Portal Box Printed Circuit Board

## Device Specifications

The Printed Circuit Board (PCB) of the Portal Box is where all of the hardware and connections, excluding peripherals, are housed and organized. All of the machine power and/or data is routed through the PCB where relays are used to manipulate the functionality of machines based on access conditions. Device logic is controlled centrally by the software housed on the box microcontroller (ESP32). Additionally, many of the peripherals such as the LEDs and Buzzer are soldered directly to the PCB. Finally, various headers are also placed throughout the board where peripheral components interface with the board itself. Version 4 of the Portal Box builds on the previous work of older Portal Box iterations, modifying the design to accommodate the microcontroller and user interface changes characteristic of the newest design. This [DigiKey BOM](https://www.digikey.com/en/mylists/list/NODQQK4VBZ) contains all components that go on the board.

## Changes to the Version 4 Portal Box

### 3.11 Board

A majority of the structure and form of the version 4 Portal Boxes is based on the 3.11 revision of the Portal Box. Developed in 2021, the rev 3.11 already includes many changes and progressions from the rev 2.0 Portal Boxes which are presently deployed in Bucknell’s Maker-E and Mooney Makerspaces, as such, much of our user feedback and research was based on the rev 2.0. Rev 3.11 is built in the Altium software, however, this will be changed in the rev 4.0. While the general layout and design of the 3.11 PCB is preserved for the 4.0, various systems are excluded and certain components have become obsolete, warranting additional changes for the next revision of the Portal Box.

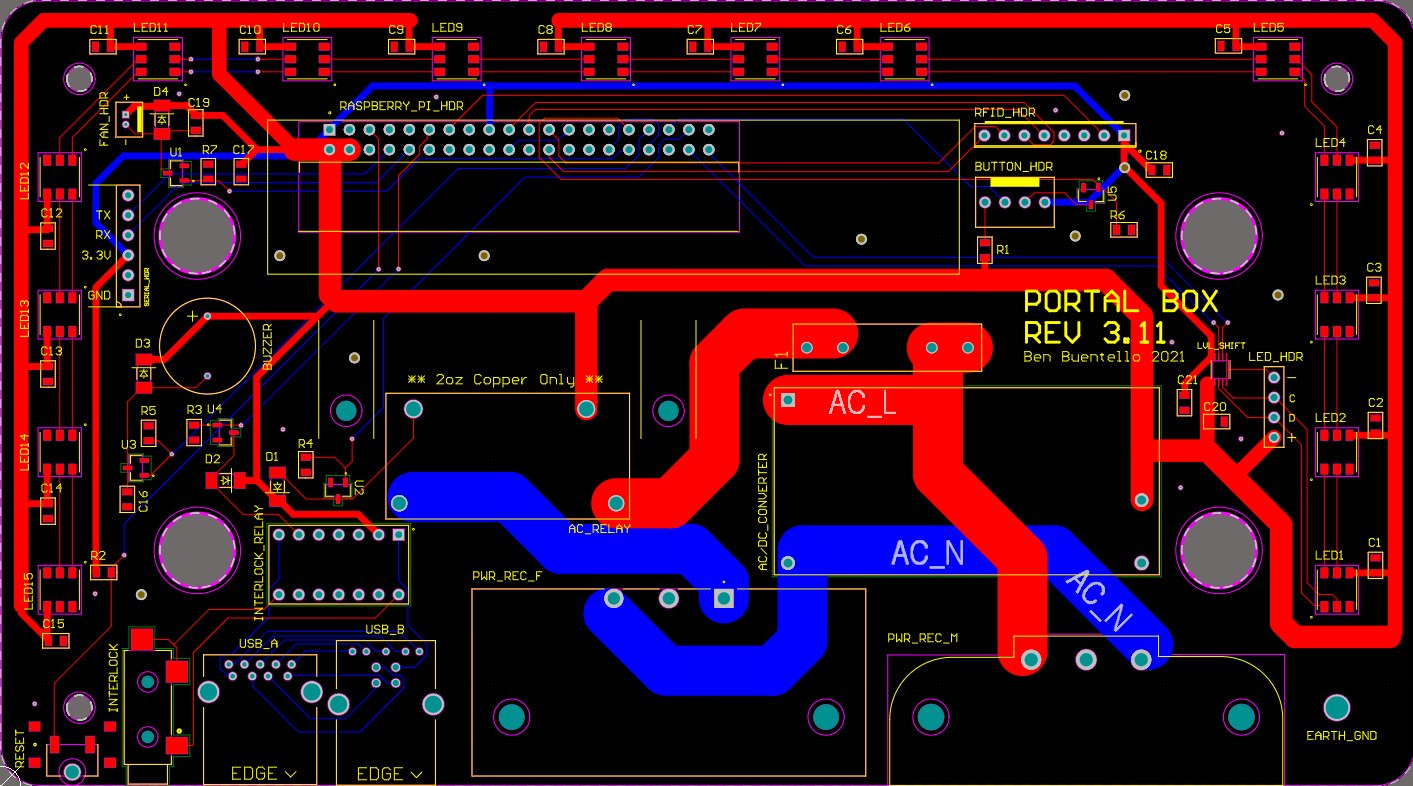


Fig 1: Rev 3.11 PCB Rendering

### 4.0 Revisions

The 4.0 version constitutes the most major changes to the system’s hardware and CAD software thus far. The most notable switch is the moving of the Portal Box schematic and PCB to KiCad, an open sourced software. Much of the structure and layout of the board from the 3.11 version has been maintained. The Raspberry Pi has instead been replaced with an ESP32. Additionally, certain systems and/or headers have been determined to be extraneous and were removed from the board including the button header, serial header, fan header, and interlock jack. The power relay and fuse from the 3.11 version are now categorized as obsolete components and have been replaced with parts equivalent in function. The USB relay has been updated to cut the Data + and Data - lines between the USB ports instead of power, however, that system is erroneously wired in the 4.0 version. All unassigned GPIO pins have been routed to 2 6-pin headers to be accessed for testing and development.

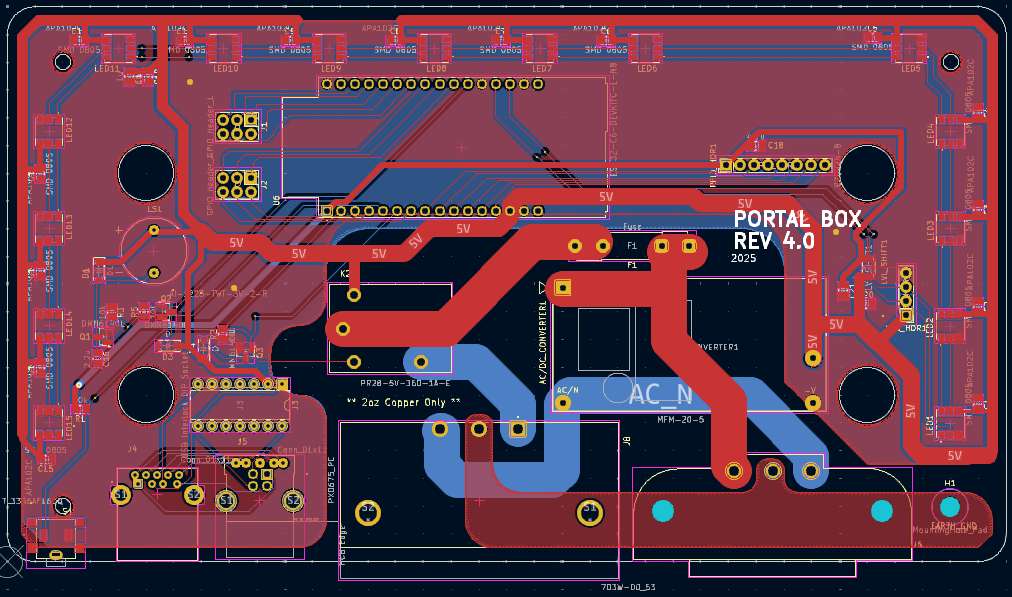


Fig 2: Rev 4.0 PCB Rendering

### 4.1 Revisions

The 4.1 version represents some design and configuration changes that improve upon the 4.0 revision. Most notably, the Neopixels that make up the lighting display within the box have instead been replaced with Dotstars. The USB B port for the USB interlock system has been replaced with a USB C port for future compatibility with a more widespread standard. However, this relay system is still wired erroneously on this version. Finally, two new headers have been introduced, the matrix keypad header and the SPI header for communication with the LCD screen. The GPIO pin header persists but is instead a single 4 pin header.

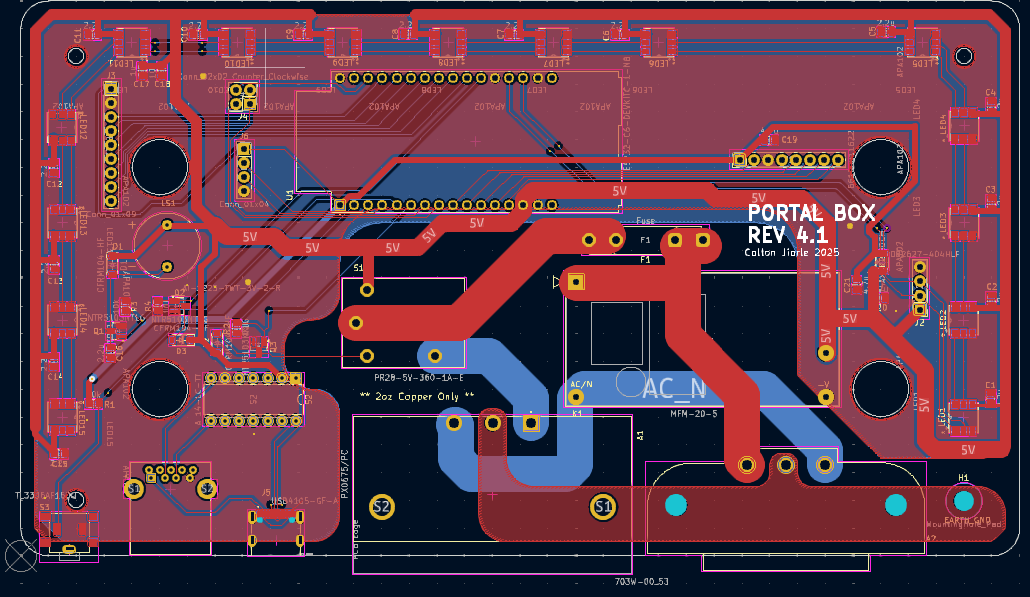


Fig 3: Rev 4.1 PCB Rendering

### 4.2 Revisions

The 4.2 revision to the Portal Box PCB includes minor improvements and fixes. The pins which control the data and clock lines for the LEDs have been switched to prevent issues with accidentally calling the boot file. An additional LED has been added to the board to fill in the gap where previously the cardholder and button would have blocked. Minor configurations have been made of the matrix keypad and SPI headers. Additionally, a new header for the optional button variant has been added. The GPIO pin header now includes a 3.3 V and GND connection. Revisions have been made to the USB relay for the proper wiring configuration. Additionally, the USB C pin wirings have been adjusted to add redundancy to the signal leaving the port.

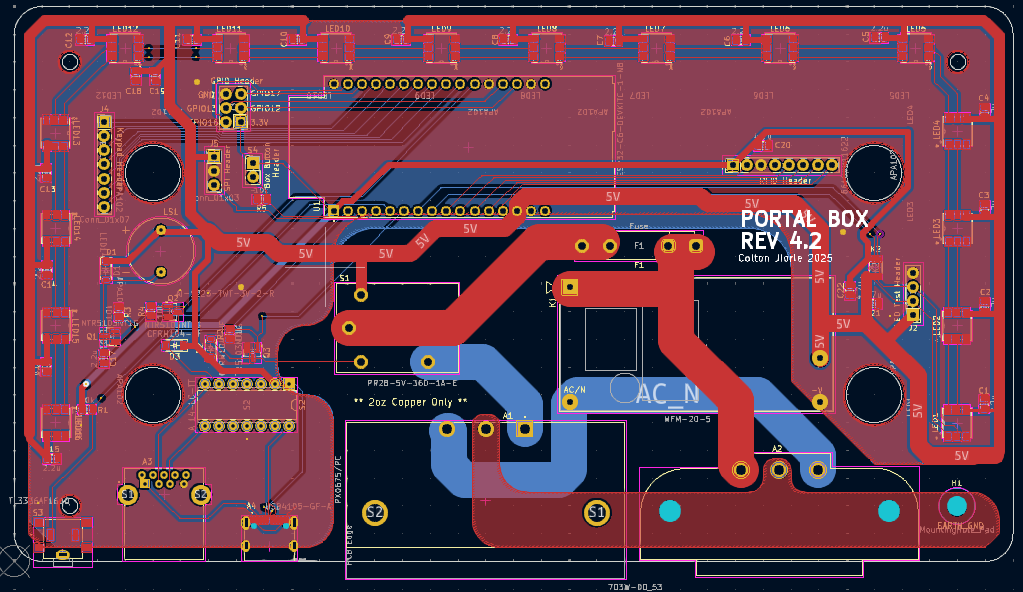


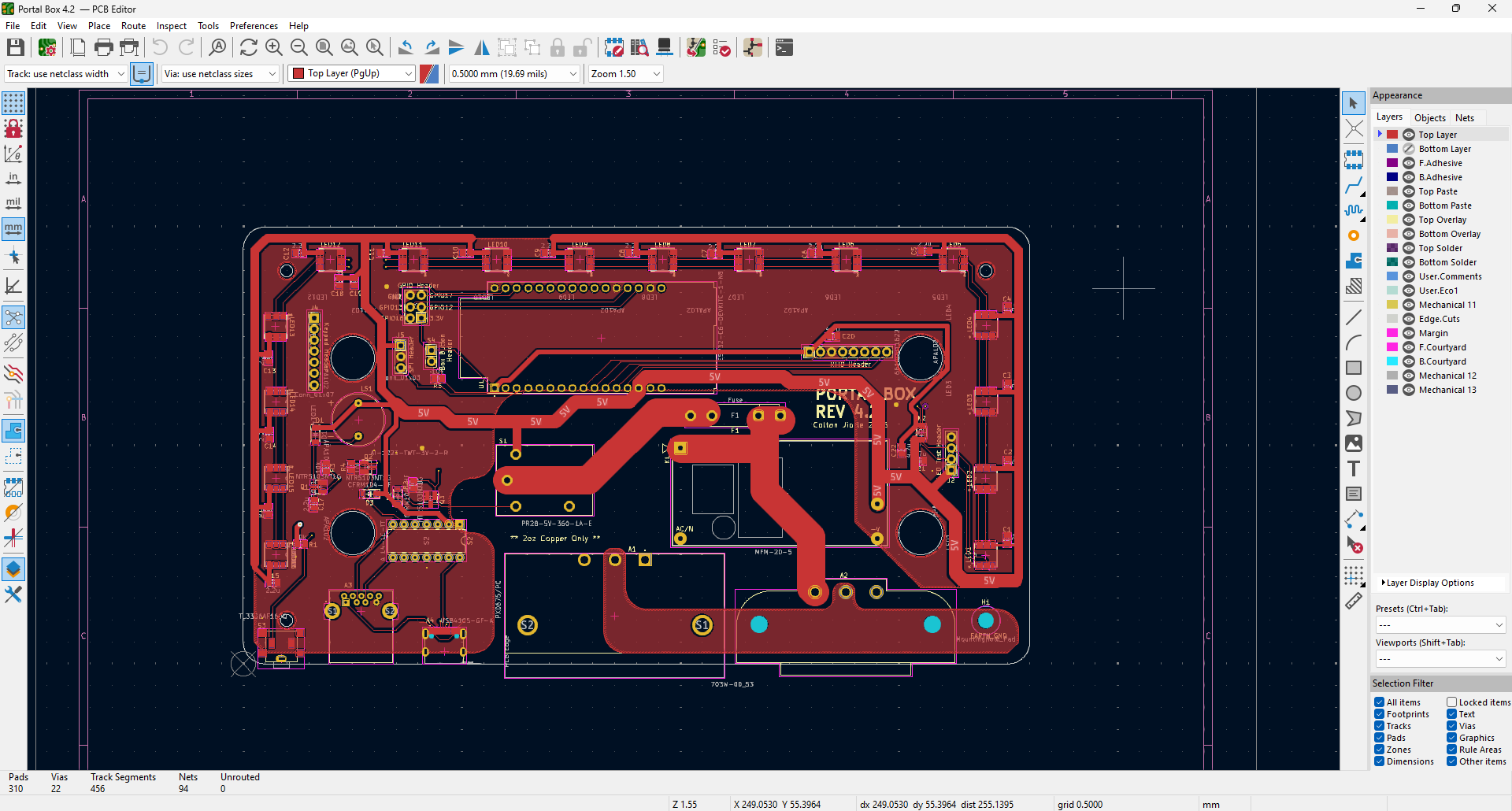
Fig 4: Rev 4.2 PCB Rendering

## PCB Ordering Guide

This guide is for ordering a PCB from JLCPCB. If you wish to make modifications from the original design, begin at step 1 for ordering. If you plan to order the Portal Box as designed, skip to step 5.

Step 1: Open KiCad Files

* First, open the Portal Box 4.2 files in KiCad (.kicad\_pro, .kicad\_sch, .kicad\_pcb)
* If you wish to make any changes to the files, they can be adjusted in this window
  + Reference the KiCad Documentation Manual for help learning the software: https://docs.kicad.org/



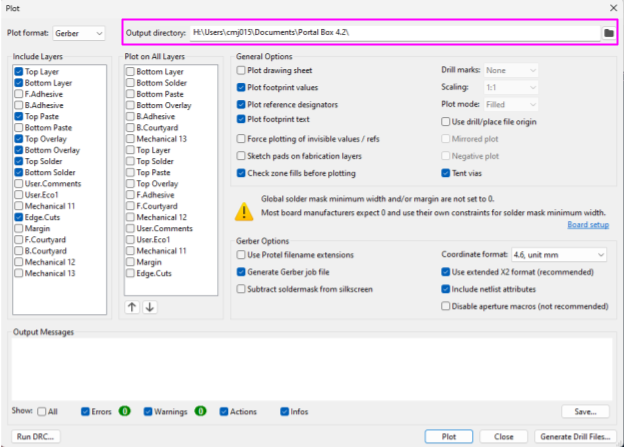
Step 2: Navigate to the gerbers output page

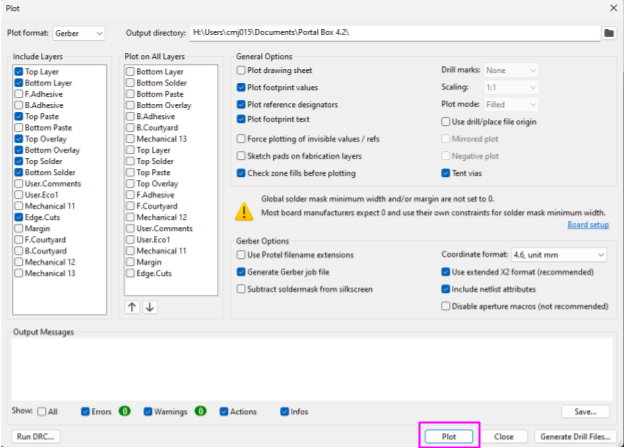
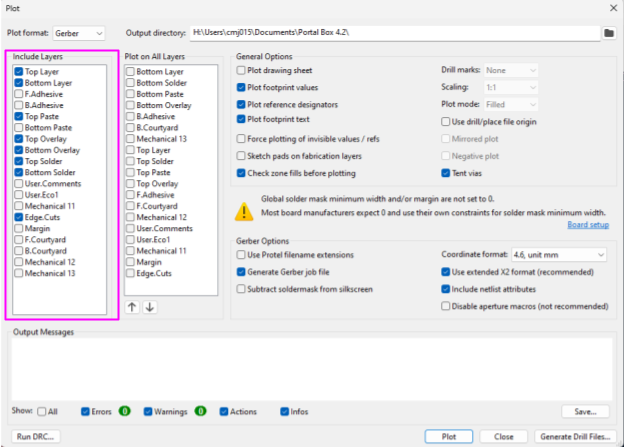
* Use File > Fabrication Outputs > Gerbers (.gbr)



Step 2: Plot Gerber Files

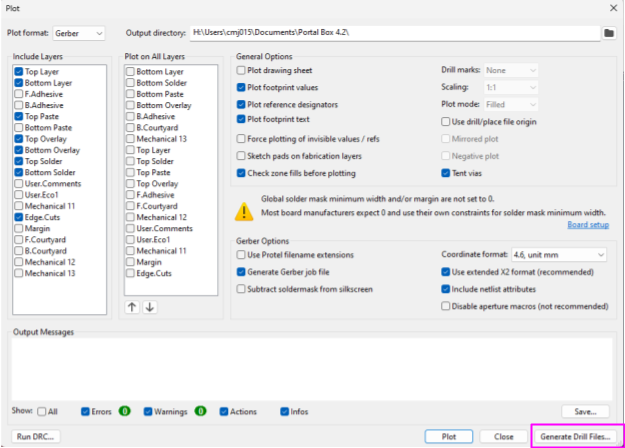
* First, you should have a file set up in your computer’s file system where you will be exporting the gerber files to
* Second, you will navigate to the output directory and select that file as the output for the gerber files
* Third, on the included layers column, select the following materials:
  + Top Layer, Bottom Layer, Top Paste, Top Overlay, Bottom Overlay, Top Solder, Bottom Solder, Edge.Cuts
* Finally, select “Plot”

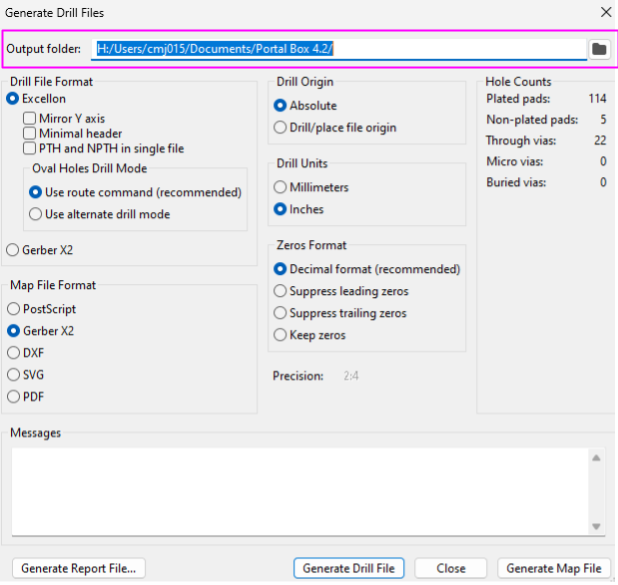


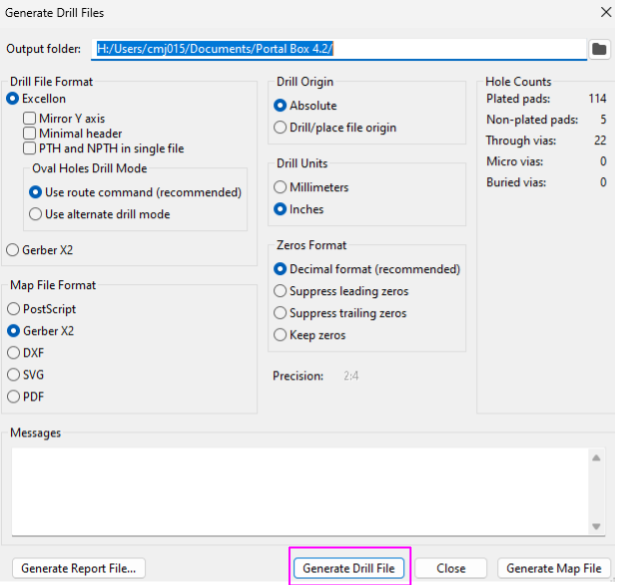


Step 3: Plot Drill Files

* On the same menu as the gerber outputs, select “Generate Drill Files..” in the bottom right corner
* Next, ensure the output folder is the same as where you copied your gerber files to
* Finally, select “Generate Drill File”

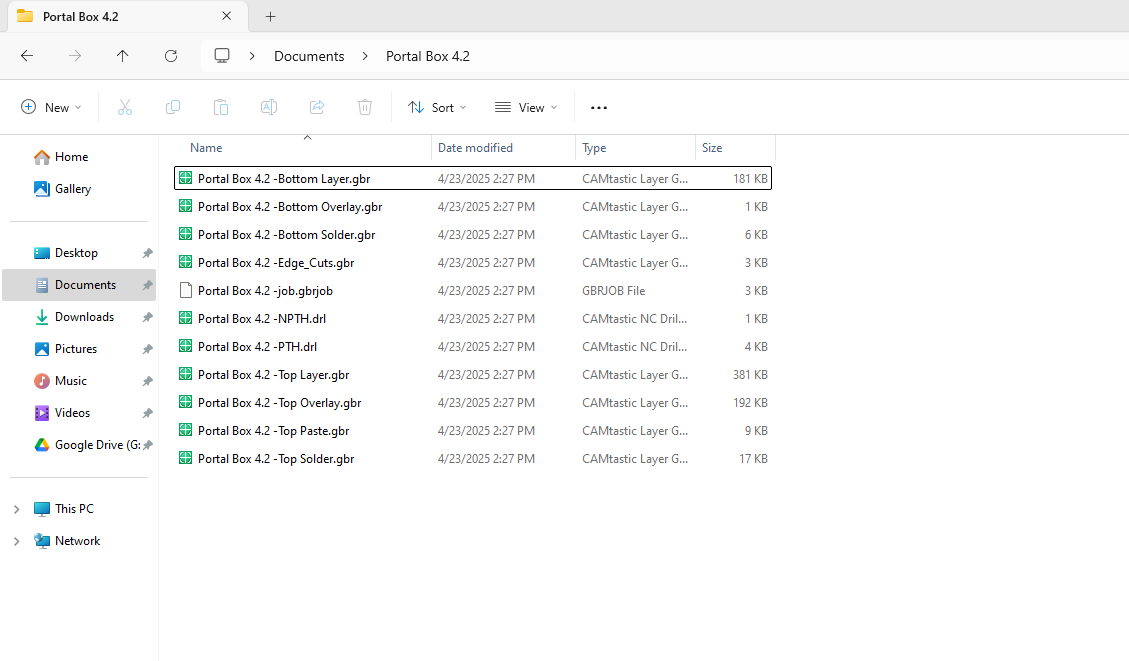


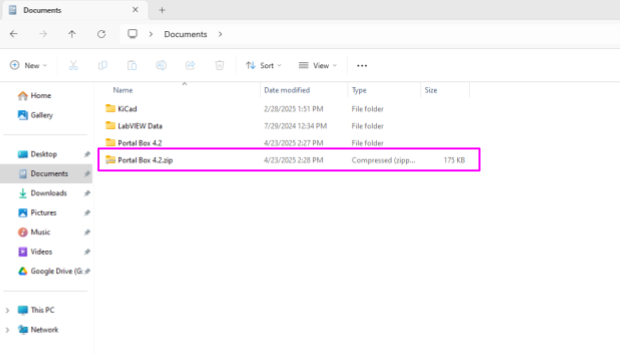




Step 4: Create Zip File

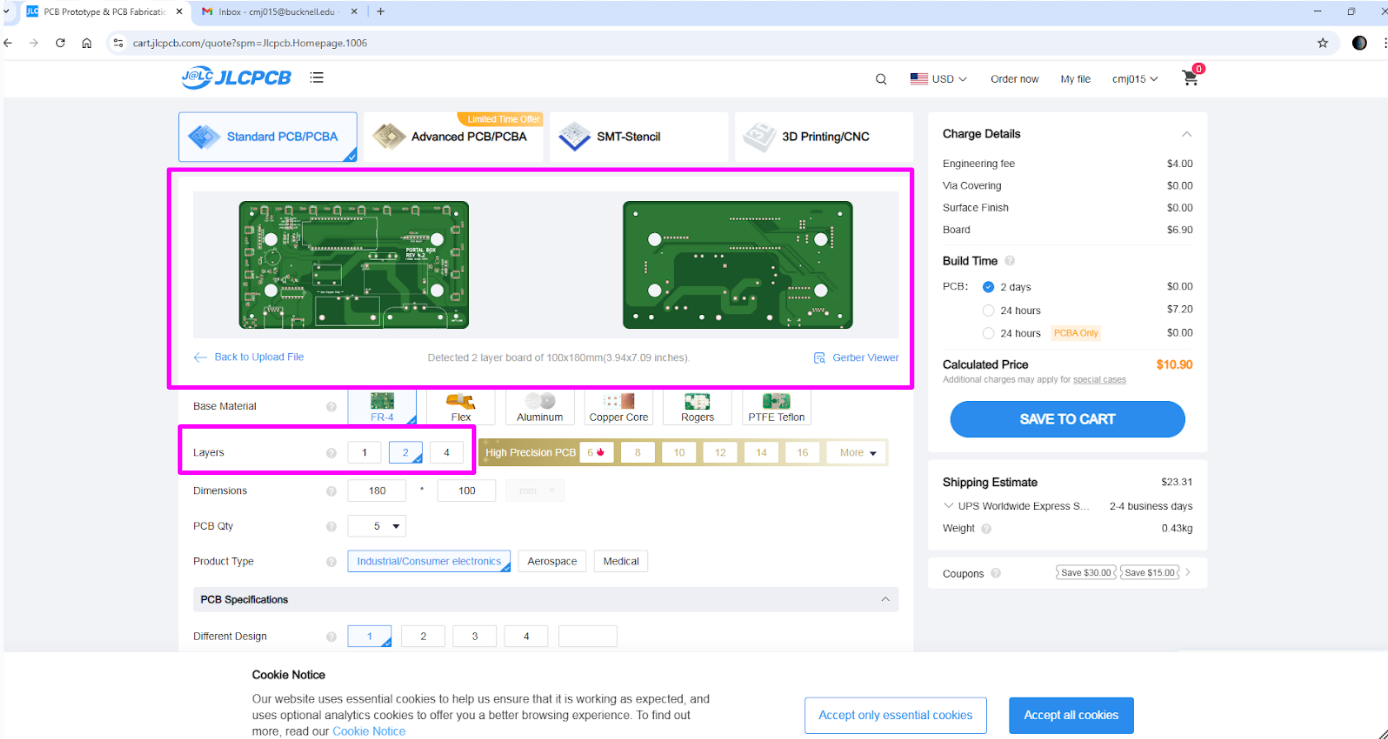
* Navigate to the folder you chose to export your files to and ensure all of the contents are there (8 .grb, 1 .gbrjob, 2 .drl)
* Select the options to compress this folder into a .zip file

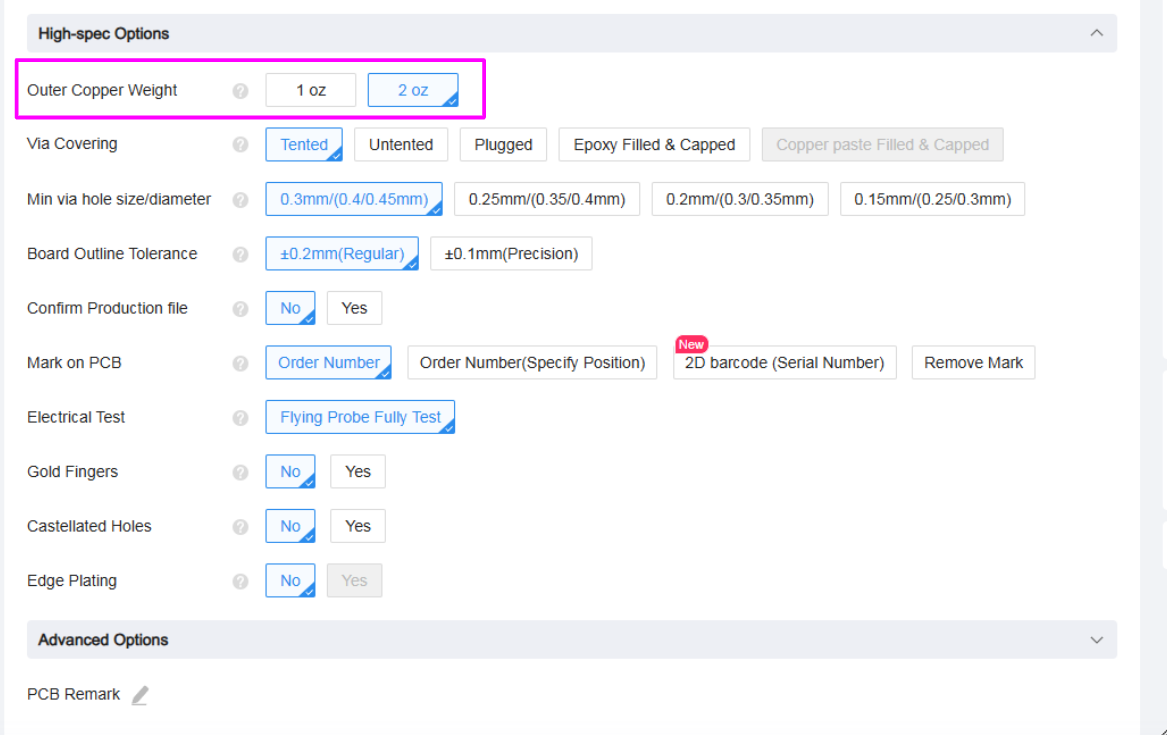


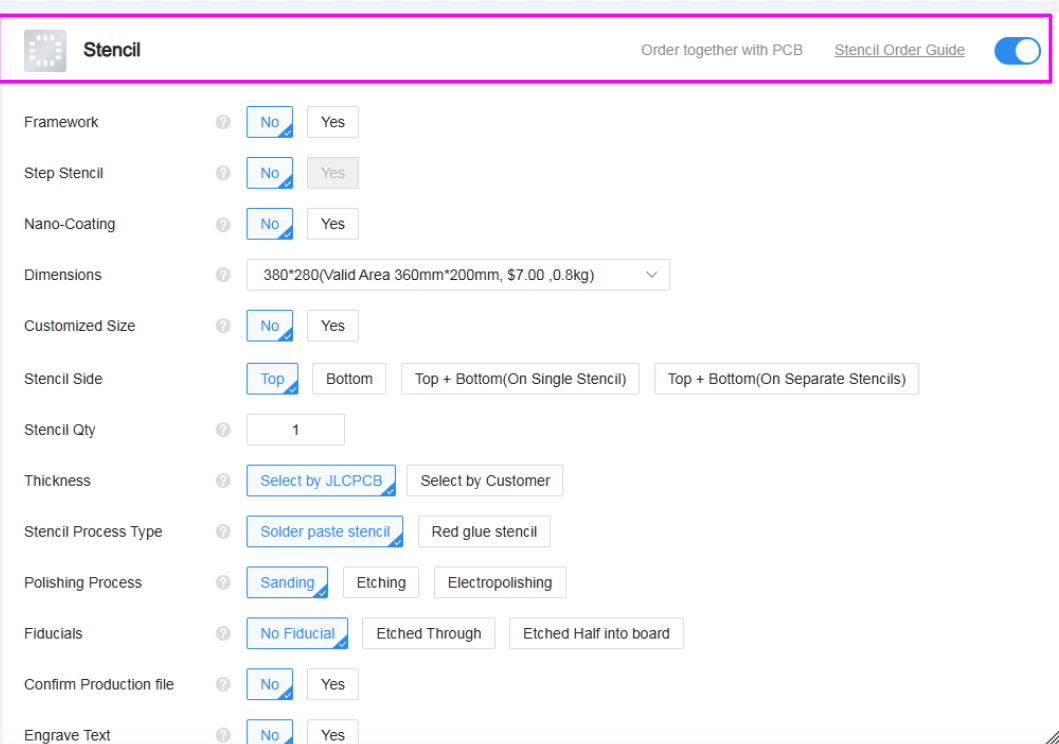


Step 5: Order From JLCPCB

* Navigate to jlcpcb.com/ and select the option to get an instant quote
* Upload the zip file you created to “Add gerber file”
* Select 2 for the “Layers” category
* \*\*\*Ensure that you select “2 oz” for Outer Copper Weight\*\*\*
  + 2 oz copper is necessary for handling the Portal Box’s intended capacity of 15 A
* Finally, select the option to also order a stencil for pasting on your pcb







Step 6: Place PCB Order

* Review your order to ensure all options were selected correctly
* Select “Save to Cart” and follow JLCPCB’s instructions for completing your order

## PCB Soldering Guide

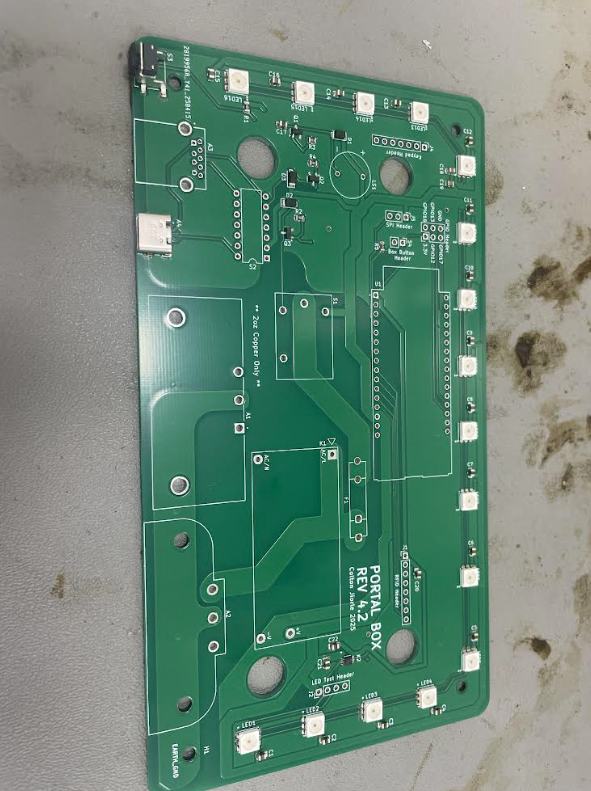
Step 1: Collect all components together

* Before beginning the soldering process, ensure that you have all the [necessary components](https://www.digikey.com/en/mylists/list/NODQQK4VBZ)
* A complete list of device components can be found in the Portal Box Bill of Materials
* It is helpful at this stage to separate the through-hole components from the surface mount components

Step 2: Surface Mount Components

* Use the stencil to distribute solder paste to the pad mounts
* Components can be placed onto the PCB by matching the component reference number in the Bill of Materials with the corresponding number on the PCB itself
  + Take special caution when placing the USB C port and Level Shift, these components have the smallest pins on the board and are the most likely to be misaligned, causing connectivity issues
  + The side of the diode footprint that have a closed line indicates the location of the cathode, the anode of the diode should be oriented towards the opening in the box
* Use a reflow oven to melt the solder and fuse the components to the board

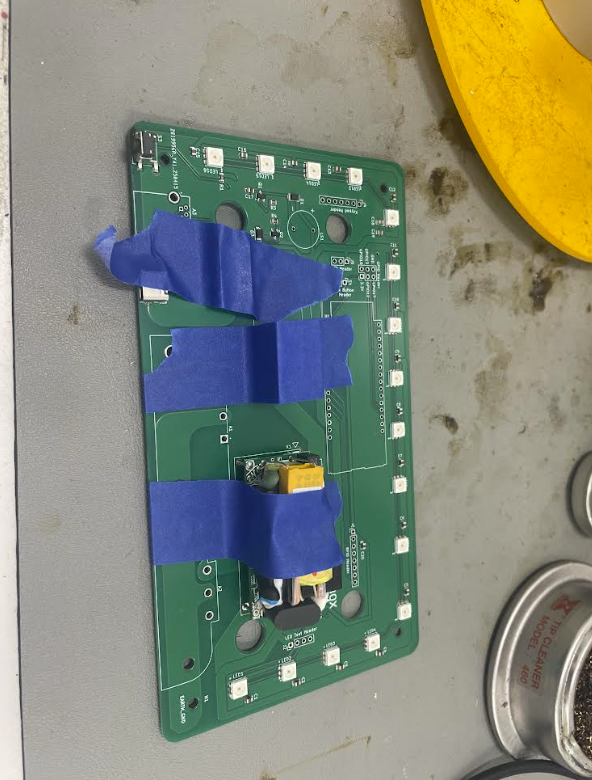






Step 3: Through Hole Components

* Use a soldering iron and wire to solder the remaining components to the board
* Note that the ground pins and AC power connections may take longer to solder due to the large planes providing heat dissipation



Step 4: Visual Inspection

* Perform a visual inspection of all solder joints to ensure connectivity