A definitive PicoFly install guide v.3

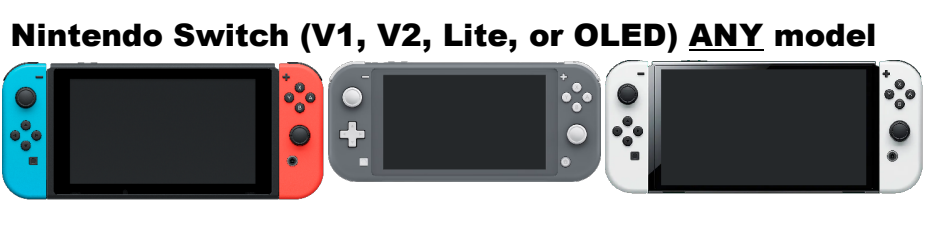
An attempt to create
an all-in-one install guide to answer questions
and help with photos

This guide will go over the basics of PicoFly setup, and wants to reiterate, this hardware mod can be installed on ANY switch (including unpatched V1's) running ANY firmware. It is CPU-glitch based, which loads a payload before Nintendo-signed software bootloader.

Disclaimer: this guide is written using many different sources of information. It is not my work, I am merely combining it here in this PDF to make it more accessible to the general public. I take no responsibility for anything that may happen to your device, good or otherwise. This guide is provided as-is, and you are responsible for anything that may happen.

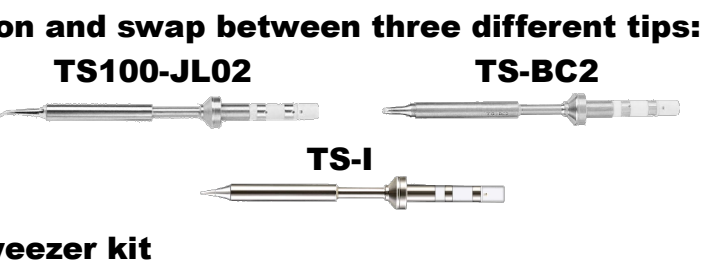
Recommended materials

1) Nintendo Switch (V1, V2, Lite, or OLED) ANY model



2) Soldering station

- Soldering iron that is preferably digitally controlled so you know what temp you solder at
- Solder (some recommend low-temp leaded fluxless solder to make some parts slightly easier)
- Flux (this is NOT optional. When soldering at these scales, flux is absolutely vital to making good solder joints) Applied with the tip of a wooden toothpick is how I usually hit small spots
- A damp sponge and/or brass wire sponge to keep your soldering tip clean. I use both.
- The tip you use can be vital depending on the locations of your soldering points. I use a ts-100 iron and swap between three different tips:



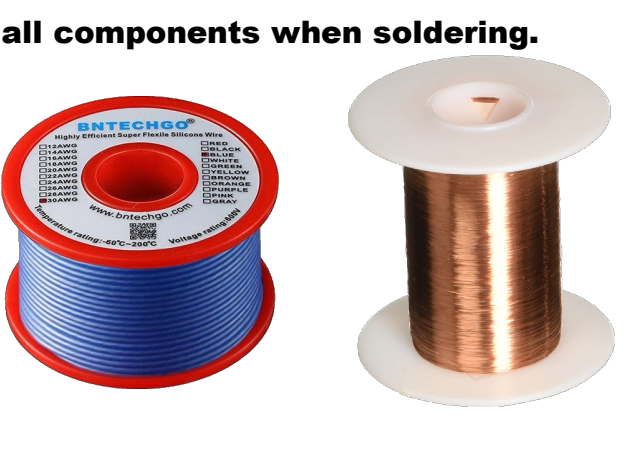
3) A tweezer kit

I know this sounds ridiculous, but trust me. Handling components this small REQUIRES fine-point tweezers. You may be able to SEE the components without a magnifier or scope, but you won't be able to manipulate and solder things like 0805 (or even smaller) resistors with bare fingers.

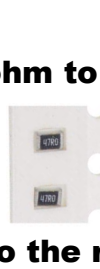
I ordered this kit off of Amazon
https://www.amazon.com/gp/product/B07S1DMKDX/

4) Wire

This would seem like a no-brainer, but the type of wire you use can ultimately make or break your install. I used a mix of 30awg Kynar and 40awg magnet wire. I have seen installs with all 30awg, and installs with all 40awg. Your luck may vary, but smaller wire tends to be a little more forgiving on small components when soldering.



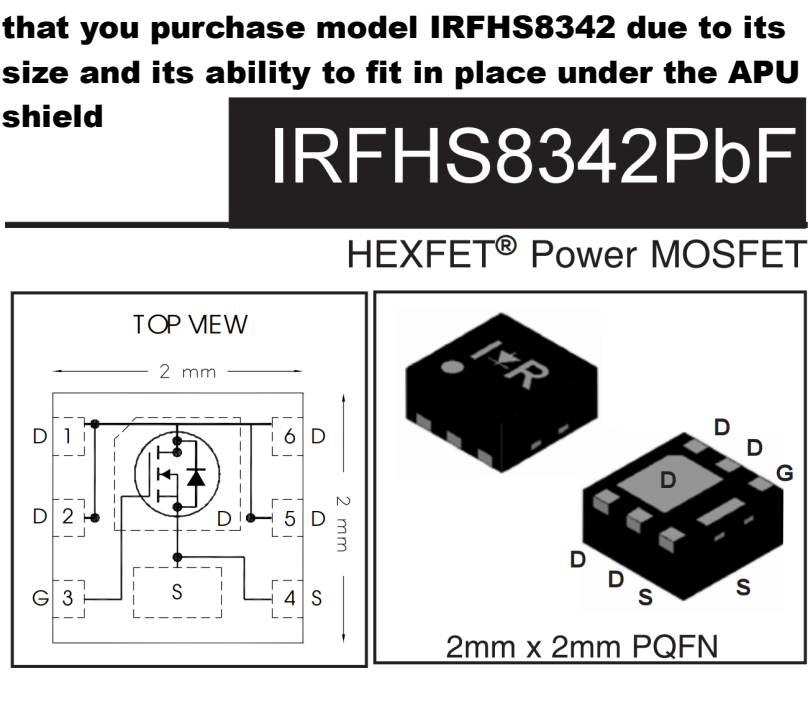
5) 47ohm to 50ohm 0805 Resistors



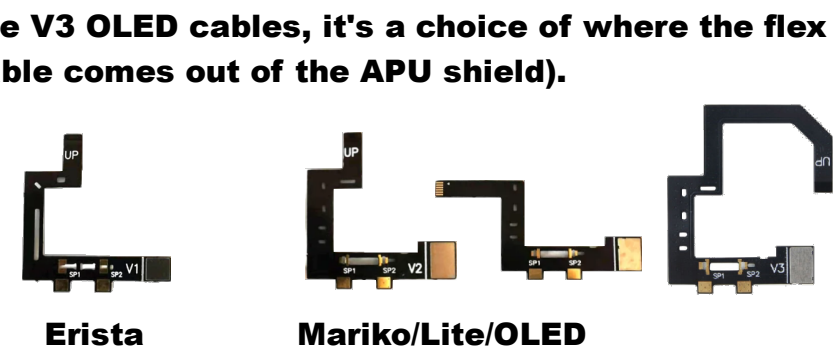
These resistors are soldered onto DAT0, CLK, and CMD on the PicoFly, then wires are soldered from the switch motherboard to the resistors. There are two types you can generally get away with using: 470 and 47R0
The 470 typing are 47 ohms +/-5% whereas 47R0 are 47 ohms +/-1%
Other sizes than 0805 will work, but 0805 seem to be easier to fit on the soldering pads of the RP2040-Zero

6) Mosfet(s) or an HWFLY flex cable appropriate to your specific console.

-If using mosfets by themselves, it is recommended that you purchase model IRFHS8342 due to its size and its ability to fit in place under the APU shield



-If using a flex cable (which has integrated mosfets) then order a V1 cable for Erista models, or a V2 cable for Mariko, Lite, or OLED models (You can also use V3 OLED cables, it's a choice of where the flex cable comes out of the APU shield).



Erista
Mariko/Lite/OLED

7) A magnifier of some kind

There are plenty of folks (myself included) that can get away with using a cellphone camera to zoom in and make sure joints aren't bridged and that good solder contact is being made. Others will say you absolutely need a scope to do this job so that you can see all the small components on a well-lit screen while working. It's up to what you're most comfortable with.

8) Kapton Tape

This seems to get overlooked in some cases, but the kapton tape helps to electrically isolate the picofly from accidentally shorting anything once it's installed and closed up.

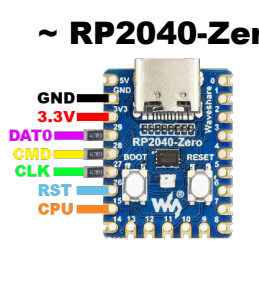
9) Thermal Paste

Another oft-overlooked item to improve your switch's lifespan after this mod. Good thermal paste being properly applied after cleaning all the old stuff off is vital to not overheating your console which leads to console shutoff into standby mode.

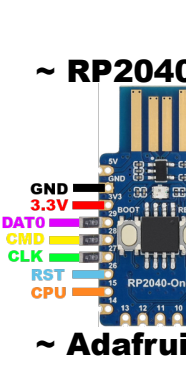
Last but not least
10) An RP2040 dev board

This is partially where the community starts to get divided, beyond what kind of switch you are modding.
A vast majority of the community has been using RP2040-Zero boards from Waveshare/Aliexpress, but thanks to Rehius, the firmware dev, several types of dev board are currently supported. I will attempt to list some of the most popularly used ones thus far. For this guide, I am focusing on The RP2040-Zero

~ RP2040-Zero



~ RP2040-One



~ Seeed XIAO-RP2040

~ Adafruit ItsyBitsy RP2040

See the end of the guide for diagrams of how to wire up the Seeed or the Adafruit

Optional, but recommended

11) UV-cure solder mask

-It cannot be understated how small these components are. Once your wires are faithfully in place, many would recommend a small amount of solder mask be applied to hold things in place, as well as help insulate connections from shorting

---

Preparing your PicoFly

Flashing your PicoFly, An Overview

At the heart of all the dev boards is an SoC (System on a Chip) known as the RP2040. It has its own built-in firmware flashing system that involves it presenting itself to your computer as a USB drive.

If it has never been programmed before, simply plugging it into your computer should bring up a window with two files in it. This is the firmware flashing window. If you drag and drop a ".uf2" file into this window, the RP2040 will read the file and attempt to flash it. This will cause the window to close, and the next time you plug the RP2040 in, it will not present itself as a USB drive to your PC.

To reprogram an already-programmed RP2040, there is typically a "boot" button that needs to be held down while being plugged into USB, but when you do this and the window shows up, your firmware file will not be displayed in the window since it IS NOT a storage device. The RP2040 just tricks your computer into treating it like a storage device. Some recommend that between flashes you use a "flash_nuke.uf2" provided by the Raspberry Pi Foundation. This basically just flashes all zeroes to the RP2040 firmware memory to ensure a clean flash of whatever you put on it next.

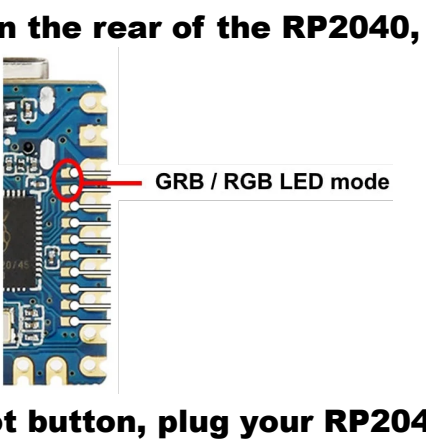
We want to flash our device with Rehius' most up-to-date PicoFly firmware, which can be located at:

https://github.com/Ansem-SoD/Picofly/tree/main/Firmwares

As of the current date (11-04-2023), "fw_2.64.uf2" is what needs to be flashed to your dev board.

With this file downloaded, and your RP2040-Zero plugged into your PC, showing the firmware flashing window, keep one eye on the LED of your RP2040-Zero. As you drop the .uf2 file into the window, your LED should flash as the window on your PC closes. This is to indicate firmware flashing success, but the color of this LED is what we are concerned with now.

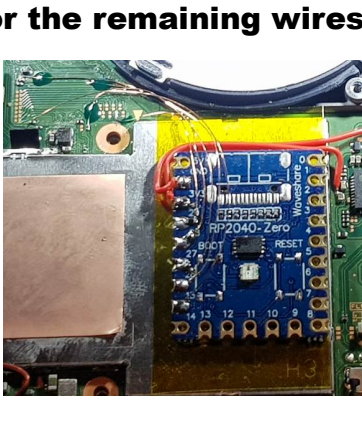
If the LED flashes Red once, GREAT, you don't need to do anything special, however, if you see a single Green flash, then you need to permanently bridge the RGB jumper with solder on the rear of the RP2040, seen below:



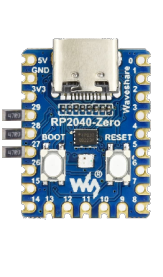
You can hold the boot button, plug your RP2040-Zero back into your computer, flash the .uf2 to the board again and look for a Red LED flash now.

At this point, with a firmware loaded on the board, we can start to prep it to go into the switch.

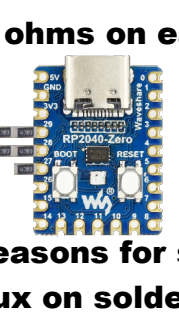
Conventionally, folks remove the USB-C port and the boot + reset buttons. Below is an example of a finished install with USB-C removed, buttons removed, resistors installed, and as you may notice, a mix of 30awg Kynar (for power supply to RP2040-Zero) and about 36awg magnet wire for the remaining wires.



During this step of removing ports, it is also recommended to add the three resistors to the side of your board. With the boot button removed, some people place their resistors to the inside of the board, rather than hanging from the edge. Both will functionally work.



If you have slow EMMC issues after putting everything together, one possible reason is that you need two additional resistors, one each on CMD and DAT0, like below, for a total of 94 ohms on each line:



There could be other reasons for slow EMMC as well though, like residual flux on solder points, or wires being too thin or too long on CLK, CMD, or DAT0.

On that same note, if you leave flux residue on RST points, your console could instantly reboot after startup, as the RST points are very sensitive.

Time to open up your console and get to work!

Modding your console

There are plenty of guides on Youtube on how to properly open your console, so I will merely reference those below:

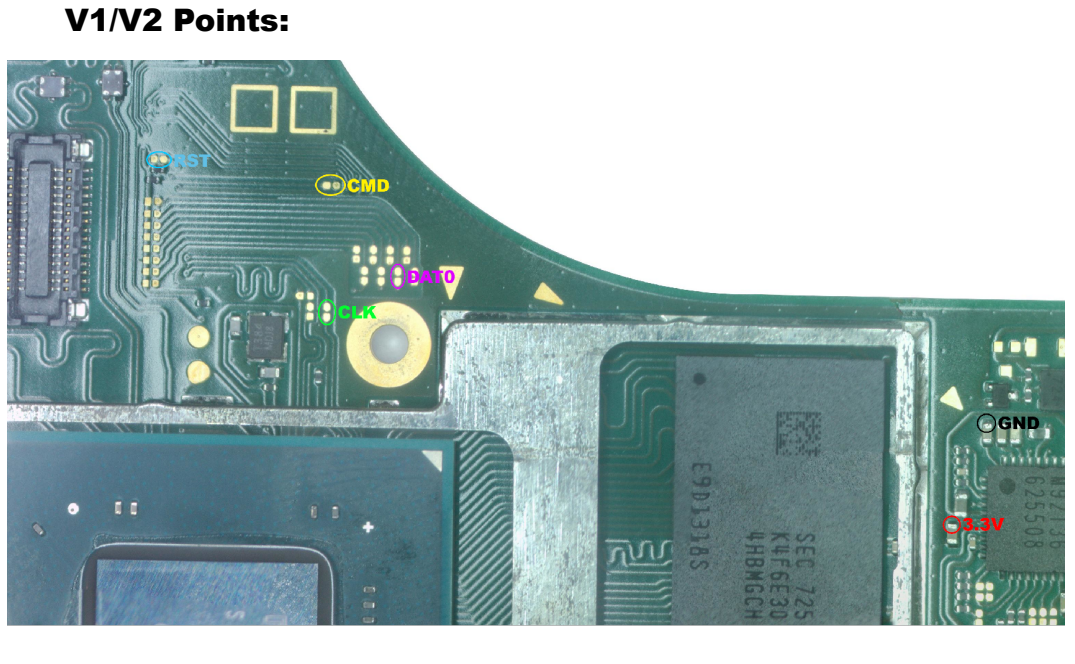
V1 and V2 disassembly:
https://www.youtube.com/watch?v=QCZ3-fYjpWo

Lite disassembly:
https://www.youtube.com/watch?v=GP1DHRs6V2Y

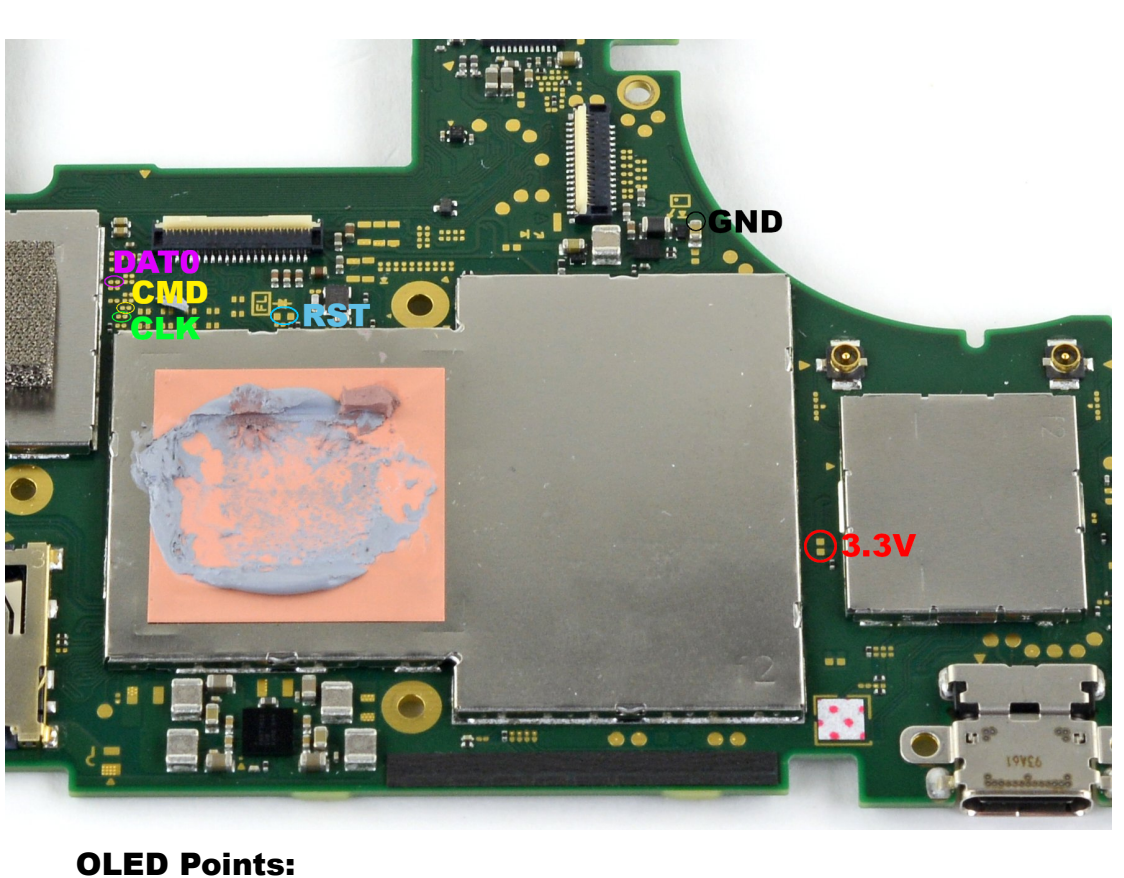
OLED disassembly:
https://youtu.be/4umniX8nX5E?t=40s

Once you're inside, you'll need to reference the following photos to solder your wires into the console:

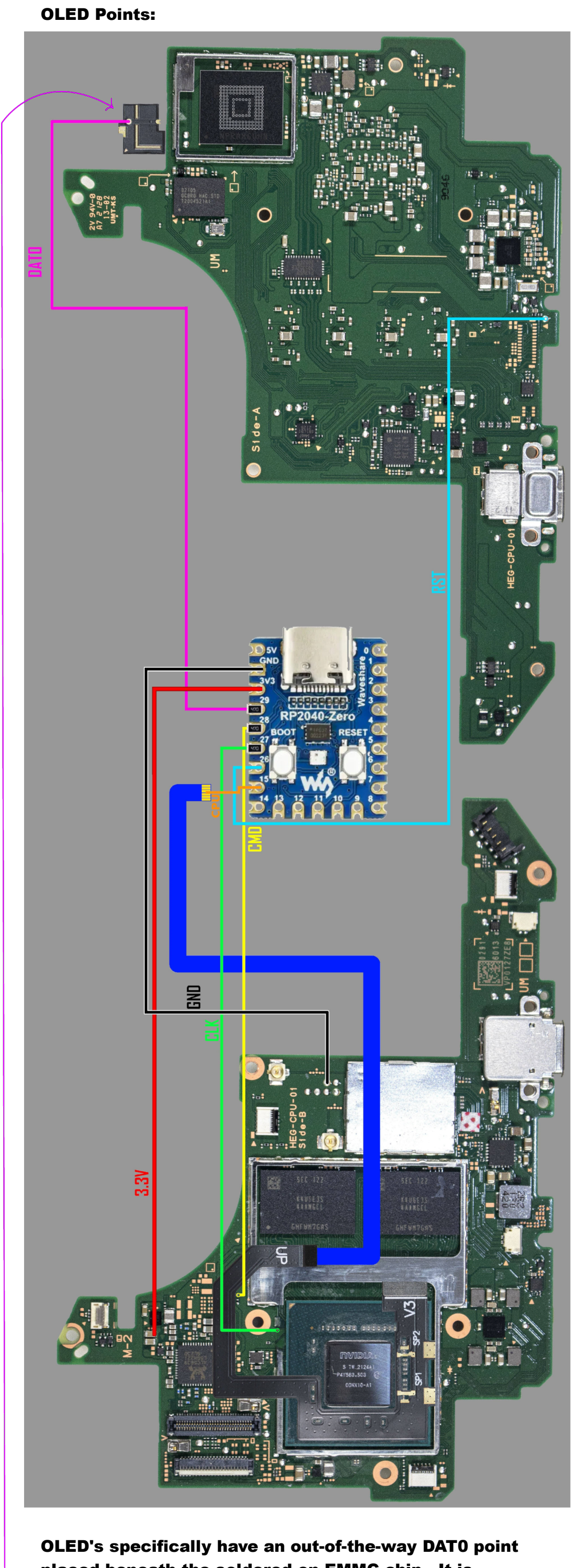
V1/V2 Points:



Lite Points:



OLED Points:



OLED's specifically have an out-of-the-way DAT0 point placed beneath the soldered-on EMMC chip. It is accessed by applying a DAT0 adapter like one of the below:

Follow Me

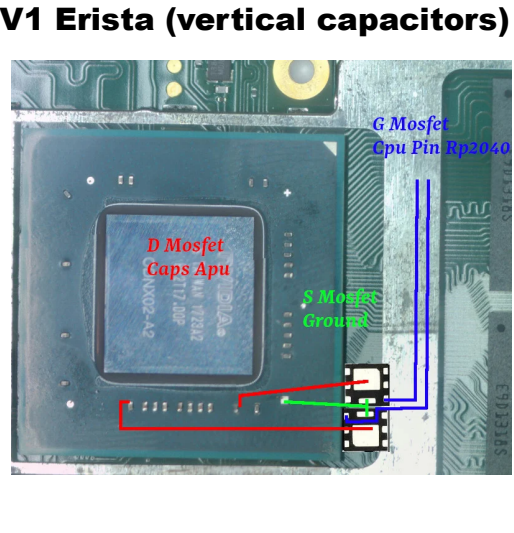
Use this link to watch youtuber Alien Retro Gaming place his adapter and wire it up:
https://www.youtube.com/watch?v=TrouatoHmBg&t=155s

The last part is the installation of mosfets or of the HWFLY cable. Functionally the install is the same for all models, the main difference is orientation of the vertical APU capacitors from the V1 model to the other 3 models that place the capacitors horizontally. Here is a section of a video with one of these cables being installed:

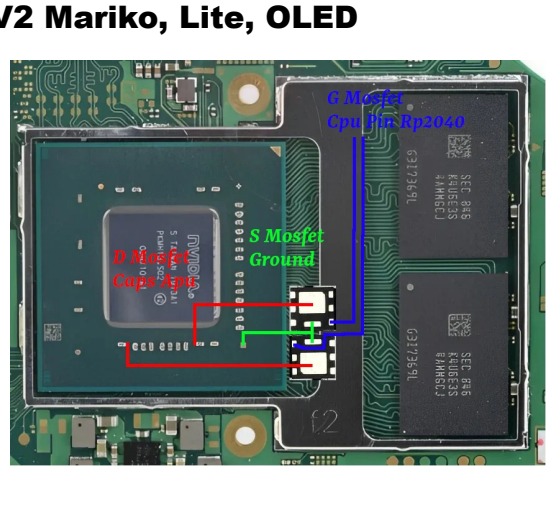
https://youtu.be/XG80TcbO5Uo?t=107

If you do not have access to a cable and are doing a bare mosfet install, here are some photos to reference of the clever ways people have been wiring theirs up:

- V1 Erista (vertical capacitors)

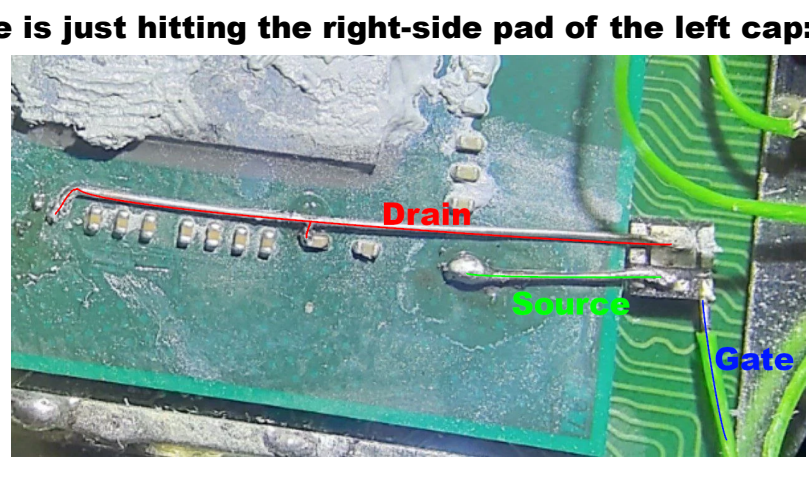


-V2 Mariko, Lite, OLED

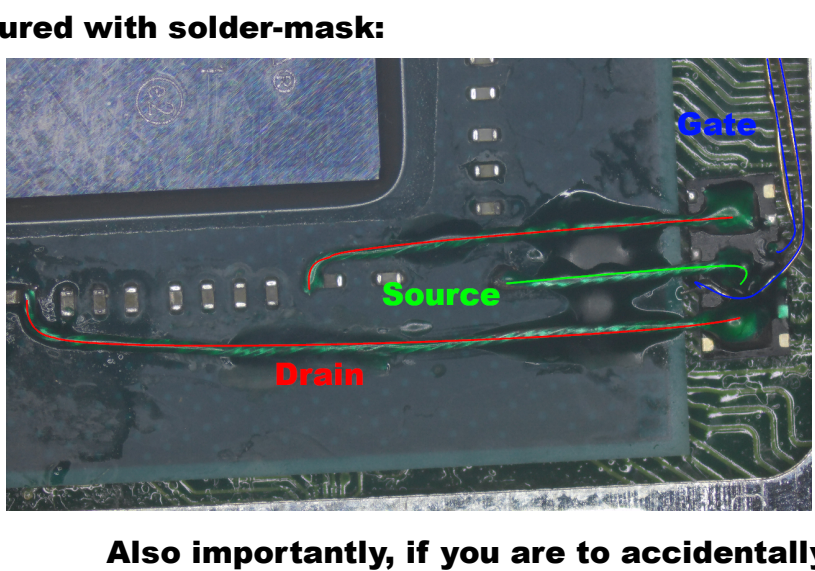


If it isn't so clear, here are a few photos of some mosfet installs:

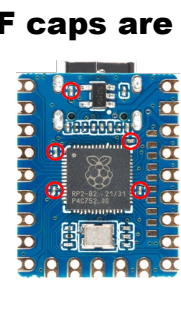
This is a single mosfet install, where the left capacitor of this V2 board has been removed, so the wire is just hitting the right-side pad of the left cap:



And here is a slightly more complex, but well-executed, install of double mosfets that have been secured with solder-mask:



Also importantly, if you are to accidentally remove one or more of these capacitors, you may be fine, but to replace them, you can rob a few off of a RP2040-Zero, without affecting the function of the PicoFly. These 100nF caps are located here:

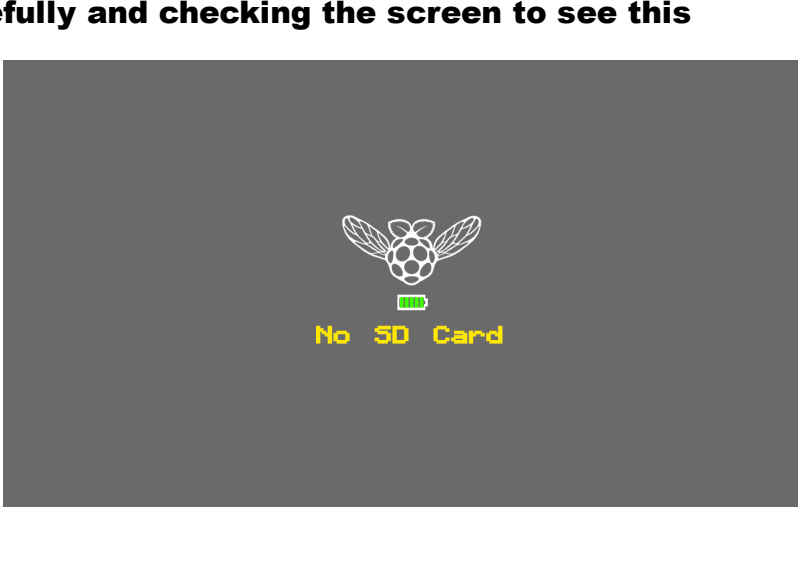


Once you have your mosfet(s) installed, make sure all your new points are electrically insulated from the APU shield as you put it back on.

Now simply solder all of your flying wire leads to their appropriate points on your RP2040.

Before attempting to boot, make sure you did not accidentally bridge any extra contacts, as this could ruin your console.

Once you're ready, push the power button on your console after plugging the battery back in. The PicoFly should flash blue, (on first boot, it should flash white after blue) then green to indicate success! Should you see this order of lights, try flipping your console over carefully and checking the screen to see this



If you're met with this screen, then that is as far as this guide takes you, as you have successfully modded your switch with PicoFly, congratulations! At this point you can follow any number of HATS pack guides to prepare an SD card for your device, but the gist is that it is looking for a "payload.bin" located in the root of the SD card. As long as all the supporting files for that payload.bin have been copied to the SD card along with it, then everything should boot fine.

On the other hand, if you didn't see Blue, White, then Green on your LED, and your console either boots to a black screen, or instead boots to OFW like normal, then look at the following LED code list to diagnose your potential issue:

- ○ Writing
- ● Glitching
- ● Success
- ● Bad CLK
- ● Bad CMD
- ● Bad D0
- ● Bad CPU pin

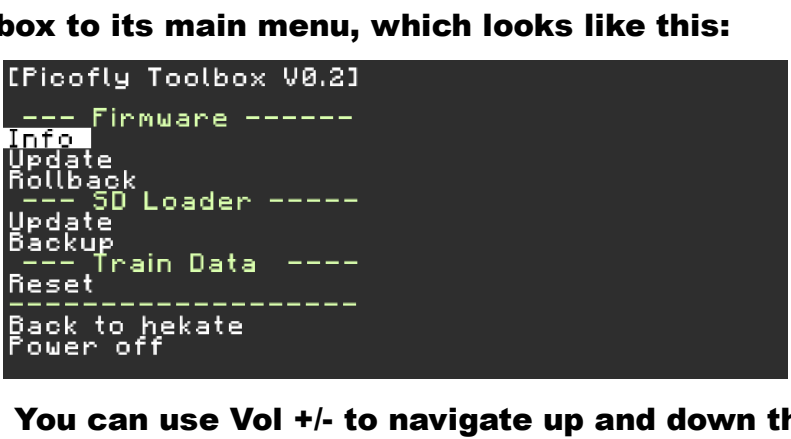
If you get Blue and then Cyan, and you have only installed a single mosfet, this could indicate that you need to install the second mosfet to get consistent glitching at the CPU level. Pink, Purple, or Orange are normally indicative of a poor solder joint somewhere, relating to whichever color is flashing.

Hopefully at this point you have managed to hack your console, and are looking forward to the following hours of configuring and tweaking your setup!

PicoFly Toolbox

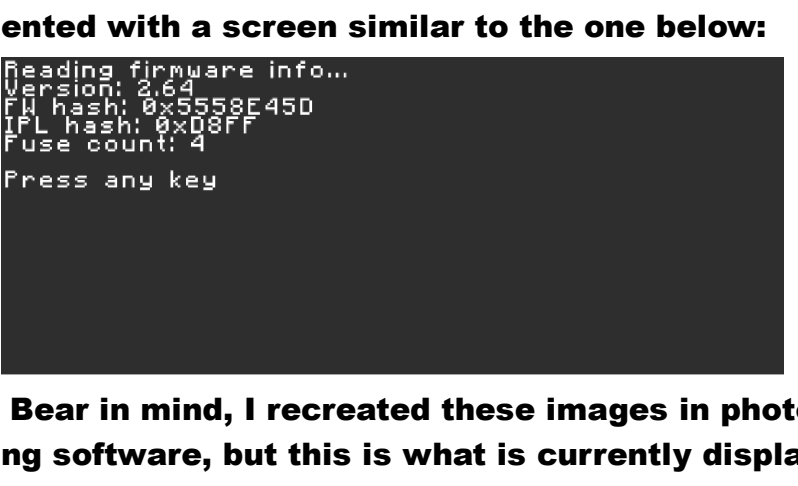
"picofly_toolbox_0.2.bin" (available on Github where the firmware is downloaded from) is the current payload that you can load through Hekate's payload launcher, to interface with your PicoFly. The caveat to this, is that you must be on at least firmware v2.6 to use the toolbox as a feature. This means, if you have previously loaded your PicoFly with an out-of-date firmware, like the frequently circulated "2.5 toshiba + unlock", you must flash at least 2.6 to your PicoFly via USB and a computer before you can interact with PicoFly using the toolbox.

Once you are on minimum v2.6, you can launch the toolbox to its main menu, which looks like this:



```
[Picofly Toolbox V0.2]
 --- Firmware ------
Info
Update
Rollback
 --- SD Loader -----
Update
Backup
 --- Train Data  ----
Reset
-------------------
Back to hekate
Power off
```

You can use Vol +/- to navigate up and down the menu, and then press the power button to select a menu option. If you hit power on the info option, you will be presented with a screen similar to the one below:



```
Reading firmware info...
Version: 2.64
FW hash: 0x5558E45D
IPL hash: 0xD8FF
Fuse count: 4

Press any key
```

Bear in mind, I recreated these images in photo editing software, but this is what is currently displayed on my switch. I have flashed several updates, so my fuse count is greater than 0, and I am currently on v2.64

Updating

To actually use the tool to update your picofly from V2.6+ to something higher, simply place the desired firmware's ".bin" update in the root of your SD card and name it "update.bin". Launch PicoFly Toolbox, then scroll down once to "Update" by hitting "Vol -" one time, then hit the power button once. The toolbox will now look for the "update.bin" where it is supposed to be, and will flash the firmware provided inside the file. You will be asked to reboot, and may experience slightly longer boot times for the first few boots after updating. This is normal, once the PicoFly re-trains, you will be back to fast boots.

Rolling Back

PicoFly stores the last firmware you flashed from, in the event of a failed flash, it can auto-rollback to whatever version you were flashing from. If in the event you have a successful flash to a higher version, but wish to intentionally rollback to your previously running version, from the main menu of PicoFly Toolbox, scroll down twice with the Vol - button being pressed twice, then select Rollback with the power button. Similar to an update, you will be asked to reboot.

SD Loader

To my understanding, SD Loader is the portion of Boot0 which is being modified to load unsigned SD payloads (such as our custom bootloader of choice, typically Hekate). The "No SD Card" screen is part of this SD Loader portion. To make a backup or restore this portion, just use the two options seen in their respective portion of the menu, but you really shouldn't have to mess with this unless directed to do so.

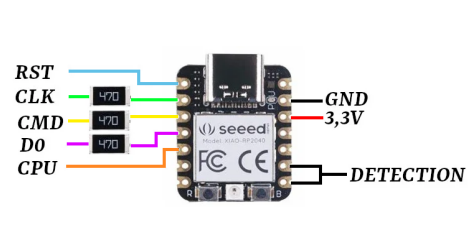
Training Data Reset

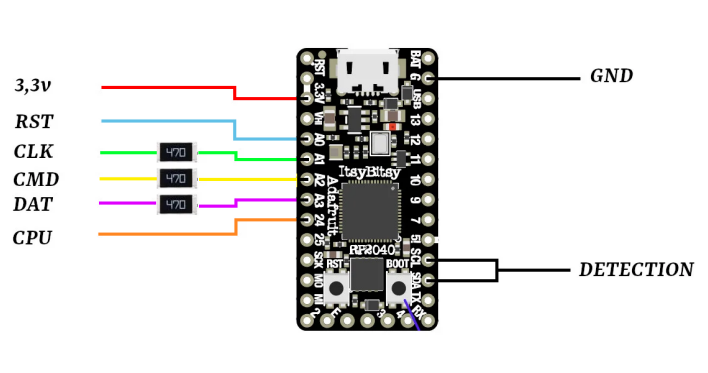
As the name might imply, this is to reset the training data of the PicoFly in the event your boot times have gone awry for some reason. This will clear your training data and allow your PicoFly to attempt relearning new glitch timings for your console. Largely this should be unnecessary, but it is here if needed.

This guide has been brought to you by Lightninjay, a member of the GBATEMP forums, and in no way implicates them for any harm that may come to you or your devices while following this guide.

As a courtesy, here are some pinouts of alternate boards that PicoFly should be compatible with, thanks to forum member, Dee87, colors added by myself

SEEED XIAO-RP2040



Adafruit ItsyBitsy RP2040