RTI Build Guide

This build guide covers the electrical side of the RTI's build. There are three assemblies that are all connected to form the RTI system, the display, the mainboard, and the dome's LEDs.

Main Board

The mainboard is the Arduino MEGA 2560 and its shield, which contains the LED drivers, matrix controls, camera shutter triggers, and other parts.

Step 1: Getting the Parts

Our circuit board was purchased from JLCPCB.com. All of the parts, listed in the "RTI Main Board BOM.csv" file, were purchased from either Mouser or DigiKey, based on availability or cost. We also purchased the solder paste stencil from JLCPCB.

When purchasing the parts, double-check a few things to make sure there won't be a problem in your configuration:

- R66, R73, and R76 are all settings resistors assigned values of 549Ω. These are all selected for a 1A LED configuration; there are three for the three independent color channels. If you decide to use any LEDs which can't operate at 1A continuously - like the UV LEDs we decided to use - you need to do one of two things. One, change one of the resistor's values and only plug the LEDs of the lower current color into that channel, or two, place a resistor in series with the potentiometer used to control current for that channel, and make sure that potentiometer and that LED color are always on the same channel (which is easier to undo.) If you're choosing the former, simply pick the new value from the chart below. If choosing the latter, find a resistor value that, when added to 549Ω, will sum to no less than the value needed in the chart below. (This is the table from the CAT4101 LED driver datasheet, https://www.onsemi.com/pub/Collateral/CAT4101-D.PDF:)

Table 6. RSET RESISTOR SETTINGS

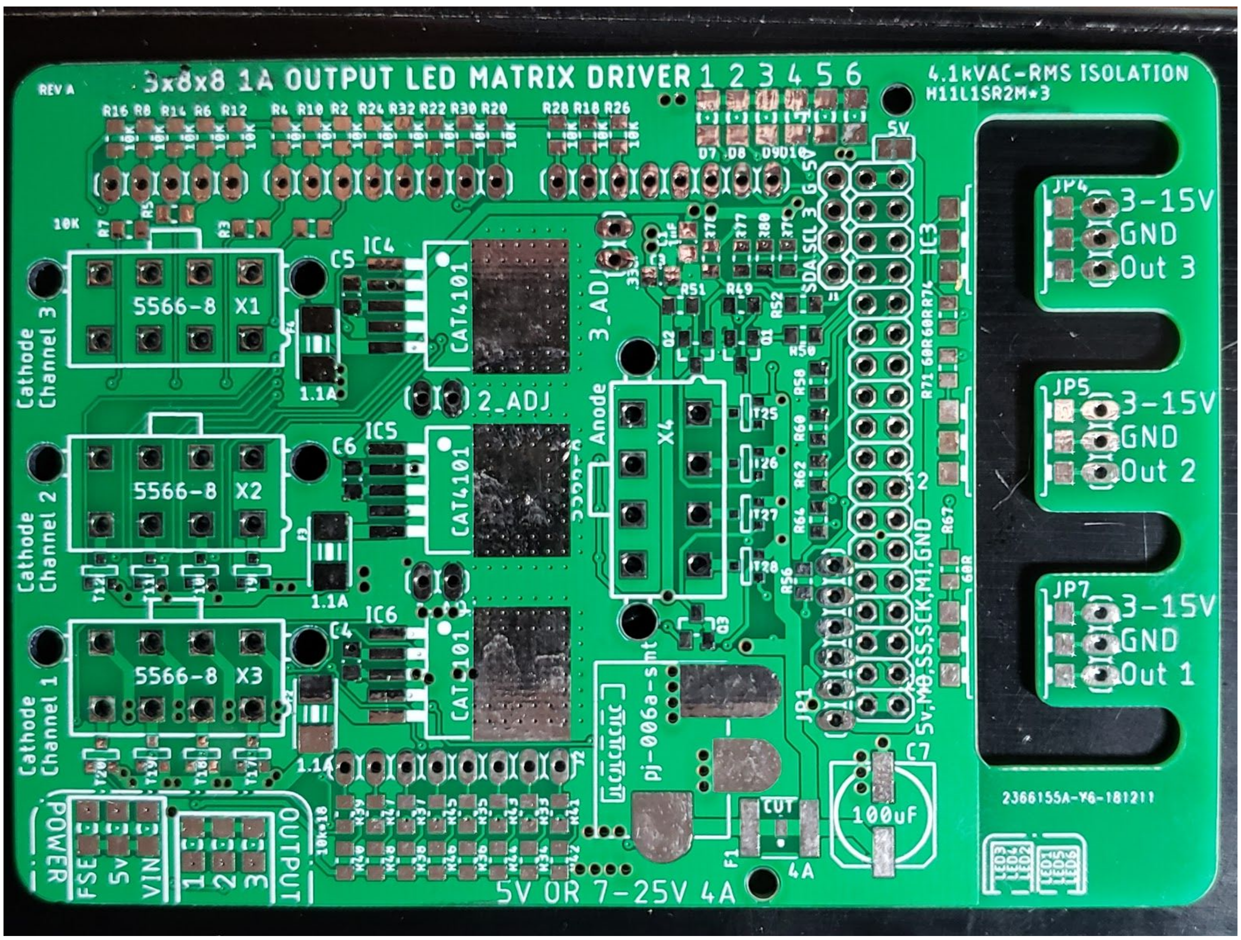
LED Current [mA]	RSET [Ω]
100	4990
200	2490
300	1690
400	1270
500	1050
600	866
700	768
800	680
900	604
1000	549

When buying these resistors, if there isn't a standard resistor value equivalent to the one in the chart, buy the next higher value (instead of 549Ω resistors, we purchased 560Ω resistors.)

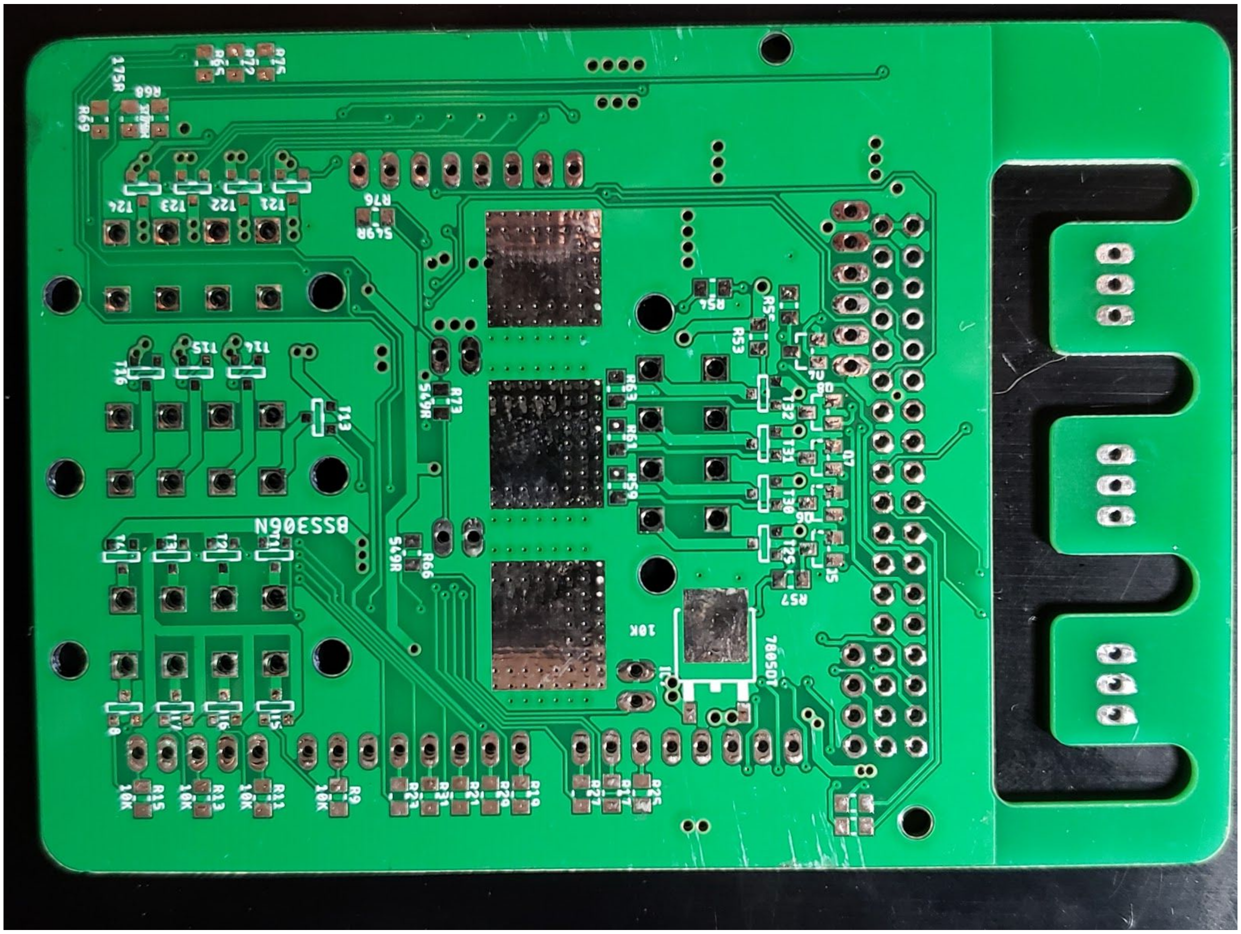
- If you elect to change one of the resistor values above, I would recommend also changing the value of a potentiometer. Since the highest usable resistance is around 5kΩ, the sum of the setting resistor (R66, R73, or R76) and the max value of the current adjustment potentiometer (3_ADJ, 2_ADJ, or 1_ADJ, respectively) should be around 5kΩ too.
- I did not use a fuse on the mainboard input, even though it is designed to allow for one. In our design, the fuse would need to be rated for a larger current than the power supply we used can provide, and that supply has a fused output anyway. Each LED channel is already individually fused.

These are photos of the circuit board before populating it:

TOP SIDE:

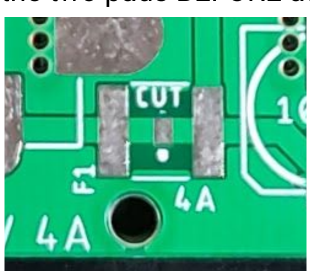


BOTTOM SIDE:

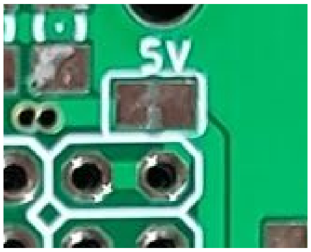


From these photos a few things are clearly visible:

- The isolation slots on the right, which the camera shutter control isolators are mounted across. The solder joint on these can be delicate since they can be ripped off when the board flexes.
- F1, the polyfuse near the barrel jack footprint, is neither required nor used in our prototype. If you'd like to use it, you will need to cut the exposed trace between the two pads BEFORE adding the fuse.

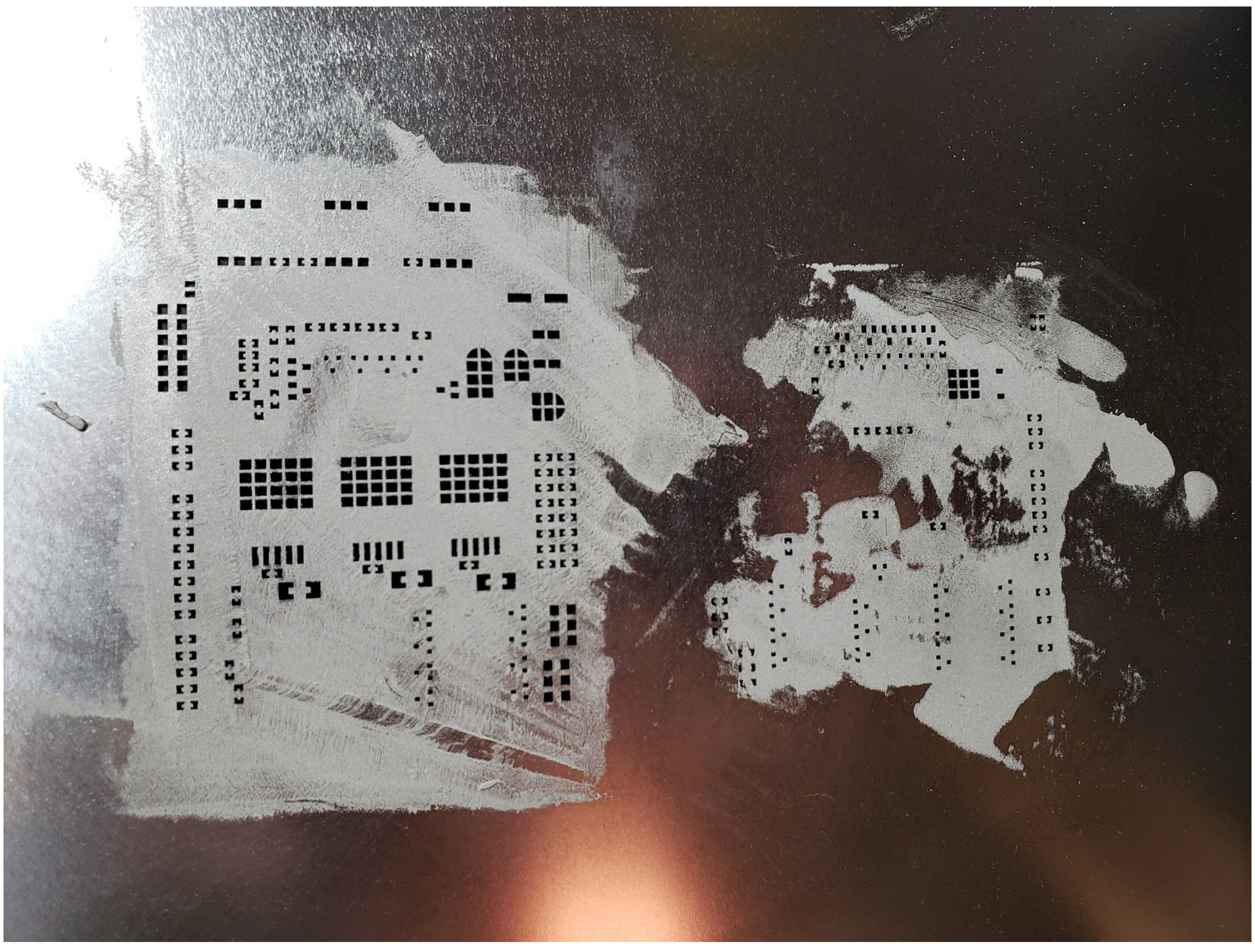


- At the top right of the top side, there is a small jumper that will supply 5v to the Arduino. If you would like the Arduino to always have power supplied separately, break this jumper. Otherwise, it is good to add solder to it to allow it to carry more current.

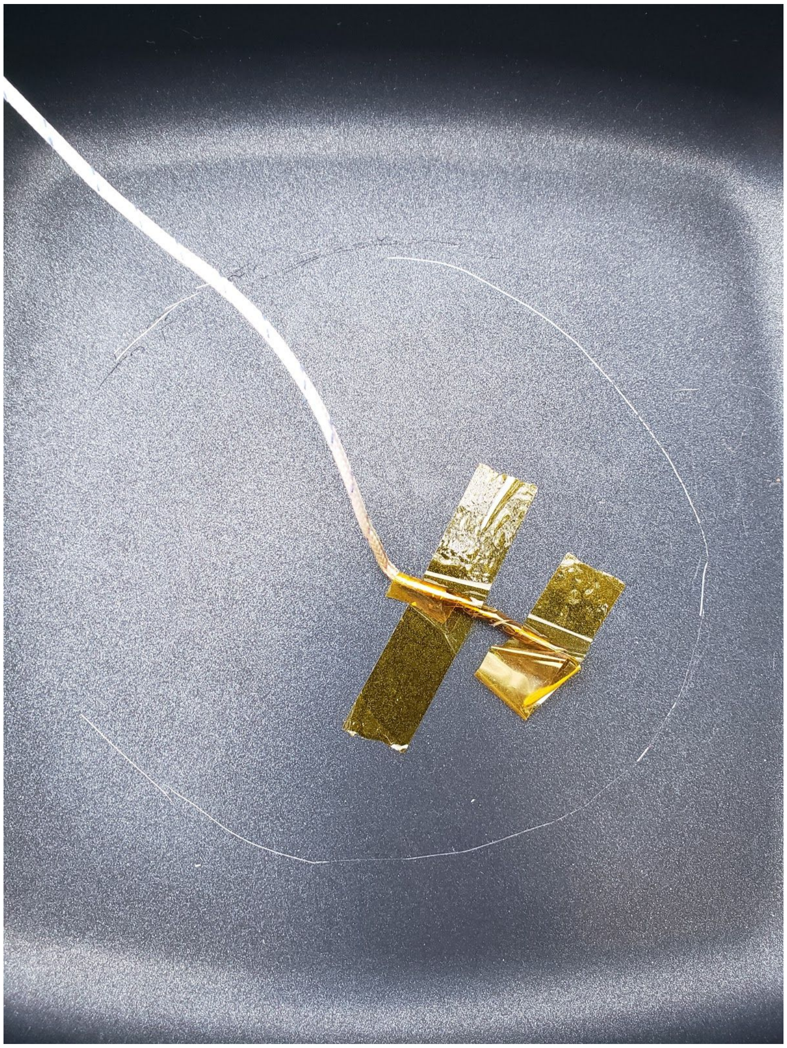


Step 2: Soldering

After getting the circuit board and all the components, a solder paste needs to be applied using a stencil. Our stencil was also purchased on JLCPCB.com, who will make and ship both items as one order. This is our stencil (after it was used).



Adding solder paste and soldering can be a complicated process, which varies depending on your tools. Personally, I used an electric skillet in place of a reflow oven for the top layer (the more complicated side,) and used a hot air station on the bottom of the board after the top was already soldered and cooled. If you’re using a hot air station, make sure to make sweeping passes across entire sections of the board to avoid heating up only one spot, which can damage the board and make parts on the underside fall off.



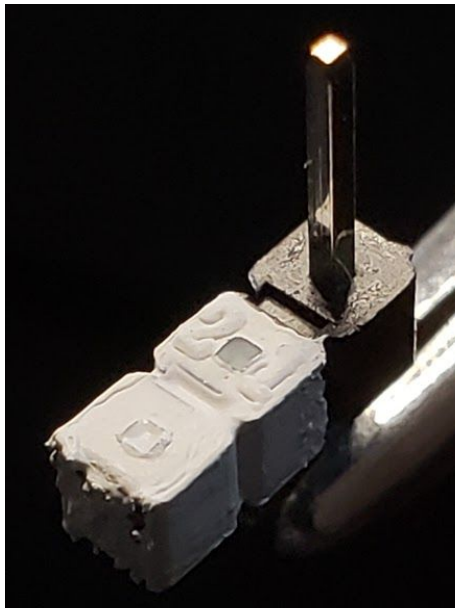
The surface of my skillet, with a thermocouple in the center. I kept the tip of the thermocouple slightly exposed so I could get the circuit board as close as possible to the thermocouple while keeping it level on the surface of the skillet. The scratched-in circle indicates where the heater element on the skillet is, so I can place circuit boards along the loop for more even heating.

There is a lot of advice online about making/using reflow ovens which is good to research if you're planning to try it for this project. I used an electric skillet, which I bought specifically for uses like reflowing boards. I controlled the temperature manually, monitoring the skillet's surface temperature and the board's surface temperature using thermocouples mounted with Kapton tape.

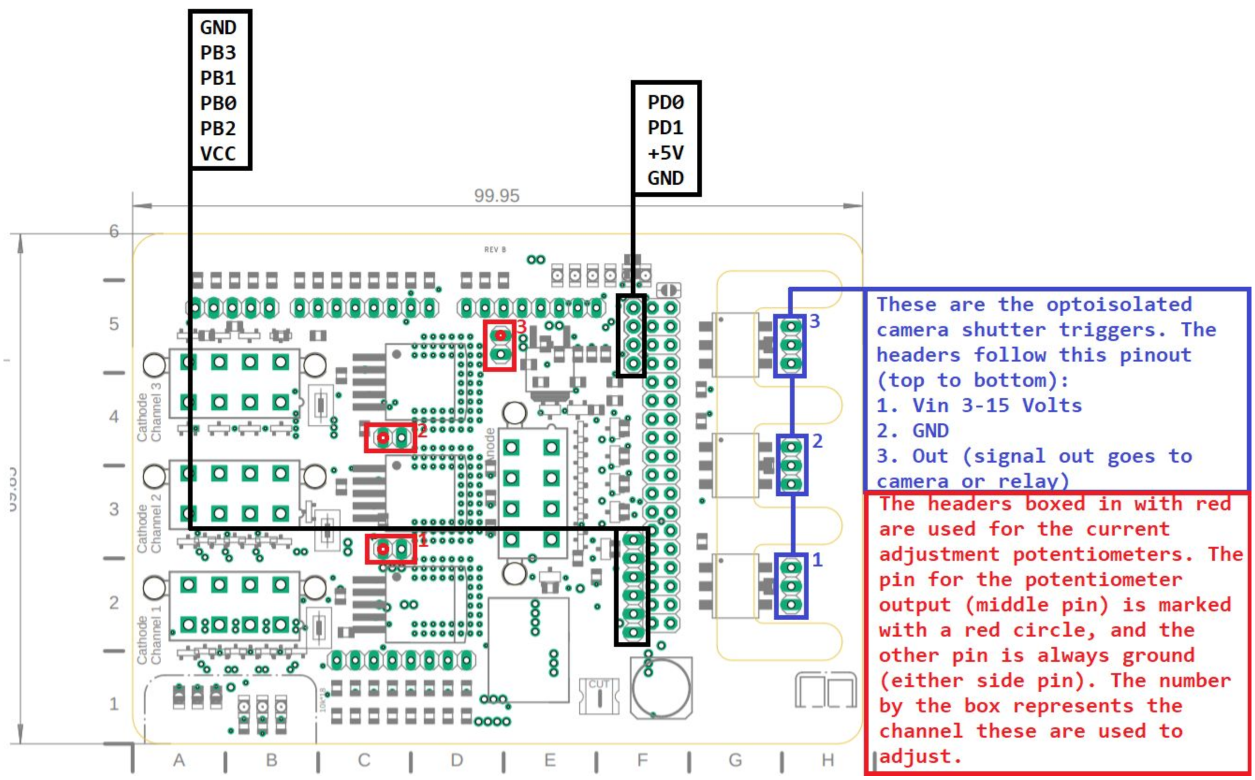
If you choose to do this, take care to read the reflow soldering charts in the datasheets of all parts chosen to make sure that the maximum temperature isn't being exceeded on the device, but also make sure that the solder fully melts and flows before stopping.

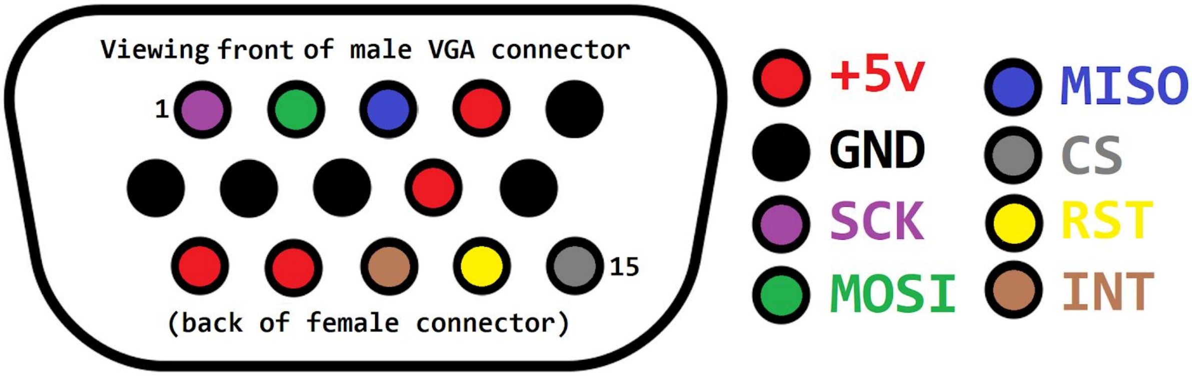
After the surface mount components are soldered, the through-hole parts will need to be manually soldered. I decided later that I would color some positions in headers using white-out.

I did this by pulling the pin (from the position that I would color) out of the plastic and painting over that plastic. I then put the pins back into the plastic holder and soldered the header back on. Later, when I crimped and assembled the mating connector, I colored the corresponding pin.



Use this pinout guide to connect the VGA connector (see more info about VGA connector in the display section:)





MISO → PB3 (Arduino pin 50)
MOSI → PB2 (51)
SCK → PB1 (52)
CS → PB0 (53)
RST → PD0 (21)
INT → PD1 (20)

Step 3: Checking the Final Product

After everything is assembled, check for shorts visually and electrically before mounting the device on the Arduino MEGA. Look for small balls and spikes of solder, which can come from the non-manual soldering processes. Check that the upright barrel jack works. Avoid powering the device from both the power jack and the Arduino USB at the same time, since it could have a slightly higher voltage than the USB supply.

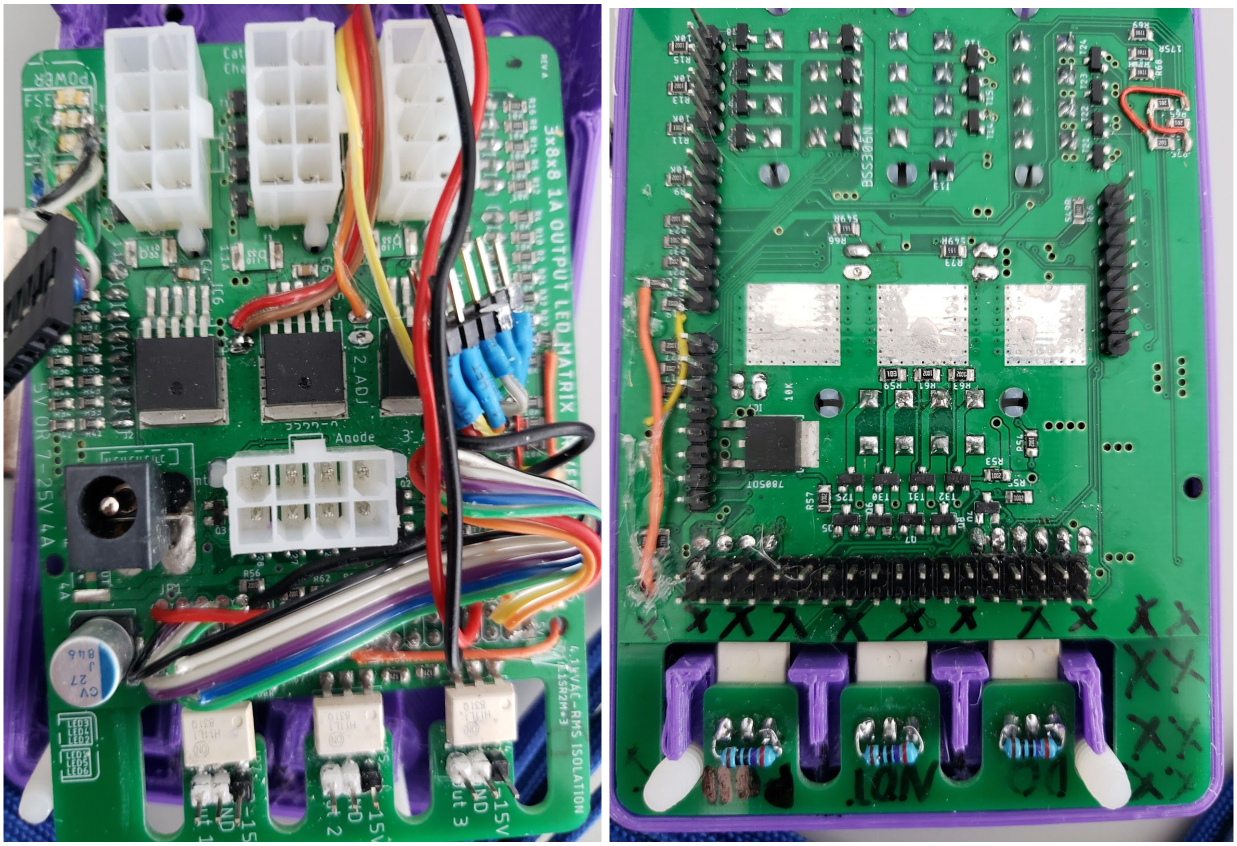
Step 4: Install and Test the Software

All Programming related files are in the 'Programs' folder. Using an Arduino-compatible IDE, upload the program. You'll need to use all the files included in the src folder. The software can be tested by opening a serial terminal, setting the baud rate to 115200, and waiting for a response.

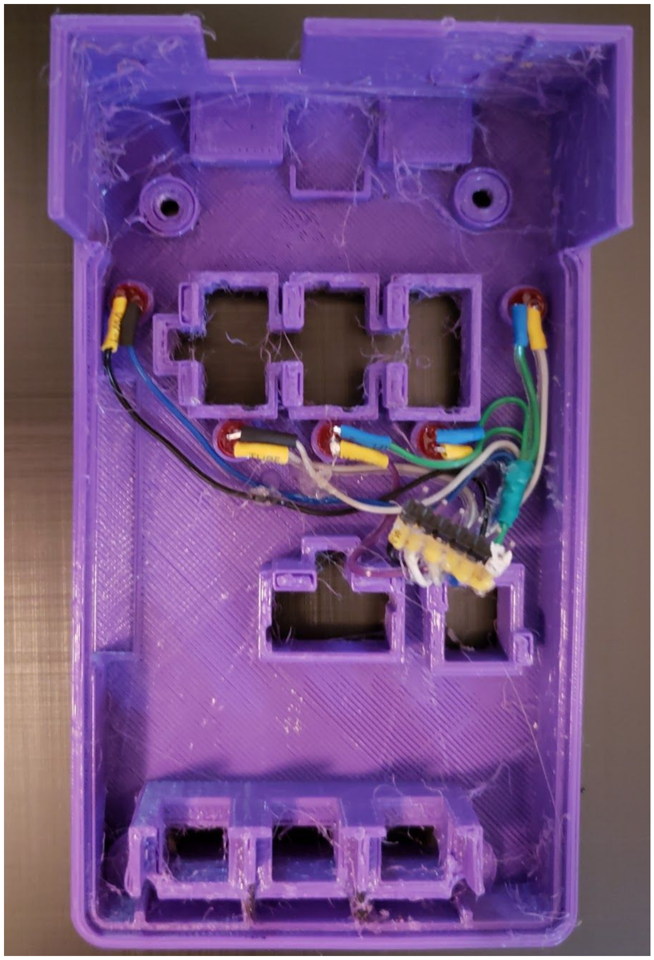
Step 5: Assemble in a Case

Once the software and hardware are working, the mainboard should be installed in a case. This can be a clear or opaque project box with holes cut in it, but we 3D printed ours.

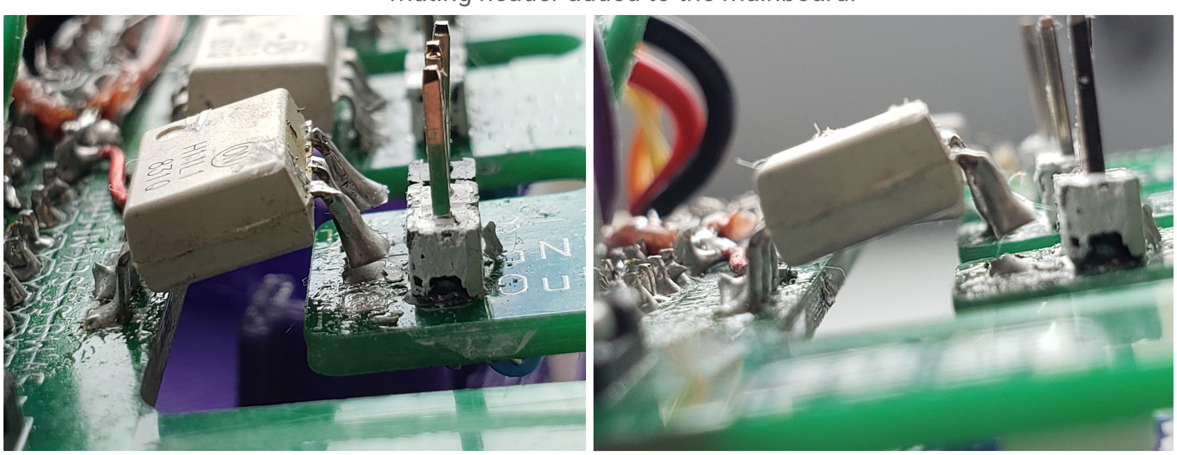
Our case has indicator LEDs for each of the three channels, LED voltage in, and logic voltage in. It also has holes for the upright barrel jack, the four 8-pin LED connectors, the Arduino's USB port, the VGA port, the adjustment potentiometers' connector, and the three camera shutter triggers. The Arduino barrel jack was <u>not</u> uncovered since it (and the USB port) should not be used for the LED power supply.



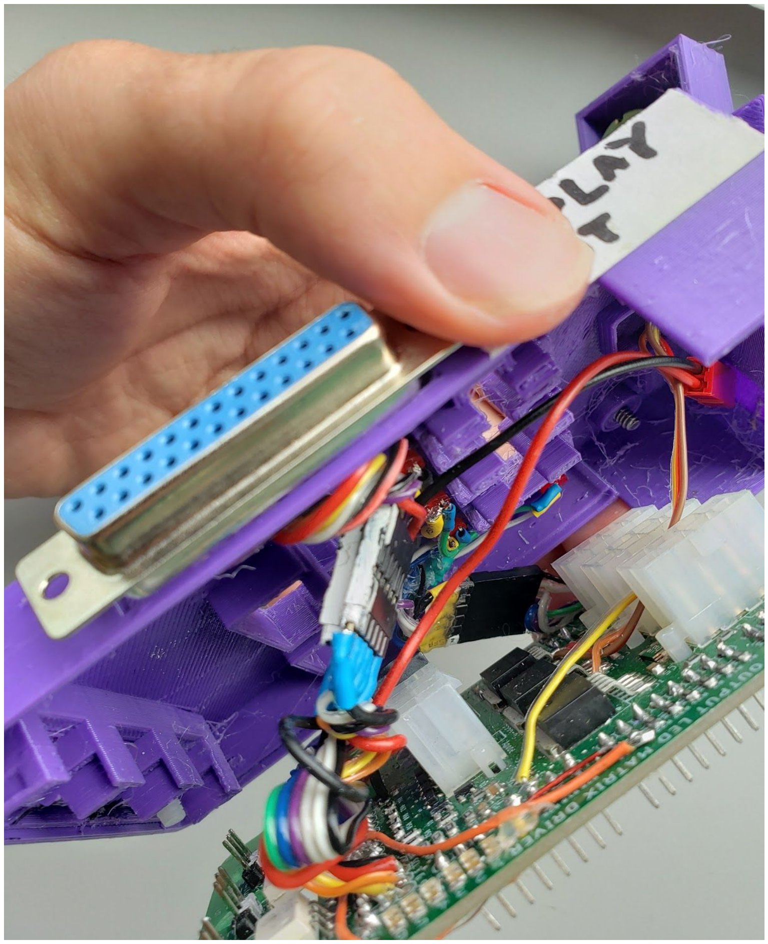
The final top and bottom sides of the mainboard. Some of the wires in these images are modifications that are already updated in the file. Any wires that go off the image or have headers at the end are connected to external parts. In the left image, the top left header connects to the LEDs in the case (shown in the image below,) and the two headers to the middle right connect to the VGA port for the display.

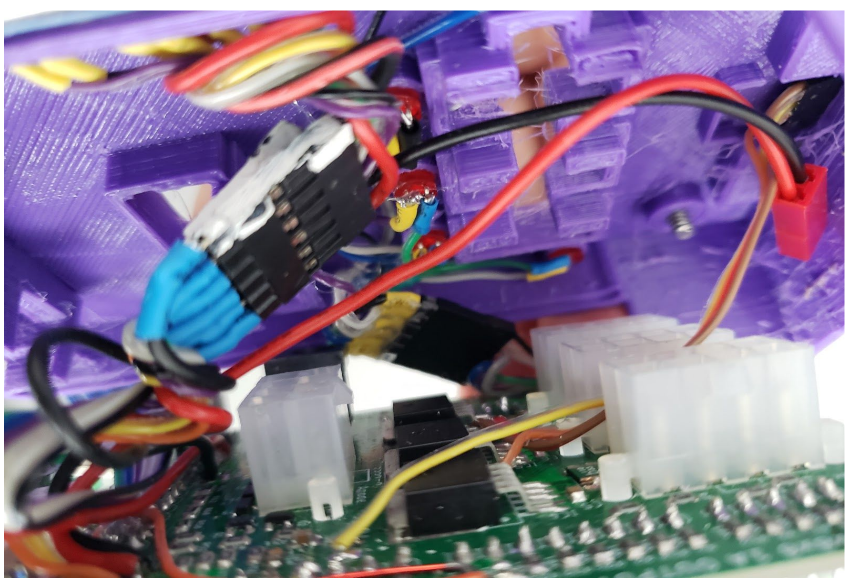


Top side with all the press-fit LEDs installed. A header is also there to connect and disconnect from a mating header added to the mainboard.



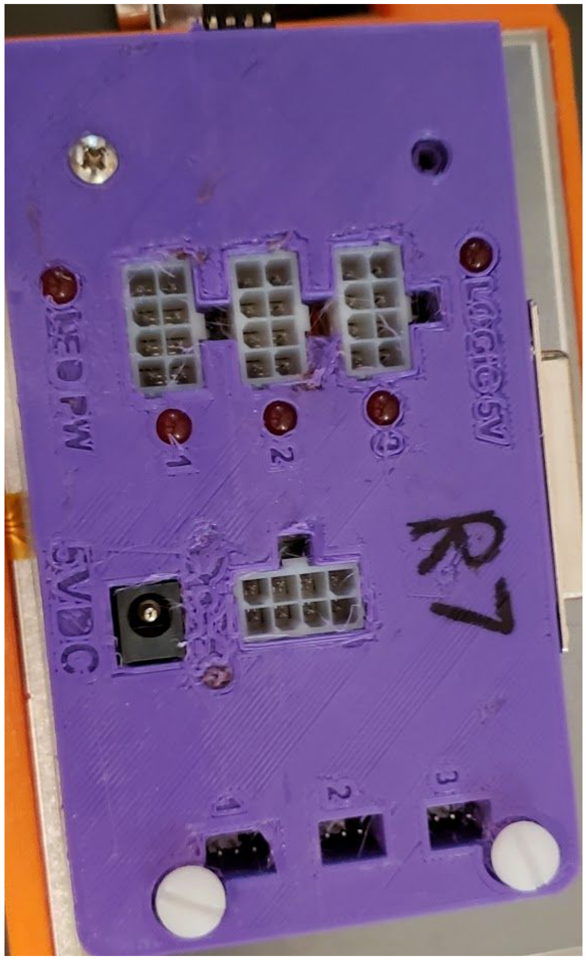
When installing the case, be careful not to push or pull at the end of the board with the optoisolators, since they can break and be bent off the board. The isolator pictured came off without any pressure, and for some reason would never stay on the board. The channel can still be used, however, because by changing a setting, all optoisolators can be made to trigger unanimously, regardless of their channel number. Since we are only connecting to these using relays, which provide more isolation, we can work around this problem by simply shorting across the signal pads with a wire (shorting pins 1 and 4). If that is done, however, the signal will be inverted, and the camera’s profile settings need to be adjusted accordingly on the device.



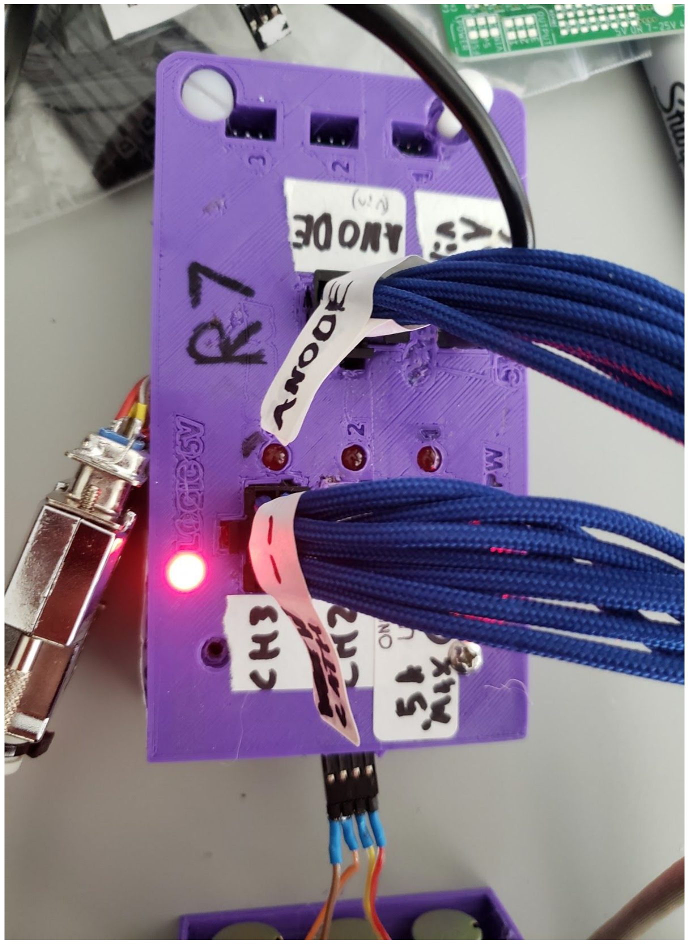


Two images of the connections made in the case during assembly. In these images some of the white-out used to color the headers is visible.

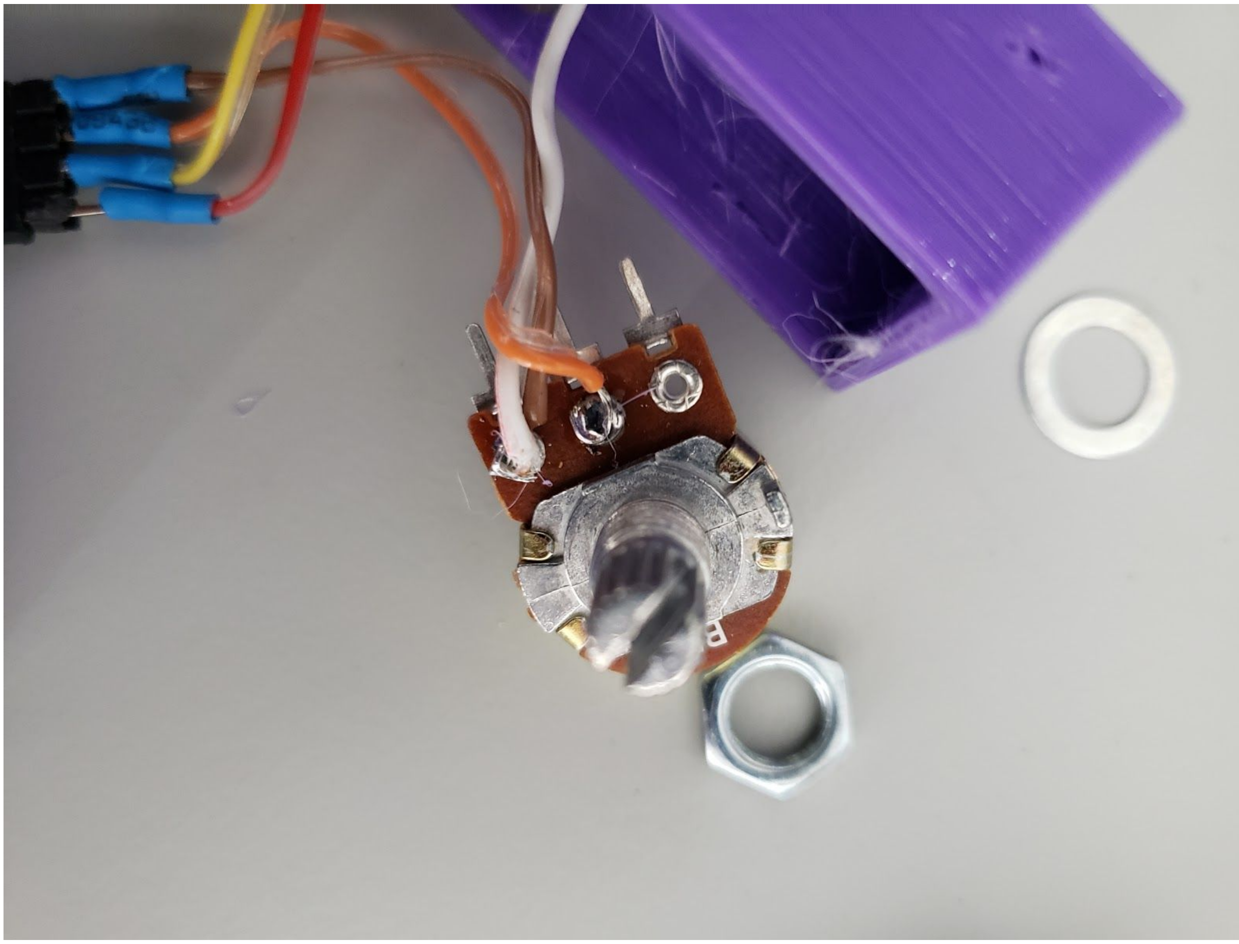
Once the top side of the case is attached fully to the circuit board, it should be nearly flush on all sides. This image also shows the isolation slots and dividers in the 3D-printed case.

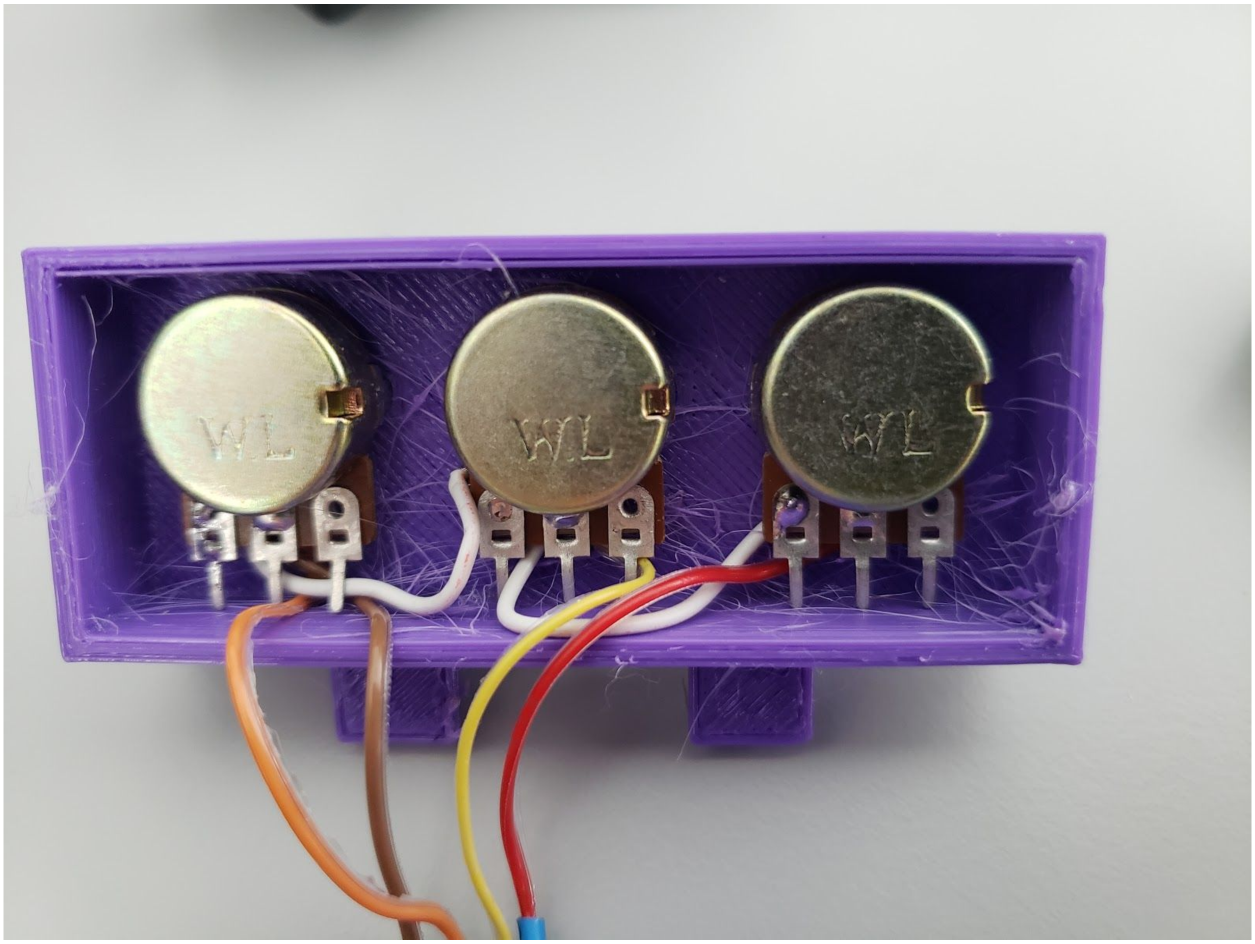


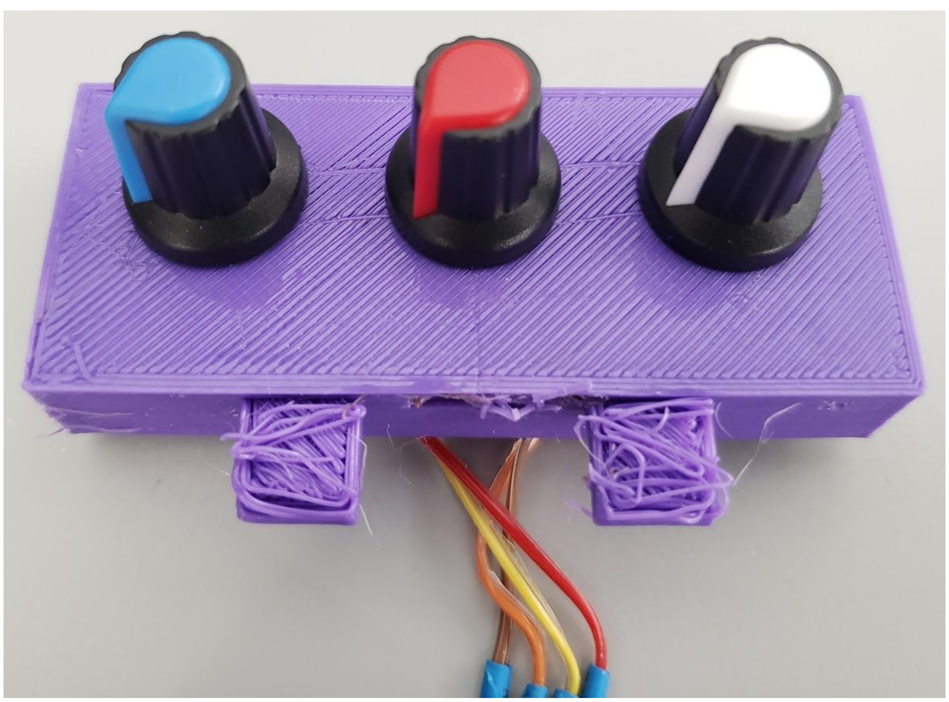
Top side with all electronics installed.



Top side again, but with wires plugged in and labels made. The first channel and UV cables had to be clearly marked so that they could not be mixed up with other channels since running the UV LEDs on another channel at high power could destroy them.







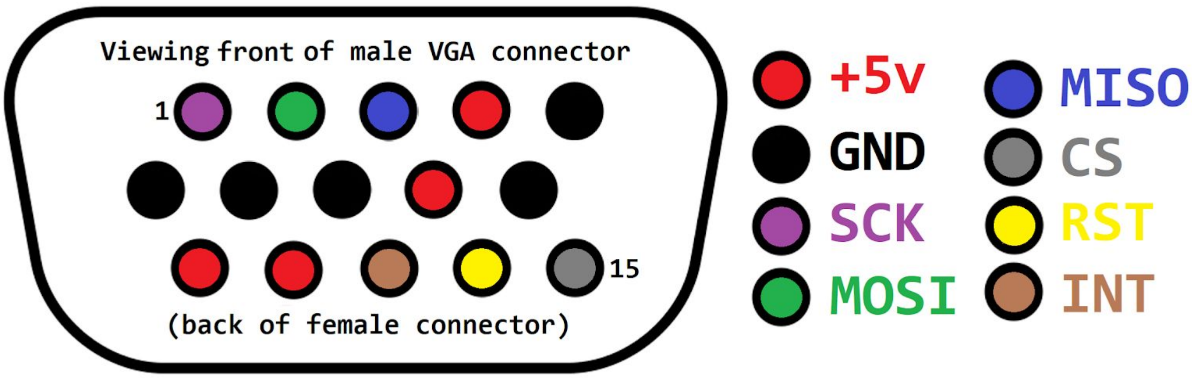
The adjustment potentiometers connect from ground to the current-setting resistor from the two-pin headers labeled 1_ADJ, 2_ADJ, and 3_ADJ.

Display

The display assembly is three different parts. The display itself is a 40-pin TFT resistive touch display, which was purchased from Adafruit (Product ID: 2354). That display is driven by Adafruit's RA8875 driver board (ID: 1590). These connect using the flat flex cable that comes with the display. The RA8875 is connected to the Arduino using a VGA cable, which is then connected to headers on the shield. A VGA cable is used because it has all the necessary connections, has three individually shielded signal wires, and has extra cables which can be used for power. <u>It is important to mark the device so that a regular VGA display is not connected to the mainboard, and so that a regular VGA output is not plugged into the display</u>.

Step 1: VGA Connections

The VGA male connectors should be wired as pictured (viewing the back of the connectors or the front of the cable):



- Red: +5v (Pins 4, 9, 11, 12)
- Black: Ground (Pins 5, 6, 7, 8, 10)
- Purple: SCK (Pin 1)
- Green: MOSI (Pin 2)
- Blue: MISO (Pin 3)
- Grey: CS (Pin 15)
- Yellow: RST (Pin 14)
- Beige: INT (Pin 13)

With these connections, a standard VGA cable can be used to connect the display to the mainboard. The graphic above is what the female connectors (which are used on the display and mainboard) will look like from the back, where wires are connected.

Step 2: The Display and the RA8875 Driver

The display comes with a 40-pin flat-flex connector, which connects directly to the RA8875 and has the display control and touch sensing connections built in. The RA8875 only requires a 5V power source to operate. The device communicates with the Arduino using an SPI interface.

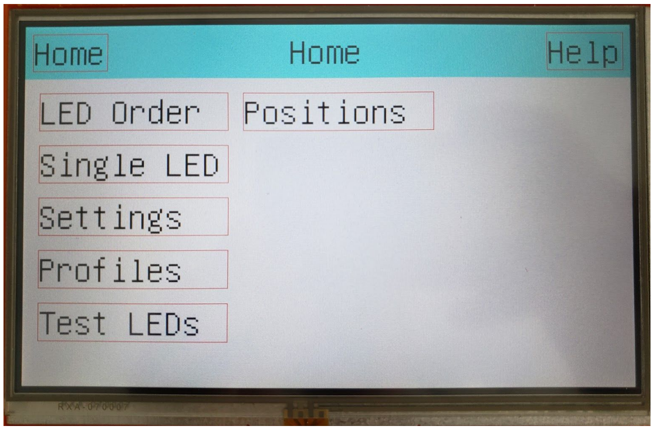
Flat flex cables can be very delicate, so be careful when handling the cable on the device. The connector on the RA8875 that mates with the flat flex cable is also delicate, since the part that pushes out (to allow the cable in) can be easily broken if not handled carefully.

Step 3: Putting it All Together

Connect the labeled pins on the RA8875 to the VGA connections with the same names. I did this using a DuPont connector between the two devices for serviceability, but the wires can also be directly soldered to each part.

Get the flat flex cable in place, metal fingers facing up, and push it in the connector. Push in the black part that extends out from the edge of the board to lock in the connector.

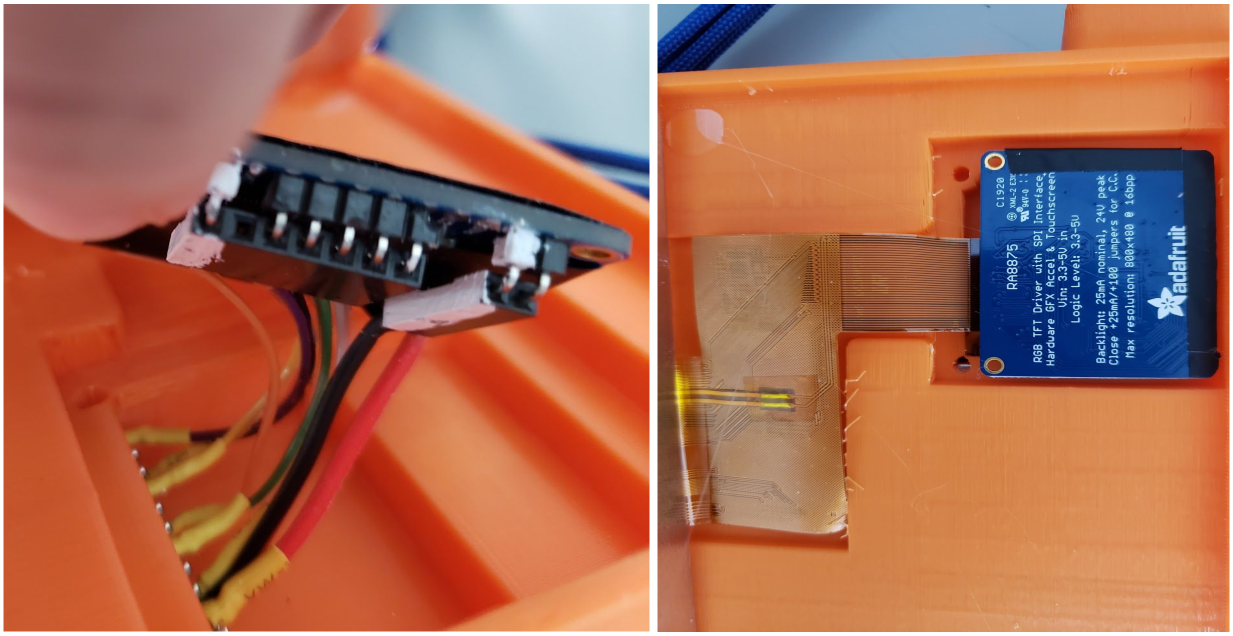
Before putting the display in a case, check that it boots up correctly. If it doesn't, check the connections before worrying about anything else (wiggling the cables around might reveal a loose connection.)



After giving some time for the board to perform a boot-up, this is what the display should look like.

Once everything is known to work, all the electronics should be placed in a case. Our display was assembled in a 3D printed enclosure, with a slot for the VGA connector. If that's not an option, a plastic project enclosure with manually cut holes and openings will do the same job.

For our enclosure, the pins on the RA8875 must be clipped with a flush side cutter and sanded down to be close to flush to the board (not too close - leave around 2-3mm of solder and pin.) All exposed metal (other than the screw holes) then needs to be taped over with electrical tape. This is all to avoid accidentally shorting pins out with the display's metal backing.



(Left) The connections from the RA8875 to the VGA connector. The RA8875 in this image is being held over the slot it's held in.
(Right) The RA8875 is placed into its slot, with the display's flat flex cable connected. After this, the display is simply pushed into place over the RA8875. The electrical tape covering metal pins on the RA8875 is also visible here.

Dome LEDs

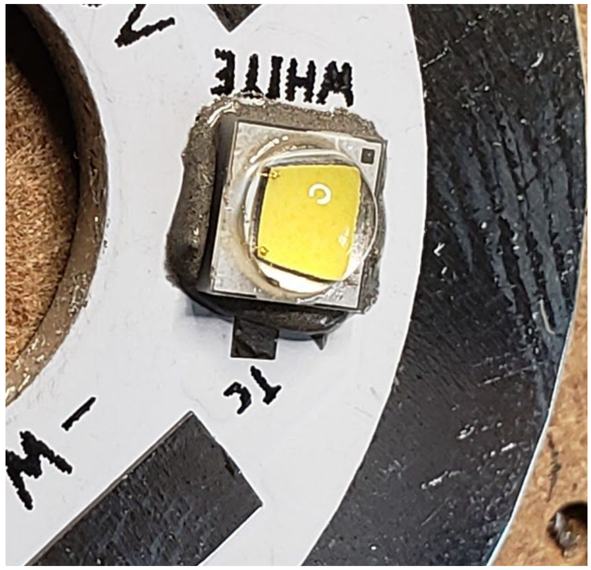
The LED boards for the dome assembly are aluminum boards, purchased from PCBWAY.com. They connect the common anode for the three LEDs and place those three on one board.

Step 1: Soldering

Because this board was designed without knowing the exact LED package we were going to purchase, only the anode is marked, not any specific side or corner, so check the datasheet of the LEDs if you're not sure about LED orientation. All anodes point inward.

Due to a mistake in ordering, these turned out to be colored white instead of matte black, which is what we intended to buy.

These were soldered using the electric skillet, but with manually applied solder paste. I found that I could run the boards in batches of 16 each.



An LED placed on the circuit board, not yet soldered, with a bit too much solder paste.

Two images of LEDs on the skillet; the top image is of test LEDs with standard boards and the bottom image is of the final custom LED board design. I found that I could fit 16 in one run.

Close up of LEDs on the skillet with a thermocouple.

When I soldered these, I had to pay special attention to the maximum temperature of the LEDs so as to not destroy any of them. I manually ran the skillet, stopping when all the solder was flowing.

I had one non-functional LED of the 84 White and UV LEDs, which was just placed 90° sideways, causing a short across the leads. Resoldering the LED rotated the right way fixed the problem.

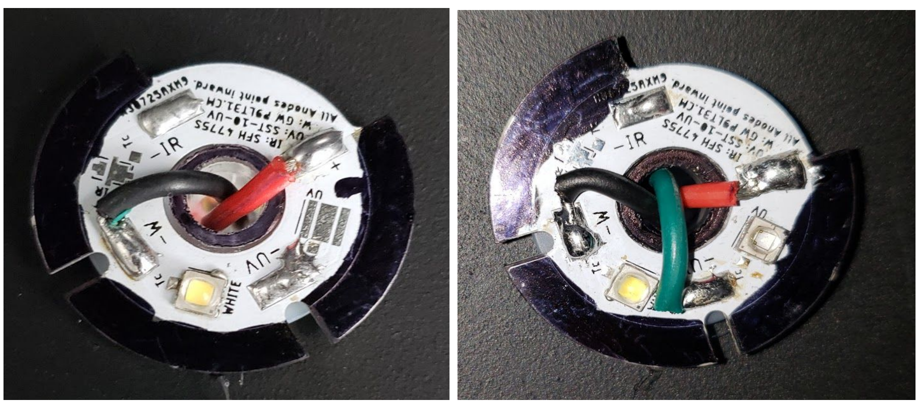
Like the mainboard, there will be some manual soldering involved. The circuit boards all need wires manually attached to them in order to work. The wires need to be soldered to the pads on the boards.

Step 2: Wires for the Dome

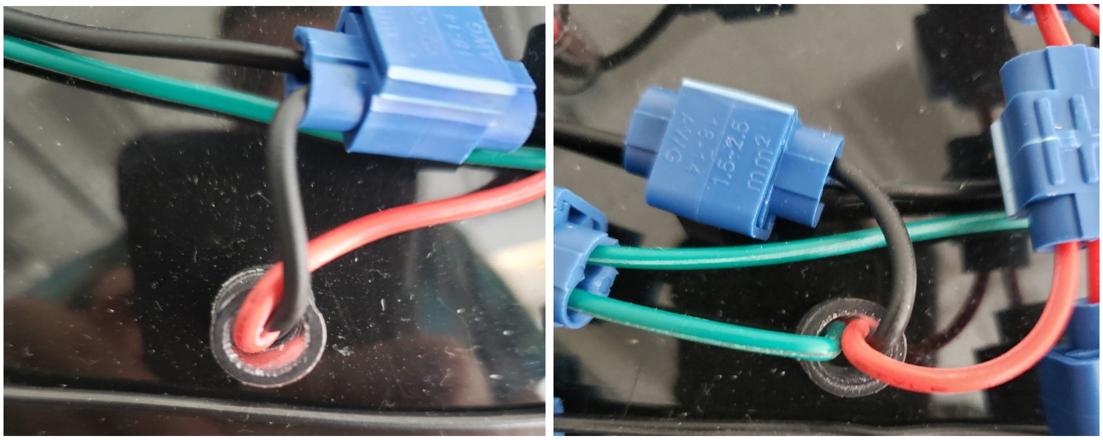
I cut all the wires we might need in case we expand the system in the future - 64 2.5” sections of each of the four colors of wire - then stripped and tinned one end of each wire. These wires were 18 AWG wires, but anything from 14-18 AWG will work for the inline splices we used. Bigger wires will be better, so 14 AWG will be better than 18 AWG since it will fit in the splices better and will waste less power (long, thin wires will waste power which can make LEDs darker on one end of the dome than on the other.)

64 of the wires after cutting and striping. This was one of the four colors of wire used since the different connections on the dome were color-coded.

With these aluminum circuit boards, there is a very high thermal mass, meaning that when using a soldering iron, it may be necessary to heat up the circuit board itself using a hot air station/gun, <u>without bringing the temperature beyond the safe range of the LED!</u> I ran my hot air station at 100ºC, which would heat the board to 80°C. After the circuit board heats up, it will be much easier to solder the wires without making a cold joint.



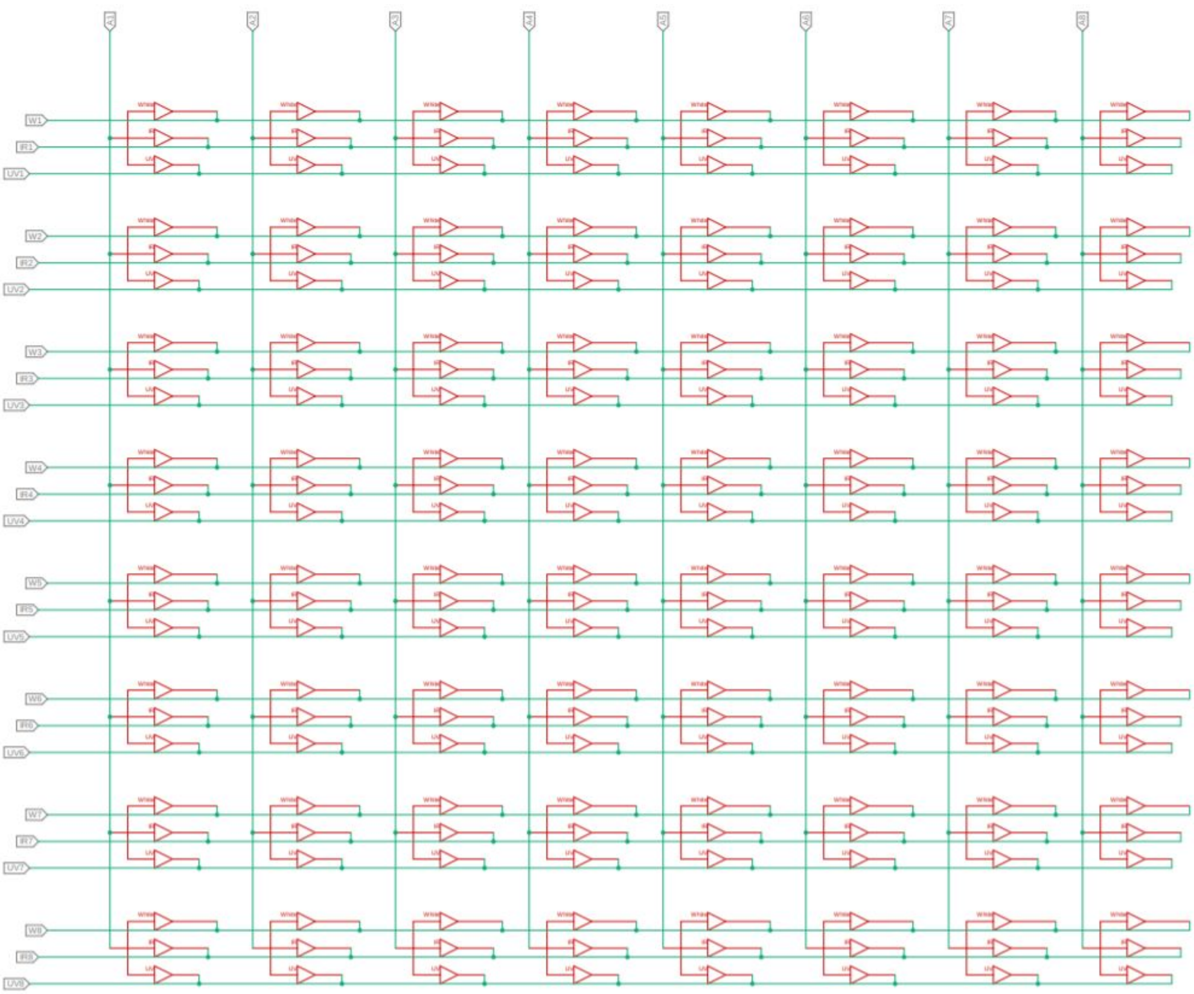
Our two different LED board wiring configurations. To the left is the white LED only board, with the common anode (red wire) and the white cathode (black wire). To the right is the white/UV LED configuration, with the common anode, white cathode, and UV cathode (green wire). Care needs to be taken to make sure the wires do not cover the LEDs (they are best off flush on the board). All the wires are run through the tubes to the outside of the dome. These boards were hot glued to plastic mounts so they could be modified in the future, but the hot glue can slightly melt if the LED is left on for too long.



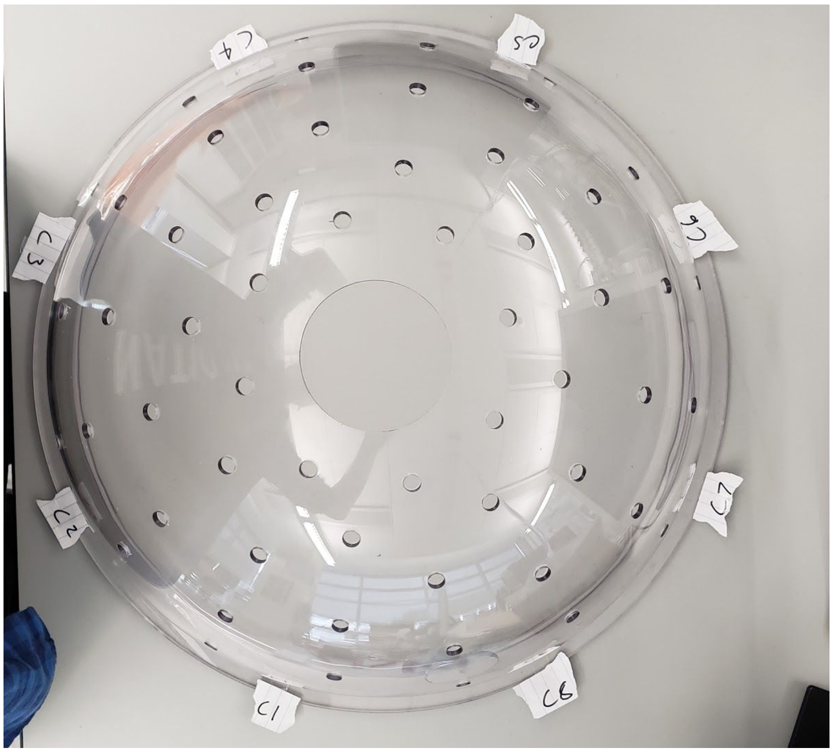
LED wires coming out of the dome and being connected to the power rail wires via inline splice taps (blue plastic parts). These LEDs are the same as the two configurations above.

Step 3: Wiring the Dome

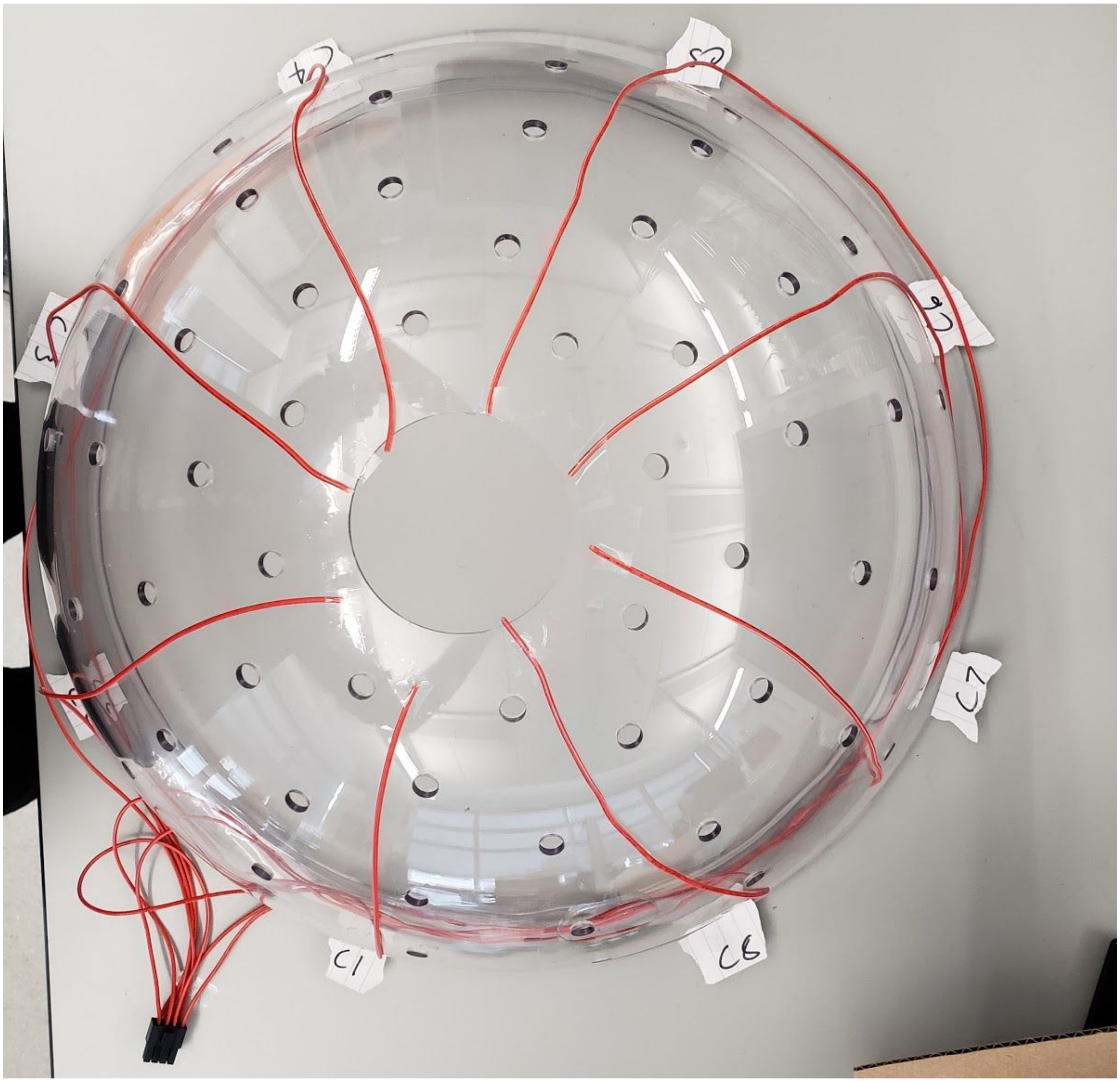
Our wiring scheme follows that of a normal 3-color LED matrix. The three colors are driven similarly to a common anode RGB LED, since any cell of three LEDs are all powered by the same anode wire, but are all controlled by their cathodes.



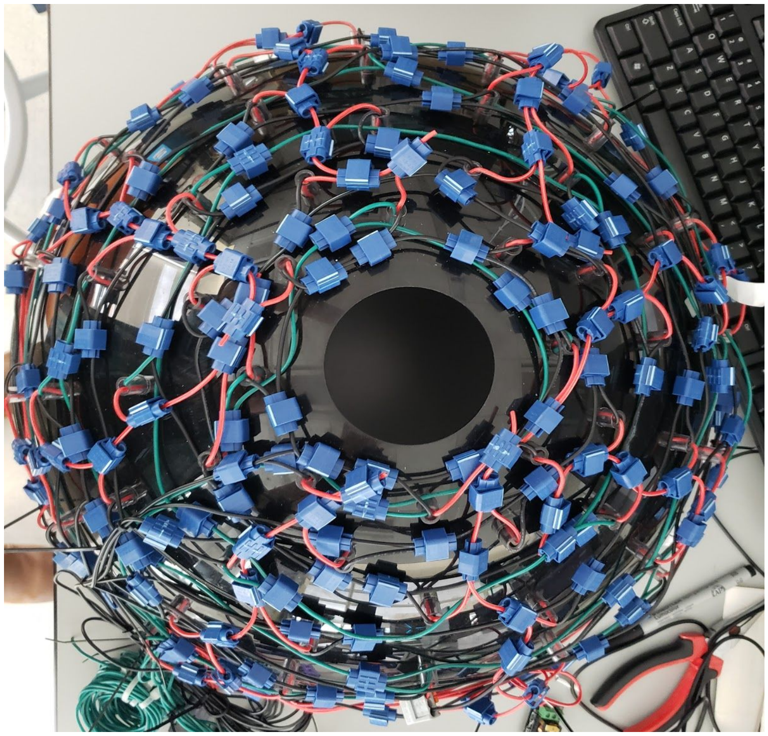
After the boards were done, the LEDs were installed in the dome by leading the wires through the tubes that were already attached to the dome. We then connected the wires of each LED module to the wires running on the outside of the dome. The anode wires were run last so that the more complicated wiring would be on top of the bundle of cabling. When I created this, I made the mistake of pre-measuring and cutting wires, then stripping and crimping them. It would have been much better if the wires were attached to the LED boards using the splices, then run around the dome, then cut, then finally crimped into the connectors.



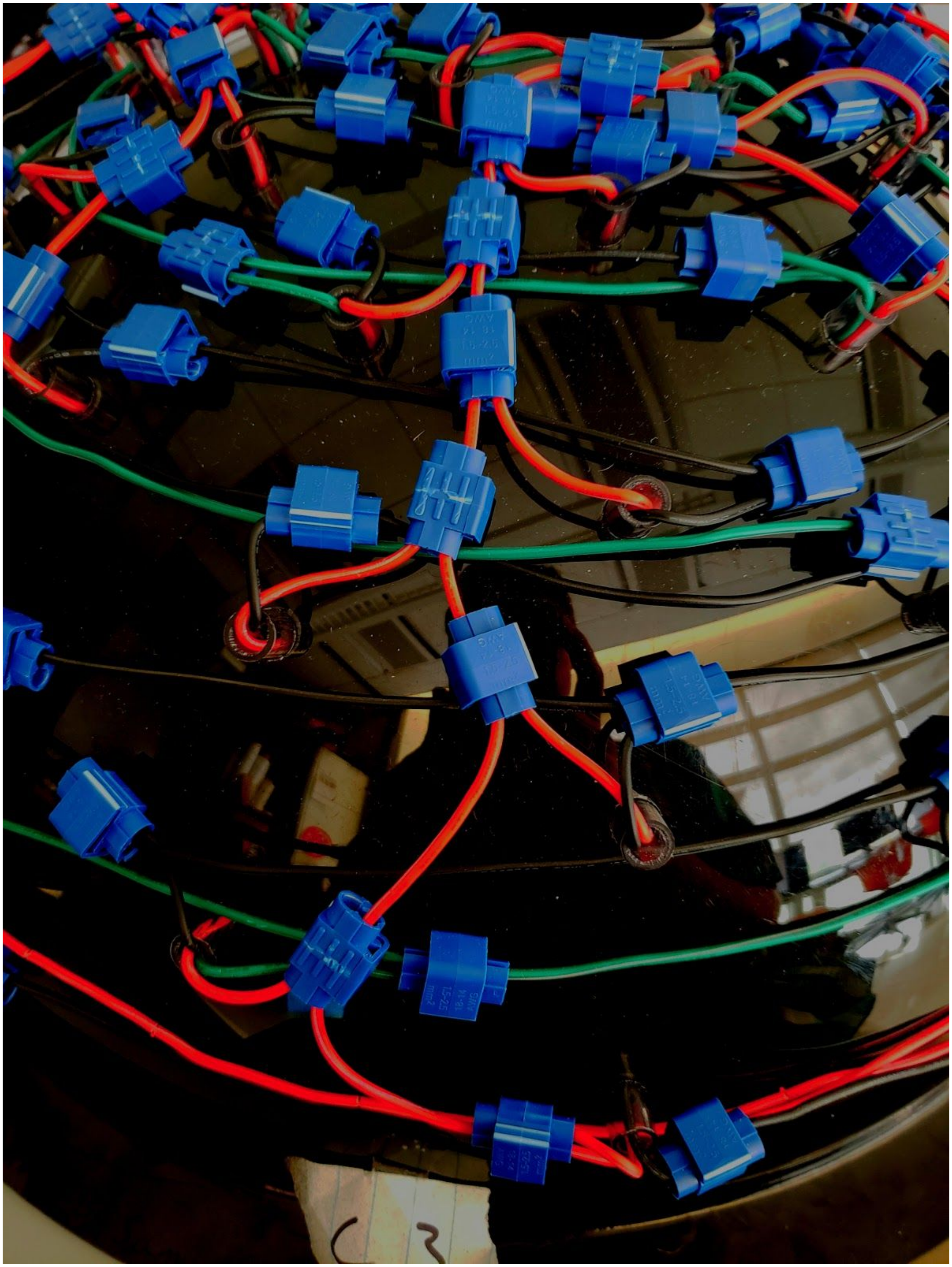
Columns on the dome are all labeled - one “column” according to the software is actually two of the staggered columns of holes on the dome itself. The labels are helpful in keeping the wires organized. I labeled between the two physical columns that would make up one logical column.



Wires were run to the columns, then were run up the dome itself. In this image I was using a pre-measuring method which I now know is a bad way to wire the dome. As I later learned, this was not enough slack in the wires that go up the dome, which need to bend to reach over to the holes at the side of each wire.



The dome, as viewed from above, when all wiring is finished.



Final wiring configuration. This image highlights the travel of the red wire (anode) going up one of the columns, connecting to all 8 LEDs in both physical 4-LED columns. The UV cathode (green) and white cathode (black) wires are also visible as they go perpendicularly around the dome in rings.