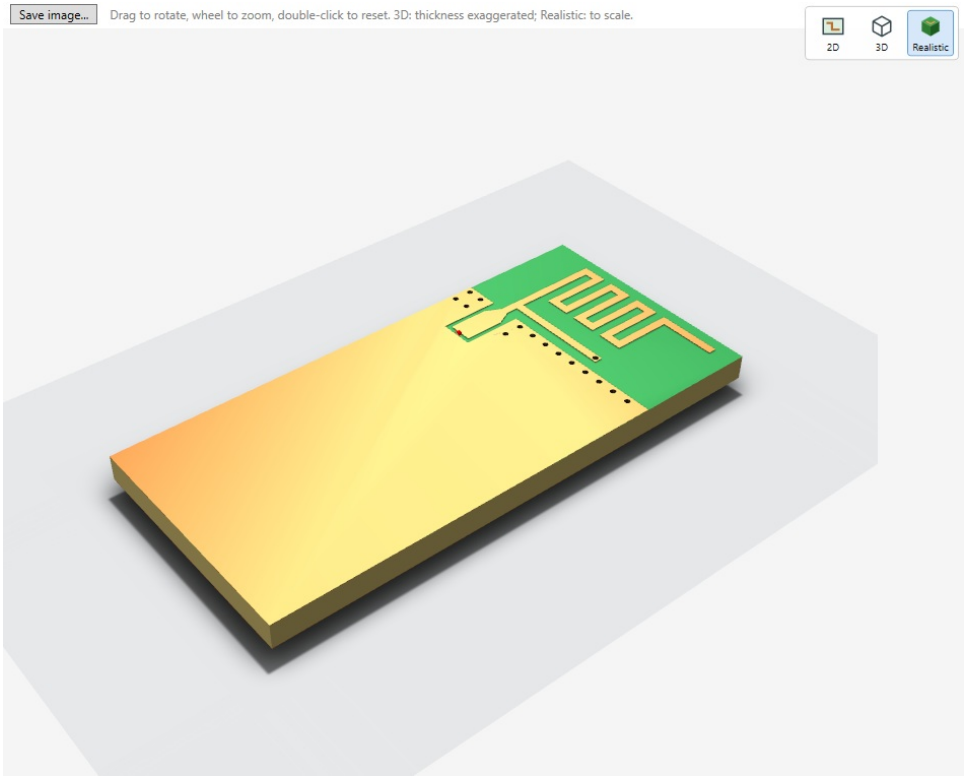
2D, 3D and Realistic: the buttons at the top right of the view change how the board is drawn:

- **2D**: the board from above, as above.
- **3D**: the layers one above the other, each in its colour, with the substrate see-through. A thin board is drawn thicker than it is, so that every layer shows.
- **Realistic**: the board to scale as it will be made: green solder mask, copper-coloured copper, the vias and holes drilled through, a light and soft shadows.

In 3D and Realistic, drag to rotate, turn the wheel to zoom and double-click to return to the start view. **Save image** saves the view as a PNG file.



The 3D view: each layer in its colour, the substrate see-through.



The Realistic view: the board to scale, with solder mask, copper, drills and shadows.

Grid, snap and moving shapes

The **Grid** box in the toolbar shows a grid in the Geometry view, with a darker line every fifth. Its spacing is set in File > Project properties (1 mm at first). When the view is zoomed out too far for it, the grid is drawn at 2, 5, 10 ... times the spacing.

To move shapes, select them first (click a shape in the view, or its row in the Shapes tab). Then:

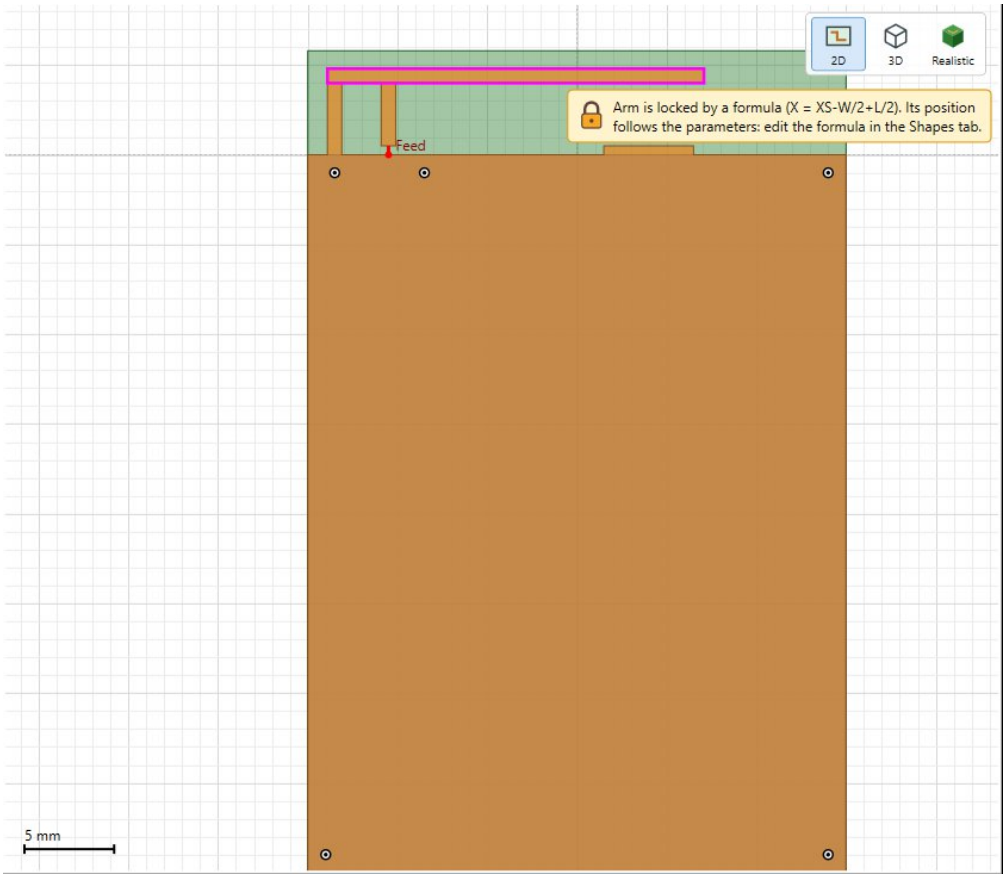
- **Drag** a selected shape with the mouse. A left drag elsewhere does nothing; the middle mouse button moves the view.
- Press the arrow keys with the view in focus (click it first): one grid step per press, ten steps with Shift. They move the selected shapes, volumes, feeds, vias and holes.
- With **Snap** ticked, the first selected shape lands on the grid: a drag goes to the nearest grid point, and an arrow key to the next grid line. Without Snap, a drag follows the mouse exactly and an arrow key moves by one grid spacing.
- Every move is a step of **Edit > Undo**, and a whole drag is one step.
- Ctrl+click on a shape, a feed, a via, a hole or a volume puts it on the grid: it moves to the nearest grid point by its origin (the cross of a shape, the start of a feed, the centre of a via, a hole or a volume). The status bar says how far it moved; it is one step of Edit > Undo.

A shape can be moved when its position is written as plain numbers: X and Y. When a position is a formula, such as XS-W/2+L/2, the shape follows its parameters, so it is **locked**: a padlock with the reason appears when you try to move it. Edit the formula in the Shapes tab, or change the parameters. Moving a shape rewrites its X and Y as numbers. Feeds and vias are edited in their tab.

Resize a selected rectangle, circle or polygon with the small square handles on it: drag a corner or an edge of a rectangle (the opposite side stays where it is), an end of a circle (its radius changes), or a corner of a polygon. While you drag, the handle snaps onto a corner of another figure (a corner of a shape, the end of a feed, a via or a hole) when one is within 10 pixels, with Snap on or off; elsewhere Snap puts the edge or the corner on the grid. When a formula sets the size or the position, the handles are grey and a padlock explains. Meanders and rotated shapes have no handles.

Polygon corners: double-click on an edge of the selected polygon to add a corner there, and double-click on a corner to remove it (a polygon keeps at least three). **Add polygon** starts with a four-corner shape in the middle of the board. The X and Y of a polygon are its origin: its corners are measured from it, so changing X and Y moves the whole polygon.

Select and delete: click a shape to select it, Shift+click to add or remove one, and click where there is no shape to deselect. Press Delete to remove the selected shapes, feeds, vias and holes (in a table, the selected rows), and Esc to deselect everything. A double-click on a selected shape keeps it selected, even where another shape lies behind it. Boxes and casing walls of the Volumes tab are selected and moved the same way.



The grid, and the padlock on a shape whose position is a formula (the arm of the IFA).

The Geometry toolbar

In the 2D view, a toolbar on the left holds the tools to draw and arrange the antenna. Hover over a tool to see what it does. The tools act on the selection: click an item, Shift+click to add more, or draw a box around them. The order of the clicks counts for Join: the first selected item is the one that stays.

- **Select and Box select:** the arrow selects by clicking. With Box select, drag a box: every shape, volume, feed, via and hole inside it is selected, and the arrow comes back.
- **Ruler:** press on a point and drag to another, or click two points: the ruler shows the distance, dx, dy and the angle between them. The points snap to the grid with Snap on, and to the corners of the figures with Snap off. Esc clears it.
- **Add:** a rectangle, a circle, a polygon, a meander, a feed, a via, a hole, a metal box, a casing or a parameter. The new item appears in the middle of the view, on the grid, and in its table, ready to edit.
- **Duplicate and Remove:** copy or delete the selected rows of the open tab: parameters, shapes, feeds, vias or holes, volumes. On the Feeds & vias tab they act on the feeds when some are selected, else on the vias, else on the holes. On another tab they act on the selected shapes. A copy keeps the name with the next number: Rectangle1 gives Rectangle2.
- **Join:** the other selected items move, together, until their nearest edge touches the first selected item exactly. Use it to put a stub on the ground edge, or an arm against another.
- **Align:** one button with a menu: Align left, centres, right, top, middles or bottom. The selected items line up on the edges of all of them: Align left on the lowest left edge, Align top on the highest top edge, Align centres on the middle of the box around them.
- **Rotate left and Rotate right:** turn the selected shapes by 90°, together, about the centre of the box around them; a single shape turns in place. A right click on a shape also turns it (and the shapes selected with it) by 90°.
- **Mirror left-right and Mirror up-down:** flip the selected shapes about the vertical or the horizontal line through the centre of the box around them.
- **Parameters from shapes:** turns the numbers of the selected shapes into parameters (see below).

Each of these tools is one step of Edit > Undo. A shape whose position or rotation is a formula cannot be moved, turned or flipped: it is left out, and the status bar names it. The toolbar is shown in the 2D view only.

The origin of a selected shape is marked with a small cross: the centre of a rectangle or a circle, the first corner of a polygon (its anchor), and the start of a meander (the left end of its first segment, in the middle of the trace). X and Y of the shape are the position of this cross. A rectangle turned by a quarter turn keeps its resize handles.

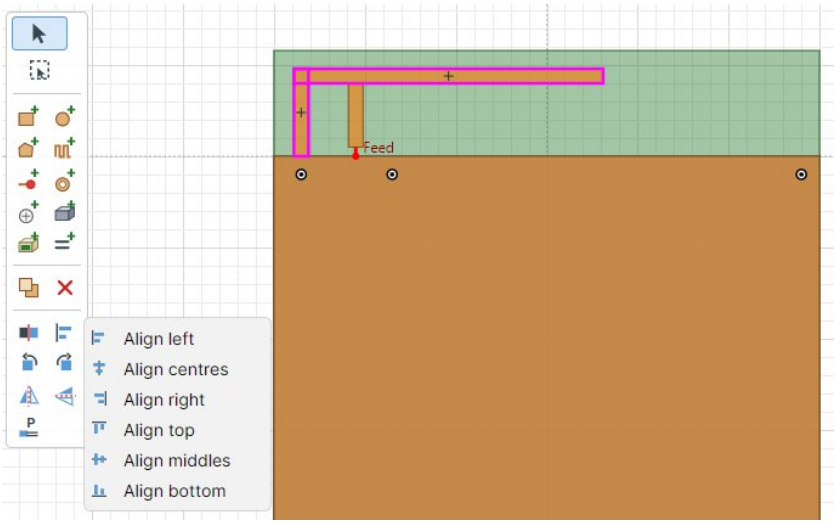
Feeds, vias and holes in the view: click one to select it (its row opens in the Feeds & vias tab), then drag it or use the arrow keys to move it, like a shape. A selected in-plane feed also has a handle on its other end: drag it to move that end. The end stays along x or along y from the feed point, as an in-plane port must, and snaps like the handles of a shape. When the feed point is a formula, X2 and Y2 are written from it (for example Y0-1 - 2), so the feed keeps following the parameters.

Parameters from shapes

Select one or more shapes and click Parameters from shapes (the last tool). Tick what becomes parameters: Position (X, Y), Size (length, width, radius, pitch, folds) and Rotation. The window lists each parameter with its name and present value; change a name if you want. Create parameters adds them to the Parameters tab and writes their names in the shapes, so the board does not change. A field that is already a formula stays as it is.

Names: the shape name and a letter for the field, such as Arm_L, Arm_W, Short_X. The shapes the program names itself are shortened: Rectangle1 gives R1_L, Circle2 C2_R, Meander3 M3_X, Polygon4 P4_Y. The values are rounded to 2 decimals.

Linked shapes: when a selected rectangle touches one selected before it (after Join, for example), the window offers to link them. The position of the second becomes a formula of the first, such as R2.X = R1_X + R1_L/2 + R2_L/2, so it stays against it when the first moves or grows. The link also works when the first shape already uses formulas: then no new parameter is needed, and Create parameters only writes the link.



The Geometry toolbar with the Align menu open; the short and the arm of an IFA are selected, with their origin crosses.

Units

File > Project properties sets the unit of the lengths the program shows: mm, inch or mil (a thousandth of an inch). It changes the cursor position, the scale bar, the grid spacing and the step of Snap and of the arrow keys. The project itself stays in mm: the expressions, the Value column and the headings of the tables are in mm, so a file means the same in any unit. The same window sets the grid spacing, in the unit chosen.

In an expression, a number can carry a unit, which is converted to mm: **10mil**, **0.5in**, **2mm**, **1.5cm**, **100um** (the units are mm, cm, um, mil or mils, in or inch; a space before the unit is allowed). For example W = 12mil, or XM = X0+25mil. This works in every field that takes an expression.

4. Designing an antenna

A project describes the antenna in four parts: named **parameters**, the copper **shapes** drawn with them, the **feeds and vias**, and the board **stackup**. All lengths are in millimetres (a number can carry a unit, for example 10mil: see Units in section 3). Every number in a shape, feed, via or dielectric layer can be an expression of the parameters, so changing one parameter moves everything that depends on it. The Geometry view redraws as you type.

Start from a template

File > New from template offers the antennas below. Each card of the window shows a picture and what the antenna is.

Template	What it is
Printed IFA	A classic inverted-F on the board edge, ground on both layers, fed in-plane across a gap.
Reference MIFA (mil drawing)	The widely used meandered IFA drawn in mils: CPW feed, three meanders and a shorting via.
TI AN043 MIFA	The meandered IFA of TI application note AN043 (CC2511 USB dongle), with TI's dimension names.
TI DN007 IFA	The inverted-F of TI design note DN007 (CC2430DB), with TI's dimension names.
Printed monopole (2.4 GHz)	A quarter-wave printed monopole next to a ground plane.
IFA with 5 GHz branch	The Printed IFA with an extra branch for the 5 GHz band (tuned at 2.4 GHz).
Stacked double-T monopole	A double-T monopole for Wi-Fi at 2.4 and 5.2 GHz.
Meandered monopole (868 MHz)	A meandered monopole for 868 MHz; it needs a matching network.
Bow-tie (UWB 3-10 GHz)	A printed bow-tie dipole for ultra-wideband, fed against 150 Ω (a balun).
Rectangular patch (2.45 GHz)	A probe-fed rectangular patch.
TI DN023 IFA (868 / 915 MHz)	The meandered inverted-F of TI design note DN023 (CC1110EM), with TI's dimension names.
TI DN034 Yagi (2.4 GHz)	The directional Yagi of TI design note DN034.
TI DN038 helical antenna (868 / 915 MHz)	The miniature helical antenna of TI design note DN038: strips on both layers joined by vias, at the edge of the ground plane.

Template	What it is
Silicon Labs AN1088 IFA	The 2.4 GHz inverted-F of Silicon Labs AN1088, for a 0.8 mm and a 1.6 mm board.
CC-Antenna-DK #5 (868 MHz)	The L-shaped monopole of TI's antenna kit CC-Antenna-DK.
CC-Antenna-DK #6 (868 / 2440 MHz)	The dual-band meander monopole of the same kit.
CC-Antenna-DK #10 (868 MHz)	The small L-shaped monopole of the same kit.

The same projects are also in the **Samples** folder of the installation. Dimensions that the TI documents do not give (board thickness, ground size) are parameters you can set.

Parameters

Each row has a **Name**, an **Expression** in mm and a **Description**. The **Value** column shows the result.

Expressions follow MATLAB syntax:

- Numbers, other parameters, `+ - * / ^` (and `.* ./ .^`), and parentheses.
- Numbers with a unit: 10mil, 0.5in, 2mm, 1.5cm, 100um (converted to mm).
- A comma or a point as the decimal separator, in every number field and expression: 1,8 or 1.8. In a function the comma still separates the arguments: min(1,8) is the smaller of 1 and 8.
- `pi` and the functions `sin cos tan asin acos atan sqrt abs exp log round floor ceil min max`.
- Example: `XM = X0+LF-W/2`.

A parameter may use any other parameter, in any order, as long as there is no loop.

Edit > Paste dimension table reads a dimension table copied from a datasheet or a spreadsheet: one Name value [unit] per line, such as L1 3.94 mm or W2 = 20 mil. The units mm, mil, µm (or um), cm, in and m are converted to mm, and a decimal comma is accepted. Existing parameters are updated by name and new ones are added.

Tune slider: select a parameter whose expression is a plain number, then drag the slider under the table. The geometry follows as you drag. The range is the parameter's Min and Max, or ±50% of its value when they are empty. Min and Max also set the range of Simulate > Tune to frequency (section 7).

The Value column, the Tune slider, Tune to frequency and Parameters from shapes write numbers with at most 2 decimals.

Shapes

Each shape has a **Name**, an **On** box (untick it to leave the shape out without deleting it), a **Layer** and a **Shape**:

Shape	Fields used
Rectangle	Centre **X**, **Y**; **Length (x)** and **Width (y)**, as in MATLAB; **Rot°** about the centre.

Shape	Fields used
Circle	Centre **X**, **Y** and **Radius**.
Polygon	Origin X, Y (the corners are measured from it); Points as x1, y1; x2, y2; ..., each coordinate an expression; Rot° about the first corner.
Meander	See below.

A **meander** starts at (X, Y) on the centre line of the trace. It steps **Pitch** along +x, then runs a leg of **Length** along y (a negative length goes down first), and turns back for each of the **Folds** legs. **Width** is the trace width, and **Rot°** turns the whole meander about (X, Y).

Shapes on the same layer are merged, so overlapping copper is joined.

Select several rows with Shift or Ctrl: Duplicate in the Geometry toolbar copies all of them (Rectangle1 gives Rectangle2) and Remove deletes them all. They act on the tab that is open: the Parameters, Shapes, Feeds & vias and Volumes tables work the same way. The add tools are in the Geometry toolbar too.

Sort a table: click a column header, and click again to reverse the order; a small arrow marks the sorted column. Numbers are sorted by value (-14 before -13.5 before 2), then formulas and names in alphabetical order. Sorting changes only the order on screen: the project keeps its own order, so the first feed stays port 1. The stackup is not sorted: its order is the order of the layers.

Edit a cell: select it and type, or double-click it. Ctrl+C and Ctrl+V copy the value of a cell to another, a drop-down list (a layer, a kind) applies at once, and a column can be moved by dragging its header.

Hover over a column header of the Shapes tab to see what the column means for each kind of shape. A click on a shape in the Geometry view opens its row.

The Board layer: shapes on the layer called **Board** are not copper. Together they set the outline of every dielectric layer. Without a Board shape there is no dielectric, and the copper is simulated in air.

Feeds

The feed is where the radio connects. **Port 1** is excited; any other feed is terminated in its impedance.

- **Vertical:** a port at (X, Y) from the **Signal layer** down to the **Ground layer**, through the substrate. Use it when the antenna is fed from a via or a pad over the ground.
- **InPlane:** a port in the plane of the signal layer, from (X, Y) to (X2, Y2), for a feed across a gap in the copper. The two points must share an x or a y: the port runs along x or along y. Its other end has a handle in the Geometry view (section 3).
- **Z0 (Ω):** the reference impedance, normally 50.

An empty Signal layer means the top metal layer, and an empty Ground layer the bottom one.

Vias

A via joins two metal layers at (X, Y), typically the short of an inverted-F. Leave **Ø** empty for a thin wire, or give the drill diameter in mm to model the barrel as a post of the same perimeter.

Holes

The Holes table, under the vias on the Feeds & vias tab, holds the holes that are not plated: mounting holes, connector pegs. Add hole in the Geometry toolbar adds one: give its drill diameter Ø in mm, and its centre X and Y. A hole goes through the whole board: in the simulation it is air, through the copper and the dielectric of every layer. It is drawn at its size in the 2D view, as a white circle with a cross, and drilled through in the 3D views. Select it, drag it or move it with the arrow keys like a via.

On the Feeds & vias tab, drag the grey bars between the Feeds, Vias and Holes tables to give one of them more height.

Stackup

The stackup is on the **Board & simulation** tab, listed from the top of the board to the bottom:

- **Metal** layers are copper sheets of zero thickness (perfect conductors). Shapes, feeds and vias refer to them by name.
- **Dielectric** layers have a **Thickness** (mm), a relative permittivity **εr** and a loss tangent **tanδ**. Typical values are εr 4.2–4.6 and tanδ 0.015–0.02 for FR-4, and εr 3.55 and tanδ 0.0027 for Rogers RO4003C.

Use **Add metal**, **Add dielectric**, **Up**, **Down** and **Remove** to change the stack. A higher εr lowers the resonance; tanδ mainly lowers the efficiency.

Volumes: metal parts and a casing

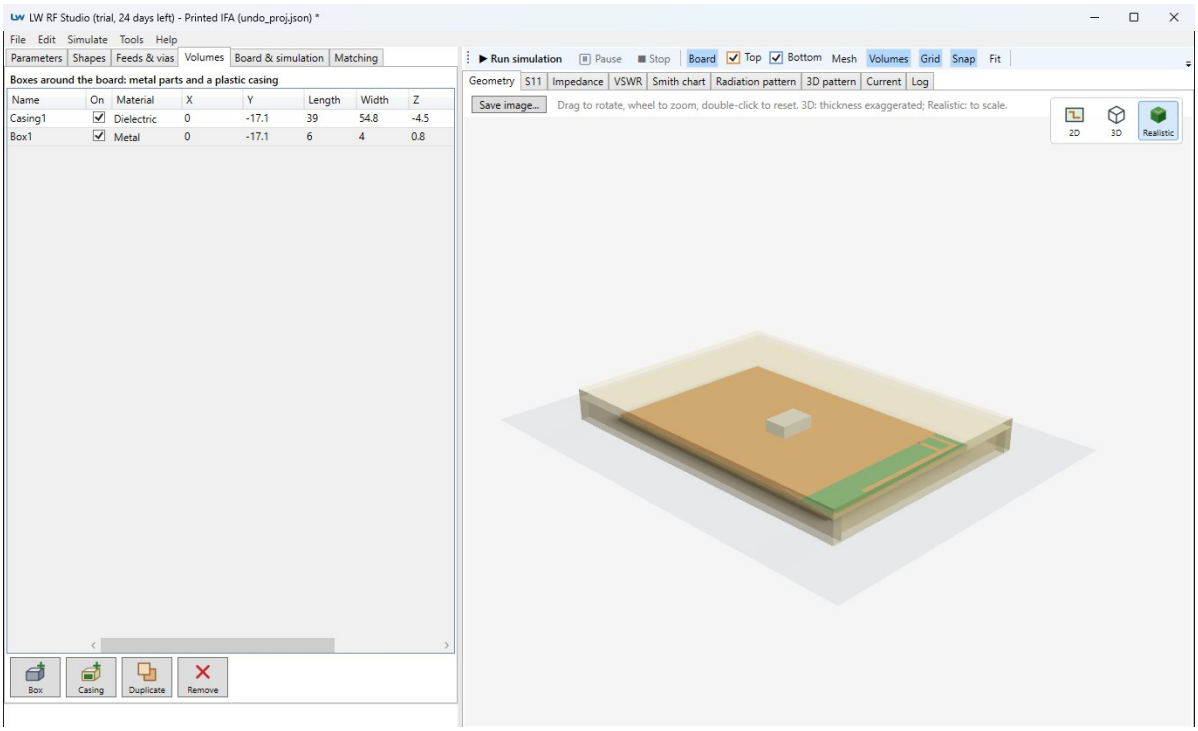
The **Volumes** tab adds boxes around the board that the simulation takes into account: metal parts near the antenna (a shield can, a connector, a battery) and a plastic casing around the whole board. Each row is one box:

- **Material**: Metal is a perfect conductor. Dielectric is plastic, with its **εr** (about 2.5 to 4) and **tanδ** (0.005 to 0.02) in the last two columns.
- **X**, **Y**: the centre of the box in the board plane. **Length** and **Width** are its size along x and y.
- **Z** and **Height**: the bottom of the box and its height. z = 0 is the bottom of the board, so the top of a 1.6 mm board is at 1.6 and a casing around the board starts below 0.
- **Wall**: empty for a solid box. A number above 0 makes the box hollow and closed, a shell with walls of that thickness; Length, Width and Height are then the outside size.

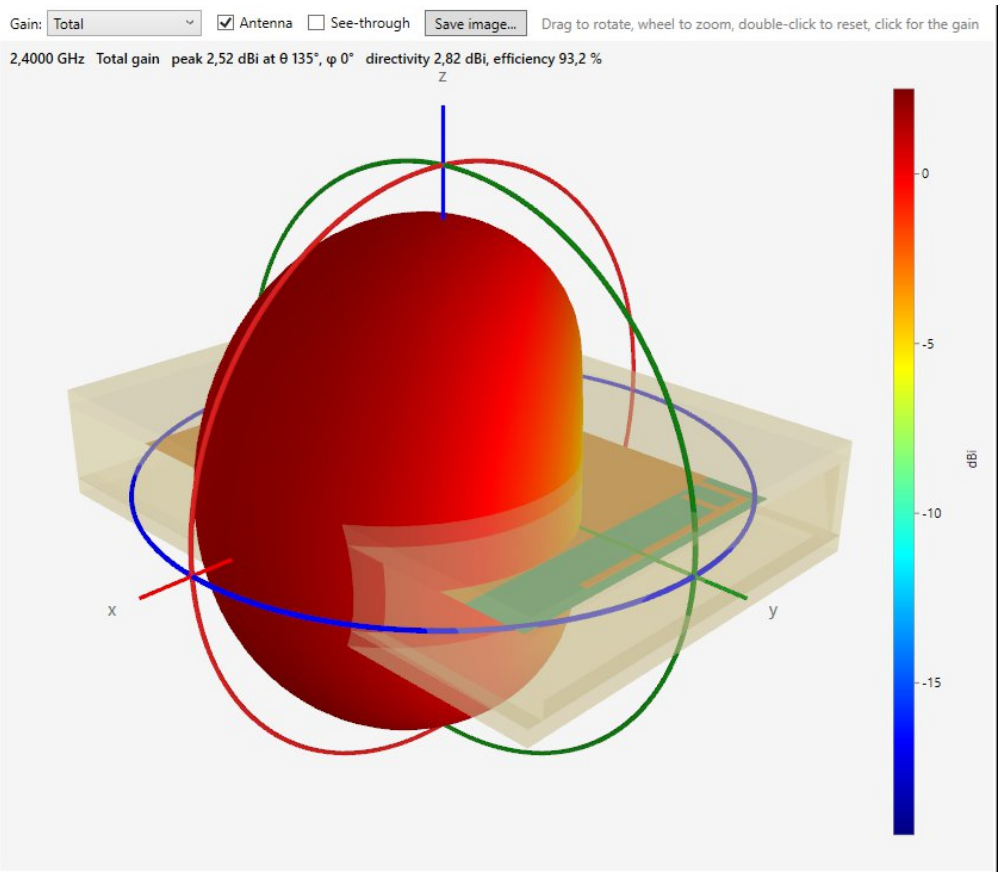
Box adds a metal box on top of the board, in the middle. **Casing** adds a plastic casing around the board, with 3 mm of air between the board and the walls and walls of 1.5 mm (εr 3, tanδ 0.01); change the numbers to your plastic and your housing. **Duplicate** and **Remove** act on the selected rows, and **On** leaves a volume out without deleting it. Every number can be an expression of the parameters, so a height can be tuned.

In the Geometry view, the **Volumes** box of the toolbar shows or hides them. In 2D a metal box is a grey rectangle and a plastic one a dashed outline (for a casing, its four walls). Click a box or a wall to select it, then drag it or use the arrow keys to move it, like a shape; a box whose X or Y is a formula is locked. In 3D and Realistic the boxes have their true height: metal grey, plastic see-through. The 3D pattern tab shows them around the antenna, and the Current tab as wire frames.

What the simulation does with them. Metal volumes are perfect conductors and plastic ones are dielectrics with loss. The mesh gets lines on every face and finer cells inside plastic, and the air margin goes around the volumes. A metal box that touches ground copper or a via is connected to it; a box with a gap under it floats. Expect the resonance to move down when a casing is added, and a metal part close to the antenna to detune it strongly. A casing adds cells, so a run takes longer: compare with the Fast preset first. The currents on the surface of a metal volume are not shown; the Current tab shows the copper layers.



The Volumes tab, and the Realistic view of a plastic casing and a metal box on the board.



The 3D pattern with the casing and the metal box around the antenna.

Model errors

Problems such as an unknown parameter, a loop between parameters, a shape on a missing layer, or a mesh that cannot be built are listed in red under the editor tabs. Fix them before you run.

Undo and redo

Edit > Undo (Ctrl+Z) and **Edit > Redo** (Ctrl+Y, or Ctrl+Shift+Z) step back and forward over the changes to the design: the parameters, shapes, feeds, vias, the stackup, the mounting holes and the simulation settings, and the project name and description. Adding, removing, moving or editing a row is a step, and so is applying a tuning result.

Changes made within a moment of each other, such as dragging the tuning slider, are one step. Up to 100 steps are kept. Opening, creating or importing a project starts a new history, and Undo and Redo are not available while a simulation runs. While you are typing in a cell or a field, Ctrl+Z first undoes the typing there.

5. Importing designs

An import replaces the open project (you are asked to save it first), and the new project is untitled until you save it. Any warnings are shown in a message box and repeated on the Log tab; read them before simulating.

PCB boards: Eagle, Fusion, KiCad and Altium

File > Import PCB board (Ctrl+I) reads a whole board, so you can simulate the antenna with the real ground plane and the copper around it on your layout.

	Eagle `.brd`	KiCad `.kicad_pcb`
Versions	Eagle 6 to 9, Fusion Electronics	KiCad 5 to 10
Copper	Wires, polygons, rectangles, circles, and the pads and SMDs of placed parts, on layers 1 to 16	Tracks, arcs, filled zones, graphic copper and footprint pads, on F.Cu, In*.Cu and B.Cu
Board outline	Dimension layer (20)	Edge.Cuts
Vias	Vias and plated holes	Vias and plated holes
Stackup	Layer setup and thicknesses from the design rules	Board Setup > Physical Stackup

Autodesk Fusion Electronics (.fbrd, or a .f3z archive that holds one): the board is stored inside as Eagle XML and is read like an Eagle board.

Altium Designer (.PcbDoc): the copper, the vias and plated holes, the board shape, and the layer stack with the thickness and εr of each dielectric.

After the import, a window asks for what the file does not say: the dielectric material, the feed when several pads qualify, and the signal and ground layers. Its Notes box lists, one per line, what else the importer has to say; it scrolls, and OK and Cancel always stay at the bottom of the window.

Place the feed with a click: in that window, click on the board where the feed must be. The feed goes exactly at the click, on the layer of the smallest piece of copper under it (a pad rather than the plane around it). In the picture, turn the wheel to zoom and drag with the middle mouse button to move. A large board shows a progress bar while it is read and drawn. After the import, the feed can still be dragged in the Geometry view.

What to check after an import:

1. **Stackup** (Board & simulation tab). Eagle files do not store εr or tanδ, so the dielectrics are set to FR-4 (4.4 / 0.02). A KiCad board without a physical stackup gets FR-4 (4.5 / 0.02) split evenly between the copper layers. Enter your real material.
2. **Feed** (Feeds & vias tab). The importer puts a vertical feed on a pad named ANT, RF or FEED, or, in KiCad, on the signal pad of a footprint whose reference starts with AE or ANT. The feed runs from the pad down to the nearest copper layer under it. An antenna pad often sits just past the ground edge: then the feed moves along the pad's copper (up to 3 mm) to where the ground starts, and a warning says so. If no such pad exists, add the feed yourself.
3. **Frequency range** (Board & simulation tab).

Other things the importer does:

- **No closed board outline:** the board is taken as the rectangle around the copper, and a warning gives its size. Plated holes count by their centre, so castellated holes on the edge are cut through the middle as on the real board. Draw the outline in your CAD tool for the exact shape.
- **Polygon pours (Eagle)** are stored as outlines in Eagle. They are clipped to the board and kept clear of other nets, which is close to, but not exactly, what Eagle pours.
- **Unfilled zones (KiCad)** are approximated. Fill all zones in KiCad (press B) and save before importing to get the exact copper.
- **Copper outside the board** (parts parked off the board, edge overhangs) and vias whose centre is off the board are left out.
- Text on copper layers is not imported. Non-plated holes (mounting holes, connector pegs) go to the Holes table: they are cut through the copper and the dielectric in the simulation, and drawn in the 2D and 3D views.
- **Coordinates:** the board is moved so that its bottom-left corner is at (0, 0), with y up. For KiCad, whose y axis points down, the Log gives the conversion from KiCad coordinates.

Large boards take a few seconds to import and make large meshes. To speed up the simulation, untick the **On** box of shapes far from the antenna, or simplify the board in your CAD tool before you import it.

Gerber and drill files

File > Import Gerber and drill files reads the fabrication files of a board: the Gerber copper layers, the board outline and the Excellon drill files, as a .zip, the job file (.gbrjob) of a folder, or files you select. A window shows what each file was taken for (copper, outline, plated or non-plated drills) and lets you change it before importing. Without a job file the stackup is a guess: check it after the import.

MATLAB pcbStack scripts

File > Import MATLAB pcbStack script reads the `.m` script that MATLAB's PCB Antenna Designer exports. It reads the `pcbStack` object with its rectangles, circles, triangles and polygons, the dielectric layers, and the feed and via locations.

- MATLAB works in metres and this program in millimetres; lengths are converted.
- Numeric workspace variables become parameters, so the imported design keeps its dimension names.
- Shape additions are kept; subtractions are not supported, and a shape with a cut-out is kept without it (a warning says which).
- A balanced MATLAB feed (delta gap) on one layer is imported as a vertical port; check it on the Feeds & vias tab.

After the import, check the frequency range on the Board & simulation tab.

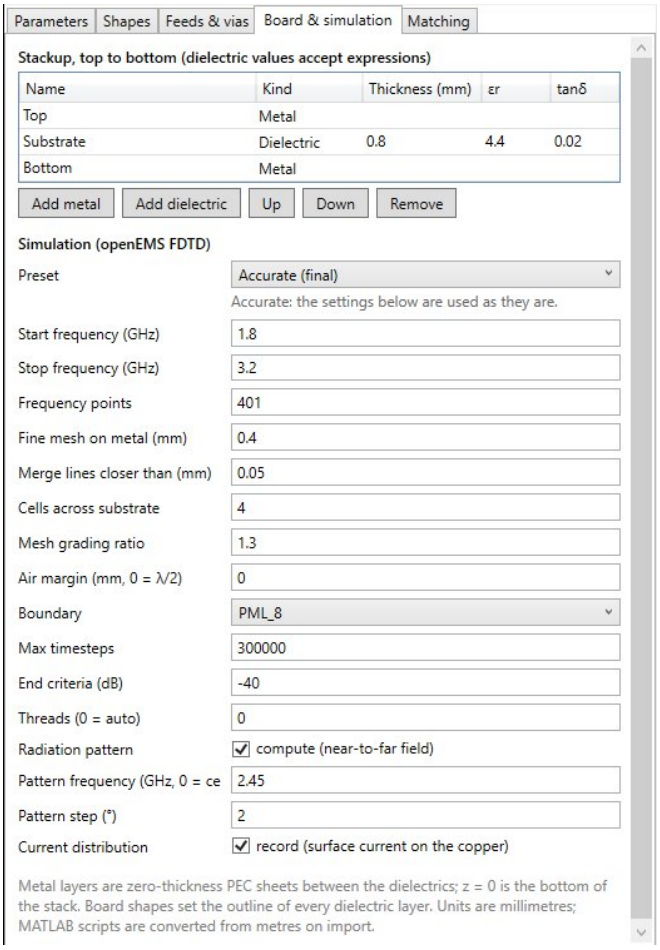
6. Simulation settings

The settings are on the **Board & simulation** tab, under the stackup, and are saved with the project. Hover over any label for an explanation. The status bar shows the size of the mesh they give, for example `Mesh 96×120×42 = 0.48 M cells, min 0.050 mm`.

Fast, Standard and Accurate presets

- **Fast estimate**: for a first look at the VSWR. Standard, then cells another 1.5 times larger, two cells across the substrate and a stop at −20 dB. Several times faster than Standard; the resonance can be a few percent away from the accurate result.
- **Standard** (tuning): fine cells 1.6 times larger, merge distance doubled, one cell less across the substrate, λ/4 of automatic air instead of λ/2, stop at −30 dB, and no radiation pattern. On the TI DN007 IFA it runs 2.4 times faster, with the resonance 0.6% higher.
- **Accurate** (final): puts back the values you had before choosing Standard or Fast. Without them, it uses the defaults below.

Choose the preset in the **Preset** list; the line under it repeats what the preset sets. Your own values are kept, and Accurate puts them back. Use Fast or Standard while you tune the dimensions, then Accurate for the final result.



The Board & simulation tab: the stackup, the preset and the settings.

Settings

Setting	Default	What it does
Start / Stop	2 / 3	The sweep shown in the results. The stop frequency also sets the largest mesh cell (λ/20), so a higher

Setting	Default	What it does
frequency (GHz)		stop frequency means a finer mesh and a longer run.
Frequency points	401	Points in the result curves. Only affects post-processing, not the run time.
Fine mesh on metal (mm)	0.25	Largest cell around the antenna copper, feeds and vias. Halving it gives about 4 times the cells and twice the timesteps. Ground planes get larger, graded cells.
Merge lines closer than (mm)	0.05	Mesh lines closer than this become one. Very small cells force a very small time step. Lower it only to resolve a thin gap or trace. Feed ports always sit on a mesh line.
Cells across substrate	4	Cells through the thickest dielectric; thinner layers get proportionally fewer, at least 2.
Mesh grading ratio	1.3	How fast cells may grow away from the antenna (1.05 to 2). Lower is smoother and more accurate but adds cells.
Air margin (mm)	0 (automatic)	Distance from the board to the absorbing boundary. Automatic is λ/2 at the centre frequency, at least 20 mm. Too little air distorts the result.
Boundary	PML_8	The absorbing boundary. PML_8 absorbs well. MUR is simpler, reflects more and needs more air.
Max timesteps	300 000	Safety limit. A run that reaches it usually has a very high-Q structure or a feed that excites nothing.
End criteria (dB)	−40	The run ends when the field energy has fallen this far below its peak. −40 dB is enough for a matched antenna; −50 dB gives cleaner curves for high-Q or badly matched antennas.
Threads	0 (automatic)	CPU threads openEMS uses on the CPU. Automatic is one thread per performance core: openEMS waits for its slowest thread at every timestep, so using the efficiency cores and hyper-threads too makes it slower (on an i7-14700, 8 threads run 1.6 times faster than all 28). This is why the CPU load stays well below 100% during a run. On Linux and macOS it counts the physical cores (in a virtual machine, its

Setting	Default	What it does
		virtual CPUs). The GPU engine does not use it.
Radiation pattern	on	Records the fields on a box around the antenna and computes the far field (section 9). S11 is the same either way.
Pattern frequency (GHz)	0 (centre of the sweep)	The frequency of the pattern and of the current distribution. Set it before running: another frequency needs a new run.
Pattern step (°)	2	Angular step of the far field. 1° takes 4 times as long to compute as 2°.
Current distribution	on	Records the surface current on the copper at the pattern frequency (section 10). Adds little to the run time.

GPU engine

With an NVIDIA graphics card, openEMS can run the simulation on the GPU: 10 to 15 times faster than on the CPU, with the same results. On an RTX 5060, the Printed IFA with its radiation pattern (0.83 M cells) takes 22 s instead of 280 s on 8 CPU cores.

The program uses the GPU by itself when all of these are true:

- **Tools > Use the GPU engine when available** is ticked (it is by default);
- the openEMS in use has its GPU engine: the one the Windows setup downloads has it;
- the PC has an NVIDIA graphics card with its driver (version 527.41 or later);
- the licence includes it: the Professional licence, each seat of the Studio Pack, and the 30-day trial. The Student licence, and the free KiCad check after the trial, run on the CPU.

Otherwise the run is on the CPU, as before. The first line of the Log names the engine and, for the CPU, why: for example **Engine: GPU (NVIDIA GeForce RTX 5060, 8 GB)** or **Engine: CPU, 8 threads (no NVIDIA GPU)**. If the GPU engine cannot start (no usable device, too little GPU memory), the run starts again on the CPU by itself. Untick the menu item to compare with the CPU: the results can differ very slightly, because the run stops a few timesteps earlier or later.

Remote GPU (beta)

Beta: the remote GPU is still being tested. Contact us to test this feature (Tools > Remote GPU > Contact us, or lwtech.org/contact): you get the keys for it.

With the remote GPU, Run simulation sends the model through lwtech.org to a PC with an NVIDIA graphics card, where the LW RF Studio Agent runs openEMS. The log, the progress and the results come back as for a run on this PC, and this PC does not need openEMS for these runs. Neither PC opens a port: both only call lwtech.org, so it works through company firewalls.

- **On this PC:** Tools > Remote GPU (beta). Tick Run the simulations on the remote GPU, paste the client key and press Test: it lists the GPU computers that are online. The Run button then reads Run simulation (remote GPU).
- **On the GPU PC:** install openEMS (section 2), copy the LW RF Studio Agent folder to the PC and start LWRFStudioAgentApp.exe. Paste the agent key, give the PC a name and press Start. Start with Windows (ticked) starts the agent hidden in the notification area when you log on; closing its window keeps it working, and Exit in the menu of its icon stops it.
- **The agent window** draws the project being simulated, shows the progress, the simulations done and the log, and saves every project it receives in Documents\LW RF Studio Agent\Received projects, which LW RF Studio opens.

Stop on this PC stops the run on the GPU PC. A remote run cannot be paused, and Tune to frequency still runs on this PC.

Making a run faster

The run time grows with the number of cells times the number of timesteps. In order of effect:

- Use the GPU engine (see above) on a PC with an NVIDIA graphics card.
1. Use the Fast or Standard preset while tuning.
2. Raise the **Fine mesh on metal** a little (0.3–0.4 mm is often fine for a 2.4 GHz IFA).
3. Raise **Merge lines closer than** if the mesh has very small cells (shown as `min` in the status bar).
4. Narrow the sweep: a lower stop frequency gives larger cells.
5. Untick **Radiation pattern** and **Current distribution** when you only need S11.
6. For imported boards, untick the shapes far from the antenna.

Do not reduce the air margin or loosen the end criteria to save time: both change the result.

7. Running a simulation

Press F5 or click ▶ **Run simulation**. The program writes the model and the mesh, starts openEMS and switches to the Log tab. ■ **Stop** ends the run at any time. ⏸ **Pause** suspends openEMS: it keeps its memory but uses no CPU, and ▶ **Resume** goes on where it stopped. Closing the program ends a paused run.

How a run works

openEMS solves Maxwell's equations step by step in time (FDTD). It sends a short pulse covering the whole frequency band into the feed, then lets the field ring down. The pulse energy is radiated by the antenna or lost in the board, so the energy in the model falls. When it has fallen by the **end criterion** (−40 dB by default) the field is gone and the run stops. The frequency response is the Fourier transform of the voltage and current recorded at the feed.

That is why the run time depends on the antenna: a well-matched antenna radiates its energy quickly, while a badly matched or high-Q one rings longer.

Progress

The top of the Log tab shows:

- **The phase:** writing the model and mesh, setting up the FDTD operator, exciting the antenna, waiting for the energy to decay, reading the port results, and computing the far field.
- **A progress bar and the time left.** While the pulse is going in, the time left is not yet known. Once the energy starts to fall, it is estimated from how fast it falls.
- **The energy chart:** the energy in dB below its peak, against time, with the end criterion as a line. The run ends when the curve reaches the line.
- **Details:** the energy reached, the timestep, the pulse length, and the speed in million cells per second.
- The CPU and memory use of the PC, refreshed every few seconds.

The phase and percentage are also shown in the status bar and on the program's taskbar button, so you can follow a run while working in another window. Below the progress panel is the full openEMS output.

When it finishes

The program reads the results, shows the **S11** tab and writes a summary in the toolbar: the resonance frequency, the best S11, the input impedance at resonance and the −10 dB bandwidth.

The simulation folder

Each project simulates in its own folder, named after the project with `_sim` added:

- Next to the project file, for a saved project.
- In %TEMP%\LW RF Studio (the temporary folder of the system on Linux and macOS), for an unsaved project or one opened from a read-only place, such as the installed samples.

The folder holds the openEMS model and output, the log (`openEMS.log`), and the results as `s11.csv`, `antenna.s1p` and `pattern.csv`. A new run of the same project overwrites it.

Results saved with the project

File > Save also writes the last results into the project file: S11 and impedance, the radiation pattern, the current distribution, the reference trace and the matching network. Opening the project shows them again, even without the simulation folder, for example on another PC. If the design has changed since that simulation, the status bar says so: run it again to update the results. A project with a radiation pattern grows by about 0.5 MB.

After a simulation the project is marked as changed (* in the title bar), so you are asked to save the results when you close it. A project file without saved results takes them from its simulation folder, if it still exists.

Simulate > Load results from simulation folder shows the results of any earlier run of the open project. It also computes the radiation pattern if the run recorded the fields but nf2ff was not run.

If the run stops with an error

- **“openEMS ignored the feed port”** or **“No energy entered the model”:** the feed does not touch the copper, or it is shorted. The run is stopped early instead of running to the timestep limit. Check the feed position and layers on the Feeds & vias tab.
- **“openEMS exited with code …”:** see `openEMS.log` in the simulation folder.

- **“Fix the model errors first”:** the model has errors, listed in red under the editor tabs.

See also section 15.

If the program meets an error

When something goes wrong in the program itself, an error window opens instead of the program closing. Continue goes back to your work; save the project then. Close ends the program. The box Send this report to lwtech is ticked: it sends a short report, the error and the function where it happened (at most 255 characters, with the version and the system), and nothing from your project. Untick it to send nothing. An error that closes the program at once is shown at the next start.

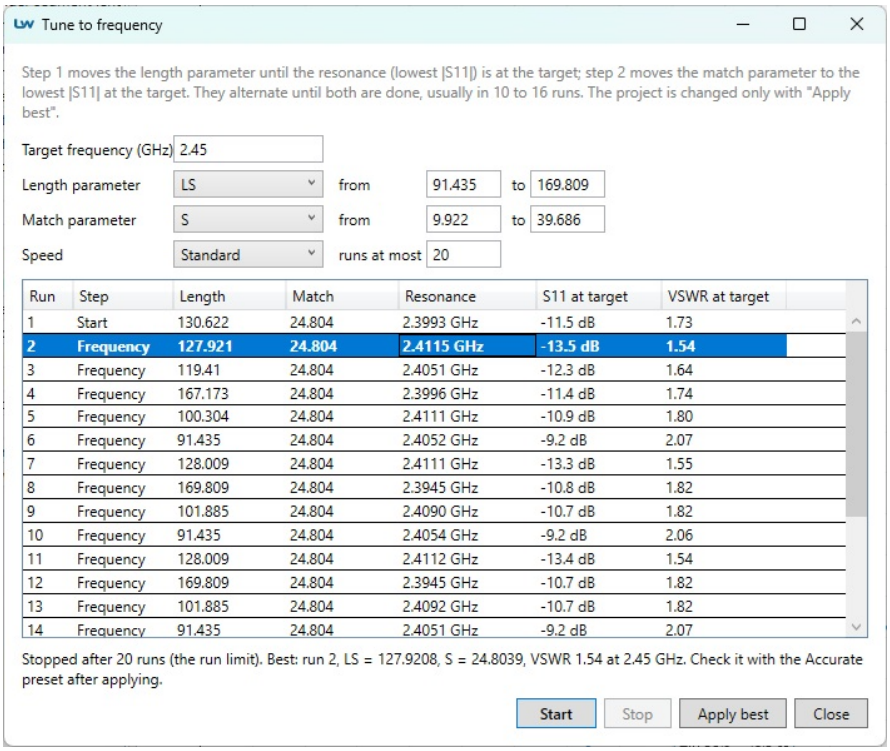
Tuning to a frequency

Simulate > Tune to frequency finds the dimensions that put the antenna at your frequency. It runs a series of simulations and changes two parameters between them:

- **Length parameter**: the one that sets the resonance, usually the length of the radiating arm. It is moved until the resonance, the lowest |S11|, is at the target. After the first run it is scaled by the found frequency divided by the target, as a resonant length goes with 1/f; then it is refined from the last two runs.
- **Match parameter** (optional): the one that sets the impedance at resonance, for example the feed-to-short spacing of an IFA. It is moved to the lowest |S11| at the target: one step on each side, then the bottom of the parabola through the three results.

The two steps alternate until the resonance is within 0.5% of the target and the match no longer improves by more than 0.5 dB. This usually takes 6 to 16 runs; with the Fast estimate preset, a run of a small antenna takes a few tens of seconds.

1. Check that the target frequency is inside the sweep (Board & simulation tab), and that the two parameters are plain numbers, ideally with a Min and a Max in the Parameters tab. Parameters that depend on them by formula follow automatically.
2. Open Simulate > Tune to frequency. Check the target, the two parameters and their ranges (without Min and Max: ±30% for the length and ±60% for the match), the speed and the maximum number of runs.
3. Click Start. The table lists every run: the values, the resonance, and S11 and VSWR at the target. Stop ends the tuning after the current run.
4. The best run is in bold. Click Apply best to write its values into the project; nothing changes before that. The values are written with 2 decimals.
5. Run the simulation with the Accurate preset for the final check: the Fast and Standard presets can be a few percent off.



The Tune to frequency window after a tuning to 2.45 GHz: every run with its resonance, S11 and VSWR at the target. Apply best writes the best run into the project.

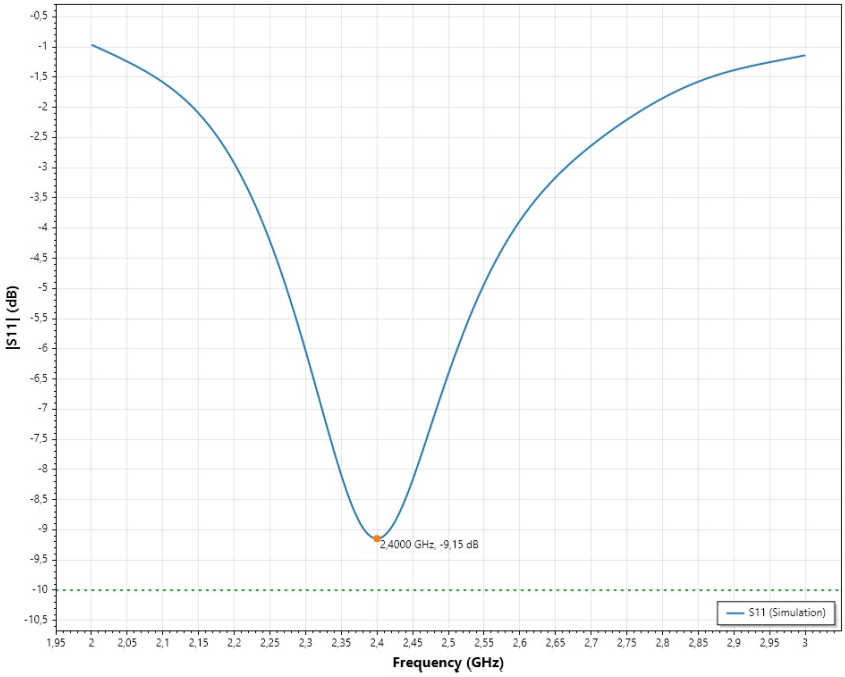
If a parameter reaches its bound before the resonance reaches the target, the tuning stops and says which bound to widen. The same tuning is available on the command line (section 13).

8. Results: S11, impedance, VSWR and Smith chart

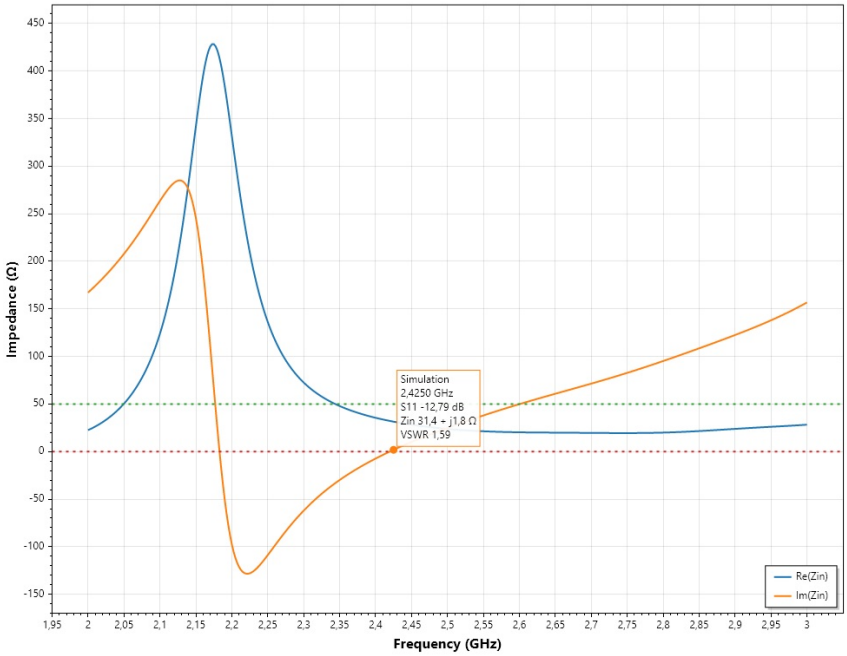
Four tabs show how well the antenna is matched to the radio, across the sweep. All are referred to the feed impedance Z0 (normally 50 Ω).

Tab	Shows	What to look for
S11	The reflection coefficient |S11| in dB, with a dotted line at −10 dB and a marker at the minimum.	The dip is the resonance. Below −10 dB, less than 10% of the power is reflected.
Impedance	The input resistance Re(Zin) and reactance Im(Zin), with dotted lines at Z0 and at 0 Ω.	At a good match Re(Zin) is near Z0 and Im(Zin) crosses 0.
VSWR	The voltage standing-wave ratio, with a dotted line at 2.	VSWR below 2 is about the same as S11 below −10 dB.
Smith chart	The input reflection on the Smith chart.	A good match passes close to the centre.

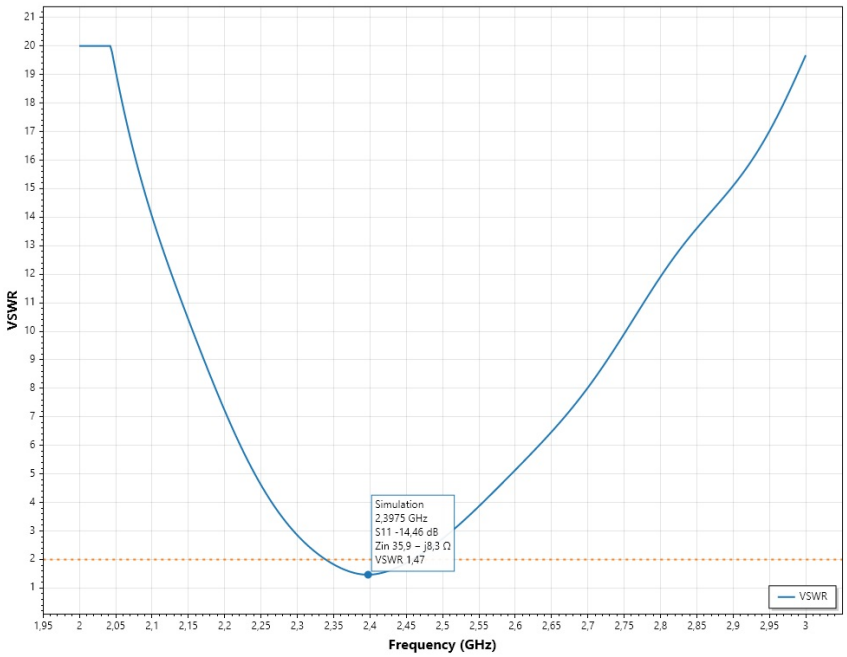
The toolbar summary gives the resonance frequency, the best S11, the impedance at resonance and the −10 dB bandwidth.



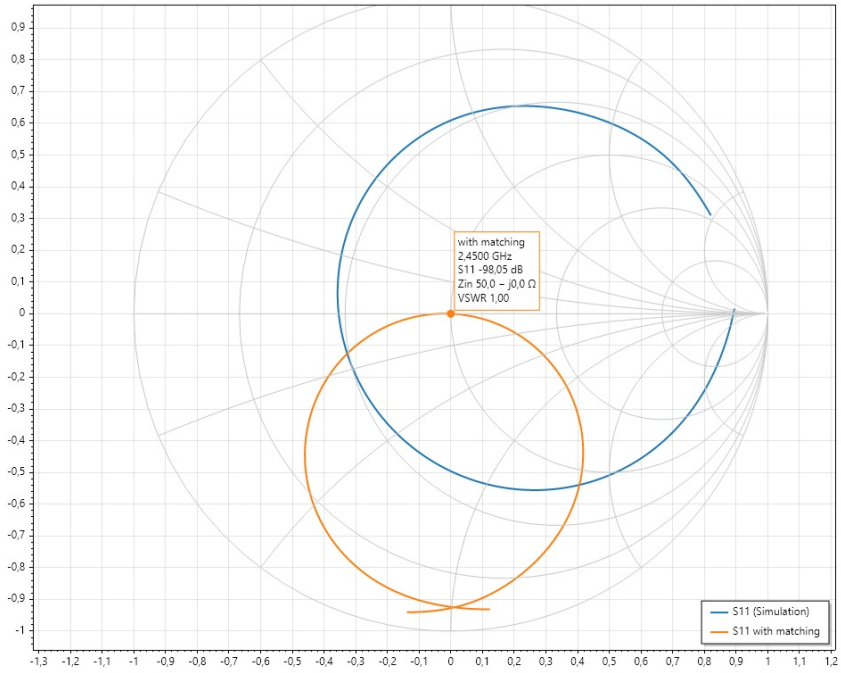
The S11 tab: the marker is at the lowest point, with the dotted line at −10 dB.



The Impedance tab: Re(Zin) and Im(Zin) with the 50 Ω and 0 Ω lines; a click gives S11, Zin and VSWR at that frequency.



The VSWR tab, with the dotted line at 2.



The Smith chart tab: the simulated antenna (blue) and the same antenna with the matching network (orange), which brings it to the centre.

Reading values on the charts

- **Data tips:** click near a curve to pin a tip with the values of the nearest point (frequency, S11, impedance, VSWR). Ctrl+click adds another tip. Click in empty space, or press Esc, to remove them.

- **Zoom and pan:** drag to pan, and turn the mouse wheel to zoom. Middle-click to fit the curves again.
- **Save or copy a chart:** right-click it.

Comparing with a measurement

Simulate > Load reference S11 overlays a measured trace, for example from a VNA, as a second curve. It reads:

- A Touchstone `.s1p` file.
- A `.csv` file with the frequency in the first column (in Hz, or GHz if all values are below 1000) and S11 in dB in a column named `S11_dB`, or else the second column.

The reference stays on the charts until **Simulate > Clear reference**.

Matching network

The **Matching** tab, with the editors on the left, shows the effect of an L/C network between the antenna and the radio without a new simulation. Set the parts in three places, as on a VNA's L/C match screen: a **source shunt** part on the radio side, a **series** part, and a **load shunt** part across the antenna. For each, choose None, L or C and enter the value in nH or pF. The charts on the right update as you type, and the tab shows the matched result.

Component values: choose **Ideal values**, or the **Würth design kits**: WE-MK 0402 inductors (kit 744784, 33 values from 1 to 270 nH) and WCAP-CSRF 0402 NP0 capacitors (kit 885390, 36 values from 0.2 to 27 pF). With the Würth kits, each value box lists the kit values, the Würth order code is shown next to it (or "not in the kit"), and switching to the kits moves every L and C to the nearest kit value. The values and order codes come from the kit datasheets; check the current datasheets before ordering.

Automatic proposal: enter the frequency in **Match at** (the pattern frequency by default) and click **Propose**. The tab lists L-networks that match the antenna to Z0 there, as SmithMatch's Auto-match does:

- With **ideal values**, the exact solutions: a shunt part next to the antenna then a series part, or a series part then a shunt part, each in its low-pass and high-pass form where it exists.
- With the **Würth kits**, every layout (one part, or a shunt and a series part in either order, each an L or a C) is tried with every kit value, and the best pair for each layout is listed. Pairs whose second part hardly helps, and parts that barely improve on the bare antenna, are left out.

Each proposal gives its parts (with the order codes), S11 at the chosen frequency, and the −10 dB band it reaches on the simulated curve, so you can prefer a wider band to a deeper dip. Click a proposal to put it in the three places and try it on the charts.

SmithMatch…, next to **Propose**, opens SmithMatch with the simulated antenna impedance at the **Match at** frequency as its start datapoint, and sets its Z0 and frequency to match. Use it to design the network by hand on the Smith chart (section 12). It needs a simulation result.

The tab also has a **Ladder** field with the same network as text, listed from the antenna towards the radio, which takes networks with any number of parts. Type it and press Enter:

```
shuntC=1.8p; seriesL=6.8n
```
This is a 1.8 pF capacitor across the antenna port, followed by a 6.8 nH inductor in series towards the radio.

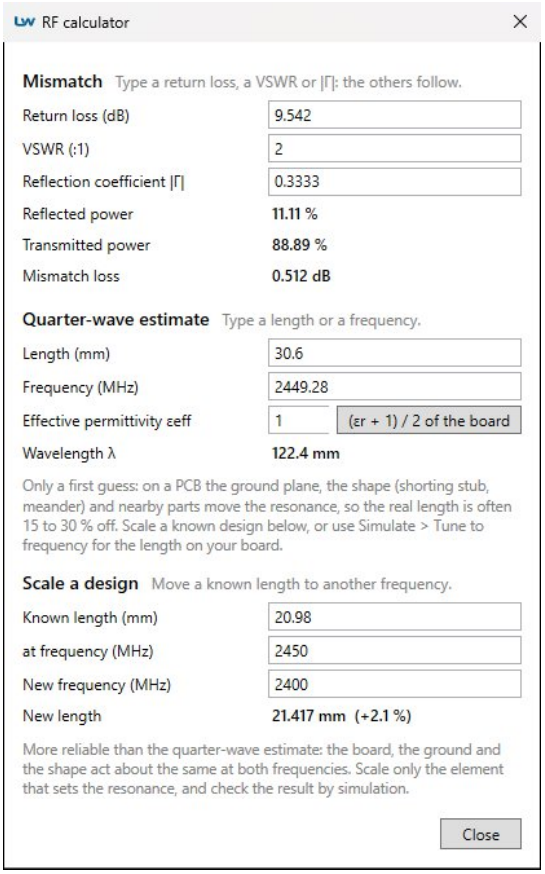
- Each component is `series` or `shunt`, followed by `L` or `C`, `=` and a value.
- Values take the prefixes f, p, n, u, m, k and M, for example `6.8n` (H) or `1.8p` (F).
- The components are ideal (no losses or parasitics).

The charts then show the curves with matching next to the originals, and the toolbar summary adds the matched resonance and bandwidth. **Clear** removes the network. The network is saved with the project.

RF calculator

Tools > RF calculator converts the figures of a match into each other and estimates lengths. It stays open beside the main window.

- **Mismatch**: type a return loss, a VSWR or the reflection coefficient |Γ|, and the other two follow, with the reflected power, the transmitted power and the mismatch loss. An S11 in dB is accepted too: −10 and 10 are the same match.
- **Quarter-wave estimate**: the frequency at which a length is a quarter wavelength, or the length for a frequency. The effective permittivity εeff is 1 in air; the button (εr + 1) / 2 of the board uses the substrate of the project, the most a printed antenna sees. On a PCB the real length is often 15 to 30% off this estimate, because the ground plane, the shape and the parts nearby move the resonance.
- **Scale a design**: moves a known length to another frequency (length × known frequency / new frequency). This is more reliable than the quarter-wave estimate, because the board, the ground and the shape act about the same at both frequencies. Scale only the element that sets the resonance, and check the result by simulation.



The RF calculator.

9. Radiation pattern

The radiation pattern shows how the antenna radiates in each direction. It needs **Radiation pattern** ticked on the Board & simulation tab before the run, and it is computed at the **Pattern frequency** only (the centre of the sweep if left at 0). During the run openEMS records the fields on a box around the antenna; afterwards its nf2ff tool turns them into the far field. To see the pattern at another frequency, change the pattern frequency and run again.

The 2D pattern

The **Radiation pattern** tab shows three polar cuts through the pattern, in the XY, XZ and YZ planes. The heading line gives:

- The frequency.
- The **directivity** (dBi): how strongly the antenna concentrates its radiation in the best direction.
- The **efficiency**: the share of the accepted power that is radiated. The rest is lost in the copper and the board.
- The **peak gain** (dBi) and its direction, as θ (from the z axis) and ϕ (from the x axis in the board plane).

Each cut shows three curves: **vertical** (Eθ), **horizontal** (Eϕ) and **total**. Vertical and horizontal are the polarisations seen with the board lying flat, z up. Click a curve for a data tip.

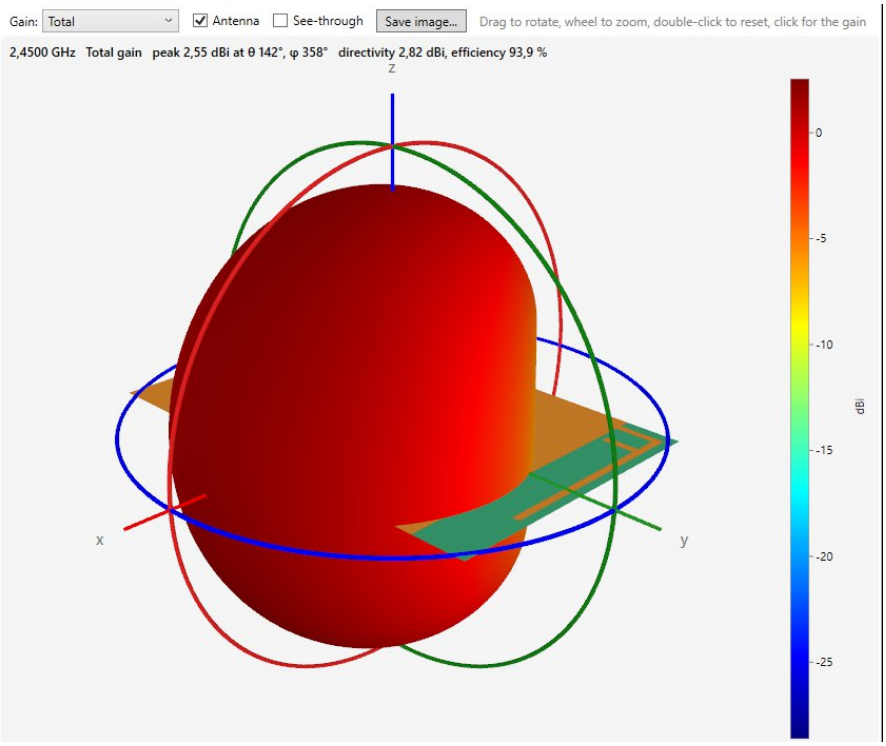
Axes: x runs along the board, y from the ground towards the antenna, and z is normal to the board.

Gain here is computed from the power the antenna accepts: it includes the losses in the copper and the dielectric, but not the mismatch at the feed. A poorly matched antenna radiates less in practice than its gain suggests, so check S11 too.

The 3D pattern

The **3D pattern** tab draws the gain in every direction as a coloured surface, with the antenna inside it. The distance from the centre and the colour both show the gain, over 40 dB below the peak; deeper nulls are drawn at the centre.

- **Gain:** show the **Total**, the **Vertical (Eθ)** or the **Horizontal (Eφ)** gain.
- **Antenna:** show or hide the board and copper in the middle.
- **See-through:** makes the pattern transparent, so you can see the antenna through it.
- **Mouse:** drag to rotate, turn the wheel to zoom, and double-click to return to the start view. Click the pattern to read the total, vertical and horizontal gain in that direction.
- **Save image:** saves the view as a PNG file.



The 3D pattern of the Printed IFA at 2.45 GHz.

10. Current distribution

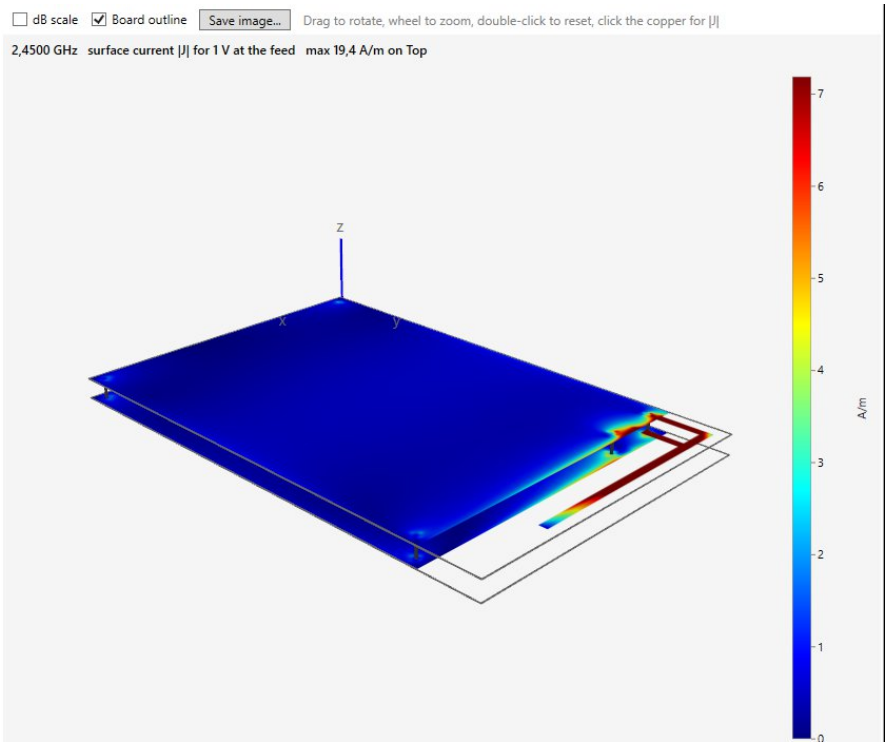
The **Current** tab shows the surface current flowing on the copper, like MATLAB's `current()` function. It tells you which parts of the antenna carry the current and therefore do the radiating, where the current crowds, and how far it spreads into the ground plane.

It needs **Current distribution** ticked on the Board & simulation tab before the run. Like the radiation pattern, it is recorded at the **Pattern frequency** only. It adds little to the run time.

Reading the view

Each copper layer is drawn in 3D and coloured by the magnitude of the surface current |J|, on a jet colour scale (blue for weak, red for strong). The current is in A/m for 1 V at the feed. The heading line gives the frequency and the highest current, and the layer it is on.

On a resonant inverted-F the current is highest at the feed and the short, and falls to nearly zero at the open end of the arm.



The surface current on the Printed IFA: strongest on the arm near the feed and the short.

Controls

- **dB scale:** colours by 20·log10(|J| / maximum), down to −40 dB. This shows the weak currents too, for example on the ground plane.
- **Board outline:** shows or hides the board edge.
- **Mouse:** drag to rotate, turn the wheel to zoom, and double-click to return to the start view. Click the copper to read the layer, the position and |J| at that point.
- **Save image:** saves the view as a PNG file.

How it is computed

openEMS records the magnetic field just above and just below each copper layer. The surface current is the jump in the magnetic field across the copper sheet. This is accurate on large copper areas and along traces. Very close to a via or the feed, where the field changes fast over one cell, treat the value as approximate.

11. Exporting

All exports are on the **File** menu.

The antenna as a PCB part

Once the antenna works, export it as a part to place on your board.

	KiCad footprint (`.kicad_mod`)	Eagle library (`.lbr`)
Opens in	KiCad 7, 8 and 9	Eagle 6 to 9 and Fusion Electronics
Copper	Filled polygons on F.Cu, In*.Cu and B.Cu	Rectangles and polygons on layers 1 (Top), 2–15 (inner) and 16 (Bottom)
Feed	SMD pad 1	SMD pad, pin FEED
Vias	Plated holes, pad 2	Plated pads, pin GND
Also	Pads 1 and 2 are a net tie, because the antenna shorts the feed to ground on purpose	A two-pin symbol (FEED, GND) and a device; the board outline on layer 51 (tDocu)

In both, the origin of the part is the feed point. The project name names the part, and the project description is written into it. All copper in the project is exported, including the ground planes. If your board has its own ground, untick the **On** box of the ground shapes before exporting, or delete them from the part afterwards.

Vias without a diameter are given a default drill. Check the drill and pad sizes against your PCB rules before ordering boards.

Export Altium footprint library (.PcbLib) writes the same part for Altium Designer: the copper as regions, the feed as pad 1 and the vias as pad 2.

Results

- **Export results (CSV / Touchstone):** the simulated S11 as a Touchstone `.s1p` file, for a circuit simulator or VNA software, or as a `.csv` file with the columns `f_Hz, Re_Zin, Im_Zin, S11_dB, VSWR`.
- **Export radiation pattern (CSV):** the gain at every angle, with the columns `theta_deg, phi_deg, G_theta_dBi, G_phi_dBi, G_total_dBi`. The first line gives the frequency, the directivity and the efficiency.

PCB Antenna Lab also writes these files to the simulation folder after every run: `s11.csv`, `antenna.s1p` and `pattern.csv`.

Images

Right-click a chart to save or copy it. The 3D pattern and the Current views have a **Save image** button that saves a PNG file.

Reports

File > Print report (Ctrl+P) and **File > Export report as PDF** make a report of the design, one view per A4 page: the geometry, the parameters and settings, S11, impedance, VSWR, the Smith chart, the radiation patterns and the current distribution. A window lets you choose the pages.

openEMS model

Export openEMS XML writes the model and its mesh (`antenna.xml`) to a folder you choose, without running it. Use it to run openEMS yourself, for example on another computer, or to inspect the model in openEMS's AppCSXCAD viewer.

12. SmithMatch

SmithMatch designs an L/C matching network on an interactive Smith chart. You enter the impedance of the load, for example the simulated antenna or one measured with a VNA, add parts to a pi network and see live how each part moves the impedance towards Z0. Values can be limited to the parts of the Würth Elektronik design kits, so the network can be built on the bench at once.

Opening SmithMatch

- **Tools > SmithMatch** (Ctrl+M) in PCB Antenna Lab opens it in its own window, so you can keep both in view. It needs no simulation.
- **SmithMatch…** on the Matching tab (section 8) sends the simulated antenna impedance at the **Match at** frequency. It becomes the start datapoint, and Z0 and the frequency are set to match.

Closing PCB Antenna Lab also closes SmithMatch, which first asks to save an unsaved design.

The window

Area	What it holds
Smith chart (left)	Impedance (Z) and admittance (Y) grids scaled to Z0, the datapoints, the path of the network and the + / − arrows. The corner gives the VSWR and return loss of the whole network.
Toolbar	**Mouse adds DP**, **Keyboard DP...**, **Z0 [Ω]**, **Freq [Hz]**, **Würth kit values only**, the part buttons (**Load**, **Serie** and **Source**, each **L** or **C**), **Auto-match** and **VSWR sweep...**
Datapoints	The impedances entered, with their Q and frequency. Tick **Start** on the one to match; right-click to delete one.
Schematic	The network drawn as a circuit with values and Würth order codes, the tuning slider, and the result: Zin, |Γ|, VSWR and return loss.
Cursor	Return loss, VSWR, Q, Γ (magnitude and angle), Z and Y at the mouse position, with Z0 and the frequency.

Turn the mouse wheel on the chart to zoom; **View > Zoom 1:1** resets it, and the **View** menu shows or hides each grid. Values accept engineering suffixes, such as `2.45G`, `3.3n` and `1.5p`.

Designing a network

The network is a fixed pi: a **Load** part to ground across the antenna, a **Serie** part, and a **Source** part to ground on the radio side. Each place holds an L, a C or nothing. This is the same layout as the load shunt, series and source shunt places on the Matching tab.

1. Set **Z0** and the frequency in the toolbar.
2. Add the load: tick **Mouse adds DP** and click the chart, click **Keyboard DP...** to type R and X, or send it from PCB Antenna Lab.
3. Click a part button. A green circle shows every point that part can reach, and the value follows the mouse along it. Click to fix the value, or press Esc to cancel.
4. Or click **Auto-match**. It lists every L-network (low-pass and high-pass) that matches the start datapoint to Z0, with the VSWR each one reaches. Select one to preview it on the chart, then click **Use**.
5. Tune: click a part in the schematic and drag the slider; double-click a part to type its value, right-click to remove it. The + and − arrows on the chart show which way each point moves when a value goes up or down, which helps when tuning on the bench.
6. Check the bandwidth with **VSWR sweep...**, then save and print.

The parts are ideal: no losses, parasitics or self-resonance. To see a SmithMatch network on the simulated curves, enter its parts on the Matching tab.

Würth design kits

Tick **Würth kit values only** to snap every value to the nearest part of the kits. They are the same kits as on the Matching tab:

Parts	Kit	Values
Inductors, WE-MK 0402	744784	33 values, 1 to 270 nH
Capacitors, WCAP-CSRF 0402 NP0	885390	36 values, 0.2 to 27 pF

With the kits on, the slider steps from one kit value to the next, Auto-match rounds its values to kit parts, and the order codes are shown in the schematic, the Auto-match list and the printout. Check the current Würth datasheets before ordering.

VSWR sweep, files and printing

- **VSWR sweep...** draws the VSWR against frequency over a range you choose (±20% around the design frequency by default), with the minimum VSWR and a cursor readout. The load is modelled as its resistance plus the L or C that gives its reactance at the design frequency, so the sweep is approximate far from that frequency.
- **File > New** (Ctrl+N), **Open** (Ctrl+O), **Save** (Ctrl+S) and **Save as**: designs are saved as `.smith` files.
- **File > Export image** saves the chart as a PNG file.
- **File > Print** (Ctrl+P) and **Print preview** open the same dialog with a live preview. Choose the chart layers (Z and Y grids, labels, traces, arrows, datapoints), the sections (schematic, datapoints, components, results, VSWR sweep), and portrait or landscape. The chart can fill page 1 on its own.

13. Command line

LWRFStudioCli (LWRFStudioCli.exe on Windows), in the installation folder (on macOS inside the app, LW RF Studio.app/Contents/MacOS), does the work of PCB Antenna Lab without the window. Use it for scripts, batch runs and parameter sweeps. It runs during the trial or with a license. After the trial, without a license, only licence, status, engine, vswr and feed-preview run (the last three are used by the KiCad plugin); the others stop with exit code 3.

Run it from a command prompt or a terminal in the installation folder, or add that folder to the PATH. Without arguments it lists the commands.

Command	What it does
`new <template#> <project.json>`	Creates a project from a template. Without arguments, lists the templates.
`import <script.m> <project.json>`	Imports a MATLAB pcbStack script.
`import-pcb <board> <project.json>`	Imports an Eagle .brd, Fusion .fbrd, KiCad .kicad_pcb or Altium .PcbDoc board.
`import-gerber <folder, set.zip or files> <project.json>`	Imports a Gerber set with its Excellon drills.
`check <project.json>`	Evaluates the geometry and the mesh and prints statistics.
`geom <project.json> <out.png>`	Draws the layers, feeds and vias, seen from above.
`xml <project.json> <dir>`	Writes the openEMS model to a folder.
`run <project.json> <dir> [openEMS.exe]`	Simulates, prints a summary of S11 and writes `s11.csv` and the `.s1p` file.
`vswr <board> <dir> [--start GHz] [--stop GHz] [--fast | --standard] [--feed pad | --feed-at x,y] [--gain [--gain-ghz GHz]]`	Imports a board, simulates its antenna and writes s11.csv; free, also after the trial (used by the KiCad plugin). **--gain** adds the far field and prints the efficiency and the peak gain at --gain-ghz (default the middle of the sweep) on a GAIN line; only with the Accurate preset (not with --fast or --standard), and free too. --feed-at x,y puts the feed at a point in mm, on the copper there.
`feed-preview <board> <out.png> [--feed pad | --feed-at x,y] [--view x,y,half]`	Draws the area around the feed, or the window --view (its centre and half width in mm), and prints where the picture lies in mm (used by the KiCad plugin to zoom the picture and place the feed with a click).
`post <project.json> <dir>`	Post-processes an existing simulation folder again.

Command	What it does
`plot <dir> <out.png> [reference]`	Plots the S11 and VSWR of a run, optionally against a `.csv` or `.s1p` reference.
`match <project.json> <dir> "shuntC=1.8p; seriesL=6.8n" [out.csv]`	Applies a matching network to the result of a run.
`automatch <project.json> <dir> <GHz> [wurth]`	Proposes L-networks matching the result of a run at a frequency, with ideal values or Würth kit parts.
`tune <project.json> <dir> --f0 GHz --length NAME [--match NAME]`	Tunes the length parameter to resonate at f0, then the match parameter to the lowest S11 there; writes tuned.json (section 7).
`pattern <project.json> <dir> [GHz]`	Computes the far field of a finished run: `pattern.csv` and polar plots as PNG.
`currents <project.json> <dir> [out.csv]`	Writes the surface current of a finished run, per copper layer.
`export-kicad <project.json> <out.kicad_mod>`	Exports a KiCad footprint.
`export-eagle <project.json> <out.lbr>`	Exports an Eagle library.
`export-altium <project.json> <out.PcbLib>`	Exports an Altium footprint library.
`status`	Prints full, trial or over for this PC, with the details and the PC ID.
`engine [openEMS]`	Shows the openEMS it uses and the engine of the simulations: GPU, or CPU with the number of threads and why not the GPU. Runs also after the trial.
`licence [licence.txt]`	Shows the license or trial status and this PC's ID, or installs a license file.

The command line uses the settings saved in the project, and finds openEMS in the same places as the program unless you give its path to run. It uses the GPU engine like the program (section 6); **--cpu**, with any command, runs on the CPU.

Example: a quick sweep

A project is a JSON text file, so a script can change a parameter's expression, run the simulation and collect the results. For example, in PowerShell:

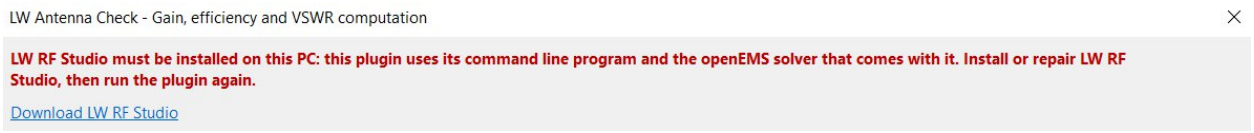
```
$cli = "C:\Program Files\LW RF Studio\LWRFStudioCli.exe"
foreach ($l in 9, 10, 11) {
  $p = Get-Content ifa.json -Raw | ConvertFrom-Json
```

```
  ($p.Parameters | Where-Object Name -eq 'LE').Expression = "$l"
  $p | ConvertTo-Json -Depth 10 | Set-Content "ifa_LE$l.json"
  & $cli run "ifa_LE$l.json" "sweep_LE$l"
}
```

14. KiCad plugin: LW Antenna Check

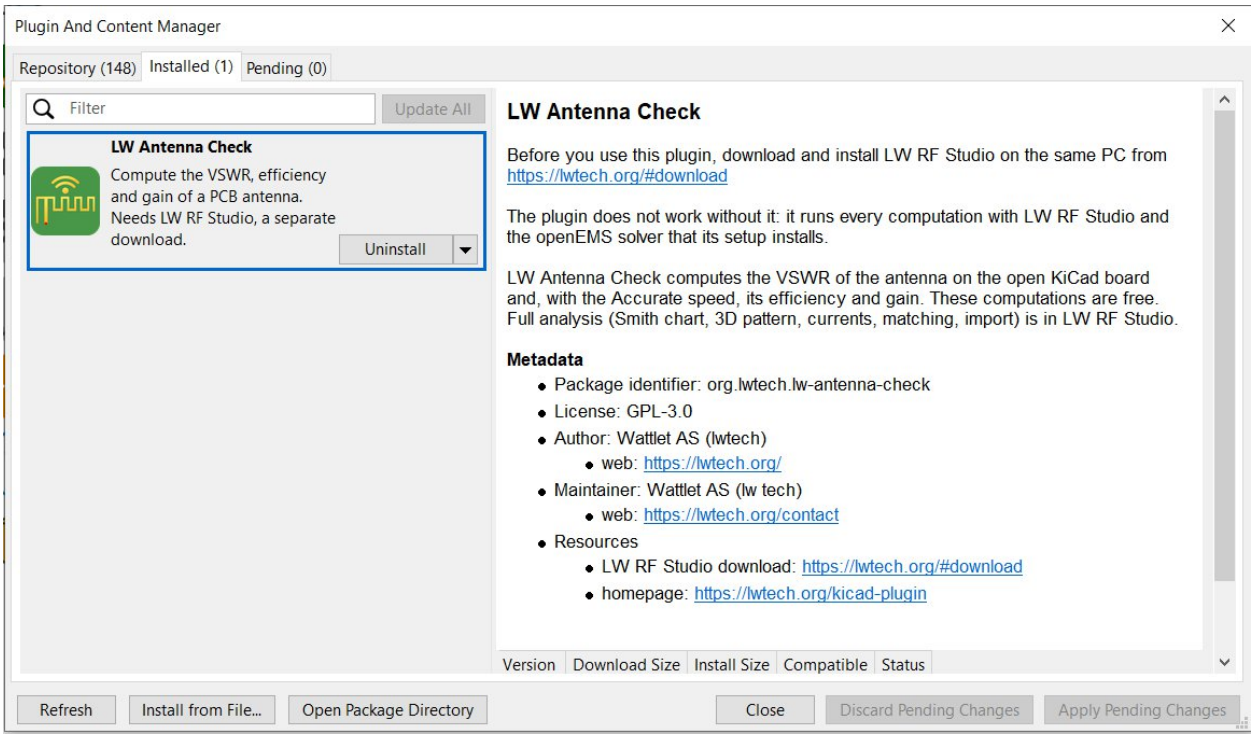
LW Antenna Check simulates the VSWR of the antenna on the board open in KiCad's PCB editor (KiCad 7 to 10, on Windows, Linux and macOS) and, with the Accurate speed, computes its efficiency and gain. It uses the command line of LW RF Studio, so LW RF Studio must be installed on the same PC; the plugin finds it by itself (on Linux through the menu entry of install.sh or in ~/opt, on macOS in Applications).

It shows a warning if the application is not installed. The Kicad plugin is free and 100% unlimited in time but the process engine is a part of the application, Even if the trial period has expired, the plugin is still working. Click on the link to install the application. Download LW RF Studio



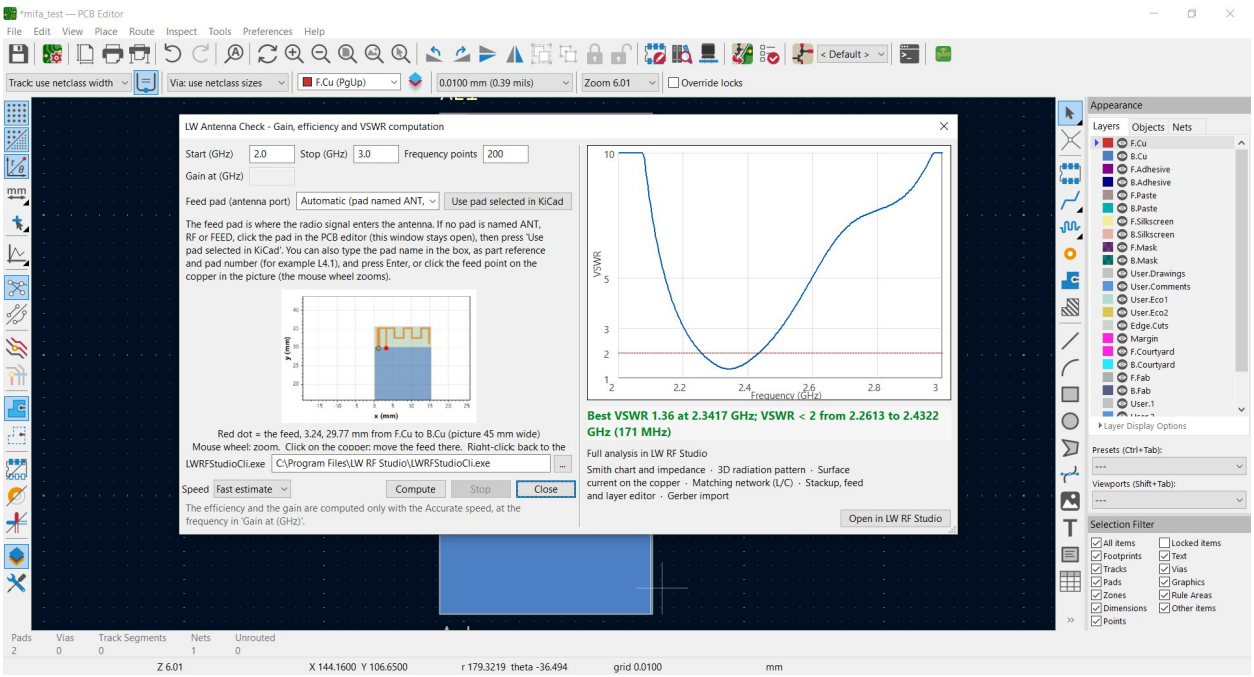
On Windows, setup installs the plugin when it finds KiCad (tick the KiCad option). On every system it can be installed with KiCad's Plugin and Content Manager, or by copying the folder kicad-plugin/lw_antenna_check of the package to KiCad's plugin folder:
~/.local/share/kicad/<version>/scripting/plugins on Linux,
~/Documents/KiCad/<version>/scripting/plugins on macOS.

To install it with KiCad's Plugin and Content Manager, download lw_antenna_check-pcm-<version>.zip from lwtech.org/kicad-plugin (the same file for Windows, Linux and macOS; do not unzip it). In the KiCad main window click Plugin and Content Manager, then Install from File..., choose the zip file and click Apply Pending Changes. Restart the PCB editor if it was open.



KiCad's Plugin and Content Manager with LW Antenna Check installed.

1. Save the board: the last saved version is simulated.
2. Open Tools > External Plugins > LW Antenna Check.
3. Set the frequency range and the number of points.
4. Choose the feed pad, where the radio signal enters the antenna: Automatic takes a pad named ANT, RF or FEED; or pick a pad from the list, or select it in the PCB editor. A picture shows the area around the chosen feed: turn the mouse wheel over it to zoom in and out, click on the copper to put the feed exactly there, and right-click to go back to the view around the feed. Compute and Open in LW RF Studio use the point you clicked.
5. Choose the speed (Fast estimate, Standard or Accurate) and click Compute. The VSWR curve shows the best match and the band below VSWR 2; click the curve to read a value. Only **Accurate** also gives the **efficiency** and the **gain**: the box **Gain at (GHz)** is the frequency they are computed at (empty: the middle of the range), because the far field is computed at one frequency. They appear under the curve as, for example, “Efficiency 81.6 %, peak gain 1.03 dBi (directivity 1.91 dBi) at 2.4500 GHz”. The gain is the directivity times the efficiency, in the direction where it is highest; the mismatch of the feed is not included.



LW Antenna Check in the KiCad PCB editor: the settings and the feed picture on the left, the VSWR curve and the best match on the right.

The VSWR, efficiency and gain simulations are free and keep working without a licence (after the trial, on the CPU). Open in LW RF Studio opens the board, with the chosen feed, in the full program. When LW RF Studio or openEMS is missing, the window says so, with a link to download LW RF Studio.

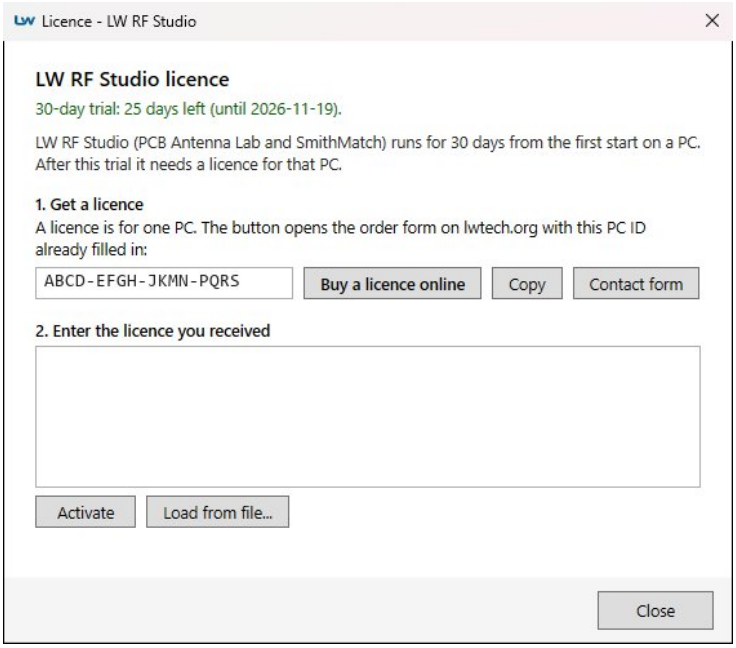
15. Trial and license

The trial period is valid 30-days and it is locked to the PC. It cannot be renewed.

Getting a license

1. Open **Help > Licence**, or use the license window that opens after the trial. It shows this PC's ID, a 16-character code.
2. Click **Buy a license online**: the order form of lwtech.org opens with this PC's ID already filled in (it is also copied to the clipboard), and you pay by card. To ask a question first, use **Contact form**, or **Copy** to copy the ID.
3. Wattlet AS sends you the license by e-mail, usually within one working day, with the invoice: a block of text from a BEGIN line to an END line, or a license.txt file.
4. Paste the whole license in the box and click **Activate**, or click **Load from file** and choose the file.

The program is unlocked at once; there is no need to restart. The same window then shows who the license is for and, if it has one, its end date.



The license window (the PC ID here is a sample).

Good to know

- A license works on one PC. The PC ID comes from the installation of the operating system (Windows, Linux or macOS): it stays the same when you update the program.
- The Professional licence, also each seat of the Studio Pack, includes the GPU engine (section 6). The Student licence runs the simulations on the CPU; the licence window shows Student after the name. The 30-day trial includes the GPU engine.
- A license can be revoked by Wattlet AS, for example after a refund.

Licence messages

Message	Meaning
This license is for the PC with ID …	The license was issued for another PC. Send this PC's ID to get one for it.
This license expired on …	The license has an end date that has passed. Contact lwtech.org to renew it.
The license text is damaged	Part of the text is missing. Copy all of it, including the BEGIN and END lines.
The license is not valid	The text was changed, or it was not issued by Wattlet AS.
This is not an LW RF Studio license	The license is for another program, such as PCB Antenna Designer. Send this PC's ID to get an LW RF Studio license.

Message	Meaning
The license of this PC was revoked	Wattlet AS revoked the license, for example after a refund. Contact lwtech.org.

Legal

LW RF Studio is Copyright © 2026 Wattlet AS, all rights reserved; LW RF Studio and lw tech are brands of Wattlet AS. Its use is governed by the license agreement shown at installation (EULA.txt in the installation folder). The program uses third-party components under their own licenses, listed in THIRD-PARTY-NOTICES.txt and the third-party-licenses folder. The openEMS solver, by Thorsten Liebig and contributors, is free software under the GNU GPL v3 and is not part of the program: on Windows the setup downloads it from lwtech.org, with its licence (COPYING.txt) and the address of its source code (SOURCE.txt); on Linux and macOS you build it from its source. Help > About LW RF Studio shows the version and these credits.

16. Tips and troubleshooting

A tuning workflow

1. Start from the template closest to your antenna, or import your board.
2. Set the real stackup: thickness, εr and tanδ of your PCB material.
3. Choose the Fast or Standard preset and run. Check that the resonance is inside the sweep.
4. Tune with the slider and re-run until the resonance and the match are where you want them, or let Simulate > Tune to frequency do it (section 7). For an inverted-F:
    - A longer arm lowers the resonance; a shorter arm raises it.
    - Moving the feed further from the short raises the input resistance; moving it closer lowers it.
5. Choose **Accurate** and run again for the final result.
6. If needed, try a matching network on the Matching tab, or design one in SmithMatch.
7. Export the antenna as a KiCad footprint, an Eagle library or an Altium footprint library.

Why simulation and measurement differ

A few percent between the simulated and measured resonance is normal. Common causes:

- **εr of FR-4** varies between manufacturers and with frequency. Use the value your board maker gives at your frequency.
- **The ground plane:** an IFA uses the ground as part of the antenna, so its size matters. Simulate the ground you will really have.
- **Things not in the model:** the enclosure, the battery, nearby components, the solder mask and the copper thickness. Copper is modelled as a zero-thickness sheet.
- **The measurement:** the test cable and the hand near the board detune the antenna.

To compare, load the measured S11 with **Simulate > Load reference S11**.

Problems and solutions

Problem	What to do
“openEMS was not found”	Install openEMS (section 2) and select it with Tools > Locate openEMS.exe (Locate openEMS on Linux and macOS).
The run stops at once: “openEMS ignored the feed port” or “No energy entered the model”	The feed does not touch the copper or is shorted. Check its position and layers, and that copper exists on both of its layers at that point.
The run takes very long or reaches the maximum timesteps	The antenna is badly matched or high-Q, or the feed excites almost nothing. Check the feed, and that the resonance is inside the sweep. See “Making a run faster” in section 6.
No dip in S11	The resonance is outside the sweep. Widen the start and stop frequencies, or check the dimensions.
The mesh is very large (millions of cells)	Look at the mesh with the **Mesh** box. Very small cells come from tiny gaps or offsets between shapes; align the shapes or raise **Merge lines closer than**. For imported boards, untick the shapes far from the antenna.
No radiation pattern	Tick Radiation pattern and run again. Check that nf2ff (nf2ff.exe on Windows) is in the openEMS folder.
The Current tab is empty	Tick **Current distribution** and run again.
An imported board has no dielectric	The board outline is not closed. Close it on Edge.Cuts (KiCad) or the Dimension layer (Eagle), or add a Board shape.
The results are from an older design	Results on the tabs are from the last run. Run again after changing the design.
The PC is slow during a run	Set **Threads** to fewer than the number of cores.
The run uses the CPU, not the GPU	The first line of the Log says why (section 6): the GPU engine is turned off in the Tools menu, this openEMS has no GPU engine (on Windows, run the setup again to get it), there is no NVIDIA GPU, or the licence is a Student licence.
SmithMatch… says “Run a simulation first”	SmithMatch takes the simulated antenna impedance. Run a simulation, or open a project

Problem	What to do
	saved with its results.
Only the license window opens	The 30-day trial has ended. Enter a license (section 14); closing the window without one closes the program.
LW RF Studio is not in the application menu (Linux)	install.sh was run with sudo: run sudo install.sh --uninstall, then install.sh as yourself.
macOS says the app is damaged or cannot be opened	The app is not signed by Apple yet: remove the download mark once (section 2): xattr -dr com.apple.quarantine "/Applications/LW RF Studio.app".

When a problem persists, send the project file and openEMS.log from the simulation folder with the contact form on lwtech.org.

17. Quick reference

Keyboard

Key	Action
F5	Run the simulation
F1	Open the user manual (PDF)
Delete	Remove the selected shapes, feeds, vias and holes (Geometry view) or rows (the table that has the focus)
Ctrl+O	Open a project
Ctrl+S	Save the project and its results
Ctrl+I	Import a PCB board
Ctrl+M	Open SmithMatch (Tools > SmithMatch)
Ctrl+P	Print a report
Ctrl+Z	Undo the last change to the design
Ctrl+C, Ctrl+V (in a table)	Copy the value of a cell and paste it into another
Ctrl+Y, Ctrl+Shift+Z	Redo
Arrow keys (Geometry view, items selected)	Move the selected shapes, volumes, feeds, vias and holes by one grid step

Key	Action
Shift+arrow keys (same)	Move them by ten grid steps
Esc	Deselect everything and clear the ruler (Geometry view); remove the data tips from the charts
Enter (in the Ladder field of the Matching tab)	Apply the matching network
Esc (in SmithMatch)	Cancel placing a part
Ctrl+N, Ctrl+O, Ctrl+S, Ctrl+P (in SmithMatch)	New, open, save or print a SmithMatch design

Mouse

View	Drag	Wheel	Double-click	Click
Geometry	Move the selected item, or a handle; with the middle button: move the view	Zoom	Fit the board	Select the item (opens its row); Shift+click adds or removes one; Ctrl+click puts it on the grid
Geometry in 3D or Realistic	Rotate	Zoom	Start view	—
Charts	Pan	Zoom	—	Data tip (Ctrl+click adds one); middle-click fits; right-click to save or copy
3D pattern, Current	Rotate	Zoom	Start view	Read the gain or current at that point
SmithMatch chart	—	Zoom	—	Fix the part being placed, or add a datapoint (with Mouse adds DP ticked)

Expressions

Operators	`+ - * / ^`, and `.* ./ .^` as in MATLAB

Constant	`pi`
Functions	`sin cos tan asin acos atan sqrt abs exp log round floor ceil min max`
Units	Millimetres everywhere. A number can carry a unit (10mil, 0.5in, 2mm, 1.5cm, 100um); Paste table converts mil, µm, cm, in and m

Files

File	What it is
`*.json`	A project
`*.smith`	A SmithMatch design (section 12)
`<project>_sim\`	The simulation folder of a project (section 7)
`openEMS.log`	The solver output of the last run, in the simulation folder
`s11.csv`, `antenna.s1p`, `pattern.csv`	Results of the last run, in the simulation folder

Contact

Licences, questions and bug reports: lwtech.org/contact, lw tech (Wattlet AS).