Constructing A Virtual Tasting Device

Lukas Graf
November 6, 2020

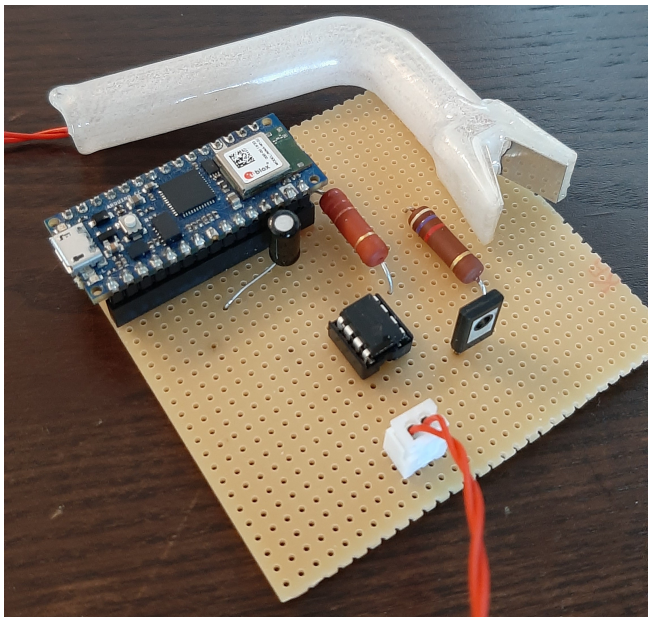


Figure 1: Photo of the completed prototype

Contents

1 Introduction

The virtual tasting device is a system designed to digitally simulate taste sensations, by supplying electricity to the tip of the tongue, through silver electrodes. There is already some work, creating such taste interfaces, for example by Nakamura and Miyashita [3] [4], and by Ranasinghe et al. [8] [5] [7], which I used as a base to build my own taste interface.
Some existing approaches also try incorporating other stimuli, like thermal stimuli, or smell sensations. My approach however will only focus on the electrical taste stimulation.

2 Components

The Hardware of the prototype can be categorized into two parts: the mouthpiece, which holds the electrodes used to apply the voltage to the user's tongue, and the control module, which regulates the voltage delivered to the electrodes.

2.1 Control Module

The design of the control module is taken from an approach by Adrian David Cheok and Kasun Karunanayaka [1], where the control module is similar to, but a bit simpler than the one by Ranasinghe [8].

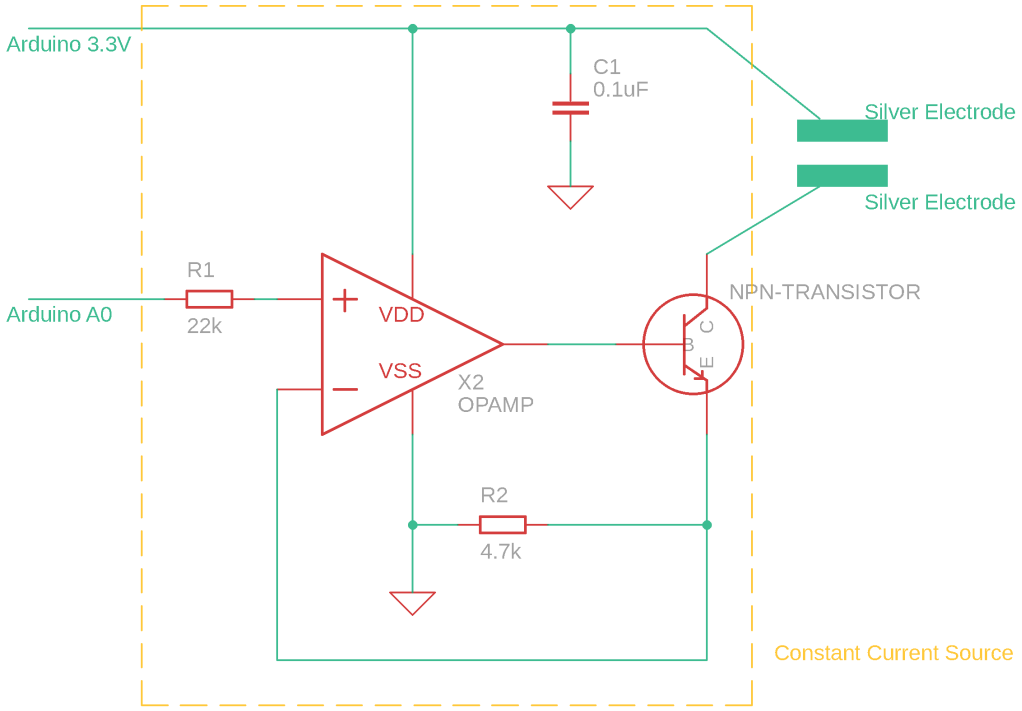


Figure 2: Control module schematic, based on Fig. 4.4 on page 57 in [1]

The control module consists of a microcontroller and a constant current source circuit. The microcontroller (an Arduino 33 IOT) receives operating instructions via a serial connection over USB and then sends PWM square wave pulses through the constant current source circuit to the users tongue, via two silver electrodes on the mouthpiece.
The resistance of the tongue can however vary from person to person and it can also vary depending on the location and the amount of saliva. Therefore the constant current source is implemented to compensate for these variations in resistance and still consistently deliver the same current, when running on the same device settings.
The output current is in a range of 0-200μA, which is achieved using a digital-to-analog converter

(DAC) on the Arduino. This magnitude of current is then output in PWM pulses with a frequency range of 50-1500Hz. This combination of a DAC and PWM is used, because both, magnitude of current, as well as the frequency, can influence the taste sensation.

2.2 Mouthpiece

The design of the mouthpiece is based on a straw-like design by Nimesha Ranasinghe et al. [5] [6], with a few adjustments, like removing the actual straw functionality.

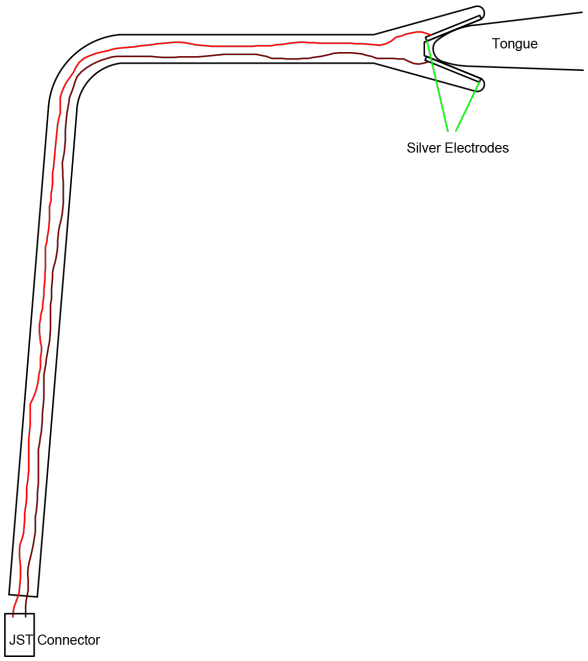


Figure 3: Sketch for the tongue interface (cross-section)

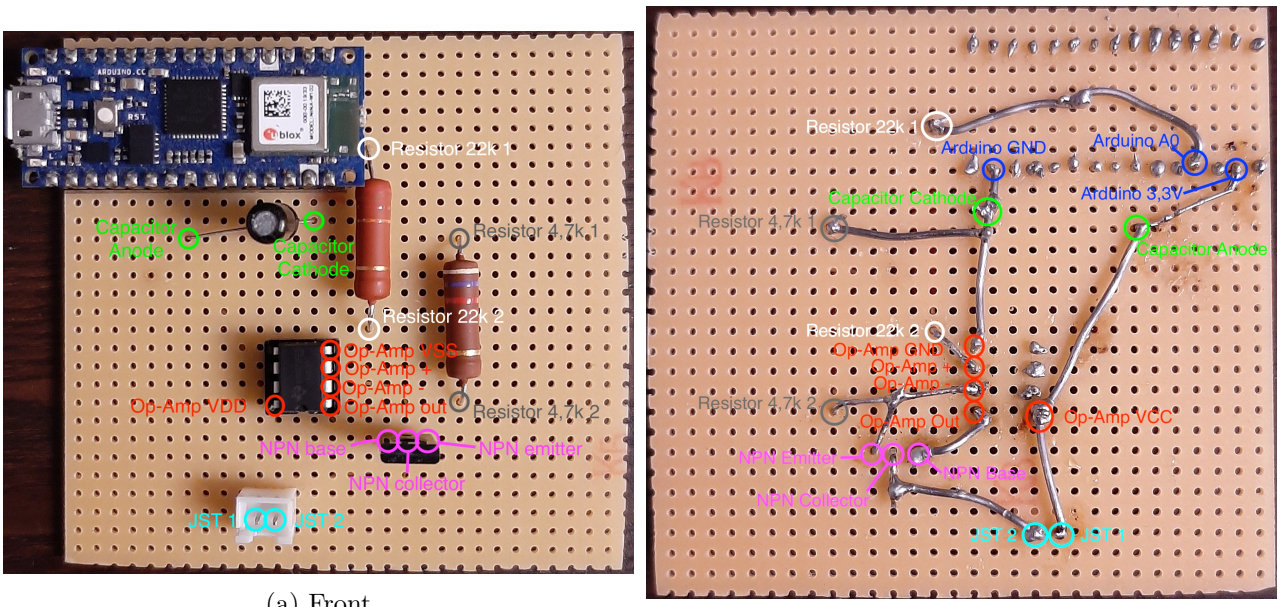
The mouthpiece consists of a 3D printed plastic casing with two silver electrodes. The user's tongue has to be put between the two electrodes, so that electricity can flow through the tongue. The wires connecting the electrodes to the control unit run inside the plastic casing and have a JST connector at the end, to make it easy to disconnect the mouthpiece.

3 Construction

3.1 Control Module

Parts

- Microcontroller Arduino Nano 33 IoT
- Op-Amp
- NPN Transistor
- 22kΩ Resistor
- 4.7kΩ Resistor
- 0.1μF Capacitor
- Perfboard
- JST-XH Connector



(a) Front

(b) Back

Figure 4: Arrangement of the components on the perfboard

Steps for assembly:

1. **Soldering**
   All the parts need to be soldered together, according to the schematic. The following figures show my placement of the parts on the perfboard, so that there are no overlappings. It would also be possible to design a PCB for this, to reduce the size of the circuit.

2. **Upload Arduino code**
   The code must be uploaded to the Arduino microcontroller, for example using the Arduino IDE. **But be sure to never use the device while uploading code, for safety reasons.** For explanation of the Arduino code see section 4: Device Firmware or the comments in the code.

3. **3D-Print The Case**
   There is a 3D design for a case for the control module, which can be printed and then assembled, using 2,9x6mm screws. This is important, when using the device with VR, as you can attach some kind of clamp to the case, to fix the case somewhere onto the person using the VR Headset. This has the advantage, that the range of motion is then only restricted by the usb cable connecting to the Arduino.

3.2 Mouthpiece

Parts

- 3D printed casing
- 2 Silver plates (at least silver 925/sterling silver), dimensions 12x10x0.5mm
- Wire (at least 25cm)
- Epoxy resin

Steps for assembly:

1. **3D print the plastic casing**
   The model was created using FreeCAD[1] and can be printed with almost no support structures, with the right orientation.

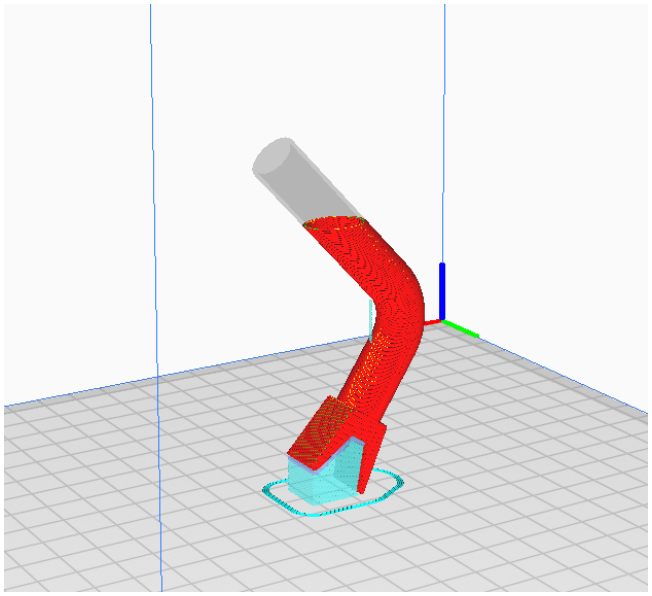


Figure 5: 3D model, sliced with support only on the outside, so it is easy to remove

2. **Coat the printed piece**
   As there could be health risks with directly putting the 3D printed piece in your mouth, it should be coated in epoxy resin. Around 3 rounds of applying the resin will be necessary, until the whole piece is covered enough. This is because the resin will only slowly start to harden after around an hour, so a lot of it will run off the print.

3. **Put in the wires**
   Once the epoxy is hardened, it is time to put in the wires. Getting them all the way through can be a bit fiddly, however using some harder garden wire like a cable puller can help.
   With the wires through you can solder the ends onto the electrodes, using some extra flux. After that, attached the electrodes to the mouthpiece using some more epoxy resin or glue.

4. **Connect to the control unit**
   On the other end of the wires put a JST-XH connector to connect to the control unit.

4 Device Firmware

The Arduino microcontroller runs a program, that reads incoming commands over a serial connection and according to these commands, it outputs a signal to the mouthpiece. The combination of a PWM signal and the DAC of the Arduino is achieved, by implementing the PWM via an interrupt timer system, because normally no timer/counter can be connected to the same pin as the DAC, on the Arduino's SAMD21G chip.

4.1 Communication Protocol

The device awaits a single line string of the following format:

[1] `https://www.freecadweb.org/`

```
command_nr dac_value pwm_duty_cycle pwm_frequency
```

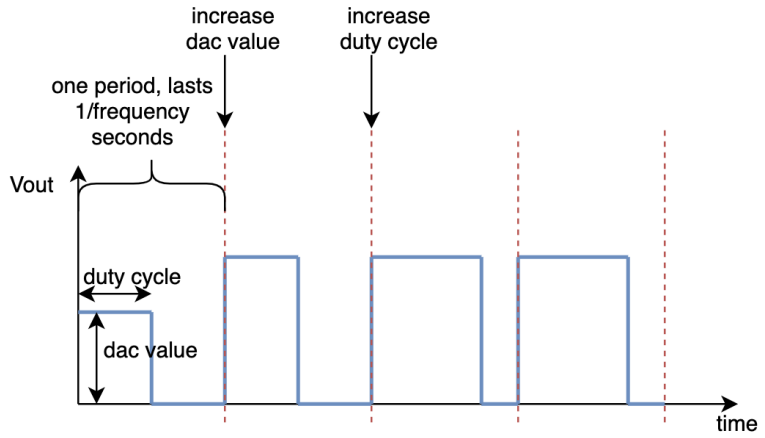


Figure 6: Sketch of a PWM example, showing how the variables control the signal

That is the following four variables, separated by spaces:

- **command_nr:**
  Integer; Any number to identify the current command. Will be included in the response, so the client can correctly match the response to the corresponding command.
- **dac_value:**
  Integer, [0,100]; The value to use for the DAC. Will be mapped to [0,1024) (10-bit), where 0 is 0V output and 1024 is full 3.3V output.
- **pwm_duty_cycle:**
  Integer, [0,100]; The duty cycle used for PWM in percent.
- **pwm_frequency:**
  Integer, [0,100]; The frequency for PWM. Will be mapped to [50Hz,1500Hz]

The response from the Arduino is a line of the following form:

```
command_nr "success"
```

in the case of a successful command, or the following if an error occurs, because of an invalid input:

```
command_nr "err" error_message
```

Where command_nr is the command_nr of the incoming command and error_message is a short text, explaining what kind of error occured.

5 Unity API

The Unity API allows to easily use the device from within unity, by sending commands over a serial connection. It relies on Ardity (`https://ardity.dwilches.com/`) for communication over the serial port.
It has been tested to work with Unity versions 2019.4.13f1 (LTS) and 2020.1.11f1 and with Ardity version

5.1 Getting started

1. Download the Ardity Unity Package.
2. Import the Ardity Unity Package and the TastingDevice Unity Package.
3. If there is an error, that the namespace Ports can not be found in System.IO, then change the Api Compatibility Level to .NET 4.0 under Edit -¿ Project Settings -¿ Player -¿ Other Settings.
4. Put the TasteController script on a GameObject, and configure it, by specifying the correct port name for the Arduino.
5. Now you can retrieve the taste controller from some script and start the device.

```
TasteController tasteController = gameObject.GetComponent(typeof(TasteController)) as TasteController;
tasteController.Activate(25, 25, 25);
```

5.2 API functions:

- **void Activate(int dacValue, int dutyCycle, int frequency, CallbackFunction onSuccess=null, CallbackFunction onFailure=null, CallbackFunction onTimeout=null)**
  Activates the device with given dacValue, dutyCycle and frequency. CallbacksFunctions have to be functions without return type, that take a single string, which contains a more detailed message of what happened.
  If callbacks are supplied, they are executed on receiving successful response from the Arduino, on failure, or on timeout while waiting for the response from the Arduino. If no callbacks are provided, nothing happens on success, while failures and timeouts will be logged.
- **void ActivatePreset((int, int, int) preset, CallbackFunction onSuccess=null, CallbackFunction onFailure=null, CallbackFunction onTimeout=null)**
  Activates the device with a given preset, which is just a tuple of dacValue, dutyCycle and frequency. The following default presets are available:
  TasteController.presetSour – 50, 50, 50
  TasteController.presetBitter – 100, 50, 0
  The callbacks work the same as with Activate.
- **void Deactivate(CallbackFunction onSuccess=null, CallbackFunction onFailure=null, CallbackFunction onTimeout=null)**
  Deactivates the device, by setting dac value and duty cycle to 0. The callbacks work the same as with Activate.

6 Conclusion

The prototype works, however there were some findings, which could maybe be improved in a new version.
First of all, 3D printing the mouthpiece and coating it in epoxy resin was more work than expected and it is almost impossible to apply to coating completely evenly. Maybe the health risks could be overcome in other ways, using different printing materials or techniques. Or maybe a different production process could be used, like casting the mouthpiece in a mold.
Another weakness of the prototype is, that it is hard to reliably create any sensation other than sourness. This limitation also applies to other electric taste based systems, like the ones, that this prototype was based on [8] [1]. There is however one approach to mildly improve this, by using a relay to have the option of inverting the flow of electricity. This has lead to some reports of sweet and bitter taste in experiments conducted by Ranasinghe [8].

Also during the development of this prototype, Miyashita published the norimaki synthesizer [2], which takes a different approach to taste stimulation, by using actual chemicals responsible for the taste sensations, instead of using only electricity. This approach probably has the advantages, that any of the five basic tastes can be replicated and the taste sensation can be controlled more precisely.

References

[1] Adrian David Cheok and Kasun Karunanayaka. *Virtual Taste and Smell Technologies for Multisensory Internet and Virtual Reality*. Springer, 2018.

[2] Homei Miyashita. Norimaki synthesizer: Taste display using ion electrophoresis in five gels. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, page 1–6, New York, NY, USA, 2020. Association for Computing Machinery.

[3] Hiromi Nakamura and Homei Miyashita. Augmented gustation using electricity. page 34, 01 2011.

[4] Hiromi Nakamura and Homei Miyashita. Development and evaluation of interactive system for synchronizing electric taste and visual content. *Conference on Human Factors in Computing Systems - Proceedings*, 05 2012.

[5] Nimesha Ranasinghe, Kuan-Yi Lee, and Ellen Do. Funrasa: An interactive drinking platform. pages 133–136, 02 2014.

[6] Nimesha Ranasinghe, Kuan-Yi Lee, Gajan Suthokumar, and Ellen Do. Virtual ingredients for food and beverages to create immersive taste experiences. *Multimedia Tools and Applications*, 75, 01 2016.

[7] Nimesha Ranasinghe, Thi Ngoc Tram Nguyen, Yan Liangkun, Lien-Ya Lin, David Tolley, and Ellen Yi-Luen Do. Vocktail: A virtual cocktail for pairing digital taste, smell, and color sensations. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, page 1139–1147, New York, NY, USA, 2017. Association for Computing Machinery.

[8] R. A. Nimesha Ranasinghe. *Digitally stimulating the sensation of taste through electrical and thermal stimulation*. PhD thesis, 2012.