LipSync

DESIGN RATIONALE

Outline

This Design Rationale is intended to provide designers and makers information about the design process and design decisions behind the development of the LipSync, a mouth-operated sip-and-puff joystick that can emulate a mouse, wireless mouse, or gamepad.

Table of Contents

Introduction

The LipSync is a mouth-operated sip-and-puff joystick that can emulate a mouse, wireless mouse, or gamepad. It enables people with disabilities that limit mobility to control a computer, gaming console, smart phone, or tablet.

The LipSync is intended for people who have limited hand function and some head and neck movement, such as a person with:

- high-level paraplegia
- bilateral arm amputations
- muscular dystrophy
- amyotrophic lateral sclerosis
- multiple sclerosis

Previous LipSync Designs

The Neil Squire Society first began development of the LipSync in 2004. Since that time, the design has gone through multiple iterations. The earliest version of the device was traditionally manufactured and later it was re-designed into a 3D printable version that could be made by volunteer makers in the community. The maker-friendly version was updated several times based on feedback from users, disability professionals, and makers.

Aluminum LipSync

The original LipSync began development in 2004. The device was comprised of an aluminum body manufactured using CNC machining. The enclosure had a smooth, organic shape specifically intended to look sleek and futuristic and less clinical and clunky.

Figure 1: Aluminum LipSync. 2004

Functionally, the device emulated a USB HID Mouse. Two buttons on the rear of the enclosure provided the ability to change cursor speed and initiate a calibration. The mouthpiece section covered a replaceable filter.

3D Printable LipSync

A 3D-printable version of the LipSync was designed and released in 2016. This version was designed to be made by a volunteer maker and take advantage of the growing availability of alternative small scale manufacturing options like small, consumer-level 3D printers, inexpensive custom printed circuit board (PCB) services, and open source microcontrollers.

This version of the LipSync was comprised of off-the-shelf, commercially available electronics and hardware, three custom PCBS, and several 3D printed components. The sip and puff joystick was a novel design based on a similar design created by the Asterics Foundation for the FlipMouse.

Figure 2. Cross-sectional view of early 3D Printable LipSync. 2016.

Enclosure

The enclosure was designed to mimic the size and shape of the original aluminum LipSync. The enclosure consisted of a front and rear housing that contained and covered the internal components of the LipSync.

Joystick

The joystick of the LipSync consisted of the mouthpiece, filter, airpath, and movement mechanism. The mouthpiece and filter were commercially available medical components with Luer lock connections. The airpath lead from the mouthpiece, through the filter, to a pressure sensing board. The movement mechanism consisted of four force-sensitive resistors (FSRs). The resistors could sense force in one direction only, so were arranged on the four cardinal directions (up, down, left, and right). There were also four springs which would centre the joystick when it was released by the user.

Electronics

The LipSync used an Arduino Micro microcontroller for its brains, a pressure transducer to measure sips and puffs, and FSRs to measure the position of the joystick. All electronic components were connected through the three custom PCBs.

Functionality

The LipSync emulated a USB HID mouse or a Bluetooth HID mouse. There were also two buttons at the back to allow users to calibrate the device and change cursor speed. Most LipSync users could not access and use these buttons independently. A bi-colour LED provided feedback to the user.

Later, a version of the code was developed to allow the LipSync to emulate a USB HID Gamepad.

LipSync V3

This version of the LipSync introduced several incremental changes from the previous version including an updated enclosure and an overhauled firmware.

Enclosure

The enclosure of the LipSync V3 was updated by adding a low poly texture to the outside. The low poly texture masked the layer lines of the print and made the device seem more professional. There were also some improvements to the design to make it easier to print.

Joystick

The joystick mechanism was unchanged from the previous version.

Electronics

The electronics were unchanged from the previous version.

Functionality

The LipSync V3 could emulate a USB HID mouse, a Bluetooth HID mouse, or gamepad. Due to memory limitations on the microcontroller, the LipSync V3 could only be programmed as one of the three options. The firmware had to be changed to change operating mode of the LipSync V3.

This version also added an application programming interface via a serial connection that allowed some of the settings to be adjusted through a computer without having to modify the code.

User Feedback

At least 1,400 of the 3D printable LipSyncs have been built and used. The most common feedback is summarized below:

- The joystick is too stiff / requires too much force to move.
- The cursor can drift
- The device is too large
- Some users are unable to use sip and puff at all, or one or the other as effectively
- The mouthpiece was uncomfortable to use with the bottom lip

New Version Goals

The following goals were created based on user, maker, internal, and stakeholder input:

1. Decrease the force required to operate the joystick
2. Decrease or eliminate cursor drift
3. Improve the user experience
4. Update the hardware
5. Add integrated Bluetooth
6. Create an intuitive settings manager the main user can access and control
7. Make the LipSync easier to build
8. Add alternate controls for sip and puff

The redesign of the LipSync needed to meet these goals, or have a plan to meet them with future iterations.

Research

There are several commercially available and DIY designs that provide similar functionality to the LipSync. This section provides a summary of these designs.

Commercially Available Sip and Puff Joysticks

The main commercial options for sip and puff joysticks are summarized in the table below.

Table 1: Commercial Option Summary

Name	Cost	Link
Jouse	$1495 CAD	https://www.compusult.com/assistive-technology/our-at-products/jouse3
QuadLife	$1190 USD ($1618 CAD)	https://quad.life/
Quadstick	From $449 USD ($611 CAD)	https://www.quadstick.com/
IntegraMouse	$3300 CAD	https://www.integramouse.com/startseite/
Celtic Magic Feather	$600 CAD + international shipping	https://www.celticmagic.org/feather

Jouse

Joystick operated, plug and play mouse control.

Price

$1495 CAD

User Interface

- Mouth joystick
- Pressure sensing

User Feedback

- Two LEDs on control unit
- DIP switches on control unit

Mouthpiece

- Short/long dental straw

- Short/long hard plastic
- Both installed into metal adapter tube

Switch Inputs

- Two 3.5mm jacks

Mounting

- Manfrotto-style Magic arm with custom clamp (Jouse 3, Jouse+)
- Gooseneck (JouseLite)

Enclosure

- Pros
  - Multiple versions of device (3, +, and Lite)
- Cons
  - Device is slightly bulky
  - Core of device in line of sight
  - Large mount

Connection

- USB connection
- HID mouse output
- Joystick output

Cleaning

- Wash the mouthpiece with warm soapy water

Alternate Versions

- Jouse 3
- JouseLite
- Jouse+

Link

https://www.compusult.com/assistive-technology/our-at-products/jouse3

QuadLife (formerly QuadJoy)

Figure 3. QuadLife Unit.

Figure 4. QuadLife Mouthpiece

Mouth mouse device that allows the user hands free operation of smart phones, PCs, and smart TVs

Price

$1190 USD

User Interface

- Joystick for mouse control
- Sip and puff for inputs

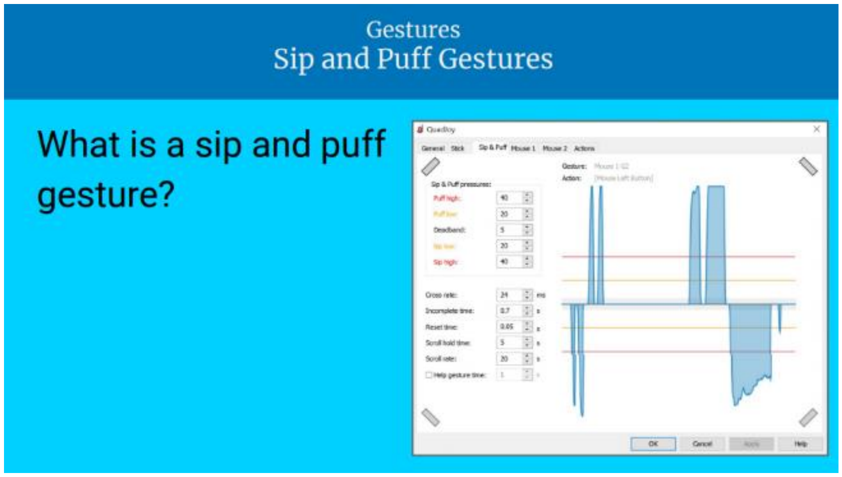


Figure 5. QuadJoy Sip and Puff Gestures.

User Feedback

- LEDs built into case

Mouthpiece

- Custom stick kit, starting from $20 USD

Switch Inputs

- Separate external switch interface
- 3 3.5 mm jacks

Figure 6. QuadLife external Switch Interface.

Mounting

- Goose neck with Super clamp or C clamp

Enclosure

- Pros
  -
- Cons
  - Bulky box
  - Clear plastic face, looks like an electronic device

- Mouthpiece has yellowish tinge
- Device in line of sight
- Tube looks like medical device

Connection

- Bluetooth (Android, Windows, Apple Max, iOS Version 13+)
- USB (Windows Devices, Apple Mac)
- Infrared (Internet ready TVS, Devices that can be controlled by an IR remote)

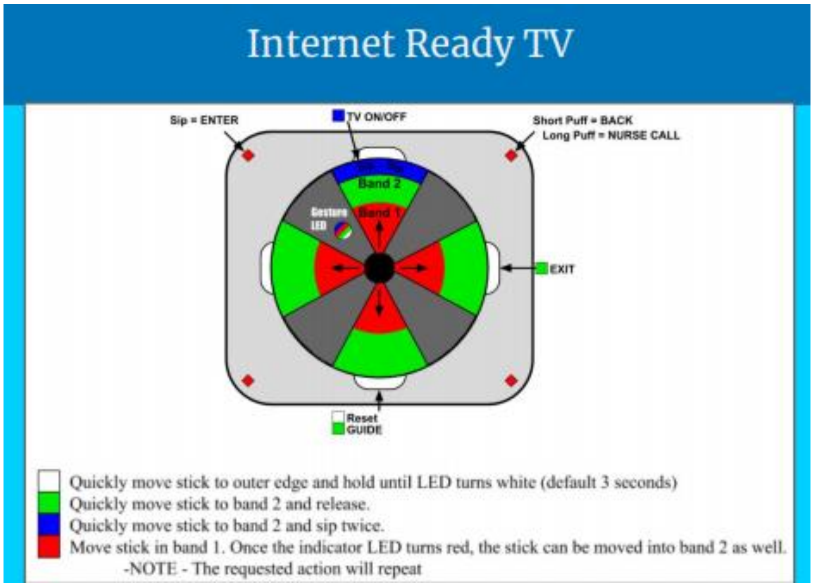


Figure 7. QuadLife Quadrant Interface.

Cleaning

- To remove the stick, firmly hold the base of the magnet holder and gently twist the stick counterclockwise.
- Remove the white spongy saliva filter from the stick.
- Clean the stick in the dishwasher or by hand

Alternate Versions

- Quadlife 4 mouse only

Link

https://quad.life/

Quadstick

https://www.quadstick.com/

Figure 8. QuadStick FPS Game Controller

Figure 9. QuadStick Singleton

Figure 10. Original Quadstick Game Controller.

A game controller for quadriplegics

Price

From $449 USD

User Interface

- Joystick: Mechanical potentiometer
- Pressure: digital; +/- 2kPa

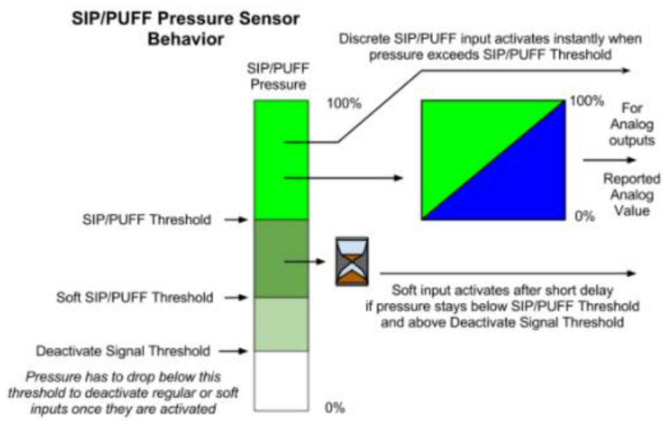


Figure 11. QuadStick Sip and Puff Explanation.

User Feedback

- Visual Feedback
    - Puff- Red light
    - Sip – Blue light
- Audio Feedback
    - Hard sip/puff – click
    - Soft sip/puff – beep

Mouthpiece

- Mouthpiece
    - 3D printed
    - 10-32 UNF Thread, ¼” Hex, Barbed 3/32” ID Tubing, white nylon
    - Nylon thumb screw: 10-32, ½” long
- Pressure tubing
    - 3/32” ID, 7/32” OD, 1/16” Wall x 25 feet: https://www.amazon.com/gp/product/B000FMYW5W
    - NSF-51 food equipment standard
- Filter
    - Syringe Filter, PTFE, Hydrophobic, 0.22 µm Pore Size, 13 mm diameter

Switch Inputs

- Default two external switches with one lip sensor
- Optional two more switch inputs
- Optional two relay outputs and two switch inputs

Mounting

- 1” Size B RAM Mount System
- Magic Arm
- Flexible mounting arm

- Articulated mounting arm

Enclosure

- Pros
    - Various mouthpieces
    - Removable mouthpieces
    - Various versions of device
    - Wide mouthpiece
- Cons
    - Bulky due to light attachment
    - Device is slightly bulky, even without the light attachment
    - Too many tubes, reminiscent of medical devices

Connection

- USB dongle

Cleaning

The documentation for the Quadstick provides the following cleaning recommendations:

"The mouthpiece can be removed and cleaned with soap and water, rubbing alcohol or hydrogen peroxide. Any debris blocking the air holes can be pushed out from the barbed fitting end with a small wire or compressed air. Soaking can help.

In multi-user settings, individual mouthpieces are recommended.

Once clean, any calcium deposits, which can show up as white specks, can be removed by soaking in a mild acid, like vinegar or lemon juice.

The PLA material currently being used is compatible with ETO (Ethylene Oxide) and can be autoclaved after cleaning."

Alternate Versions

- FPS
- Singleton
- Original

Link

https://www.quadstick.com/

3D Printable Mouthpieces: https://github.com/QuadStick

http://quadstick.s3.amazonaws.com/documents/user_manual/um/dropdown_list_for_ouput_functions.htm

IntegraMouse

The IntegraMouse is a wireless oral computer mouse

Figure 12. IntergraMouse.

Price

$2160 EUR / ~ $3300 CAD

User Interface

- Joystick: Unknown
- Pressure: Unknown, possibly mechanical

User Feedback

- Battery power indicator
- Joystick mode display
- Cursor mode display

Mouthpiece

- Mouthpiece
  - Description: IntegraMouse Plus mouthpiece
  - Material: SABIC Lexan HP3NREU
  - Application part: Type B
  - Measurement: 65 x 25 x 28 mm
  - Weight: 7 g
  - Silicone tube for making mouthpiece more comfortable
  - Shouldn’t be used by people at risk of swallowing / choking
- Membrane
  - Description: IntegraMouse Plus Membrane
  - Material: Hard part: SABIC PP PCG H10; Soft part; TPE - TM3MED
  - Measurements: Diameter: 17mm, height: 6 mm

    - Weight: <1 g
- Silicone Tube
    - Description: VERSILIC S60V34B
    - Material: Silicone
    - Hardness: Shore A 62±5
    - Diameter: inner/outer 6/10 mm
    - Length: 19 ± 1.5 mm
    - Authorisations: European Pharmacopoeia §3.1.9, FDA 21CFR §177.2600

Switch Inputs

- One 3.5 mm jack for changing operating mode

Mounting

- 3X ¼”-20 UNC holes on bottom portion – left / right / bottom

Connection

- Wireless, 12 hours battery life at continuous operation
- USB power connection
- Joystick mode
- Cursor mode
- Keyboard mode
- USB dongle for connection

Cleaning

- The mouthpiece is disinfected in boiling water for 3.5 minutes. The ‘membrane’ is removable and disposed.
- Mouthpiece should be cleaned at least once per week; more in cases of contamination or illness
- Mouthpiece should be disposed after 6 months

Alternate Versions

- IntegraMouse Plus
- InegraMouse Plus Connect
- Powerpack

Link

https://www.integramouse.com/

Celtic Magic Feather

Figure 13. Celtic Magic Feather.

The Feather by Celtic Magic is a lightweight input device that can be used as a joystick or mouse. The activation force can be adjusted between 5 – 20 grams force. It is available in a version that has a sip and puff sensor.

Price

~$600 CAD + International Shipping

User Interface

- Adjustable force
  - 5, 10, 15, 20 grams
  - +/- 6 degrees
- No springs; 18 magnets arranged internally
- Adjustable settings through configuration app

Figure 14. Screenshot of Feather Configuration App.

User Feedback

- Optional external display

Mouthpiece

- 20mm cork sphere

Switch Inputs

- 2 3.5 mm switch inputs

Mounting

- ¼-20 UNC Camera mount adaptor

Enclosure

- Mostly 3D printed

Connection

- USB Connection
- 3 modes
  - Mouse
  - Simple

Cleaning

- Flush tubes with warm soapy water

Alternate Versions

- Alternate taller version

Link
https://www.celticmagic.org/feather

https://www.youtube.com/watch?v=Luvadk9oKYs

DIY Sip and Puff Joysticks

There are a variety of DIY projects.

FlipMouse

Figure 15. Lasercut FlipMouse Design.

https://www.asterics-foundation.org/projects/the-flipmouse/

- 149 euro = ~216 CAD

- same mouthpiece and filter as QuadLife

Mouth Operated Mouse

Thingiverse: https://www.thingiverse.com/thing:1090461

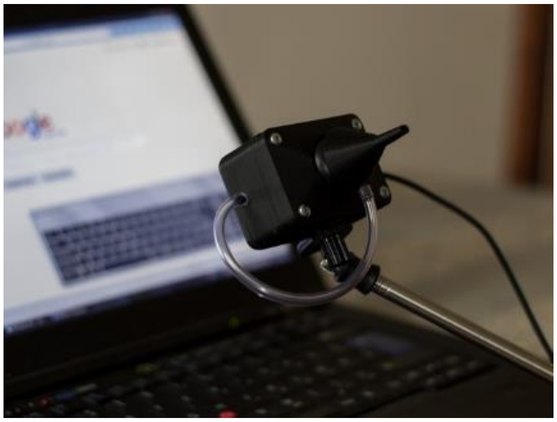


Figure 16. Mouth Operated Mouse.

- < $20 USD
- 3D printed mouthpiece
- 15 cm length 6 mm diameter food-safe PVC tubing (~0.40 USD hardware store)

Enclosure

- Pros
  - Compact
  - Cost effective
- Cons
  - Dull cube
  - Device in line of sight
  - Tube looks like medical device

Related Devices

NetCle

https://tetragear.com/products/netcle/

Developed by a group of Tetra volunteers

- I2C over a TRRS cable
- Joystick and switch interface device

Research Summary

The devices researched included the Jouse, QuadLife (Formerly QuadJoy), Quadstick, IntegraMouse, and Celtic Magic Feather for commercial devices, and the FlipMouse, Mouth Operated Mouse, and NetCle for DIY devices.

The research shows that all sip and puff devices should use a filter, primarily to protect the interior pressure sensor from moisture, but also replaceable and cleanable/sterilisable mouthpieces for sanitary health. Most devices use a hydrophobic filter, similar to what the LipSync has used in previous iterations.

Some devices also use a 3D printed mouthpiece, however many seem to use a commercial medical straw of some sort. As we cannot guarantee the food safety of 3D printed parts at this time, especially in DIY situations, we will not be considering 3D printed mouthpieces.

As far as joystick mechanism goes, the research suggests moving towards Hall effect sensors for more accurate readings that will not deteriorate and produce drift like potentiometers. Looking at the Celtic Magic Feather specifically, it is seen that we can create a very low force joystick using magnets.

Devices such as the FlipMouse have an online configuration tool that can be used to create custom slots and adjust settings. This requires a host device with an internet connection, but is a user friendly way to configure the device.

One gap in the existing solutions is a way to change settings without needing access to some application or browser on the host device.

Conceptual and Detailed Design

There are several key decisions that impacted the design of this version of the LipSync. First, the selection of a more reliable joystick that would be easier to use and make. Second, feedback from users signifying a desire to change settings easier.

Architecture

Previous iterations of the LipSync have been unibody designs that have the joystick, microcontroller, PCB and components all within the singular body. A modular architecture would separate the joystick from the microcontroller and potentially some feedback features.

Unibody

A unibody design with the joystick mechanism, user feedback, and microcontroller all in the same enclosure.

Pros

- Fewer parts to mount

- Fewer cables to manage

Cons

- Larger size
- Limits the available size of interior components
- Requires feedback lights to always be in the user's face

Modular

Creating multiple enclosures to separate the joystick mechanism from the feedback and microcontroller.

Pros

- May be easier to mount
- Opens up options for the joystick mechanism as there is more space to work with because the electronics are removed from the main body
- More flexibility in the set-up

Cons

- Need to mount additional devices
- More cables to manage

Architecture Decision

While both unibody and modular approaches were explored, the modular architecture was selected to reduce the size of the portion near the user's face and provide a wider range of options for visual feedback.

The Alpha iterations are the Unibody concepts and the Beta iterations are the modular concepts. The Beta designs built off lessons learned from the Alpha prototypes.

LipSync Major Systems

The following is a brief overview of the main systems in a LipSync.

Pressure Measurement System

The pressure measurement system measures the difference in the ambient pressure and the pressure from the mouthpiece to determine if the user has done a sip or a puff.

Movement Measurement System

The movement measurement system is how the LipSync captures movement from the mouthpiece and can convert it into a format the microcontroller can process.

User Interface

The user interface is how the user can interact with the LipSync beyond just the mouthpiece, such as onboard buttons, or assistive switches.

User Feedback System

The user feedback system is how the LipSync gives the user feedback on their actions. This feedback can be in the form of a text display, LED lights, or audio feedback.

Mouthpiece System

The mouthpiece is the how the user primarily interacts with the LipSync, it allows them to move the joystick to control the movement of the mouse, as well as sipping and puffing to click and perform other inputs.

LipSync Joystick Design

The joystick is the portion of the LipSync that contains the sensors that capture the user's mouth input. It consists of the gimbal mechanism which captures the movement of the mouthpiece and the sip/puff inputs, the mounting hardware, and the hub connection. The gimbal mechanism is being redesigned to move away from the force sensitive resistors and towards a lower force, more sensitive gimbal mechanism.

Requirements for Standalone Unit

Goals

1. Try to avoid a custom PCB of any kind within the joystick unit.
2. Minimize or avoid the use of screws and other mechanical fasteners
3. Keep mouthpiece in line with the axes of rotation of the gimbal
4. Extend physical range of motion of the joystick (dependent on linearity of response beyond Alpha II range)

Requirements

1. Secondary user must be able to change the filter.
2. Must be able to change filter in comparable time to the previous LipSync iterations.
3. Comparable or improved resolution to the LipSync 3.0 joystick
4. Comparable or improved linearity of response (both force and output response profiles) to LipSync 3.0
5. Comparable or improved centering/neutral hysteresis to LipSync 3.0
6. Lower max force than LipSync 3.0
7. Comparable size to LipSync 3.0
8. Able to change the orientation of the mounting arm while keeping the joystick itself in the same approximate orientation

    a. The internal components should not interfere with one another when in use or when changing mounting angle.
9. Must limit joystick motion to a reasonable range for a user.
    a. Baseline is LipSync 3.0 at ±3 mm at mouthpiece tip.
10. The external cable connecting to the Hub must be able to route out along the mounting arm in any orientation.
11. Any features of the device must have as minimal impact within the users field of vision as possible
    a. Ideally no features above the joystick
    b. No bright lights shining in the user's eyes

Bonus Scope

1. The ability to change the range of motion of the joystick
2. The ability to change the required joystick operating force

Components being designed around

1. TLV493D sensor and magnet for joystick movement
2. LPS33HW Sip and Puff pressure sensor
3. LPS22HB Ambient pressure sensor
4. RJ11 Jack to connect to main unit with a cable
5. Centering mechanism
    a. Previously fixed magnets – could configure differently
        i. Centering magnets behind the sensors
    b. Flexure based?
    c. Variable?
6. Bearings for smooth movement
7. Mouthpiece

Primary Selection Criteria

- Makeability
- Sourcing of parts
- Printability
- Overall size
- Secureness of airpath attachments
- Functionality
    - Centering force
    - Magnetic sensor response
- Design effort/time

Conceptual Design and Rapid Prototyping

The following are the concepts that were explored for the modular joystick unit design. Note that this was following the Alpha designs, which are unibody and a separate design phase with different concepts than the ones explored here.

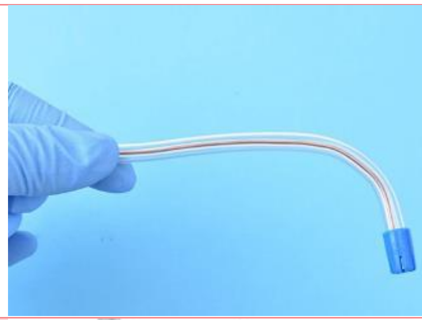
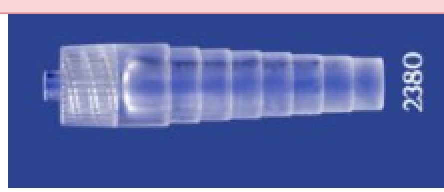
Sip and Puff Airpath System

Air Path:

- Two main options
  - Direct to pressure board
    - Most secure connection as it sandwiches the tubing and board together
    - Requires inline mouthpiece and filter
    - Requires Luer components
      - Likely possible without, but this is the easiest and most secure connection we currently have
    - Requires pressure board on the gimbal
  - Route tubing through gimbal to pressure board elsewhere on the device
    - Potential to remove Luer components
    - Requires some secure way to securely connect tubing to mouthpiece, filter, and pressure sensor
    - Requires some way to securely connect the mouthpiece and filter while having them easily removable.
    - Pressure board location is relatively arbitrary
    - Requires thought into tubing routing and keeping it secure in place
      - Bend radii
      - Snag risk
      - Internal features to fasten to

Mouthpiece Options

Item	Image	Summary	Price	Link
Spray Nozzle Tips		Bent plastic straw with groove for location. Separate internal straw inside the hard plastic outer.	$20.49/250 $0.08 Each	Amazon

Dental Straw		Flexible dental straw with plastic tip.	$22.11/100 $0.22 Each	[AliExpress]
Dental Straw		Flexible dental straw with plastic tip.	$23.99/100 $0.24 Each	[Amazon]
Stepped Male Luer Lock Connector		A hard plastic connector with a male Luer on one end and a stepped plastic connector on the other. Currently used in the LipSync Alpha II	Unknown	[Quosina]
Fir Tree to Male Luer Lock Connector		A hard plastic connector with a male Luer on one end and a stepped plastic connector on the other. Currently used in the LipSync Alpha II	Unknown	[Rocket Medical]
Christmas Tree Connector Male Luer		A hard plastic connector with a male Luer on one end and a stepped plastic connector on the other. Currently used in	49.99 GBP/10 4.99 GBP Each Current 83.12 CAD	[DMS Veterinary]

		the LipSync Alpha II	Oct 10 2023	

Filter Options

Item	Image	Summary	Price	Link
PTFE Syringe Filter		30mm Diameter 0.22 um pore size Female Luer one side Luer Slip one side	\$39.99/100 \$0.40 Each	Amazon
Barbed Air Line Filter		5/16 barb connectors both sides Site says it filters down to 0.5, no units given	\$19.71/4 \$4.93 Each	Amazon
Venting Filter Disc		50mm Diameter 0.2 um pore size ¼-1/2 in barbs both sides	\$117.18/10 \$11.71 Each	Amazon
Hydrophobic Luer Filter		17mm Diameter 0.2mm Male Luer one side Female Luer one side Currently in use on prototypes	Unknown	Quosina

Hydrophobic Dual Luer Lock Filter		25mm Diameter 0.2 um pore size Female Double Luer Lock one side Male Luer Slip outlet	$8.81 (Possibly for 5)	[Foxx Life Sciences]
Inline Filter		6-10mm ID barbs Pore size from 0.1-10um	$258.87/10 $25.89 Each	[McMaster]
PTFE Syringe FIlter		0.22 um pore 25mm Double Luer lock	$132.18/100 $1.32 Each	[Tisch]
Nylon Syringe Filter		0.22 um pore 25mm Double Luer lock Reseller of the above Tisch filter. Has poor reviews and $30> shipping and import fees	$104.43/100	[Amazon]
Double Luer Lock PTFE Syringe filter		0.22-10 um pore 25mm Double Luer lock	$142.99/100	[Aurora Scientific]

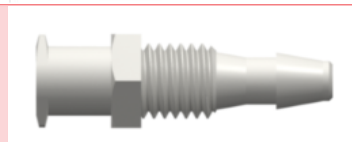
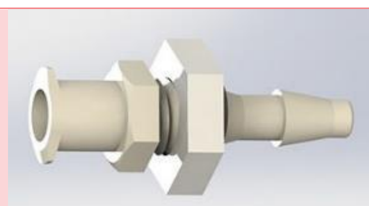
Tubing Options

Item	Image	Summary	Price	Link

Silicone Food Grade Tube		2mm ID 4mm OD 3 meter long silicone tube	$12.49/3m	Amazon
Soft PVC Plastic Tubing		1/8” ID ¼” OD plastic tubing	$12.00/50ft	McMaster

Connector Options

Item	Image	Summary	Price	Link
Female 1/8” Luer Lock Barb Connector		A connector with a female Luer on one end, a 1/8” barb on one end, and has a knurled knob in the middle.	$11.99/30 $0.40 Each	Amazon
Female 1/4” Luer Lock Barb Connector		A connector with a female Luer on one end, a 1/4” barb on one end, and has a knurled knob in the middle.	$10.99/30 $0.37 Each	Amazon
Male 1/4” Luer Lock Barb Connector		A connector with a male Luer on one end, a 1/4” barb on one end.	$11.99/30 $0.40 Each	Amazon

Male 1/8" Luer Lock Barb		A connector with a male Luer on one end, a 1/8" barb on one end.	Unknown	Cole Parmer
Female Luer Lug Panel Mount 1/8th" barb		Panel mount connector with female luer lock on one end, 1/8" barb, and ¼-28 UNF threads. Most similar to one currently used in Alpha II prototype	$0.2409 Each 1000 minimum order Total $240.09	Nordson Medical
Plastic Quick Turn Tube Coupling			$4.47/10 $0.48 Each	McMaster
1/8" hose barb Female Luer Lug Panel Mount		Panel mount connector with female luer lock on one end, 1/8" barb, and ¼-28 UNF threads. Most similar to one currently used in Alpha II prototype	$1.32 Each Currently out of stock	Industrial Spec

Commercial Options

Item	Image	Summary	Price	Link
QuadLIFE Stick Kit		1X Stick (Straw) 1X Hydrophobic Filter 1X Line & Luer	$20 USD + Shipping ($10 USD to US / $35 USD to Canada)	QuadLIFE

QuadLIFE Supplies Pack		2X Sticks 2X Hydrophobic Filters 4X Saliva Filters 2X Line & Luer	$40.50 USD + Shipping ($10 USD to US / $35 USD to Canada)	QuadLIFE
QuadLIFE Stick (Straw)		2X Stick (Straw) Polycarbonate	$24 USD + Shipping ($10 USD to US / $35 USD to Canada)	QuadLIFE
QuadLIFE Saliva Filters		12X Saliva Filter Blocks saliva from entering sip and puff tubing	$11.50 USD + Shipping ($10 USD to US / $35 USD to Canada)	QuadLIFE
QuadLIFE 2 Line & Luer		2X Male Luer w/ tubing attached	$4.50 USD + Shipping ($10 USD to US / $35 USD to Canada)	QuadLIFE

QuadLIFE Hydrophobic Filters		5X Hydrophobic filters that keep saliva out of the magnet holder. Male Luer and female luer connector.	$17.50 USD + Shipping ($10 USD to US / $35 USD to Canada)	[QuadLIFE]()

Mouthpiece Testing

As a test, the dental straw, syringe filter, Luer connector, and plastic tubing were ordered. The test fit of the dental straw on syringe filter connection was found to be very weak, slipping on and off with no resistance. A zip tie was tested [1] and it was found that the walls of the dental straw were too firm to be crushed against the filter to secure it. A length of the silicone tube from the original LipSync was cut and placed over the syringe tip before the dental straw was placed over [2], and it was found to be a very secure connection, requiring 2600 grams of force to dislodge. This was repeated again, but with a small amount of hot glue around the outside of the tube on the syringe filter [3], and with hot glue 'caulked' around the base of the tube and the filter. This connection was found to be so secure that it could not be undone with any amount of force that could be reasonably applied to a LipSync, and more force than could be measured with the test apparatus (>3kg). Another mouthpiece was tested with just the filter and the dental straw held in place with hot glue [4]. This connection also maxed out the scale used to measure the force at over 3kg of force.

When the silicone sealant was trialed, two mouthpieces were constructed using the sealant, one with the inner piece of tubing as a sock and silicone on the outside, and the other held together with just the silicone sealant. After letting both mouthpieces dry for 5 days, they were tested on the force testing jig from the initial mouthpiece selection. The mouthpiece that was held together with just the silicone failed at 1.4kg, and the mouthpiece with the sock and the silicone withstood more than the 3kg that the testing jig could measure. Both mouthpieces did not have any air leaks before testing, and the sock and silicone mouthpiece did not have any leaks after testing.

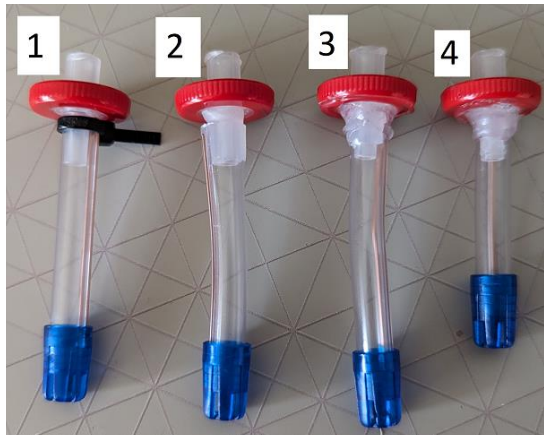


Figure 17. Mouthpiece Options

A male Luer to 1/8th inch barbed connector was used to connect the mouthpiece assembly to the joystick assembly. The male Luer lock has a series of ridges along the outside that can be used to prevent rotation of the mouthpiece when the clamshell of the joystick housing is clamped around it, although the specific pattern is different between suppliers.

Since the grip pattern on the ordered Luers was a series of 10 indents along a cylindrical surface, a decagon was used on both sides of the clamshell Luer to prevent rotation. After the clamshells were secured with bearings, there was a secure enough connection to allow for the Luer lock connection to be made.

Sip and Puff Sensor Concepts

For the sip and puff functions, there are two methods initially considered to measure and trigger these inputs. The first way, used on the previous iterations of LipSyncs, is a closed-loop control, or differential pressure readings. This means that the sensor(s) take two readings of pressure, one ambient for reference and one directly from the mouthpiece and compares them to measure if a meaningful sip or puff has been performed. The second way is through an open-loop control system, where there is a single sensor reading the mouthpiece pressure and the criteria for a meaningful sip of puff is based on the pressure change over time. This means that the reference pressure changes, making this a very dynamic system. A sudden change in pressure would trigger the input, while gradual changes through the day would not.

To reduce the risk of error, the differential pressure approach was selected.

Soldered Differential Pressure Sensor

The LipSync 3 used a soldered differential pressure sensor on its PCB. A tube was attached to one of the ports and the other was left open to read the ambient temperature.

This was a surface pad soldered part, and approximately $30 CAD.

https://www.digikey.ca/en/products/detail/nxp-usa-inc/MPXV2202DPT1/3743114

LPS33HW + Onboard Pressure Sensor

(i.e., Adafruit Feather nrf52840 Sense)

Used in the Alpha II design, the LPS33HW sensor has a barbed outlet that a tube can be pushed on to and the Adafruit Feather nrf52849 Sense microcontroller has an onboard pressure sensor that can read the ambient pressure. This concept is dependent on the microcontroller having an on board pressure sensor.

https://www.digikey.ca/en/products/detail/adafruit-industries-llc/4414/11201428

Dual LPS33HW Concept (Dual STEMMA Breakout Board Pressure Sensors)

- Need to solder pads on second LPS33HW to change i2C addresss
- I2C Address 0x5D (0x5C w/ Jumper)

Using two STEMMA breakout boards would improve the ease of assembly as they can be daisy chained together. The LPS33HW breakout is not the ideal selection for both boards as it is approximately $18 CAD individually.

https://www.adafruit.com/product/4633 - LPS22HB - $6.95 USD

The LPS22 is a similar pressure sensor breakout board that is much more cost efficient. It does not have a barbed port, but this is irrelevant as it is only reading the ambient temperature.

LPS33HW and Secondary Pressure Sensor

This approach requires the ambient pressure sensor to be a soldered component on the board. This would take up space on the PCB, but opens up the options to a larger number of pressure sensors.

Majority of these sensors are still expensive and are surface pad soldered parts, making assembly difficult.

Sensor Comparisons

Device	LPS33HW	LPS35HW	LPS22	MS8607
Cost (USD)	12.50	12.50	6.95	14.95
Availability	8 (Digikey 2023-10-11)	9 (Digikey 2023-10-11)	29 (Digikey 2023-10-11)	19 (Digikey 2023-10-11)
Type	Absolute	Absolute	Absolute	Absolute
Waterproof	Resistant	Resistant	No	No
Voltage	5V or 3.3V	5V or 3.3V	5V or 3.3V	5V or 3.3V
Range	260-1260 hPa	260-1260 hPa	260-1260 hPa	10-2000 hPa
Accuracy	+/- 0.1% hPa	+/- 0.1% hPa	0.1 hPa	+/- 2 hPa (0.016)
Resolution	24 bit	24 bit	24 bit	16 bit

Device	LPS33HW	LPS35HW	LPS22	MS8607
Notes	Identical to 35HW, but with nozzle	Identical to 33HW, but with no nozzle		Adafruit changed units partway through the page, converted for easy comparison

Gimbal Concepts

Concept 1 – Inline Joystick – Reach around centering – No boards on gimbal

Similar setup to **Concentric.f3,** the modified CAD model below and a previous design concept from the Alpha design phase.

- Similar central bearing housing to Alpha II
- Outer bearing housing integrated into enclosure
- Alternate air pathway (no Luer lock pieces)
  - Barbed filter
  - Tubing
  - Dental straw
- Centering magnets positioned to the rear
  - Longer moment arm
  - Potential for adjustable centering force
- Pressure sensor locations arbitrary
- Magnet sensor in line with joystick

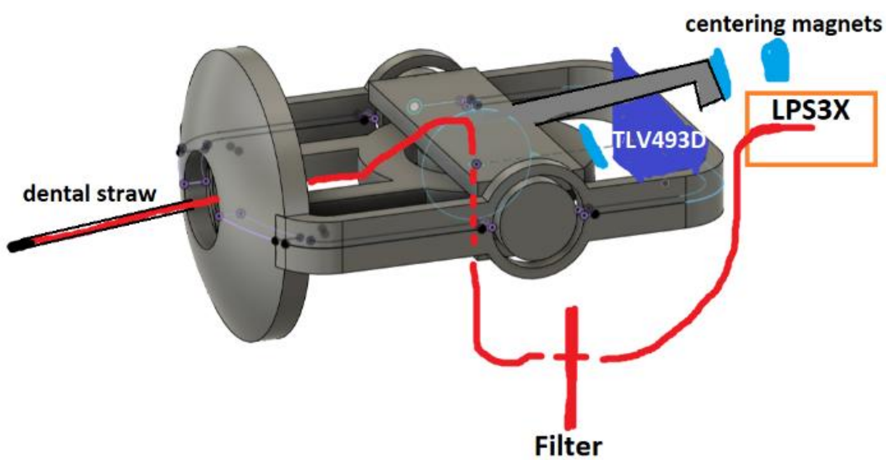


Figure 18. Inline Gimbal Concept.

Need to have central bearing mount held together around bearings like the Alpha II design. This allows the central bearing to be held in place without fasteners or print in place required.

Nothing to securely hold the mouthpiece part to that central bearing otherwise.

Proposed advantages

- Builds off components within the Alpha II design
- Reduces overall required fasteners
- Low weight (No breakout boards) on the joystick mechanism itself
  - Less momentum causing oscillation or drooping based on orientation
- Very modular in terms of where components have to be situated within the enclosure
  - Pressure sensor placements can be anywhere
- Keeps magnets away from the front of the device
- Magnet and sensor are located through the same print (bottom enclosure)
- Access to centering mechanism from the back could allow for adjustments
- The width of the clip and swing arm means no overhangs.

Proposed disadvantages

- Different printer tolerances may result in bearing fits to be too tight or too loose
- Would turn into a long and narrow device, making it very tall if it were modified into a regular joystick, like the Oak joystick
- The width of the swing arm is not wide enough to fit the original mouthpiece parts.
  - Could be modified to the Alpha II swing arm width

Test Print

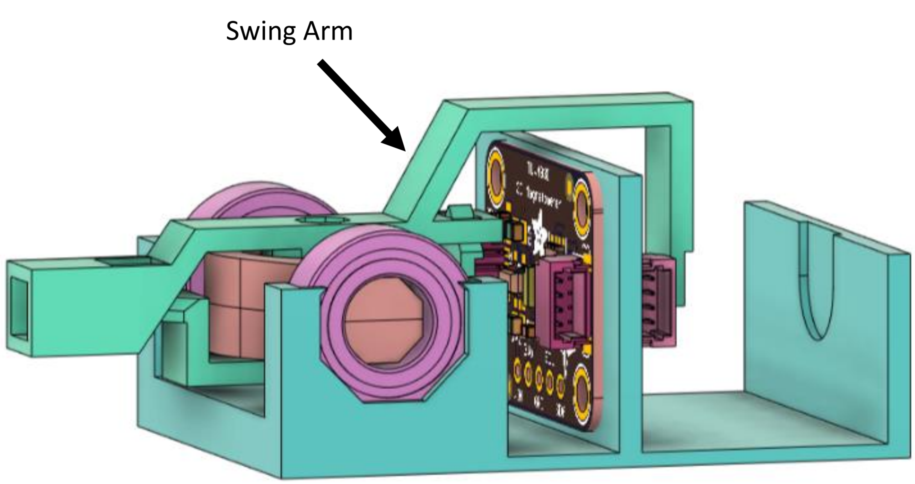


Figure 19. Swing Arm Concept.

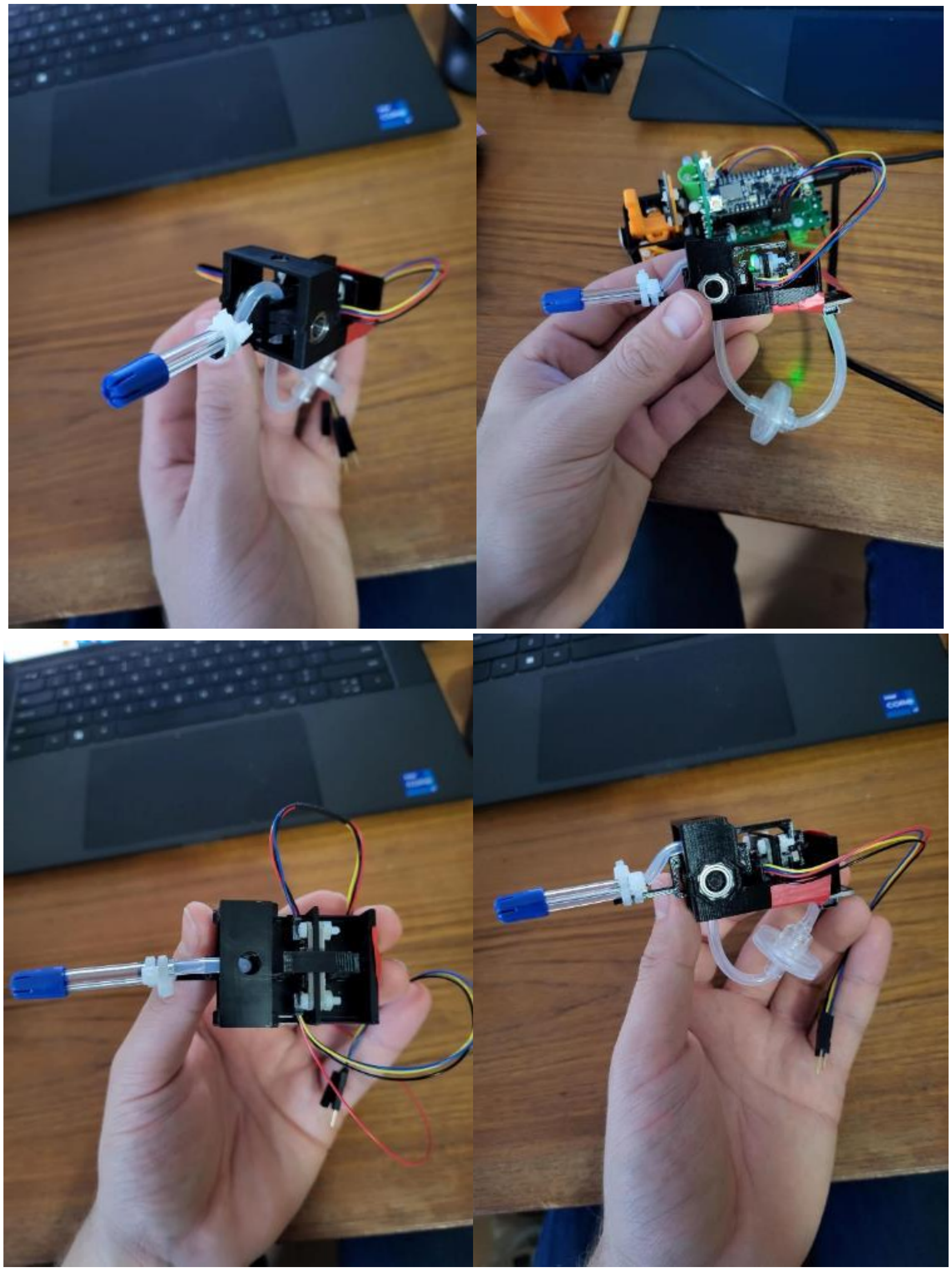
Approximate print time of existing rough model: 1 hour

OFIs

- Requires top housing piece to hold the bearings in place (integrated into top enclosure)
- Requires second pressure sensor (the air path one) and RJ11 connector
- Need some way to securely connect air pathway tubing

- Need to have some way of securing the tubing so it doesn't pull on the joystick mechanism or pull off the ports it's attached to.
    - Requires air path routing through bottom enclosure
- Requires a way to secure the mouthpiece in place
    - Cable tie on a flat portion?
- Requires a better way to secure magnets in place (can make parts thicker)
- More secure clip for the swing arm
- Integrate bottom housing into enclosure design (print orientation will matter)
- Line up magnetic sensor with the center line properly
- Explore if the overall width can be reduced (bearing to bearing for the central bearing mount)
- Find a way to remove fasteners (M2.5 required for the boards)

Concept 1 Prototyping



Plug and play with the Alpha II board.

The joystick is very solid and doesn't pick up vibratory movements but remains low force to move.

Drift only occurred if the tubing leaving the bottom got pulled forward or backward. This could be mitigated easily with a smaller hole to route the tubing through the enclosure and/or by adding some tube management features to hold the tubing and/or filter in place below the unit.

Pros

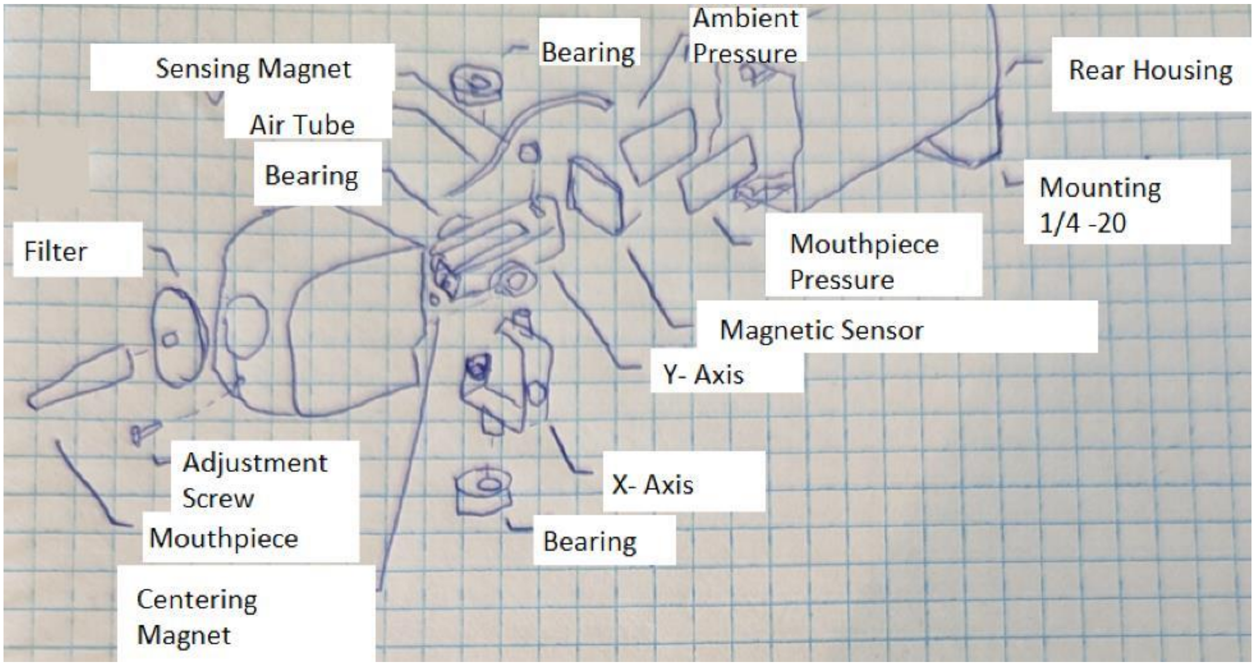
- Improvement on existing Alpha II in terms of centering force and drift-free movement
- Works as is with current Alpha II code and hardware
- Mouthpiece parts easy to source
    - No Luer parts
- Fewer fasteners than alpha II
    - Two screws and nuts for TLV493D and secondary LPSXX board.
    - Likely required screws and nuts for RJ11 connector and possibly LPS33HW board

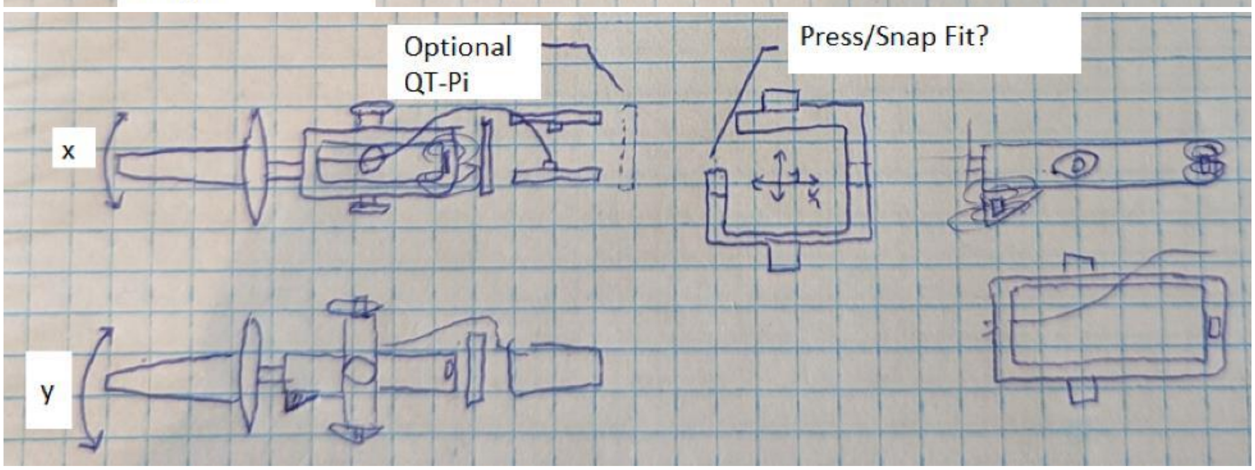
Cons

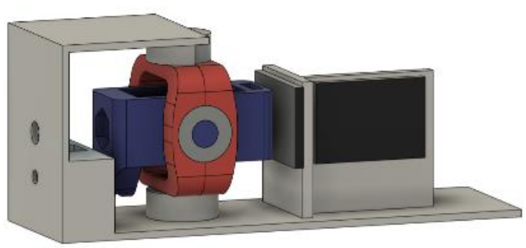
- Mouthpiece difficult to change (cable tied on)
- Dangling filter could get caught and pulled off
- Pushing the tubing through the 3D prints is very doable, but can be difficult to start
- PLA creep could cause centering to drift over time due to the long, thin swing arm
- full extent, some minor oscillation can occur
    - Results in the cursor bouncing back a touch.

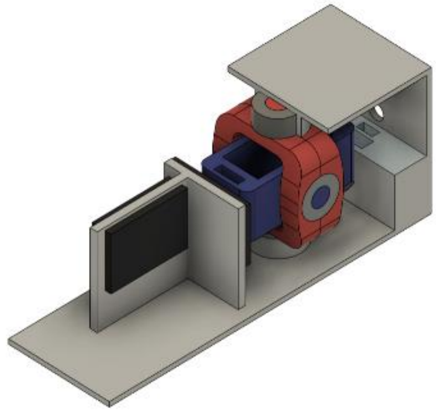
Concept 2: In-line Mouthpiece, Offset-Centering Magnet at Front, Pressure Tube Through Gimbal

This concept brings the airpath into the center of rotation of the gimbal to solve the offset axis problem with the Alpha II. All the sensor boards are off the gimbal, and the magnet is shifted to be off axis below the air path.


Sensing Magnet
Air Tube
Bearing
Bearing
Ambient
Pressure
Rear Housing
Mounting
1/4 -20
Filter
Mouthpiece
Pressure
Magnetic Sensor
Y- Axis
Adjustment
Screw
Mouthpiece
X- Axis
Bearing
Centering
Magnet


Optional
QT-Pi
Press/Snap Fit?
x
y





Features

- Prioritize minimizing diameter over minimizing length
- Aligns mouthpiece with actual axis of rotation
- Move pressure sensors off swing assembly
- Use screw/captive nut on the front to adjust magnet
    - Also potentially replace the dime?
- Mounting could be done with a fixed ¼-20 t-nut, or a rotating ring like current prototype
- Housing used as outer bearing mount, inner bearings captured with a snap fit

Proposed Advantages

- Reduces the profile of the LipSync so that it takes up less of the field of vision of the user
- Allows the user to adjust the distance of the centering magnet to adjust centering force
- Less mass in the joystick assembly itself

Proposed Disadvantages

- The tubing going to the pressure sensor may skew the centering of the joystick
- Narrow but long design makes it difficult to use as a joystick
- Front and back halves of enclosure hold the bearings, force on joystick may work the two halves of the shell apart if not properly secured
- Putting the magnet assembly in front of the joystick will create a larger hole in the front of the enclosure because the angle over the longer distance

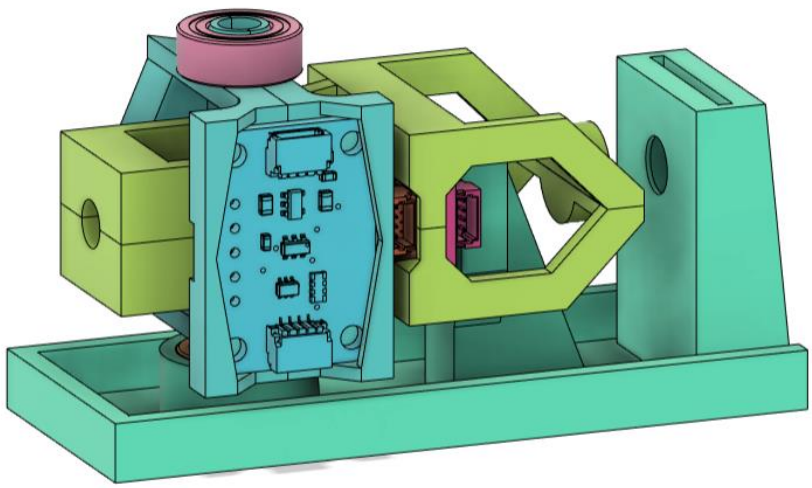
OFIs

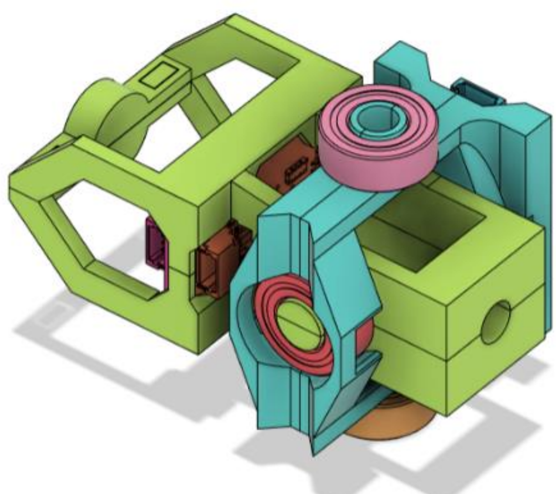
- The location of the bearings on the Y-swing arm could be tuned so that it "counterweights" the mouthpiece, reducing the necessary magnet strength
- How will the pressure tubing components (mouthpiece, filter, tubing) be connected to the joystick assembly without luer lock?
- Not captured in drawing, but filter could be moved between the magnetic sensor and the dual pressure sensors to bring it into the enclosure
    - This would make it more difficult for users to change the filter though

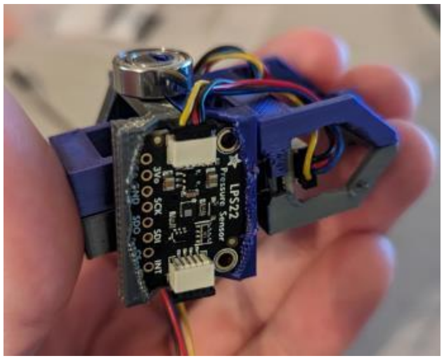
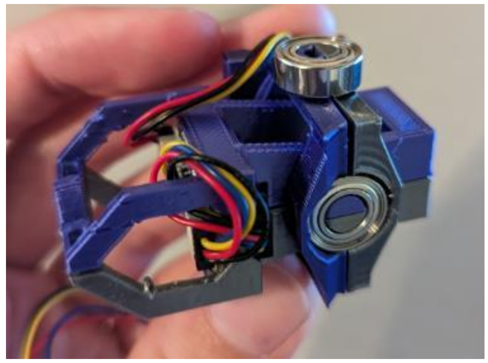
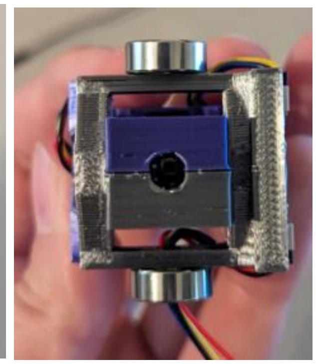
- Built in cable managing to prevent kinks in air line

Concept 3: Direct connection to pressure sensor - All boards on gimbal – Reach around centering

Built off Concept 2, this design adds all three STEMMA QT boards onto the gimbal so that the pressure sensor to mouthpiece connection is direct, like the alpha II design, and the only routing away from the gimbal is the STEMMA – DuPont cable going to the RJ 11 connector.





After printing, the joystick was found to be plug and play with the current LipSync Alpha II hardware. Air path components were not available to test with, but the concept uses dental straw, PTFE Syringe Filter, Female 1/8" Luer Lock Barb Connector, and Silicone Food Grade Tubing from the air path components list. This has a maker cost of $88.46 and a per unit cost of $1.14.

Centering force was found to be stronger than concept 2, and the mechanism felt stiff. Joystick was also very sensitive when connected to the Alpha II PCB. The sensing magnet could be moved further back to test how that affects the sensitivity.

Filter will be mounted the same as the Alpha II prototype so that the filter is easily accessible and replaceable by a secondary user.

Features

- Prioritize minimizing diameter over minimizing length
- Aligns mouthpiece with actual axis of rotation
- Moves all sensors onto the gimbal
- Wires leave the gimbal through the axis of rotation to add minimal rotational bias to the system
- Mounting could be done with a fixed ¼-20 t-nut, or a rotating ring like current prototype
- Housing used as outer bearing mount, inner bearings captured with a snap fit

Proposed Advantages

- Reduces the profile of the LipSync so that it takes up less of the field of vision of the user
- Sensors in the back of the gimbal should counteract the weight of the mouthpiece
- Brings the mouthpiece into the axis of rotation of the gimbal.
- Dual clamshell design allows all the boards and the magnet to be securely held in place without the need for fasteners.
- Back magnet bracket design is as stiff and sturdy as the rest of the joystick
- Wires going through the bearing prevents any rotational bias/drift

Proposed Disadvantages

- Front and back halves of enclosure hold the bearings, force on joystick may work the two halves of the shell apart if not properly secured

- All the sensors being on the gimbal adds weight
- Back mounted magnet adds significant length to the assembly

OFIs

- Needs another pass for printability, some fit issues with the bearings and the boards
- Need to figure out a better way to mount the magnets to the housing, current sensing magnet placement causes interference

Magnet retainment

The sled that the gimbal sits in contains two magnets. One is used to provide a centering force to return the gimbal to the neutral position, and one is used to create a magnetic field for the sensor on the gimbal to read. These magnets need to be securely fixed in place in the sled to keep the joystick accurate. A few different methods of magnet retainment were trialed.

Slot and pin

The slot and pin method extends the magnet pillars up further and adds two slots. A pin can be inserted through these slots to hold the magnet in place and prevent it from shifting as the joystick moves.

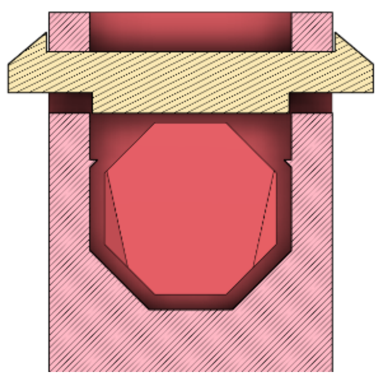


Figure 20. Slot and Pin Concept.

Pros:

- Easier to get around printer tolerances by allowing the pin to flex around the magnet
- Magnet is more removeable

Cons:

- Taller magnet mount
    - makes installing the gimbal into the sled more difficult
    - could interfere with the gimbal at its extent
    - more material

- adds an extra part that is very small

Slot with bumps

This method of securing the magnet adds two small bumps into the slot that the magnet sits in. The magnet is pushed past these slots, and they prevent the magnet from slipping back out or shifting around in the slot.

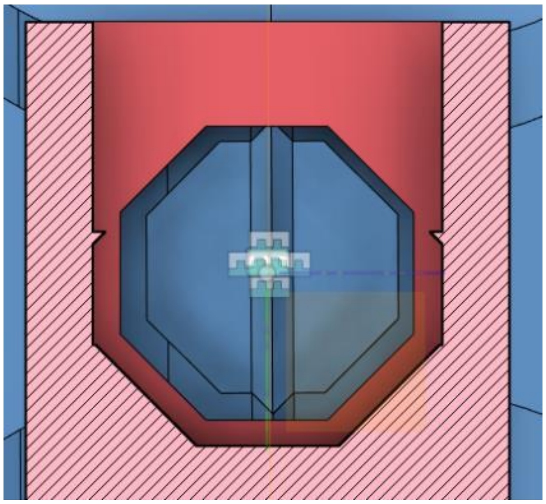


Figure 21. Deformable Bump Concept.

Pros:

- Shorter magnet mount
    - Less material
    - Less likely to interfere with other components
- Fewer parts overall

Cons:

- Larger risk of slop in fixed position of magnet due to printer tolerance variations
- The magnet is less removeable

Glue

This method of securing the magnet uses cyanoacrylate (CA) glue (Superglue/ Crazy glue) to secure the magnet in the slot with no restraints in the 3D model itself

Pros:

- Shorter magnet mount
    - Less material

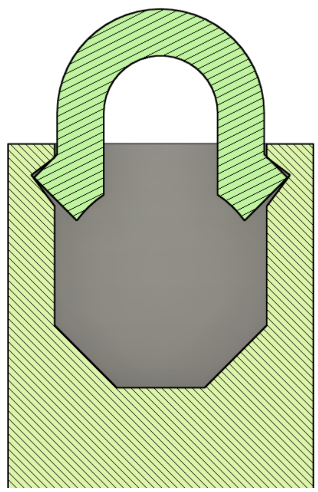
    - Less likely to interfere with other components
- Fewer parts overall

Cons:

- Sled needs to be destroyed to remove the magnet
- The magnet is likely to be destroyed/be covered in glue when removed from the sled
- Requires glue/adds to the tool list

Horseshoe Clip

The horseshoe clip method of magnet retainment adds a pair of indents into the sides of the magnet slot, then uses a horseshoe shaped clip with barbs on the end to secure it in place. The horseshoe shape allows the clip to flex and be inserted into the slot. The clip was made with the lower face that will push on the magnet to be slightly oversized so that once the barbs on the clip are in their indents, the clip will always be pushing down on the magnet to hold it in place.



Pros:

- Easier to get around printer tolerances by allowing the clip to flex into the slot
- Very easy to assemble
- Magnet is more removeable

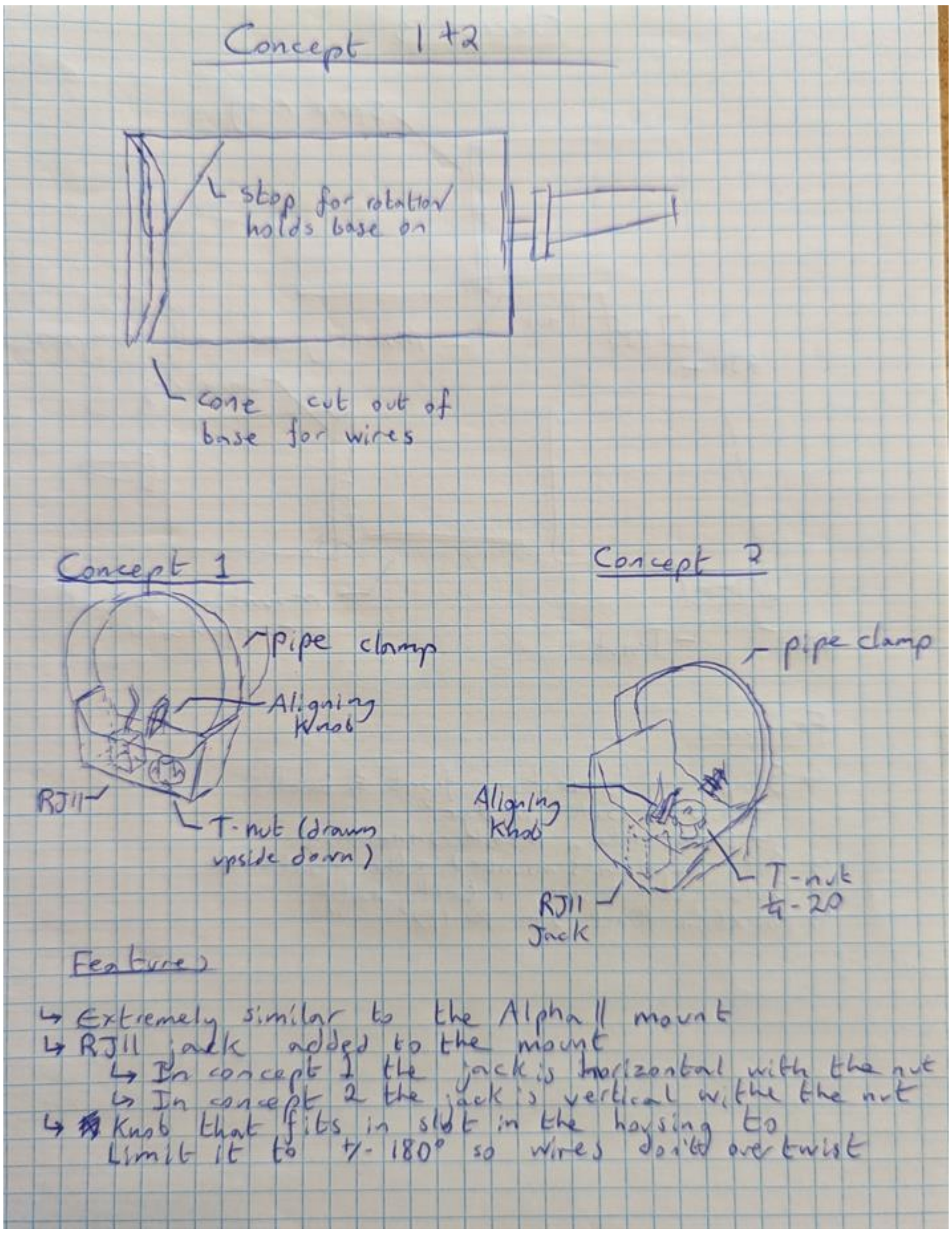
Cons:

- Taller magnet mount
    - more material
- adds an extra part that is very small

Joystick Mounting Concepts

Concept 1 & 2 – Hose Clamp

Both concepts 1 and 2 are based on the Alpha II hose clamp mounting system, with the RJ11 jack added into the hose clamp along with the preexisting T-Nut. The primary difference between the two concepts is whether the RJ11 jack and T-Nut are side by side or in line with each other. The i2c wires would feed out the back of the enclosure and into the hose clamp, with a rotation stop to limit it to +/- 180 degrees of rotation.



Features

- Concept 1 and 2 are a modified version of the current Alpha II mounting
- Housing has a circular slot to allow the wires to spin with the mounting ring
- Housing has a gusset that both supports the back and limits rotation of the mounting ring to +/-180 degrees instead of full 360
- Concept 1 has the same mounting ring as the Alpha II, but the T-Nut has been moved to the side and is mounted side by side with the RJ11 jack
- Concept 2 has the same mounting ring as the Alpha II, but it has been thickened and the RJ11 jack has been mounted in line with the T-Nut.

Proposed Advantages

- Pipe clamp would help secure both halves of the main enclosure together
- Mounting and cable come out on the same axis

Proposed Disadvantages

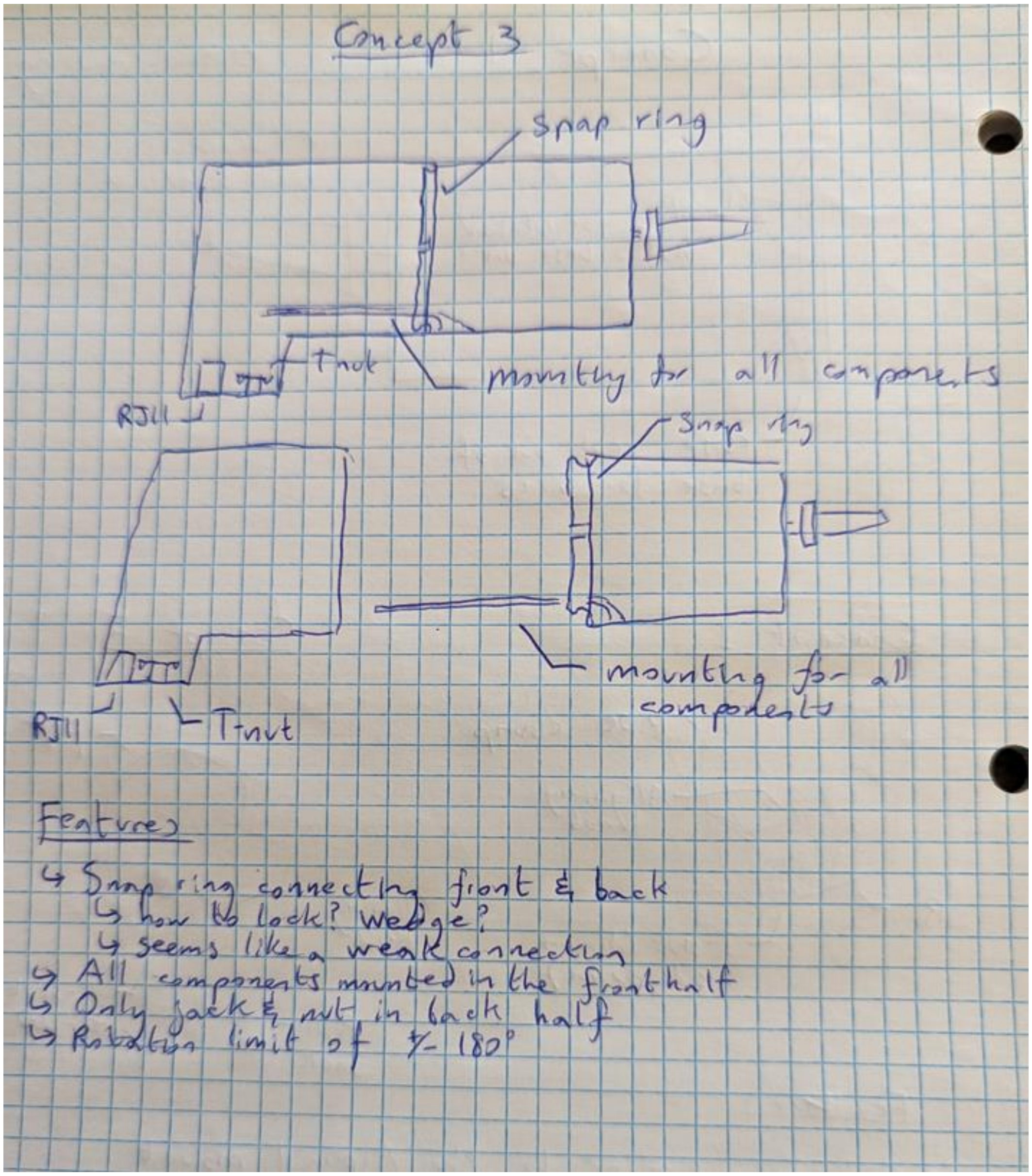
- Back of the housing would be pretty fragile
- Wires would be outside the enclosure going from main unit to ring

OFIs

- Two gussets could be used to secure the back further
- Some sort of cowling would be useful to cover the wires going from main unit to the base

Concept 3 – Compliant Rotatable Snap Ring

Concept 3 has the sections of the housing rotate relative to each other, with the rear of the enclosure housing the T-Nut and the RJ11 jack, and the front of the enclosure housing the gimbal assembly. The two halves of the enclosure fit together with a snap ring and are held in place by two push tabs. The tabs can be pushed in to allow the halves to rotate, then released to lock them in place.



Features

- Two halves of the housing connected by a snap fit
- Almost all components contained in the front half
- Only RJ11 and T-Nut in the back half
- Limited to +/- 180 degrees to protect cables

Proposed Advantages

- Aesthetically closest to the original LipSync design
- Does not require any hardware, no screws, no pipe clamps
- All wires and cables kept inside the enclosure

- All one part when assembled, no separate ring

Proposed Disadvantages

- Snap fit ring could wear down over time
- Snap fit ring would be fragile if dropped

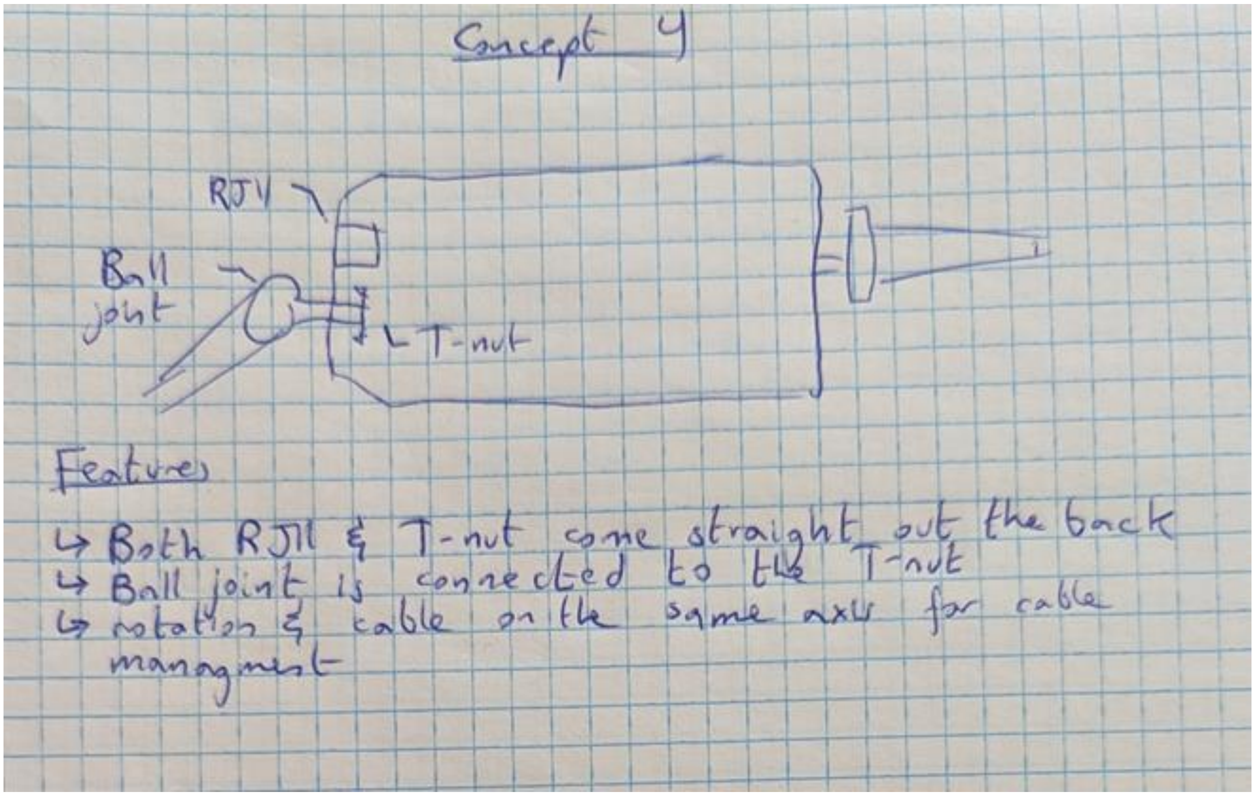
OFIs

- Need to determine how it locks in place
    - Current plan is a series of slots around the rim of the front housing in 20 degree spacing
    - Back housing has a tab (or multiple) that slots in and locks in place and can be pressed in to allow the housing to rotate

Concept was given a test for printability, and it was found to potentially be viable, but the print line orientation weakened both the push tabs on the rear, and the clips attaching the two sides together.

Concept 4 – Cylinder, Both Mount and RJ11 Out back

Concept 4 is the simplest mounting concept, with the enclosure being a single piece with both the RJ11 and the T-nut come out the back of the enclosure. A ball joint mounting arm is required for proper positioning, but it keeps the RJ11 jack colinear with the mounting arm.



Features

- T-Nut and RJ11 come out the back of the enclosure
- Ball joint is connected to the T-Nut
- Rotation and cable along the same axis for cable management

Proposed Advantages

- Sturdy, one piece enclosure

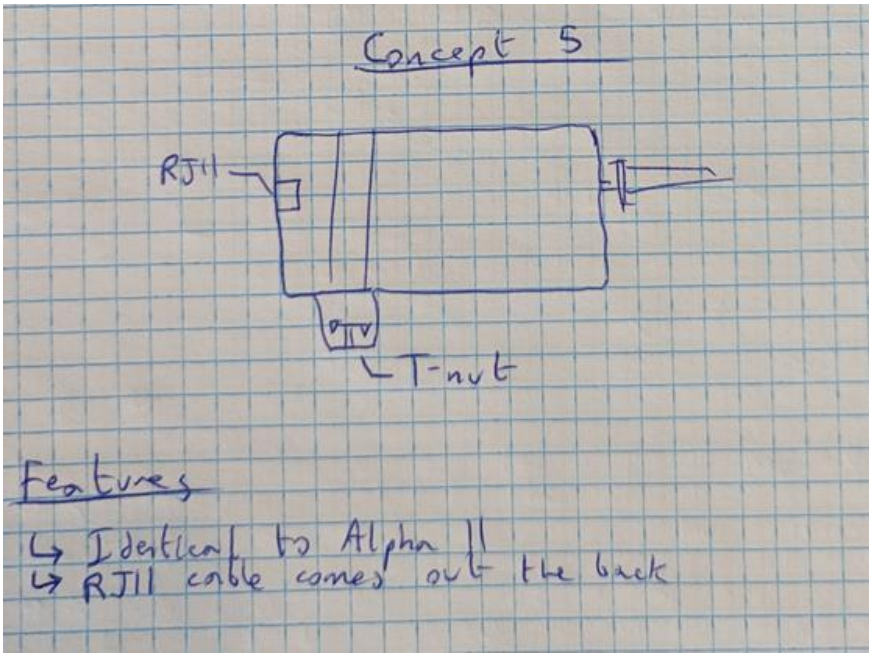
Proposed Disadvantages

- An extra joint in the mounting arm would be required to get the same range of motion as the other/ mounting would have to be in a hemisphere around the back of the device

OFIs

Concept 5 – Cylinder, RJ11 Out back

This is the same mounting solution as the Alpha II, with the microcontroller port replaced with the RJ11 jack. This lets the hose clamp rotate freely without worrying about twisting cables inside the device, but the mounting arm and cable come out of the device in different directions.



Features

1. The same design as the Alpha II
2. RJ11 comes out the back the same as the USB on the Alpha II

Proposed Advantages

- Minimal work required, completely reuse the existing Alpha II mount

Proposed Disadvantages

- Cable comes out a separate direction from the mounting
- Mounting ring is disjointed aesthetically

OFIs

1. Make the mounting ring look less disjointed from the rest of the enclosure

Concept 6 – Rotatable Back, Hirth Thread

This concept has the sections of the housing rotate relative to each other, with the rear of the enclosure housing the T-Nut and the RJ11 jack, and the front of the enclosure housing the gimbal assembly. The two halves of the enclosure are prevented from rotating using a Hirth joint

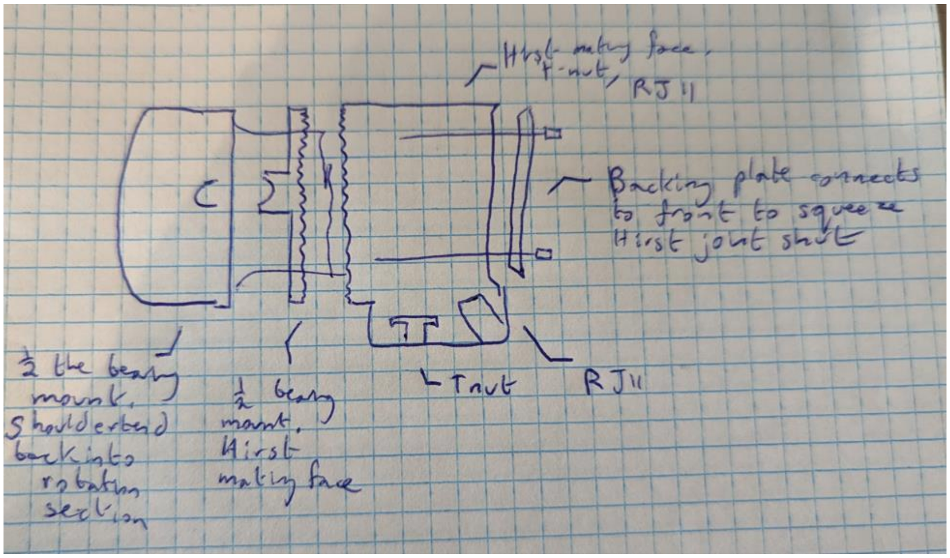


Figure 22. Hirth Joint Concept

Features

1. Joystick assembly could be inserted from front or back depending on sizing.
2. Offers 8 locking positions (or more, with a differently shaped polygon)
3. Locking ring doesn't require tools
4. Cable cover directs cables down behind mount
5. Hirth joint allows for 35-40 different angles
6. Everything is contained within the enclosure, no rotation ring like the Alpha II

Proposed Advantages

- Modular mounting section could have different mounts (e.g., Pipe thread, tee nut)
- Axially symmetric
- Everything is enclosed within the enclosure
- No flexure components

Proposed Disadvantages

- Finite number of mounting positions
- Print orientation could be an issue
- Screws have to be backed off to rotate, so needs tools/can't be done on the fly

OFIs

- Unclear method for attaching rear cover to mounting section

Concept 7 – Rotatable_Mount_Threaded

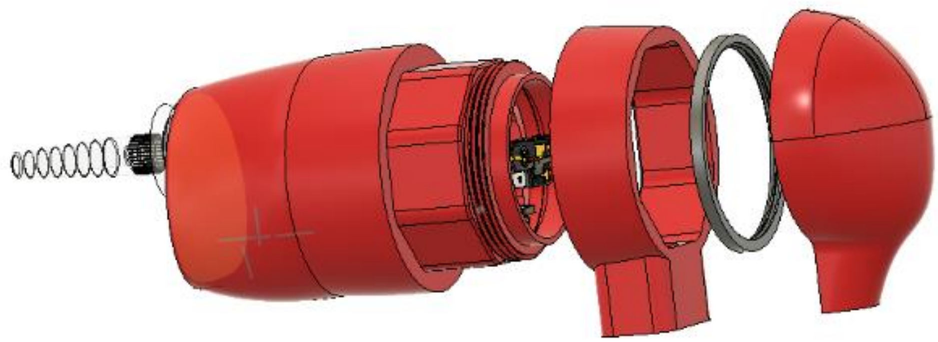


Figure 23. Rotatable Threaded Mount Concept.

Mounting concept using an octagonal mounting ring with a cable cover.

Concept 8 – Joystick Sled

Features

- 4 main pieces
  - Threaded front cap
  - Middle joystick housing
  - Joystick and fixed magnets mount
  - Rear mounting adaptor
- Front cap
  - Possibility of changing the joystick range of motion through mouthpiece opening shape
  - Secures the joystick in place within the middle housing
- Middle joystick housing
  - Holds the joystick secure so that the mounting arm is free to be adjusted without affecting any internal components of the joystick itself or requiring any disassembly of it.
  - Could possibly house all cables so that there is no rotation of connected components relative to one another.
- Joystick and fixed magnets mount
  - Holds the fixed magnets and the joystick securely relative to each other to allow for consistent builds and reduce the risk of shifting during use.
- Rear mounting adaptor
  - Allows for different mounting methods without having to reprint the whole enclosure
  - Will not affect any joystick components (cables, magnets, etc.)
  - Can be designed to allow 360° mounting angles

Proposed Arrangements of Components

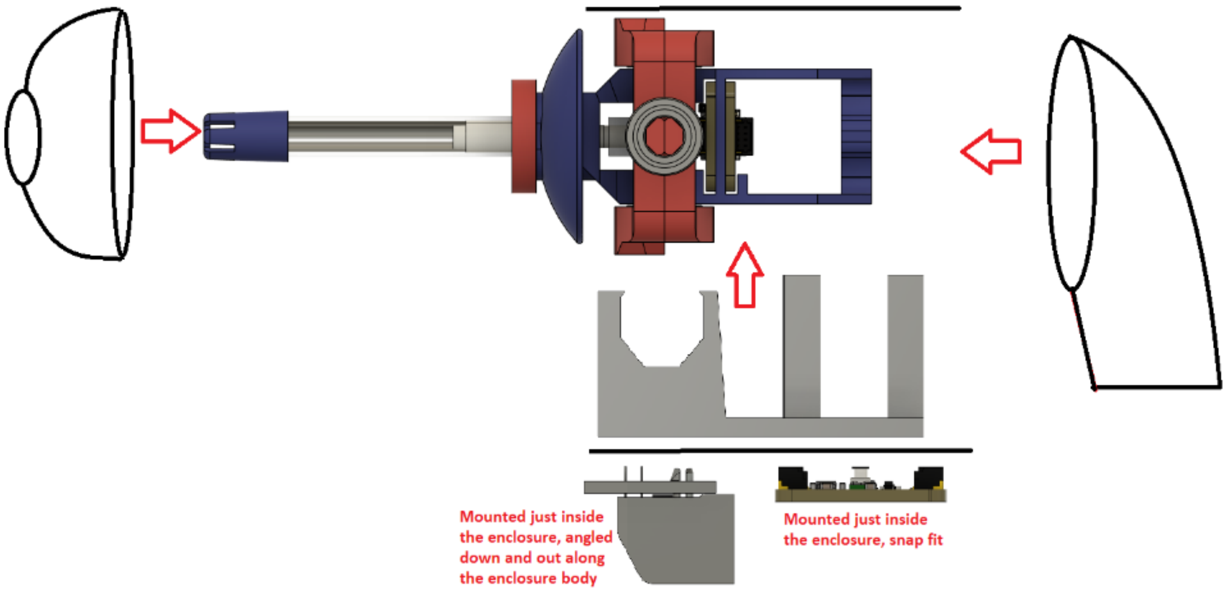


Figure 24. Component Arrangement Proposal

Would require a more secure bearing and magnet retainment and rails for the gimbal/magnet mount to slide into

Print orientation would also be an issue with this concept, and it would require some optimization.

Mounting Concept Selection

A combination of Concept 6 and Concept 8 were selected for the fact that Concept 6 provided a reliable locking mechanism, did not require additional hardware like a pipe clamp, did not require tools to adjust. It also allows for a slimmer overall connection that using a pipe clamp. Meanwhile, Concept 8 provides a secure retainment of the joystick gimbal and ensures the magnets are positioned more consistently relative to the sensor, and aids in the ease of assembly to insert the gimbal into the enclosure.

There was debate as to using splines/teeth as in a traditional Hirth joint or two use flat mating faces. The traditional teeth allows for greater rotational strength and kept it in place while tightening the outer ring, but made it less printable and restricted the LipSync to a fixed amount of mounting angles. Using no teeth makes it easier to print and gives infinite printing angles, but it can shift while tightening, and it needs to be very strongly tightened to keep it from rotating, which can make it difficult to untighten.

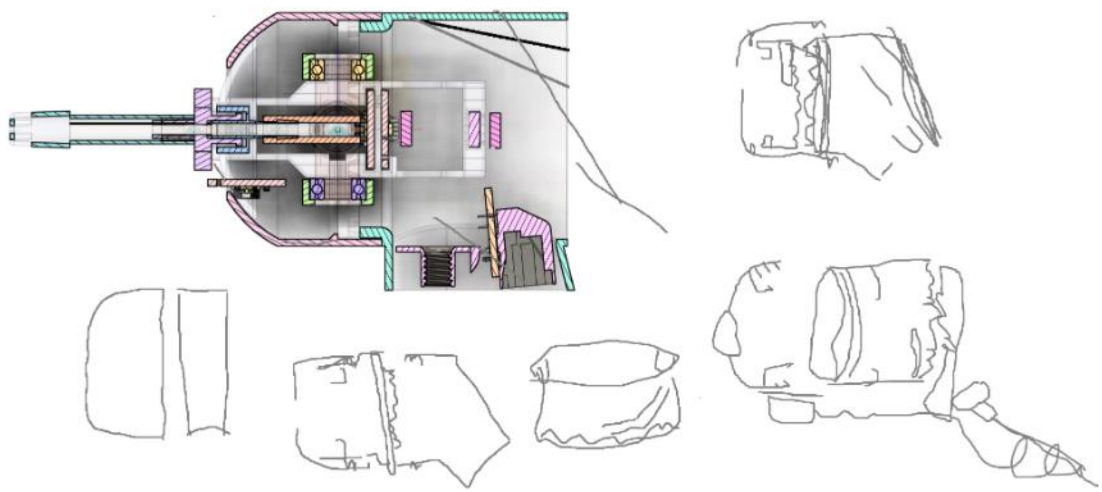
The final decision was to keep the teeth to reduce the risk of overtightening and prevent the angles from slipping while tightening.

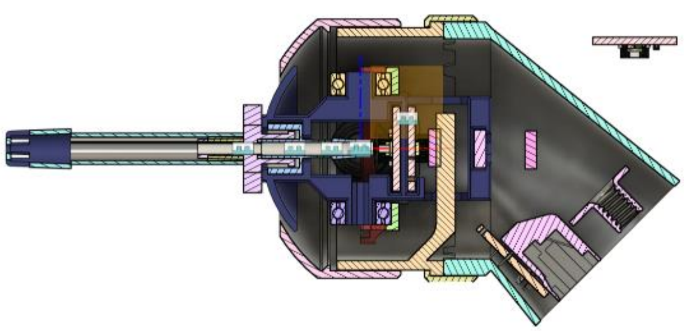
Beta Joystick and Enclosure Prototyping

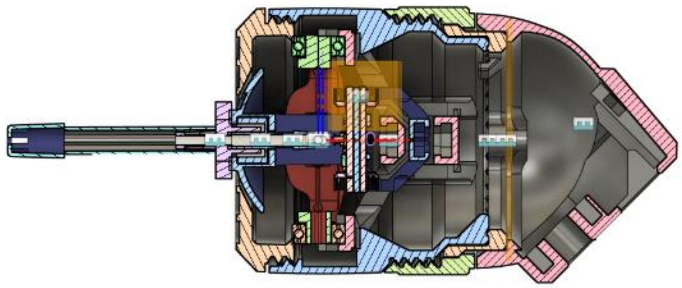
The selected concepts to move forward with were a combination of joystick concepts 1 and 3, where the LPS33HW and TLV493D boards are mounted on the gimbal, while all other boards are mounted off the gimbal. This allows a direct pressure connection to be made without having to design supports around air tube routing and air path connections (tube to filter, tube to sensor, mouthpiece to gimbal, etc.)

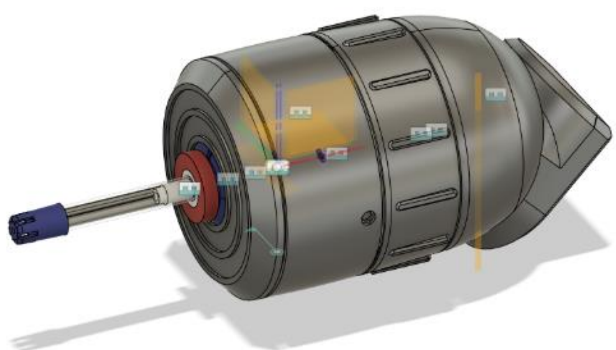
Enclosure Design

After the Hirth joint concept was selected, work began on creating an enclosure that incorporated the Hirth joint and locking rings while keeping the look and feel of the LipSync.









Determining Magnet Direction

The LipSync firmware reads what direction the sensing magnet is facing and corrects the output to accommodate either direction of magnet. However, it is important to be able to determine the polarity of a magnet for the ease of changing out parts in a given LipSync joystick. For example, if the Sled component breaks but the rest of the joystick gimbal is fine, only the sled needs to be changed out, but the direction of the magnets in the sled must match that of the one in the gimbal.

Using Compass

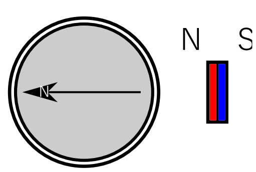


Figure 25. Using compass to determine magnet polarity.

The north side of the magnet will be attracted to the south side of a compass needle.

Small magnetic compasses are available for ~$1 each, but only in quantities of 10-12 (e.g., Amazon: https://www.amazon.ca/Button-Compass-Filled-Camping-Boating/dp/B094JTQZDX). The cheapest option is about $5 (e.g., Coghlan's Ball-type Pin-on Compass, Amazon: https://www.amazon.ca/Coghlans-0126-1200-Ball-Type-Pin-On-Compass/dp/B000LC844Q).

Most phones also have magnetic sensors and have apps available that can act as a compass.

Using Magnetic Sensor

The magnetic sensor can also be used to determine the polarity of a magnet. When the reading from the magnetic sensor direction z-axis is positive, the north side of the magnet will be oriented up from the top of the board.

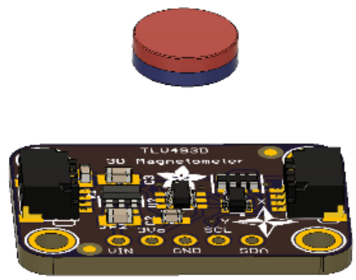


Figure 26. Using Magnetic Sensor to determine magnet polarity.

Using Known Magnet

A marked or known magnet can also be used to determine the polarity of an unmarked magnet. An alternative method could be used to determine a single magnet, and then that known magnet could be used to determine the polarity of additional magnets.

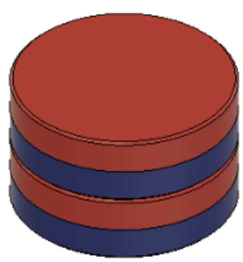


Figure 27. Using a known magnet to determine magnet polarity.

A jig may be a useful way to determine the polarity of multiple magnets at a build event.

In this case, we can also consider the other magnet in the joystick be matched to as a “known” magnet as we know the magnet(s) being added must match the other magnet’s direction.

Marking Magnet Direction

Once the orientation of the magnet is determined, it could be marked by the maker to aid in further steps, for example using tape, permanent marker, or perhaps nail polish. A permanent marker (Sharpie) was tested, but the marking did not stay on the smooth coating of the magnet long enough to be useful.

Magnet Stacking

The concept of stacking magnets was explored as a potential option for changing the operation force of the joystick to create a stiffer or looser mechanism for different users.

When stacking magnets, the magnetic force increases until the thickness of the stack reaches the diameter of the magnets [first4magnets.com]. Since the magnets used in the LipSync are 2.5mm x 8mm, 3 to 4 can be stacked before reaching diminishing returns. This is demonstrated in a graph showing the force vs thickness of the magnet. Note the X axis does not increase linearly. [totalelement.com]

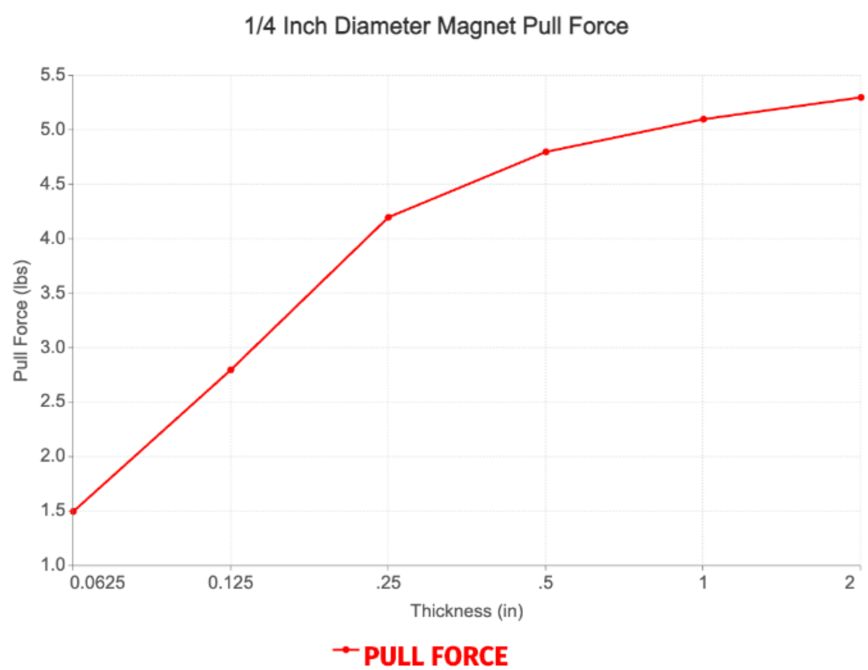


Figure 28. Magnetic Force with Greater Thickness.

Final Joystick Design

The final Joystick design is based around the following primary components:

- STEMMA TLV493D triple-axis magnetometer break out board from Adafruit
- STEMMA LPS33HW pressure sensor breakout board from Adafruit
- STEMMA LPS22 pressure sensor breakout board from Adafruit
- 4 Bearings (0.25” ID x 0.5” OD ball bearings)
- RJ11 jack and breakout board from Sparkfun
- ¼-20 t-nut
- Male Luer Lock Connector
- 3 Magnets

Joystick System

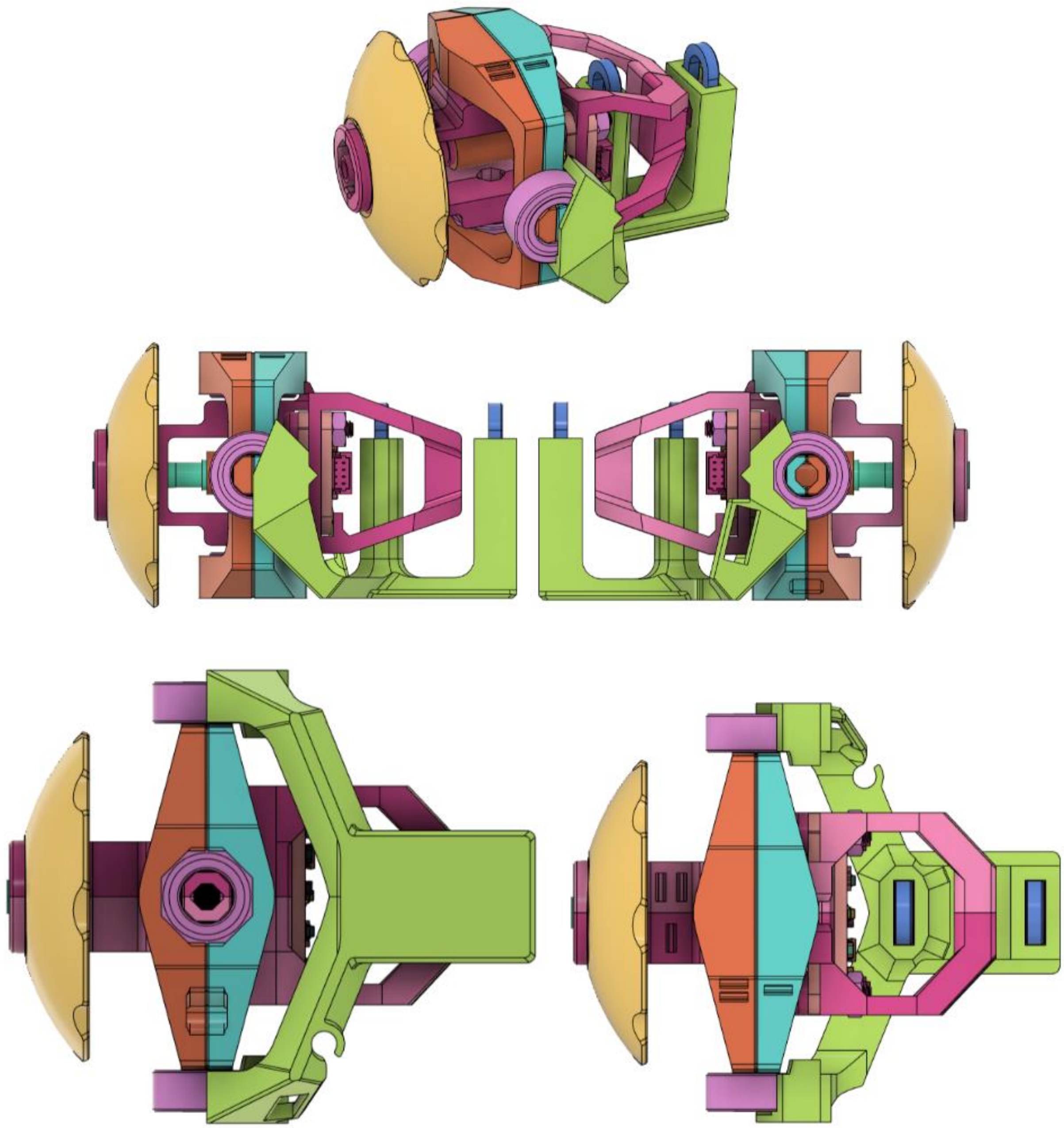
- Joystick Mechanism
  - Using magnetometer breakout board and a magnet for measuring deflection
  - Using bearings in a gimbal arrangement for axis movement
  - Magnetically centered
- Sip and Puff System
  - Direct connection airpath within the gimbal
  - Mouthpiece
  - Mouthpiece connection using Male Luer Lock Connector
  - Using LPS33HW sensor for reading sips and puffs
  - Using LPS22 sensor for reading the ambient pressure
- Enclosure
  - Rotatable Mounting
  - Mounting Interface
    - ¼-20 T-Nut
  - RJ11 port for connection to the Hub

Joystick Mechanism

The Joystick Mechanism is modified version of the Gimbal from Concept 3, with the ambient pressure board moved off the gimbal. The selected design features a moving magnetic sensor mounted on the gimbal and a static sensing magnet for deflection measurements. For centering, the gimbal uses a pair of magnets located to the rear of the sensing magnet. One magnet is in line with the magnetic sensor, and the other is static, in line with the sensing magnet. This keeps the magnets further from the front of the device and lowers the risk of interference with outside sources. The Airpath features a direct connection within the gimbal from the mouthpiece Luer connector to the pressure sensor.

1. The cable routing path has been made such that there is only one way the gimbal pieces can be assembled with the cable successfully routed through them.

2. The octagonal bearing shafts are dimensioned 6.35 mm between opposite corners and drafted inward at a 1-degree angle for ease of sliding the bearing on. The length of each shaft from base to end is 4.9 mm.
3. Joystick has hard stops set to ±10°
4. Horseshoe pin for magnet retainment



Magnetic Arrangement

The lever arm length was derived through testing with the Infineon magnetic design tool and then through prototype testing. The arrangement of the magnets is primarily dictated by the lever arm and air gap, selected to give an ideal output range for the angle of deflections being used in the joystick. The pair of centering magnets are spaced as close as possible while still leaving clearance for the joystick movement. The central axis of the joystick, from the mouthpiece to the rear most magnet, lines up along the magnetic sensor center point.

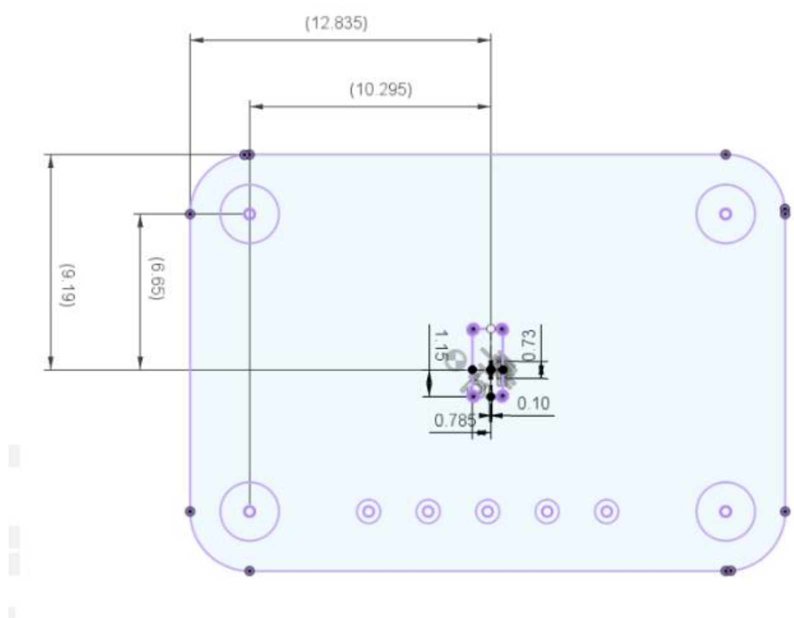


Figure 29. Magnetic Sensor Center on Breakout Board.

The sensor board is mounted using nylon M2.5 machine screws and nuts within the inner gimbal.

Magnet Specifications

The LipSync uses a rare earth magnet to produce the magnetic field for the sensor to measure and to produce the centering force.

The magnet is an 8.00 mm diameter, 2.50 mm thickness cylindrical magnet from Radial Magnets, Inc. This shape of magnet may also be referred to as a disc or pill magnet. This magnet is a Neodymium Iron Boron (NdFeB) magnet with a grade of N35SH. The magnet is magnetized through the thickness of the magnet (i.e., one of the round surfaces has a north pole, the opposite round surface has the south pole). To prevent corrosion, this magnet has a Nickel-Copper-Nickel coating.

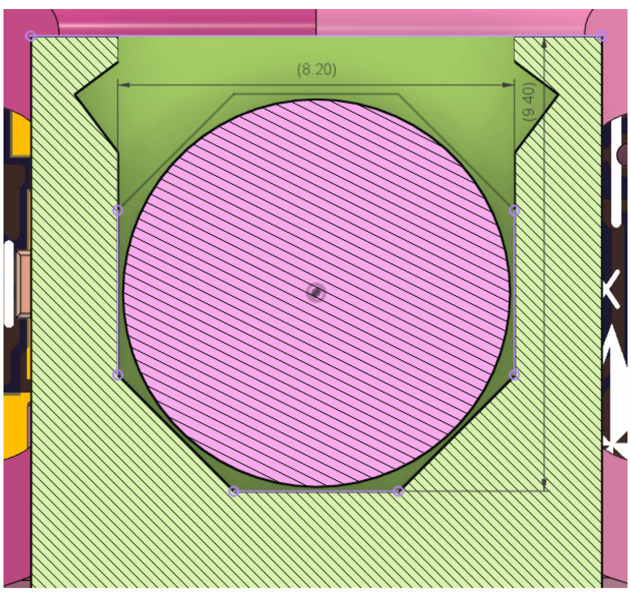
Table 2. Neodymium Magnet Physical Properties[1]

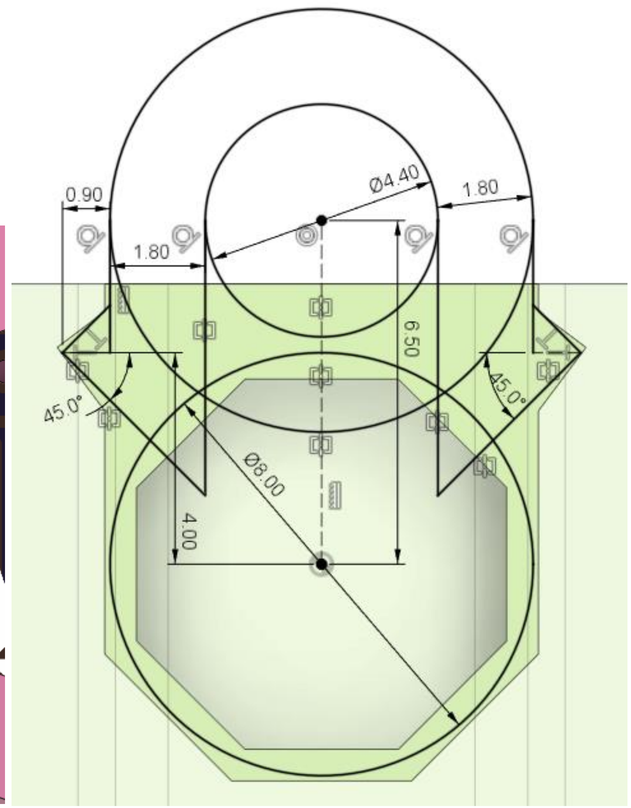
Grade	Residual Flux Density (Br)	Coercive Force (Hc)	Intrinsic Coercive Force (Hci)	Max. Energy Product (BH)max
N35SH	11.7-12.1 KGs	>10.8 KOe	>20 KOe	33-35 MGOe

This magnet model was selected by compiling a list of magnets available through Digi-key and Mouser and using the Infineon Magnetic Design resources.

Magnet Retainment

The horseshoe pin was selected for the final magnet retainment in the sled due to its compliant nature that allows it to be inserted even with tighter print tolerances, and because it leaves the magnets removeable. Due to the shape of the bump, the pin will push down on the magnet even if not fully engaged with the cutouts in the magnet slot.



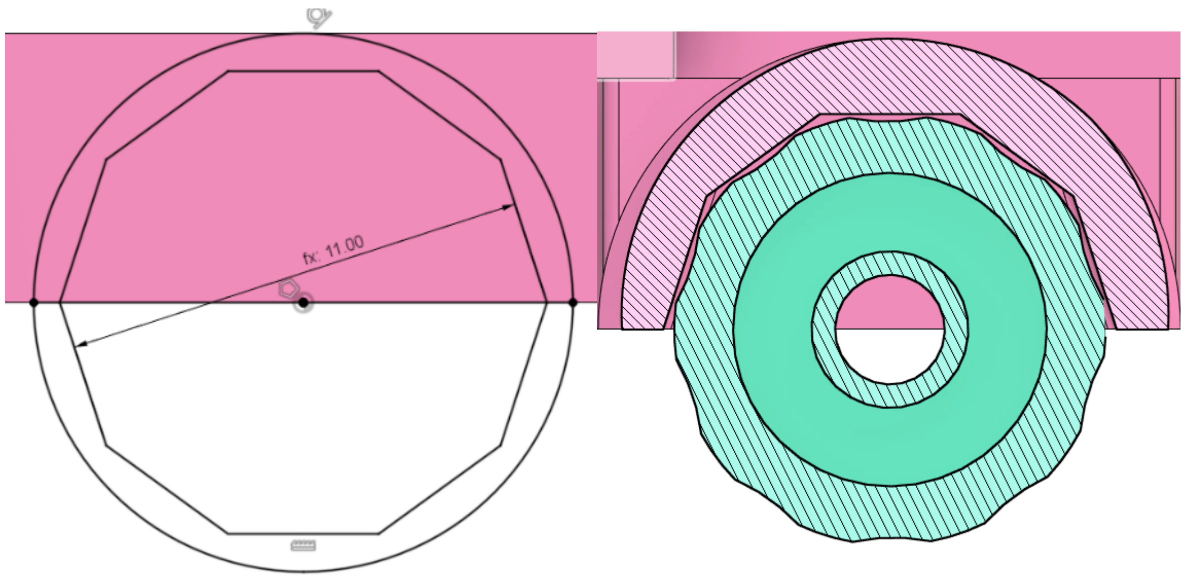


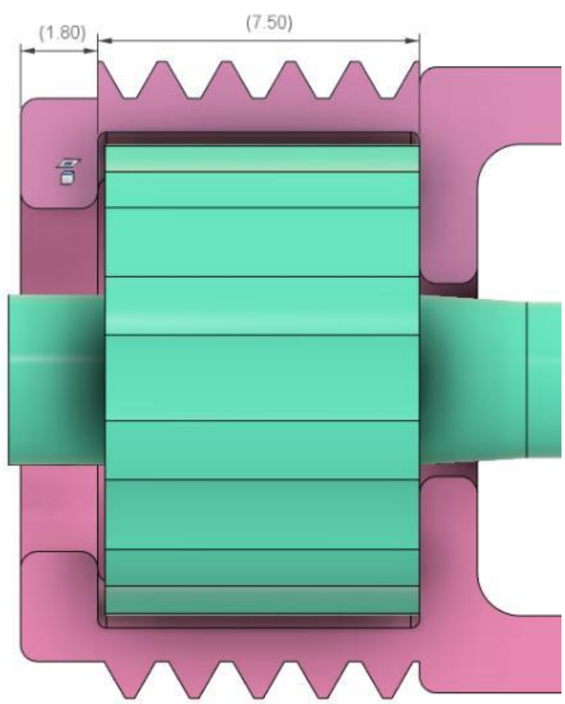
The thickness of the magnet slot is 2.6 mm.

Male Luer Lock Connector Retainment

The Luer Lock connector retainment is a decagon dimensioned 11 mm between parallel flat sides.

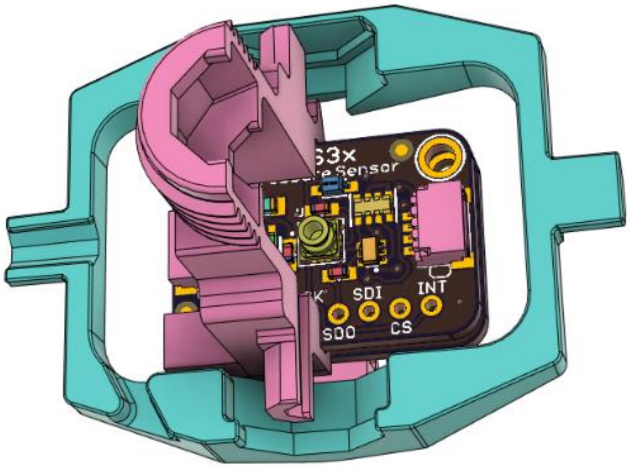
[1] Neodymium Magnet Physical Properties. K&J Magnetics, Inc. https://www.kjmagnetics.com/specs.asp

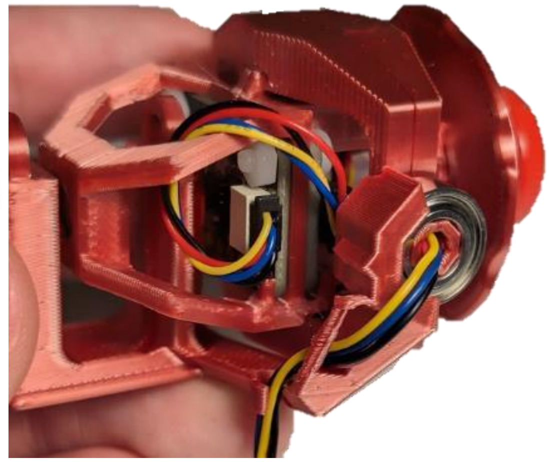




Cable Routing

The cable routing is designed such that the cable gimbal can only be assembled in one orientation with the cable successfully routed through. This is through blocking off the axes that we don’t want the cable to pass through. The sled has some additional cable management features to help protect and guide the cable when the gimbal subassembly is inserted into the enclosure.





CAD Model Parameters

Table 3. User Parameters Used in CAD Model

CAD Parameter	Value	Description
Outer_Bearing_Width	40.5 mm	Inner distance between outer gimbal bearings
Inner_Bearing_Width	22.5 mm	Inner distance between inner gimbal bearings
Lever_Arm	12.5 mm	Distance from center of rotation to magnetic sensor face
Air_Gap	4 mm	Distance between magnetic sensor face to the sensing magnet
Moving_Magnet_Gap	6.4 mm	Gap between the inner faces of the sensing and moving magnets
Centering_Magnet_Gap	5 mm	Distance between centering magnet inner faces
Added_Clearance	0 mm	To accommodate 3D printer tolerances
Wall_Width	0.45 mm	Single extrusion width for 3D printers

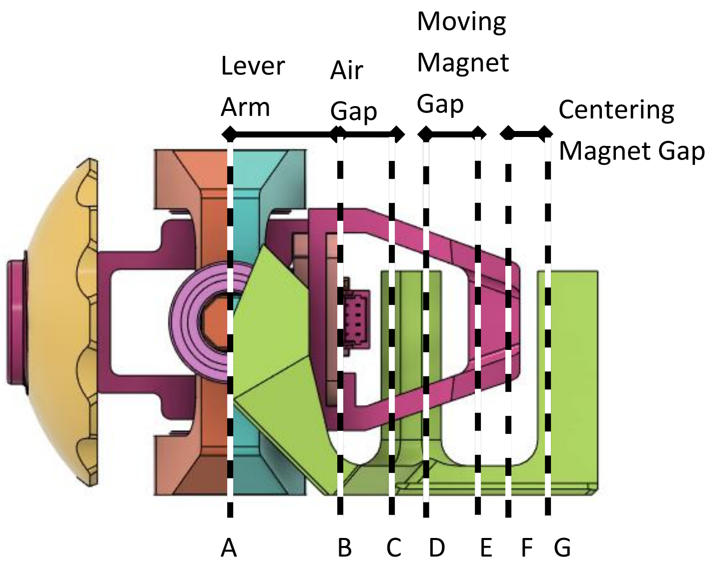


Figure 30. Parameterized Gimbal

A: The Center of Rotation (CoR) of the gimbal.
B: The TLV493D magnetic sensor face.
C&D: The front and back face of the sensing magnet.
E&F: The front and back face of the moving centering magnet.
G: The front face of the fixed centering magnet.

Wall Width

The wall width was chosen as integer multiples of Prusa's default extrusion width 0.45 mm. Further research showed that the ideal wall thickness for 0.2 mm layer height is actually 0.86 mm[2].

Added Clearance

The added clearance parameter is to accommodate the tolerance amongst various 3D printers. There is clearance built into each fitted feature of the model, and the added clearance parameter is included in these dimensions to increase or decrease the fit sizes marginally. As a default it is 0.

Sip and Puff System

The Sip and Puff System is made up of the airpath and the pressure sensors. Within the joystick design, the important factors are how the mouthpiece is connected to the pressure sensor and that we have both pressure sensors connected. The decision was to have the LPS33HW sensor mounted within the gimbal, opposite the magnetic sensor. The LPS22 sensor will be located external to the gimbal, within the enclosure.

[2] Layers and perimeters | Prusa Knowledge Base. https://help.prusa3d.com/article/layers-and-perimeters_1748

Airpath Connections

From the Luer connector to the pressure sensor, the airpath consists of an 1/8" tube fit to both the pressure sensor spout and the Luer connector barb. The Luer Connector and pressure sensor are located near each other so the required tubing is only up to 1.5 cm long.

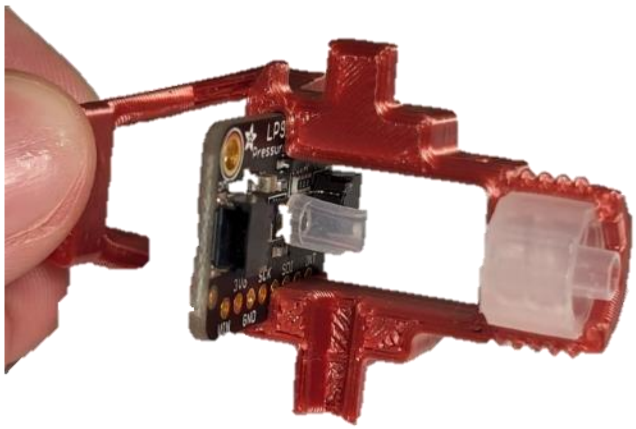


Figure 31. Sip and puff airpath components

Airway Path

In the original LipSync, the airpath components were largely sourced from medical device components suppliers like Qosina. Additionally, one of the parts was only available to be purchased by organizations, not individual makers. The LipSync redesign offered a chance to source a new mouthpiece, filter, and other connectors to create a new airpath that was both cheaper and easier to source.

Mouthpiece Design

The goal of the airpath selection was to find a new airpath for the LipSync that was cheaper than the existing airpath, and all the parts are easily available to individual makers.

Objectives

Requirements

1. Each mouthpiece should cost less than 5 dollars
2. The mouthpiece should be able to be assembled within 5 minutes
3. The mouthpiece should not pose any danger to the user, either as a choking hazard or a food safety hazard
4. All parts must be easily obtained by an individual maker
5. The mouthpiece must form a mechanically secure connection to the gimbal

Initial Decision

The initial mouthpiece design used a section of dental straw, a syringe filter, and a section of plastic tubing to make a mouthpiece similar to the original LipSync, but at a fraction of the cost. The mouthpiece connects to the gimbal using a Luer lock connecter. The dental straw/slip fit filter mouthpiece was the cheapest and most accessible combination at under 2 dollars per mouthpiece and an eighty-dollar maker cost, but the slip fit was potentially a mechanically weaker joint than the Luer lock connection from the previous airpath.

Name	Image	Price ($per piece/ $per package)	Link
Dental Straw		$0.40/$23.99	Amazon
PTFE Syringe Filter		$0.40/$39.99	Amazon
Male Luer Lock Connector		$0.40/$11.99	Amazon
Silicone Food Grade Tube		$0.10/$12.49	Amazon

First Iteration

Testing in the internal build event showed that the dental straw's D shaped internal profile led to a slight air leak in the mouthpiece between the straw and the filter. Food safe silicone sealant was used to seal this leak, as well as provide additional strength to the mouthpiece connection.

However, the silicone sealant takes 24 hours to fully dry, and up to a week to reach full strength. This does put a delay on how quickly mouthpiece can be made and delivered to the user, as well a delay on how quickly a maker can use the mouthpiece for testing.

Second iteration

In the first user testing, the blue tip of the dental straw detached in the user's mouth, both posing a severe choking hazard, as well as leaving an exposed length of wire coming out of the mouthpiece. Because of this, the mouthpiece was redesigned to no longer include the blue tip of the dental straw, replacing it with a length of silicone straw that fits snugly over the shortened dental straw, and is sealed at the base of the filter to maintain airtightness. A zip tie is optionally used to further attach the silicone straw to the dental straw.

Pressure Sensor Tube Connection

Objective

1. Need a reliable way to connect the sip and puff pressure tube to the LPS33HW sip and puff pressure sensor.

Requirements

1. The connection must be secure.
2. The connection must provide adequate bend radius so that the pressure tube does not get kinked.
3. The connection must not exert side shear force on the pressure sensor.
4. The connection should not require the use of specialized tools or equipment.

First Iteration

The description on the supplier website suggest that a o-ring can be used, and links to a [Oil-Resistant Buna-N O-Ring, 1/32 fractional width, Dash Number 001-1/2]().

a. 1/8” OD (0.15” Actual)
b. 1/16” ID (0.07” Actual)
c. 1/32” Width (0.04” Actual)

This approach was abandoned due to difficulty sourcing the o-ring and the tiny size of the device.

Second Iteration

A piece of silicone tubing 2mm ID and 4mm OD was used to connect the barbed end of the Luer connector to the port on the pressure sensor. The overall assembly was squeezed

Joystick Enclosure

The Joystick enclosure is made up of a front cap, a Joystick housing, inner and outer locking rings, and a rear housing.

The Joystick Housing houses the gimbal and the LPS22 pressure sensor, while the Rear Housing houses the RJ11 port and ¼-20 t-nut.

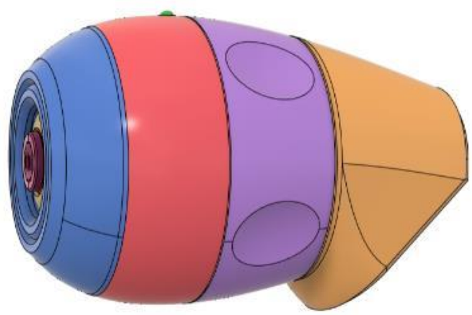
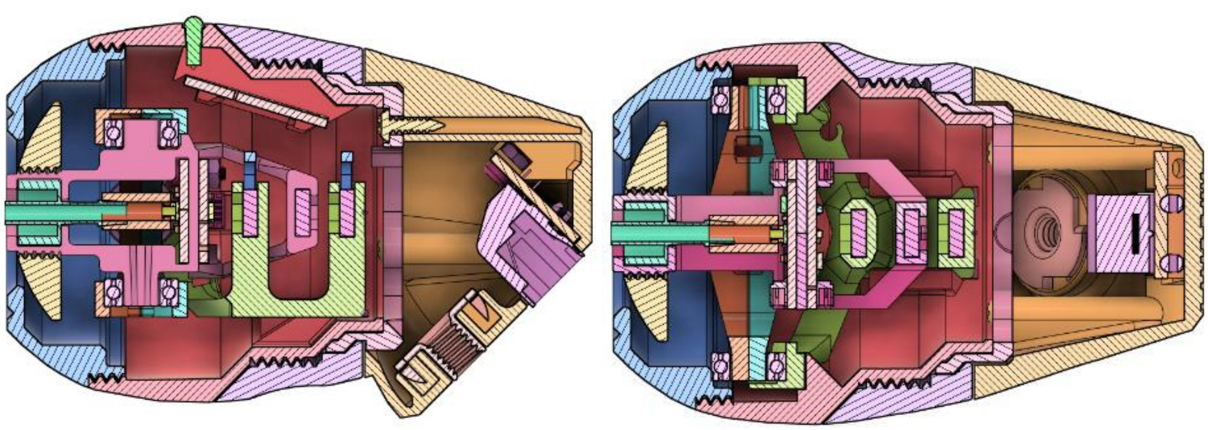


Figure 32. Joystick components.

The design goal for the outer profile was to reduce the front diameter of the device and keep a smooth continuous profile from front to rear. To do this, a spline was created as a base parent sketch and referenced inside the shell sketch of each exterior enclosure component.

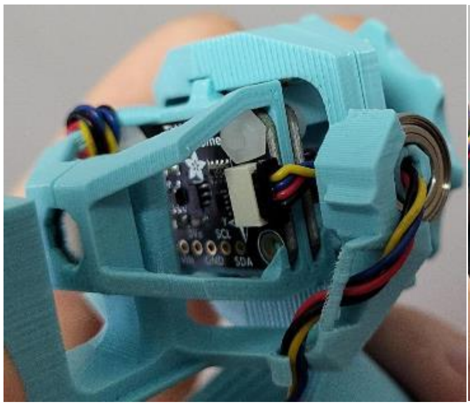
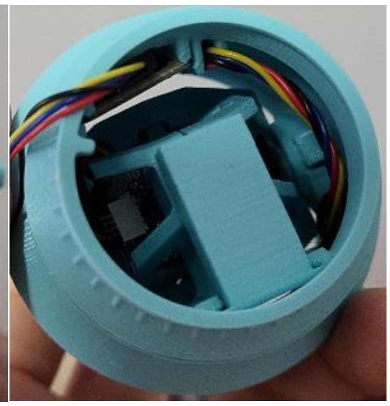
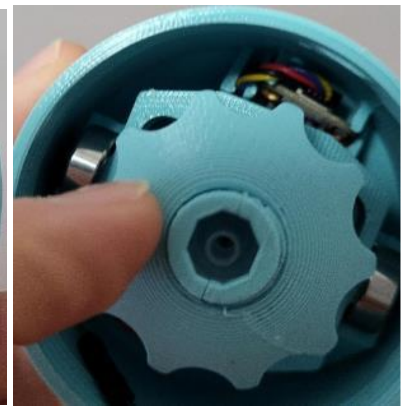


The front inner diameter of the joystick is set to approximately 60 mm. This is required so the gimbal assembly can be inserted into the joystick housing. The resulting final front diameter is 40 mm, the maximum outer diameter is 65 mm, and the length is 100 mm.

The LPS22 sensor board was placed in the Joystick_Housing at the top to use the LED to indicate power and the "up" direction of the joystick.

Enclosure Cable Management

The cable management feature used on the Joystick_Housing test print proved to be difficult to print, and the way the cable was mounted through the gimbal, there was not enough excess cable to utilize the cable management feature. For the final design, cable management was incorporated into the sled to guide the cable around to the rear without risking pinching it in the process of inserting the gimbal into the Joystick_Housing. To deal with the excess cable sitting in the back of the Joystick_Housing, a press fit section was added up beside the LPS22 snap fits so the cable could be pushed into a slot that keeps it from interfering with the moving gimbal.


Locking Ring

The outer Locking Ring uses M52x2 threads for simplicity and ease of assembly.

Rotational Locking

The rotational locking on the Beta II design and model rebuild both removed material from the print face of the Joystick Housing. As the Joystick Housing is a tall print, reducing the surface area for print bed adhesion in any way is undesirable. To remedy this, the cutouts were moved onto the inner locking ring, and the bumps/ribs were moved the first 45-degree expansion of the Joystick Housing body.

RJ11 Jack Placement

The RJ11 connector was mounted flush with the rear of the enclosure to improve the ease of access to the cable, reduce the overall enclosure size, and to keep all mounting and cable attachments streamlined.

Low Poly Texturing

LipSync V2 and V3 featured a "low-poly" or low polygon texture on the outside surfaces of the housing. This texture not only adds an aesthetic look, but it functionally helps to reduce the appearance of the layer lines, making the device look less 3D printed. This technique was also used in V4 both for the functional reasons and to link it visually with the previous version.

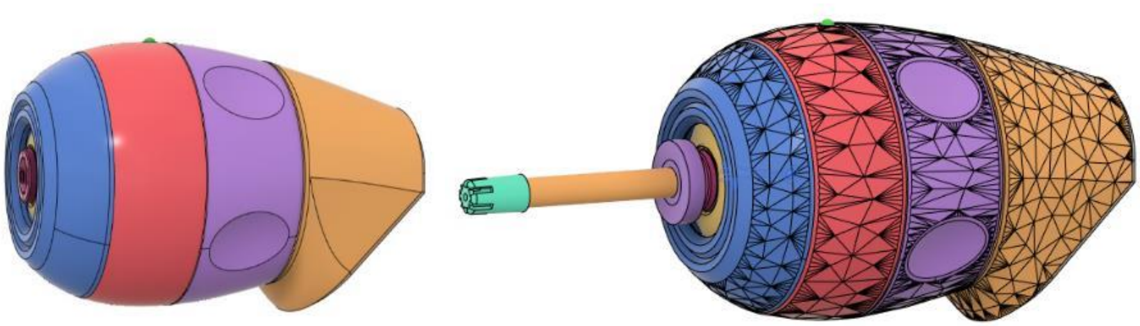


Figure 33. Joystick before (l) and after (r) low-poly texturing.

LipSync Hub Design

LipSync Hub Overview

The LipSync Hub is intended to serve two primary purposes. First, it is intended to reduce the size of the joystick unit. Second, it is intended to provide a better interface for providing feedback to the user.

Hub User Requirements

1. Provides visual and audible feedback to the user on the operation and state of the LipSync
2. Provides a way for the primary user to adjust settings and control the operation of the LipSync
3. Can be mounted so that it is visible to the user and close to the host device.
4. Is simple to understand and operate
5. Provides a way for secondary user to activate sip and puff inputs
6. Provides a way for secondary user to adjust settings and control the operation of the LipSync
7. Provides a way to connect external switches for an alternative to sip and puff or as additional inputs.

Hub Functional Requirements

1. The Hub must contain the microcontroller.
2. The Hub must contain a means for providing audible feedback.
3. The Hub must contain a means for providing visual feedback.
4. The Hub must contain ports for attaching external switches.
5. The Hub must contain tactile buttons intended for use by a secondary user.
6. The Hub must contain a means for connecting to the LipSync Joystick.
7. The Hub must contain a means for a secondary user to reset the microcontroller.
8. The Hub must contain a mounting interface.
    8.1. The mounting interface must comprise a ¼”-20 UNC mounting thread.
9. Input ports on the Hub must be labelled.
10. The primary user must be able to disable the audible feedback.
11. The primary user must be able to disable the visual feedback.

Hub Constraints

1. The Hub must be comprised of components that are one of the following: off-the-shelf, commercially available parts; 3D printed components; or custom printed circuit boards (PCBs).
2. Custom printed circuit boards (PCBs) must be able to be produced at prototyping services.
    2.1. Custom PCBs must have a maximum dimension of 100 mm x 100 mm or less.
    2.2. Custom PCBs should have a maximum routing area of area of 80 mm x 100 mm.
    2.3. Custom PCBs must be limited to 2 layers.
3. Custom printed circuit boards (PCBs) must be friendly for volunteer makers.
    3.1. Custom PCBs must contain components that are one of the following: thru-hole components, breakout-boards that interface through thru-hole headers, components that connect via STEMMA QT cables.
    3.2. Custom PCBs must minimize or eliminate the use of surface mount components.

Hub Design Process

Selection

The overall criteria for selecting the optimal Hub design are as follows:

1. Size
2. Ease of use
3. Makeability
4. Availability of suitable microcontroller.

LipSync Hub Conceptual Design

The LipSync Hub design was based around selecting an option for visual feedback, choosing an appropriate microcontroller and circuit board concept, and then the overall enclosure.

Visual Feedback Approach Possible Solutions

Several options were considered for how visual feedback would be portrayed to the user:

1. Discrete LEDs
2. Discrete Neopixel LEDs
3. Neopixel LED Ring
4. Display

Screen

Pros

- More intuitive use for changing settings
- Less reliance on user memory / cognition
- Easier to connect than individual neopixels
- Easier to add additional features to menus / screens
- More intuitive visual calibration feedback

Cons

- Added cost
- Need to size / position appropriately to see
- Relies on vision
- Limited lifetime of the OLED display (starts to dim after 1000 hours, operating lifetime 10,000 hours, storage lifetime 20,000 hours)

Microcontroller

Several different options were considered for the microcontroller.

Table 1: Technical information on considered dev boards.

Dev Board	Adafruit Feather nRF52840 Sense	Circuit Playground Bluefruit	XIAO nRF52840
Ecosystem	Adafruit Feather	Arduino, CircuitPython	XIAO/QT py
MCU	nRF52840 Cortex M4 at 64 MHz	nRF52840 Cortex M4 at 64 MHz	nRF52840 Cortex M4 at 64 MHz
Clock Speed(MHz)	64 MHz	64 MHz	64 MHz
Total pins	28	14 pads	11
Digital pins/pads	21	8	11
Analog pins	4 (12-bit)	6 (12-bit)	6 (12-bit)
DAC output	2	?	1
Capacitive touch pins	Through Software and requires 1M resistor	7	Through Software
PWM pins	12	2	11
Castellated pads	None	No	Yes
FLASH memory	1 MB	3 MB	1 MB, 2 MB QSPI flash
SRAM	256 KB	256KB	256 KB
EEPROM memory	None	None	None
Exposed Interfaces	1 I2C, 1 UART, 1 SPI ,1 SWD	1 I2C, UART, SPI, SWD	1 I2C, 1 UART, 1 SPI ,1 SWD
Input Voltage (Limit)	3.5 to 5	3.3V	3.3V
Power Voltage	3.3V	3.3V	3.3V
Voltage Regulator	AP2112K-3.3	AP2112K-3.3	BQ25100
USB Type	Micro-USB B	Micro-USB B	USB C
Battery Charger	Yes, 200mA+ lipoly charger	None	None
Buttons	RESET, SWITCH	RESET, 2 built in buttons, slide switch	RESET
Native USB	Yes	Yes	Yes
Onboard RGB LED	RGB NeoPixel LED	RGB NeoPixel LEDs, 10 built-in and controllable	3-in-one LED
Qwiic/ Stemma/ Grove Connector	None	None	None
USB HID	Yes	Yes	Yes
XAC HID Compatible	Yes	Yes	Yes
Bluetooth Built-in support	Yes, nRF52840	Yes, nRF52840	Yes, nRF52840
Bluetooth Mouse HID Support	Yes	Yes	Yes
Bluetooth Switching			

Wi-fi	None	None	None
IMU/ Accelerometer/ Gyroscope	LSM6DS33	LIS3DH 3-axis XYZ accelerometer	None
Microphone	MP34DT01-M	SPW2430HR5H-B microphone,	None
Proximity, Light, Gesture Sensor	APDS9960	ALS-PT19-315C/L177/TR8 (Connected to Pin 8)	None
Temperature, barometric pressure Sensor	BMP280	Murata NCP15XH103F03RC (temperature only)	None
Humidity sensor	SHT30	None	None
Bootloader	Arduino or CircuitPython	Arduino or CircuitPython	Arduino or CircuitPython
Software compatibility	Arduino or CircuitPython	Arduino or CircuitPython	Arduino or CircuitPython
Dimensions (mm)	51.0 x 23.0 x 7.5	50.6 diameter	20 x 17.5x5.7
Mounting holes	4 x Corners	Pads	None
Weight (g)	5.5	8.9	5
Cost (US$)	32.5	24.95	$10.38
Cost (CAD$)	$57.28	$36.18	$15.51
Release Date	Mar 11, 2020	Sept. 12, 2019	2021
Other		Internal speaker for sound (can play songs and tones)	NFC

Table 2: Availability of the three boards as of 2023-07-11

Distributor	Adafruit Feather nRF52840 Sense	Circuit Playground Bluefruit	XIAO nRF52840
Digikey	0	0	1,569
Mouser	273	0 (151 expected September 15)	602
Adafruit	0	0	None
Other		In stock (numbers not given)	In stock (numbers not given)

PCB Concepts

Initial PCB Concepts

Concept 1: Circuit Playground Bluefruit with axially assembled jacks/plugs.

Concept 2: Can use either the Circuit Playground Bluefruit or the Xiao with a Neopixel ring.

Concept 3: Xiao with the Neopixel ring that is part of the joystick. The joystick mechanism would sit in front of the LED ring. It would then connect to a Hub like the Forest Hub. Switches and everything would be on the Hub. Can use a Seesaw board instead of the Xiao in the Forest Hub.

Concept 4: Similar to Alpha II, but using the Xiao. Rest is the same as the Alpha II.

Concept 5: Hub concept with three buttons, and three jacks (which map to the same functions). Status light for showing mode/is on/off. Additional Seesaw to add on Neopixel ring if user wants.

Circuit Playground Bluefruit Concept

Microcontroller: Circuit Playground Bluefruit https://www.adafruit.com/product/4333

Breadboard-Friendly 3.5 mm Stereo Jack: https://www.adafruit.com/product/1699

Adafruit: https://www.digikey.ca/en/products/detail/adafruit-industries-llc/1699/8605093?s=N4IgTCBcDaIIYBM4DMBOBXAlgFwAQEYA2ATmJAF0BfIA

Generic: kycon stx-3120-5B - https://www.digikey.ca/en/products/detail/kycon-inc/STX-3120-5B/9990114

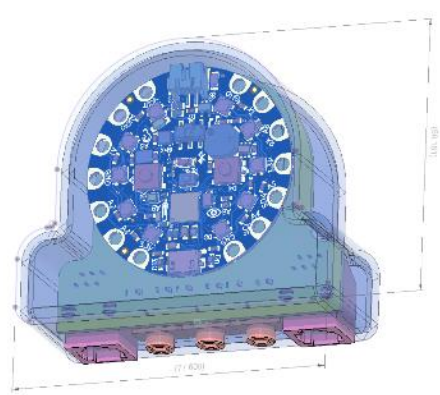


Figure 34. Circuit Playground Hub Concept

XIAO Concepts

Concept A: 3 LEDs and Designated Mode Change Inputs

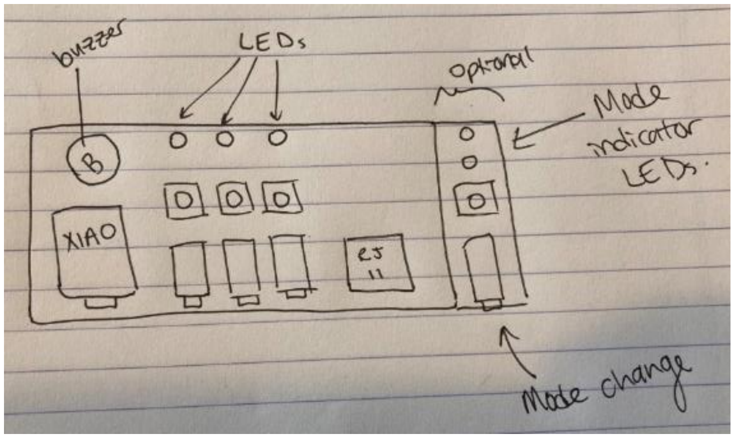
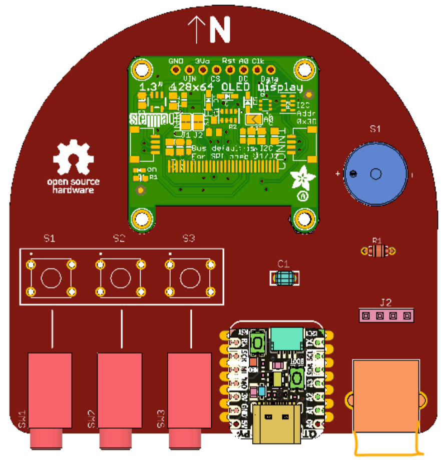


Figure 35. Initial PCB Concept featuring 3 LEDs, 3 Switch inputs, and additional dedicated input for changing mode.

Concept B: Hub w/ Integrated Screen

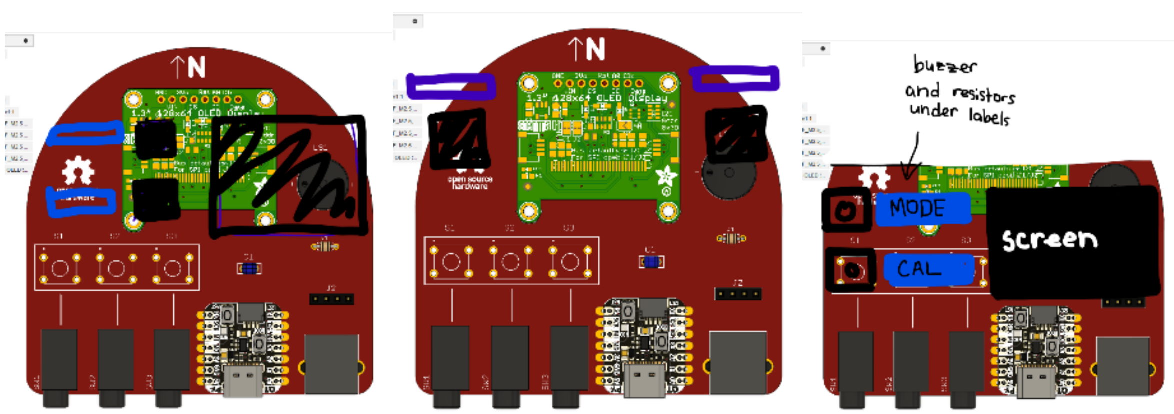
Custom PCB

- Microcontroller
  - Xiao -NRF52840
    - Integrated LED?
- 1 RJ11 jack
- buzzer
- 1 i2c screen
- 3 switch jacks
- 1 tactile button for calibration
- 1 tactile button for slots / mode
- Gamepad LED
- Mouse LED
- Bluetooth LED
- Second RJ11 Jack
- ~~(analog jack for training joystick?)~~
- (additional buttons for cursor speed?)



PCB Concept Review

Could consider PyPortal as screen + i2C option: https://www.adafruit.com/product/4116. This would still require a solution for switch jacks.



Could consider a daughterboard with switchjacks, connected via a SeeSaw board.

Xiao NRF52840 Reset

The Xiao NRF52840 development board does not have a dedicated pin for the reset button, but it does have an exposed pad on the bottom.

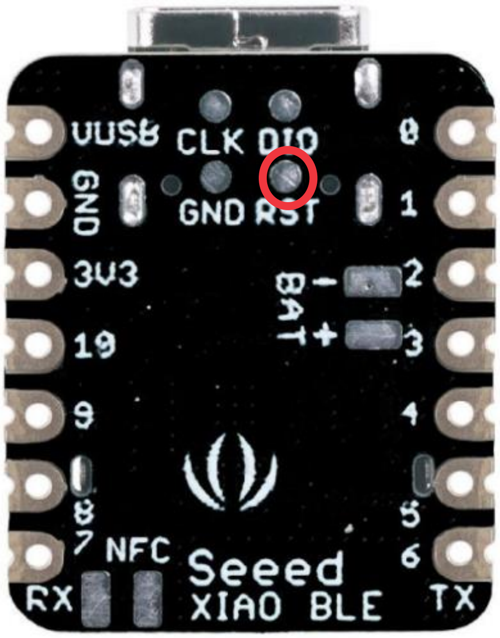


Figure 36. Xiao NRF52840 Reset Pad Location.

As a workaround, a wire could be directly soldered to the RST pad to provide accessible access to the reset pin.

As an alternative, a spring-loaded contact pin could also be added to the PCB so that the reset function could be accessed through a header.

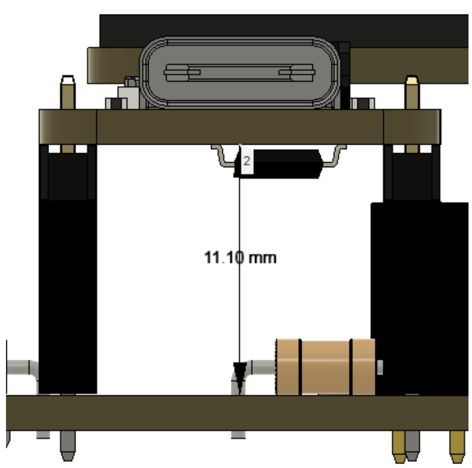


Figure 37. Reset Pad Height

With the current set of headers, the distance from the top of the PCB board to the bottom of the development board is 11.1 mm.

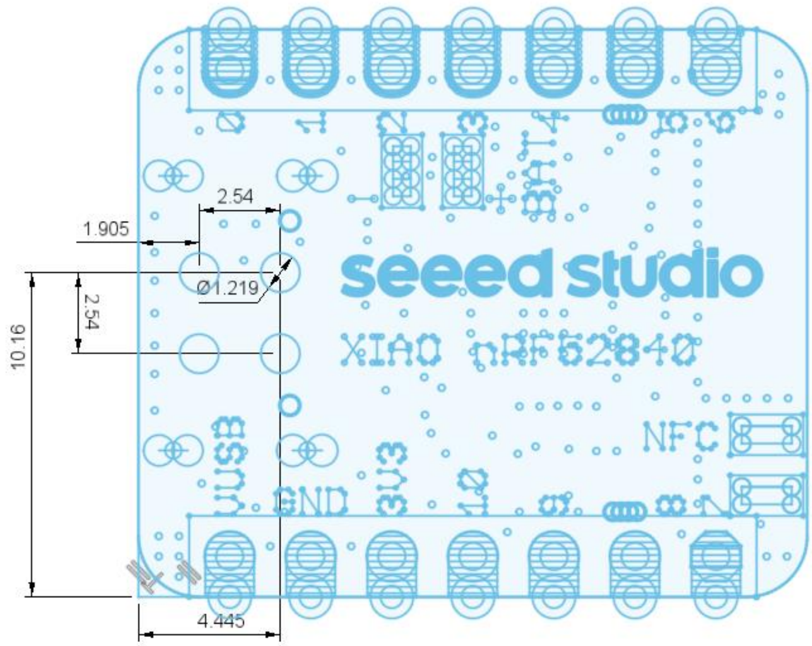


Figure 38. XIAO NRF52840 RST PAD Position.

With the USB port oriented towards the left, the RST pad is located 10.16 mm (0.40 in) up and 4.445 mm (0.175 in) to the right from the bottom left projected corner of the board. The pad is 1.22 mm (0.048 in) in diameter.

Adam Tech PH-MVP-26515-10-10-WP (Digikey 2057-PH-MVP-26515-10-10-WP-ND)was identified as a potential option, with a minimum working height of 10.200 mm, recommended working height of 10.90 mm, maximum working height of 12.70 mm, and plunger diameter of 1.20 mm.

Figure 39. Spring Loaded Pin. Representative image DigiKey. Copyright Digikey.

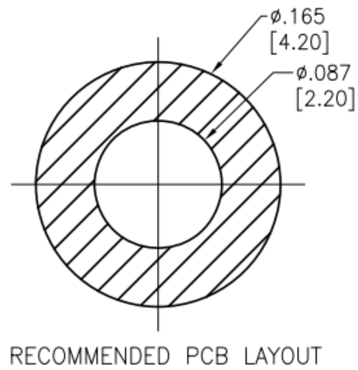


Figure 40. Spring-Loaded Pin Recommended PCB Layout[3]. Copyright Adam Tech.

Ultimately, the use of a spring-loaded pin was abandoned due to the cost of the pin and the difficulty in reliably assembling the pin and having it connect to the microcontroller.

LipSync Hub Enclosure Design

The LipSync Hub Enclosure is intended to the protect the PCB and internal components. It must also have openings for the ports, display, LEDs, and some form of button pushers to press the two buttons t=on the Hub PCB.

Shadow Line

A shadow line is used between the two halves of the enclosure to provide an assembly aid and improve the appearance of the split line.

[3] DIP Spring Loaded Pin, Straight Waterpoof. PN PH-MVP-26515-10/10-WP. https://app.adam-tech.com/products/download/data_sheet/192729/ph-mvp-26515-10-10-wp-data-sheet.pdf

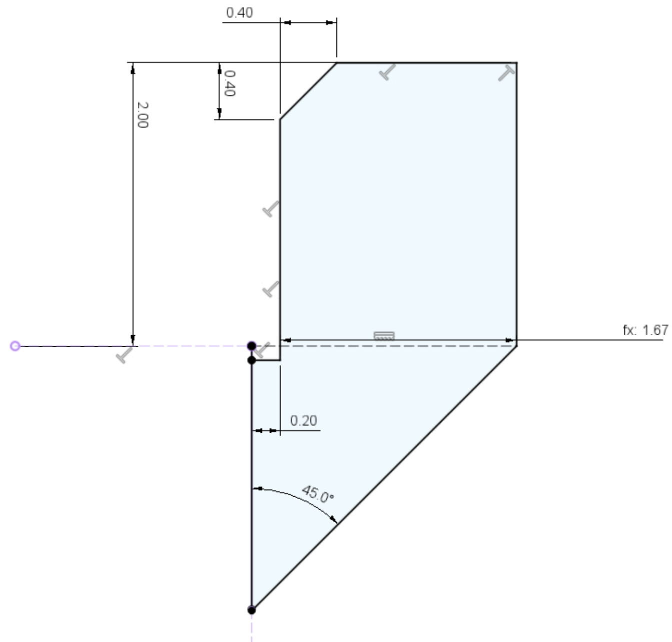


Figure 41: Shadow Line Geometry

Button Pushers

Button Pusher Requirements

The buttons on the PCB are well below the height of the enclosure, so some sort of mechanism is required to activate the buttons.

Button Pusher Options

Options for button pushers could include a separate component that is assembled with the enclosure, a separate component that is captured as a print-in-place with the enclosure, or an integrated compliant component.

Integrated compliant button

- Pros
  - Fewer separate components
  - the compliant element should avoid any rattling
  - no assembly required
- Cons
  - Takes up greater surface area
  - Unclear fatigue life

Captured button elements

- Pros

  - Small footprint
- Cons
  - Tricky to assemble with rotation of pusher relative to button
  - May rattle

Separate Element

- Pros
  - Easier to print
  - Can print in contrasting colours
- Cons
  - Additional components
  - Requires assembly
  - May rattle

Initially, captured button elements were used, but these were changed to separate elements for printability and for contrast / customization options.

Screen Securing Method

A secure method for attaching the screen to the enclosure was required. Several options were considered:

1. Screws
- the screen has 4X 2.5 mm diameter holes that could be used for mounting holes
2. Snap-fit
3. Additional mounting frame

Screws were selected for simplicity.

M2.5 Standards

- Close fit: 2.7 mm
- Normal Fit: 2.9
- Loose Fit: 3.1
- Counterbore Diameter: 5.5 mm
- Counterbore Depth: 3.0 mm

Final Hub Design

The Hub is built around the following primary components:

- Seed Studio XIAO nRF52480 microcontroller
- 3 mono switch jacks
- 1 RJ11 jack
- Adafruit Monochrome 1.3" 128x64 OLED graphic display
- 3 red diffusible LEDs
- Piezo electric speaker
- ¼-20 tee nut

Hub Display

The Hub Display provides a visual interface for the user. The Hub Display is monochrome 1.3" 128 x 64 OLED graphic display – STEMMA QT / Qwiic from Adafruit. This display has an I2C connection.

Figure 42. Hub Display.

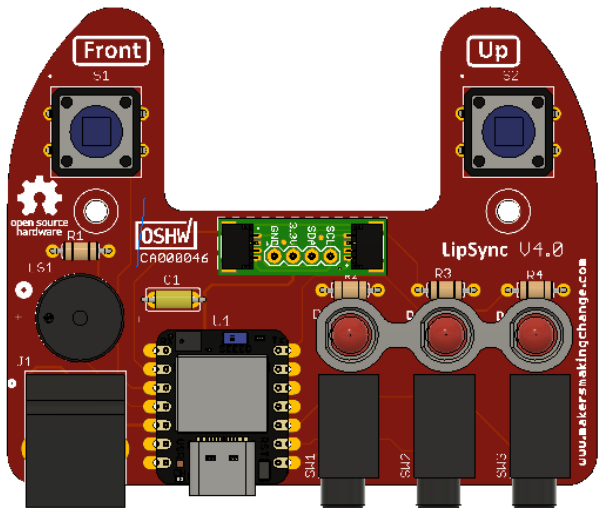
The display has four (4) holes that fit M2.5 mounting screws, used to fasten the display to the enclosure. Partially recessed counterbored holes were added to the Enclosure Top to mount the display, with hex nuts on the rear side.

Key features of the Adafruit Monochrome 1.3" 128x64 OLED graphic display include:

- I2C connection with Sparkfun quiic compatible STEMMA QT connectors
- Power: depends on how much of the display is lit but on average the display uses about 40mA from the 3.3V supply
- OLED lifetime: each pixel has a lifetime of ~1000 hours if kept on before it starts to dim, or a storage lifetime of ~20,000 hours (when not on).

Hub PCB

The Hub PCB is a single, 2-layer custom PCB. All of the components are thru-hole and are soldered to the front of the board.



PCB Key Components

Microcontroller – the microcontroller is the interface between the other hardware and the host device. This microcontroller is the Seeed Studio Xiao nrf52840.

RJ11 Connector - the RJ11 connector provides an I2C connection to the LipSync Joystick.

3.5 mm Switch Jacks – provide a connection for external assistive switches

Piezo Speaker – provides audible feedback

SparkFun QWIIC Adapter – an off-the-shelf board that provides a convenient way to connect a STEMMA QT cable via a set of headers on the PCB. This allows the display to be disconnectable.

LEDs – provide visual feedback

Tactile Buttons – provide a way for support user to provide input and interact with display menu

Microcontroller

The LipSync Hub utilizes a XAIO nRF52840 microcontroller from Seeed Studio. Some key features of this microcontroller are listed in Table 4. And a pinout diagram is shown in Figure 43. A CAD model of this microcontroller created by Maurice Pannard[4] was found and incorporated into the design to check spacing and clearances.

Table 4. Microcontroller Key Features

Item	Seeed Studio XIAO nRF52840
Processor	Nordic nRF52840

[4] https://grabcad.com/library/seeed-studio-xiao-nrf52840-sense-1

	ARM® Cortex®-M4 with FPU runs up to 64 MHz
Wireless	Bluetooth 5.0/NFC
On-chip Memory	1 MB flash and 256 kB RAM
Onboard Memory	2 MB QSPI flash
Interface	1xUART, 1xIIC, 1xSPI, 1xNFC, 1xSWD,
	11xGPIO(PWM), 6xADC
Dimensions	21 x 17.5mm
Power	Circuit operating voltage: 3.3V@200mA
	Charging current: 50mA/100mA
	Input voltage (VIN): 5V
	Standby power consumption: <5μA

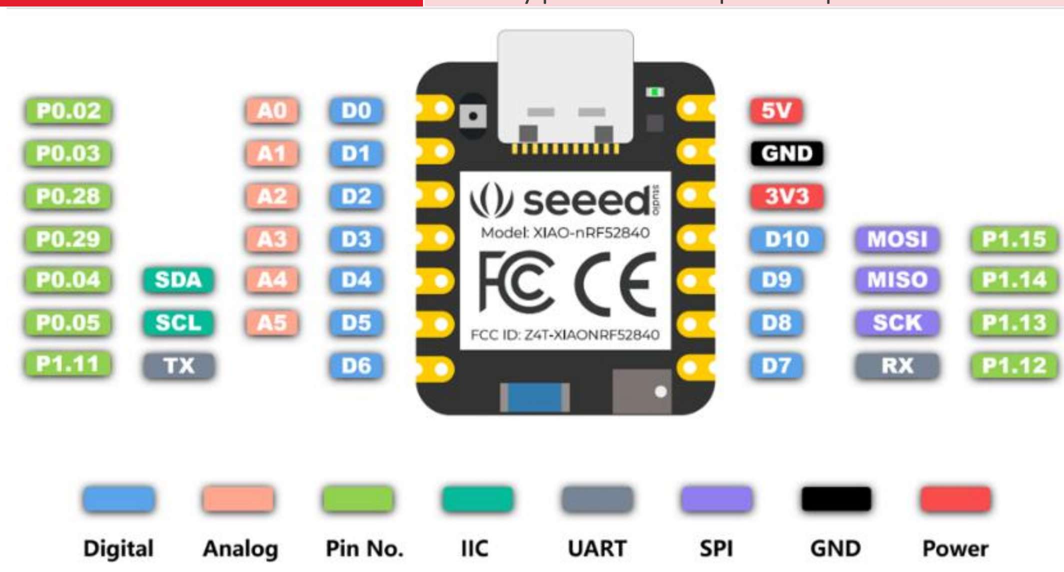


Figure 43. XIAO nrf52840 Pinout. Copyright Seeed Studio. https://wiki.seeedstudio.com/XIAO_BLE/

All of the pins on the microcontroller are utilized, as indicated in Table 5.

Table 5. Microcontroller Pin Utilization

PIN ID	Pin Label	Connection
0	A0	Assistive Switch 1
1	A1	Assistive Switch 2
2	A2	Assistive Switch 3
3	A3	Hub Button 2 (Select)
4	SDA	SDA Serial Data
5	SCL	SCL Or Serial Clock
6	TX	LED 3
7	RX	LED 1
8	SCK	LED 2
9	MI	Hub Button 1 (Next)
10	MO	Buzzer

11	3.3V	Power
12	GND	Ground
13	5V	5 V

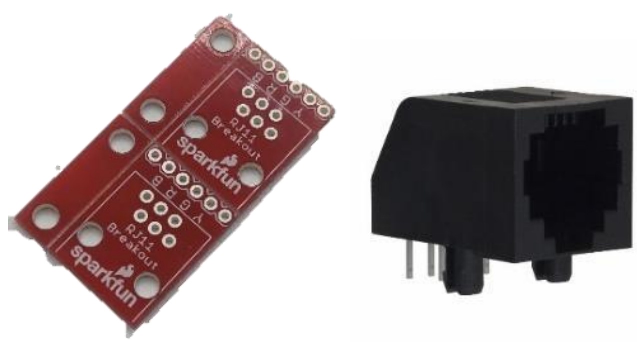
LipSync Joystick Connection – RJ11

The I2C connection takes place using the STEMMA QT cables and breakout boards from Adafruit. To connect the Hub to the Joystick, a RJ11 phone cable is used. The RJ11 is a 6P4C cable, meaning it has 6 positions and 4 are connected.

RJ11 Breakout	Microcontroller Pin	STEMMA QT Cable
2 (B)	12	Black (Gnd)
3 (R)	11	Red (Vcc)
4 (G)	4	Blue (SDA)
5 (Y)	5	Yellow (SCL)

This wiring scheme was chosen so that the order matches the pinout on other I2C adapter boards.

The RJ11 connector for the Hub was chosen to match the footprint of the SparkFun RJ11 Breakout board used in the joystick design so two of the same component can be ordered.



Feedback Speaker

A piezo speaker provides for simple audible feedback. The piezo speaker can generate tones in a range of frequencies. It is currently connected directly to the microcontroller without any additional driving circuitry, so the volume is not adjustable.

Visual Indicators

The LipSync Hub contains three (3) 5 mm red LEDs that are arranged in a horizontal layout that roughly aligns with the three switch input ports.

The LEDs are mounted using a 3D printed spacer to bring the height of the LEDs to be flush with the front face of the enclosure. The spacer has a flat side to provide a visual clue on the correct orientation of the LED during installation.

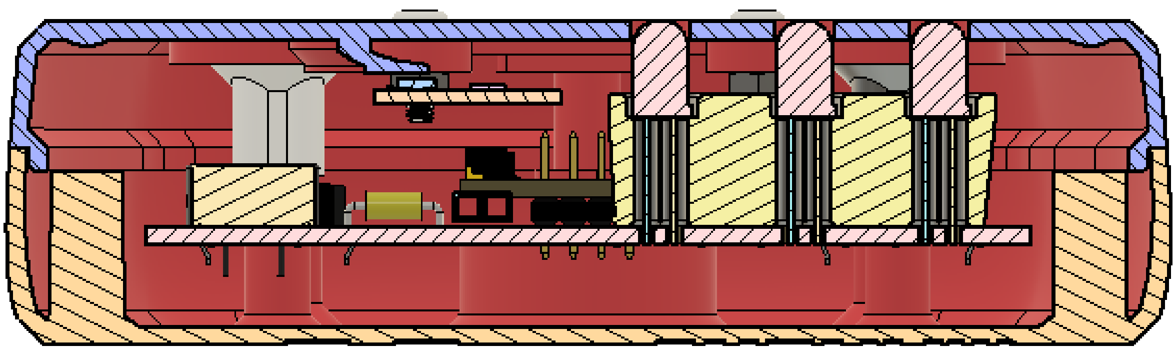


Figure 44. LED Spacer Cross Section

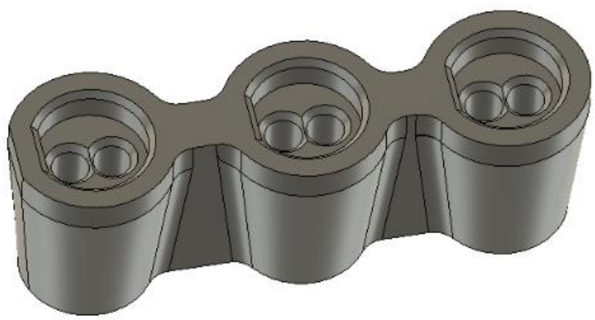


Figure 45. LED Spacer

The LEDs each use a current limiting resistor of 68 ohms.

Tactile Buttons

The LipSync Hub has two 12 mm tactile buttons. These buttons provide a way for a secondary user to provide input to the LipSync and to navigate the Hub Menu.

The tactile buttons are Omron model B3F5050. This model was selected because it has a low activation force and is used in a variety of other MMC OpenAT designs (e.g., Raindrop switch).

LipSync Firmware

Overview

The LipSync Firmware is the code that is flashed to the microcontroller. It handles the inputs from the different sensors, buttons, and assistive switches and generates the corresponding output to the host device as well as user feedback. The code is written in C++ using the Arduino Development Environment.

User Requirements

For the LipSync firmware to be effective, some of the main user requirements are:

1. Emulate the functions of a USB mouse, Bluetooth mouse, and/or gamepad.
2. User must not experience unintended movement of the mouse cursor/gamepad joystick when LipSync joystick is at rest.
3. A user must be able to perform inputs using either sip and puff, or assistive switches.
4. A user must be able to independently access and change key settings that are vital to the operation of the LipSync.
5. The user must be able to independently switch operating/communication mode (i.e., USB mouse, Bluetooth mouse, USB Gamepad).
6. The user must be able to easily identify what communication mode the LipSync is using.
7. The primary user must be able to identify if the LipSync is powered on.
8. A secondary user must be able to easily change settings on the LipSync without needing to use sip and puff inputs or assistive switch inputs.
9. User must be able to change settings once and not have to adjust each time the LipSync is powered off and on.

Functional Requirements

To meet the user requirements defined above, technical requirements need to be defined.

1. The LipSync must be capable of emulating a computer mouse.
   1.1. The LipSync must be capable of emulating a USB computer mouse.
   1.2. The LipSync must be capable of emulating a Bluetooth Mouse.
   1.3. The LipSync must be able to emulate a right click.
   1.4. The LipSync must be able to emulate a left click.
   1.5. The LipSync must be able to emulate a middle click.
   1.6. The LipSync must be able to emulate a press and hold of left click, or “drag”.
   1.7. The LipSync must be able to emulate a mouse scroll wheel input.
2. The LipSync must be capable of emulating a gamepad.
   2.1. The LipSync must be capable of emulating a joystick.
   2.2. The LipSync must be capable of emulating two gamepad buttons.
3. The LipSync must be capable of accurately detecting inputs.
   3.1. The LipSync must record readings of the joystick movement.

3.2. The LipSync must be able to differentiate between an intentional movement of the joystick and when the joystick is at rest.
3.3. The LipSync must not produce cursor or gamepad joystick drift.
3.4. The LipSync must record readings of pressure input.
3.5. The LipSync must be able to differentiate between an intentional sip and puff and differences in ambient pressure .
3.6. The LipSync must read inputs from assistive switches and hub buttons.
4. The LipSync must provide the user with useful feedback.
4.1. The LipSync must inform the user when the LipSync is initialized and ready to use.
4.2. The LipSync must inform the user of which operating mode they are in.
4.3. The LipSync must inform the user when the outputs are different than typical operation, such as when left click is being held down.
4.4. The LipSync must inform the user when they are adjusting settings.
4.5. The LipSync must provide feedback to guide the user through calibration processes.
5. The LipSync must allow the user to adjust key settings.
5.1. The LipSync must allow the user to independently switch operating/communication modes (i.e. USB mouse, Bluetooth mouse, USB Gamepad).
5.2. The LipSync must allow the user to independently adjust the mouse cursor speed.
5.3. The LipSync must allow the user to independently initiate calibration.
5.4. The LipSync must be able to calibrate the neutral position of the joystick.
5.5. The LipSync must be able to calibrate the extent or maximum positions of the joystick, to allow for the maximum input to equal the maximum output.
5.6. The LipSync must allow for settings to be changed via serial through the API.
6. The LipSync must be able to conduct timed functions accurately.
6.1. The LipSync timers must allow timed processes to be completed without blocking the codes other functions.
6.2. The LipSync timers must allow functions to be conducted at a defined frequency.
7. The LipSync must use non volatile memory to save values after powering off.
7.1. The LipSync must save to non volatile memory any user adjustable settings.
7.2. The LipSync must load values from memory on startup.
7.3. The LipSync must be able to reset the memory to defaults.

The main functions of the LipSync are grouped into the following categories below, and then explored in further detail:

- Reading inputs
    - Joystick movement
    - Sip and puff (Pressure)
    - Assistive switches
    - Buttons

- Processing inputs and generating actions
    - Input output mapping
- Process outputs
    - Mouse
        - Cursor movement
        - Left click, middle click, right click
        - Drag mode
        - Scroll mode
    - Gamepad
        - Joystick displacement
        - Gamepad buttons
- Communicate with host device
    - USB Mouse
    - USB Gamepad
    - Bluetooth Mouse
- Adjust settings
    - Mode (communication/operating mode, i.e. USB mouse, Bluetooth mouse, USB Gamepad)
    - Cursor speed
    - Calibration (center reset and full extents calibration)
    - API
- Provide user feedback
    - LEDs
    - Screen
    - Sound
- Timers
- Store values in memory

Firmware Overview

The firmware for the microcontroller is written in C++ using the Arduino Development Environment.

The firmware consists of a main Arduino sketch (LipSync_Firmware.ino) and several additional files. The files are described in Table 6.

Table 6. Firmware File Summary

File Name	Description
LipSync_Firmware.ino	Main Arduino sketch
LSAPI.ino	Application Programming Interface (API) functions and implementation

LSBLE.h	Definition and implementation of LSBLEMouse class. This class is used to provide a BLE HID mouse using the Bluefruit library.
LSBuzzer.h	Definition and implementation of the LSBuzzer class. This is used to handle commands to the buzzer to provide sound feedback.
LSCircularBuffer.h	Implementation of the LSCircularBuffer class. This is used to store and manage multiple elements of data.
LSConfig.h	This file contains the settings and parameters for the LipSync.
LSInput.h	Definition and implementation of the LSInput class, used to measure and store the current and previous state of buttons and switches.
LSJoystick.h	Definition and implementation of LSJoystick class. This class handles low level communication with magnet sensor, manages the calibration of the sensor, implements a deadzone, and scales the data for mouse output or gamepad output, depending on operating mode.
LSMemory.h	Definition and implementation of the LSMemory class. This implementation uses the Adafruit_LittleFS librarr, InternalFilesSystem and ArduinoJson to store and retrieve data to flash to provide non-volatile storage of settings.
LSOutput.h	Definition and implementation of LSOutput class, used to control LED output.
LSPressure.h	Definition and implementation of the LSPressure class, used to provide sip and puff functionality and manage communication with the pressure sensors.
LSScreen.h	Definition and implementation of the LSScreen class. This is used to handle commands to the OLED screen and manage the menu. This class handles the menu actions for next and select and calls on other functions to change settings.
LSTimer.h	Definition and implementation of the LSTimer class. This class manages multiple timed tasks and provide basic callback functionality.
LSUSB.h	Definition and implementation of LSUSBMouse, LSUSBKeyboard, and LSUSBGamepad classes. These classes implement a mouse, keyboard, and gamepad using the Adafruit Tiny USB library.
LSUtils.h	Definition of a number of structures used across classes.
LSTest.ino	Defines several tests that can be activated to test device functionality.
LSWatchdog.h	Defines several functions for implementing a hardware watchdog.

Basic Connections

Description	PCB	Microcontroller	Software	Pin ID
Onboard LED				
User LED - Red	-			LED_RED
User LED - Blue	-			LED_BLUE
User LED - Green	-			LED_GREEN
Inputs				
Next Button	S1	MO	D9	9
Select Button	S2	A3	D3	3

Description	PCB	Microcontroller	Software	Pin ID
Switch Jack S1	SW1	A0	A0	0
Switch Jack S2	SW2	A1	A1	1
Switch Jack S3	SW3	A2	A2	2
Outputs				
Left LED	D2	MI	D8	8
Center LED	D1	RX	D7	7
Right LED	D3	TX	D6	6
Buzzer	LS1	MO	D10	10
Communication				
I2C-SDA	SDA	SDA	P0.04/SDA/A4/D4	4
I2C-SCL	SCL	SCL	P0.05/SCL/A5/D5	5

Table 7. I2C Addresses

Device	Model	I2C Address	Note
Magnetometer	TLV493D	0x5E	
Sip Puff Pressure Sensor	LPS33HW	0x5D	
Ambient Pressure Sensor	LPS22	0x5C	This is the secondary I2C addressed enabled by soldering the pads on the pressure sensor.

User LED Control

The Xiao nRF52840 User is an RGB LED controlled by three different channels. It is a common anode, so control is the reverse of what is expected (e.g., LED turn on when set LOW, and turns off when set HIGH).

Third Party Libraries

A number of third party libraries are utilized in the LipSync firmware.

Table 8. Third Party Libraries

File	Name	Author	License	Description	Dependency
	Adafruit_BusIO	Adafruit	MIT	I2C helper;	nil
	Adafruit_GFX	Adafruit	BSD	Display Graphics	Adafruit_BusIO
	Adafruit_LPS2X	Adafruit	BSD	LPS22 Ambient Pressure Sensor	Adafruit_Sensor, Adafruit_BusIO

File	Name	Author	License	Description	Dependency
	Adafruit_LPS33HW	Adafruit	BSD	SaP Pressure Sensor	Adafruit_BusIO
	Adafruit_Sensor	Adafruit	Apache 2.0	Unified sensor abstraction;	nil
	Adafruit_SSD1306	Adafruit	BSD	Monochrome OLED Driver	Adafruit_GFX
	ArduinoJson	Benoit Blanchon	MIT	Efficient JSON library	
	TLV493D-A1B6	Infineon Technologies	Custom	Magnetic sensor	nil

Basic Functions

The Basic Functions the LipSync needs to do, and which input needs to be able to do these functions is explored in the table below:

Function	Joystick (sip/puff, switch)	Hub buttons	API	Menu	Modify Code
Turn on/off Bluetooth Mode			X	X	
Change Operating Mode (e.g., Mouse, Gaming, Wireless)			X	X	
Initiate joystick neutral calibration[1]	X* (switch 2, or through menu)		X	X	
Change pressure threshold			X	X	
Initiate joystick extent calibration			X	X	
Change joystick sensitivity			X	X	
Adjust cursor speed			X	X	
Adjust scroll speed			X	X	
LED On / off			X	X	
Modify LED Brightness?			Future	X	
Audible Feedback on/off			X	X	
Change sound feedback mode			X	X	
Initiate hard factory reset			X	X	
Initiate reset (power cycle)			X	X	
Control Cursor	X		X		
Input click functions	X	X	X		
Swap sip and puff			Future	Future	X
Enable debug mode			X	Future	X

Function	Joystick (sip/puff, switch)	Hub buttons	API	Menu	Modify Code
Enable/Disable mouse output					X
Set up / enable keyboard macro			Future		Future
Trigger keyboard macro	Future				
Enable morse code input			Future	Future	
Set custom keyboard macro sequence			Future		
Activate alternate sip[2]		X			
Activate alternate puff[2]		X			
Change Slot	Future			Future	
Turn on/off screen / enter/exit menu	X	X	X	X	
Select Menu Item	X (Sip/Puff)	X	X	n/a	
Next Menu Item	X (Sip/Puff)	X	X	n/a	

Future features to consider:

- Wakeup / go to sleep / power management
- Bluetooth:
  - Initiate scanning
  - Swap connection
- Baseline pressure calibration??
- Button remapping

Reading Inputs

For all inputs, the values are read and a defined length of buffer values are stored. This section goes over the reading and storing of these values.

Joystick Movement

To measure the movement of the joystick, using the magnetic sensor, the library TLV493D-A1B6 by Infineon Technologies was used ([https://github.com/Infineon/TLV493D-A1B6-3DMagnetic-Sensor)](https://github.com/Infineon/TLV493D-A1B6-3DMagnetic-Sensor)

Some functions from the library:

- getX() – strength of magnetic field in cartesian x direction [mT (milliTeslas)]
- getY() – strength of magnetic field in cartesian y direction [mT(milliTeslas)]
- getZ() – strength of magnetic field in cartesian z direction [mT(milliTeslas)]
- getAmount() – overall strength of magnetic field in mT; sqrt(x^2+y^2+z^2) [mT (milliTeslas)]
- getAzimuth() – arctan(y/x)

- getPolar() – arctan(z//(sqrt(x^2+y^2))

Coordinate Systems

For intuitive use, the movement of the joystick must correspond with the movement of the cursor on the screen in mouse mode or the expected movement of the joystick in gamepad mode. To work as expected, the mouthpiece movement and sensor coordinate system must be set up properly to match the coordinate system on the host device screen.

Joystick Sensor Coordinate System

The TLV493D 3D magnetic sensor measures the magnetic field in x, y, and z-directions as shown in Figure 46. The magnetic field for the pill-shaped is primarily oriented along the main axis.

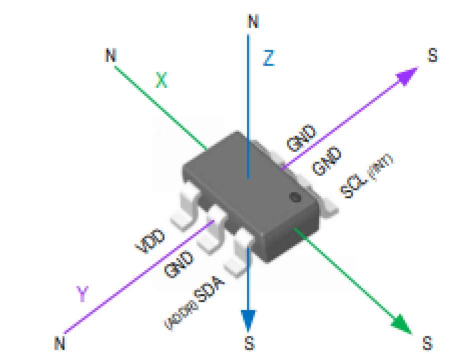


Figure 46. TLV493D-A1B6 Coordinate System

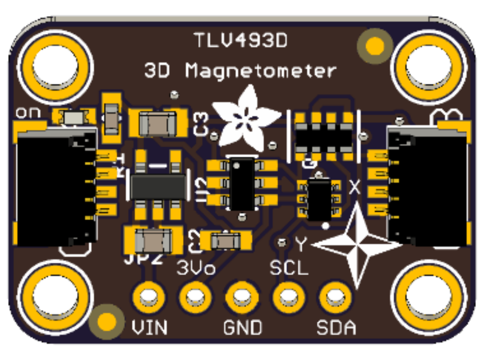


Figure 47. Adafruit TLV493D Board Coordinate System.

LipSync Coordinate System

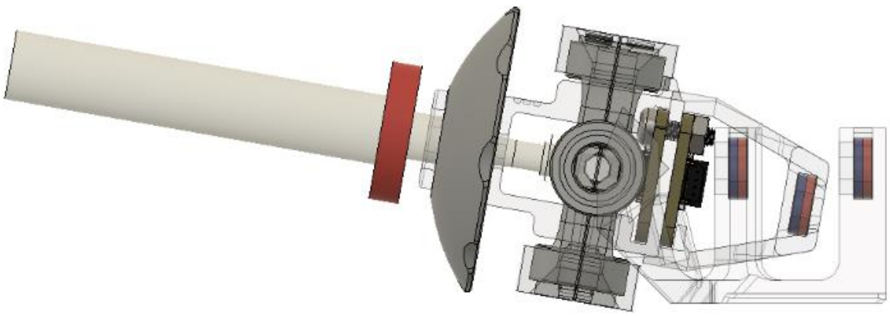


Figure 48. LipSync Up Movement

As the mouthpiece is moved up, the sensor rotates down relative to the fixed sensing magnet. With respect to the sensor, the magnet moves up.

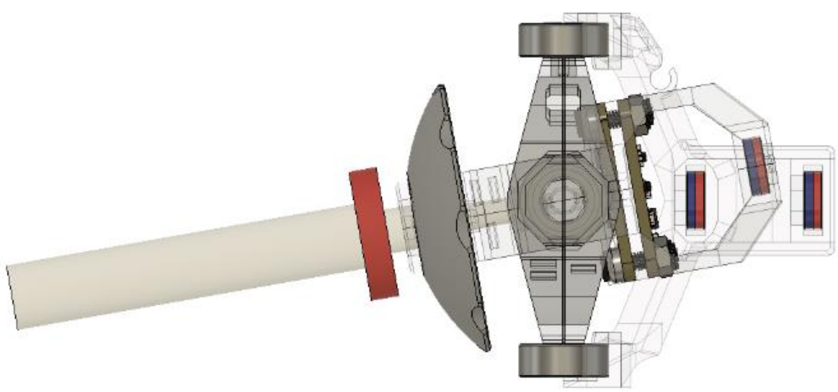


Figure 49. LipSync Right Movement.

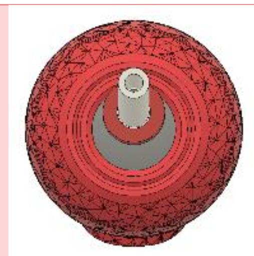
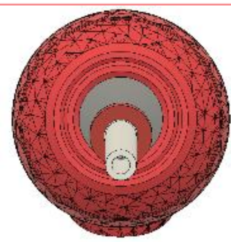
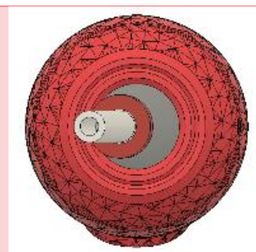
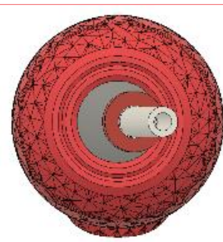
As the mouthpiece is moved right, the sensor rotates left relative to the fixed sensing magnet. With respect to the sensor, the magnet moves left.

Magnet Orientation

The value that the sensor reports depends on the orientation of the magnet. The firmware detects the orientation of the magnet by looking at the sign of the z-component and inverts the reading as needed..

Table 9. Mouthpiece Movement Mapping

Mouthpiece Movement	Mouthpiece	Sensor X	Sensor Y

Up		Pos	0
Down		Neg	0
Left		0	Pos
Right		0	Neg

Joystick Stored Values

Joystick values are read, stored, the input is processed and then the output is processed. These are defined in the LSJoystick class as _rawPoint (float), _inputPoint (int), and _outputPoint (int), and stored in a buffer (10 values are stored for raw values, 5 values are stored for input and output points) defined as joystickRawBuffer, joystickInputBuffer, and joystickOutputBuffer, respectively.

Recording Raw Joystick Input

The raw joystick input values (_rawPoint) are the values read from the magnetic sensor. These values are dependent on the hardware and how far the magnets are from the magnetic sensor. With this setup, and the expected values, the code caps out the at max values to +/- 30 for both x and y measurements (JOY_RAW_XY_MAX). Through experimentation these max values are typically in the range of +/- 13 and +/- 26. When at rest the joystick should read close to 0,0. These readings are stored as a float data point.

Processing Joystick Input Values

The raw joystick values (_rawPoint) read from the sensor are then processed using the function processInputReading to give _inputPoint, stored in joystickInputBuffer. This function centers the values (using the center point calibration stored or measured on startup) and maps the raw values (float) to be an integer between +/- 1024. The significance of this number is it is the max resolution for 10 bit analog to digital converter, which is used for Arduino analog readings, and therefore a common range for joystick readings.

Mouse Cursor Mapping

Host Device Coordinate System

The default coordinate system for screens has the origin in the top left corner, with positive X oriented towards the right and positive Y oriented down. This coordinate system is illustrated in Figure 50.

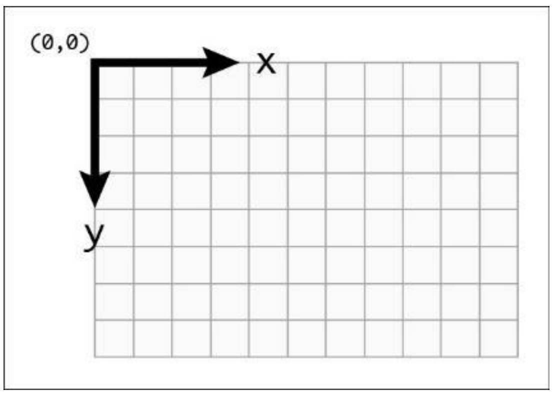


Figure 50. Host Device Screen Coordinate System

Sip and Puff Inputs

Sip and Puff inputs are determined by measuring the pressure applied to the mouthpiece. Two separate pressure sensors are used. One pressure sensor measures the absolute pressure applied to the mouthpiece. A second pressure sensor measures the ambient pressure inside the LipSync enclosure. The sip and puff pressure is measured as the pressure of the tube relative to ambient, or the difference between the tube pressure sensor and the ambient pressure sensor.

Having two sensors prevents false positives of sip and puff inputs from changes in pressure in the user’s environment.

Sip and puff inputs are activated when the sip and puff pressure reaches the sip pressure threshold or puff pressure threshold. By default, these thresholds are both set at 3.0 hPa. Different actions are triggered based on the duration of time beyond certain thresholds.

The code for measuring the pressure is in the LSPressure class. The main tubing sensor uses the Adafruit LPS33H library by Adafruit https://github.com/adafruit/Adafruit_LPS35HW. The ambient pressure sensor used the Adafruit LPS2X library by Adafruit https://github.com/adafruit/Adafruit_LPS2X.

Each time the sip and puff state gets recorded (using the inputStateStruct) it gets recorded into a buffer, and the oldest value gets forgotten.

Assistive Switch Inputs and Hub Button Inputs

Assistive switches and Hub buttons are both handled in a similar way, since they are both digital readings of a pin. In the main LipSyncFirmware.ino file, an instance of the class LSInput is defined for both the assistive switches and the Hub buttons (is and ib respectively). This class instance uses the array of button/switch pins and the total number of buttons/switches.

In the LSInput class, the digital reading is taken of each button or switch and then three values are recorded to define the state of all of the buttons or all of the switches. This is captured using the inputStateStruct(mainState, secondaryState, elapsedTime).

When a new state is recorded, it gets saved to the buffer, and the oldest state gets forgotten. In this case the input buffer is 5 states long.

Storing Input States

To store input states for sip and puff actions, assistive switches, and buttons, inputStateStruct is used. The readings for the inputs are done in the classes LSInput and LSPressure, and are recorded using this structure.

The input state structure is made up of:

- mainState
    - This is an integer binary representation of one or more input states. This single value represents which inputs were involved in the recorded state. ([button1 + 2*button2 + 4*button3 or none : 0 ,sip : 1, puff : 2]
- secondaryState [waiting = 0, started = 1, released = 2]
    - Waiting – same state as previously (e.g. both on-on and off-off)
    - Started – has transitioned from off to on
    - Released – has transitioned from on to off
- elapsedTime [ms]
    - Time since last state was recorded

Processing inputs and generating actions

Once the inputs have been read, these inputs must be translated to output actions. The input output mapping is represented in the following table:

Table 10. Input - Output Actions

Input	Mouse	Gamepad	Menu/Settings Mode
Joystick			
Joystick Movement	Mouse movement	Joystick movement	--
Sip and Puff			
Short Puff	Left click	Button 1 press	Next Menu item
Long Puff	Start drag mode	Button 3 press	
Very Long Puff	Enter settings mode	Enter settings mode	Exit settings mode
Short Sip	Right Click	Button 2 press	Select menu item
Long Sip	Start scroll mode	Button 4 press	
Very Long Sip	Middle click <slot change when implemented>	--	
Hub Buttons			
Next Button	Left click	Button 1 press	Next menu item
Next Button Long Press	Start drag mode	Button 3 press	
Next Very Long Press	Enter settings mode	Enter settings mode	Exit settings mode
Select Button	Right Click	Button 2 press	Select menu item
Select Button Long Press	Start scroll mode	Button 4 press	
Select Very Long Press	Middle click	--	

	<slot change when implemented>		
Next + Select Button together	Enter settings mode	Enter settings mode	Exit settings mode
Assistive Switches			
SW1 Press	Left click	Button 1 press	Next menu item
SW1 Long Press	Start drag mode	Button 3 press	
SW1 Very Long Press	Enter settings mode	Enter settings mode	Exit settings mode
SW2 Press	Middle click	Button 5 press	
SW2 Long Press	-- <slot change when implemented?>	Button 6 press	
SW2 Very Long Press	Start Center Reset	Start Center Reset	
SW3 Press	Right Click	Button 2 press	Select menu item
SW3 Long Press	Start scroll mode	Button 4 press	
SW3 Very Long Press	Middle click	<slot change when implemented>	
SW1 + SW3 Press	Enter settings mode	Enter settings mode	Exit settings mode

Summary:

Table 11. Input-Output Action Summary

Interchangeable LipSync Inputs			**Duration**	**Hub Menu**	**Mouse**	**Gamepad**
Sip and Puff	**Assistive Switches**	**Hub Buttons**				
Puff	**S1**	**Select**	< 1 sec	Select	Left click	Button 1
			1-3 sec	-	Start Drag Mode	Button 3
			> 3 sec	Exit Hub Menu	Enter Hub Menu	Enter Hub Menu
-	**S2**	**-**	< 1 sec	-	Middle Click	Button 5
			1-3 sec	-	N/A	Button 6
			> 3 sec	-	Perform Center Reset	Perform Center Reset
Sip	**S3**	**Next**	< 1 sec	Next	Right click	Button 2
			1-3 sec	-	Start Scroll Mode	Button 4
			> 3 sec	-	Middle Click	Button 5
-	**S1 + S3**	**Next + Select**	< 1 sec	Exit Hub Menu	Enter Hub Menu	Enter Hub Menu

Joystick	-	-	Cursor movement	Joystick movement

The mapping of the inputs to outputs in the code is found in the LSConfig.h file, and then this array is used by the code to check which output should happen when a certain input occurs for a certain duration. This is an example of the input output array for the assistive switches:

```
// External Assistive Switch Jacks
const inputActionStruct switchActionProperty[]{
  { INPUT_MAIN_STATE_NONE,            CONF_ACTION_NOTHING,       CONF_ACTION_NOTHING,        CONF_ACTION_NOTHING,           0,    0 },

  { INPUT_MAIN_STATE_S1_PRESSED,      CONF_ACTION_LEFT_CLICK,    CONF_ACTION_B1_PRESS,       CONF_ACTION_SELECT_MENU_ITEM,    0,  1000 },
  { INPUT_MAIN_STATE_S1_PRESSED,      CONF_ACTION_DRAG,          CONF_ACTION_B3_PRESS,       CONF_ACTION_NOTHING,        1000,  3000 },
  { INPUT_MAIN_STATE_S1_PRESSED,      CONF_ACTION_START_MENU,    CONF_ACTION_START_MENU,     CONF_ACTION_STOP_MENU,      3000, 5000 },

  { INPUT_MAIN_STATE_S2_PRESSED,      CONF_ACTION_MIDDLE_CLICK, CONF_ACTION_B5_PRESS,        CONF_ACTION_NOTHING,           0,  1000 },
  { INPUT_MAIN_STATE_S2_PRESSED,      CONF_ACTION_NOTHING,       CONF_ACTION_B6_PRESS,       CONF_ACTION_NOTHING,        1000,  3000 },
  { INPUT_MAIN_STATE_S2_PRESSED,      CONF_ACTION_CURSOR_CENTER, CONF_ACTION_CURSOR_CENTER, CONF_ACTION_NOTHING,        3000,  5000 },

  { INPUT_MAIN_STATE_S3_PRESSED,      CONF_ACTION_RIGHT_CLICK,   CONF_ACTION_B2_PRESS,       CONF_ACTION_NEXT_MENU_ITEM,  0,  1000 },
  { INPUT_MAIN_STATE_S3_PRESSED,      CONF_ACTION_SCROLL,        CONF_ACTION_B4_PRESS,       CONF_ACTION_NOTHING,        1000,  3000 },
  { INPUT_MAIN_STATE_S3_PRESSED,      CONF_ACTION_MIDDLE_CLICK, CONF_ACTION_NOTHING,         CONF_ACTION_NOTHING,        3000, 5000 },

  { INPUT_MAIN_STATE_S13_PRESSED,      CONF_ACTION_START_MENU,  CONF_ACTION_START_MENU,      CONF_ACTION_STOP_MENU,         0, 1000 },
};
```

Figure 51. Assistive Switch Action Mapping

Processing Outputs

Joystick Movement Output

As covered above, joystick readings are stored in buffers joystickRawBuffer, joystickInputBuffer, and joystickOutputBuffer. The process to record and process values in joystickRawBuffer and joystickInputBuffer are covered above in the Reading Inputs section.

Values stored in joystickInputBuffer will be mapped between -1024 and 1024, and then each _inputPoint will be processed using processOutputResponse to give the _outputPoint and stored in joystickOutputBuffer.

processOutputResponse takes the _inputPoint and applies the deadzone to prevent drifting, and maps the input values to output values. The deadzone is applied using applyDeadzone, where any input value below the deadzone value gets assigned an output value of 0,0. Then the deadzonedPoint gets mapped to output values using linearizeOutput, which maps each point to an output value, using the variable “_rangeValue” as the max output. _rangeValue is calculated using the function setOutputRange, depending on whether you are in Mouse Mode or Gamepad Mode.

In setOutputRange, if the mode is set to mouse mode, _rangeValue is calculated using the cursor speed level (function variable rangeLevel) using this equation:

_rangeValue = (int)((0.125 * sq(rangeLevel)) + (0.3 * rangeLevel) + 2); Equation 1

Depending on the speed level, _rangeValue will end up being between 2 and 17.5.

In setOutputRange, if the mode is set to gamepad mode, _rangeValue is set to be 127 (JOY_OUTPUT_XY_MAX_GAMEPAD)

Mouse Output

Cursor movement

The cursor movement value is calculated and stored in the LSJoystick class, and stored in the joystickOutputBuffer. These values are calculated as explained in Joystick Movement Output. The value sent to the mouse move command will depend on the amount the joystick is deflected, and the cursor speed level. Depending on the cursor speed level, the expected maximum magnitude of the output values sent to the mouse move command are between 2 and 17.5. At rest the output cursor movement would be 0,0. Movements down or to the left will result in negative values.

Mouse clicks

When a mouse click action is conducted, the button will be clicked, or pressed and then immediately released. This will occur when the action mapped to the click is completed, so for example when switch 1 is pressed for less than 1 second and then released, once it is released, the left click action will be sent, which will cause it to be pressed and immediately released.

Left click and right click are used more and are therefore mapped to more easily activated inputs, while middle click can also be conducted but requires a longer activation. See the Processing Inputs and Generating Actions section to see which input actions are mapped to these outputs.

Drag Mode

Drag mode is where left click is pressed and held, instead of being released. A defined input will toggle drag mode to be on, and then any sip and puff, button, or switch input while in drag mode will turn it off.

Scroll Mode

When scroll mode is activated, joystick movements will be translated to scrolling the mouse scroll wheel up or down, instead of cursor movement. A defined input will toggle scroll mode to be on, and then any sip and puff, button, or switch input while in drag mode will turn it off.

The speed of the scroll is defined by the scroll speed level (variable scrollLevel), which can be adjusted in the API.

Gamepad Output

Joystick displacement

The cursor movement value is calculated and stored in the LSJoystick class, and stored in the joystickOutputBuffer. These values are calculated as explained in Joystick Movement Output. The expected output values sent to the mouse move command are between -127 and 127.

Gamepad buttons

On an HID Gamepad, there are 8 buttons that can be pressed, the LipSync currently has inputs set to activate 6 of these buttons. Currently, when the associated input is released, the Gamepad button will be pressed and then immediately released. For example, when a short puff is completed, Gamepad button 1 is pressed and then immediately released.

Input Modes / Activation Mode

Previous LipSync gamepad code had two modes called “digital” and “analog”. This is not in scope for this version of the firmware, but is explained below for future work:

- “Digital”
  - Length of input maps to different buttons and functions (e.g., short puff is Button 0, long puff is Button 3).
  - Button is pressed on release of input.
  - Button is released after fixed duration timer.
  - Long press on input could also toggle Gamepad button (still need some way to access menu)
- “Analog”
  - Gamepad button is pressed on initial press of input
  - Gamepad button is released on release of input
  - Gamepad button could be toggled on after a certain duration of input

User Feedback

There are a number of items that the user requires feedback on:

1. User needs to be able to identify when device is powered on.
2. User needs to be able to turn Bluetooth mode on / off
3. User needs to be able to determine if the device is paired.
4. User need to be able to determine if the device is connected.
5. (Handling multiple devices – e.g. Custom names)
6. User needs a way to identify when device is in scroll mode.
7. User needs a way to identify when device is in drag mode.

Visual Feedback – LEDs

There is a total of four LEDs on the LipSync Hub. There are three discrete red LEDs, and a single RGB LED on the microcontroller.

Several options were considered for how to use the three discrete LEDs. The following options were considered:

1. Operating mode – one light per operating mode (USB Mouse, USB Gamepad, Wireless Mouse)
2. Input Feedback – one light per input channel (e.g. left click, middle click, right click)
3. Input Timing Feedback – one light per input timing level (e.g., short, long, very long)
4. Operating slot – one light per slot

The initial release will use the three LEDs to provide input and timing feedback.

Early iterations of the LipSync hardware utilized Neopixels. The IBM Colorblind Friendly Color Palette[5] was chosen to represent a discrete set of colors that could be easily distinguished amongst by users. The RGB values of these colors are listed in Table 12. This set of colors is still used for the RGB microcontroller LED.

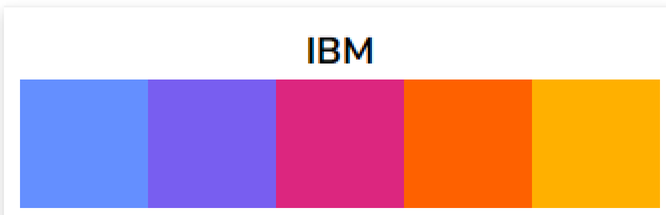


Figure 52. IBM Colorblind Friendly Color Palette

Table 12. RGB Color codes

Color	Red	Green	Blue
White	255	255	255
Blue	100	143	255
Purple	120	94	240
Red	220	38	127
Orange	254	97	0
Yellow	255	176	0
Black	0	0	0

User Feedback Summary

The current set of user feedback is outlined in Table 13. This table lists the discrete visual (i.e. LED) feedback, display feedback, and audio feedback for a number of situations and states.

[5] https://davidmathlogic.com/colorblind

Table 13. Feedback Summary.

Item	Discrete Visual	Display	Audio
Power	Steady LED on Enclosure		Power-up sound
Initialization	3 LEDs long flash	"LipSync v4.0" "Makers Making Change"	
Ready for use	3 LEDs short flash	"Ready to use" "Mode: __"	Ready sound
Bluetooth - Pairing	Dev board user LED		
Bluetooth - Connected	Dev board user LED		
Mouse Mode	Dev board user LED	Display on startup	
Gamepad Mode	Dev board user LED	Display on startup	
Scroll Mode	Right LED stays on		
Drag Mode	Left LED stays on		
Calibration Mode (Don't touch)	Dev board user LED? Or slow blink both mode LEDs at once?	"Center reset, do not move joystick"	
Calibration Measurement			Corner Tone
Center Measurement			Center Tone
Menu / Settings Mode		Screen on	
Sip/puff/switch has been activated	Left / Right LED		
Sip/puff/switch long	Left / Right LED Slow Blink		
Sip/puff/switch very long	Left / Right LED Double Blink		
Device Resetting			Device reset sound
Hardware Error			Error sound

Indicating Power

A light on the joystick enclosure indicates that the joystick is powered and the up direction of the joystick. This light is a light pipe that transmits the light from the LED on the ambient pressure sensor.

Operating Mode

The operating mode is indicated by the color of the microcontroller LED on the LipSync Hub. It is also indicated on startup using the visual display.

Light Brightness

The brightness of the LEDs are user adjustable through the Menu and the Serial API Interface. The following equation maps the 11 light brightness levels to the LED brightness level:

```
ledBrightness = round(pow((float(_ledBrightnessLevel)/CONF_LED_BRIGHTNESS_LEVEL_MAX), 2.2) * CONF_LED_BRIGHTNESS_MAX);
```

The LED brightness is controlled using analog write which uses pulse width modulation that varies from 0 (always off) to 255 (always on)

```
analogWrite(CONF_LED_LEFT_PIN, ledBrightness);
```

Table 14: Light Brightness Mapping

Light Brightness Level	LED Brightness
0	0
1	2
2	7
3	18
4	34
5	55
6	83
7	116
8	156
9	202
10	255

Sounds

The LipSync Hub has a speaker that plays different tones to provide user feedback. The user can choose to turn sounds off by changing the sound mode. It is not currently possible to adjust the volume.

Sound	Description	Notes
Startup Sound	Plays when the LipSync is first powered.	F5 for 0.2 s
Ready Sound	Plays when the LipSync has finished the center reset and is ready for use.	F5 for 0.5 s, delay 0.5 s, C6 for 0.25 s
Error Sound	Plays when a hardware error is detected.	G4 for 0.5 s, delay 0.5 s, C4 for 0.5 s
Shutdown Sound	Plays when the LipSync is shutting down in preparation for a reset.	C6 for 0.5 s, delay 0.25 s, G5 for 0.5 s, delay 0.25 s, C5 for 0.3 s, delay 0.5 s
Calibration Corner Tone	Plays when measuring corner point during full calibration.	A4 for 0.3 s
Calibration Center Tone	Plays when measuring center during center reset.	A6 for 0.5 s

User Interface

In addition to the visual and audible feedback elements, the Hub also has an OLED screen to provide additional visual feedback as well as a user interface.

Menu Access

To ensure that most users can access and navigate the menu, the menu needs to be accessible using minimal inputs. Since activating joystick calibration is a common task that should be accessible through the menu, it was decided to limit menu navigation to just sip and puff. In case the joystick calibration is off or doesn't work properly, the joystick isn't required to navigate the menu. This also means that the menu can't be on at all times – it needs to be possible to activate and deactivate the menu using the same inputs so that the device can be used normally.

The Hub buttons and assistive switches can also be used to access and navigate the menu for use by users who can't use sip and puff, or a support person.

Activating Hub Menu

The Hub Menu can be activated by the user through a very long puff. It can also be activated by a very long press of assistive switch 1, a very long press on the "Next" button on the Hub, by pressing "Next" and "Select" simultaneously, or by pressing assistive switch 1 and 2 simultaneously.

Switch Scanning Menu Navigation

Once the Hub Menu is activated, the menu can be navigated using two inputs like two-switch switch scanning. One input is "Next" and one input is "Select.

For menus that are longer than the 4 items that can be displayed on the screen, when the cursor is on the final item on the screen, and the "Next" input is given, it will automatically scroll the menu page to the next menu item, and the first menu item will disappear. When the cursor is on the final item in the menu, "Next" will go back to the first menu item.

Table 15. Hub Menu Inputs and Outputs

Input			Output
On board button	Assistive Switch	Sip / puff	
Next	S3	Sip	"Next" – goes to next option
Sel	S1	Puff	"Select"
Next very long press	S1 very long press	Puff – very long	Exit menu

There is potential in the future to also use the joystick inputs when in settings mode, where up and down / next and previous could be controlled through moving the joystick up and down, and keeping the same method for select as above.

Menu Structure

This is the current overall structure of the settings menu:

- Exit menu
- Center reset
- Mode
    - Mouse USB
    - Mouse BT
    - Gamepad
- Cursor speed
    - (show current cursor speed)
    - Increase
    - Decrease
- More
    - Sound on/off
    - Light Brightness
    - Scroll Speed
    - Sip and Puff
        - Sip threshold
        - Puff threshold
    - Full calibration
    - Restart LiPSync
    - Factory Reset

Some sample screen layouts, depending on text size:

With “>” beside the current selection

Initial settings menu:

```
> Exit Menu
  Center Reset
  Mode
  Cursor speed
```

```
> Calibrate
  Mode
  Cursor speed
> More
```

Calibration menu:

```
> Center reset
  Full Calibration
  ... Back
```

Center reset:

```
Performing
center reset… Do
not move
joystick
```

```
Center reset
complete
```

Then back to main menu

Full Calibration:

```
Hold
Joystick
Top left
```
->
```
Hold
Joystick
Top right
```
->
```
Hold
Joystick
Bottom
right
```
->
```
Hold
Joystick
Bottom
left
```
->
```
Release
Do not
Mode
joystick
```
->
```
Joystick
Calibrated
```
Then back to main menu

Change mode:

```
 MOUSE USB
>MOUSE BT
 GAMEPAD USB
  ... Back
```
Select ->
```
Confirm mode
change to
Gamepad Mode
```
Select ->
```
Changing mode
and resetting
device
```
*only reset when switching between USB Mouse and USB Gamepad

Cursor speed:

```
Cursor speed: 2
> Increase
  Decrease
  ... Back
```

Scroll speed:

```
Scroll speed: 2
> Increase
  Decrease
  ... Back
```

Info:

```
Info:
V4.1.0
LS_abcd1234
Decrease
```

Menus to Add

There are a few menu options that would be useful to add in future version of the code:

- Menus to add in future versions:
  - Deadzone
  - Reversing sip and puff
- Menus to add later down the road:
  - Cursor acceleration settings
  - Sip and puff automated calibration

- Debug mode, using small text, show pressures and joystick outputs, states of buttons and switches
- Training mode, wizard that prompts moving joystick and doing sips and puffs
- Response curves

Menu functions

Exit Menu

The top menu item is to exit and deactivate the Hub Menu. After performing any other setting changes, or backing out of submenus, the menu display will always return to the top of the main menu, so it is easy to exit the menu after changing settings. When selected, you will be asked to confirm you wish to leave the menu and can select “Confirm” to exit, or “… Back” to return to the menu.

As shown in the table for navigating the menus, any function used to enter the menu, can also be used to exit it.

Center Reset

The Calibrate submenu contains the options “Center Reset” and “Full Calibration.”

A center reset will reset the neutral resting position of your joystick and should be used if you are experiencing drift. Do not touch the joystick while performing a center reset. This function calls on the API function “setJoystickInitialization”

Change Mode

The Change Mode submenu allows you to select one of three modes: USB Mouse, BT Mouse, or Gamepad. After changing modes, the Hub will reset and perform a center reset upon powering on. Do not touch the joystick until you see the Hub display read “Ready for use” and the mode.

Mouse Cursor Speed

The Mouse Cursor Speed submenu allows you to increase or decrease the mouse cursor speed in increments of 1 from a scale of 0 to 10. While changing the increment, the cursor can still be moved across the screen to test the current speed setting.

More

The More submenu contains further submenus for settings that likely won’t be used as often.

The current features included are Sound On/Off, Sip and Puff Threshold Adjustment, Restart Device, and Factory Reset:

- Sound On/Off will toggle the sound feedback for LipSync inputs on or off. There is no volume control.
- Sip and Puff Threshold Adjustment allows you to fine tune the required air pressure for sips and puff, independently. This means that you can have the puff threshold different than the sip threshold.
- Full Calibration resets the neutral and extent positions of the joystick. The Full Calibration of the LipSync should be completed upon initial assembly of the device. It may also need to be repeated if the user is experiencing strange cursor movements or no change in movement at the extents of the joystick. This function calls on the API function “setJoystickCalibration”
- Restart device is a way to power the device off and on again without unplugging it from power.
- Factory reset will return all settings back to default. This includes the operating mode, cursor speed, calibrations, and any settings adjusted through the API. It is suggested to perform a full calibration after a factory reset.
- Info display the firmware version and the device’s unique hardware ID / Bluetooth name.

Text size

The screen being used is 29.4 x 14.7 mm

Below is an example of text on the selected screen, where “Hello world” and “1x Scale Text” are written in the default text size, and “2x Scale Text” is written in 2x text size.

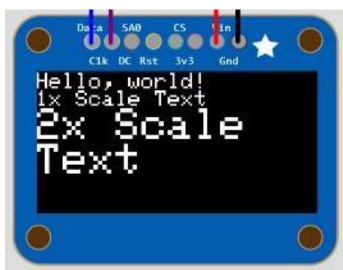


Figure 53. Display Text Size

For readability, the 2x text size was chosen.

Text scrolling

When the line of text is too long for the screen, it will appear cutoff when not selected, but will scroll when it is the current selection/when the cursor is beside it. A text scroll function was added so that the entire text can be seen.

Display Graphics

Monochrome images can be converted into static code using the following tools:

- https://javl.github.io/image2cpp/

Audio Feedback – Piezo Buzzer

- Start-up Sound plays when the device is first powered.
- Ready Sound plays when the device is fully initialized and ready for use.
- Error Sound plays when a hardware error is detected.
- Audio feedback in full calibration (beep before each corner/center reset)

Communication with host device

The LipSync is able to emulate an HID mouse, HID gamepad, and wireless Bluetooth mouse. At the beginning of the code, depending on the selected operating mode, an instance of either of the above options will be initialized, by calling the begin function for the corresponding class.

Currently, the LipSync appears as only one device. This is primarily for compatibility with the Xbox Adaptive Controller, which will only accept a sole USB HID Gamepad.

Adjust Settings

To adjust the settings on the LipSync, most settings a user would want to adjust are accessible through Serial commands within the API, and a subset of these are available through the Hub menu. The limits of the adjustable settings are only modifiable through values hard-coded in.

A full list of the API commands and settings are found in the API section below.

The main settings a user might want to adjust are listed in Table 16.

Table 16. User Settings

Setting	Default	Min	Max	Note
Cursor Speed	5	0	10	Changes the cursor speed in any Mouse Mode.
Sip Threshold	3.0	1.0	100.0	Changes the pressure required for a Sip input.
Puff Threshold	3.0	1.0	100.0	Changes the pressure required for a Puff input
Sound Feedback	On	Off	On	Turns the sound on or off.
Scroll Level	5	0	10	Changes the scrolling speed. **Only adjustable through API, not Hub menu.**
Inner Deadzone	0.05	0.0	0.5	Changes the distance the joystick must be moved from rest to cause cursor/gamepad movement. **Only adjustable through API, not Hub menu.**
Outer Deadzone	0.95	0.5	1.0	

Changing Mode

The operating/communication mode can be changed in the Hub menu by navigating to “Change Mode” and then selecting which mode to use.

In the code, this will change both the operating mode variable and the communication mode variable.

- USB Mouse Mode: comMode = 1 (USB), operatingMode = 1 (Mouse)

- USB Gamepad Mode: comMode = 1 (USB), operatingMode = 2 (Gamepad)
- Wireless Bluetooth Mouse: comMode = 2 (Bluetooth), operatingMode = 1 (Mouse)

When a mode change occurs through the Hub menu, a soft reset will occur (power cycle). This is so it gets initialized properly when the LipSync powers on.

The reason that instances of multiple modes cannot be initialized at once is because the Xbox Adaptive Controller will not allow a device that has been initialized as both a mouse and gamepad. Initializing both instances could also potentially lead to undesirable outputs.

When changing the mode through the Hub menu, the code calls on functions in the LSAPI file: setCommunicationMode and setOperatingMode.

If the mode has been changed through the API, a soft reset or power cycle must be conducted for the device to work properly.

Cursor Speed

Available through the API or Hub Menu. The Cursor Speed is set to levels between 0 and 10, which correlate to the values sent to the mouse move command. These levels between 0 and 10 correlate to maximum mouse movement commands between 2 and 17.5.

When the cursor speed level is changed, it is saved to memory and saved in the class instance for the joystick (LSJoystick) using js.setOutputRange.

Sip and Puff Thresholds

The sip and puff thresholds are adjustable through the API or Hub Menu.

The thresholds are individually set to levels from 1.0 to 100.0 hPa, which correlate directly to the required pressure difference in sensor readings to activate a sip or puff input.

When the sip and puff thresholds are changed, they are saved to memory and saved in the class instance for the pressure functions (LSPressure) using ps.setPuffThreshold.

Calibration

Center Reset and Full Calibration are available through API or Hub Menu.

Neutral Calibration

A Center Reset disables the mouse or joystick output, measures the position of the joystick, uses that measurement to set the dead zone around, and then reenables the mouse or joystick output.

When the center reset is being conducted, the software reads multiple values and saves them to a buffer. These are then used to evaluate the center position of the joystick input, using js.evaluateInputCenter, the minimum radius is updated using js.setMinimumRadius, and then saved to the global variable CenterPoint.

Extent ("Full") Calibration

A Full Calibration disables the mouse output, prompts the user to move the joystick to each corner and records measurements of the joystick's position at these maximum extents. The joystick output maximums are set to these maximum input readings. Lastly, a center reset is performed, and the mouse or joystick output is reenabled.

Current Corner Point Measurement

Process:

1. Process is initiated.
2. (Prompt the user to position and hold mouthpiece in sequence of corner (e.g., top left, top right, bottom right, bottom left).
3. Measures a corner point
    a. Takes a number of measurements
    b. Saves the maximum value
4. Stores the 4 points into memory.
5. (Prompt user to release the mouthpiece)
6. Measures and stores the neutral position.
7. Notify user that process is finished – returns to main menu.

Calculations:

1. From the 4 corner point values, the smallest distance from center to corner, or the smallest radius, is then found.
2. This minimum radius is stored in memory.
3. For future inputs, the minimum radius is used so that any values above this will be capped at this radius, which will be equivalent to the maximum output.

Once the Extents calibration is complete, the values are all saved to memory.

Reset Device

A soft reset performs a power cycle of the device. When this menu option is selected it calls the software reset function which releases all outputs, ends instances of usbmouse gamepad and btmouse, and then calls NVIC_SystemReset().

Factory Reset

The factory reset will make it easier to return the setting of the LipSync to a known good default state.

The factory reset sets key values saved in memory to their default values, using their API functions, which then saves the default values into memory. It then calls the softwareReset function, which performs a power cycle.

The values that are reset to defaults are:

- Communication Mode
- Operation Mode
- Debug mode
- Joystick deadzone
- Sip pressure threshold
- Puff pressure threshold
- Cursor speed level
- Scroll speed level
- Joystick Acceleration level

API

The LipSync has an Application Programming Interface that can be used to change settings through a serial interface. This will provide additional functionality for users as well as aiding troubleshooting and testing.

The API provides a way for more tech-savvy users to change settings that would otherwise be unavailable through the Hub Menu. In the future, it will also make it possible to use a user-friendly configuration app.

Serial Interface Setup

To use the API, you will need a host device capable of supporting a serial connection and a serial terminal emulator program. The Arduino IDE[6] is a convenient option.

Serial Settings

Select the appropriate COM port and set the speed to 115200.

Sending Commands

Once the serial connection is established, text commands can be sent to the LipSync. Make sure the line ending is set to 'No Line Ending'. The commands should be sent in all capitals. Commands are sent in a two step process.

First, send the following command to activate the interface:

SETTINGS

If the serial connection is setup properly, the LipSync will respond with:

SUCCESS,0:SETTINGS

[6] Available at no cost: https://www.arduino.cc/en/software

Next, send the desired command. The command consists of a two letter end-point code followed by a comma and either a zero or a one. Then, a colon is used to separate the parameter. All of the end-points are listed in **Error! Reference source not found.**.

Table 17. API Commands

Command	Function	Description	Action
SETTINGS	-	Readies LipSync for API Command	Enter API mode
EXIT	-		Exit API mode
MN,0:0	getModelNumber	Get Model number. Originally used to differentiate between USB / Gaming / Wireless / Macro variants of LipSync.	Reads Model number from memory.
VN,0:0	getVersionNumber	Get version number (V{N.NN}) This is shown as an integer (e.g., V3.0 = 30)	Reads version number from memory
ID,0:0	getIDNumber	Get unique hardware ID number. This is an 8 character alphanumeric value.	Reads ID number from hardware.
OM,0:0	getOperatingMode	Get operating mode (1 = USB Mouse, 2= USB Gamepad)	Reads operating mode from memory.
OM,1:{OM}	setOperatingMode	Set operating mode (1 = USB Mouse, 2= USB Gamepad)	Changes global variable operatingMode to inputted value and saves to memory.
LM,0:0	getLightMode	Get light mode (0 = All LEDS off, 1 = Minimal LEDs, 2 = Normal LEDs)	Reads light mode from memory.
LM,1:{LM}	setLightMode	Set light mode (0 = All LEDS off, 1 = Minimal LEDs, 2 = Normal LEDs)	Checks if input light mode value is valid, changes global variable lightMode, and saves to memory.
LL,0:0	getBrightnessLevel	Get current light brightness level (0 = Light off, 10 = brightest)	
LL,1:{LL}	setBrightnessLevel	Set light brightness level (0 = Light off, 10 = brightest)	
SM,0:0	getSoundMode	Get sound mode (0 = Sound off, 1 = Basic Sound, 2 = All sounds)	
SM,1:{SM}	setSoundMode	Set sound mode (0 = Sound off, 1 = Basic Sound, 2 = All sounds)	Checks if input sound mode is valid, changes global variable soundMode, calls buzzer.setSoundModeLevel (inputSoundMode), and saves to memory.
SS,0:0	getCursorSpeed	Get the mouse cursor speed value (Level 0-10)	Reads the cursor speed from memory.

Command	Function	Description	Action
SS,1:{Cursor Speed Level:0-10}	setCursorSpeed	Set the mouse cursor speed value (Level from 0-10)	Checks if input cursor speed is valid, saves value to memory and calls js.setOutputRange(tempCursorSpeedLevel)
SL,0:0	getScrollLevel	Get the mouse scroll level value (Level 0-10)	Reads the scroll level from memory.
SL,1:{Scroll Level:0-10}	setScrollLevel	Set the mouse scroll level value. (Level 0-10)	Checks if input scroll speed level is valid, and saves the value to memory.
PV,0:0	getPressureValue	Get pressure value (Pressure Difference in hPa)	Gets the current pressure values by reading both pressure sensors.
ST,0:0	getSipPressureThreshold	Get sip pressure Threshold (Pressure Threshold in hPa)	Reads the Sip pressure threshold from memory.
ST,1:{Threshold 1.0 to 100.0}	setSipPressureThreshold	Set sip pressure Threshold (Pressure Threshold in hPa)	Checks if input Sip pressure threshold is valid, saves value to memory and calls ps.setSipThreshold(inputSipThreshold)
PT,0:0	getPuffPressureThreshold	Get puff pressure Threshold (Pressure Threshold in hPa)	Reads the Puff pressure threshold from memory.
PT,1:{ Threshold 1.0 to 100.0}	setPuffPressureThreshold	Set puff pressure Threshold (Pressure Threshold in hPa)	Checks if input Puff pressure threshold is valid, saves value to memory and calls ps.setPuffThreshold(inputPuffThreshold)
IN,0:0	getJoystickInitialization	Get joystick neutral values (x and y)	Reads the joystick neutral value from memory.
IN,1:1	setJoystickInitialization	Perform joystick center reset to set new joystick neutral values (x and y)	Calls performJoystickCenter, takes neutral position readings, saves these to a buffer, calculates the center position, and saves to global variable CenterPoint.
CA,0:0	getJoystickCalibration	Get joystick calibration values (point 0, point 1, point 2, point 3, point 4)	Reads the joystick calibration values from memory.
CA,1:1	setJoystickCalibration	Start joystick full calibration	Calls performJoystickCalibration and goes through extents calibration as explained above.
IZ,0:0	getJoystickInnerDeadzone	Get joystick inner deadzone (0.0 to 0.5)	Reads the joystick inner deadzone level from memory.
IZ,1:{ Joystick Inner DeadZone Factor 0.0 to 1.0}	setJoystickInnerDeadzone	Set inner joystick deadzone (0.0 to 0.5)	Checks if input inner deadzone factor is valid, saves value to memory and calls js.setDeadzone(inputDeadZone)
OZ,0:0	getJoystickOuterDeadzone	Get joystick outer deadzone (0.5 to 1.0)	Reads the joystick outer deadzone level from memory.
OZ,1:{ Joystick Outer DeadZone Factor 0.0 to 1.0}	setJoystickOuterDeadzone	Set joystick outer deadzone (0.5 to 1.0)	Checks if input outer deadzone factor is valid, saves value to memory and calls js.setDeadzone(inputDeadZone)
CM,0:0	getCommunicationMode	Get communication mode value (0=No HID Output, 1=USB mode enabled 2=Bluetooth mode enabled)	Reads communication mode value from memory.

Command	Function	Description	Action
CM,1:{CM}	setCommunicationMode	Set communication mode value: (1 = USB mode, 2 = Bluetooth mode)	Checks if input deadzone factor is valid, saves value to memory, releases all output actions, and ends instances of btmouse and usbmouse.
DM,0:0	getDebugMode	Get debug mode value (0=debug mode off, 1=joystick debug mode 2=pressure debug mode 3=button debug mode 4=switch debug mode 5=sip and puff debug mode)	Reads debug mode from memory.
DM,1:{DM}	setDebugMode	Set debug mode value (0=debug mode off, 1=joystick debug mode 2=pressure debug mode 3=button debug mode 4=switch debug mode 5=sip and puff debug mode)	
SR,1:1	softReset	Perform a soft reset.	Calls the softwareReset function which releases all outputs, ends instances of usbmouse gamepad and btmouse, and then calls NVIC_SystemReset()
FR,1:1	factoryReset	Perform factory reset	Calls resetMemory(), then sets key values saved in memory to their default values, using their API functions, which then saves the default values into memory. It then calls the softwareReset function.

Timers

The LipSync implements a custom timer class called LSTimer manager to manage multiple timed tasks and provide basic callback functionality. This code is based on the Simpletimer class by Natan Lisowski. This allows the LipSync to do timed tasks like blinking an LED, while sequencing other tasks like sensor measurement.

The main code implements four main timers: actionTimer, calibTimer, pollTimer, and ledStateTimer.

1. actionTimer
   1.1. This timer keeps track of output actions. It is currently used to automatically release the gamepad buttons after a set duration.
2. calibTimer – There are two calibration timers.
   2.1. The first calibration timer is set up to make a set of joystick readings at a fixed interval (e.g., 5 readings every 100 ms).
   2.2. The second calibration timer is used to trigger the different steps of the calibration process.
3. pollTimer – There are a number of poll timers that trigger different processes in the main joystick loop.

3.1. Joystick – triggers the joystickLoop function. Gets new joystick values and generates output.
3.2. Pressure –triggers the pressureLoop function. Gets new pressure values and triggers actions.
3.3. Input – triggers the inputLoop function. Gets button and switch input values and triggers actions.
3.4. Bluetooth – triggers the btFeedbackLoop function. Checks on the status of Bluetooth connection.
3.5. Debug – triggers the debugLoop function. Generates outputs when Debug mode is activated.
3.6. Scroll – triggers the joystickLoop function while scroll mode is activated.
3.7. Screen – triggers the screenLoop function. Updates the screen.
3.8. Watchdog – triggers the watchdogLoop function to feed the hardware watchdog.

4. ledStateTimer – There are five LED state timers.
    4.1. The first LED timer is used to trigger the ledIBMEffect.
    4.2. The second LED timer is used to trigger the different steps of the ledIBMEffect
    4.3. The third LED timer is used to handle LED blinks.
    4.4. The fourth LED timer is used to handle Bluetooth status light blinks
    4.5. The fifth LED timer is used to handle error LEDs

Pressure – Sip and Puff

A timer is also implemented in the LSPressure.h code. The mainStateTimer is used to keep track of how long the pressure value has been held in different states.

Screen

A timer is also implemented in the LSScreen.h code. The screenStateTimer is used for timing of certain tasks like scrolling text.

Memory

Memory is required to store and retrieve settings. This improves the user experience by keeping the settings the same for a user if the device is powered off or switched to another device.

The LipSync uses the internal flash memory of the microcontroller. The nRF52840 contains 1024 kB of flash memory and 256 kB of RAM that can be used for code and data storage.[7] The LSMemory class is used to handle the implementation of the memory. This class uses the Adafruit Little File System to store and retrieve a JSON-formatted text file within the flash memory. The ArduinoJSON library (v7) is used to serialize and deserialize the individual settings.

Stored Settings

The settings stored in memory are listed in Table 18.

[7] https://docs.nordicsemi.com/bundle/ps_nrf52840/page/memory.html

Table 18. Flash Memory Settings

Key	Default Value	Data Type	Description	
MN	0	int	**M**odel **N**umber	
VN1	4	int	**V**ersion **N**umber - Major	
VN2	1	int	**V**ersion **N**umber - Minor	
VN3	0	int	**V**ersion **N**umber - Revision	
ID	0	int	Hardware Unique **ID**	
OM	1	int	**O**perating **M**ode	
CM	1	int	**C**ommunication **M**ode	
SS	5	int	Cur**s**or **S**peed	
SL	5	Int	**S**croll **L**evel	
~~**PM**~~	~~2~~	~~int~~	~~**P**ressure **M**ode~~	~~Removed 4.1~~
ST	3.0	float	**S**ip **T**hreshold [hPa]	
PT	3.0	float	**P**ressure **T**hreshold [hPa]	
AV	0		Joystick **A**cceleration **V**alue	
IZ	0.07	Float	**I**nner Dead**z**one	
OZ	0.95	Float	**O**uter Dead**z**one	
CA0	[0.0,0.0]	pointType	Center Point	
CA1	[-13.0,13.0]	pointType	Calibration Point 1	
CA2	[13.0,13.0]	pointType	Calibration Point 2	
CA3	[13.0,-13.0]	pointType	Calibration Point 3	
CA4	[-13.0,-13.0]	pointType	Calibration Point 4	
SM	1	int	**S**ound **M**ode	
LM	1	int	**L**ight **M**ode	
LL	5	Int	**L**ight Brightness **L**evel	
DM	0	int	**D**ebug **M**ode	

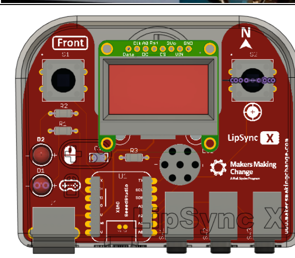
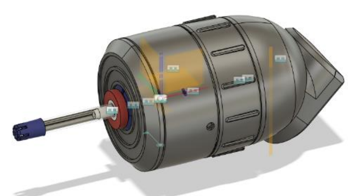
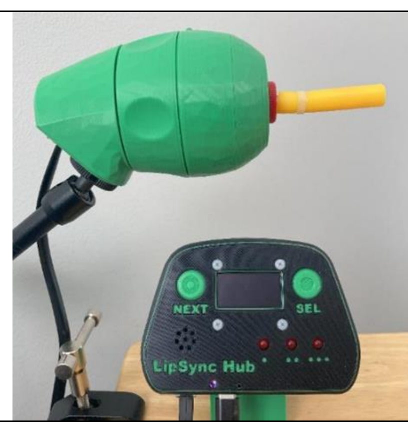
Memory Processes

The memory file is read on device startup. If the file is not found (e.g., on initial programming), then the file is created with the default values stored in LSConfig.h.

The default values are also saved into memory when a factory reset is triggered.

Design Iterations Snapshot and Feedback

The LipSync 4 went through several design iterations.

The_V4-Alpha I Design was the first integrated functional prototype featuring a unibody architecture, and a Hall-Effect, gimballed bearing joystick. This version was built and tested internally.	
The V4-Alpha II Design was featured a unibody architecture, a Hall-Effect, gimballed bearing joystick with an offset mouthpiece.	
The V4-Beta I Design was a functional prototype of the two-part modular architecture with a Joystick unit and a separate Hub. The Joystick had a revised, in-line Hall-Effect, gimballed bearing joystick. This version underwent internal testing of the joystick mechanism.	
The V4-Beta II Design was a revised version of the Beta I Design. This design was built internally.	
The V4.0- First Release was released in March 2024. At least 50 units have been built and deployed to users.	

V4-Alpha I Design

V4-Alpha I Design Overview

The goals of the Alpha I, improving on the LipSync 3.0, were the following:

- Better joystick with lower force and reduced drift
- Easier to adjust settings for the user
- Easier to source components
- Incorporate multiple input types beyond sip and puff (assistive switch access, buttons to be used by support person)

The Alpha I design features an offset axis joystick with a direct connection for the sip and puff pathway.

The 2 PCB setup features 3 NeoPixel LEDs and 3 buttons on the top of the device. These are intended for user feedback and secondary user access. There are also 3 switch jacks at the bottom of the rectangular PCB for assistive switch inputs. The smaller square PCB is intended to face upwards and the long rectangular PCB will go down the back of the device.

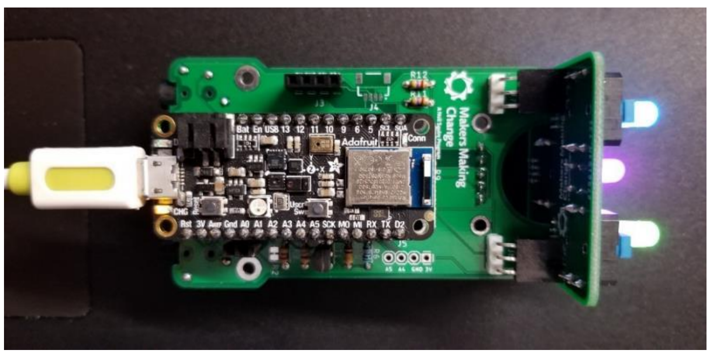


Figure 54. V4-Alpha I PCBs

The Alpha II design uses a TLV493D magnetic sensor breakout board and magnet for detecting movement, a LPS33HW pressure sensor breakout board and the onboard Arduino Feather Sense pressure sensor (BMP280) to detect sips and puffs.

Advantages

- Direct and secure connection from mouthpiece to pressure sensor
- Magnets all held in place with 3D printed parts (front enclosure and clip pieces)

Disadvantages

- A tall offset
    - Awkward non-symmetric mouthpiece range
- Magnets right at the front of the device
    - No option to add magnets from stronger centering
- STEMMA cables route from the gimbal to elsewhere in the device

    - Must be routed carefully to not affect joystick centering
  - Difficult to install in enclosure
  - Extra weight on the joystick makes it more prone to oscillation or accidental movement from bumps

Build Feedback

- Surface pad soldering component (SETMMA port) is very difficult to solder
- STEMMA-DuPont cables take up a lot of space on the DuPont side in female headers
- NeoPixel LED footprint is very small and difficult to solder.

Testing Feedback

Joystick

- The Luer bulkhead connections come in two lengths – the CAD model is for the short one, so additional space is required to accommodate both types.
- The STEMMA QT wires to the pressure sensor affect the center position of the joystick.
  - Silicone wires?
  - Increase centering force to compensate (e.g., more / closer centering magnet)
  - (carefully) remove the heat shrink on the longer (100 mm) cable
- The offset joystick axis complicates the design of the enclosure
  - Also results in too low of a centering force in the current arrangement
- The nylon fasteners don't secure the gimbal components reliably – they tend to strip out.
- Non-symmetrical inner bearing arm/mount
- Device must be mounted vertically
  - Joystick neutral position changes with different mounting angles
  - Joystick cardinal directions difficult to change depending on the mounting angle

PCB

- device must be mounted vertically
  - Buttons and LEDs have to stay on the top of the device to be accessible and functional

Opportunities for Improvement

- Reduce front profile and make it so the device can be mounted at various angles.
- Increase centering hysteresis and stability
- Create an enclosure

Sound Feedback

- A simple piezo speaker (e.g. https://www.digikey.ca/en/products/detail/pui-audio-inc/AB2040B-LW100-R/4147328), attached directly to the analog port provides suitable volume.

Tactile Feedback

- The Adafruit Vibrating Mini Motor Disc (i.e., https://www.adafruit.com/product/1201)) and the STEMMA QT DRV2605L provide a simple way to add-on tactile feedback.
- Attaching to the base didn't transfer much vibration to the mouthpiece; attaching the motor to the inner joystick component worked much better

V4-Alpha II Design

Figure 55. V4-AlphaII Design

Alpha II Design Overview

The Alpha II Design was a full function prototype featuring a one-piece integrated architecture. The enclosure features a separate, modular mounting interface that could be rotated around the main body of the enclosure. To accommodate the mounting arrangement, the PCB orientation was changed from vertical to horizontal.

The joystick was an updated version of the Alpha I design, with an offset mouthpiece and three-bearing gimbal.

This version featured three Neopixel LEDs oriented on the top of the enclosure and three corresponding tactile buttons for input.

This version of the design was tested internally and went for user testing with five users.

Figure 56. V4-AlphaII PCBs

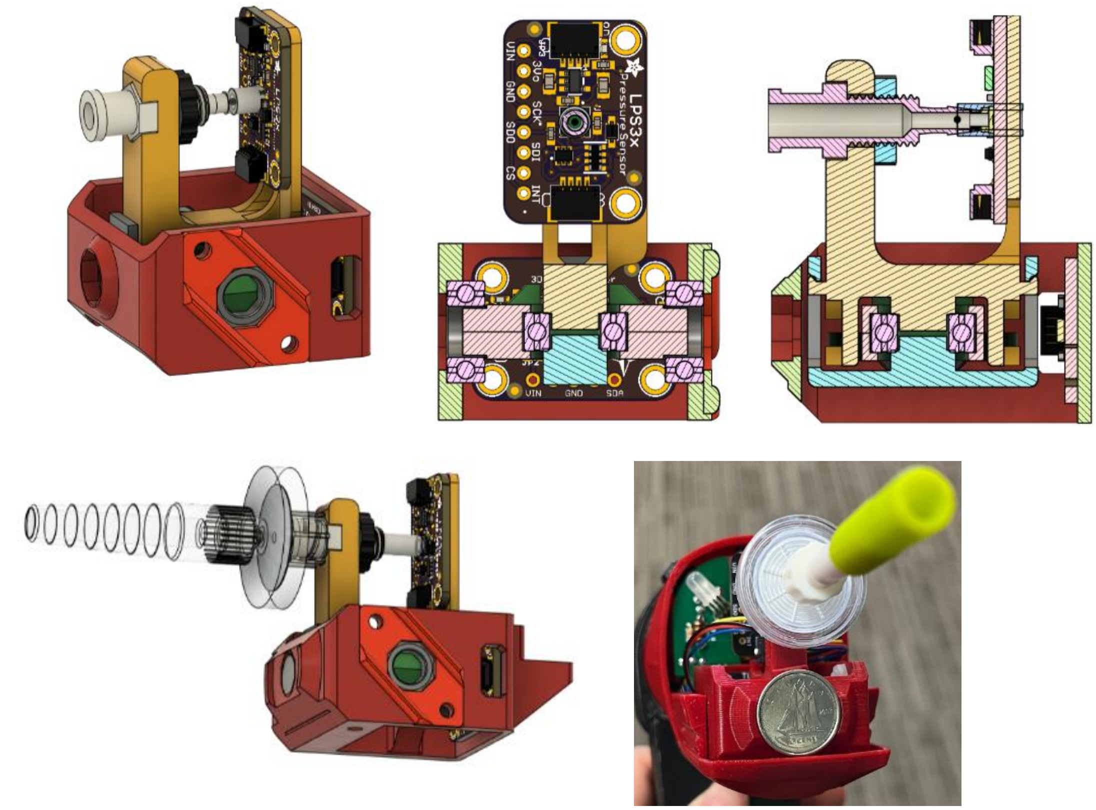


Figure 57. V4-AlphaII Joystick Mechanism

Reasons for current configuration:

- Mouthpiece is offset vertically so the magnets can be in line with the magnetic sensor.
- Pressure sensor is oriented the long way vertically so the swing arm can be printed on its side without supports.
- A dime was used to help strengthen the magnetic centering force.

Firmware

- Functional API
- Buttons and switches able to trigger center reset, calibration and start Bluetooth mode
- Neopixel feedback for inputs

Build Feedback

- Reduce different values of resistor
    - Match switch resistors and Neopixel resistor (300-500)
- Buttons are difficult to assemble
- Front of the enclosure is not aesthetically pleasing
- The closely spaced Neopixel pins are difficult to solder and easy to bridge
- If the Neopixel is soldered in backwards, it can burn out and explode if powered

Issues to address

- Neutral centering (hardware)
- Joystick drift (software)
- Finalizing / selecting user interface lights
- Input Mapping (Buttons / External Switches)
    - What to do with middle button
    - middle switch?
    - Do buttons map to sip and puff, or?
- Assembly improvements
- Light feedback visibility (enclosure)
- Joystick calibration process
    - Method / approach
    - Order of points
    - Visual feedback
- Joystick processing
- Linearity
    - Cursor speed scaling
    - (Acceleration?)
- Single absolute pressure sensor / pseudo differential
    - Automatic update
    - Manual update?
- Pressure threshold calibration?
- PCB changes
- External reset button – pusher
- External reset button – PCB
- Rotational mounting adjustments requires tools

Internal Testing Results

Neutral Hysteresis / Drift

- Tends to drift

Double Clicks

- Difficult to perform

Calibration Process

- Activating by pushing two buttons simultaneously was difficult / unreliable
- Process is confusing – lots of different colors and timing is challenging

Bluetooth

- Cursor speed is far too slow at Bluetooth

Feedback Light Orientation

- Can be difficult to see far side light if not oriented directly in front of the user

User Testing Feedback

- Joystick force is much lighter
- Movement is much smoother
- Small movements can be challenging
- Enclosure too large, too bulky
- LEDs from inside white enclosure are distracting while using in dark environments

User 1: EP

Nature of disability:

LipSync Experience: >3 years

LipSync Usage: Daily

Initial Impressions: "I frickin love this new LipSync" / "Smooth like butter"

Compared to previous LipSync:

- Force: Much
- Range of motion: better
- Fine positioning / Ease of accurate clicking:
- Gross positioning / Ability to move quickly:
- Movement Symmetry:
- Calibration:

Setup: Windows PC via USB

Feature Usefulness:

- Rotatable Mounting:
- Separate visual feedback lights:
- Additional visual feedback light colors:
- External Switch Inputs:
- Application Programming Interface:

Preferred Mouthpiece Style: Yellow tip with rubber end – flared tip helps control the tip better; better able to wrap lips around

Preferred USB cable connection:

Suggestions:

- Very accurate and sensitive
- “So much easier to use” “would have a hard time going back”
- Far less fatigue

1. Mouthpiece Options
   1.1. Yellow tip with rubber end
   1.2. Blue tip was too difficult
   1.3. Hard plastic tip – fine; not nearly as nice on teeth as yellow tip

Mouse control

- Sensitivity for left click is perfect
- Drag activation time is too long - Would like to decrease time to activated

Struggle with short sip – difficult to get good seal without moving mouthpiece

2. Increased sensitivity would help

Very long puff could be remapped to something else

Won’t be able to do medium sip or long sip

- Being on vent / muscle strength is a limiting factor

Mouse Control

Had to turn down cursor speed to be able to get fine movement control

- Would be handy to have cursor acceleration to maintain low end speed with higher high-end speed
- This is the biggest challenge – especially dragging windows / files
- Different speed levels for different joystick activations

Visual Lights

- Internal blue flashing light and steady green light is annoying
  - This is especially annoying while working at night
  - This is probably made worse by the work environment (they typically work at night)
  - Dimmable / or recess them into the case a bit
- Color choices seem intuitive (Blue-Bluetooth / Red-don’t touch)
- Having flash every time you click is annoying (left led) / distracting
- Drag activation light is good

Has failed to start a couple time when windows starts. Cycling the power fixed the problem.

3 or 4 times in last week, the left and right click has swapped. This is fixed by cycling the power

- (i.e. applied puff and instead of a left click, the right click would be activated)

Mounting

Uses the LipSync slightly off to the Left

- Being able to rotate the axis of the mount is nice
- Liked the range of options for mounting (e.g,. RAM Mount, Gooseneck)
- Mounting angle was off compared to other LipSync (by about 90 degrees) – used plumbers tape to avoid having to make more significant changes. Had to angle the straw upwards, and lower the height of the gooseneck

External Switches

- Hasn't tried – doesn't work for their setup

USB

- likes having the ability to be able to remove the USB cable from the device
- easier to install (e.g. threading device onto gooseneck)

Overall Size

- Too big - "A little jarring at first"

Drift

- Very minimal issues – tended to drift to the left – fixed after slight nudge

Would shortcuts be useful?

- Would like to be able to do combinations (e.g., combined movement and pressure)
- Four timed levels of puff would provided greater option due to difficulty with sip
- Additional puff thresholds
    - Soft puff – left click, hard puff – right click
- E.g. a hard puff then a direction
    - Even a long puff, then moving joystick in a direction

Hard puff to activate menu

Quadrants or left and right - colors to

GUI / Interface to be able to (mostly?) independently change settings and mappings

User 2: KS

Nature of disability: Physically weak, cannot use my hands

LipSync Experience: >3 years

LipSync Usage: Daily

Initial Impressions: It's a little large. But I like that it requires very little force to move the curser.

Most liked feature: How it requires very little force to move the curser.

Least liked feature: Size

Compared to previous LipSync:

- Force: Much better
- Range of motion: Much better
- Fine positioning / Ease of accurate clicking: Similar
- Gross positioning / Ability to move quickly: Similar
- Movement Symmetry: Similar
- Calibration: Similar

Setup: USB connected to Host device – Windows Computer

Feature Usefulness:

- Rotatable Mounting: Neither helpful nor unhelpful
- Separate visual feedback lights: Somewhat helpful
- Additional visual feedback light colors: Somewhat helpful
- External Switch Inputs: N/A
- Application Programming Interface: N/A

Preferred Mouthpiece Style: Original LipSync mouthpiece (hard plastic tube)

Preferred USB cable connection: Removable

Suggestions: I couldn't get the long sip and puff for scrolling to work. I'm not sure if it's a problem with my laptop or the LipSync. But I recently bought a new laptop that I can test on further.

User 3: OA

Nature of disability: Spinal cord injury; quadriplegia

LipSync Experience: >3 years

LipSync Usage: Occasionally

Suggestions:

1. Fine positional accuracy
    a. Difficulty: position cursor on small things
    b. Idea: more physical resistance
    c. Idea: Cursor mapping dependent on deadzone setting
2. Cursor movement during long sips and puffs
    a. Idea: disable cursor output when sip and puff
3. Activating long puff is too long
4. False activations during movement
    a. Putting lips onto the mouthpiece to move mouthpiece
5. Add timing intervals to API so user can adjust delays between short, long, very long
6. Idea: Would like drag behaviour more to be more consistent with Quad Stick.
7. Two input modes
    a. Analog
        i. Puff and release = Sip
        ii. Puff and hold = Drag
        iii. Puff and hold for a while = Drag Toggle
        iv.
    b. Digital
        i. Activate on release: short, long, very long
    c. Mode Selection Mode
        i. Very long puff?
        ii. Single-input scanning
        iii. Double-input scanning
8. Compatibility with dwell-click software

User 4: DD

Nature of disability: Spinal Cord Injury, quadriplegia

LipSync Experience: >3 years

LipSync Usage: Daily

Initial Impressions: Much improved; “A lot easier to move around”

Most liked feature:

Least liked feature:

Compared to previous LipSync:

- Force: Better
- Range of motion: Better
- Fine positioning / Ease of accurate clicking: Not answered
- Gross positioning / Ability to move quickly: Not answered
- Movement Symmetry: Better
- Calibration: Not answered

Setup: Worked with laptop and telephone

Feature Usefulness:

- Rotatable Mounting: Not answered
- Separate visual feedback lights: Not answered
- Additional visual feedback light colors: Not answered
- External Switch Inputs: Not used / suitable
- Application Programming Interface: Not used

Preferred Mouthpiece Style: Blue dental tip (matches what user uses with different devices)

Preferred USB cable connection: Not answered

Suggestions:

- Phone – default speed was too quick for phone
- Was drifting a bit on phone – unplugged and plugged it in again
- Cursor acceleration would be useful – especially on desktop with larger monitors
- Diagonal movement are much better
- Mounting is still challenging, but the Loc-Line may be a good solution
- Ability to independently to answer / hang-up phone
- Sip and puff morse code with repeat functionality and sufficient speed (32 WPM? ~160 characters / 60 seconds)
- Different tones for dots and dashes – ideally option for headphone jack

Opportunities for Improvement (OFIs)

1. Tee Nut Insert
    1.1. Modify tee nut slots to handle 3, 4, and 6 prong variants
2.

1. Tolerancing of parts for improved fit
2. Reduce offset from center
    a. Reduces height
    b. Better centering under low centering force
3. Strength/printability of parts
    a. Clip? Typically breaks
4. Reduce fasteners

3. Main PCB
    3.1. Add small circle in corner of optoisolator footprint stencil to indicate pin 1 more clearly than notch.
    3.2. Remove surface mount connector from Main PCB
    3.3. (Consider spacing STEMMA QT holes to accommodate commercial JST connector PCB) (Flip?)
    3.4. Add reverse-facing tactile button for external reset
    3.5. Increase space between Feather header and Switch jack.
    3.6. Reduce number of different valued resistors if possible
    3.7. Differentiate / adjust value of 470 ohm resistors from 4.7K resistors (i.e. so they aren't one colour apart and confused)
    3.8. Align SW3 Jack footprint with other switches
    3.9. Move Switch Jack position as close as possible to edge.
    3.10. (Explore repositioning switch jacks to align surfaces with Feather / USB Port
    3.11. (e.g., make board a little wider, move switches outside of Feather Header
    3.12. A4 and A5 labels seem to be reversed
4. Enclosure
    4.1. (Aesthetics / Style)
    4.2. Mass and size reduction
    4.3. Enclosure is too thick for center light pipe – need to thin out or add counterbore.
    4.4. Light pipes are too close to the LEDs, and not centered
    4.5. Joystick is not centered in the enclosure
    4.6. Filter interferes with sides of enclosure
    4.7. Placing screw to attach joystick to enclosure is very difficult.
    4.8. Side supports on the enclosure are too narrow to allow the switch jacks on the PCB to be slid in
    4.9. Reset pusher is too tall and interferes with enclosure button when attempting to slide in PCB
    4.10. Button holes don't align with buttons
    4.11. Side mounting screw counterbore is not deep enough; screws protrude out the side.
    4.12. Add branding: MMC / LipSync
    4.13. Add labelling or something to distinguish external switch inputs
    4.14. Add labelling or something to indicate reset button slot
    4.15. Ensure that all LEDS are visible (side LED may be obscured if mounted to the side of users' mouth)
    4.16. External button pushers can stick
    4.17. Re-orient so layers parallel w/ motion
    4.18. Optimize button connection so buttons move more independently

4.19. Re-print at higher resolution
4.20. Tamper proof front opening (child safe and keeps it from being broken accidentally)

5. Joystick Assembly
   5.1. Add means to enable more repeatable mounting options
      5.1.1. E.g., secondary screw hole / nut insert
   5.2. Add cable routing features for STEMMA QT wires to reduce chance of abrasion during assembly
   5.3. Optimize / look at cable routing.
   5.4. Mitigate side bearing printing issue – additional clearance, etc. (done)

V4-Beta I Design

Beta I Design Overview

The Beta I design was a functional prototype of the two-part modular architecture with a Joystick unit and a separate Hub. The Joystick had a revised, in-line Hall-Effect, gimballed bearing joystick. This version underwent internal testing of the joystick mechanism.

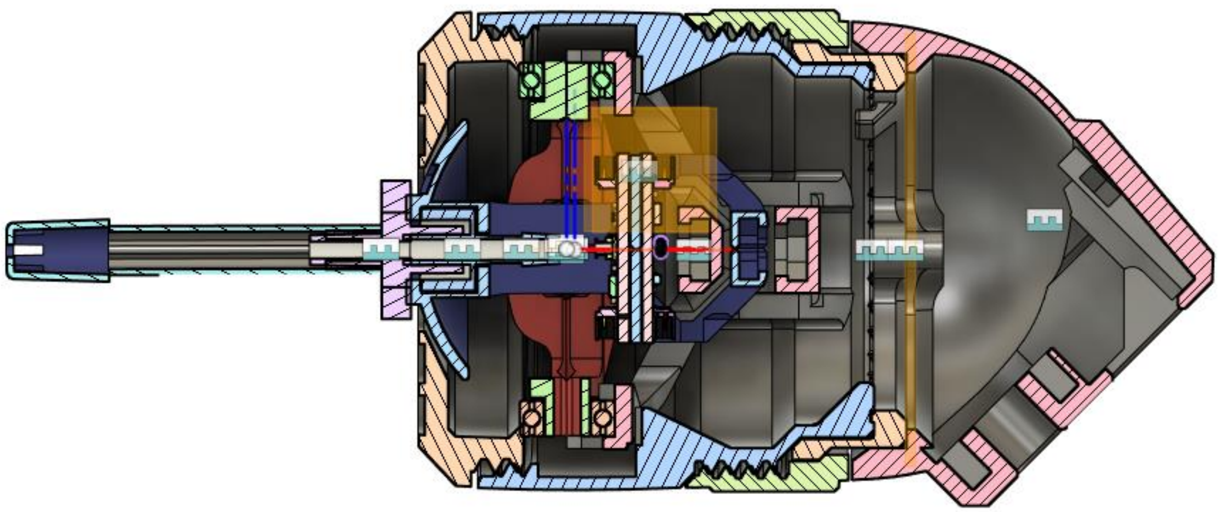


Figure 58. V4-BetaI Joystick Design

The Beta I joystick still requires mounting for the LPS22 breakout board and some type of cable management, but is the first iteration of the split LipSync concept.

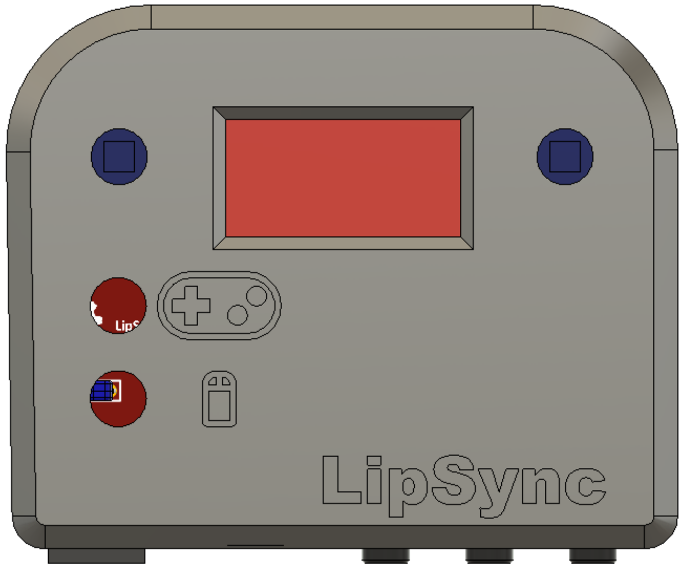


Figure 59. V4-BetaI Hub Design

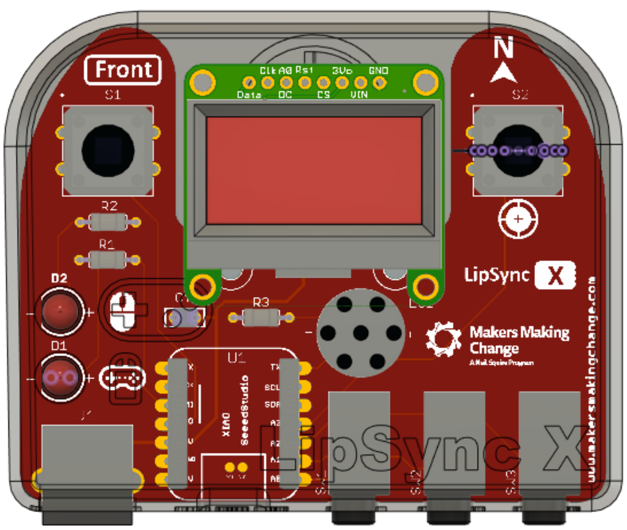


Figure 60. V4-BetaI Hub PCB

The Beta I Hub had an OLED screen and 2 LEDs for indicating operating mode. It is built around a Seeed Studio XIAO nrf52840 microcontroller with a buzzer for audible feedback, three switch jacks for assistive switches and an RJ11 jack to connect to the Joystick.

Beta Testing Process

The beta testing will be conducted by the internal team designing the LipSync. This testing process will assess general functionality, including whether changes to the PCB are needed, and some long term testing as was performed on the Alpha II design. All tests are detailed in the following sections.

All beta testing code can be found in this GitHub repository branch.

Test Results

Test	Tester	Pass/Fail	Notes
Screen suitability test	Josie	Pass?	See table below
LED test	Stephan	Pass	The LEDs are very easy to distinguish as long as the light is bright enough to see the Hub. If it is too dark to see the hub, you can still distinguish them but it is more difficult as light bleeds into one LED from the other, and glows through the enclosure a little bit (3D printed in a light blue, darker filaments may not have this issue).
Buzzer test	Stephan	Pass	The buzzer is audible 2m away in a room with sound levels up to 75 dBA (measured on a phone with the NIOSH Sound Level Meter app).
Number of buttons test	Stephen	Pass	Can very easily get through all options within 10 seconds. Even if we added more layered menus I don't forsee an issue with the number of buttons.
Disconnect/reconnect test	Jake	N/A	Cursor stops when RJ11 cable is unplugged. Requires reset after plugging the RJ11 cable back in.
Reset button test	Josie	Pass	Can press with paperclip or tweezers through hole in hub enclosure
Joystick mass	Stephan	N/A	135 g RJ11 cable was not included in mass.
Neutral Offset test	Stephen	N/A	Test was not used for verification or validation, but to measure the neutral position.
Hysteresis test	Stephen	N/A	Test was not used for verification or validation, but to determine the range of output values the joystick neutrally rests at.
Extents test	Stephen	N/A	Test was not used for verification or validation, but to measure the readings at the joystick extents to determine the symmetry and shape of the joystick analog output
Sip/puff functionality			
Bluetooth	Jake	Pass	Tested BLEKeyboard connection to smartphone using demo code. No additional LEDs light up – the code uses the user LED to indicate connection status (Blinking blue and red)
Software Reset	Jake	Pass	Succesfully able to reset micocontroller through software.

Screen suitability test

Test	Condition	Able to read easily?	Notes
Angle	Flat	Yes	
	Tilted forward/backward up to 45°	Yes	
	Tilted left/right up to 45°	Yes	
Lighting	Dark room	Yes	
	Bright room	Yes	
	Outside	Yes	
Distance	0.3 m away	Yes	
	0.6 m away	Yes	
	0.9 m away	Yes	
	1.2 m away	Yes	
	1.5 m away	No	Only with considerable effort
	1.8 m away	No	Not well. Can distinguish words from each other but without knowing what the menu says it would be difficult

OFIs for V4-Beta I

1. Improve printability of threads on rear piece
    a. Remove a portion of threads at the top of the rear enclosure piece for printability
    b. Horizontal holes: https://www.hydraresearch3d.com/design-rules#unsupported-holes
    c. Thicker threads
    d. Machine screws and nuts
    e. #4 screws (also works for RJ11)
2. Add some type of outer grip for locking ring
    a. Change mating surface from flat to Hirth
3. Chamfer/dogpoint the threads to easily start threads
4. Bring in the back (make shorter overall)

V4-Beta II Design

V4-Beta II Design Overview

The Beta II joystick was a rapid iteration of the OFIs from the Beta I version. This version of the design was built internally.

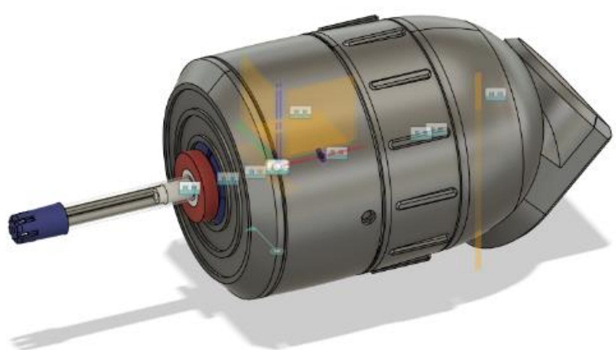


Figure 61. V4-BetaII Joystick Design

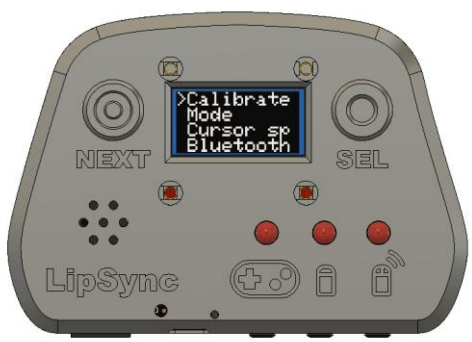


Figure 62. V4-BetaII Hub Design

The V4 Beta II Hub has an additional feedback LED in a different spacing and arrangement. The labels on the LEDs were kept to represent operating mode.

The V4-BetaII firmware had minimal changes from the V4-AlphaII to accommodate the different hardware (e.g., LEDs instead of Neopixels). The menu structure was created and tested, but was not integrated into the main code.

Figure 63. Initial Hub Menu Testing

Internal Testing of Internal Beta

1. Physical joystick Interference
2. Long-term magnet induced PLA creep
3. Manual function verification
    a. Short / Long / Very Long Puff
    b. Short / Long / Very Long Sip
    c. Joystick Motion
4. Automated displacement testing
5. Automated X-Y movement testing
6. (Possibly) Automated pressure cycle testing

V4-Beta II Internal Build Feedback

Joystick

1. Investigate Super-glue free method to retain magnets
2. Need more clearance between ¼-20 tee nut and Rj11/25 connector – at least 25 mm diameter + some space for Loc-Line connector / tripod pivot.
3. Need a better method for securing magnet into sled – can here it clicking and moving around when moved in the same direction. (e.g. horizontal)
4. Modify shape of front so it's a little less blunt / less visual impact.
5. No way for user to know if cable is connected / device is powered.
6. Had a hard time getting the sled into the enclosure. Could longer rails be added to help with alignment? Or maybe a chamfer to help with alignment? I also found the joystick would twist while trying to inert into rails
7. Mouthpiece has air leak

Hub PCB

1. Increase PCB footprint for capacitor. Radial Capacitor lead spacing is ~5 mm, PCB footprint holes are only 2.5 mm apart. Fit is possible but awkward.
2. Reduce microcontroller footprint to one set of holes instead of two.
3. Update LED graphics
4. Buzzer footprint has + and – on terminals, but the buzzer appears to not have a polarity

Hub Enclosure

1. Printable separate bottom panel for dual colors
2. V1 - Printability of buttons – some fused together with the enclosure and are not movable, and the circle around the square indent for the button itself has thin walls at the corners of the square that didn't print well.
3. V2 – Not quite enough space for buttons to move up and down, if the enclosure is closed tightly it keeps the buttons pressed
4. Possible to insert PCB screws in too far and dent out the back of the enclosure
5. Increase height of internal PCB screw posts so they cannot bulge out the back

Joystick CAD

1. Enclosure shape / size refinement
2. Add low poly texturing
3. Options for customizable aesthetic elements?
4. Investigate backwards-compatible option for air path / filter
5. Investigate options for sealing mouthpiece (e.g., Hot glue, etc)
6. Add feature / label to distinguish between Inner_Gimbal_1 and Inner_Gimbal_2.
7. Add feature / label to distinguish between Outer_Gimbal_1 and Outer_Gimbal_2.
8. Improve printability of Inner Gimbal prints (corners on print face, surface area)
9. Improve printability of shield (remove any hard corners)
10. Increase strength in magnet arms
11. Increase clearance for LUER connector in inner gimbal prints
    a. Add set screw? Glue in place?
12. Add clearance for the bearings (Effort: 15 min Priority: HIGH)
13. (Possibly) Add ruler on a gimbal part for cutting tubing (Alternative is put it on the printed instructions)
14. Match t-nut housing to the hub t-nut/fix up tolerances **]**
15. Add clearance to screw slots on rear enclosure
16. LPS22 snap is too stiff
17. Add clearance between cylindrical mating faces of rear enclosure and inner locking ring – Or add a point to the top of the rear enclosure round opening to accommodate drooping
18. Outer locking ring too thin in the threads (minimum walls required)
19. Cable management feature on rear housing needs modified for
    a. Also needs modified for the length of cable left after routing

20. Add versioning number to the model (internal or on the bottom)
21. Investigate options for mounting future lip switch
22. Cutaway version for demos?
    a. Gear on front cap (for a demo/SM post at least)
23. Consider modifying the way the rear enclosure and inner ring are keyed (currently a small bump and hole), right now it is possible for someone to assemble it the wrong way and then have it bulge out and things not line up properly.

Hub PCB

1. Modify C1 Footprint to match radial capacitor – it's too small [Effort: 15 minutes, Priority: HIGH]
2. Update repository link
3. Got several comments about finding a jack that had better spaced pins
4. Add silkscreen to indicate orientation of stemma breakout board
5. Rotate the labels on the stemma J2 to match breakout board
6. Remove buzzer polarity markers
7. Remove second set of holes from microcontroller
8. Check/modify jack footprint to avoid overlapping slot / hole error
9. Remove J3 label to avoid confusion
10. Add version number
11. Reset button hole (enlarge or add an internal post to guide to the button)

Airpath Safety Improvements

1. Blue tip on dental straws proved to be a choking hazard in internal user testing
    a. Silicone straw was chosen to replace the tip and length of the dental straw was reduced
2. Future OFI to find a food safe material with the same ID as the dental straw with a stiffness between dental straw and silicone straw

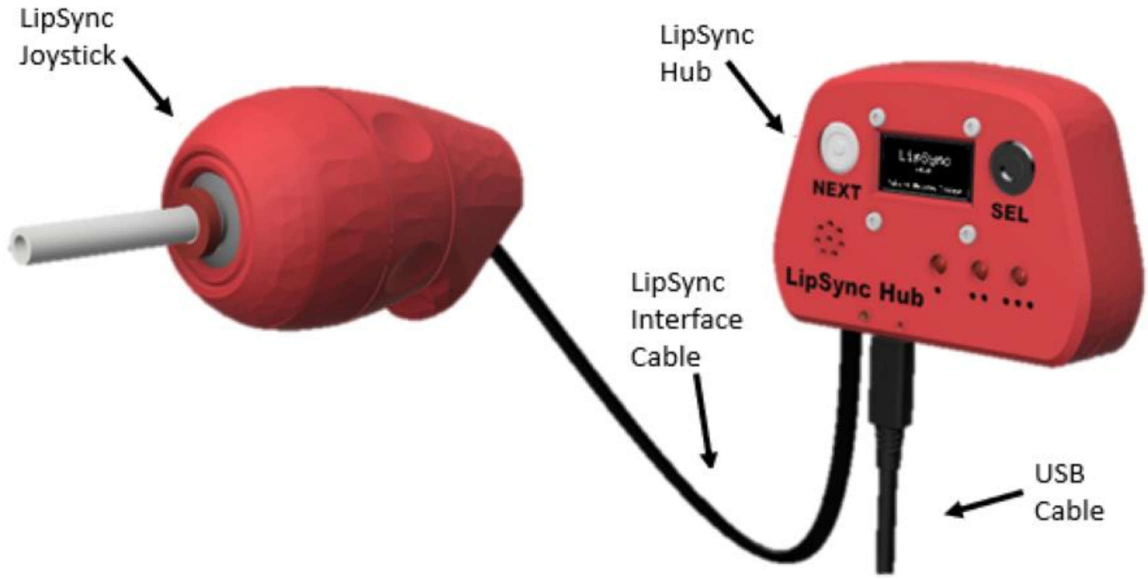
V4.0- First Release

V4.0 Design Overview

Architecture

The main architecture of the LipSync v4.0 consists of the LipSync Joystick, the LipSync Hub and the LipSync Interface Cable (RJ11 phone cable) in between the two, with a USB-C cable to connect to the host device.

Figure 64. V4 Final Design



Item	LipSync Joystick	LipSync Hub
Size (Length x Width x Height) [mm]	170 mm x 53 mm x 66 mm	104 mm x 29 mm x 72 mm
Mass	123 g	90 g
Power consumption	~0.1 W (20 mA @ 5 V)	
Joystick Angular Range	± 10°	N/A
Joystick Movement Range	± 10 mm (using standard mouthpiece)	N/A
Operating Force	~30 grams-force	N/A

Joystick

The LipSync Joystick is comprised of a 3D printed enclosure that contains a Hall-Effect based joystick and a sip and puff pressure system. The enclosure has a rotatable mounting interface that can be rotated and locked in place by a threaded ring. A ¼”-20 UNC mounting connection and an RJ11 connector provide the physical and communication connection.

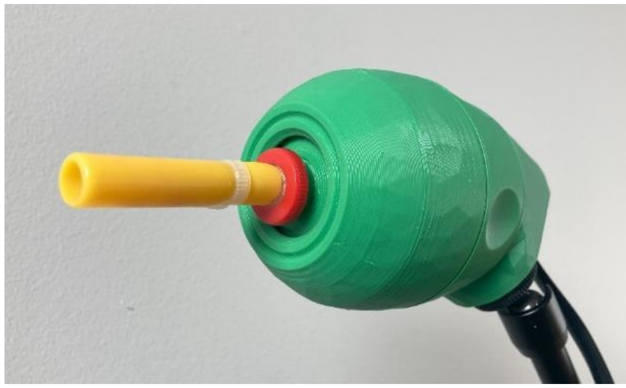
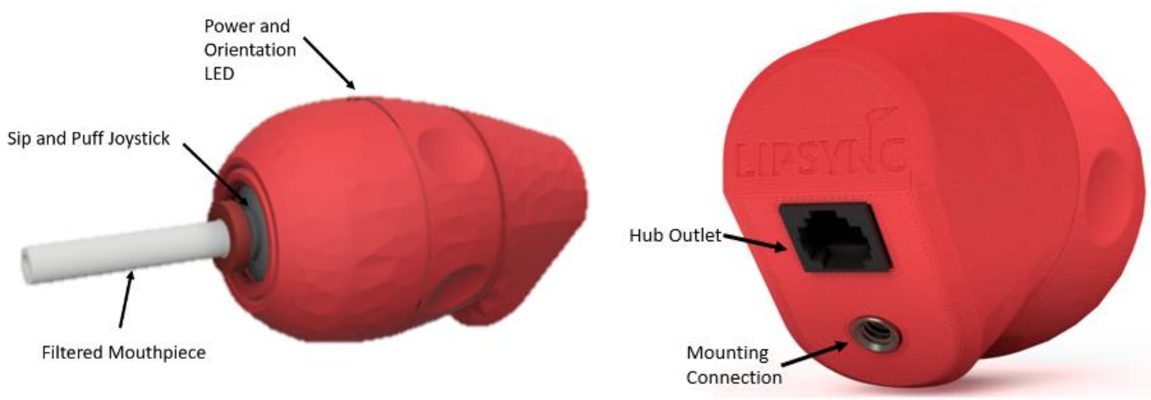
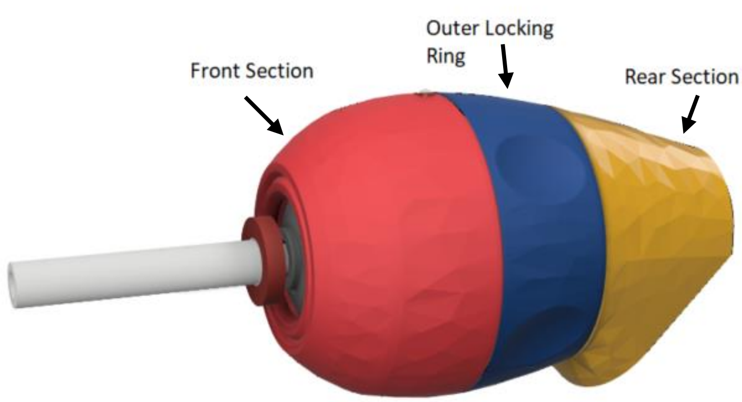


Figure 65. V4 Joystick

Joystick Features





Sip and Puff Joystick: Measures the movement of and pressure applied to the mouthpiece.

Filtered Mouthpiece: The mouthpiece is a consumable part that uses a hydrophobic filter to protect the interior pressure sensor.

Power and Orientation LED: The Power and Orientation LED signifies that the joystick is plugged into the Hub when powered and is also used to properly orient the joystick. The LED is on the very top middle on the joystick.

Rotatable Mounting: The Mounting Angle of the LipSync can be adjusted by loosening the Outer Locking Ring, rotating the Front Section, and retightening the Outer Locking Ring.

Hub Port: The Joystick plugs into the Hub using the LipSync Interface Cable.

Mounting Connection: The joystick can be mounted using the ¼-20 (standard camera mount) threads on the rear section.

Joystick Main Components

The main subassemblies that make up the joystick unit are the Gimbal, Mouthpiece, and Enclosure. Within each are key components that the subassembly is built around or made up of.

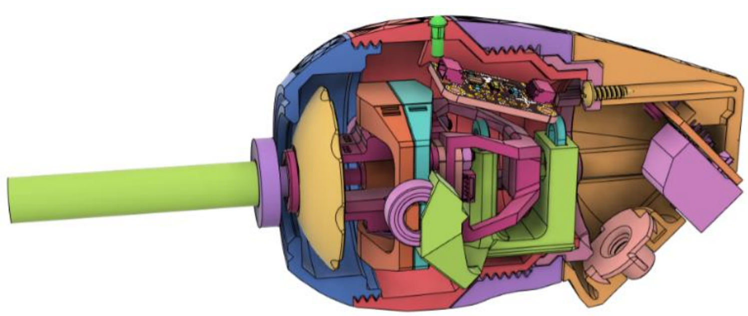


Figure 66. V4 Internal View

Joystick Mechanism

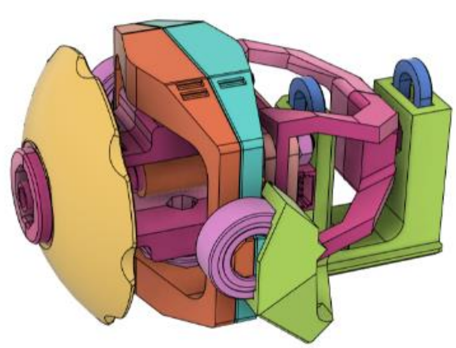


Figure 67 V4 Joystick Mechanism

The gimbal is made up of 3D printed components utilising ¼” ID x ½” OD ball bearings for smooth axial movement, 8mm x 2.5 mm (Diameter x Thickness) N35SH pill magnets for magnetic sensing and centering, an [Adafruit TLV493D breakout board](#) for magnetic sensing, and an [Adafruit LPS33HW](#)

breakout board for gauge pressure sensing. These breakout boards connect through STEMMA QT cables, which carries an I2C signal.

Mouthpiece

The mouthpiece is made up of a hydrophobic syringe filter, dental suction tube, 1/8" silicone tubing, food-safe silicone sealant, and a large silicone straw.

Figure 68. V4 Mouthpiece

Enclosure

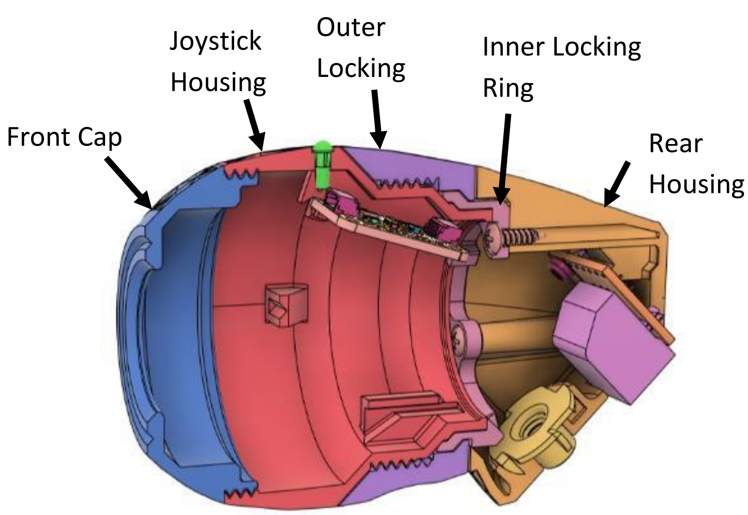


Figure 69. V4 Joystick Enclosure

The enclosure is made up of 3D printed components comprising of a screw-on front cap, a joystick housing, a screw-on outer locking ring, an inner locking ring, and a rear housing.

The joystick housing encloses the joystick gimbal, and an Adafruit LPS22HB breakout board used for ambient reference pressure sensing. The light pipe above the LPS22HB breakout board uses the onboard LED to signify power and the upright position of the joystick. This board connects to the gimbal breakout boards through a STEMMA-DuPont cable.

The inner locking ring and joystick housing have small protrusions and cuts that mesh together to lock the relative rotational movement between the front and rear sections when the overall joystick is assembled.

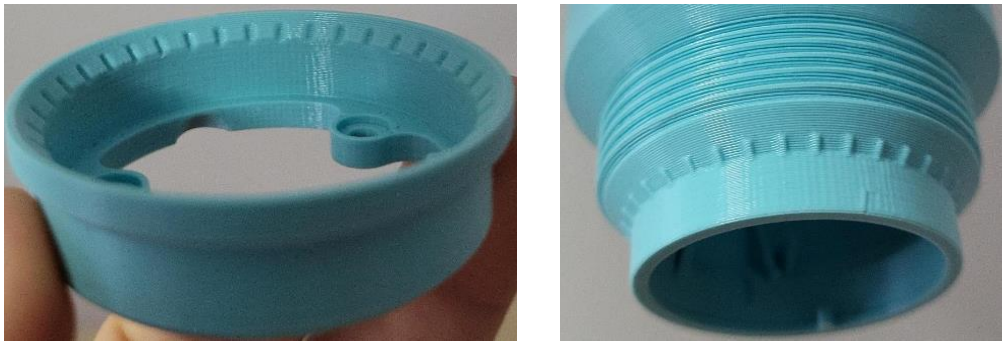


Figure 70. V4 Rotable Mounting Joint

The rear housing is fastened to the inner locking ring with #4 3/8" screws and encloses the ¼-20 tee nut used for mounting and the RJ11 jack used for connecting to the Hub. The RJ11 jack is soldered to the DuPont headers coming from the LPS22HB and carries the I2C signal to the Hub through an RJ11 phone cable.

Enclosure Specifications

The threads interfacing between the front cap and joystick housing are M58x2 and the threads interfacing between the outer locking ring and the joystick housing are M52x2.

The opening on the front cap is 25 mm in diameter with a 1.2 mm chamfer, expanding the diameter to 28.4 mm on the very front face. The front face of the cap in inline with the front of the inner gimbal, and the interior ridge of the front cap is used to hold the gimbal bearings in place.

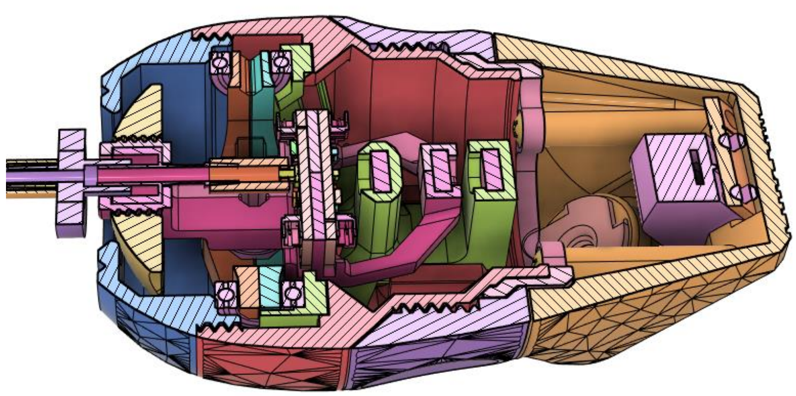


Figure 71. V4 Joystick Horizontal Cross Section

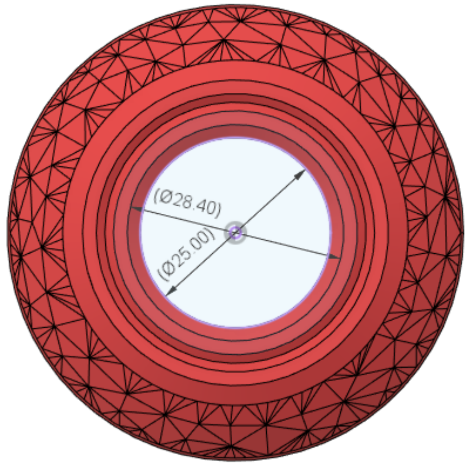
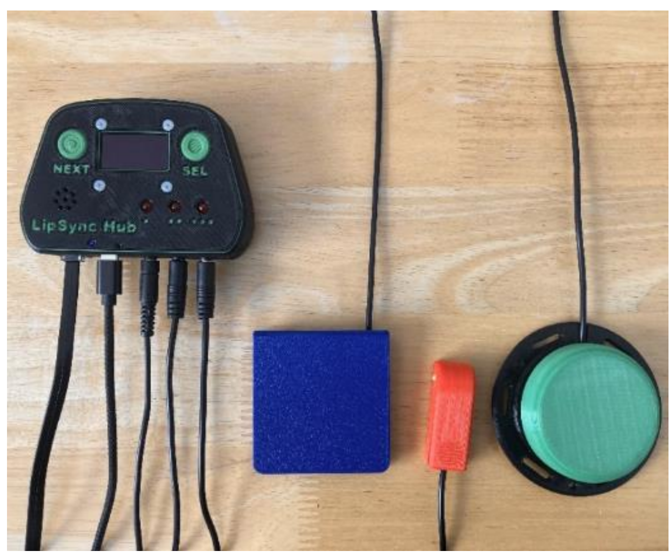


Figure 72. V4 Joystick Front Opening

Hub

The LipSync Hub is comprised of a 3D printed enclosure that contains a custom PCB, a display, and a mounting interface. The custom PCB contains the microcontroller, LEDs for visual feedback, a speaker for audible feedback, and the connection ports for the LipSync Joystick, the Host Device, and external assistive switches.



Hub Features

Hub Display: The Hub Display allows the users to make quick adjustments to device settings, calibrate the joystick, and change the operating mode. It also displays mode and version information upon start-up.

Next & Select Buttons: The Next and Select Buttons on the Hub allows a support person to access the Hub Menu without the need to use sip and puff, or the assistive switch ports.

Feedback Speaker: The Feedback Speaker gives auditory feedback through beeping tones. The sound can be turned on or off through the Hub Menu.

Feedback Lights: The Feedback Lights give visual feedback through flashing on input durations. Each light represents a specific set of inputs.

Operating Mode Indicator: The Operating Mode Indicator is used to show whether the device is in USB Mouse, Wireless Mouse, or USB Gamepad mode.

Microcontroller Reset: The Microcontroller Reset hole allows access to the reset pin on the microcontroller using a paperclip or similar sized object.

Mounting Connection: The Hub can be mounted in view beside the host device screen so that it is easy to see/hear the feedback and read the Menu Screen when adjusting settings. The ¼-20 threads are the same as a standard camera mount.

Joystick Port: The LipSync Joystick plugs into this port using the LipSync Interface Cable.

USB Port: This port provides the device with power and can also be the direct connection to the host device.

Assistive Switch Ports: Up to 3 assistive switches with 3.5 mm audio plugs can be connected to the LipSync Hub. By default, these external switches can be used in place of the Sip and Puff controls.

Hub Main Components

The LipSync Hub consists of a mountable 3D printed enclosure that contains a display and a custom PCB with tactile buttons, a feedback speaker, and connection ports.

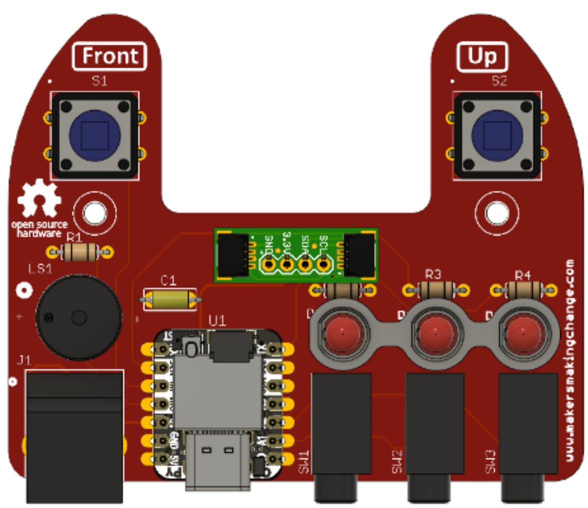


Figure 73. V4 PCB

For more details, links, and quantities see the Bill of Materials.

Hub Enclosure

The Hub Enclosure is a two-part 3D printed enclosure secured by three screws. The Hub Enclosure also has two 3D printed button pushers and a spacer for the LEDs on the PCB.

The Hub Enclosure is intended to be mounted vertically, with the display and indicator lights facing the user. The cable connections exit the bottom, and the mounting interface is located on the back.

Hub Mounting Interface

The Hub Mounting Interface consists of a ¼”-20 UNC tee nut that is inserted into the Hub Bottom. A bolt or machine screw can be inserted up to 20 mm into the enclosure. The tee nut is 51.2 mm from the bottom surface of the hub and centered left to right.

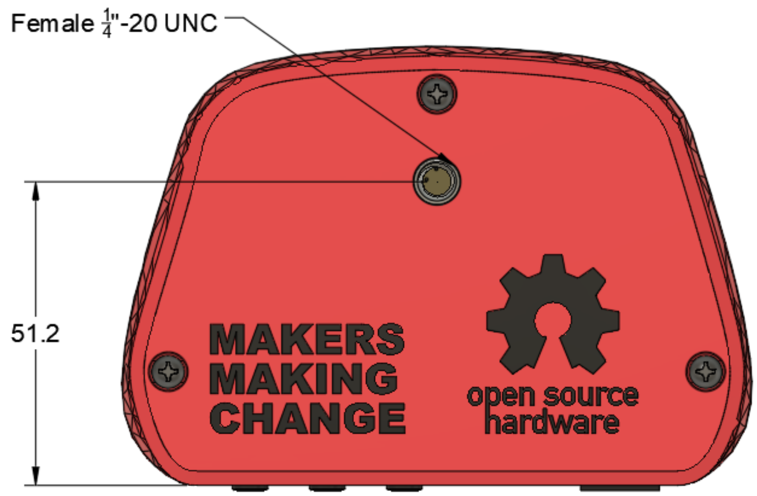


Figure 74. Hub Mounting Interface

A 5/16” tee nut was chosen, as this is one of the more commonly available lengths. Different lengths could be used (7/16”, 9/16”) but they would protrude from the back of the enclosure or require a thicker enclosure. The hole pattern for the tee nut should fit 3, 4, and 6 prong variants.

Hub Display

The Hub Display provides a visual interface for the user. The Hub Display is monochrome 1.3” 128 x 64 OLED graphic display – STEMMA QT / Qwiic from Adafruit. This display has an I2C connection.

Figure 75. Hub Display.

The display has four (4) holes that fit M2.5 mounting screws.

Key features of the [Adafruit Monochrome 1.3" 128x64 OLED graphic display]() include:

- I2C connection with Sparkfun quiic compatible STEMMA QT connectors
- Power: depends on how much of the display is lit but on average the display uses about 40mA from the 3.3V supply
- OLED lifetime: each pixel has a lifetime of ~1000 hours if kept on before it starts to dim, or a storage lifetime of ~20,000 hours (when not on).

Hub PCB

The Hub PCB is a single, 2-layer custom PCB. All components are thru-hole and are soldered to the front of the board. The board has two mounting holes for securing it into place.

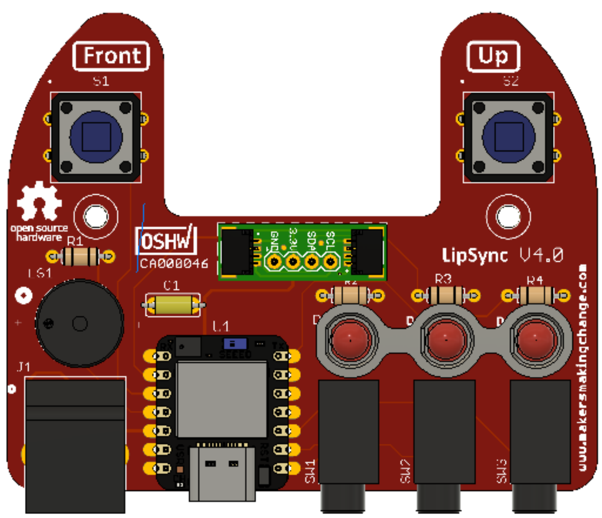


Figure 76. Assembled PCB

PCB Key Components

Microcontroller – the microcontroller is the interface between the other hardware and the host device. This microcontroller is the Seeed Studio Xiao nrf52840.

RJ11 Connector - the RJ11 connector provides an I2C connection to the LipSync Joystick.

3.5 mm Switch Jacks – provide a connection for external assistive switches

Piezo Speaker – provides audible feedback

SparkFun QWIIC Adapter – an off-the-shelf board that provides a convenient way to connect a STEMMA QT cable via a set of headers on the PCB. This allows the display to be disconnected and replaced in the future.

LEDs – provide visual feedback

Tactile Buttons – provide a way for support user to provide input and interact with display menu

For a complete component list along with specific parts, links, and quantities see the Bill of Materials.

Microcontroller

The LipSync Hub utilizes a XAIO nRF52840 microcontroller from Seeed Studio. Some key features of this microcontroller are:

Item	Seeed Studio XIAO nRF52840
Processor	Nordic nRF52840
	ARM® Cortex®-M4 with FPU runs up to 64 MHz
Wireless	Bluetooth 5.0/NFC
On-chip Memory	1 MB flash and 256 kB RAM
Onboard Memory	2 MB QSPI flash
Interface	1xUART, 1xIIC, 1xSPI, 1xNFC, 1xSWD,
	11xGPIO(PWM), 6xADC
Dimensions	21 x 17.5mm
Power	Circuit operating voltage: 3.3V@200mA
	Charging current: 50mA/100mA
	Input voltage (VIN): 5V
	Standby power consumption: <5μA

Table 19: Microcontroller Pin Utilization

PIN ID	Pin Label	Connection
0	A0	Assistive Switch 1
1	A1	Assistive Switch 2
2	A2	Assistive Switch 3

PIN ID	Pin Label	Connection
3	A3	Button 2
4	SDA	SDA Serial Data
5	SCL	SCL Or Serial Clock
6	TX	LED 3
7	RX	LED 1
8	SCK	LED 2
9	MI	Button 1
10	MO	Buzzer
11	3.3V	Power
12	GND	Ground
13	5V	5 V

LipSync Joystick Connection – RJ11

An RJ11 connector provides the connection to the LipSync joystick via the LipSync Interface cable. The RJ11 connector is connected using i2C in the scheme indicated below:

RJ 11 Breakout	STEMMA QT Cable
2 (B)	Black (Gnd)
3 (R)	Red (Vcc)
4 (G)	Blue (SDA)
5 (Y)	Yellow (SCL)

This wiring scheme was chosen so that the order matches the pinout on other I2C adapter boards.

The RJ11 connector was chosen to match the footprint of the SparkFun RJ11 Breakout board.

Feedback Speaker

A piezo speaker provides simple audible feedback. The piezo speaker can generate tones in a range of frequencies. It is currently connected directly to the microcontroller without any additional driving circuitry, so the volume is not adjustable.

Visual Indicators - LEDs

The LipSync Hub contains three (3) 5 mm red LEDs that are arranged in a horizontal layout that roughly aligns with the three switch input ports.

The LEDs are mounted using a 3D printed spacer to bring the height of the LEDs to be flush with the front face of the enclosure. The spacer has a flat side to provide a visual clue on the correct orientation of the LED during installation.

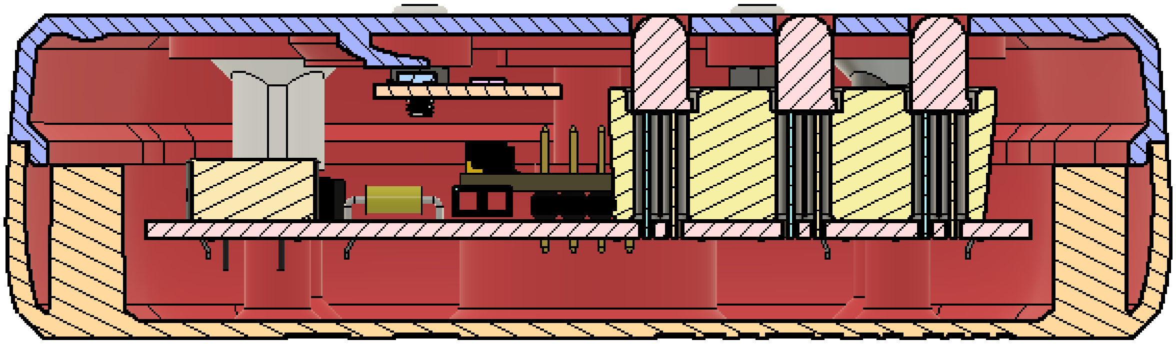


Figure 77. LED Spacer Cross Section

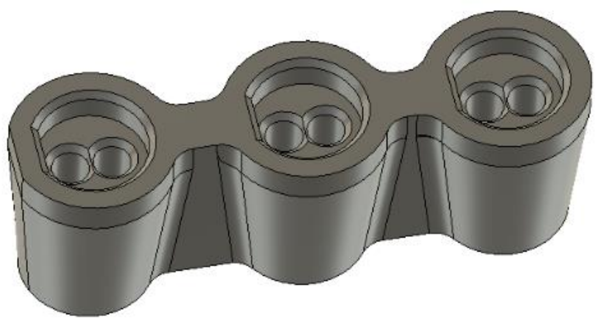


Figure 78. LED Spacer

The LEDs each use a current limiting resistor of 68 ohms.

Tactile Buttons

The LipSync Hub has two 12 mm tactile buttons. These buttons provide a way for a secondary user to provide input to the LipSync and to navigate the Hub Menu.

The tactile buttons are Omron model B3F5050. This model was selected because it has a low activation force and is used in a variety of other MMC OpenAT designs (e.g., Raindrop switch).

Firmware

Firmware Main Features

- 3 Operating Modes: USB Mouse, Wireless Bluetooth Mouse, USB Gamepad
- Inputs can be entered using either sip and puff, assistive switches, or the Hub buttons
- Menu on OLED Screen to adjust key settings, that can be navigated by user, including:
  - Calibration
  - Operating mode
  - Cursor speed
  - Sound on/off
  - Sip and puff threshold
- Neutral position calibration on startup
- LEDs for feedback
- Memory to remember user adjustable settings

Compatibility

Operating System		Compatibility per Mode		
		USB Mouse	Wireless Mouse	USB Gamepad
Phone/Tablet	Android	✓ [1]	✓	✓ [1]
	iOS[2]	✓	✓	
	iPadOS	✓	✓	
	Windows	✓	✓	✓
Desktop	macOS	✓	✓	
	Linux	✓	✓	
	Windows	✓	✓	✓

[1] Only with devices with USB OTG HID (Universal Serial Bus On-The-Go Human Interfere Device)
[2] Only for iOS13+ and iPadOS.

Inputs and Outputs

Input			Input Duration	Hub Menu	Mouse	Gamepad
Sip and Puff	Assistive Switches	Hub Buttons				
Puff	S1	Select	< 1 sec	Select	Left click	Button 1
			1-3 sec	-	Start Drag Mode	Button 3
			> 3 sec	Exit Hub Menu	Enter Hub Menu	Enter Hub Menu
-	S2	-	< 1 sec	-	Middle Click	Button 5
			1-3 sec	-	N/A	Button 6

Input			Input Duration	Hub Menu	Mouse	Gamepad
Sip and Puff	Assistive Switches	Hub Buttons				
			> 3 sec	-	Perform Center Reset	Perform Center Reset
Sip	S3	Next	< 1 sec	Next	Right click	Button 2
			1-3 sec	-	Start Scroll Mode	Button 4
			> 3 sec	-	Middle Click	Button 5
-	S1 + S3	Next + Select	< 1 sec	Exit Hub Menu	Enter Hub Menu	Enter Hub Menu
Joystick			-	-	Cursor movement	Joystick movement

Menu Options

- Exit Menu
- Calibrate
    - Center reset
    - Full calibration
- Change mode
    - Mouse USB
    - Mouse BT
    - Gamepad
- Mouse cursor speed
    - (show current cursor speed)
    - Increase
    - Decrease
- More
    - Sound on/off
    - Sip and Puff threshold adjustment
        - Sip threshold
        - Puff threshold
    - Restart device
    - Factory Reset

Table 20. Hub Menu Settings

Setting	Default	Min	Max	Note
Cursor Speed	5	0	10	Changes the cursor speed in any Mouse Mode.
Sip Threshold	3.0	1.0	100.0	Changes the pressure required for a Sip input.
Puff Threshold	3.0	1.0	100.0	Changes the pressure required for a Puff input
Sound Feedback	On	Off	On	Turns the sound on or off.
Scroll Level	5	0	10	Changes the scrolling speed.

Setting	Default	Min	Max	Note
				Only adjustable through API, not Hub menu.
Deadzone	0.12	0.0	1.0	Changes the distance the joystick must be moved from rest to cause cursor/gamepad movement. **Only adjustable through API, not Hub menu.**

Input Feedback Features

Each Feedback Light is designated to a set of inputs:

Left Light •: Puff, S1
Middle Light ••: S2
Right Light •••: Sip, S3

The time durations Short, Long, and Very Long for the Sip/Puff and switch/button inputs are the following:

Short: Less than 1 second. Feedback Light blinks once when released.
Long: Between 1 to 3 seconds. Feedback Light turns on and stays on.
Very Long: 3 seconds or longer. Feedback Light turns back off.

V4.0 Firmware Features

- Joystick Functionality
  - Adjustable deadzone via API
  - Neutral Position Reset
  - Extents Calibration
  - Linear input/output response
- Sip and Puff Functionality
  - Short / long / long press
- Wired Mouse Mode
  - Joystick cursor control
  - Sip/puff short/long/very long maps to different buttons and functions
  - Adjustable cursor speed
- Wireless Mouse Mode
  - Joystick cursor control
  - Sip/puff short/long/very long maps to different buttons and functions
  - Adjustable cursor speed
- Wired Gamepad Mode
  - Joystick Thumbstick control
  - Sip/puff short/long/very long buttons and functions
- Display
  - Ability to enter / leave menu mode
  - Splash screen

        - Appears on startup
        - Once center calibration is complete, new screen showing device is ready to use and the mode appears
        - Disappears after set time
- Reset
    - Restart device and Factory Reset are options in menu
- Sound Feedback
- Visual Feedback
- Serial-based API
    - Get/Set operating mode
    - (neutral calibration) get/setJoystickInitialization
    - (Full calibration)
    - Get / set cursor speed
    - Get / set sip /puff threshold
    - Get/set deadzone level
    - Get/set sound mode

V4.0 Internal Testing

V4.0.rc1 Firmware Feedback

- Gamepad mode lights aren't giving the correct feedback (Seems random)
- Indicator LED not turning yellow in gamepad mode
- Onboard LED blinks yellow in menu for any input
- Onboard LED blink when waiting to pair in Bluetooth Mode
    - Remains solid blue for a while
    - Does start blinking eventually, but blinks over a solid purple light and switch inputs can stop the blinking for a bit
- Scroll mode
    - very slow on iOS and Mac
    - Ok on Android and Windows
    - Tested iOS and Android via BT, Windows via USB
- Potentially something weird going on between scroll speed and cursor speed

V4.0 OFIs

Joystick CAD

1.1. Update model to use proper thin wall widths
1.2. Create alternate front cap hole outline
    1.2.1. Octagon, reduced range, square/diamond, cardinal directions, etc.
1.3. Add parameters for specific clearance values in the CAD model.

1.3.1. Bearing_Clearance, Luer_Clearance, Nut_Clearance, Sensor_Board_Clearance, RJ11_Clearance, etc.

1.4. Get feedback on the rear mounting angle and adjust if required.

1.5. Add interface on Joystick enclosure for mounting a lip/chin switch and/or microphone for voice recognition

1.6. A number of assembly steps include risk of pinching the STEMMA cables. Rework component design or build instructions to mitigate.

Hub CAD

1.7. Improved display mounting (e.g., Counterbore screen connections, add frame)

1.8. Swap PCB mounting posts to top?

1.9. Add version number to enclosure

1.10. Reduce height of USB Connection? (e.g., Shorter headers, Direct solder)

1.11. Increase Light pipe height tolerance

Firmware

1.12. Mouse Functionality

1.12.1. Add Joystick position based acceleration

1.13. Menu

1.13.1. Ability to disable sip/puff (for someone using switches) in menu

1.13.2. Ability to toggle LEDs on/off

1.13.3. Add scroll speed

1.13.3.1. (Need faster Scroll on Bluetooth / iOS / Mac)

1.14. Wired Gamepad Mode: Button Hold then Toggle

1.15. Add error handling

1.15.1. If joystick is not plugged in, add screen to tell user to plug in

1.15.2. And/or remove code that blocks code from executing if joystick is not plugged in

1.16. Add middle click and hold option as an output (Useful in CAD and some graphics programs)

1.16.1. S2 long press maybe?

1.17. On “risky” confirm pages (like mode change, factory reset, etc) make the default selection “Back” instead of “Next”

1.18. Add ability to recognize sip and puff sequences. E.g., Integra Mouse mode change: 2 short puffs; <2 second delay; 2 short sips; <1 second delay; 1 short puff

1.19. Diagnose device naming USB Mouse mode on Mobile (iOS and Android) – showing up as “XIAO nRF52840” instead of LipSync.

1.20. BUG: does not allow you to enter menu mode when in USB mouse mode and connected direct to power (plugged into wall, not into a device). So you need to connect to a device to change to Bluetooth mode before plugging in directly to power

1.21. User adjustable input durations (i.e., Time required for long and very long inputs)

1.22. Add scroll speed adjustment in menu, currently only in API)
1.23. Separate classes for low-level joystick and mouse / gamepad stuff
1.24. Create class for LipSync device
1.25. Implement Sip and puff sequence recognition / Morse Code
1.26. Add Sound Feedback
1.27. Joystick Functionality
    1.27.1. Add adjustable response / sensitivity curve
    1.27.2. Explore utility of ability to disable menu activation
    1.27.3. Easier remapping of actions
1.28. Mouse Mode
    1.28.1. Add Time-based cursor acceleration
1.29. Gamepad Mode
    1.29.1. Add ability to create separate functions for Sip and Puff vs. switches
1.30. LSJoystick.h
    1.30.1. Separate out mouse mapping stuff
1.31. Implement Full calibration wizard / helper
1.32. Add walkthrough/prompts during calibration process
1.33. Add real-time display output of Sip / puff / cursor
1.34. Diagnose potential conflict between cursor speed and scroll speed settings

V4.1 Update

V4.1 Firmware Changes

- TinyUSB Library bug identified, now works with latest version (3.4.2) of the library
- Full calibration changes
    - Moved to "More" menu to prevent Full Calibration from being selected when not necessary
    - Added warning message and confirmation page
    - Added error checking, if joystick was not moved enough during full calibration, use default value instead
- No USB auto mode switch fix
    - 4.0.1 added a feature that if no USB was detected the LipSync would automatically switch to Bluetooth mode, but this caused unintentional effects
    - Now, if no USB is detected, the screen displays an error message and buzzer plays an error tone, but continues to try to connect in the background
    - To change modes, the menu can now be used even if no USB is mounted.
- User Feedback
    - Light Brightness can now be adjusted through Hub menu or API
    - Added additional sounds for startup, powering off, and errors
    - Added information screen to menu
- Error Handling
    - I2C devices are checked on startup and a corresponding error message is displayed
    - Added watchdog implementation
- Added safe mode accessible by Hub Buttons, or triggered during hardware error or watchdog
- Scroll speed can now be adjusted through Hub menu
- Scroll is now calculated differently, at a higher poll rate but not performed every loop, depending on the scroll speed
- Fixed bug where scroll speed was dependent on cursor speed
- Fixed bug in timers, so timer offset now works properly
- Deadzone improved and separated
    - Deadzone was separated into inner deadzone and outer deadzone
    - Output now starts at 0% at the end of the inner deadzone, and goes to 100% at the edge of the outer deadzone

V4.1 Hardware Changes

- Ribs were added to the screw bosses on the LipSync Hub Top and LipSync Hub Bottom to increase strength
- LipSync Mouthpiece was simplified

Updated Mouthpiece

The updated mouthpiece is comprised of the same hydrophobic syringe filter, a Luer adapter, and a large silicone straw. This simplified design is easier to assemble, easier to clean, and has the added benefit of reducing the overall cost.

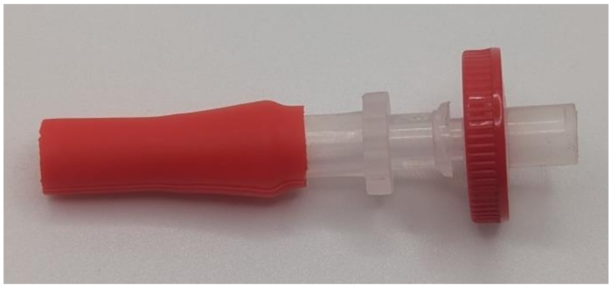


Figure 79. V4.0 Mouthpiece (top) V4.1 Mouthpiece (bottom)

V4.1 Documentation Changes

- Maker Guide
    - TinyUSB library - can now use latest version (3.4.2)
    - Added additional assembly step to test joystick components before assembly into Joystick Gimbal and housing
- User Guide
    - Updated to reflect changes to menu structure and full calibration process
    - Expanded troubleshooting with error messages
    - Updated menu to represent changes
- Created LipSync Firmware Upgrade Guide
- Created LipSync Information Sheet
-

Appendix

Material Properties

Table 21.: PLA Material Properties

Property	Value	Notes	Source
Tensile Modulus	2.3 GPa		
Ultimate Tensile Strength	26.4 MPa		
Tensile Strength @ Yield	35.9 MPa		
Elongation at Yield	2%		
Elongation at Break	4%		
Hardness	95D		
Solid Density	1.24 g/cc		

<Source: http://www.sd3d.com/materials>

Low Poly Surface Texturing Method

The LipSync uses a "low-poly" or low polygon texture on the outside surfaces of the housing. This texture helps to reduce the appearance of the layer lines and seam lines and adds an interesting aesthetic look.

Processed STL

In earlier LipSync designs, a low poly surface texturing was applied after the 3D model had been converted into an STL. First, the STL was imported into a mesh program (Meshmixer). Next, the vertices of the surface to apply the texture upon were manually selected. Then, the Reduce function was used to reduce the number of triangles. Lastly, the modified mesh was exported as an STL.

Fusion 360

Relatively recent updates to Fusion 360 have provided a way to do the low poly surface texturing completely with Fusion. This workflow works best with surfaces that have curvature in two directions. It is harder to apply this texture to flat or rotated surfaces.

1. Create the solid geometry.
2. Split the desired faces using the Split Body tool to offset any faces that need a clean, undisturbed edge, like the transition between two circular parts that rotate relative to one another.
3. In the surfacing environment, use the Offset Tool to create a zero-distance offset of the surface to be textured. You may need to uncheck the Chain Selection option to ensure that only the desired surface is offset.
4. In the surface environment, delete the original face.
5. In the mesh environment, select the new offset surface and use the Tesselate tool to convert the surface to a triangular mesh.
6. In the mesh environment, use the Remesh tool to change the look of the mesh. Setting the Type to Adaptive, and turn on the Preserve sharp Edges and Preserve Boundaries options. Adjust the density and shape preservation to achieve the desired look.
7. In the mesh envrironment, use the Convert Mesh tool to convert the triangular mesh back to a surface.
8. In the surface environment, select the newly converted surface body and the rest of the surface bodies, then use the Stich tool to recombine the surface into a solid body.

Test Methods

General Functionality

PCB Suitability

These tests will be completed to see if there are any changes required to the PCB. These tests should be completed first to allow for ordering and delivery lead times of the PCBs.

LED Position/Colour/Number

To test the position and colour of the LEDs, the hub should be placed in multiple positions and distances from a user. The test will check to see the LEDs are visible and distinguishable from different angles and positions. The following conditions should be tested:

- Test line of sight: (different mounting positions)
    - Directly in front of user in line of sight
    - Upper/lower/sides peripheral vision
        - +/- 30 degrees from focal center (need citation for this but think it's in Ergo)
- Angle of hub (different mounting positions)
    - Flat
    - Tilted forward/backward up to 45 degrees
    - Tilted left/right up to 45 degrees
- Distance to hub (different mounting positions)
    - 0.5 m to 2 m (2 m the length of the cable we have in the BOM currently, 0.5 m is arbitrarily close)
- Lighting conditions (use in different spaces)
    - Test in a dark room
    - Test in a fully lit room
    - Test with sunlight/outside (weather permitting)
        - Can we get PWM control of brightness of LEDs?

These tests should be conducted in combination. For example, all lines of sight would be tested at each distance and angle of the hub and should be tested with each LED.

Software required for testing this will simply require the ability to turn the LEDs on and off. This could be from the main software and have the tester switch between cursor and joystick modes, or a simple custom code that only controls the LEDs.

Questions:

1. LED Spacing
    a. Can the two LEDs be differentiated at the different distances, angles, and lighting conditions?

2. LED Number (e.g., additional LED for wireless?)
    a. Do we need to have a third LED to show any other states?
3. Hole / light pipe for microcontroller User LED?
    a. Can the User LED or TX/RX LED be used for these?
4. Hole / light pipe for microcontroller TX / RX LED?

Buzzer

The buzzer needs to be tested for loudness and size. The buzzer is not meant to provide detailed auditory cues, such as speech or music, but a simple buzz/tone. To test this, the same distances and angles as the previous LED tests should be used. The current circuitry does not allow the buzzer volume to change unless a different resistor is used to change the current through the buzzer. The buzzer volume should be tested in the following setups:

- Environmental noise
    - Test in quiet room (no sound), medium loudness room (office space type of environment, a few people talking or music playing at a reasonable level), and loud room (large number of people speaking loudly, music playing loudly, etc.)

The software required for this testing will only need to cause the buzzer to buzz at the different loudness levels.

Question:

1. Do we need physical / digital potentiometer?
    a. How well can we hear the buzzer as is?
2. Do we remove the buzzer all together?
    a. Based on user feedback?

Number of Buttons

We need to test if the hub needs extra dedicated switches for navigating the menu, changing modes, or other functions a secondary user may perform with the device. To test this, we will need to be able to access the menus in the screen and navigate between them. A secondary user should be able to enter the menu selection mode, navigate the menus, and select the desired outcome in under one minute using only the two buttons. If they are unable to do this successfully, we should add a third button to increase the number of inputs for the secondary user. Additionally, we will have to consider if we want a third, dedicated button for the most commonly used function for a secondary user (such as entering calibration or switching operation mode).

The code required to test the number of switches will need to read and interpret the button presses and include a skeleton of different modes (mouse and joystick mode), and the menus/navigation of the menus.

Dedicated Reset Button

We will need to test how easy it would be for someone to reset the Hub in the current configuration to see if there is a need for a dedicated reset button. If the reset cannot be completed within 30 seconds, or requires the secondary user to open the enclosure, we will need to add a dedicated reset button.

Testing the need for a dedicated reset button does not require any specific code. If there is a regular need to reset the microcontroller, or need for precise timing when resetting it (like with the QT PY) then a dedicated reset button will be required.

Screen Suitability

The following tests will be used to assess the suitability of the selected screen. The screen will be assessed for legibility at different distances, angles from the user, and lighting conditions. The tests will follow similar conditions as the LED testing; however, the tests of having the hub in the periphery of a user's vision will not be included as it is expected a user will focus on the screen to read it. Under each of the following conditions, the tester will ensure they can read the screen without difficulty.

- Angle of hub (different mounting positions)
  - Flat
  - Tilted forward/backward up to 45 degrees
  - Tilted left/right up to 45 degrees
- Distance to hub (different mounting positions)
  - 0.5 m to 2 m (2 m the length of the cable we have in the BOM currently, 0.5 m is arbitrarily close)
- Lighting conditions (use in different spaces)
  - Test in a dark room
  - Test in a fully lit room
  - Test with sunlight/outside (weather permitting)

The firmware required for these tests of legibility simply needs to display lines of text on the screen at the desired font size (2x).

Disconnecting and Reconnecting

Test what happens when you disconnect and reconnect the mouthpiece/joystick module while the Hub is on. Test this in both joystick and mouse mode to see if there is different behaviour. There is no specific code to test this with, just code that runs the mouse/joystick software.

- General functionality (sip/puff, detect boards are connected)
  - Sip/puff works
  - Boards are connected
  - Basic joystick motion
- External switch jacks

Firmware Testing

The firmware testing is to ensure the firmware behaves as intended and to find any bugs that require attention.

Mode Switching

Switch modes between mouse and joystick. Test that the mouthpiece/joystick function in those modes using a computer and/or an XAC as needed. The LED to indicate the modes should switch as indicated to the correct mode. This test will be successful if you can switch between modes correctly and in under one minute.

Navigating Menus

Test that the menu navigation works properly. Navigate through each menu using the buttons on the enclosure. Each menu should open, display the correct information, and behave as it should. For example, the calibration menu should initiate calibration.

The code required for this test is the regular LipSync firmware that a user would have flashed to the device.

The navigating menus test will be successful if all menus are active and show the expected behavior when a menu item is selected.

Check sensors are connected and on

The firmware should test that the joystick mechanism is connected and all sensors are connected/on.

Mouthpiece Testing

The testing of the mouthpiece involves testing the joystick mechanism and sip/puff functionality.

Extents and Neutral Testing

The extents and neutral testing will be the same as the previous extents and neutral testing, but with updates for new wiring/buttons.

Sip/Puff Functions

Test the sip and puff functions on the device. Sib and puff on the device to test the right/left click functions. Also test the short, medium, and long sip/puff functions to ensure the correct behaviour.

The code required to test this functionality is the regular LipSync firmware a user would have flashed to the device.

The sip/puff function test will be successful if the expected behaviour occurs with the sips/puffs.

Weight Testing

Record the total mass of the assembled mouthpiece/joystick. This will be used as a benchmark for potential improvements.

Rubbing Interference

Check for rubbing/interference for the movement of the joystick. Listen and feel for any changes in friction/sticking at any point while operating.

Hardware Testing

This hardware testing section is to test the different input options on the Hub. The test will go through each input on the Hub (integrated and external switches). The hardware testing will require the regular LipSync firmware that a user would have flashed to the device.

Joystick Movement

- Sensors connected/on (general code testing
- Test buttons
    - Connected
    - Scanning properly
- Testing screen
    - Testing readability
    - Test menu navigation
        - Sip/puff with joystick
        - With buttons
- Testing joystick movement
    - Part of calibration
- Testing sip and puff
- Testing external switches
    - Each jack and their input/output
- Testing centering
    - Individual testing
    - Testing rig testing
- Same testing as long term stuff previously
- How we access settings menu
    - User access
    - Caregiver access
- How we navigate settings menu
- Does the screen fall asleep? How to reactivate it?
    - Does it stay at a general info screen? (cursor speed, mode, etc.)
- Bluetooth
- Mouse options (left click/right click)
- Changing between joystick and mouse modes
- Dead zone and drift testing
- What happens when something is detached?
    - How do we sense everything is still connected?

- What happens when you reconnect?

Joystick Test – PLA Creep

This test is to determine if the attraction between the embedded magnets imposes enough constant stress to cause creep to occur in the PLA and bend the two magnet slots/posts towards each other.

Method

The test will measure the inner distance between the magnet posts and the values that the sensor is reading at the neutral position. The Data Capture code will be used to capture the data. The test will be repeated after 2 weeks' time to see if the measurement or sensor values have changed noticeably.

Measurements

1. Type in the date you are taking the measurement.
2. With the gimbal assembly sitting neutrally, take one measurement using the data capture code and copy/paste the results (**From the headings down to the end of the data, so start your highlight at "Trial" and copy everything to the right and below)** into the Excel document under the sheet with your name.
3. Using calipers, measure the distance at the very top of the two magnet posts on the sled. Note, it might be easier to remove the sled to measure this gap. Type your results into the Excel document under the sheet with your name.
4. Measure the distance between the two magnet posts at the bottom of the flat section of the sensing magnet post. Type your results into the Excel document under the sheet with your name.
5. Reassemble your gimbal/full joystick unit and repeat steps 1-3 in 2 weeks.

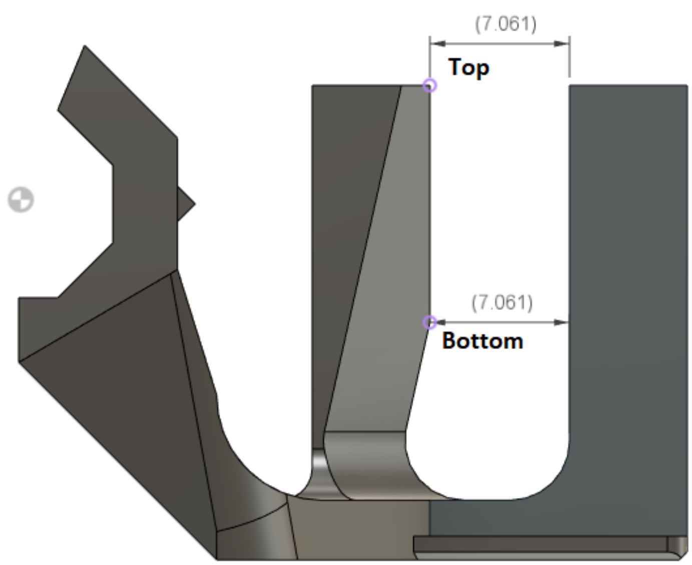


Figure 80. Sled Geometry

Data

PLA Creep Test.xlsx

Analysis

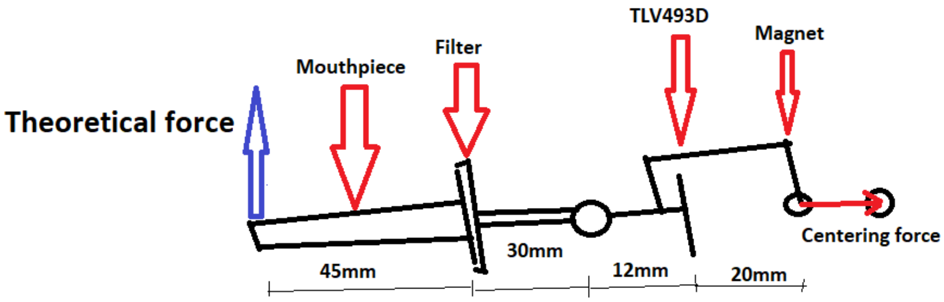
Looking at the measured distances, we can see if the two magnets have pulled the posts closer together. Looking at the magnetic Z value, we can see it the sensing magnet post has been pulled back away from the sensor.

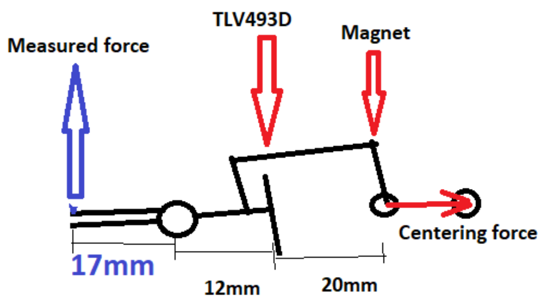
Results

It was found that there was some creep occurring from the magnetic forces between the magnets in the Beta II sled. In the final design, the sled was designed to have stronger walls to resist this effect.

Force Testing – 3D Printer

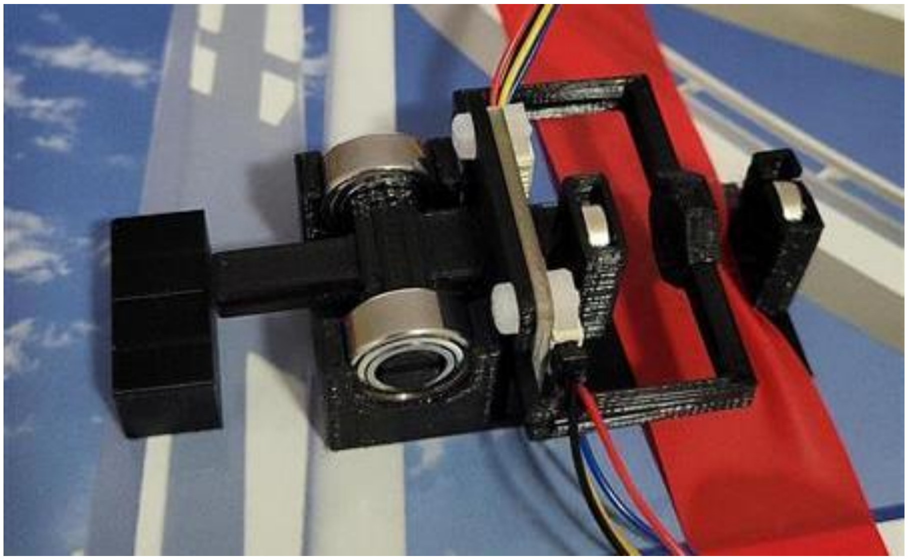
Setup: Testing unit attached to hot end of a 3D printer and a kitchen scale placed on the print bed. An empty spool was used to depress only the end of the test unit "Mouthpiece" lever. The scale was tared to start and the vertical axis of the printer moved manually using the dial. This moves the hot end vertically in 1 mm increments. The force was recorded over 10 steps for various set-ups. The test was repeated thrice per set-up and the mean value was plotted. Further testing with a more finalised set-up would be required to determine the force at the end of a mouthpiece when a Qosina mouthpiece and filter are attached. This current arrangement also does not take into account the pressure sensor on the gimbal.





Magnetic Sensor Testing

Testing concept 3 configuration in only the vertical axis.



The above unit was connected to the Alpha II PCBs and tested using the LipSync_DataCapture.ino code. The joystick was constrained to vertical movement only and the deflection angle is controlled with a stepped 3D printed block. The height difference in each step was 0.3 mm, determined through the CAD model to move the joystick 1 degree with each step. The lever arm distance is 12.5 mm and the air gap between the sensor and magnet is 4 mm.

Lever arm distance is defined as the distance from the center of rotation of the gimble to the face of the magnetic sensor chip on the TLV493D breakout board. The air gap distance is defined as the distance between the face of the magnet and the face of the magnetic sensor chip on the TLV493D breakout board.

Testing Process

Setup

1. Print and assemble test setup.
2. Secure test set-up on a flat, level surface.
    a. Taped down to a textbook in this case.
3. Plug the STEMMA-DuPont cables into the Alpha II PCBs.
4. Upload the testing code.
5. Plug an assistive switch into port 2 on the Alpha II

Collecting Data

1. Connect the Alpha II boards to your computer via the microcontroller USB cable.
2. Press the switch to take a reading of the neutral position.

3. Using a block to move the joystick a specific distance, move the joystick to the first increment and press the switch to take another reading.
4. Release the joystick back to neutral and take another reading.
5. Repeat steps 3 and 4, each time using a new increment for step 3, until you have gone through all increments of joystick movement.

Sip And Puff Maker Testing

Currently, a spare mouthpiece is used to test the pressure system. An ear syringe can be used to apply sips and puffs to the mouthpiece as an optional testing component