Trichrome Stipple Light (TSL)

Assembly Guide

Matthew Hamilton for Lab64

Part 0 – Orienting Yourself

The day of the workshop focuses on integration – basically putting everything together to make the final product. It's also about sharing skills! Here you will be soldering the guts of the stipple holder, assembling the potentiometers, and putting everything into the enclosure for a clean package. All the while, your SVG files will be laser-cut out of acrylic and will be ready during the workshop.

If you're doing this project outside of a workshop, these directions will guide you through soldering your connections, laser cutting your acrylic squares, and assembling the final product.

However, these directions don't function as an introduction to soldering, 3D printing, or laser cutting! That's where skill sharing comes in – if you need help along the way, reach out to a Lab64 Course Assistant to aid you in any of the techniques discussed here. You should also talk to anyone else that's working on this project. Experience is a great teacher and they'll be able to pass on tips and tricks not covered in this document.

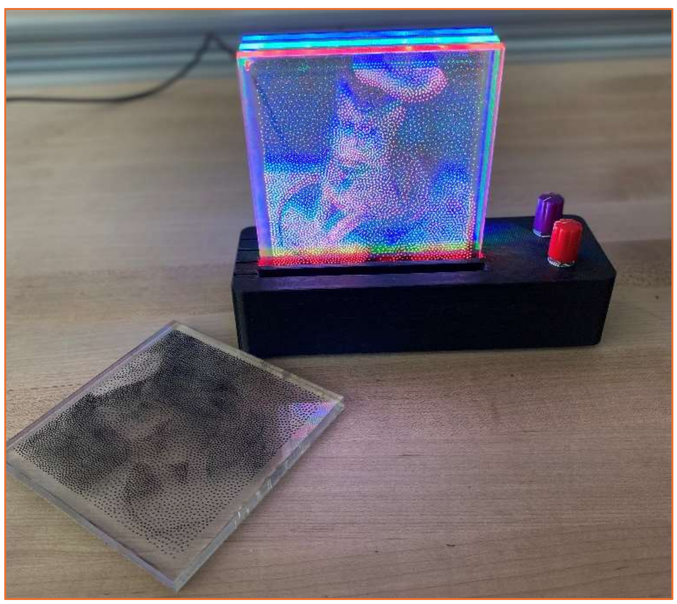


Figure 1: Completed TSL prototype with Black Channel Removed

You can follow these directions sequentially, but can also choose to start at Part 1, Part 2, or Part 4. You'll eventually hit every part on the way to completing this project.

Part 1 – Print your Stipple Holder

3D printing usually takes several hours, so it's a good idea to start this step early. You can solder and assemble the electronics portion ahead of time while your holder is being printed. Download the Stipple Holder on the Lab64 GitLab site and import it into the Prusa Slicer.

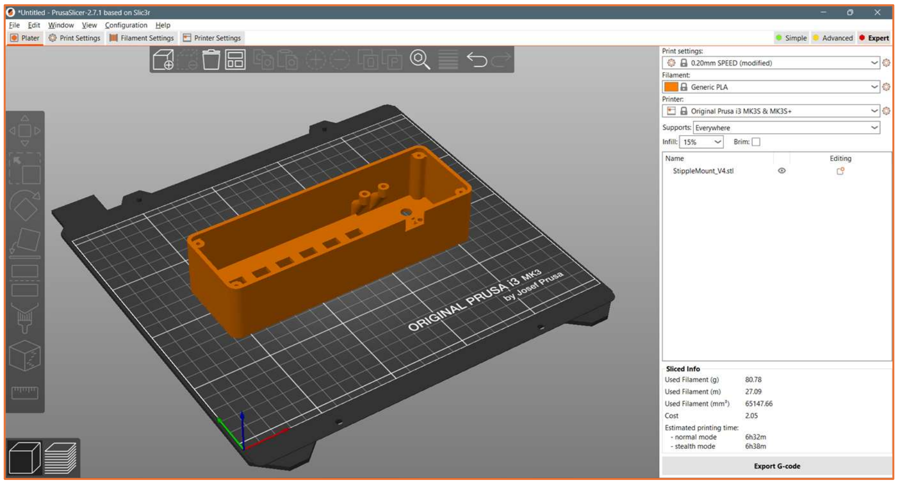


Figure 2: PrusaSlicer for the Stipple Holder

You'll want to print it upside-down to minimize wasteful supports inside the cavity. Keep the infill to 15% with supports everywhere. Make sure "Brim" is NOT selected, or you'll have a hard time removing supports and cleaning the edges of the box. You can use any detail and layer height that works for you. The screenshot above is just a recommendation.

Assuming your bed has no temperature issues, bed adhesion should not be a problem. In testing, we observed it takes between 8 and 11 hours to print a holder. As always, you can find the 3D model, directions, and code on our website at https://lab64.stanford.edu/.

1.1 Adding Heatset Inserts

You might have noticed that the underside of the Stipple Holder has 4 mounting holes in a rectangular pattern. These are sized for M2 4mm "Heatset Inserts" which couple with M2 screws and solidly secure the microcontroller to the holder. Since this is where you'll plug and unplug your completed TSL, a mechanically stiff mounting point is critical!

If you've never used a Heatset insert before, you're in for a treat. They're both an elegant way to mount components onto a 3D print (or another thermoplastic) and are immensely satisfying to use. You'll need a hot soldering iron, your 4 brass inserts, and a pair of tweezers. With your holder upside-down on a workbench, heat up your soldering iron. Position one insert so that the wider diameter is facing up. Use your tweezers to move the insert to one of the mounting holes in your stipple holder.

Next, make sure the soldering iron tip is completely clean and free of solder. Touch the tip of the soldering iron to the brass insert to heat it up. At the same time, use your tweezers to guide the insert into the mounting holes until the brass insert reaches the bottom, and the top of the insert is flush with the sides of the mounts. CAREFULLY reposition your tweezers to the top of the insert so that they're right next to the soldering iron tip. Remove the soldering iron tip and use the tweezers to keep the insert in place. This part is critical so that you don't rip out the brass insert on accident!

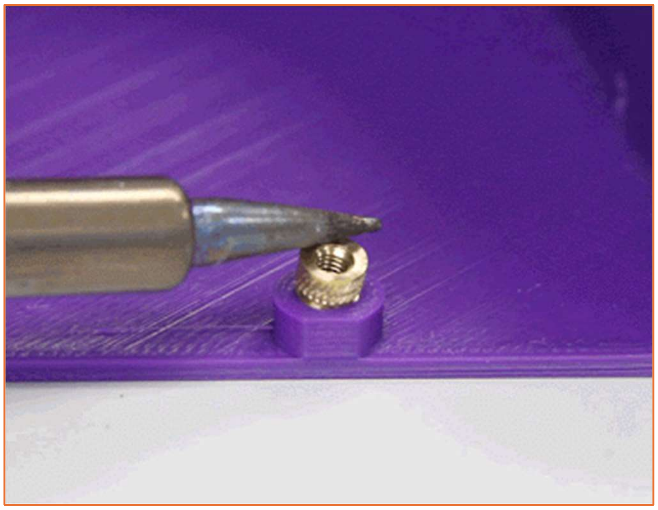


Figure 3: Animation of heatset insert process

Part 2 - Soldering

Soldering the connections for this project takes the most time out of the steps listed here. If you're not an experienced solderer (and even if you are) you might find some difficulty soldering the wire and LED strip together. This is because the LED strips are thin and metallic making them excellent heat conductors and heatsinks. This isn't great for soldering since we need to heat up the surface to ensure a good joint. Before starting off, here are some quick tips to think about before starting. We'll discuss them in detail in the following directions.

- Tin your tip! Tin your pads! Tin your wires! – Solder bonds to solder easier than to metal
- Heatshrink tubing is critical for ensuring a good mechanical connection
- Use clamps and holders to make soldering wires together easier
- Don't burn your fingers – yes, it can happen to you!
- Solder the backside of the LED strips
- Daisy-chain your connections

The wiring illustration in the last page of this document will orient you though the assembly. The TSL uses an Adafruit Metro Mini V2 as the microcontroller for this project. It uses the excellent atmega32 chip commonly found in Arduino microcontrollers. Since it uses 5 volts, it easily powers LED strips without issue and has more than enough analog inputs for our purposes. (It also means you can expand the TSL with many more functions if you want to.)

The illustration shows the red Vdd wires and black Ground wires "daisy chained" together, except in a few spots. This means that instead of every device being wired back individually to the same ground or Vdd pin creating a clumsy ball of wire and solder, these wires jump from one device to another to complete the connections.

2.1 Tinning the Wires

Start by cutting your wires to length and stripping the tips. You'll need enough wire to connect 2 potentiometers and 3 LED strips. You can do this all at once or as you go, but it helps to cut them ahead of time so that you can tin your wires efficiently. Consider the length of your stipple holder and give yourself some slack so the wires are not stretched when you assemble them later.

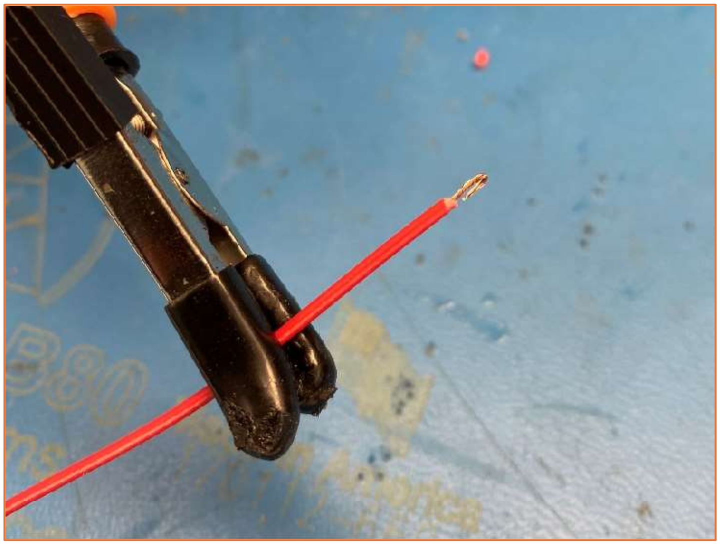


Figure 4: One wire ready for tinning, you'll want to do multiple at a time.

Next, tin all of your wires. Tinning means using your soldering iron to apply solder to the bare wire threads without connecting them to anything. This serves 2 purposes. First, it helps keep your wire strands together. Second, it makes it MUCH easier to bond them to other components. Liquid solder bonds more easily with solder than it does to cold metal like your wire or solder pad. This is an important trick, and it will make assembly easier and less frustrating.

It's more efficient to do this with several wires all clamped at the same time, rather than individually. Use a table vise or wire grips at your station to clasp several wire ends, but don't let them touch each other. Apply a small amount of solder to the length of stripped wire to tin them. Then, do the same with the other ends of the wires. This may seem tedious, but it will save you time in the next steps. Don't be tempted to hold the wires with your fingers! Copper conducts heat quickly and will easily burn you! Use a clamp or vise grip; it's safer that way.

2.2 LED Strips

With the wires tinned, turn your attention to the LED Strip pads. You'll tin these too, but this can be tricky. Clasp the LED strips in your vise grip and use a little flux to clean the copper pads. Make sure you tin the back side of the pads! Otherwise, the LED Strips won't lay flat into the stipple mounting holes. All wires going in and out of the LED strips should attach to the back side of the strip, including your signal wires. It's super important that these connections are snug with the pad and no extra length of wire is exposed. This is a common point of failure – make sure there's no chance that any of the wires can touch each other! You'll stabilize this connection later during assembly with hot glue.

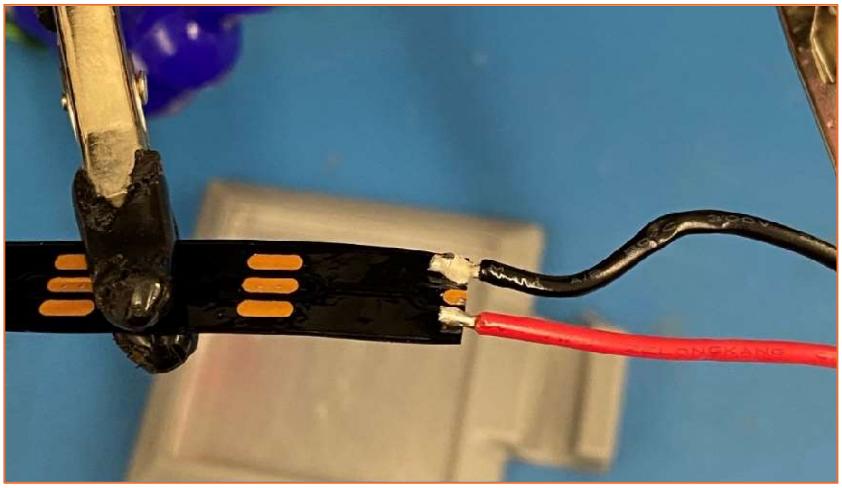


Figure 5: Power wires soldered to the LED strips. Note that these are on the back, and the wires insulation is snug with the solder pad

2.3 Soldering the Potentiometers

Now you can solder the wires to the potentiometers. Tinning the potentiometer leads can help here too. During the workshop, we noticed that some of the potentiometers are either defective or break during soldering. Try not to apply too much heat to the pot leads when soldering them to reduce this risk. Reference the wiring diagram on the last page and take care to slip on heat shrink tubing on your wires *before* you solder if you're soldering multiple wires. Forgetting to put on heat shrink tubing after you've made your solder joint is a rite of passage. Since you are daisy chaining the connections, there will be a few points where you'll slip heatshrink over 2 wires instead of just one!

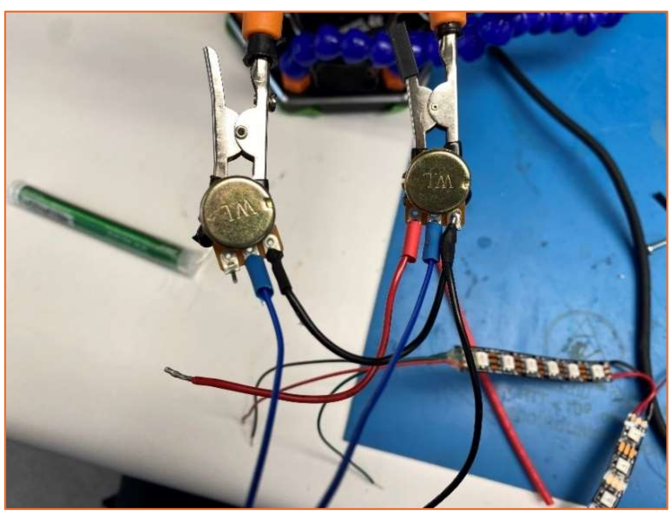


Figure 6: Wires going to the potentiometers. The "signal" wires are blue.

2.4 Header Pins

Last but not least, you'll need to add female header pin connectors to the ends of your signal wires going to the microcontroller. These will connect to the male header pins soldered onto the microcontroller and simplify assembly at the end. (On the wiring diagram, these are the little black rectangles pointing towards the microcontroller.) Secure this splice with a bit of heatshrink.

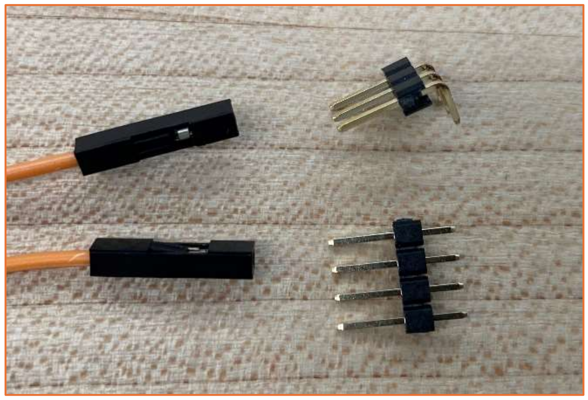


Figure 7: Female wire header connectors (left) with male header pins in straight and angled variety (right)

Turning to your micro controller, solder the male header pins to your board at the appropriate spots. The angled pins will connect to the LED signal outputs, and the straight ones will connect to power and analog inputs. This is an important detail for assembly later. Just solder the pins you need; don't use the whole strip or the micro won't fit into the stipple holder!

2.5 Final Steps and Checks

As you work through the solder connections, pay careful attention to the connections, and take your time. When you're finished, take a look at all of your solder joints to make sure they're mechanically stable. Could you add heat shrink anywhere? Is there any chance your wires could cross and short out? Are you sure you wired the ground pins to the ground wire, and the power wire to the power pins? Did you connect the LED strip signal wires to the Din side, and are your Dout pads unconnected? Have you had a snack? These soldering steps can take a good amount of time so keep your blood sugar up and don't get frustrated. If you're satisfied with your job, congratulations! This is a challenging soldering job that takes a good amount of skill. The next step is testing it with your microcontroller.

Part 3 - Testing your Circuit and Microcontroller

At this point, your LED strips, potentiometers, and microcontrollers should all be connected together but unattached to the stipple holder. This is a good chance to make sure the guts of this project work right so you can fix issues now. Grab your laptop, microcontroller, and a USB-C connector. You'll also need the following software:

- Adafruit MCU Drivers, to get the configuration of your MetroMini V2
- Arduino IDE, the development environment for Arduino-based microcontrollers.
- USB to Serial Drivers for connecting to your microcontroller on Windows (if you're using mac, refer to Adafruit's guide here
- StippLED Code from Lab64's Gitlab.

The software above is everything you need for testing, connecting to your microcontroller, and uploading the TSL code. If you have issues with this step, follow the manufacturer's instructions on their website at learn.adafruit.com. Start by downloading and installing all the software listed above.

3.1 Connecting to the MCU

With the Arduino IDE page open, connect your computer to the microcontroller using a USB-C cable. Remember, don't connect any of the electronics in the previous step to your board!

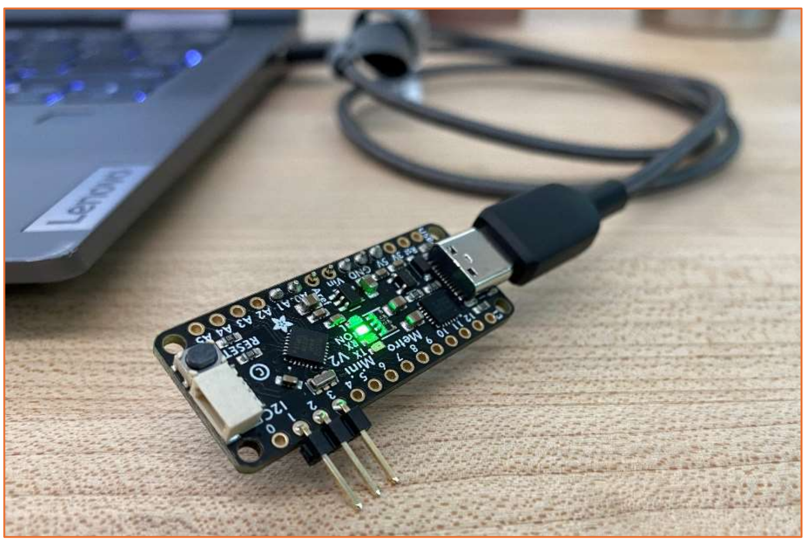


Figure 8: MetroMini V2 connected to a PC. Note there's no other electronic attached to this yet!

If you haven't already, add the MetroMini v2 to your board manager in the Arduino IDE settings. Reference the screenshot below to connect to your board, but keep in mind that your serial port number will be different. It's typically a number higher than 6. For me, it was COM23 Serial Port (USB). Notably, the serial port number changes every time you reconnect your metro so you may have set the com port again if you disconnect the microcontroller. With your board connected, you can now download the StippLED code and copy it into your board's .ino file. You'll also need to install the "fastLED" library from the IDE's Library Manager, as well as Arduino AVR Boards from the

Board Manager. Otherwise, you'll get an exit error message from the serial terminal whenever you try to upload.

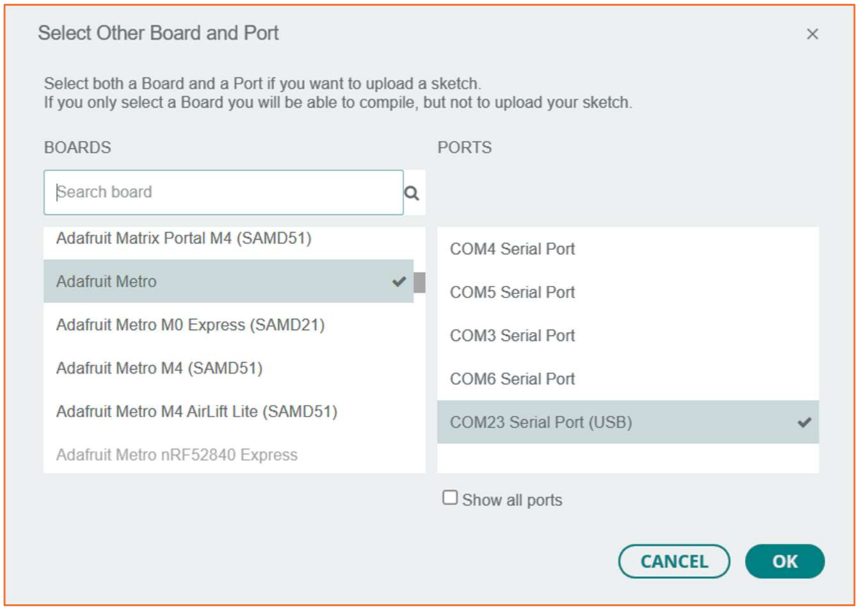


Figure 9: Arduino IDE Boards Manager (under Tools > Board > Boards Manager)

It's important not to look at the code for too long – it's probably not up to any CS Major's standards (from Stanford or anywhere else). This is, however, the best code you can expect from an EE Major who learned Arduino in a few weeks. If you're comfortable with Arduino this project is an excellent opportunity to develop your own firmware with custom LED animations (once you've assembled everything). You're encouraged to add your own code and share it with everyone else, so that they too can judge your code :)

Now, upload the code using the arrow button. If your code uploaded successfully without error, then your IDE is configured correctly. If not, you might have issues with the board manager, libraries, serial drivers, or MCU drivers. Go back and check these before proceeding to the next step.

3.2 Testing the Electronics

Bring over your mess of LED strips and wires to your microcontroller. Unplug your microcontroller from your computer so that it's completely powered off. While looking at the schematic diagram, connect the female header sockets to the male header pins on your microcontroller. Take special care in ensuring that you're not shorting any connections accidently. Then reconnect the microcontroller to your computer. The micro should immediately begin illuminating the LEDs depending on the position of the potentiometers. Turn both POTs and see if it's working as expected. One POT will shift through color schemes, and the other will vary the intensity of the lights.

3.3 Troubleshooting

If your lights are not working as expected but you uploaded code in the last step, the issue is with your circuit. A short in your wiring can cause issues compiling code to your microcontroller! This is why we tried uploading the code to the micro before testing the circuitry. Here are common issues we found during the workshop.

1. **LED strips are flashing and/or sensitive to small adjustments**: *Check the wiring at the LED strip pads and make sure there are no shorts. Are your wires soldered closely to the pads like in figure 4?*
2. **Some strips are not illuminating:** *Check that your signal wires are connected to the Dn (or Din) pads, and the Do (Dout) pads are not connected.*
3. **Turning the POTs does nothing:** *You might have a bad potentiometer causing an open or short condition inside the canister. Test the resistance value using a multimeter. Also check that the signals inputs are connected to AD1 and AD2.*
4. **Trying to upload the code fails and nothing works:** *You likely have a short between 5v and ground pins somewhere in your wiring resulting in insufficient power to run your microcontroller. Trace these connections and find the fault.*
5. **Code uploaded successfully, but nothing works:** *Check the power and ground connections and ensure they are soldered properly. It's also possible that your POT inputs and signal outputs are shorted to ground or 5v preventing a signal from reaching your LED strips.*

Once you've worked out all of the bugs in your circuit and it works as expected, you can move on to another part.

Part 4 - Acrylic Laser Cutting and Finishes

You'll need laser cutter training to do this part alone or ask a CA for help. We used the following settings to cut the acrylic from the templates.

Engraving: Power: 80, Speed: 30, dpi: 333 to 500
Cutting: Power 70, Speed: 0.55, passes: 2

Cutting 4 acrylic squares at the same time can take around 30 minutes as the laser head rasters over the sheet. After they're done, you can start putting the finishing touches on the 4 color channels. They'll still be dusty from the cut, so the first step is to wash them vigorously in the sink.

4.1 Preparing and Painting

Take them over to the reflow room and use the lab sink to wash out the acrylic dust. The dust tends to get stuck in the stipple dots, so wash them until the acrylic appears clear under the water. You can also remove the plastic film from the blanks at this point. Dry them off using a microfiber towel. Acrylic can scratch easily, so be careful and don't use any hard tools! If you're a perfectionist, you can look at the stipples under the microscope in our lab.

Now find your **black** acrylic square. If you labeled it, this should be easy. If not, then you'll notice that one of the images has a notably different pattern than the color channels and you can hopefully find it this way. In this step, we're going to add pigment to the stipples so that they blend and add darkness in the resulting image. This isn't possible with RGB light alone, after all. Remember – we're only doing this with the black channel! Not the other color channels!

Over the sink (please), use the tube of black acrylic paint and add several drops to the stippled surface of your plate like this:

Figure 10: Black paint on the black channel

Then using the paint brushes, dab the black paint into all the little stipples. Do this several times to really get the paint in there. It can help to use a paper towel and circular movements to better put the paint into the stipples.

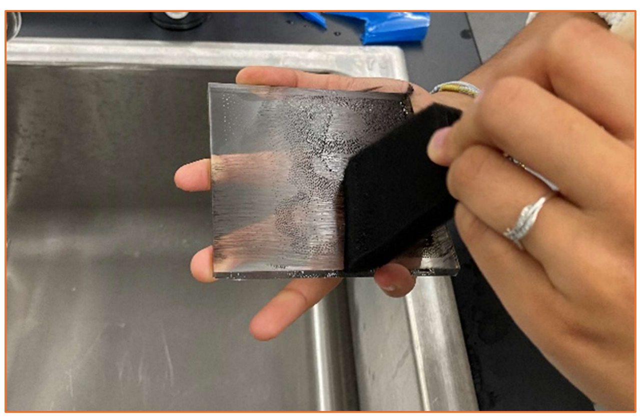


Figure 11: Brushing the black paint on

With enough work, you'll see the black paint covering the entire square. Then, use a damp paper towel to wipe the excess paint off and clean up the square. Take a look at how the acrylic paint has filled the dots. If you're not satisfied, repeat this process until you are.

4.1 For Overachievers

If you *really* want to make sure all the stipples are filled you can take the extra step of evacuating the air in the acrylic paint. You can use the small vacuum chamber along with the vacuum pump to pull all of the air out of the paint.

Apply a thick coat of paint on the stipples, then place it in a plastic container that will fit into the vacuum chamber. Place both at the bottom of the chamber then place the lid on top being sure to center it exactly. Pull a vacuum in the chamber until the pressure has stabalized. You'll see all of the air leave the paint and dry out a little bit. Leave it for about 1 minute, then return pressure to the vessel. Take out the acrylic and wipe off the surface. This is the best way to ensure the paint has entered every single stipple.

4.2 Optional Step:

We're not sure how good this effect looks, but you can add a coat of matte clear coat on the black mask to "enhance the blending effect". It makes the image look better upclose (maybe), but isn't necessary to get the full effect. This is an asthetic choice.

Figure 12: Spray lightly from about 1 foot away to apply an even coat

Part 5 - Putting It All Together

You have your clean acrylic squares and they're ready to go. You've tested your spaghetti mess of wires and LED strips with your microcontroller and it all works great. Your Stipple Holder is clean and the heatset inserts have been heated, set, and insterted. It's time to put it all together.

5.1 Hot glue for the LED Strips

Begin by carefully test fitting the led strips into the holder. You'll notice that there's a gap for every led strip. Here, you'll use a little hot glue along the edges and "bridges" of the slots to hold your LED strips in place. This can be messy but it works well.

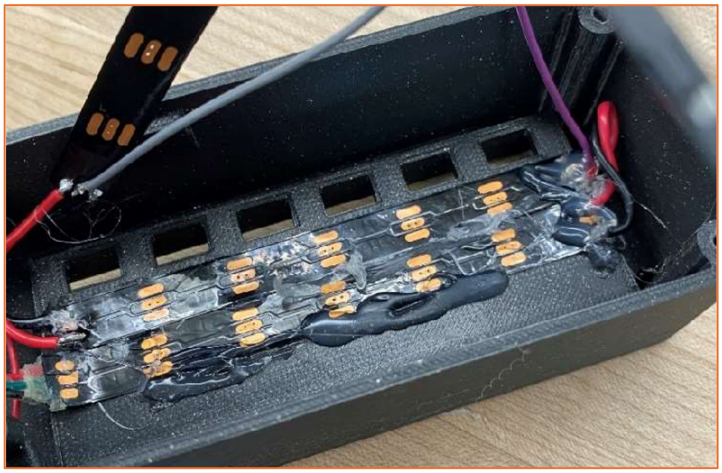


Figure 13: 2 out of 3 hot-glued LED strips. Glue on the solder pads also help prevent shorts during assembly

Next insert your potentiometers. These don't need hot glue and instead secure with a washer and nut on the top face. When that's done, add the knobs of your choice to the two pot shafts. These typically secure with pressure alone, or with a small hexscrew on the side of the knob which you'll adjust with a allen wrench (hex key).

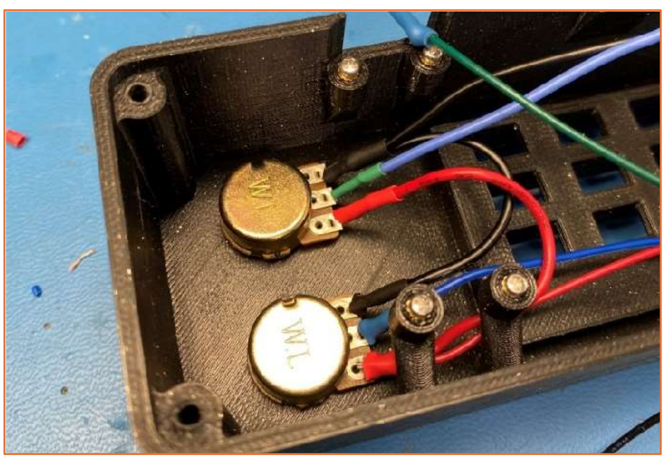


Figure 14: Pots inserted into the stipple holder

5.2 Microcontroller and connections

Next, secure your microcontroller to the 4 inserts you secured in part 1 with 4 small 2M screws. Because this microcontroller is also the main plug for your stipple holder, the heat set inserts provide a mechanically stable and secure point for attaching and detaching the USB-C cable for power. With this installed you can now attach the female header pins to their respective pins on the microcontroller to complete your circuit. If you soldered the male header pins correctly, you'll notice that the angled pins point conveniently towards the LED strips making this connection easy. At this point you're almost done! Use a little extra hot glue (if you want) to secure the loose wires to the stipple holder.

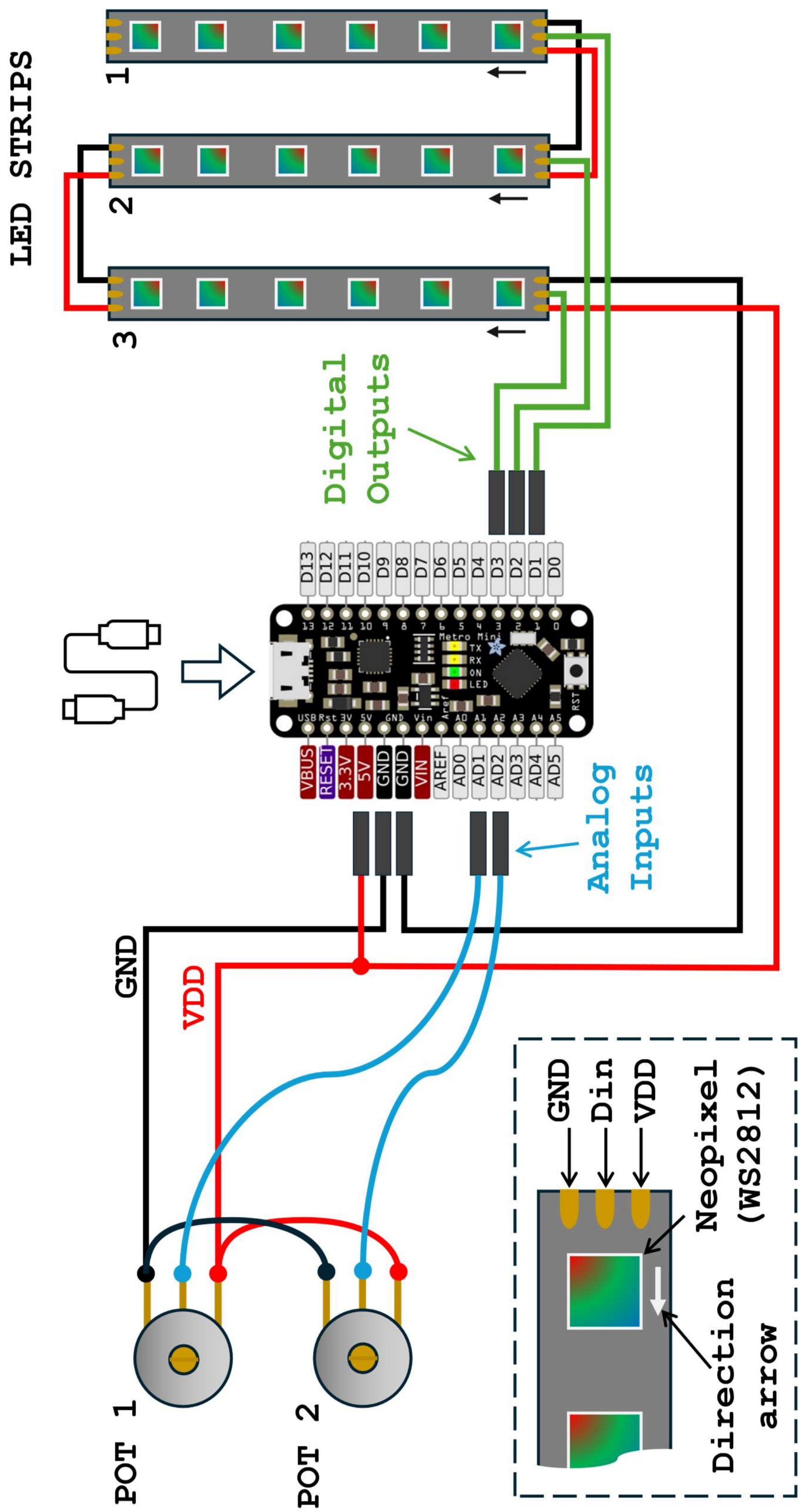
5.3 Securing the Acrylic Backing

Now with all of your wires connected and secure you can close it up using the provided acrylic backing. If you can't find one of these in the workshop supplies, you can cut one yourself using the SVG file on our Gitlab page. The four holes of the backing secure easily to the stipple holder using 4 #6-19 screws 5/8" in length (similar to M3.5 screws). If you can't find these, you can use any self-threading screws sutable for plastic. All this means is that these particular screws press into the surrounding plastic along the shaft to create threads. Speaking of which, be very careful screwing in these screws for the first time! Place your stipple holder upside down on a steady surface and use a good amount of pressure to put these in. Try this without the acrylic back`ing to make the threads first, then rescrew them with the backing in place.

Finally, stick 4 small rubber pads to the backing to give your stipple holder some legs to stand proudly on your coffee table, luxury Stanford dorm room furniture, night stand, or any other surface near an outlet. When you add your 4 color channels and plug it in, you should see the final result!

Figure 15: Final project using the example image in this document. The Stipple Holder is from an earlier revision. Knobs are not installed.

Congratulations on completing this project. The TSL was designed to give makers a more challenging project that would use a wide range of skills. Be sure to pass on what you learned to another maker and recommend this project to someone else!



LED STRIPS
1
2
3
Digital Outputs
Analog Inputs
D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0
VBUS RESET 3.3V 5V GND GND VIN AREF AD0 AD1 AD2 AD3 AD4 AD5
GND
VDD
POT 1
POT 2
GND
Din
VDD
Neopixel (WS2812)
Direction arrow