Quick Usage

Introduction

the ZIP file and extract to a location. Copy the *Example JSON file* below and save it as *test.json* using a simple text editor. To keep the things simple, save it at the same location as the executable.

To create your own JSON file, use our online visual box designer, or JSON Creator or refer to MIDILLI JSON Definition document to create your own from scratch.

Example JSON file

The following file generates all six sides for *MyPart*, placing one hole and one rectangle cutout on the *Top* side. The top size of the box is 5x5 cm and the components are placed at (x = 1, y = 1) and (x = 3, y = 3) respectively. The visible sizes of the components are the same as drill sizes.

```
{
□"Generic": {
□□"width": 5.0,
□□"length": 5.0
□},
□"MyPart_Top": {
□□"myRoundComponent#1": {
□□□"location_x": 1,
□□□"location_y": 1,
□□□"drill_diameter": 1,
□□□"call_name": "Hello",
            "visible_diameter": 1,
□□□"rotation": 0
□□},
□□"myRectangleComponent#1": {
□□□"location_x": 3,
```

```
□□□"location_y": 3,
□□□"drill_width": 2,
□□□"drill_length": 1,
□□□"call_name": "MIDILLI",
□□□"visible_width" : 2,
□□□"visible_length" : 1,
□□□"rotation": 0
□□}
□},
□"MyPart_Bottom": {

□},
□"MyPart_Back": {
□□
□},
□"MyPart_Front": {

□},
□"MyPart_Left": {

□},
□"MyPart_Right": {

□}
}
```

Depending on the command-line options, the other sides of the part will be created using different sizing. Read *Coordinate system and effect of sizing parameters* for more on different results of the arguments.

> **Note**: JSON syntax is very strict. Ensure that the last line before each '}' does **not** end with a comma, while all other lines **should**. Use an online validation tool to confirm correctness.

> **Note**: Use dot (.) if working with decimal numbers to separate the integer part.

Command line

Assuming that the JSON file is stored at the same location as the executable file, open a command window (Command Prompt, or PowerShell), then execute the following command:

```
.\GCodeGenerator.exe --json-file .\test.json --output-directory . --material-thickness 3 --cut-depth-per-pass 1 --box-height-3d 30
```

You will be provided with an output showing the current settings as following:

```
~~~~~ Settings ~~~~~
Outer sizing enabled: No
Finger joints: No
Finger size: 3 mm (default: material thickness)
Finger count: Automatic
Finger clearance: 0 mm
Components enabled: Yes
Side outline enabled: Yes
Screw diameter top in mm: 0
Screw diameter bottom in mm: 0
Screw head diameter in mm: 0
Screw head height in mm: 0
Screw extra margin in mm: 0
Recess depth in mm: 0
Feedrate on X-axis: 200
Feedrate on Y-axis: 200
Feedrate on Z-axis: 100
Bit shape: straight
Bit diameter in mm: 0
Material thickness in mm: 3
Depth per pass in mm: 1
Box height in 3D in mm: 30
Cut depth per pass in mm: 1
Simulation enabled: No
```

Each setting on the console output can be customized with command line arguments.

Note: Use dot (.) if working with decimal numbers to separate the integer part.

After completion, 6 files starting with *MyPart_* and ending with *.nc* will be visible under *test* directory. Opening *MyPart_Top.nc* in a program which can parse G-Codes should display a view similar to the following:



3D view of the test project in the Candle software.

Execution of the G-Code will move at 200 feedrate and drill 1 millimeter (mm) at each pass at 100 feedrate to open the holes for the components of a material with the thickness of 3 mm. At the end, it will cut along the outline of the *Top* to separate it from the larger material. For more customization, read the *Detailed usage* section.

Hint

Tip: If you want the overall size to be exactly as you specify in the JSON file, use the --outer-sizing option. If you want the usable inner area to be the same as you define, leave it as default.

Revision #6
Created 2025-11-07 10:19:53 UTC by MIDILLI
Updated 2026-08-14 07:37:09 UTC by MIDILLI