



Customizing the Ergodox Infinity LCD Logo

09 FEBRUARY 2016 on keyboards

The Ergodox Infinity is a highly customizable keyboard device. One of its key features allows you to tailor multiple keyboard "layers" for use with various applications.

When switching layers, an LCD on each half of the keyboard shows which layers are currently active, ranging from the default layer to any number of stackable layers.

The default firmware that ships with the Ergodox Infinity displays the input.club logo (I:C) when the default layer is active. This tutorial shows you how you can customize that logo.

These instructions are applicable to the Mac OSX operating system. While many of the steps will also apply to building the firmware in Linux, I have not tested them. An

intermediate understanding of the command-line and UNIX-ish environments is also assumed.

Prerequisites

Hardware

You will need an assembled and functioning Ergodox Infinity keyboard and an additional keyboard to type in commands while you are flashing your Ergodox with custom firmware (laptops have this additional keyboard built in).

Firmware source code

You will need to obtain a copy of the firmware source code. This is available on Github at [kiibohd/controller](https://github.com/kiibohd/controller). Download it by cloning the Git repository or by [downloading](#) and unzipping a copy of the archive.

Toolchain

I downloaded and installed the OSX toolchain for building the firmware by following [the instructions here](#). I highly recommend using [Homebrew](#) instead of MacPorts to install the toolchain.

To test that the flashing utility was installed correctly, run the following:

```
> sudo dfu-util --list
```

TK You should see your Ergodox Infinity listed in the output.

Python dependencies

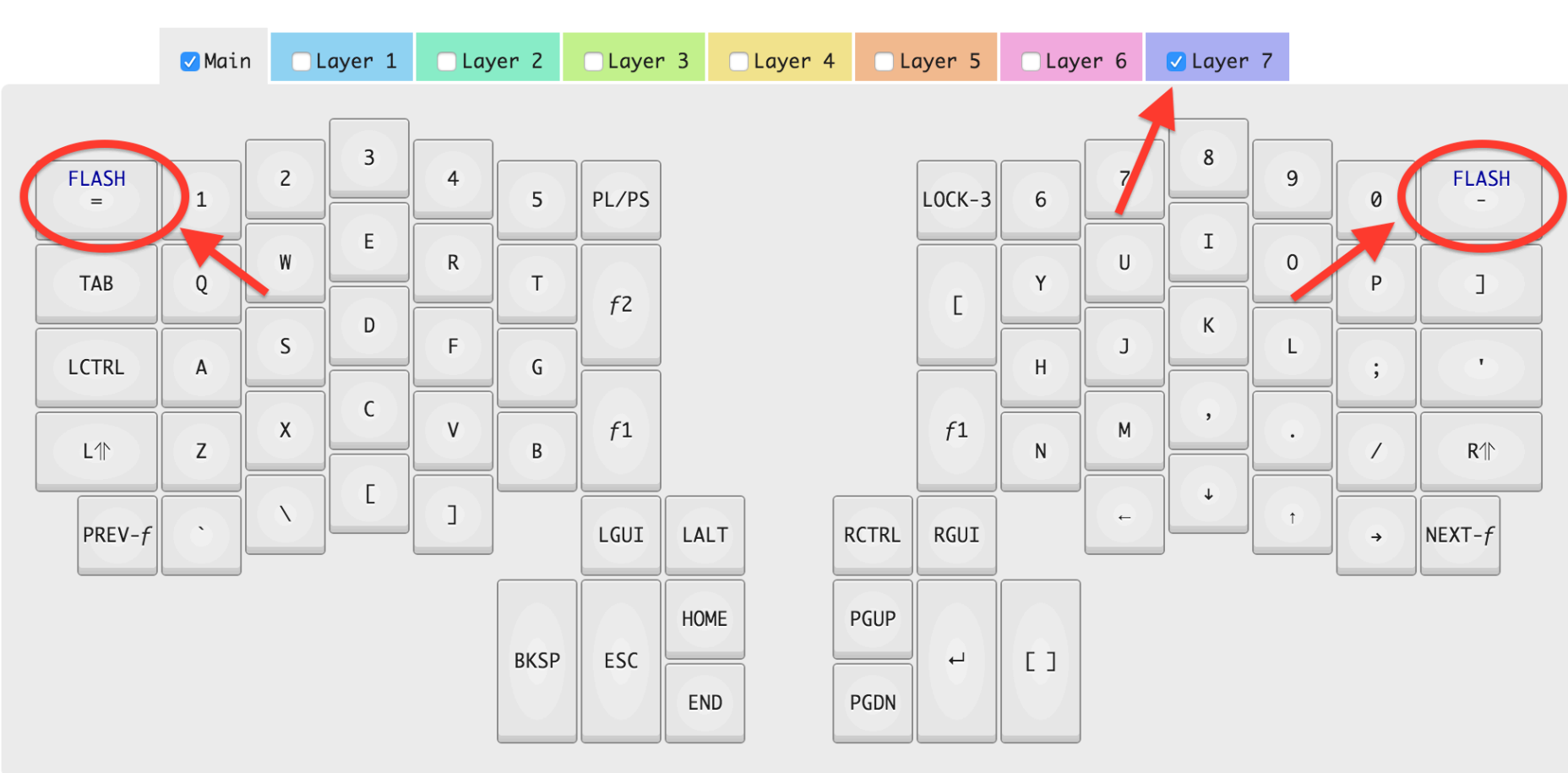
You will also need to install the Python Imaging Library (PIL). The easiest way to install this is via pip, the Python package manager, which is available via Homebrew:

```
> brew install pip  
> pip install PIL
```

Keyboard layout

Use the Configurator to build a keyboard layout. I highly recommend you have a `FLASH` command on each half of your Ergodox, this will allow you to enter the flash-mode without requiring you to press the hardware flash button.

In my layout, I configured the upper-left key (the plus key) and the upper-right key (the underscore key) mapped to the `FLASH` command on layer 7. I also mapped the lower-left and lower-right keys to the `PREV-f` and `NEXT-f` commands, respectively. This allows me to access to layer 7 even if I only have one half of the keyboard plugged in.



(You can use my layout, which includes mappings for the FLASH commands, as a starting point. [Open a browser to the Configurator](#), click on the "import map" button and paste [the contents of this JSON file](#).)

After you have created a layout, download the firmware pre-built with your layout and unzip it into a temporary location.

Logo creation

You can use any program to create a logo as long as you can save or export the logo image as a BMP file.

Here's an example of one I made using [Piskel](#):



Be sure your logo fits within the 128 X 32 pixel dimensions of

the LCD display and has a black foreground on a white background.

Included in the kiibohd source is a utility called [bitmap2Struct.py](#), which takes a BMP file and returns a byte-array representation of your logo.

This byte-array will be used to overwrite a setting in default Ergodox Infinity KLL files.

Cut and paste the byte-array into a temporary file.

Below is an example of the output from `bitmap2Struct.py`. The byte-array in this example is on lines 71—76.

```
1  +-----+
2  |
3  |                                     *****
4  |                                     *****
5  |                                     ****          *****
6  |                                     ****          **
7  |                                     ***           **
8  |                                     **      ***** *
9  |                                     **      ***** *
10 |                                     ***  **      **   **
11 |                                     **   **      **   **
12 |                                     **   **      **   **
13 |                                     **   **      ***   **
14 |                                     **      ***** *
15 |                                     **      ***** ** *
16 |                                     **           **   **
17 |                                     ***           **
18 |                                     ***           **
19 |                                     **  **      *****
20 |                                     **  **      *****
21 |                                     **           **
22 |                                     **  **
23 |                                     *** **  ** ** ** **
24 |                                     **  **  ** ** ** **
25 |                                     ** **  ** ** ***
```

```
26 |                                     ** ** **      **
27 |                                     ** ** **      **
28 |                                     ** **      ** **
29 |                                     **      **      **
30 |                                     ** ** ** ** ** ** ** **
31 |                                     ** ** ** **
32 |                                     **      **
33 |
34 |-----+-----
35 |
36 |-----+-----
37 |
38 |                                     ****
39 |                                     ****
40 |                                     ****      ****
41 |                                     ****      **
42 |                                     ***      **
43 |                                     **      ****
44 |                                     **      ****
45 |                                     *** **      **
46 |                                     ** **      **
47 |                                     ** **      **
48 |                                     ** **      ***
49 |                                     **      ****
50 |                                     **      **** **
51 |                                     **      **      **
52 |                                     ***      **
53 |                                     ***      **
54 |                                     ** **      ****
55 |                                     ** **      ****
56 |                                     **      **
57 |                                     ** **
58 |                                     *** ** ** ** ** **
59 |                                     ** ** ** ** ** ** ** **
60 |                                     ** ** ** ** **
61 |                                     ** ** **      **
62 |                                     ** ** **      **
63 |                                     ** **      **      **
64 |                                     **      **      **
65 |                                     ** ** ** ** **
66 |                                     ** ** **
67 |                                     **      **
68 |
69 |-----+-----
70 |
71 | uint8_t array[] = {
72 | 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
73 | 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
74 | 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
```

76

} ;

[view raw](#)

42

[illegible]

```
43 |                                     **      *
44 |                                     **      *
45 |                                     *** **      *
46 |                                     **      *
47 |                                     **      *
48 |                                     **      *
49 |                                     **      *
50 |                                     **      *
51 |                                     **      *
52 |                                     ***      *
53 |                                     ****      *
54 |                                     ** **      *
55 |                                     ** **      *
56 |                                     **      *
57 |                                     ** **
58 |                                     *** ** ** ** **
59 |                                     ** ** ** **
60 |                                     ** ** ** **
61 |                                     ** ** **
62 |                                     ** ** **
63 |                                     ** ** **
64 |                                     ** ** **
65 |                                     ** ** **
66 |                                     ** ** **
67 |                                     ** ** **
68 |
69 |-----+
70
71 uint8_t array[] = {
72 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
73 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
74 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
75 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
76 };
```

gistfile1.txt hosted with ❤ by GitHub

view raw

Build parameters

Next we need to create the build directories. We will need to take the KLL files from the ZIP file we downloaded from the configurator earlier. These files describe our custom key layouts.


```
> cd kiibohd-controller
> cp -r Scan/MDErgo1 Scan/MDErgo1-CustomLogo
> cp -r Keyboards/ergodox.sh Keyboards/ergodox-
custom.sh
> mkdir Keyboards/ICED-L Keyboards/ICED-R
```

Into each of the `ICED-L/` and `ICED-R/` directories, copy all of the `.k11` files from the downloaded firmware. Each of these is one layer of the keyboard layout.

```
> cp <tempdir>/*.k11 Keyboards/ICED-L
> cp <tempdir>/*.k11 Keyboards/ICED-R
```

Where `tempdir` is the location of the unzipped firmware from the "**Keyboard layout**" section above.

Edit `Keyboards/ergodox-custom.sh`:

- Change the `scanModule` value from `MDErgo1` to `MDErgo1-CustomLogo`.
- Edit `Scan/MDErgo1-CustomLogo/default.k11`. Set `DefaultLogo` to the byte-array saved from the "**Logo creation**" step described above.
- For each keyboard layer, set it as a `DefaultMap` or `PartialMap`, omitting the `.k11` extension.

The `DefaultMap` setting defines the layout for the default layer (or layer 0). Each `PartialMap` setting defines the layout for each additional layer, starting at layer 1:

For example, my layout has keys defined on layers 0 (default) to 7, so the KLL files from the firmware created by the configurator number from `MDErgo1-Default-0.kll` to `-7.kll`:

```
# define the default/layer 0 layout
DefaultMap="MDErgo1-Default-0 lcdFuncMap"

# define layers 1--7, where 4,5,6 are empty layers.
PartialMaps[1]="MDErgo1-Default-1 lcdFuncMap"
PartialMaps[2]="MDErgo1-Default-2 lcdFuncMap"
PartialMaps[3]="MDErgo1-Default-3 lcdFuncMap"
PartialMaps[4]="lcdFuncMap"
PartialMaps[5]="lcdFuncMap"
PartialMaps[6]="lcdFuncMap"
PartialMaps[7]="MDErgo1-Default-7 lcdFuncMap"
```

- Run the build script: `bash Keyboards/ergodox-custom.sh`.

This will produce a file named `kiibohd.dfu.bin` in each of the `Keyboards/ICED-L` and `Keyboards/ICED-R`. These are the firmware files you will need to flash to each half of your Ergodox.

Flashing firmware

To flash the firmware, you will need to put one half of the keyboard into flash mode. Plug in **just the left-half of the Ergodox** and (if you had followed my recommendation above) press the `PREV-f` key once to get onto Layer-7 and press the upper-left hand corner key to open flash mode. The LCD screen should be red in this mode.

Now, using a separate keyboard, enter the following command:

```
> sudo dfu-util --download Keyboards/ICED-  
L/kiibohd.dfu.bin
```

Upon successful firmware flash, you should see your custom logo displayed on the LCD screen.

To flash the right-hand side, unplug the left-hand side, plug in **just the right-hand side**, put it in flash mode, then run the following command:

```
> sudo dfu-util --download Keyboards/ICED-  
R/kiibohd.dfu.bin
```

Again, if successful, you should see your new logo on the right-hand side LCD screen.

Something wrong?

To recover, flash your Ergodox with the pre-built firmware files that you downloaded with your custom keyboard layout:

- Switch to the directory where you saved the files downloaded from the Configurator.
- Plug in just the left half of your Ergodox and run: `sudo dfu-util --download left-kiibohd.bin`.
- Unplug the left half of your Ergodox, plug in the right half and run: `sudo dfu-util --download right-kiibohd.bin`.

This should restore your Ergodox with a functioning firmware with the keyboard layout you have created.

Where can I purchase a kit?

Ergodox Infinity kits are periodically available on [Massdrop.com](https://massdrop.com). Please consider using [my invite code](#) when joining!



Ivan Tam

Read [more posts](#) by this author.

Share this post



YOU MIGHT ENJOY

Vim: Incrementing numbers.

You can increment a number in VIM by using the CTRL+a command. Decrementing a number with CTRL+x....