
GUIDES FOR TACTONE

December 2024

**Table of
contents**

1. Intro	Page 2
2. Creating custom pads	Page 3
3. Changing the pads	Page 8
4. Altering the actuation	Page 10
5. Changing the audio file	Page 13
6. Handling the codebase	Page 16

1. Intro

Do you want to design the perception of your soft robot or shape changing interface, you can use Tactone to do so. With it's premade demo pads and corresponding sounds, you can start your design journey.

However, in order to explore more combinations, we will explain how Tactone's modular elements can be changed by you. In this way you can explore different combinations, upload new sounds and make your own demo pads.

CREATING CUSTOM PADS AND CASINGS

2.0

Creating custom pads

Ingredient List per pad:

- 60 g Ecoflex 30
(for a flat top layer)
- 220 g Eurosil 33

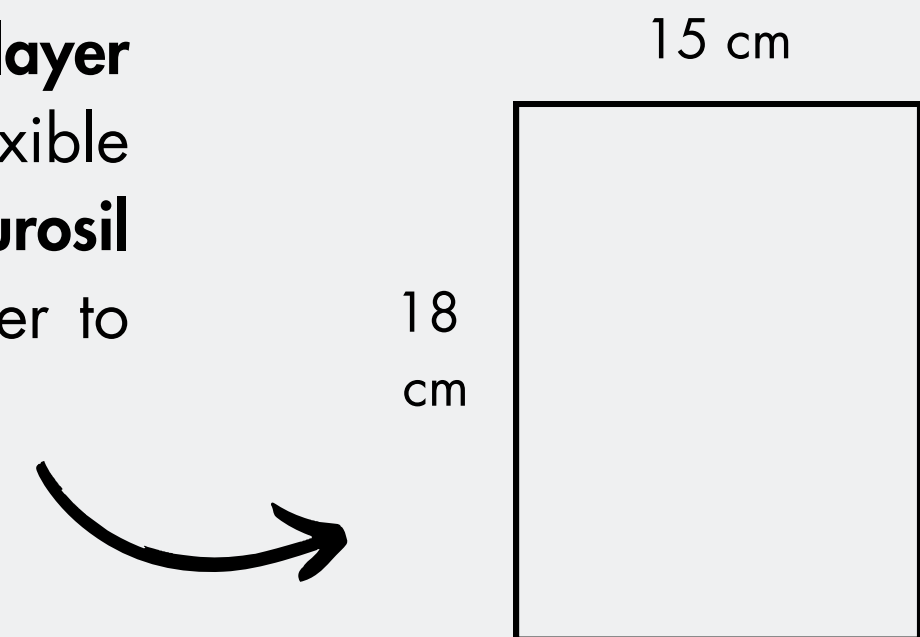
Tools:

- Vacuumchamber
- Weighing Scale
- Measuring Cup
- Something to stir with

Tactone consists of four demo pads, each allowing different tactile experiences, however, you can explore your own pads as well.

Practicalities

Each pad consists of three parts, a top layer, a middle layer, and a two-part bottom layer. For the **top layer Ecoflex 30** is used, as this silicone is more flexible allowing for more actuation. For the **other layers Eurosil 33** is used, as this silicone is more sturdy and harder to bend.

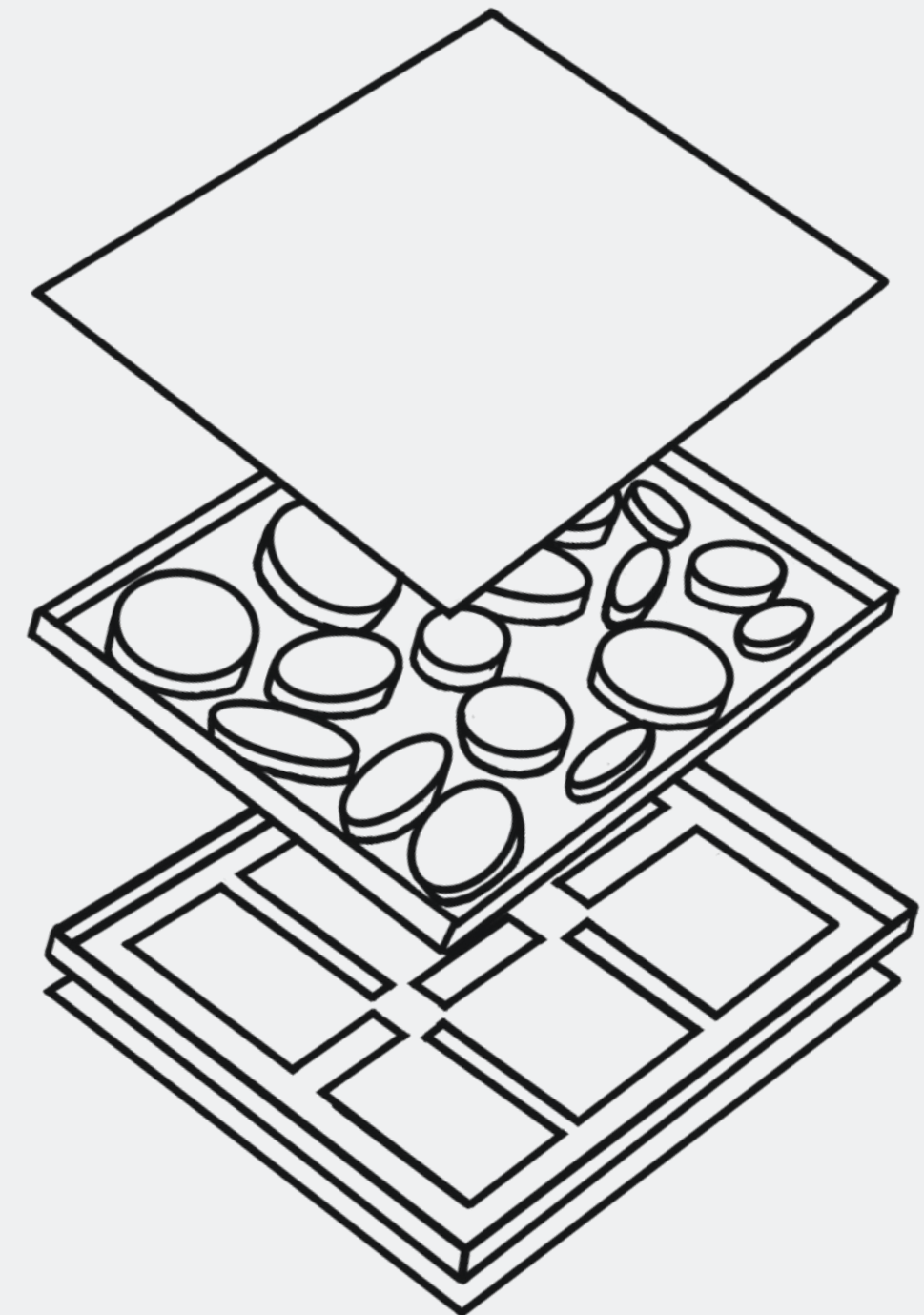


2.1 Design your pad

The top and middle layer allow customization of the pad's design. The middle layer creates air compartments, creating large or small bubbles in the top sheet. So, in the middle layer you can experiment with the size and shape of the bubbles.

For bubbles you can keep the top layer flat. For soft spikes you create a top layer with spike indents. The thickness of the top layer affects the actuation. We recommend not making the top layer thicker than 2mm.

The two bottom layers should not be changed to prevent the bottom from ballooning.



2.2 Casting process

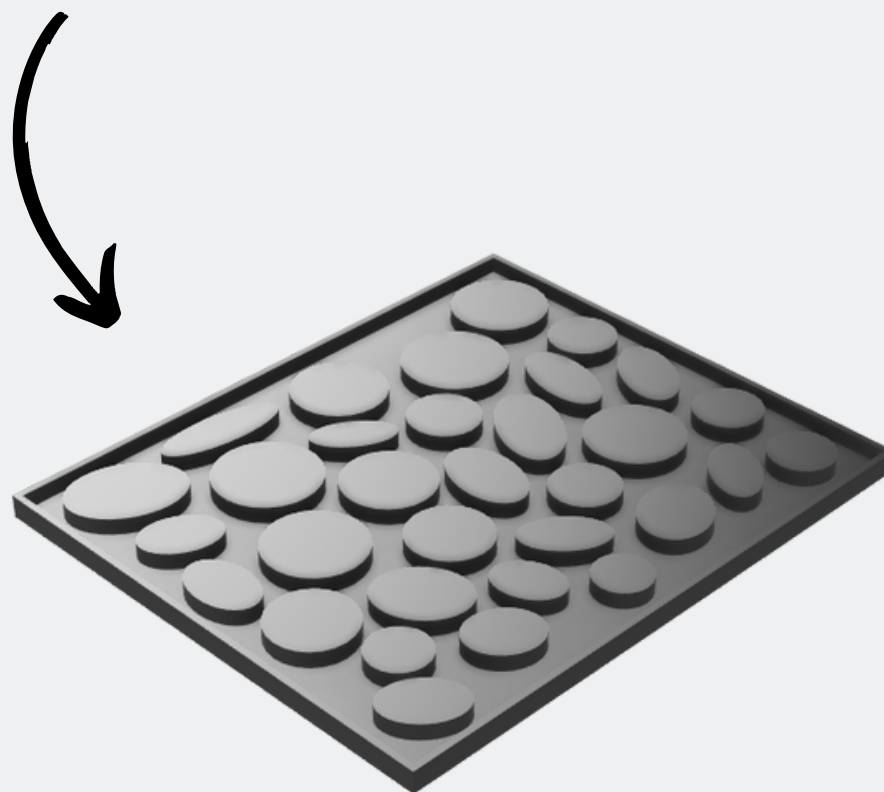
Each layer should be cast one by one. The mold files for these casting steps can be found in the GitHub.

[GitHub Link](#)

Note: After each step excess silicone should be cut away!

1. Middle layer

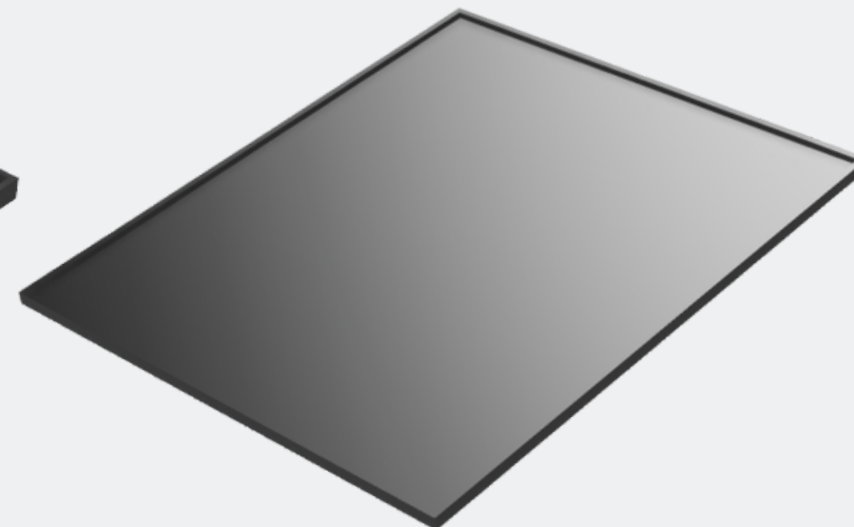
Pour the silicone in the middle layer mold, and let it set for 3 hours (the demolding time of the silicone).



2. Top layer

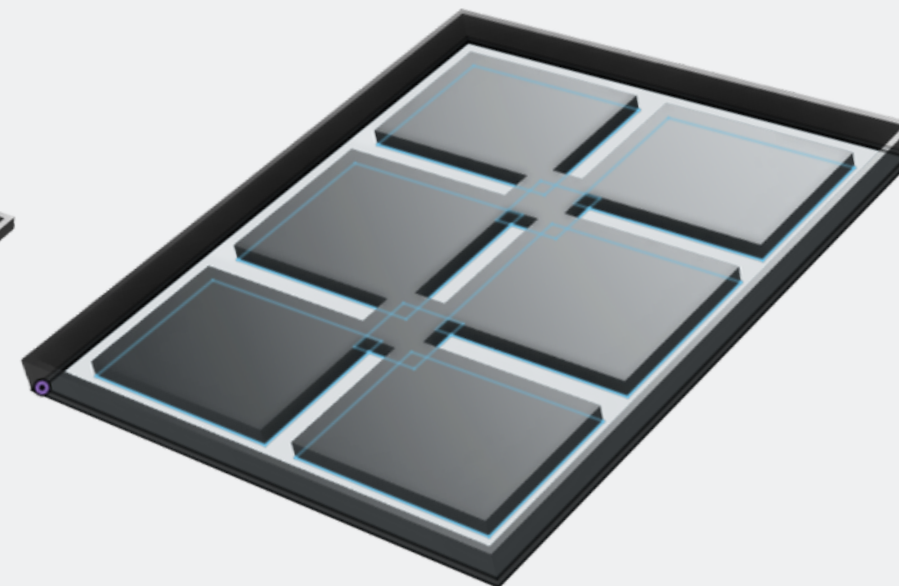
Pour Ecoflex 30 in the mold and immediately after finishing pouring, place the hardened middle layer on top.

Note: No weight should be put on top of the middle layer, as the middle layer then sinks too much into the top layer.



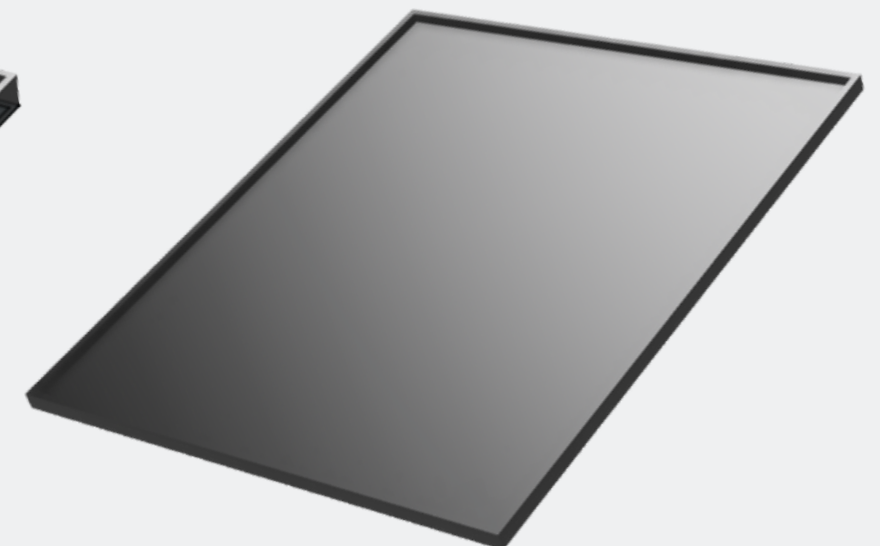
3. Bottom layer 1

Place the mold on a flat surface. Cast the silicone till the height of the inner segment. After you finished pouring, place the middle section on top.



4. Bottom layer 2

Pour the silicone, and place the pad on top. The walls don't need to be pressed hard into the bottom, just make sure all the walls and corners are inside the mold.



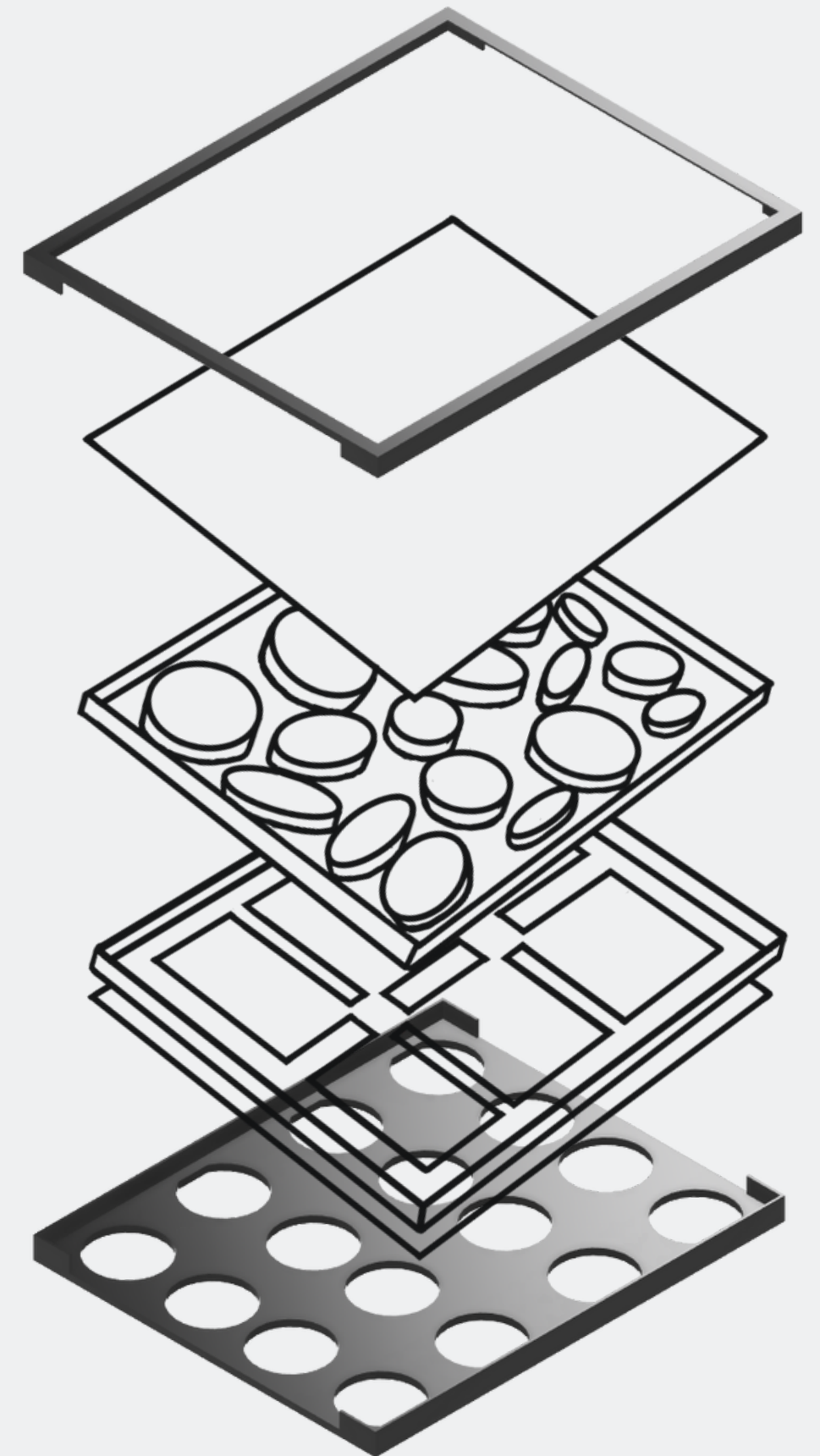
2.3

Casing measurements

Once the pads are fully cured you measure the thickness of the pad. Depending on the thickness you have to alter the 3D printed casing of the pad. As all the pads should be leveled, the thickness should maximally be 16 mm. So, if your pad is a bit thinner you need to print a thicker bottom layer, to level it out.

The 3D models of the casings can be found online as .stl's in the GitHub. You can download these models and print them. → [GitHub Link](#)

Seal the pad inside the two casings with superglue, to prevent ballooning of the pads when they are inflated. Now your pad is ready to be used.



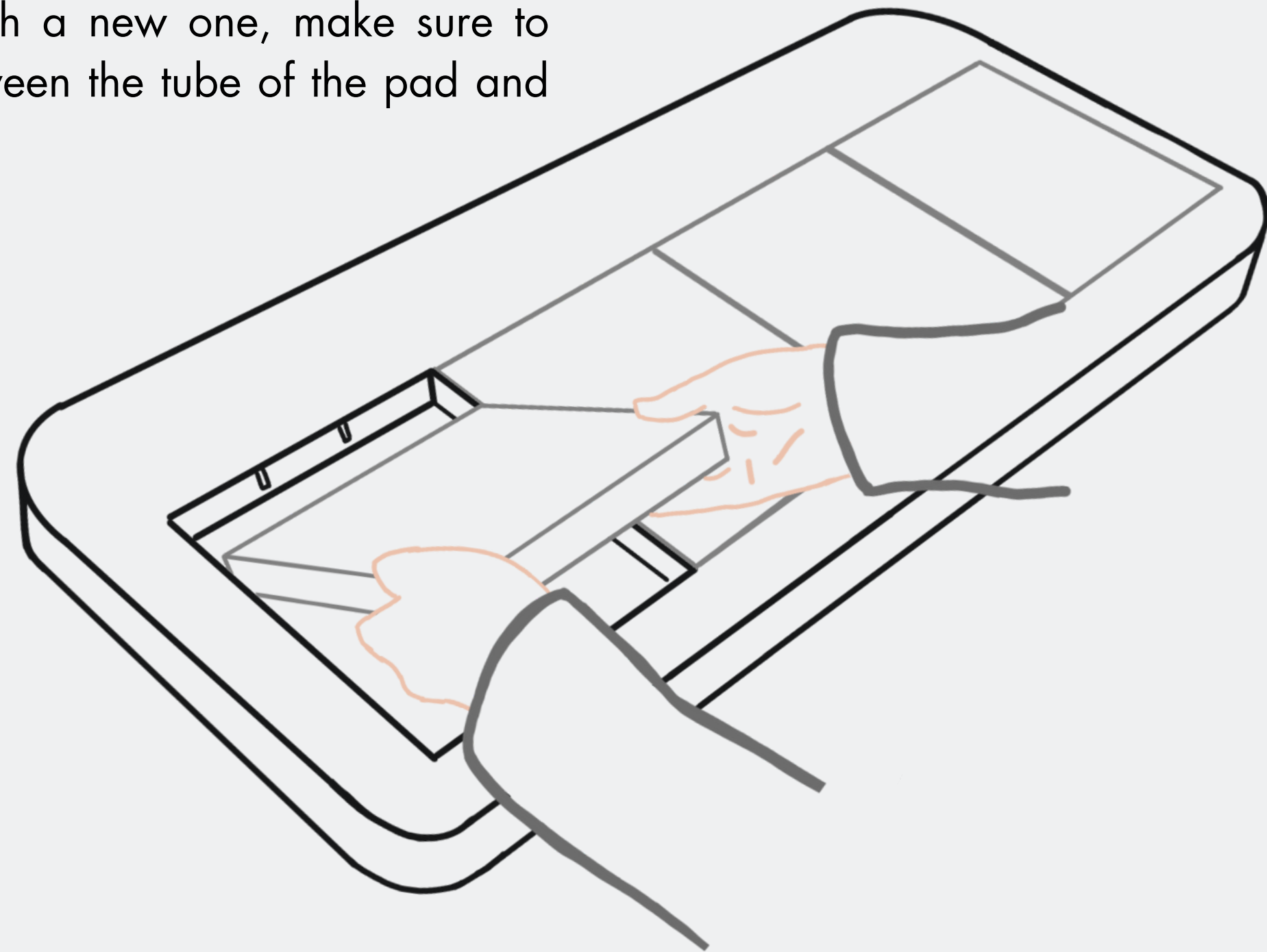
CHANGING OUT THE PADS

3.0

Changing the pads

The four silicone demo pads are put on a white MDF board above the circuit with a connection to an air pump.

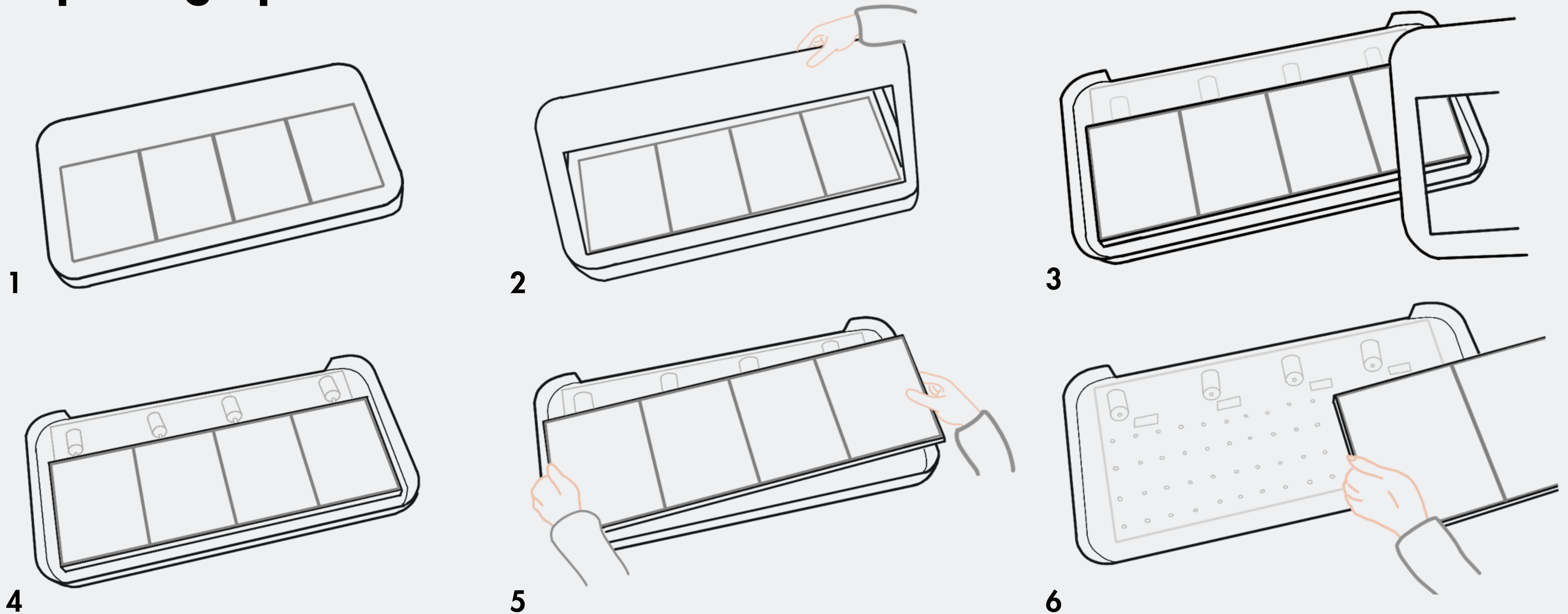
To change a pad, simply take out the pad from the box. When replacing the pad with a new one, make sure to make a tight connection between the tube of the pad and the air pump inside the box.



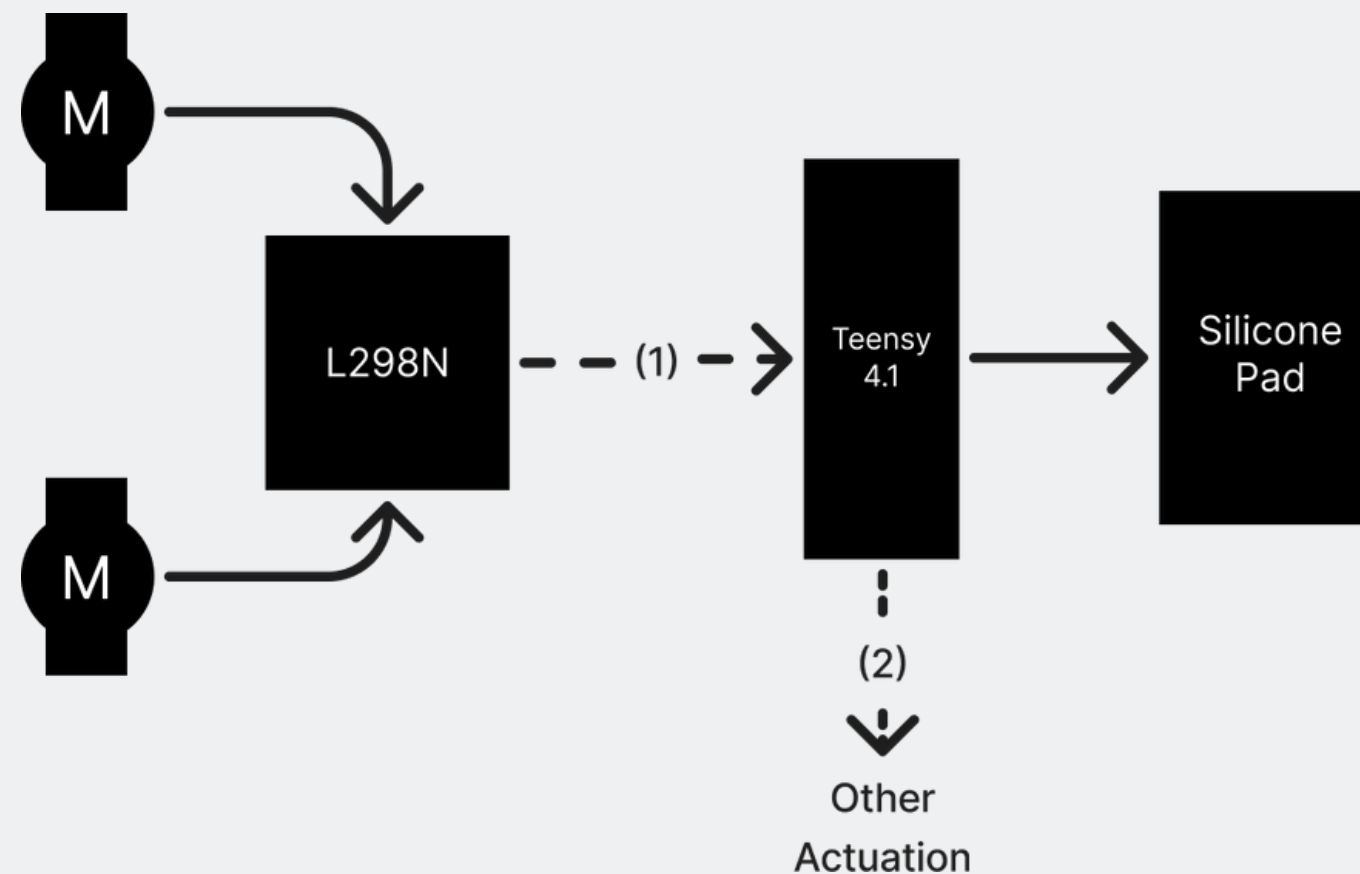
ALTERING THE ACTUATION

4.0 Opening up

To alter the actuation method, take off the top cover and take out the tray holding the silicone pads in place. Now the electronics are fully exposed and easy to adjust.



4.1 Altering the actuation



Currently 2 motor drivers control the 4 pumps used to inflate the silicone pads. If the new actuators can make use of the motor drivers (a component that requires about 12V), such as other dc motors for controlling vibration, the code does not have to be adjusted. Simply unscrew the air-pumps and connect the new actuators in their place.

It is also possible to hook up entirely different actuation (liquid crystals, etc.) by using any of the free pins on the Teensy (in the image above, unhook wire (1) and hook up wire (2)). This will include the pins now used by the motor drivers.

If you hook up an alternative actuation, or want different control logic, you will need to alter the software. Go to page 17 for more information on the software.

CHANGING THE AUDIO FILES

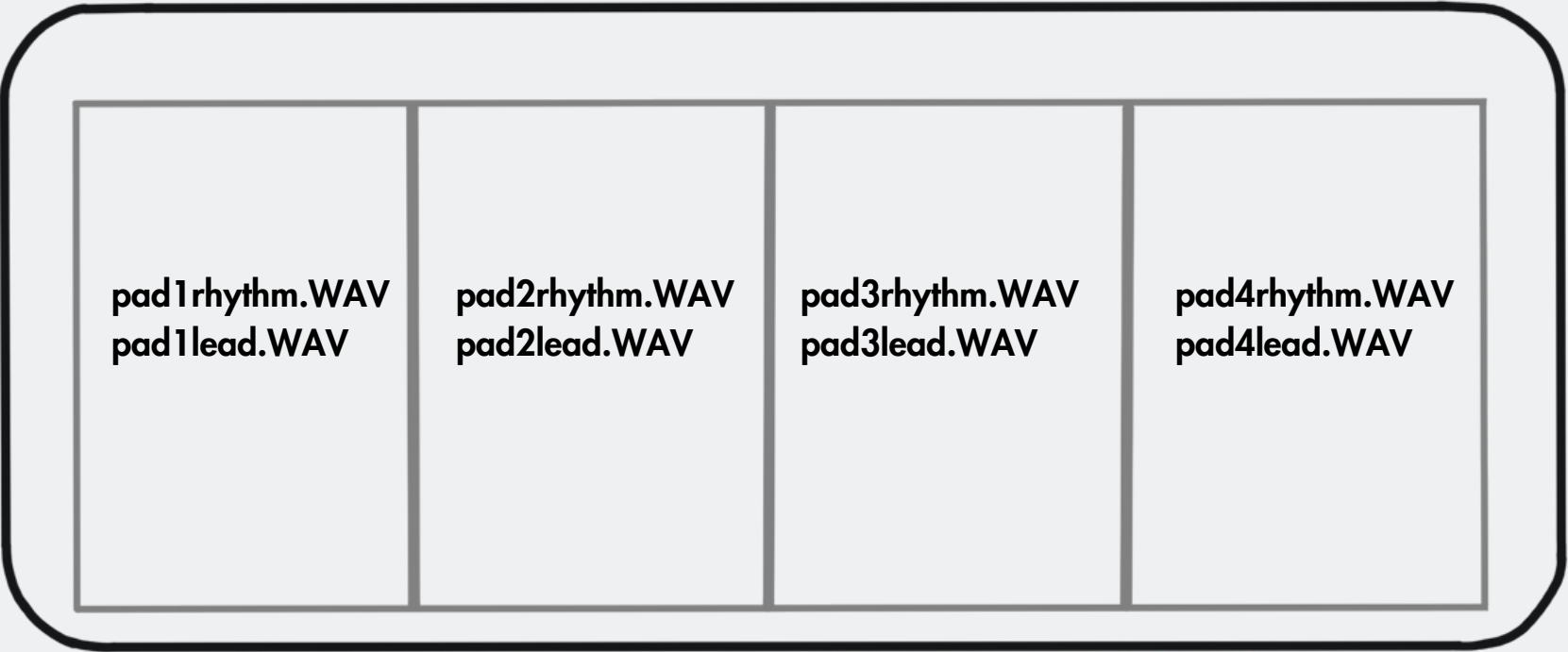
5.0

Creating audio

Audio can be either downloaded or created by the user themselves via a DAW of choice. It is important to note that the files are created as 16-bit WAV files, so the Teensy (the microcontroller) can read it. The files should be uploaded as:

pad*number*rhythm.WAV
pad*number*lead.WAV

Hint:
When creating your own audio pay attention to the volume level of both sound files, they should be about equal and not accentuate the same frequency to avoid sound clipping.

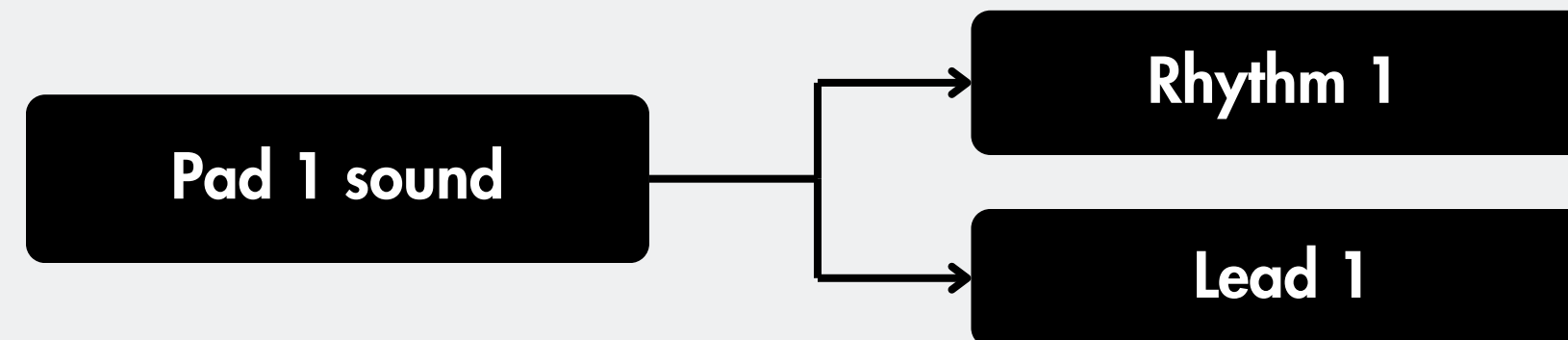


5.1 Uploading audio

The audio produced by Tactone can be uploaded via a micro-SD card found in the back of the artifact.

1. Remove the micro-SD card.
2. Insert the card into the computer.
3. Upload the audio files.
4. Reinsert the card into the Teensy audio shield.

The audios consist of 2 files per pad, a rhythm sound which will be played continuously while interacted with and a lead tone that will be modulated based on coverage.



HANDLING TACTONE CODEBASE

6.0 Setup

We recommend following this guide in the GitHub, for easier navigation. Otherwise all information can also be found here.

[GitHub Link](#)



To upload new code to Tactone, you will need the arduino IDE installed and add the teensy board manager.

[Arduino Software](#)



1. In arduino IDE go to file > preferences > Additional board managers URLS
2. Add the link that can be found in the Teensy documentation to download the board manager.
3. In the board manager window, download the Teensy board manager.
4. Select the Teensy 4.1 in the board manager to connect with the Teensy

[Teensy Documentation](#)



6.1

Changeable variables

It is relevant to know that there are a few variables that are open to change and finetune when using Tactone in a different setting.

'Lowerbound' and 'LDRtriggermap' are variables that can be changed dependent on the light level of the environment. If an environment is bright, the trigger value can be very high, and in a darker environment a lower trigger value can be used. We recommend having a direct light source above Tactone for stability.

```
//Line 15-18
int lowerBound = 400;

int LDRtriggermap1[] = { lowerBound, ... };
int LDRtriggermap2[] = { lowerBound, ... };
int LDRtriggermap3[] = { lowerBound + 5, ... };
int LDRtriggermap4[] = { lowerBound, ... };
```

'Filelist' is a list of audiofiles that are loaded on the SD card. This list consists of 8 files. Two files for each of the pads, a base rhythm and a lead. For further explanation of how to establish the audio files, check previous guide on audio.

```
//Line 22-24
const char* filelist[8] = {
  "HAPPYRITHEM.WAV", "HAPPYLEAD.WAV",
  "CALMRITHEM.WAV", "CALMLEAD.WAV",
  "SADRITHEM.WAV", "SADLEAD.WAV",
  "STRESSRITHEM.WAV", "STRESSLEAD.WAV"
};
```

6.2 Alternative actuation methods

We as authors trust that if you want to hook up a differing actuator, you have the relevant knowledge of pin declaration, and the relevant software.

If you want the actuation to be triggered by use of the pads. You can use the switch case in the void loop() function.

Case -1 is active when no pads are being triggered, also referred to as an 'idle' state. The other cases represent the active pads.

```
//Line 260-307
switch (activeMultiplexer) {
  case -1:
  {
    deactivateAllPumps();
    Serial.println("No multiplexer activated.");
    idleLED();
    ESNumber = -1;
  }
  break;
  case 0:
  {
    ColorAllLeds(255, 200, 0);
    volumeMap();
    activatePump(0, enA1, in2_1, 255, 2000,
2000);
    printActiveMulti(currentCaseTime, "0",
coverage);
    ESNumber = 0;
  }
  break;
  ...
}
```

TACTONE
