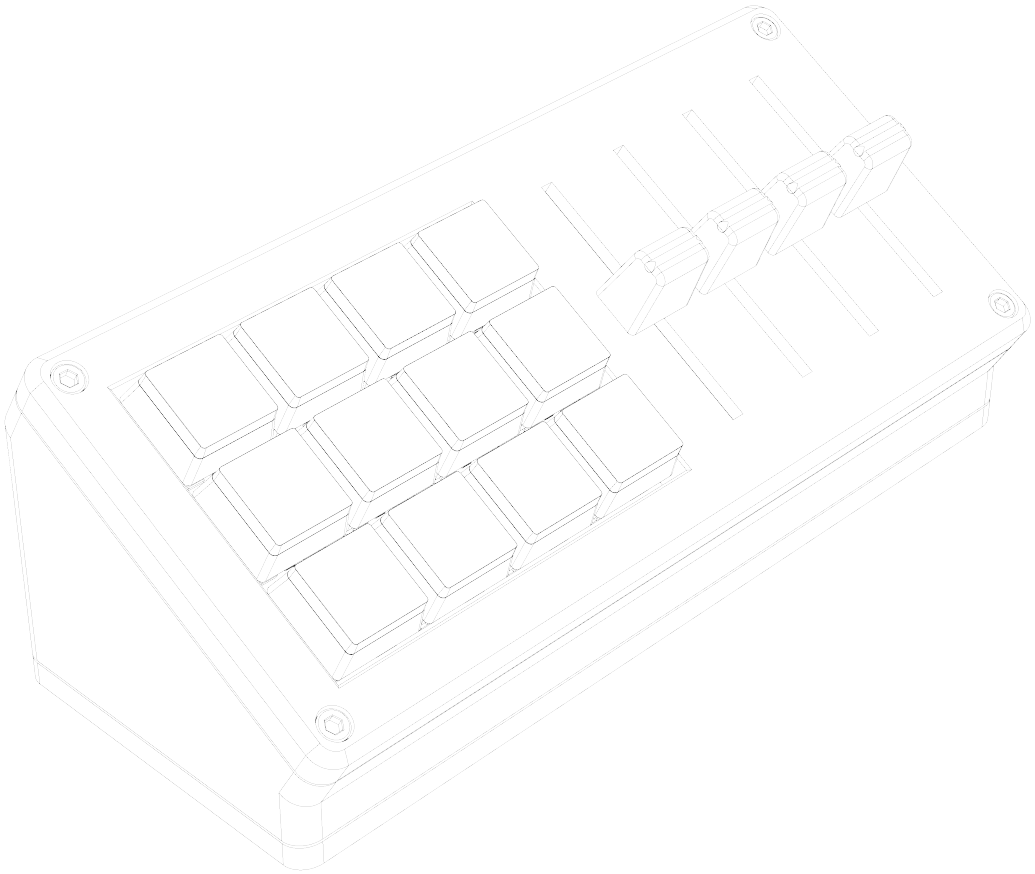


MisteRdeck

Assembly Guide

Introduction

Thank you for downloading MisteRdeck, the Arduino-based MIDI stream deck. These instructions will guide you through the process of making and using the device.

I created the deck for my own use, a few weeks before I started streaming. After fiddling with OBS for a few days I knew that a hardware device for scene switching and volume adjustment was going to prove incredibly useful. But herein lies the problem; all of the “cheaper” commercially available products lacked any kind of volume adjustment hardware. The cheapest deck, manufactured by a well-known company (I won’t name them) costs almost £100 and is essentially a box with 6 buttons. I knew I could produce something more functional at a lower cost, so I got to work.

Hours of designing and prototyping later and the MisteRdeck was born.

Massive thanks to /r/3DPrinting for the overwhelmingly positive comments, as well as the FB group Prusa Community Forum. If it weren’t for the sheer amount of interest shown from these two communities I would never have published this thing, selfishly keeping it for myself!

I also need to offer my thanks to the following;
Pieter P for the Control-Surface library
Everyone who contributed to the MIDIUSB library
Chris Yarger for the OBS-MIDI plugin
nightfox939 for the general aesthetic inspiration and genius pot-clip design

This manual is broken down into the following sections;

- Slicer settings & printing
- Stage 1 assembly (switches and pots)
- Soldering/wiring
- Stage 2 assembly (case)
- Arduino configuration
- OBS setup

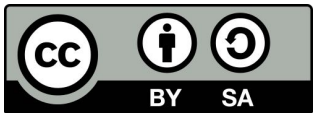
What’s required?

Basic soldering knowledge
The ability to follow instructions
A little common sense ;)

Tools

M3 hex wrench
Soldering iron + solder
Superglue/strong double-sided tape
Tweezers (super helpful for positioning wires)

License



With that out of the way, let’s crack on…

Bill of Materials

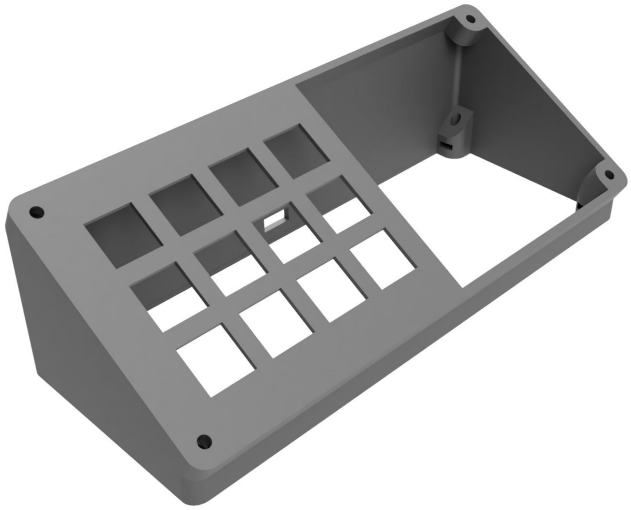
Item	Qty	Link *
10k Linear Slide Potentiometer (45mm travel, dimensions 60mm x 9mm)	4	cpc.farnell.com
Cherry MX/Compatible Switches	12	mechboards.co.uk
Cherry MX/Compatible keycaps (relegendable) **	12	ceratech.co.uk
1N4148 Diodes	12	amazon.co.uk
Hookup wire	-	amazon.co.uk
Arduino Pro Micro (clone)	1	amazon.co.uk
M3x10mm Cap Head Screws	8	accu.co.uk
M3 Square Nuts (max 1.8mm thickness) ***	8	accu.co.uk
M3 Heat-Set Threaded Inserts (4.6mm diameter) ***	8	amazon.co.uk

* Links are for example purposes, you may find them cheaper elsewhere.

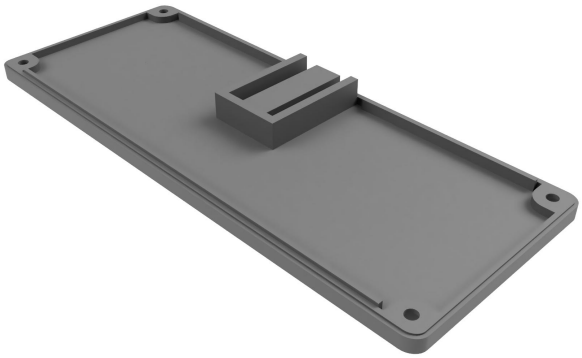
** Recommended, but not essential. 3D printed keycaps would reduce the cost somewhat. If you've got some transparent filament, Mc_gee's printable keycaps are a fantastic alternative (https://www.thingiverse.com/thing:3791028/files).

*** Either Square nuts, or Heat-set Inserts; see Print Settings for details.

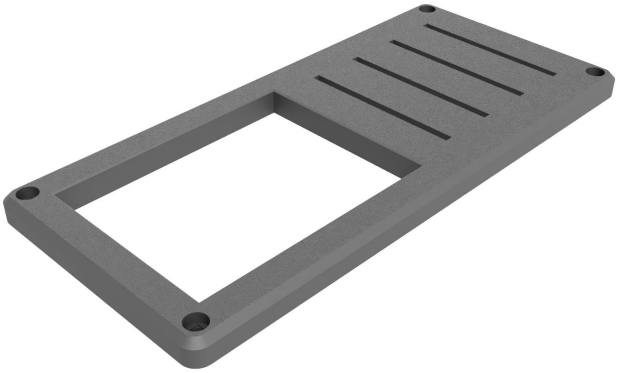
Parts Breakdown



Box_*.stl

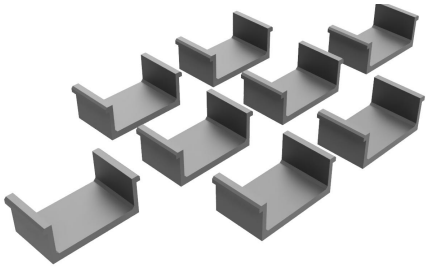


Base.stl

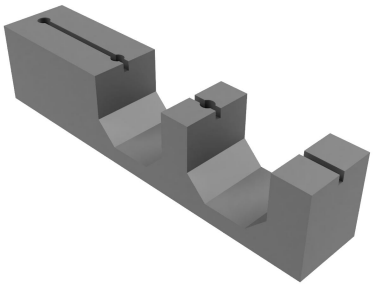


Lid.stl

Fader_knobs.stl



Clips.stl



Solder_jig.stl

Print Settings

Resolution = 0.2mm
Infill = 10%
Perimeters = 3
Supports = No
Brim = Possibly needed for clips

Estimated print time = 12h
Estimated material use = 150g

All STLs are correctly orientated, ready for printing. The models were designed around a 0.4mm nozzle with an extrusion width of 0.45mm, printed at 0.2mm layer height. This is the optimal configuration to eliminate any "gap fill" on the main box and ensure the sacrificial bridging works correctly for the countersunk screw holes. Higher resolutions will be fine, lower (0.3mm for example) might not work, but YMMV.

Box

You've got x3 options here, pick whichever is easiest for you;

- **Box_*.stl** - The original version, uses captive nuts on one side and inserted nuts on the other. I thought I was being clever here, but it's probably a pain for some users.
- **Revised Box** - The second version (remix, provided by senjak - non-logo version) does away with the captive nuts and simply lets you insert them once the print has completed.
- **Box_*_heatset.stl** - The newly-added version that skips the nuts entirely and allows you to use heat-set threaded inserts (this was requested by more than one person so I figured I'd implement it - the more options the better, right?)

(2 version of each provided; Standard and Branded. Don't like the logo? Print the Standard version ;)

Captive nuts

This version will require you to pause the print at a certain layer height, insert the nuts into the cavities, then resume the print. This is slicer/printer dependant. If you're printing at the optimal layer height (0.2mm) the pause will need to be inserted at the start of layer #29.

For PrusaSlicer, slice the file then move the layer slider to the correct layer height. Right clicking on the plus symbol will give you the option of adding custom gcode. Paste in the following;

```
G1 X10.000 Y210.000 E0; parking position
M17; turn on steppers (prevent moving)
M300 S2500 P1000; beep
M117 PAUSED; write message
M1; user stop
M105; return to current temp
```

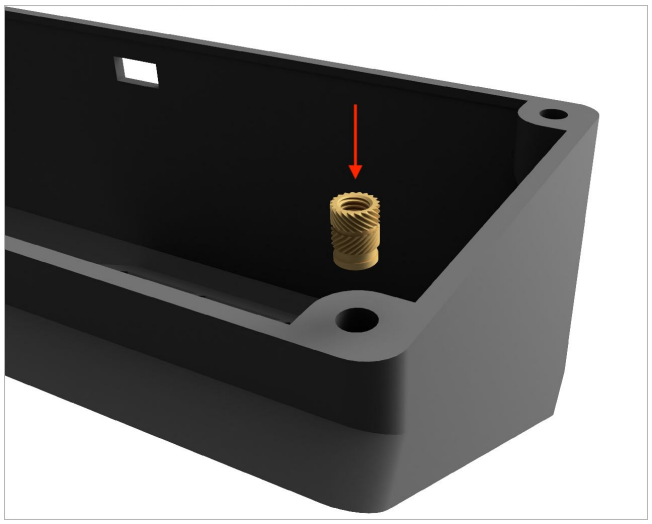
Once the print hits the correct layer it'll move the bed forwards, park the carriage to one side, lock the stepper motors in position and notify you with a "beep". Then it's just a case of inserting the nuts and hitting resume. **Note, if you're using OctoPrint you need to hit resume on the web UI, NOT the Prusa.**

WARNING: If you're not using a Prusa, please double-check the compatibility of the above gcode before sending it to your printer. There may be Prusa-specific commands listed and the last thing I want is for your printer to end up breaking itself!

Heat-Set Inserts

If you go this route you probably already know what you're doing, but I'll provide instructions anyway.

From my limited experience with these, I've found that the following procedure works well;



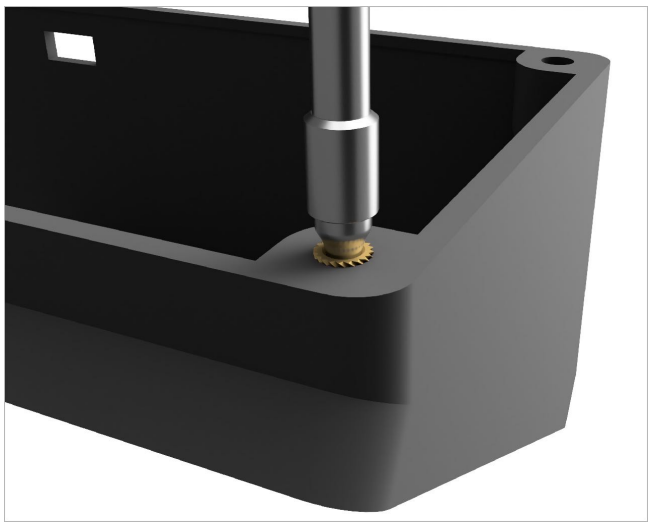
Place an insert into the cavity.

Set your soldering iron to match the hotend temperature of the material you're using (for the PLA I used the temp. was set to to 210°C).

Once the iron is up to temp. place the tip inside the insert. Wait a few seconds for the heat to transfer into the insert.

With minimal downward force the insert will start to descend into the cavity, melting/ softening the surrounding material.

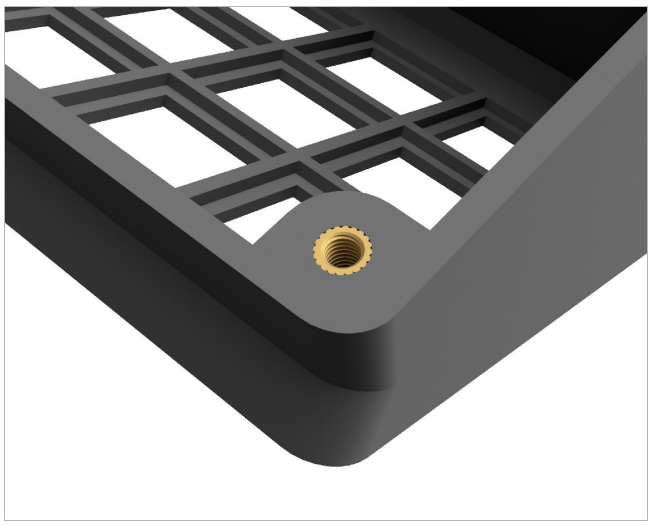
Go slow and steady; this process takes 10-15 seconds.



Once the insert is almost all the way in (roughly 90%), stop applying force, remove the iron and quickly flip the part over placing it on a flat, heat-resistant surface and gently push down.

This will set the insert in place, flush with the face of the print whilst also flattening any material bulge that may have occurred during the process.

Leave the part upside-down for 30 seconds or so to give the insert/plastic time to cool.



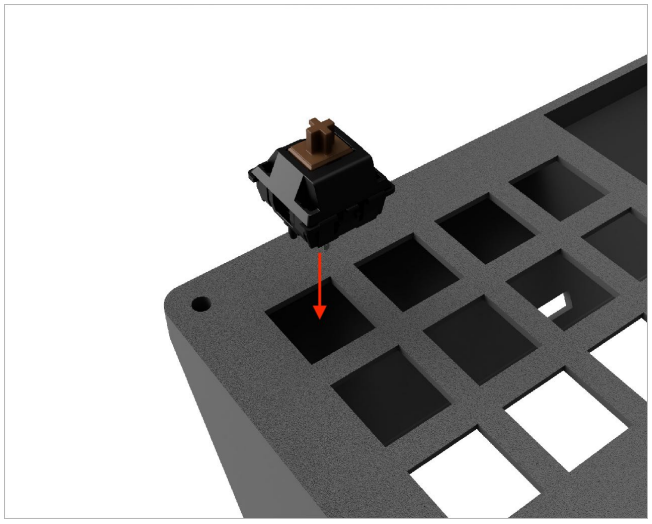
The insert is now installed and should sit flush with the face of the box.

Repeat for the remaining 3 cavities.

Follow the same procedure for the 4 cavities on the other side.

Assembly (switches and potentiometers)

Switches

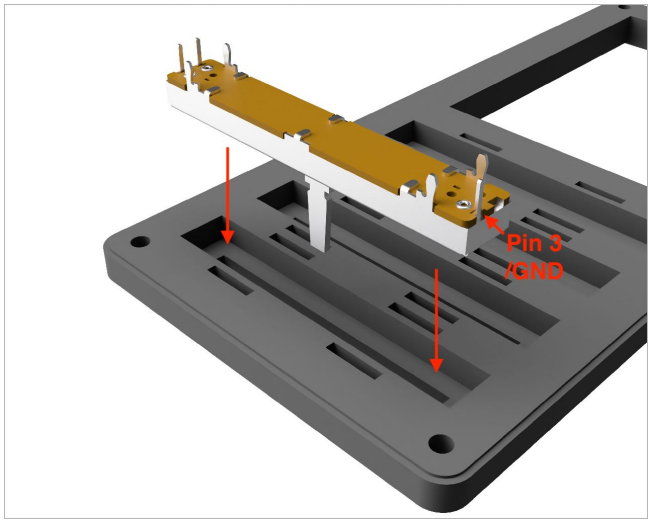


Place the Box on a flat surface and insert a switch. It should pop into place with minimal force. Note the orientation; I've mounted them upside-down on purpose to give a little extra room when soldering.

Repeat the process for the rest of the switches.

From underneath you can see the distance between the uppermost pin and the back wall of the box. With the switch mounted upside-down there is a little extra room for manoeuvring a soldering iron.

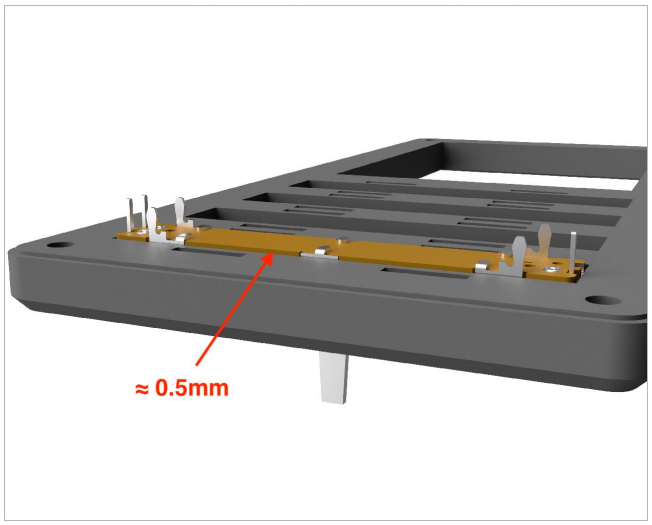
Potentiometers



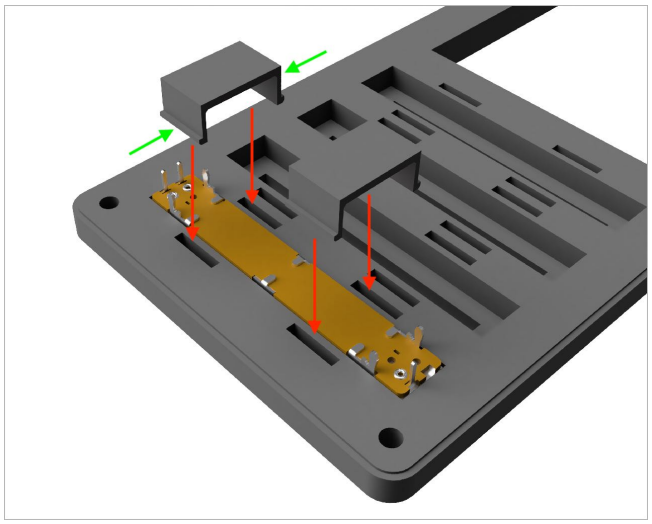
Take the Lid and position it face down (you'll have to hand-hold this one, the pot sliders will protrude from the underside when inserting).

Push a potentiometer into place. It should be a snug fit; not too tight that it requires excessive force, but not too loose that it easily falls out.

Note the orientation; the pin labeled "3" should be located at the bottom (this is for the Bourns pots listed in the BOM. Yours may differ. See the manufacturers data sheet for more info on orientation).



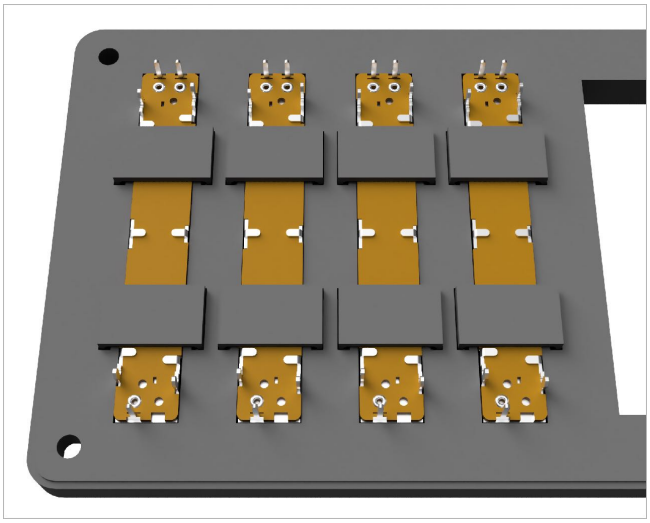
The pot should sit proud from the Lid by roughly 0.5mm This ensures that the mounting clips will butt-up against the pots rather than the Lid.



Take a clip and whilst squeezing slightly (green arrows) insert it into the slits.

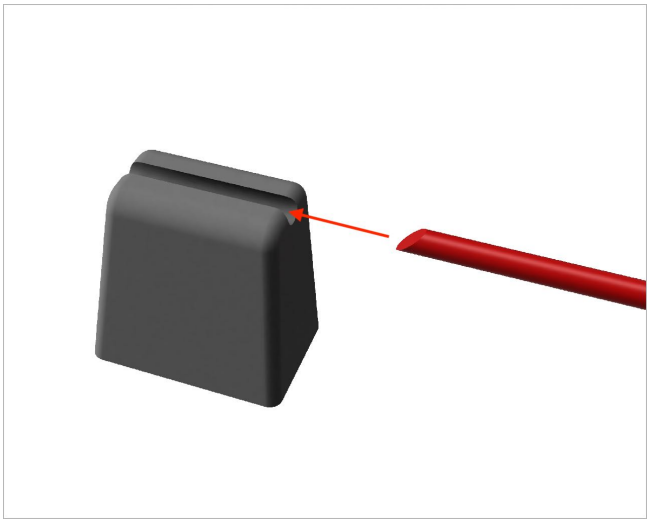
It'll click into place once pressed down.

Do the same for the other clip.



Repeat this procedure for the remaining pots.

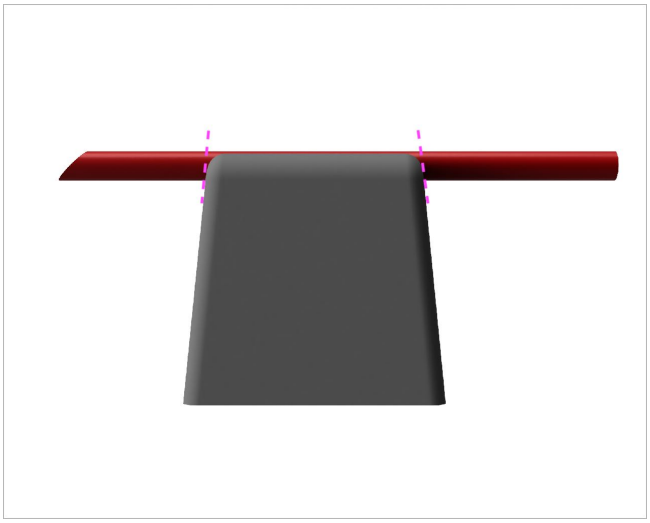
Fader Knobs



The Fader Knobs are designed to accommodate a piece of 1.75mm filament to act as an indicator.

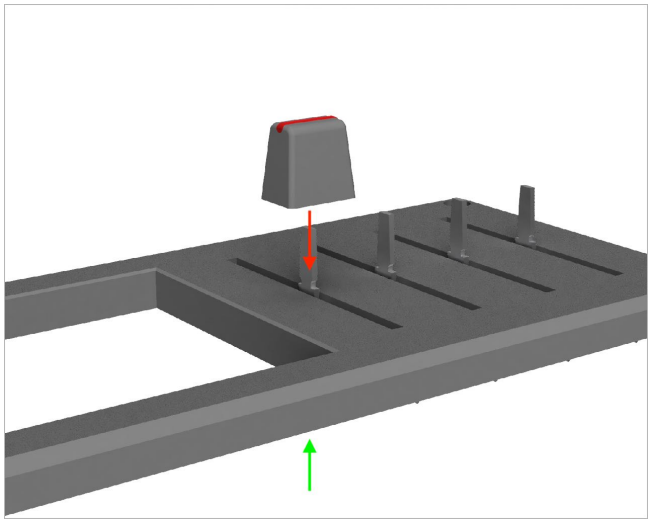
The channel may be slightly rough/tight so it's recommended to use a needle file or 1.5mm drill bit to remove any excess material.

Take a length of filament, straighten it as much as possible, clip off one end at an angle and insert it into the channel.



Using a pair of snips or a craft knife, remove the excess filament.

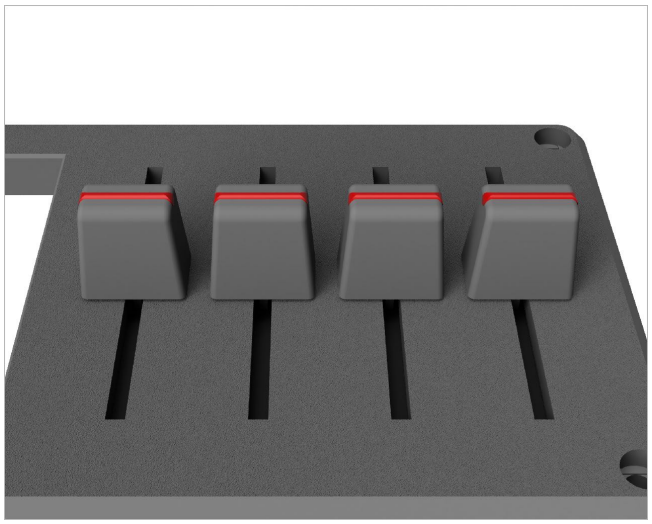
I'd recommend using some sandpaper to remove any sharp edges and round off the filament as much as possible.



Push the knob onto one of the sliders.

Make sure to also apply force from the opposite side to prevent the pot/clips from popping out.

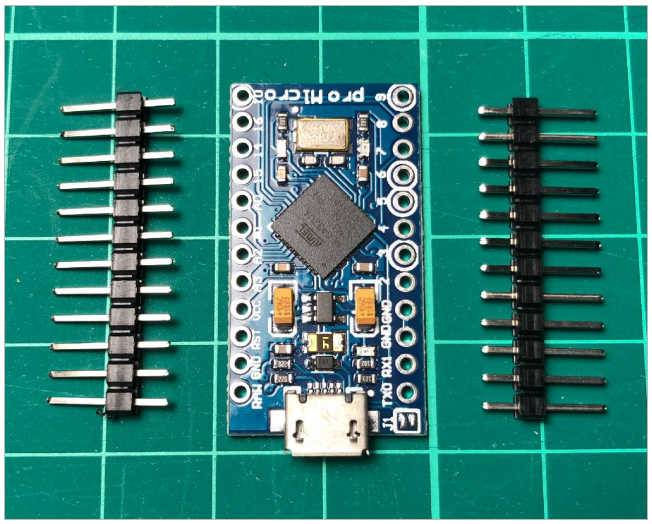
Again, this should be snug but not so tight that it requires excessive force.



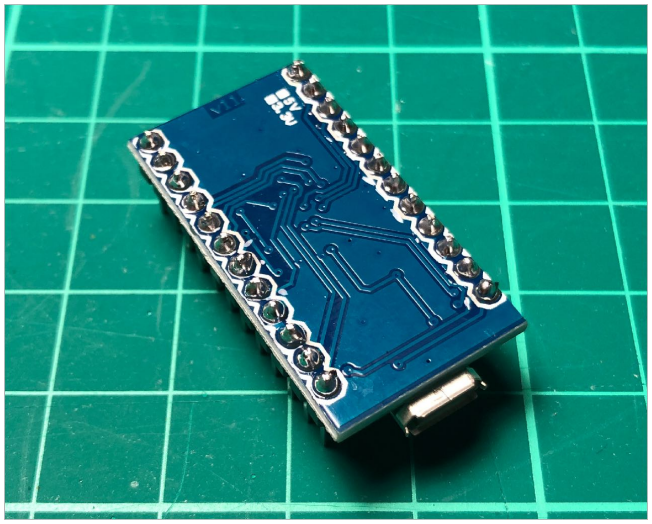
Repeat the process for the remaining knobs.

Soldering/Wiring

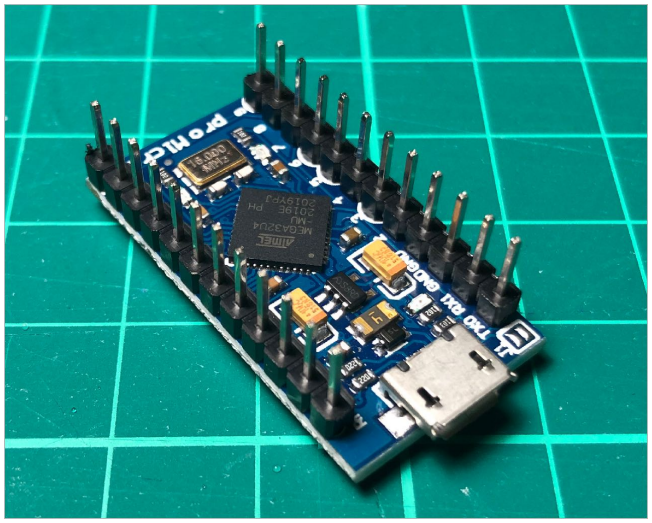
We'll start with the easiest; soldering the pins onto the Arduino.



Insert the pins into the Arduino.

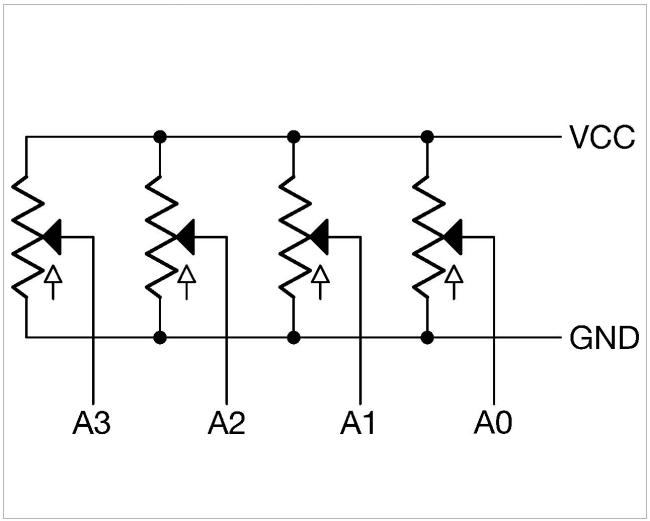


Flip it over and solder the pins into place.

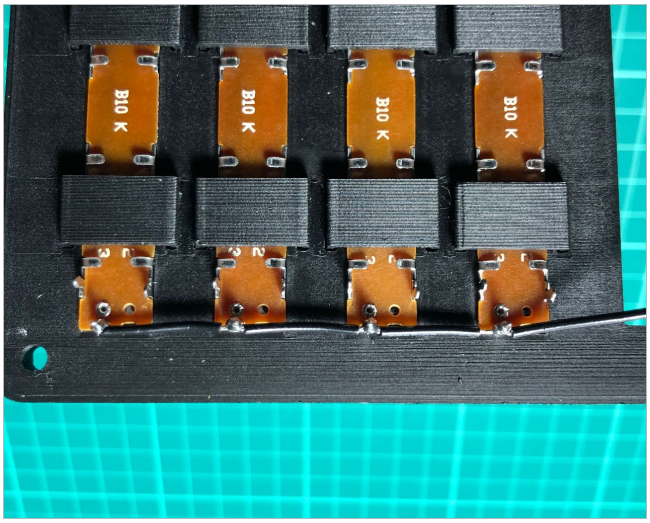


It should look like this when done; pins facing upwards.

Next we'll tackle the potentiometers.



Circuit diagram

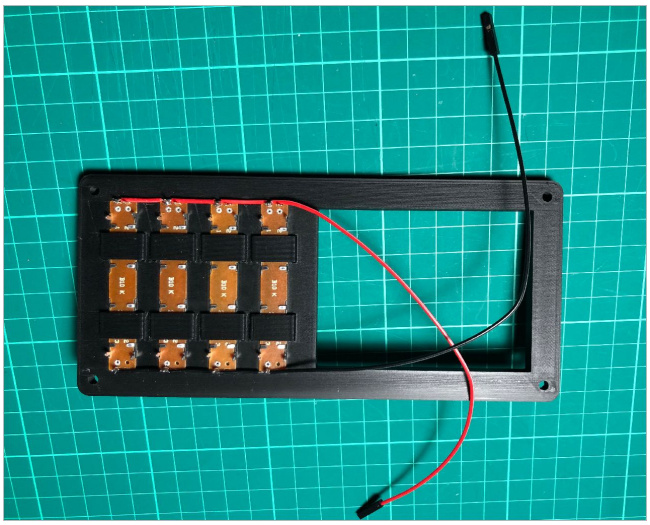


We'll start with the GND pins (labeled "3" on the Bourns pots).

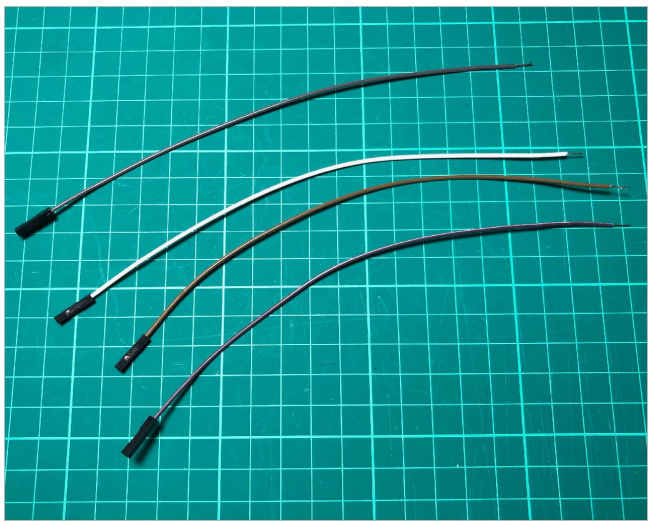
Take a length of wire and cut it into x3 small (roughly 25mm) pieces. Strip the ends then twist and tin the exposed wired. Wrap these around the GND pins, connecting them together.

For the right-most pin, take a wire and snip off one of the connectors (making sure the other end has a female Dupont connector). Strip, twist & tin. Wrap this around the pin.

Solder all wires in place.



Repeat the process for the VCC pins (labeled "1" on the Bourns pots).

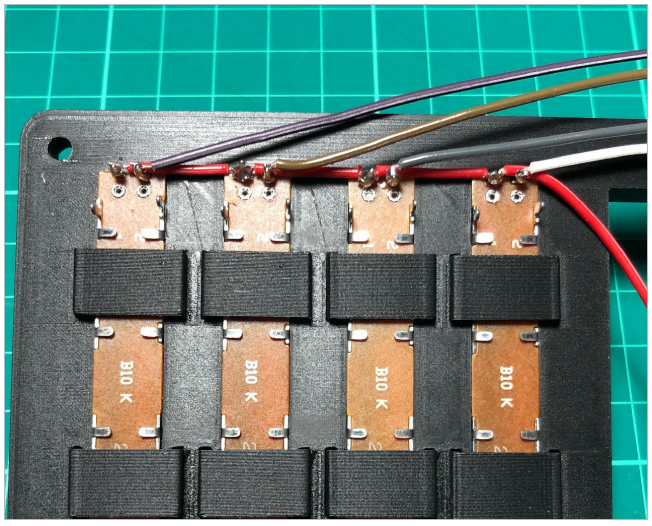


Each of the signal pins (labeled “2” on the Bourns pots) require their own individual wire.

Take 4 wires and snip off one of the connectors (again, make sure the other end has a female Dupont connector).

To make identification easier it is recommended to use a different coloured wire for each.

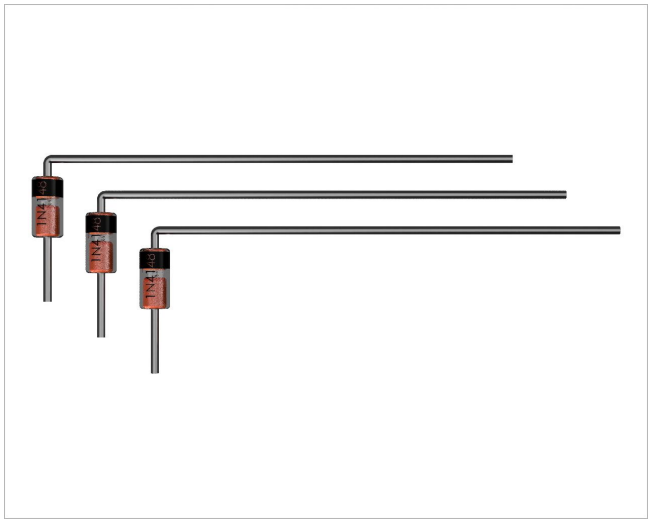
Strip a small section of the sheath on each wire. Twist & tin.



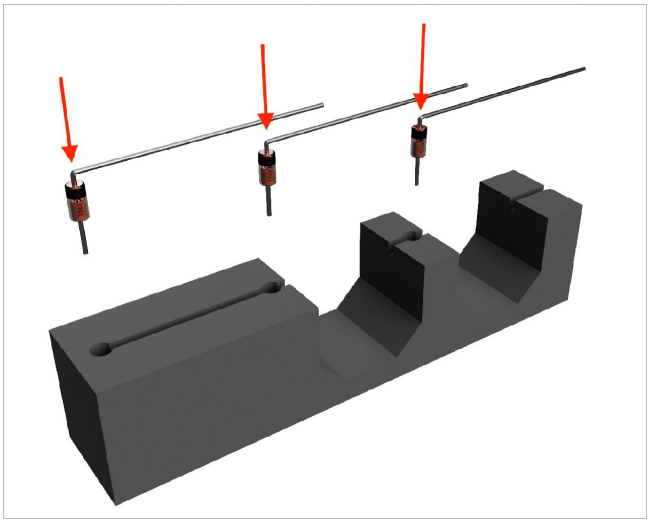
Solder each of these wires to one of the signal pins on the potentiometers.

That's the pots done. Make a note of which colour wire corresponds to which pot as it'll make it easier when it comes to connecting the cables to the Arduino.

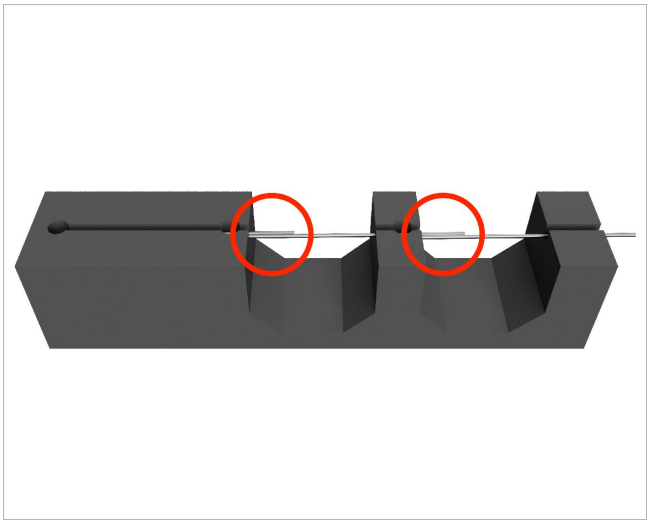
Now onto the switches. We’ll start by using the Solder_Jig to make some diode assemblies.



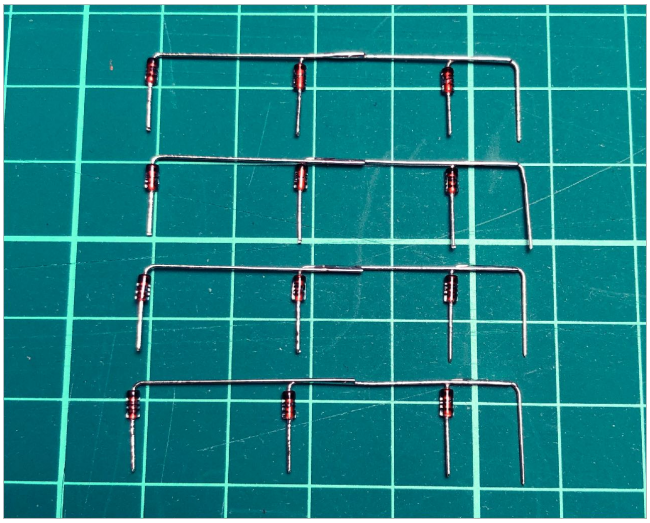
Take 3 diodes and bend the cathode leg (the one next to the black stripe) a little over 90°. Snip off the other leg leaving roughly 10mm.



Insert the diodes into the jig.

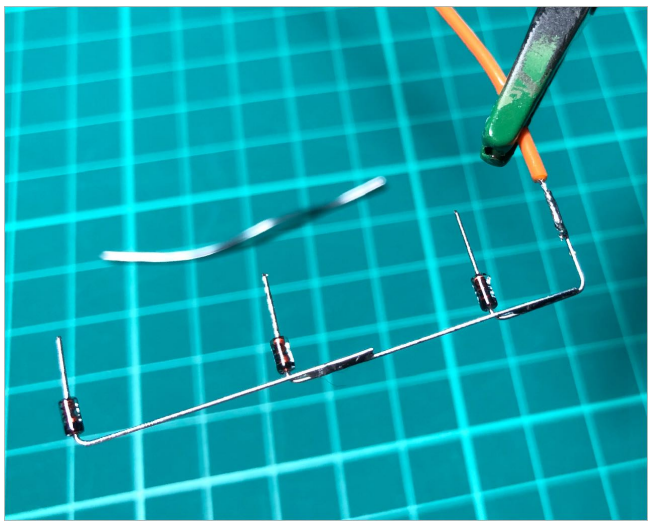


Solder them together where the legs meet.



Remove the completed diode assembly and repeat this process 3 more times.

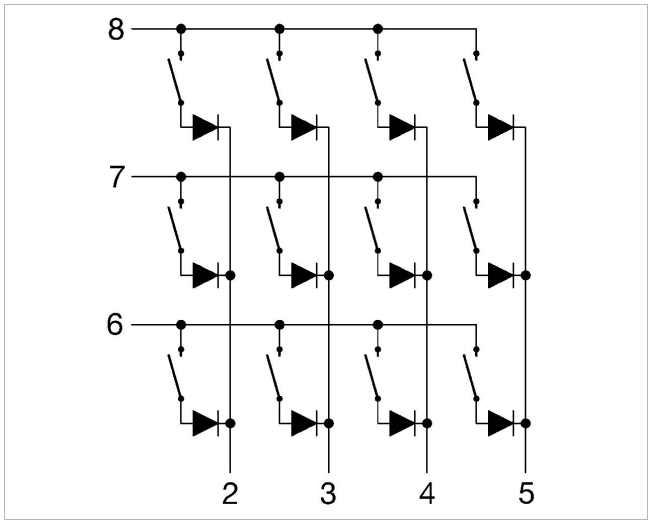
For each diode assembly, bend the long leg 90° just after the solder joint and snip off the excess (roughly in line with the other legs).



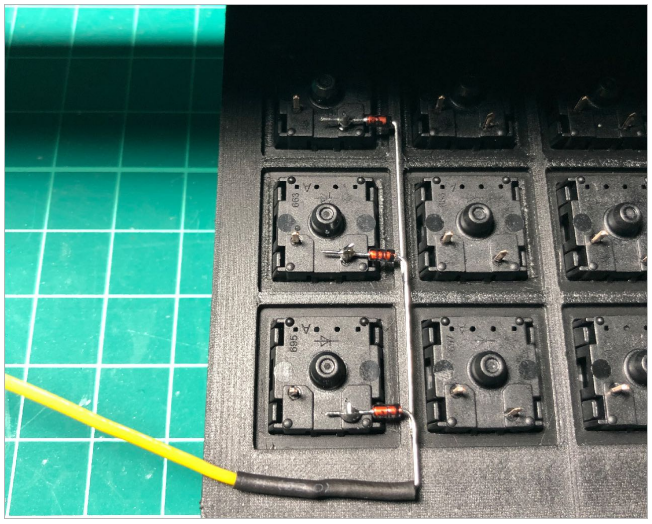
Take 4 more wires, snip off one of the connectors on each, strip, twist & tin and solder onto each of the diode assemblies.

Heatshrink not necessary, but recommended.

You're now ready to solder these onto the switches.



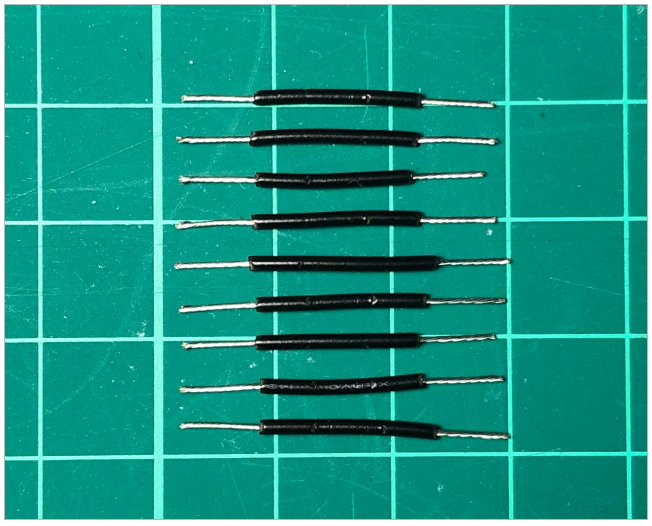
Circuit diagram



Position the assembly in place and solder each leg to the lower pin of each switch.

Repeat for the remaining 3 switch columns.

Again, it's recommended to use different coloured wires for easier identification later on.



Use some more wire to create x9 jumper cables, roughly 25mm in length.

Strip, twist and tin the ends of each.

Use these wires to attach the other switch pins together, working along the rows. If you have a pair of narrow tweezers it'll help with wrapping the wire around each pin.

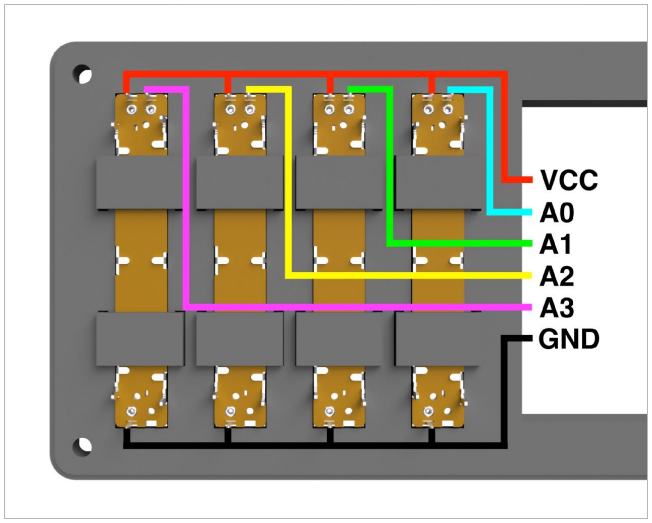
Make sure that none of these wires are shorting against the diodes.

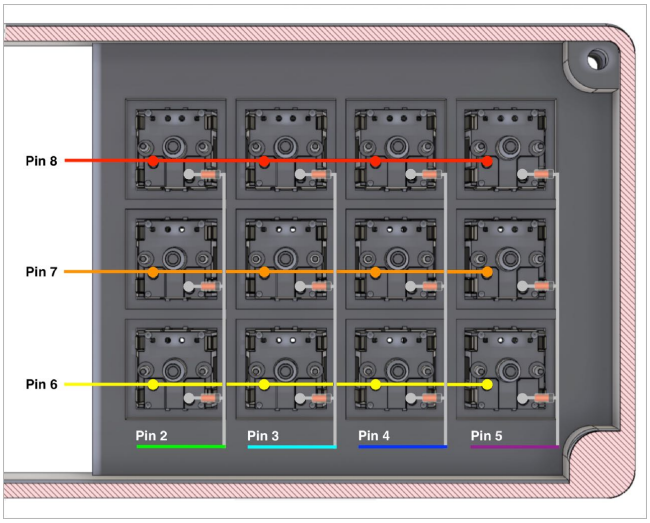
Use another 3 strips of wire, snip off one connector on each, strip, twist & tin the bare wire then wrap them around each pin on the left-most switches.

Solder all wires in place. Be careful here, there's not much room to work and it can be quite easy to accidentally touch the hot iron to the plastic box.

That’s the hardest part done. All that's left is to connect the wires to the Arduino and screw the case together.

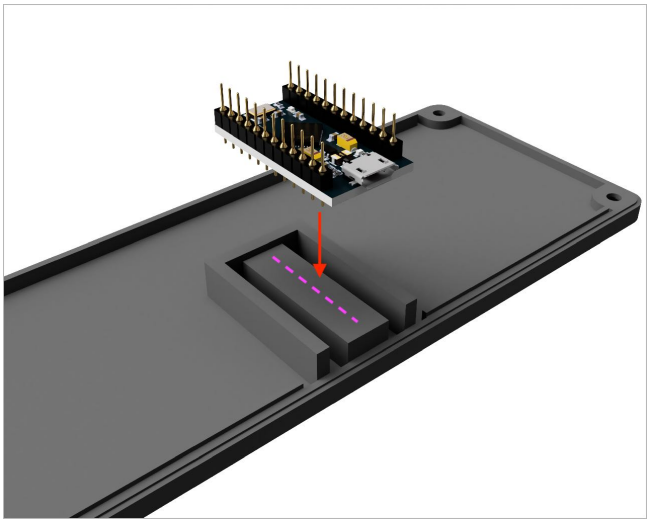
Grab a pen and piece of paper and use the following images to figure out which wires go to which pot/key (your wire colours may vary).





Assembly (case)

Pass the pot wires through the Box and position the Lid in place. Use x4 M3 bolts to attach the Lid. Don't over-tighten these as you can accidentally break the Box.

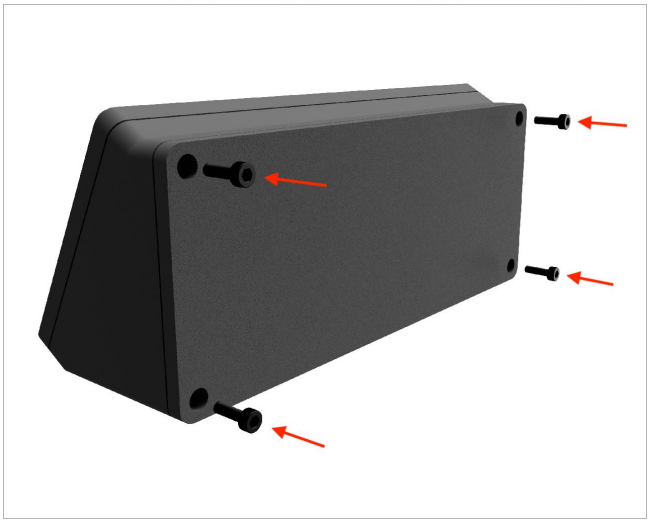


Use some strong double-sided tape (or a small amount of superglue) to attach the Arduino to the Base.

Place the Box on its back with the Base in front and connect the wires to the corresponding pins on the Arduino. Refer to your notes to determine which pins go where (pots to GND, VCC and A0-A3, keys to pins 2 though 8).

Skip this paragraph if you're using heat-set inserts.
Insert the remaining M3 nuts onto the holes. These have a tendency to fall out so a small amount of glue is recommended. Just make sure not to get any in the threads (or do what I did and cram a little Blu-Tak in there, in place of glue!)

Carefully tuck the wires into the box whilst "hinging" the Base into position.



Secure the Base with x4 M3 bolts. Again, don’t over-tighten them

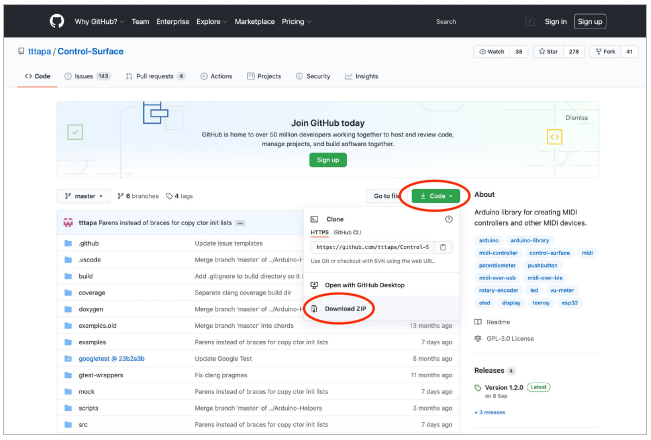
Finally, flip the deck over and attach your key caps.

…and we're done! All that remains is the software setup.

Arduino Configuration

In order to flash the code to the Arduino we must first download a couple of things.

Head to the Arduino IDE download page, grab the latest version of the software and install it.

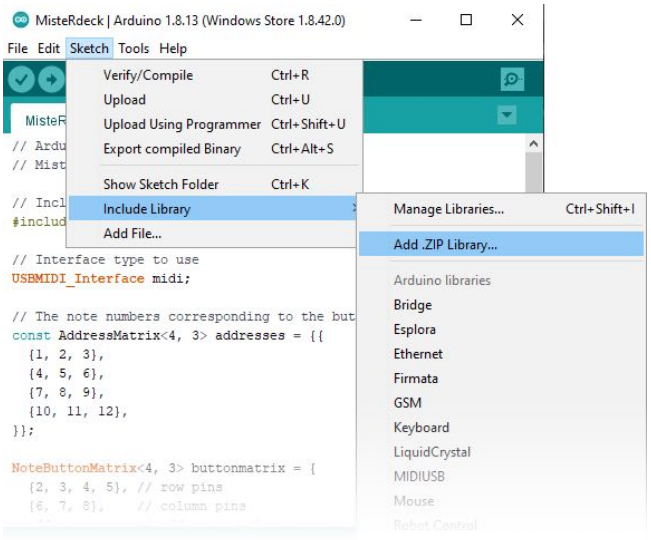


We also need to download the Control-Surface library. From their Git page click on the green “Code” button and select “Download ZIP”.

With everything downloaded and the Arduino software installed we can now connect the MisteRdeck to the PC via a microUSB cable and open the MisteRdeck.ino code supplied with the STLs.

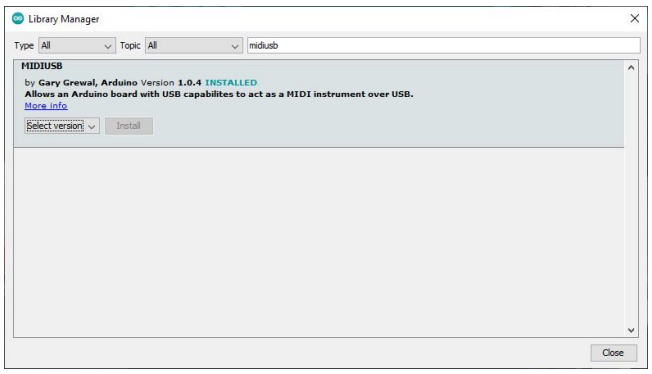
Note: If the Arduino software throws up a message about moving the file into a folder, just click the OK button.

There are just a couple more steps we need to perform before flashing the code.



From within the Arduino software, select the “Sketch” menu item, go down to “Include Library” and then select “Add .ZIP Library…"

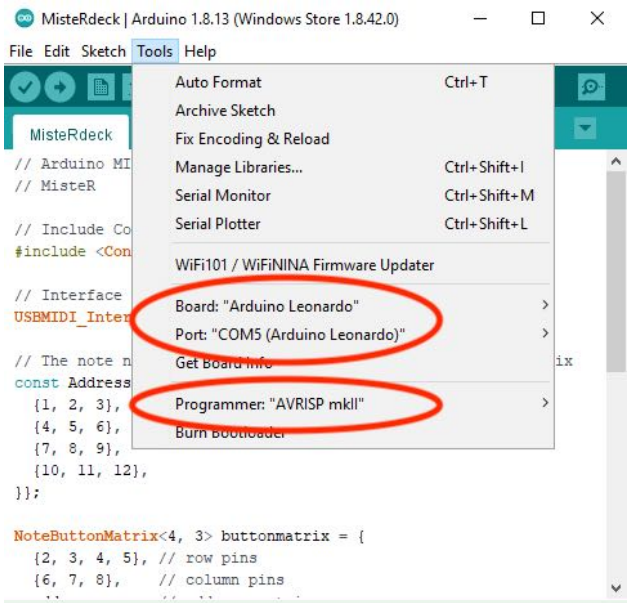
Browse to the location of the Control-Surface ZIP we downloaded earlier and select it.



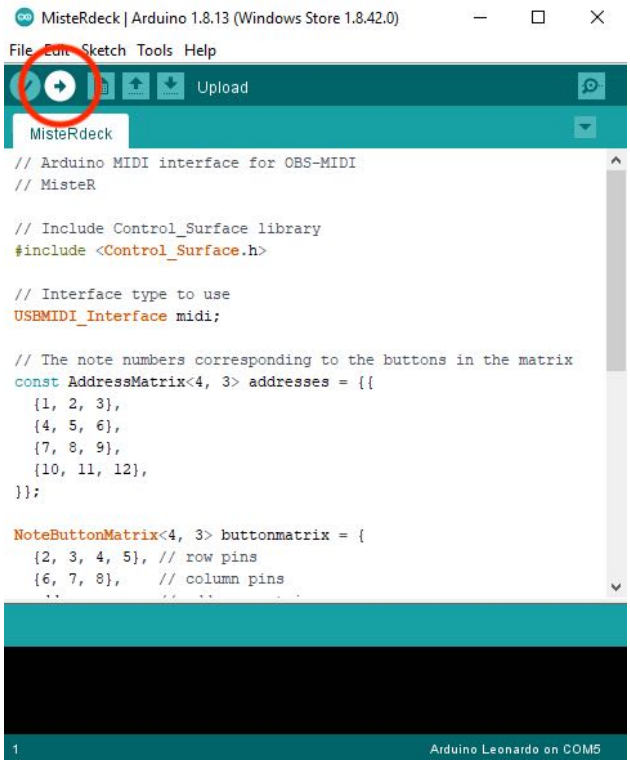
Select the “Sketch" menu item again, go down to "Include Library” and then click “Manage Libraries…”

From within the box that pops up use the search function (top right) to find “MIDIUSB”.

Select the latest version and click Install.



From the “Tools” menu make sure that the “Board”, “Port” and “Programmer” variables are set correctly (your port number may differ).



Now you just need to click the “Upload” button and wait for the “Done uploading.” message in the bottom left corner.

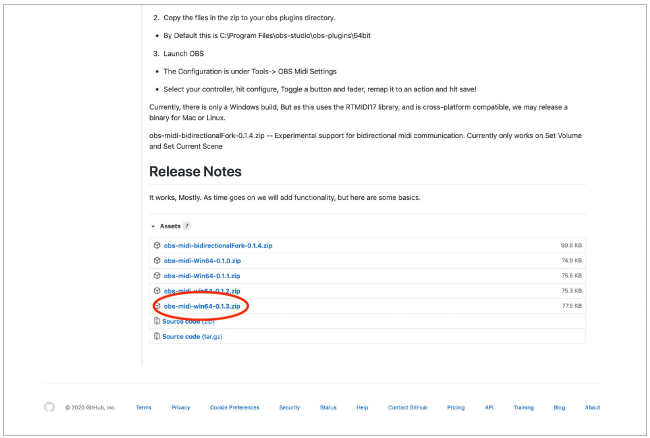
Congratulations, your MisteRdeck is flashed with the code required for operation.

Just one more section and we’re done.

OBS Setup

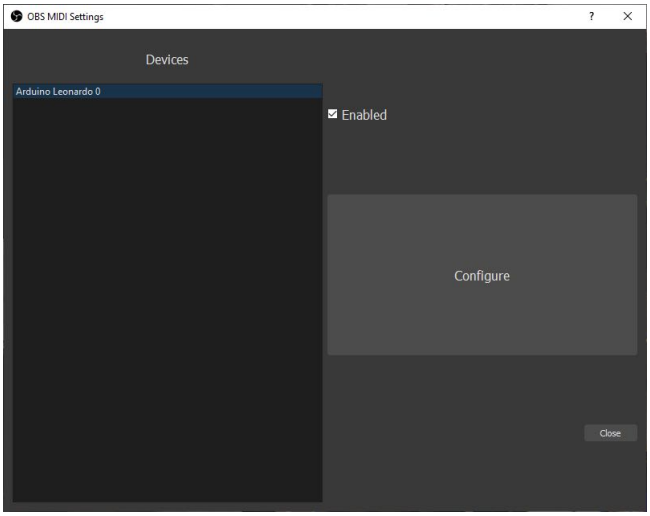
The following instructions are depreciated. I no longer use the OBS-MIDI plugin and instead use a standalone app called MIDIControl. Scroll down for details…

The final step is to get the OBS-MIDI plugin and configure the buttons/sliders in OBS.



Open the OBS-MIDI download page, scroll to the bottom and get the latest version.

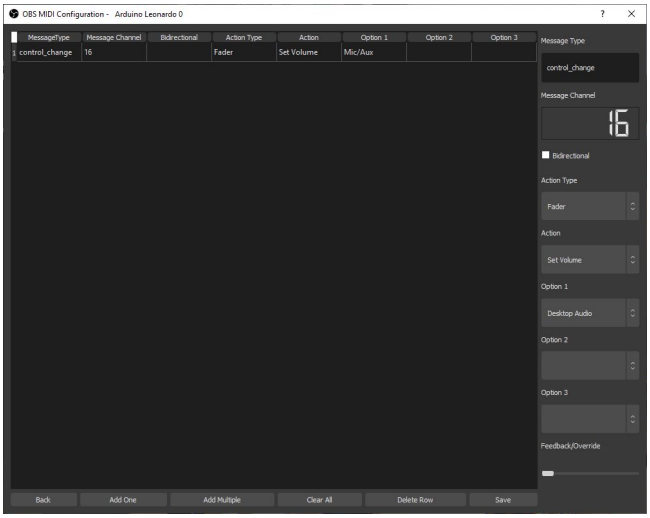
Extract the contents of the ZIP file to your OBS install folder (usually C:\Program Files\obs-studio\obs-plugins\64bit).



Open OBS, click the “Tools” menu and click “OBS Midi Settings”.

In the box that pops up you should see the Arduino listed on the left panel.

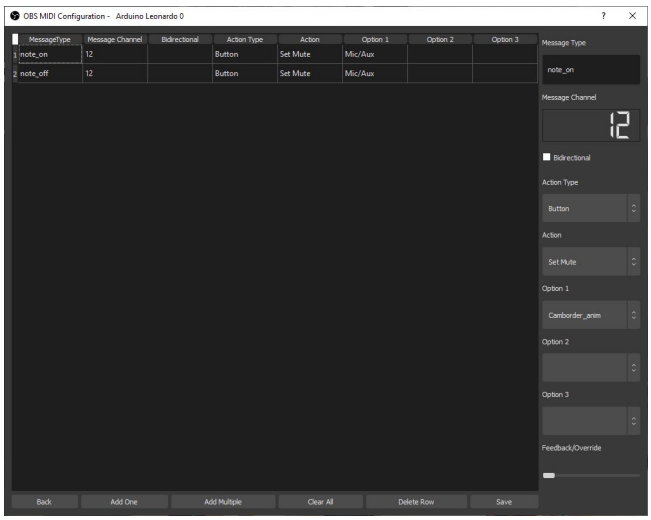
Click the “Enabled” checkbox and hit “Configure”.



When the new window opens, move one of the sliders up/down. It should appear in the left panel.

If we click on its menu entry we can set its options using the drop-down items on the right side panel.

In this example I've set mine to control the “Desktop Audio” volume, but you can set this to whatever you require.



After configuring your sliders we can now move onto the buttons.

Press one of the buttons and x2 entries will appear in the left panel.

Because the buttons work like MIDI notes there are 2 states per button; note_on and note_off (i.e., press and release).
For basic actions like scene switching, we can delete the note_off state (these are useful for things like “hold-to-mute”).

Configure each button to perform the task you require.

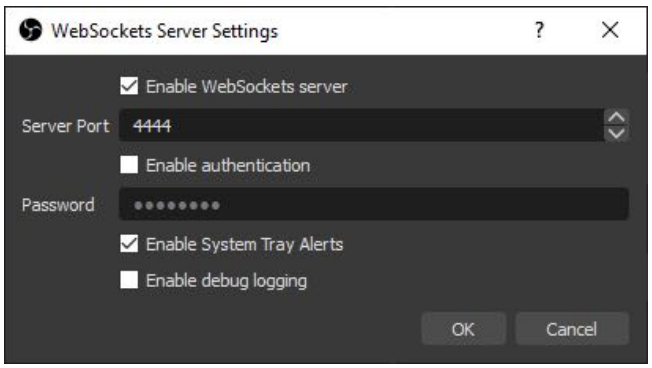
Hit “Save” and close the window. Your buttons/sliders will now perform the functions you’ve set up.

OBS Setup (websocket/MIDIControl)

I've found that the OBS-MIDI plugin isn't currently the best solution; it's prone to crashing and seems to have been abandoned. As an alternative, I'm now using an app called MIDIControl, in conjunction with the obs-websocket plugin. This app offers a few more features (soundboard!), hasn't yet crashed and is actively being developed and improved upon. I'd recommend following these instructions instead of using OBS-MIDI.

Head over to the obs-websocket site, download and install the latest version. Do the same for MIDIControl.

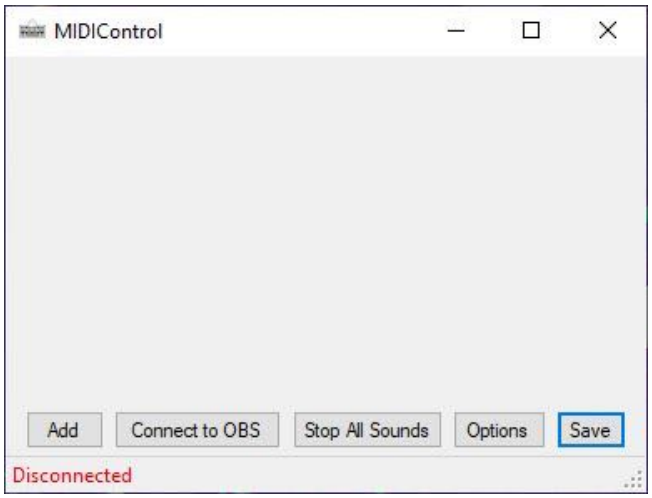
Once installed, open OBS and the MIDIControl app (this lives in your taskbar).



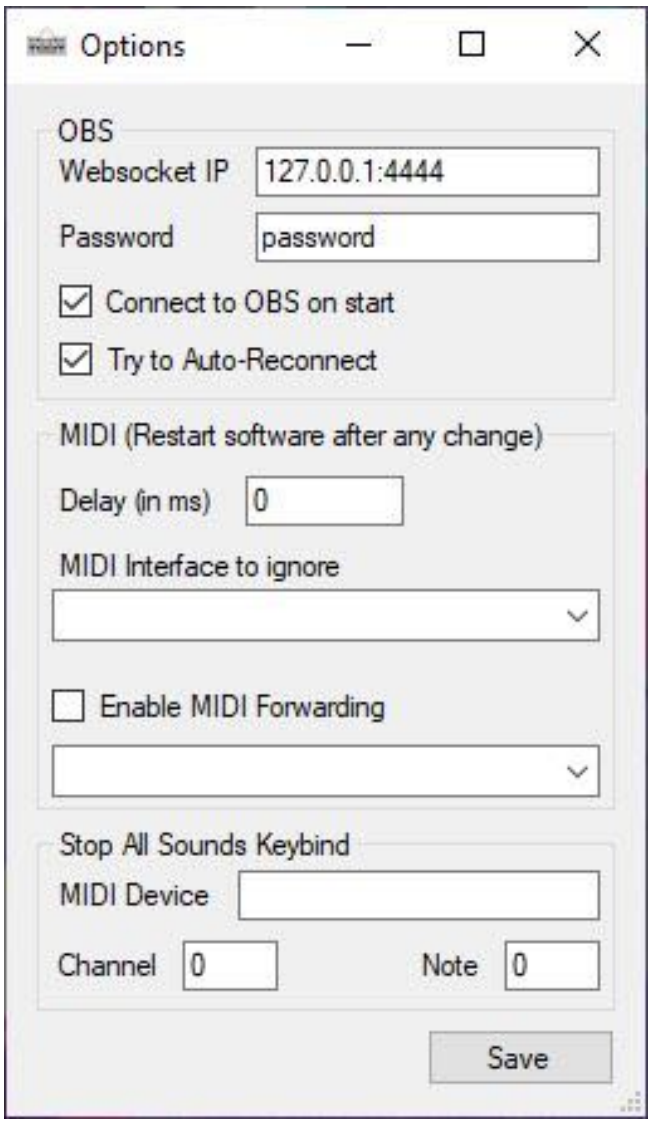
Within OBS, go to the Tools menu and open the Websocket Server Settings.

In here you'll need to enable the server and optionally set a password (recommended).

By default, the websocket runs on port 4444.



Open MIDIControl (taskbar) and click the Options button.

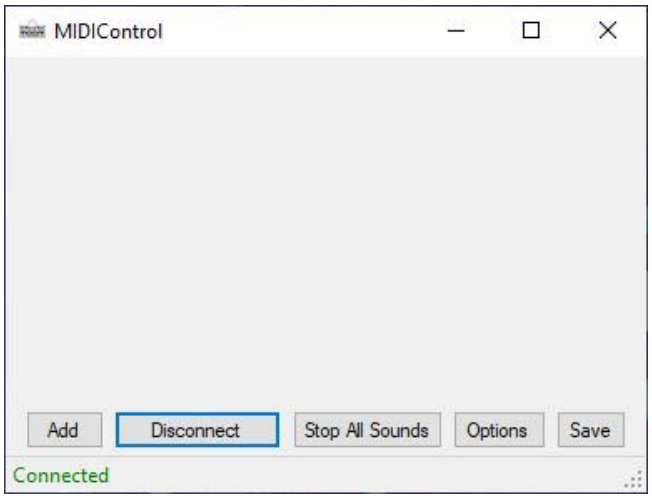


Set the following options.

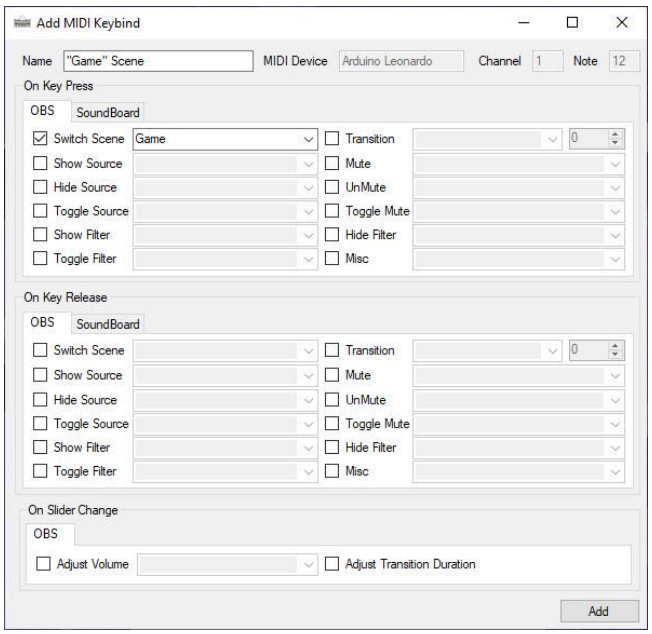
If you've set a password you can enter it here.

Also, if you're running OBS on a separate PC (dual-PC streaming setup) you can change the IP address to match your 2nd PC.

Hit Save.



MIDIControl should auto-connect to OBS (hit the Connect button if it doesn't).



You can now click the “Add” button and start defining keys, sliders and their functions.

Simply press a button, or move a slider and the Channel/Note will appear in the top-right boxes.

Click the checkbox of the function you want and set its parameter.

In this example I’ve configured a button (Channel 1, Note 12) to switch to my “Game” scene.

Once done, click “Add”. Repeat for the remaining buttons and sliders.

Remember to hit “Save” on the main screen when you’re done.

And that’s it!

Make sure that MIDIControl is set to autostart and auto connect and you should be golden.

Another great feature of MIDIControl is that it also supports MIDI forwarding which allows you to use the MisteRdeck to control 2 applications simultaneously, using a virtual MIDI port app like loopMIDI. I have mine set up so the buttons control OBS and the sliders control Voicemeeter (for audio mixing). Instructions on how to set this up are out of the scope of this guide, but some Googling should provide you with the info needed.

Conclusion

Congratulations on making it to the end of the build! If you run into any issues along the way leave a message on the Thingiverse page, or send me a DM and I'll try and help.

Please do also consider taking a quick photo and uploading a "make" to Thingiverse. I love seeing my things printed on a variety of machines.

Changelog

v1.01 - 12/11/20 - Specified M3 nut thickness (thanks Wrightboy)
v1.02 - 15/11/20 - Added instructions for using heat-set inserts in place of M3 nuts
v1.03 - 23/11/20 - Slight alteration to the wording of the BOM
v1.04 - 27/11/20 - Added link to printable relegendable keycaps
v1.05 - 07/02/21 - Added instructions for MIDIControl and obs-websocket