

PicoLX User Manual



If possible, please view this page in print layout, to ensure everything renders as intended.

Introduction.....	2
Getting Started.....	3
PIUIO firmware.....	3
Gamepad/Keyboard firmware.....	3
Firmware Flashing & First Boot.....	4
Common Issues.....	4
Customization.....	5
Keycaps.....	5
Switches.....	5
Lighting, etc. Configuration (PIUIO firmware).....	6
Adjusting LED colours.....	6
Recommended Defaults.....	6
USB Device Information.....	7
Recommended Defaults.....	7
Drivers.....	8
Windows.....	8
Linux.....	9
Troubleshooting.....	10
Schematic.....	12
Board Drawings.....	13

Introduction

The **PicoLX** is a 5-panel dance game controller, compatible with popular simulators like *Project OutFox*, *StepF2*, *StepP1/P1+*, *PumpSanity*, and more! It's also compatible with commercial 5-button rhythm games like *Pump it Up: RISE* (available on Steam) alongside the arcade *Pump it Up* titles, when running the appropriate firmware.

Whether you purchased your PicoLX directly from me (dj505), or assembled it yourself, this manual aims to provide a complete & thorough look at everything you need to know to customize, configure, maintain, and repair your controller.

Each controller features...

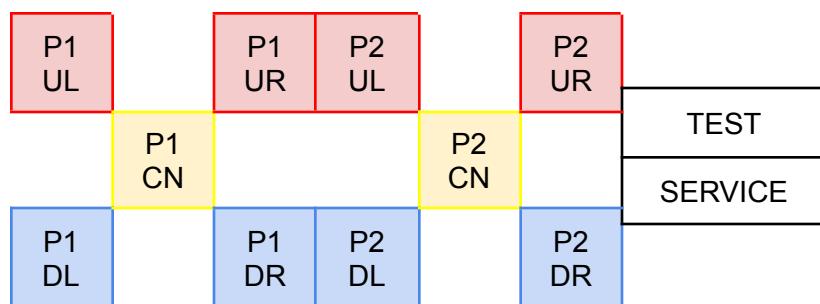
- A USB type C connector
- 10 hotswap sockets (Kailh, typically) to make customizing your switches a breeze
- 10 mechanical keyboard switches (Gateron Yellow, typically)
- 19 RGB LEDs for key lighting, underglow, and a fancy lit up logo
- A powerful Raspberry Pi RP2040 microcontroller under the hood

Getting Started

To get going with your brand new controller hardware, you'll first need to pick your preferred firmware. [Both firmwares are available here](#), inside of the "Firmwares.zip" file.

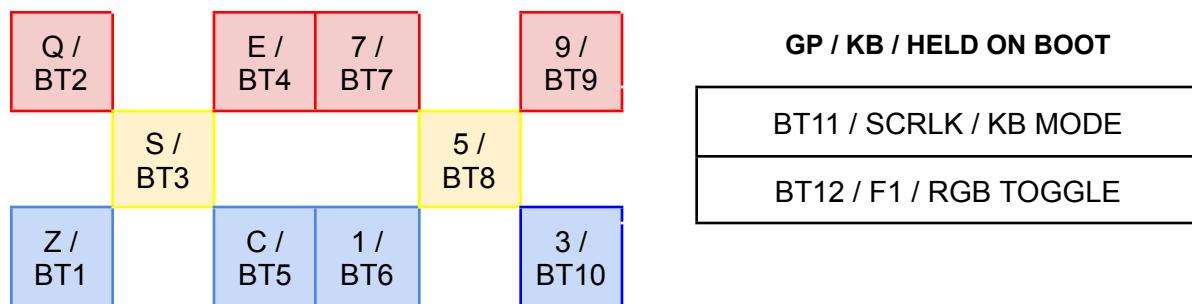
PIUIO firmware

The PIUIO (*Pump it Up* IO) firmware allows for native compatibility with the arcade *Pump it Up* titles. This is particularly handy for players who own a copy of the game and wish to play it at home, without having space for a full stage or two. In this mode, the controller's lighting is controlled by the game; the buttons are assigned red, yellow, and blue lights, and the 6 underglow LEDs act like cabinet lights. This mode can be used on Windows and Linux, with the appropriate drivers installed (See the "Drivers" section). In this configuration, the button layout is as follows:



Gamepad/Keyboard firmware

The gamepad (and keyboard!) firmware allows the controller to be used as an ordinary HID gamepad or keyboard, for use with ordinary PC and/or mobile software. In this mode, all the lights will be lit, and will cycle through the rainbow. In this configuration, the button layout is as follows:



Firmware Flashing & First Boot

On first boot, if your controller is freshly made and has not yet been flashed with any firmware, you can skip straight to the second step.

1. On the underside of your controller, you'll notice a small pin hole near the USB port. Simply poke a paper clip (or other small, thin tool) through until you feel it make contact with the button underneath. You should be able to feel it click. Hold the button down while plugging the controller's USB cable into your PC.
2. You should see a storage device appear, titled "**RPI-RP2**." Simply copy and paste the .uf2 file for your preferred firmware onto this disk.
3. Wait for the copy to finish; once it does, the controller will automatically reset itself, and the disk will disappear. **You may get an error popup during this process.** If that happens, you can safely ignore it, as the disk will only appear to disconnect after the controller has been successfully programmed.



Common Issues

- I can't see any RPI-RP2 disk!
 - Ensure your USB cable supports charge and sync. There are some cheap USB cables out there (even type C ones!) that only carry power, with no data lines. It might occasionally be necessary to try several cables.
- Some of the LEDs do not work/are acting up. What's going on?
 - The LEDs are all in one long chain, with the colour data flowing through each, one by one. It's likely that there's a "weak link in the chain," so to speak, which could be either the data input of the malfunctioning light, or the data output of the one before it. See the schematic towards the end of the manual to narrow down which LED may be acting up, and touch up the solder points for both.
- Some of the buttons aren't responding. What's going on?
 - The most common cause, I find, is bent legs on the switches. Occasionally, a slightly bent leg might not quite line up with the socket, and when the key is pushed into place, it might crumple and fail to make contact. Remove the offending keys and ensure the legs are straightened and at a 90 degree angle!
 - Otherwise, double check the soldering on the sockets. They take a lot of heat, and it's not difficult to end up with a cold joint. Don't hesitate to reflow the solder.

Customization

Since the PicoFX is built like a mechanical keyboard, you can customize your hardware in a very similar manner!

Keycaps

As far as keycaps go, you can use just about anything you want! The sky's the limit. I've seen some builds that match the keycap colours with the panels on arcade *Pump it Up* stages, and they look fantastic! They do tend to cost more, though, because you need to buy at least 3 sets of keycaps in different colours.

The stock keycaps on most units are plain white 1u DSA profile keycaps. I've found these to be the most comfortable, personally, and they're quite easy to source. I've also tried XDA profile caps, which feel quite nice, but they're a little too bulky for my liking. I avoided profiles like OEM or Cherry because the angle felt weird to play on. Pretty much all profiles will *fit* the controller, though! You don't have to worry about collisions or fitment problems.

Switches

Which switch you use is entirely up to you. The PicoLX uses **hotswap sockets** to mount the switches, so replacing them is dead simple. I would highly recommend using a **switch puller** (pictured) to avoid damaging your switches or hurting your fingers, but it could be done by hand if you really wanted to.



When installing switches...

- Remove and replace **one at a time**. The top plate and main board are held together by the switches themselves, so removing them all at once means unscrewing the plate!
- Make sure the legs are at a 90 degree angle and aren't bent! Attempting to insert a bent switch might mean poor (or no) contact, and probably a very quick stage break on your next song 😞
- Feel for the click! Unless you've gone through this process enough to wear out the top plate, the switch should snap satisfactorily into place and sit level.
- Leave the controller plugged in, and test each switch as you install it to ensure everything is working as expected!

Lighting, etc. Configuration (PIUIO firmware)

Configuring the controller's behavior, at the time of writing this, is not the simplest of tasks and requires modifying & rebuilding the controller's firmware. As it would be too much to include setting up a toolchain and development environment in a simple instruction manual, the following will assume that you already have a working environment in which to compile the firmware.

To begin, clone a copy of the [PIUIO firmware source code](#)¹.

Adjusting LED colours

LED colours for the PIUIO firmware are defined in [piui0_config.h \(line 11-31\)](#) as sets of RGB values, ranging from 0 to 255. Each line is commented to show which LED it corresponds to when the controller is viewed upright from the top.

By default, the 3 logo LEDs are set to **pure white**; adjustments to these values will not be reflected without commenting out [piui0_ws2812.c \(line 35-37\)](#) to disable the default colour cycling animation.

Regardless of colour, the same LEDs will be triggered by the same signals from the game software. If you want to change the order of these LEDs, see [piui0_ws2812.h \(line 16-33\)](#). These are listed in the physical order of the LEDs in the chain, with matching comments.

Recommended Defaults

The default LED colours are modelled after the lighting in an *FX* model *Pump it Up* cabinet, with buttons corresponding to the inputs on a stock set of stages, listed below for your convenience:

LEDs	Colour	Value	
Upper left/right backlights	Solid green	(0, 255, 0)	
Lower left/right backlights	Solid red	(255, 0, 0)	
Upper/Lower middle backlights	Solid blue	(0, 0, 255)	
P1 & P2 top row (upper diagonal) buttons	Solid red	(255, 0, 0)	
P1 & P2 middle row (center) button	Yellow	(255, 255, 0)	
P1 & P2 bottom row (lower diagonal) buttons	Solid blue	(0, 0, 255)	

¹ <https://github.com/dj505/piui0-picول>

USB Device Information

USB device information is all present in [usb_descriptors.h](#).

The manufacturer, product, serials, and vendor interface information are all listed in the array at [line 112](#), and should be fairly self explanatory. Changing these will change what shows up in the device manager/lsusb output/the little popup when you plug it into your PC.

Recommended Defaults

Descriptor	Value
Manufacturer	dj505
Product	PicoLX PIUIO
Serials	123456
Vendor Interface	PIUIO-pico

Do not change anything else in this file unless you know what you're doing; the USB PID/VID are configured so as to ensure that the game recognizes the controller as a valid IO device! Adjusting specific values, such as the device/vendor ID, may prevent the game (or third party drivers) from detecting the controller.

Drivers

This section applies to the PIUIO firmware specifically.

If you want to use the PIUIO firmware for use with genuine *Pump it Up* software, but you also want to play on simulators or other software without reflashing your controller, you have a few options to choose from.

If you're reading this as a printed paper manual, please see <https://5panel.dance/guide/piuio-drivers> for more information, troubleshooting steps, and easily clickable links!

Windows

Required software:

- [Zadig](#)
- IO2Key / [Precompiled](#) or [Source Code](#)
- [Visual C++ Redistributable Packages for Visual Studio 2013](#) (**MUST be the x86 version**)

Installation:

1. Open up Zadig and plug in your controller.
2. From the top dropdown, select the controller.
3. There will be two boxes in the "Driver" section, separated by an arrow. On the right side, make sure "libusb-win32" is selected.
4. Hit the install button and wait for the process to finish! This may take quite a long time in some cases. Enjoy the silly progress messages!
5. If the driver installation fails, or does not persist between reboots, you may need to [enable test signing mode in Windows](#).
6. Reboot your PC!
7. Unzip the IO2Key archive. Doesn't particularly matter where, but I'd recommend putting it somewhere other than your Downloads folder, so it doesn't get mistakenly deleted.
8. Navigate to where you extracted the archive and run IO2Key.exe, while the controller is plugged in.
9. You should now see a little pad icon in your system tray at the bottom right of your taskbar, with no popups. If this is the case, great! You're ready to use your PIUIO! If you get an error message saying that there is no PIUIO detected, head to the troubleshooting section on the next page.

Troubleshooting

- Open up the device manager and locate the PIUIO. If you can't find it, unplug it and plug it back in to see which item disappears and reappears. Right click it, uninstall the driver, and then reinstall the libusb-win32 driver with Zadig.
- Check your USB cable. Some cables are charge-only and don't do data transfer. If you swapped cables between install and testing, this could be the case.
- Reboot your PC! Sometimes turning it off and on again is all it takes. Especially if you skipped that step in the installation section.

Linux

Required software:

- [din's fork of dipholy's PIUIO kernel module](#)

Installation:

1. Install gcc: `sudo apt install gcc` (Debian) / `sudo pacman -S gcc` (Arch)
2. Download and extract the source code for the kernel module to `~/piuio`, or clone the repo: `git clone https://github.com/DinsFire64/piuio ~/piuio`
3. Navigate to the "mod" directory: `cd ~piuio/mod`
4. Run "make": `make`
5. Install the compiled module: `sudo make install`
6. You may encounter an error related to SSL stuff. You can safely ignore this! It's only important if you're trying to build a cryptographically signed module.
7. Reboot your PC!

Troubleshooting:

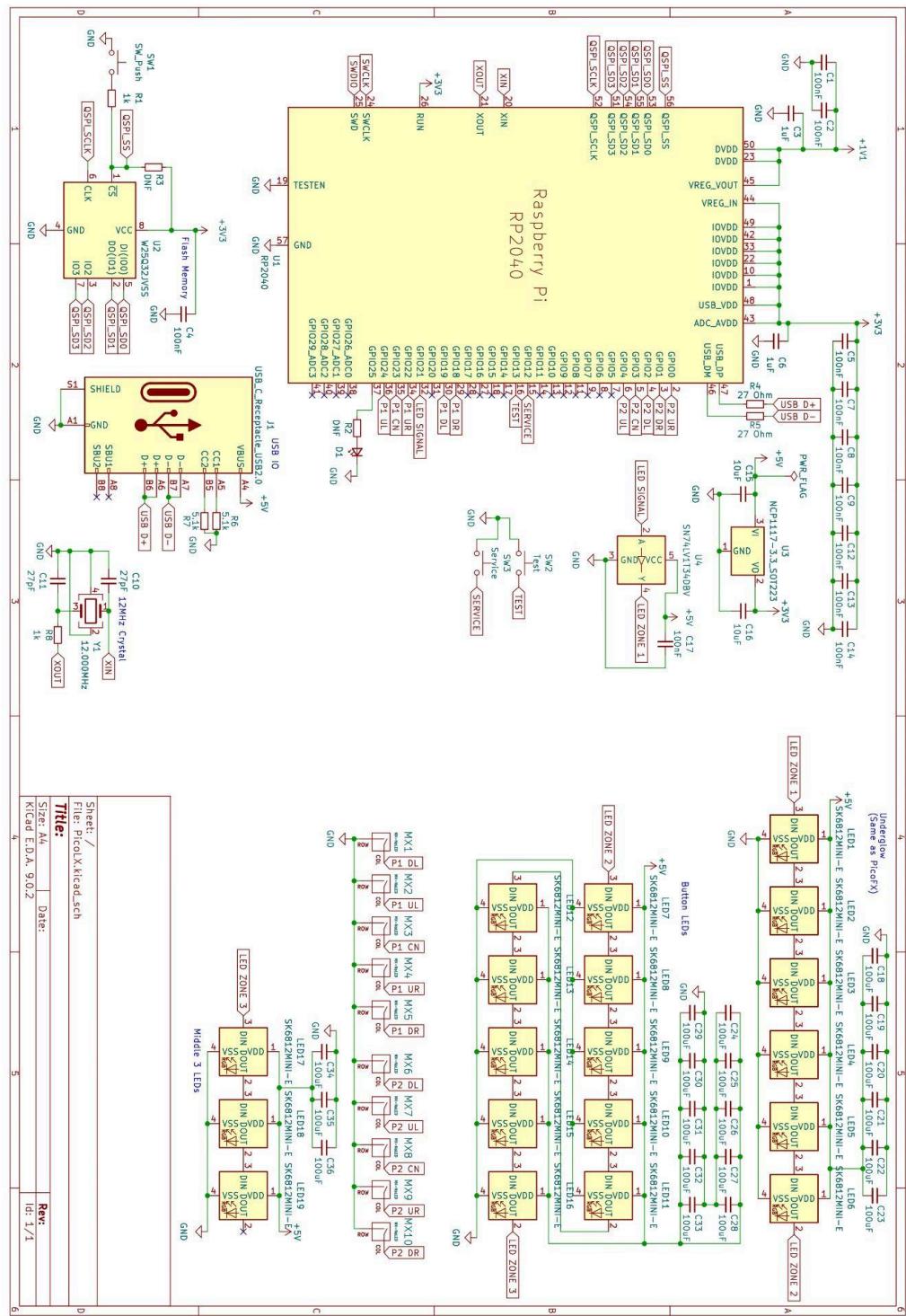
- Search for the PIUIO module to ensure it actually installed: `find /lib/modules -name 'piuio*'`
- If the search turns up nothing, reboot and redo the installation.
- See if the PIUIO module info exists and can be found: `modinfo piuio`
- If the module cannot be found, run the `depmod` command to "rescan" installed modules and hopefully help the system find it.
- Attempt to run `modinfo piuio` again to see if the module is recognized. If so, great! Run `modprobe piuio` to add the module to the kernel, followed by `lsmod | grep piuio` to ensure the module was properly loaded.
- If all else fails, double check the key bindings in whichever game you're playing! It's an easy step to miss!

Troubleshooting

- The controller is on and lights up, but my device doesn't detect it!
 - Ensure your USB cable supports charge and sync. There are some cheap USB cables out there (even type C ones!) that only carry power, with no data lines. It might occasionally be necessary to try several cables.
- Some of the LEDs do not work/are acting up. What's going on?
 - The LEDs are all in one long chain, with the colour data flowing through each, one by one. It's likely that there's a "weak link in the chain," so to speak, which could be either the data input of the malfunctioning light, or the data output of the one before it. See the schematic towards the end of the manual to narrow down which LEDs may be acting up, and touch up the solder points for both.
- Some of the buttons aren't responding. What's going on?
 - The most common cause, I find, is bent legs on the switches. Occasionally, a slightly bent leg might not quite line up with the socket, and when the key is pushed into place, it might crumple and fail to make contact. Remove the offending keys and ensure the legs are straightened and at a 90 degree angle!
 - Otherwise, double check the soldering on the sockets. They take a lot of heat, and it's not difficult to end up with a cold joint. Don't hesitate to reflow the solder.
- I was replacing my switches, and the two boards came apart completely! Now I can't put it back together properly.
 - You'll need a #2 Phillips screwdriver for this one. Remove the 4 screws holding the front plate onto the case, and separate the two. **Be careful not to lose the two side buttons!**
 - If all the switches are still attached to the plate, remove all except the 4 outermost corner switches. If most of the switches have been removed, snap the outer 4 in.
 - Make sure all the pins are at a 90 degree angle, and aren't bent. Carefully line up the plate with the main board. The two side buttons should be on the right, and facing down. Gently push the two halves together, and make sure they're flush.
 - Snap the rest of the switches in place one at a time with one hand, while holding the two halves of the PCB together with the other. After each one, ensure the two halves are still properly lined up, and that the switch is sitting flush on both ends.
 - Once all the switches are lined up, carefully reinstall the side buttons, and install the board while holding the case at an angle, so the side buttons don't fall out. Start with the corner between the USB port and side buttons, and tilt the remainder of the board into place.
 - Reinstall the 4 Phillips screws. **Do not over-tighten them!!!**
 - Test it out. If everything works, you're good to go!

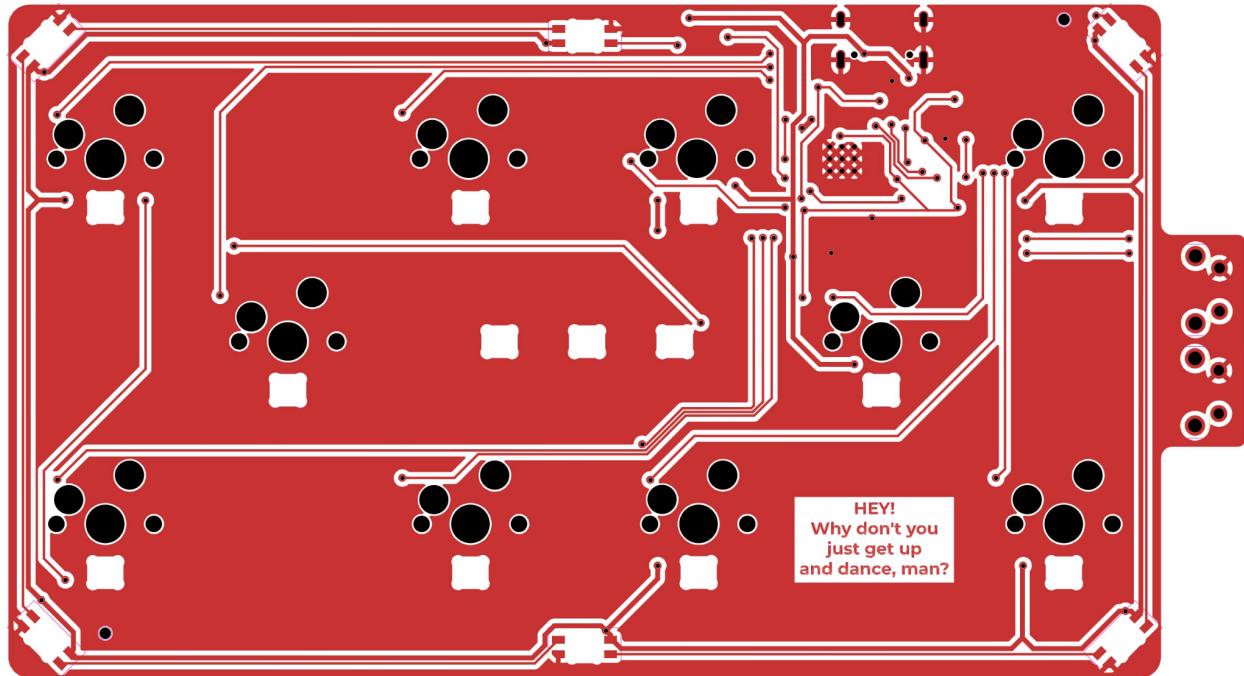
- I'm using the gamepad firmware, but I can't get it to swap to keyboard mode / I can't get the RGB toggle to work. What gives?
 - As of writing, keyboard mode is quite finicky, and I'm still trying to narrow down some firmware-side fixes for these. However, in the meantime...
 - Unplug the controller for **at least 30 seconds** between attempting to switch modes
 - Ensure you're holding down the button with solid and consistent pressure while plugging the device in
 - Only hold down the button that toggles your desired mode, and make sure nothing else is being pressed
- I broke the case! How can I get a new one?
 - As the case files aren't currently available online, please reach out to me directly, and I'll send you the model for 3D printing! If you don't have access to a 3D printer, there's a chance I may be able to send you a replacement, for the cost of shipping. Otherwise, services like PCBWay and JLCPCB also offer 3D printing services.
- I broke/damaged the front plate! How can I get a new one?
 1. Head to <https://github.com/dj505/PicoLX/releases> and download the "Fabrication Files" zip. Extract it and locate the "gerber.zip" inside of the "Plate" folder. **Don't extract gerber.zip.**
 2. Navigate to a PCB manufacturing website like JLCPCB or PCBWay, and upload the gerber.zip file (again, don't extract it).
 3. You'll see a lot of options. You can ignore most of them! Just double check the following options, and leave the rest as their defaults:
 - a. **PCB thickness:** 1.6mm
 - b. **PCB colour:** Anything! The original controllers are either black or white. Just look at the preview and make sure it looks good to you :)
 - c. **Material type:** FR4 TG135
 - d. **Surface finish:** LeadFree HASL (cheap) or ENIG (fancy shiny gold! but more expensive)
 - e. **Mark on PCB:** Remove mark
 4. Save the order to your cart and submit it! Cheap shipping takes a few weeks, but it's pretty inexpensive, and you'll be getting 5 of them. There's no way to order just a single one-off, as 5 is the minimum.
- My controller won't show any signs of life whatsoever, regardless of the cable. What should I do?
 - Please reach out to me directly, and I'll help troubleshoot!

Schematic



Board Drawings

Top layer



Bottom Layer

