# pyRouterJig 0.1.0


pyrouterjig@lowrielodge.org

## 1 Overview

pyRouterJig is a woodworking layout tool for creating box and dovetail joints. It has the following features:

1. Creates templates for the Incra LS Positioner fence. In fact, this was the main motivation for writing pyRouterJig.

2. Support for both English and metric unit systems. Currently, only English has been tested, in part because my LS Positioner is the English version. I would be happy to test the metric version if someone would like to send me a metric LS Positioner.

3. May be customized for any finger (box joint) or pin / tail (dovetail) spacing pattern. Currently, two patterns are included ("Equally spaced" and "Variable spaced"), which will be described below.

4. pyRouterJig is licensed software, but you're free to use and modify it. See Sec. 6 for the license details. I really appreciate the Internet woodworking community. I have learned a lot from the tips, plans, videos, *etc.* that numerous woodworkers, who are much more talented woodworkers than I, have contributed. Maybe this tool can help them. I also hope that by making pyRouterJig freely available, including the source code, others that are more artistic than I can devise extensions and improvements to pyRouterJig. By making this software freely available, I am not implying that others should do the same for their hard work. Woodworking is my hobby, not my career. But because pyRouterJig is free, as the saying goes, "you get what you pay for." I will consider implementing changes and feature requests you might have, but in the

end, if you desperately want to change the code, learn Python and make the changes yourself! I certainly welcome other Python developers to help improve pyRouterJig.

# 2 Installation

To develop the code for this project, all I have access to is a Mac, and in particular, OSX 10.9.2.[1] pyRouterJig depends upon the following Python packages, which must be installed in order to run pyRouterJig:

1. Python. Python is installed by default on the Mac, but I use a slighly different version, as I will discuss below.

2. Mathplotlib. This package is used for drawing the joints and Incra template.

3. wxPython. This package is used as the graphical user interface (GUI).

I install all of these packages using Anaconda, which is also available for Windows and Linux. I highly recommend Anaconda, as the packages above may have other dependencies that Anaconda also takes care of installing.

# 3 Intervals

pyRouterJig does all of its computations in terms of what pyRouterJig refers to as "intervals." By default,

$$1 \text{ interval} = \begin{cases} 1/32" & \text{for Enligh units,} \\ 1 \text{ mm} & \text{for metric.} \end{cases}$$

An interval is the resolution of pyRouterJig. All dimensions used by pyRouterJig, such as the router-bit width, are rounded to the nearest number of intervals. The reason for the default choices above is that these are the resolutions of the respective Incra LS Positioner fence. By using intervals, we ensure that it's possible to position the fence at the exact location desired. More generally, using intervals (or "integer arithmetic") means that pyRouterJig does not need to be worried about floating-point errors.

[1] I'm looking for volunteers to test on other platforms, such as Windows and Linux.

# 4 Input Parameters

Throughout the documentation of pyRouterJig, we refer to a "finger" not only as the traditional finger of a box joint, but also generically to refer to a pin or tail of a dovetail joint.

pyRouterJig has the following input parameters, for any finger spacing algorithm:

- **Board Width [inches or mm]:** The width of the board for the joint.
- **Bit Width [inches or mm]:** The maximum cutting width of the router bit.
- **Bit Depth [inches or mm]:** The cutting depth of the router bit.
- **Bit Angle [degrees]:** The angle of the router bit. Zero indicates a straight bit (box joint). Fractional values must be input as a floating-point number, such as "7.5".

These parameters are changed by entering text for the dimension in their respective text boxes. In order for the figure to update using the new value (or values), either the "Enter" key must be pressed within a text box, or the "Draw" button must be pressed.

The "inches" length may be specified as either a fraction or decimal. For example, the following are equivalent:

- 7 1/2
- 7.5
- 7 1 / 2
- 7 1 /2
- 7 1/ 2

There are currently two finger spacing algorithms:

- **Equal:** In this case, the fingers are equally spaced. There are three inputs that affect this algorithm:
    1. **B-spacing:** This slider allows you to specify additional spacing between the Board-B fingers.

2. **Width:** This slider allows you to specify additional width added to both Board-A and Board-B fingers.
3. **Centered:** This input is only available for straight bits (bit angle = 0). If this box is checked, a finger is always centered on the board. Otherwise, a full finger is started on the left edge, which will result in a centerd finger only if the finger width divides into the board width an odd number of times.

- **Variable:** In this case a large finger is centered on the board, and the fingers decrease in size proportional to the distance to the center. There is one input that affects this algorithm:

    1. **Fingers:** This slider allows you to specify the number of fingers. At its minimum value, the width of the center finger is maximized. At its maximum value, the width of the center finger is minimized, and the result is the same as equally-spaced with, zero "B-spacing", zero "Width", and the "Centered" option checked.

# 5 Improvements Needed

I am looking for help to make the improvements outlined in this section. I will certainly give credit to those who help make improvements.

## 5.1 Windows and Linux Support

If you can help test and improve pyRouterJig on Windows or Linux platforms, please contact me. Ideally, you know Python and can send me proposed patches (or pull requests on Github).

## 5.2 Known Bugs

1. The first use of Variable Spacing puts its slider (for number of fingers) in the upper left-hand corner. Resizing the window fixes the slider placement.

2. Autosizing after changing the board width is broken. Again, resizing the window should fix this issue, and it appears related to the previous bug.

3. I can't add tic labels to the sliders. This may be an issue just on the Mac. I actually can add the tic labels, it's just that they don't "hide" properly.

### 5.3 Features

I'm working on the following features, but would appreciate help:

1. Enable use of arrow keys for sliders.
2. More inline help, such as define wxPython "tooltips" for each input value. These are windows that pop up when you put the mouse over a value.
3. Define the option for "fold-over templates" that are appropriate for hand-cut joints. This would be an alternative to the Incra template.
4. More friendly error messages and handling.
5. Get rid of the dependence on matplotlib and use wxPython drawing directly. The biggest issue I've had is printing. For example, with my anaconda installation, wxPython's demo/PrintFramework.py segfaults when I try to print.
6. More spacing options.
7. An interactive GUI to allow manual adjustment of fingers positions and widths.
8. Consider relaxing the requirement that board and bit dimensions be exact multiples of intervals.

## 6 Software License and Disclaimer

pyRouterJig is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with pyRouterJig, in the file named COPYING. If not, see `http://www.gnu.org/licenses/`.

Because of the well-known dangers of using woodworking tools, please also read the following: The authors of pyRouterJig are not responsible for any injury, death, or financial loss that could conceivably be caused by using pyRouterJig. As a user of pyRouterJig, only you are responsible for using the output of pyRouterJig safely and responsibly. Woodworking is very dangerous, and in particular, using a router or other tools to cut joints is dangerous. You can loose your eyesight, fingers, and suffer other serious, even fatal, injuries. The templates and patterns for joints that pyRouterJig generate are simply suggestions for joints. There is no implication that the joints can be safely cut with your tools (even if you have a Festool router). You are solely responsible for operating your tools in a safe manner. If you feel pyRouterJig could conceivably encourage you to operate your tools in an unsafe manner, don't use pyRouterJig. pyRouterJig does no stress or strength analysis of the joints it generates (although we would happily accept code changes that would do such analysis), and there is no implication of any particular joint's strength. The joint may fail, even if you glue it up properly. pyRouterJig is not responsible for joint failures. There is no guarantee that pyRouterJig is accurate or even generates a feasible joint. pyRouterJig will likely generate a joint pattern that does not fit and destroy your otherwise perfect woodworking project. In summary, if you believe, or you believe that your survivors might believe, that it is even remotely possible that pyRouterJig could cause you and your family serious bodily injury or death, break your tools, or ruin your woodworking project, do not use pyRouterJig. You have been warned.