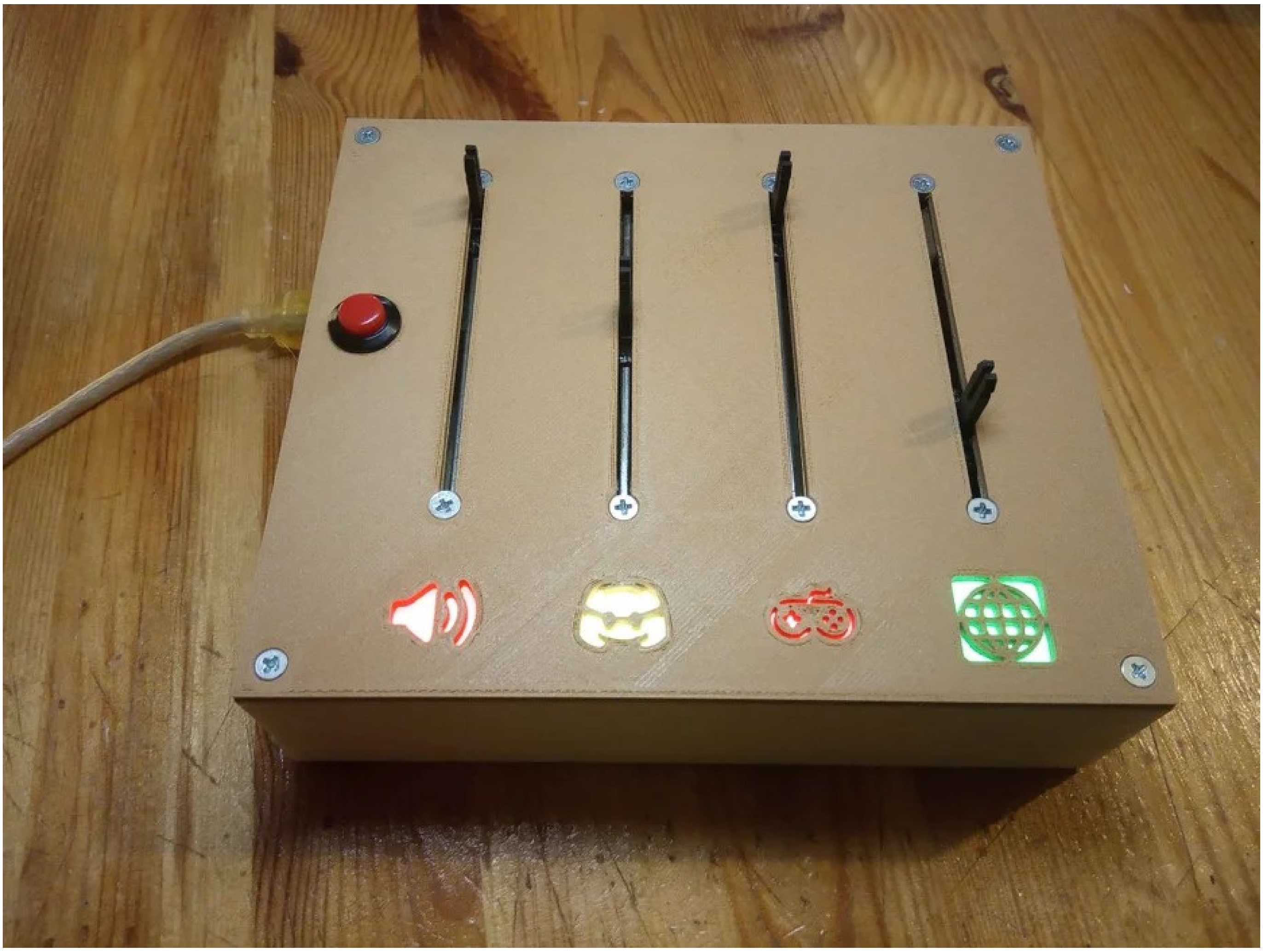
AUTODESK
Instructables

Hardware Volume Mixer for PC With RGB LEDs

By User 7291 in Workshop3D Printing

Introduction: Hardware Volume Mixer for PC With RGB LEDs



Fusion Projects »

So I found the deej project and made my own controller for it. Here I'm showing how you can build my design, or how to implement the lighting icons into yours.

Some words from the author of deej about the project:

> "deej is an open-source hardware volume mixer for Windows and Linux PCs. It lets you use real-life sliders (like a DJ!) to seamlessly control the volumes of different apps (such as your music player, the game you're playing and your voice chat

> session) without having to stop what you're doing."

This instructable will consist of two parts (one with building mine and one how you can implement LEDs into your design) these will be written somewhat simultaniously, but I'll mark them.

I tried to write this as beginner friendly as possible, but might have missed mentioning something what's obvious for me, but might not be for you, so feel free to ask! You can also join the discord of deej for some deej related questions (not about my design decisons. They might not be able to answer it).

Supplies

4x 10K sliding pots (When building my design you have to use one with the highlighted specs matching. Othervise you can even use rotating pots)

4x ws2812(B) LEDs (can be cut from a strip of neopixel or can be one like this. (I used the second option))

1x arduino nano (could also be an uno or even a Mega(why?) but those wouldn't fit in my case)

1x pushbutton which fits in a 13mm hole. (Obviously this is only specified for my box)

1x switch like this.

1x female usb B connector and a male usb mini (these are not necessary even in my design but more about this later)

8x M3 screw 4mm, countersink (Mine were cut from longer screws)

4x M3 screw 10mm, countersink

4x M3 screw 25mm, countersink

4x M3 nut

The 3D printed parts: (Thingiverse)

1x TopPlate [not white?] (this one is a bit tricky, and needs good bed adhesion)

1x Top_Wall [what you want]

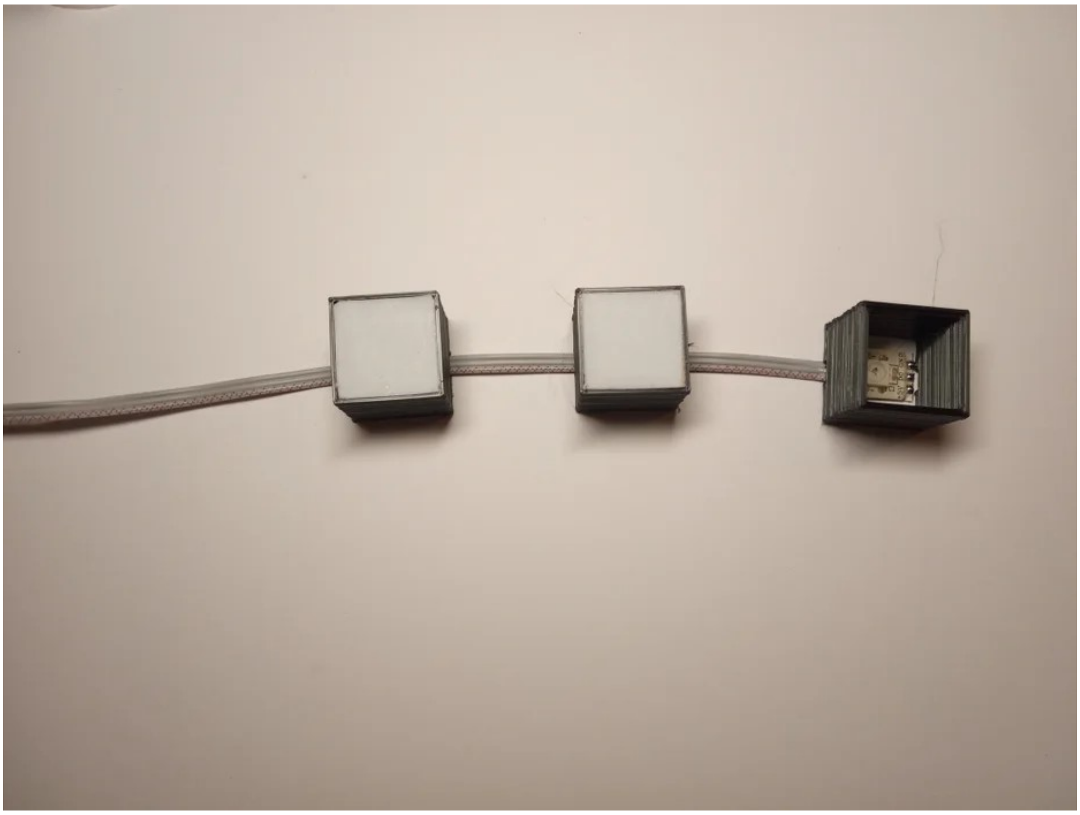
1x Bottom_V2 [what you want]

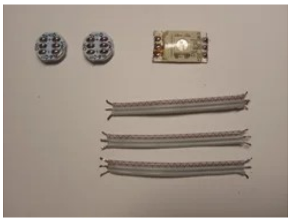
4x Backlight_Diffuser [white](transparent?)

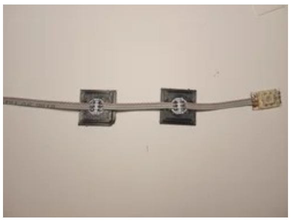
4x Backlight_Wall [I recommend something dark]

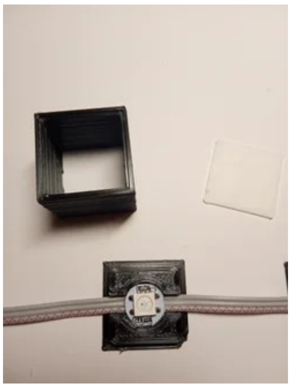
4x Led_Bracket [Doesn't really matter]

Step 1: The LED Backlights









So these are the lamps what makes the icons glow.

As you can see there are 6 pins on one led:

1x DIN (connect this to the previous LED's DOUT)

1x DOUT (connect this to the next LED's DIN)

2x 5V (They are internally connected, so it doesn't matter which you connect to which 5V source/next LED's 5V)

2x GND (Same here as for the 5V pins)

First you should cut 3x ~4cm sections of wire, and a longer one to connect the chain to the arduino. Tin everything.

Solder the long wire to the first LED's [5V] [DIN] [GND] pins. Then connect the first LED's [other 5V] [DOUT] [other GND] to the second LED's [5V] [DIN] [GND] and so on. (For my build you go for 4 LEDs in the chain, but you can make it more/ less)

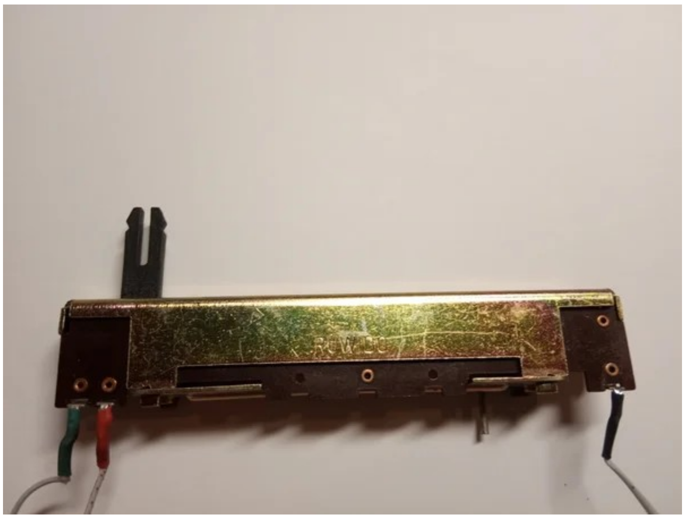
I figured I can use the LED brackets to help hold the LED while soldering. Then take the LEDs out, put them in the right side up. After that, you put the walls on, and the diffuser ontop. These are designed to be press-fit, but you might have to sand down a bit from the parts if they don't fit.

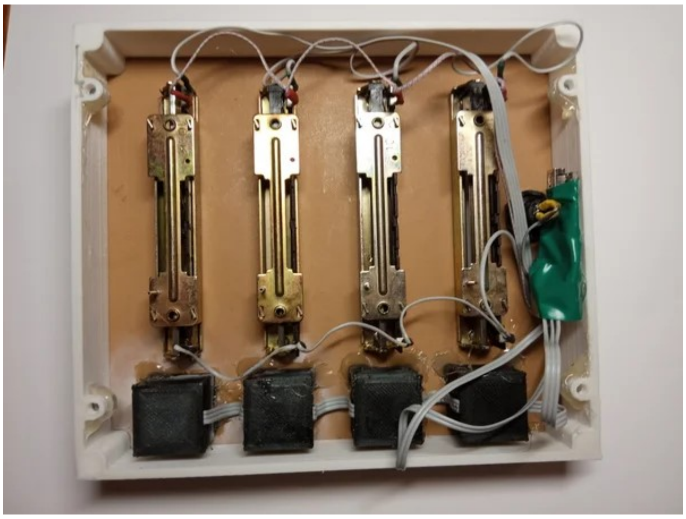
If you wired up the LEDs like this you just have to connect the first LED's GND, 5v, and DIN later.

And your cube lamp chain is ready.

(Don't mind the LED cut off a strip on the pics. That's just for demonstration that you can use such LEDs too, and I even connected it badly (5V & GND swapped))

Step 2: The Wiring





Don't worry! This isn't that hard. (I spent about one hour trying to find out how to make a drawing for this, but I couldn't get one. Sorry! (If someone can get one I'd apreciate it))

There is a picture of my slider wired up. It has 3 pins: ground(balck) 5 volts(red) "signal"(green). Pinouts may differ so you should check yours before wiring. (Not quite sure how to do so, but I wired my pots wrong the first time and got a short circuit)

You connect every +5V (red) pins together (one per component is enough)(4 slider; 1 arduino; 1 LED chain)

You connect every GND (negative, black) pins together (one per component is enough)(4 slider; 1 arduino; 1 LED chain; 1 pushbutton)

Here are the "special pins:

Call the most left slider 'slider1' the second slider 'slider2' and so on.

Connect [slider**1**'s signal] to [A**0**] on arduino

[slider**2**'s signal] to [A**1**] on arduino

[slider**3**'s signal] to [A**2**] on arduino

[slider**4**'s signal] to [A**3**] on arduino

*[slider**5**'s signal] to [A**4**] on arduino (If you're making a your own build with more channels. I think you can go for 8 channels max but I've never tried and don't know if there's a limit in the deej.exe)*

Led chain's first [DIN] to [D6] on arduino.

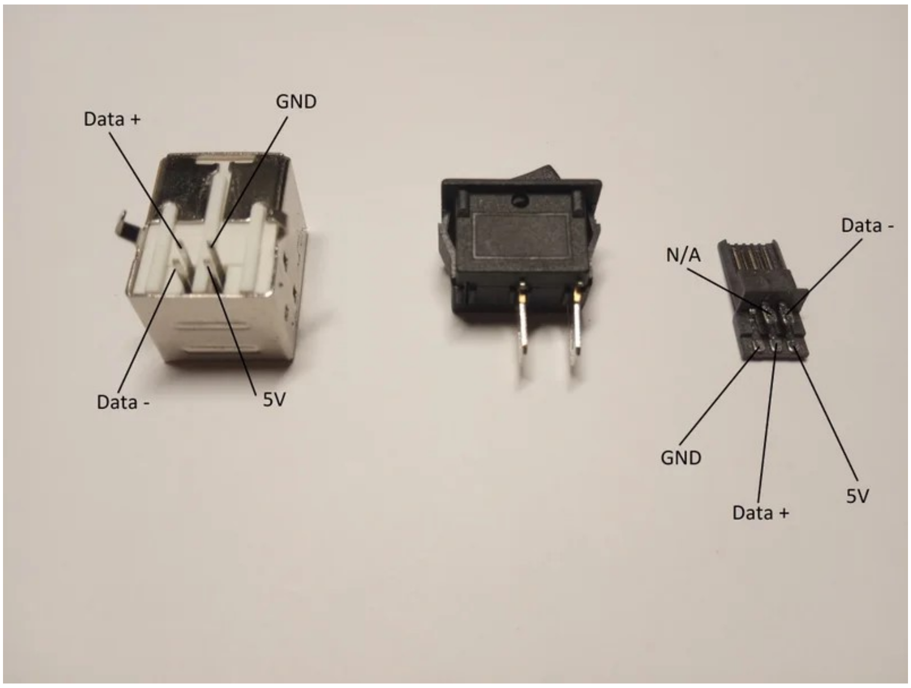
Connect [one pin] of the pusbutton to [D2] on arduino *(*use some connectors *(which fit through the hole for the button)* here, or solder it in place*)*

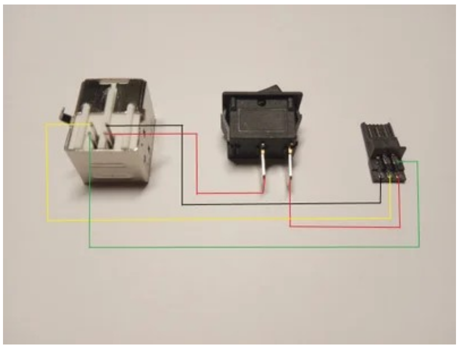
Connect [other pin] of the pushbutton to GND

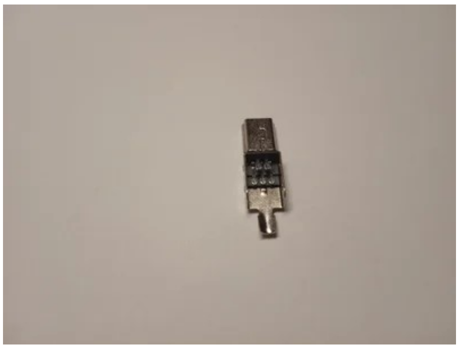
The switch and the usb Connectors are in the next step

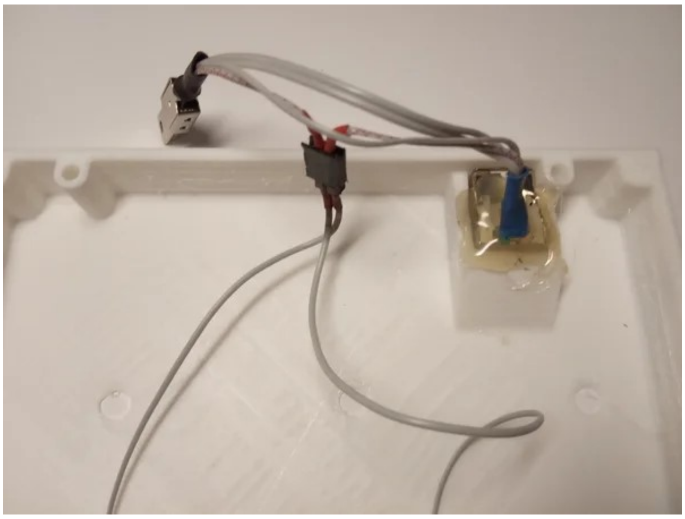
Always check for shorts. It's easier to fix now, than disassembling the whole thing when something don't work.

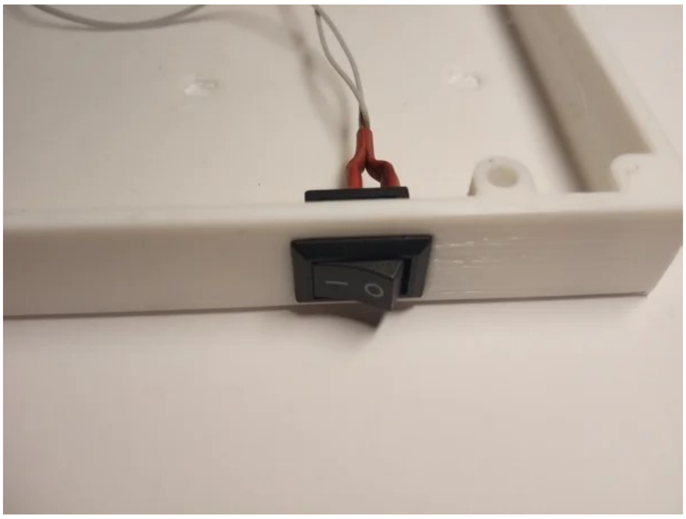
Step 3: The USB Converter











(If you're only interested in implementing LEDs into your design you can skip this step)

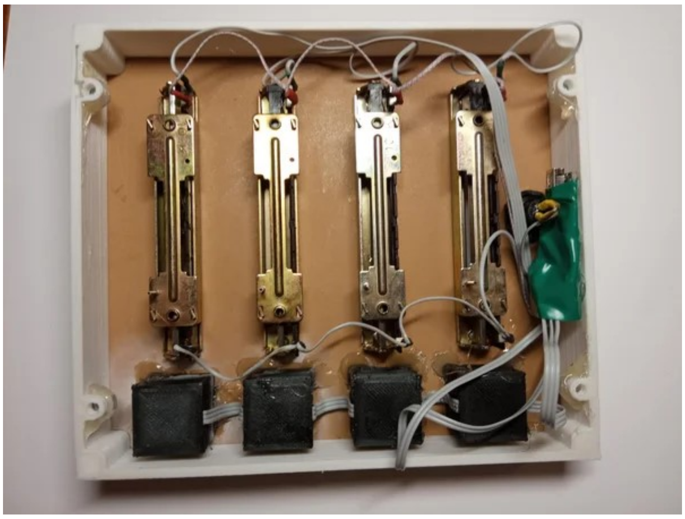
Basically you can just plug a normal USB cable into the arduino, and guide it out where the USB B connector would be. You can also cut this wire and put the switch in the middle of the 5V line, or simply don't use a switch.

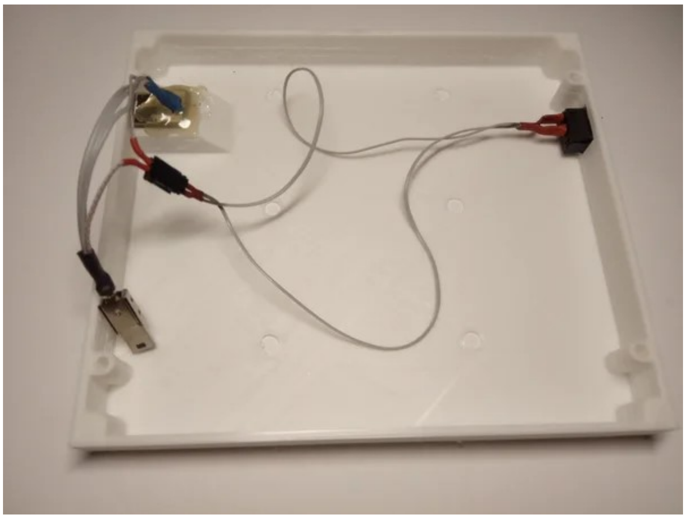
But I show you how to make it in a nicer way.

This time I got you some wiring diagrams. Based on those I'm sure you can connect 10 pins together correctly. I recommend using some heat shrink tubes, or electrical tape. You don't want a short (most modern PCs have protection for overcurrent protection, but still). After you soldered everything, push the mini USB's plastic connector part into the metal housing (If you have that kind of connector). Also, use some connectors(*which fit through the cutout for the switch*) for the switch, or solder it in place.

Now test this one too! Just see if a computer can recognise an arduino connected trough this wire.

Step 4: Assembling





(If you're only interested in implementing LEDs into your design you can skip this step)

Good news! There's no more soldering. First mount the sliders on the top plate with the 4mm screws. By now you should have inserted the button. If not, do it now.

Insert the 4 nuts into the hexagonal holes in 'Top_Wall'. Might need some force, or some superglue for this.

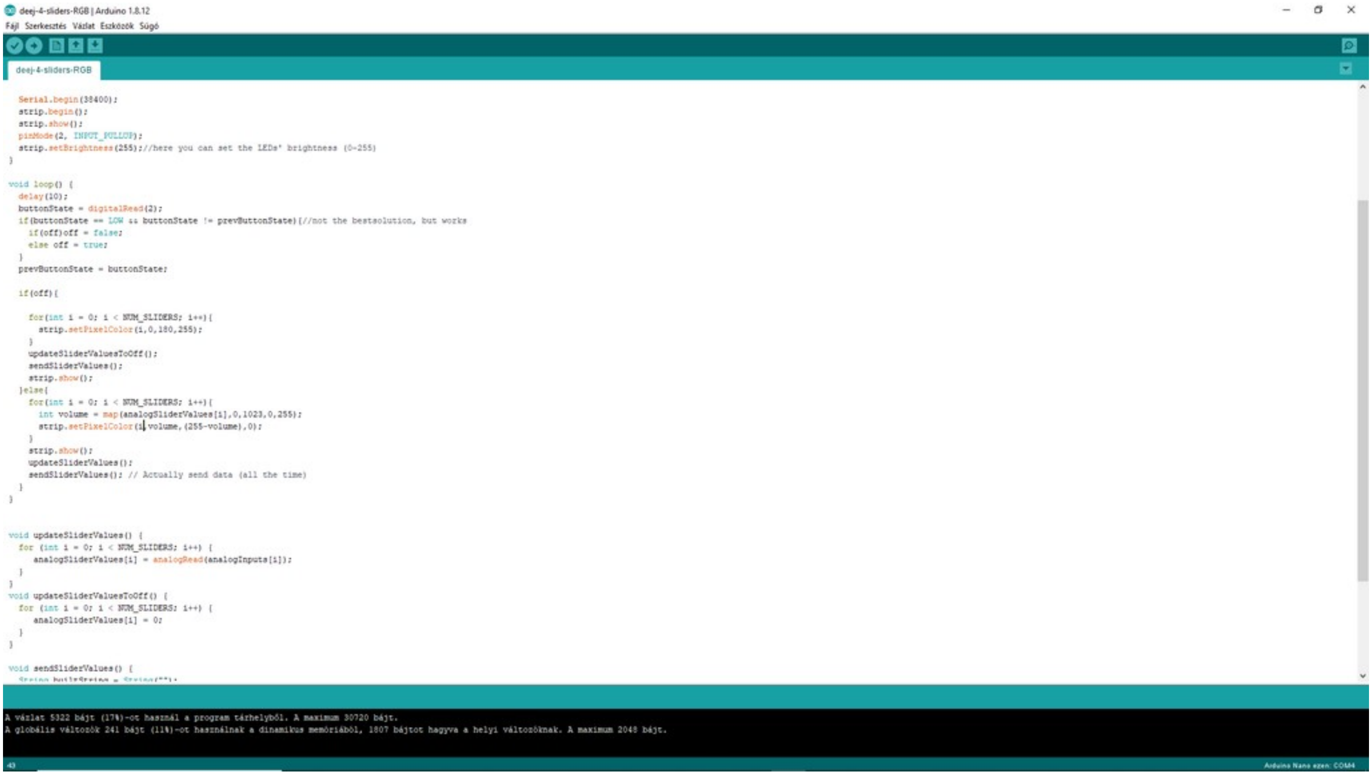
Now assemble the previous two parts (10mm screws here). Before I screwed mine in I put some superglue on the threads. It was a previous iteration and the holes were too big. It's most likely good now.

You can see that in the first iteration (picture) I put the block lamps behind the icon cutouts just by eye. Now there are guides to help you place them. Apply some glue to make sure they don't move.

Here comes the usb and the 'Bottom V2'. Put the USB B to where it belongs and then secure it with glue.

Plug the USB mini in the arduino nano, glue the it somewhere, and screw the bottom to the 'top wall'.

Step 5: Programming


deej-4-sliders-RGB | Arduino 1.8.12
Fájl Szerkesztés Vázlat Eszközök Súgó
deej-4-sliders-RGB

```
  Serial.begin(38400);
  strip.begin();
  strip.show();
  pinMode(2, INPUT_PULLUP);
  strip.setBrightness(255);//here you can set the LEDs' brightness (0-255)
}

void loop() {
  delay(10);
  buttonState = digitalRead(2);
  if(buttonState == LOW && buttonState != prevButtonState){//not the bestsolution, but works
    if(off)off = false;
    else off = true;
  }
  prevButtonState = buttonState;

  if(off){

    for(int i = 0; i < NUM_SLIDERS; i++){
      strip.setPixelColor(i,0,180,255);
    }
    updateSliderValuesToOff();
    sendSliderValues();
    strip.show();
  }else{
    for(int i = 0; i < NUM_SLIDERS; i++){
      int volume = map(analogSliderValues[i],0,1023,0,255);
      strip.setPixelColor(i,volume,(255-volume),0);
    }
    strip.show();
    updateSliderValues();
    sendSliderValues(); // Actually send data (all the time)
  }
}


void updateSliderValues() {
  for (int i = 0; i < NUM_SLIDERS; i++) {
    analogSliderValues[i] = analogRead(analogInputs[i]);
  }
}
void updateSliderValuesToOff() {
  for (int i = 0; i < NUM_SLIDERS; i++) {
    analogSliderValues[i] = 0;
  }
}

void sendSliderValues() {
  String builtString = String("");
```

```
A vázlat 5322 bájt (17%)-ot használ a program tárhelyből. A maximum 30720 bájt.
A globális változók 241 bájt (11%)-ot használnak a dinamikus memóriából, 1807 bájtot hagyva a helyi változóknak. A maximum 2048 bájt.
```

You will need the arduino IDE and the deej.exe for this. You can donload the IDE here, and deej here.

Program the arduino:

Plug the arduino (it can be fully built into the enclosure with all other hardware connected to it) in the computer and open arduino IDE. Paste the following code in it:

```
const int NUM_SLIDERS = 4;
const int analogInputs[NUM_SLIDERS] = {A0, A1, A2, A3};

int analogSliderValues[NUM_SLIDERS];
#include <Adafruit_NeoPixel.h>
Adafruit_NeoPixel strip(NUM_SLIDERS, 6, NEO_GRB + NEO_KHZ800);//pixel count ; pixel pin ; idc
int buttonState = 1;
int prevButtonState = 1;
bool off = false;

void setup() {
  for (int i = 0; i < NUM_SLIDERS; i++) {
    pinMode(analogInputs[i], INPUT);
  }

  Serial.begin(9600);
  strip.begin();
  strip.show();
  pinMode(2, INPUT_PULLUP);
  strip.setBrightness(255);//here you can set the LEDs' brightness (0-255)
}

void loop() {
  delay(10);
  buttonState = digitalRead(2);
  if(buttonState == LOW && buttonState != prevButtonState){//not the bestsolution, but works
    if(off)off = false;
    else off = true;
  }
  prevButtonState = buttonState;

  if(off){

    for(int i = 0; i < NUM_SLIDERS; i++){
      strip.setPixelColor(i,0,180,255);
    }
    updateSliderValuesToOff();
    sendSliderValues();
    strip.show();
  }else{
    for(int i = 0; i < NUM_SLIDERS; i++){
      int volume = map(analogSliderValues[i],0,1023,0,255);
      strip.setPixelColor(i,volume,(255-volume),0);
    }
    strip.show();
    updateSliderValues();
    sendSliderValues(); // Actually send data (all the time)
  }
}


void updateSliderValues() {
  for (int i = 0; i < NUM_SLIDERS; i++) {
     analogSliderValues[i] = analogRead(analogInputs[i]);
  }
}
void updateSliderValuesToOff() {
  for (int i = 0; i < NUM_SLIDERS; i++) {
     analogSliderValues[i] = 0;
  }
}

void sendSliderValues() {
  String builtString = String("");

  for (int i = 0; i < NUM_SLIDERS; i++) {
    builtString += String((int)analogSliderValues[i]);

    if (i < NUM_SLIDERS - 1) {
      builtString += String("|");
    }
  }

  Serial.println(builtString);
}
```

Select your board and COM port in tools, than upload the program with the big arrow.

Now check in serial monitor. If you get right values similar to the position of the sliders, then it's fine.

For people who are making this with more / less channels just modify these two lines:

```
const int NUM_SLIDERS = 6;
const int analogInputs[NUM_SLIDERS] = {A0, A1, A2, A3, A4, A5};
```

In the second line match the number of sliders with the pins (A0...A7)

Deej:

Actually you just have to download deej.exe and config.yaml. Put them in the same folder somewhere. When you're done with this, open the config file and set your COM port first (modify this line). This is the file where you can bind programs to sliders. There should be some example programs here already.

```
com_port: COM4 # set this to the port where your arduino is. This can be seen from the arduino ID
```

Here's an another example for mapping:

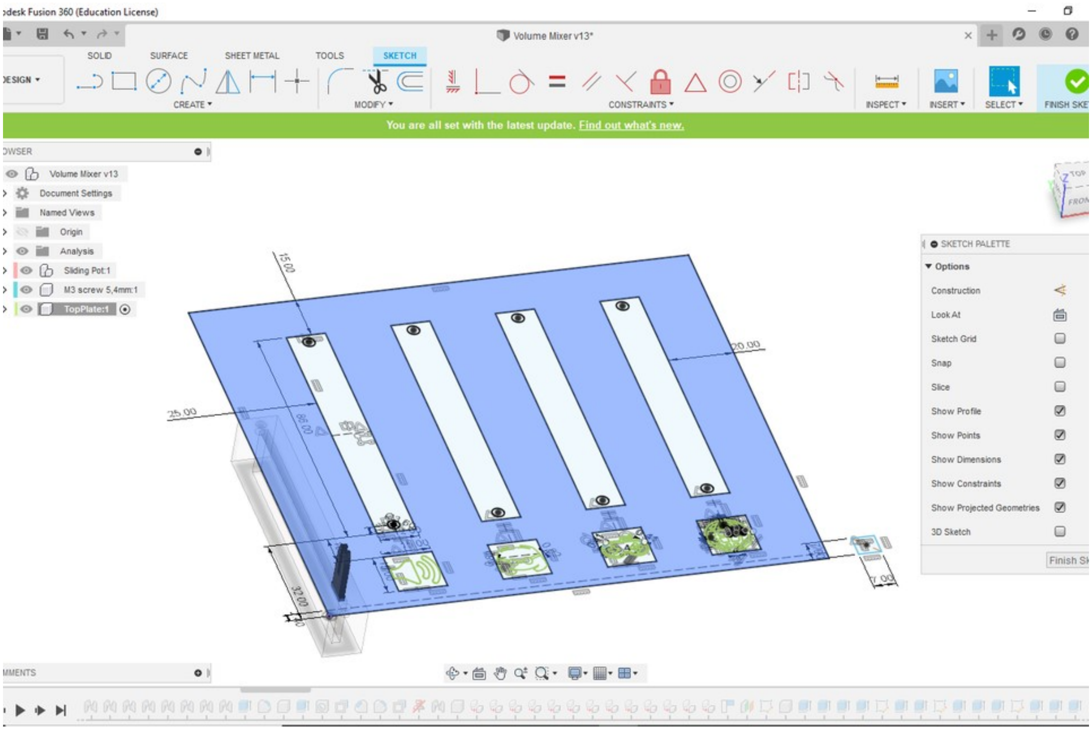
```
slider_mapping:
  0: master
  1: discord.exe
  2:
    - valorant.exe

  3:
    - chrome.exe
    - AvastBrowser.exe
```

If you wan't deej to auto start, put a shortcut of it into:

```
%appdata%\Microsoft\Windows\Start Menu\Programs\Startup
```

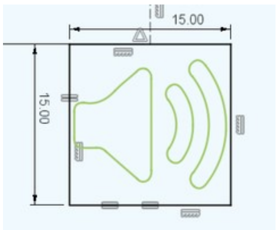
Step 6: Designing the Icons

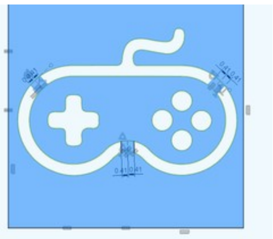


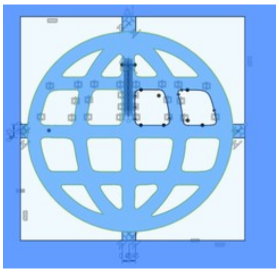
SKETCH PALETTE

Options

- Construction
- Look At
- Sketch Grid ☐
- Snap ☐
- Slice ☐
- Show Profile ☑
- Show Points ☑
- Show Dimensions ☑
- Show Constraints ☑
- Show Projected Geometries ☑
- 3D Sketch ☐







For those who built mine design you're done (I hope I didn't missed anything. And I'd love to see if you built this so share it with me!). Now it's about designing your own controller.

For the icon cutouts I made a 15 by 15mm square and then imported the .svg files and scaled them in the squares just by eye. Make sure the parts which you print are thick enough(X,Y) for the nozzle diameter. My top plate is 2.6mm for some structural strength, and to block light.

Some icons might need tabs to hold them in place(controller, browser for example). They are ,82mm thick for me. After you extruded the sketch you might want to use some fillets to make these tabs look better.

Now about the lamps. You either print mine, or make your own. Some help to the designing:

The diffusers are 0,6mm thick and about 15mm away from the led.

I've attached the .svg files for you. (oh, on my browser icon two sections are broken. Delete the broken ones, and mirror them from the other side)

I've also attached the fusion360 files so you can get some answers from there. Or just ask.

You should share your design with the deej community when you're done. Hope you like this project as much as I did.