builder User's Manual

Version 1.0

September 2024

Brian Young

Contents

1 Introduction

builder is a companion tool for OpenParEM2D and OpenParEM3D for quickly setting up projects for some common transmission line and waveguide types. Using a keyword/value text file, a project can be specified that allows builder to generate all files needed to run a simulation with the exception that gmsh [1][2] must still be manually run to generate a mesh. The project control file runs as-is but must be customized for the simulation for basic parameters such as frequency.

The supported transmission line and waveguide types are listed in Table 1. Microstrip and stripline support an optional soldermask, while the coupled versions of these support symmetric and asymmetric layouts. Note that thin soldermasks can result in slow run times and/or difficulty in converging.

Table 1: Supported transmission line and waveguide types.

- Rectangular waveguide
- Microstrip
- Stripline
- Coupled microstrip
- Coupled stripline

Input text files follow the specification in Sec. 6, where three types of transmission line or waveguide are defined. Rectangular waveguide is supported by the type `RectangularWaveguide` configured as defined in Fig. 1. Microstrip and stripline are supported by the type `Strip` as defined in Fig. 2, while coupled microstrip and stripline use `CoupledStrip` defined in Fig. 3.

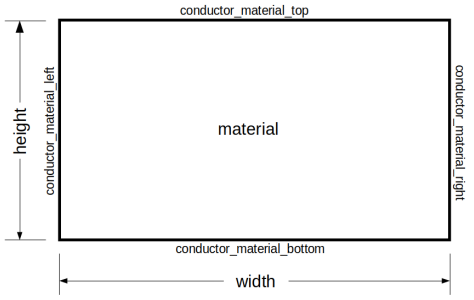

Figure 1: Structure and definitions for type `RectangularWaveguide`.

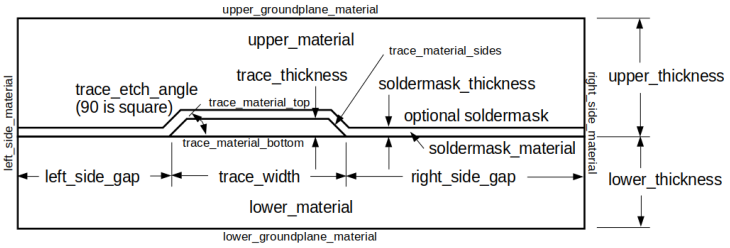

Figure 2: Structure and definitions for type `Strip` supporting microstrip and stripline.

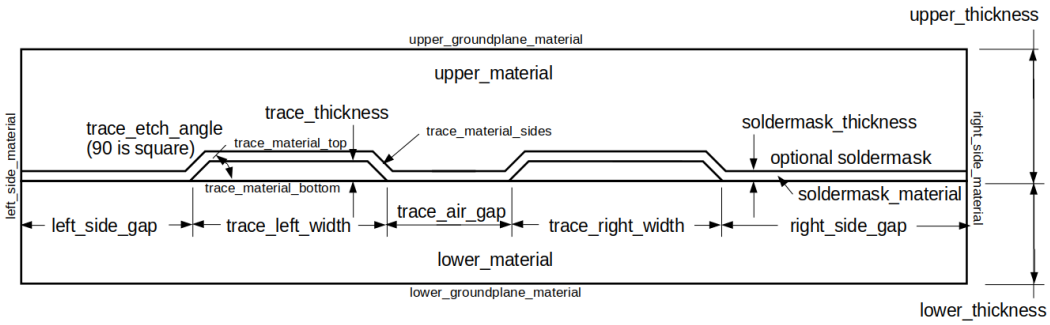

Figure 3: Structure and definitions for type `CoupledStrip` supporting microstrip and stripline.

2 Setup and Execution

Any text editor can be used to create the input file for builder following the specification in Sec. 6. One `Control/EndControl` block is required, then one or more supported transmission lines or waveguides can be defined. The transmission line or waveguide called out by the `build` keyword in the `Control/EndControl` block is the project that is constructed. The intention of allowing more than one transmission line or waveguide in the file is to support optimizations, where a transmission line or waveguide can be quickly copied, pasted, and tweaked. An example input file from the OpenParEM2D regression suite at `regression/Simonovich_stripline/builder/builder.txt` is

```
#builder 1.0

Control
   build=Simonovich_stripline
   check_limits=true
EndControl

Strip
   name=Simonovich_stripline
   use_symmetry=false
   left_side_gap=0.00486
   right_side_gap=0.00486
   upper_thickness=0.000301
   upper_material=prepreg
   lower_thickness=0.000305
   lower_material=core
   trace_thickness=0.00003175
   trace_width=0.000279
   trace_etch_angle=60
   default_conductor_material=PEC  // box left and right sides
   trace_material_bottom=copper_core
   trace_material_top=copper_prepreg
   trace_material_sides=copper_prepreg
   upper_groundplane_material=copper_prepreg
   lower_groundplane_material=copper_core
EndStrip
```

which builds files for solution with OpenParEM2D. The same input file is used in the OpenParEM3D regression suite at `regression/stripline/Simonovich_stripline/builder/builder.txt` with the simple addition of `length=0.001` to create a 3D geometry for use in OpenParEM3D.

For an input file called `builder.txt`, the command for running builder is simply

```
$ builder builder.txt
```

There are no command line options for execution. To get a short help file, run

```
$ builder -h
```

or

```
$ builder -help
```

3 Output Files

On successful execution, the output files of builder for OpenParEM2D are

- build_name.geo
- build_name_lines.txt or build_name_modes.txt
- build_name.proj

where `build_name` is taken from the `Control/EndControl` block. For OpenParEM3D, the output files are

- build_name.geo
- build_name_ports.txt
- build_name.proj

The project file is a template that must be customized. The remaining files are used without modification.

Note that the output files overwrite any previously existing versions. If a previous project file has been customized, care must be taken to not lose those customizations.

4 Meshing

The geo file can be directly imported into gmsh. Materials are already defined, so no further setup is required. After opening the file, it can be immediately meshed in 2D or 3D, then the mesh can be saved and gmsh can be exited.

5 OpenParEM2D and OpenParEM3D

After the project file has been customized and the geo file meshed, OpenParEM2D or OpenParEM3D can be run using the normal command line execution.

6 Specification

```
#builder 1.0

// enable multiple definitions in one file
// call out the one to build here
Control
   build=name
   check_limits={true|false}
EndControl

// The length parameter controls whether and OpenParEM2D or OpenParEM3D file set is
// produced.  Omitting or setting length=0 produces output for OpenParEM2D, while a
// positive length produces output for OpenParEM3D.

// The "include=name" option pulls in data from the named case to enable a change from
// one case to propagate to others. When used, any further data included in the block
// supersedes the included data. The included data must be in the same file.
```

```
RectangularWaveguide
   name=string
   [include=name]
   width=double
   height=double
   material=string                      // dielectric filling
   default_conductor_material=string    // Optional if all 4 of the materials below are specified.
                                        // Use "PEC" (without quotes) for no conductor losses.
   // material overrides useful for varying surface roughness
   // can also specify PEC or PMC
   [conductor_material_top=string]
   [conductor_material_bottom=string]
   [conductor_material_left=string]
   [conductor_material_right=string]
   [length=double]
EndRectangularWaveguide

// Use Strip to define microstrip and stripline.
Strip
   name=string
   [include=name]
   use_symmetry={true|false}            // faster solve time when true
                                        // user must divide the impedance result by 2
   default_conductor_material=string    // optional if all other materials are specified
   left_side_gap=double                 // measured from the farthest leftward trace dimension
   [left_side_material=string]          // a material, PEC, or PMC
   right_side_gap=double                // measured from the farthest rightward trace dimension
   [right_side_material=string]         // a material, PEC, or PMC
   upper_thickness=double               // measured from the top of the lower thickness
   upper_material=string
   [soldermask_thickness=double]        // thin soldermasks can cause simulation problems
   [soldermask_material=string]         // required for a non-zero soldermask thickness
   lower_thickness=double
   lower_material=string
   trace_thickness=double
   trace_width=double                   // width applies at the bottom next to the substrate
   trace_etch_angle=double              // degrees; 90 for vertical
   // material overrides useful for varying surface roughness or applying symmetry
   [trace_material_bottom=string]
   [trace_material_top=string]
   [trace_material_sides=string]
   [upper_groundplane_material=string]  // planes and sides can also specify PEC or PMC to create
                                        // symmetric coupled lines
   [lower_groundplane_material=string]
   [length=double]
EndStrip

// Use CoupledStrip to define symmetric and asymmetric microstrip and stripline pairs.
CoupledStrip
   name=string
   [include=name]
   [solution_impedance_calculation=string]   // modal or line; default=line
                                        // See the OpenParEM2D specification for details.
   default_conductor_material=string    // optional if all other materials are specified
   left_side_gap=double                 // measured from the farthest leftward trace dimension
   [left_side_material=string]          // a material, PEC, or PMC
   right_side_gap=double                // measured from the farthest rightward trace dimension
   [right_side_material=string]         // a material, PEC, or PMC
```

```
   upper_thickness=double                // measured from the top of the lower thickness
   upper_material=string
   [soldermask_thickness=double]         // thin soldermasks can cause simulation problems
   [soldermask_material=string]          // required for a non-zero soldermask thickness
   lower_thickness=double
   lower_material=string
   trace_left_width=double               // width applies at the bottom next to the substrate
   trace_right_width=double              // width applies at the bottom next to the substrate
                                         // Set to 0 to force trace_right_width=trace_left_width
                                         // for a symmetric pair.
   trace_thickness=double
   trace_air_gap=double                  // measured from from edge-to-edge at the bottom next to
                                         // the substrate
   trace_etch_angle=double               // degrees; 90 for vertical
   // material overrides useful for varying surface roughness or applying symmetry
   [trace_material_bottom=string]
   [trace_material_top=string]
   [trace_material_sides=string]
   [upper_groundplane_material=string]   // planes and sides can also specify PEC or PMC to create
                                         // symmetric coupled lines
   [lower_groundplane_material=string]
   [length=double]
EndCoupledStrip
```

References

[1] C. Geuzaine and J.-F. Remacle, "Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities," *International Journal for Numerical Methods in Engineering*, 79(11), pp. 1309-1331, 2009.

[2] `https:\\gmsh.info`