Anti-Nail-Biting Device

Daniel Dreise, Justin Turcotte, Brandon Harkness

Advanced Tech Elective

ESE

Conestoga College

August 2, 2021

Abstract

Chronic nail-biting is an exceptionally difficult habit to break. However, very little support is given for those suffering from the addiction. Of the supports available, many are misinformed and/or not useful. Therapy is one option for some, but for many it is financially unfeasible, and again is not able to help during a person's regular everyday activities. Therefore, this project aims at designing a wearable device that will help deter the user from biting their nails. The device is worn around the neck like a necklace, with an ultrasonic sensor placed near where the collarbones meet. The sensor will detect when the user brings their hand close to their mouth. Once a hand is detected, it will trigger an actuator (buzzer) on the back of the neck that will notify and deter the user from biting their nails.

Keywords: nail-biting, therapy, deterrence, ultrasonic sensor, actuator

Table of Contents

Introduction

The project will be a wearable device that dissuades the user from biting their nails by giving real-time haptic feedback to the user when their hand gets close to their mouth. The purpose of this document is to provide a comprehensive report of the designed anti-nail-biting solution.

Problem Description

Personal experience has proven that trying to stop the chronic habit of biting nails is near impossible. When people think of addictions, substance addictions (i.e., alcohol, drugs, etc.) are what come to mind. However, through previous personal academic education I have learned that the chronic biting of nails, being a habitual addiction, can be just as much or more difficult to break than substance addictions. Chronically biting one's nails has a negative impact on an individual’s well-being for several reasons:

- Unsightly biting nails in public. Resulting in feelings of awkwardness or social inferiority.
- Gives others false indication that you are nervous, anxious, or stressed
- Biting too much can cause severe pain on fingertips, disabling individual from performing normal daily tasks
- Saliva dries out fingertips to the point that they crack and bleed frequently
- Causes mental blocks when biting nails
- Surprising number of actions cannot be done by individual due to lack of fingernails (scratching, peeling tape, peeling fruit, opening key rings, removing SD or SIM cards, opening battery compartments, etc.).
- Can lead to cuticle infection
- Can lead to hangnails that get caught frequently on clothing and get ripped off, leading to pain and bleeding.
- Leaves fingers feeling tender most of the time
- Can damage and misalign teeth

Permanent damage to nail-bed tissue that makes nails

System Description

The project will consist of three main components. The first component is a wearable proximity sensor that will sense when your hand comes close to your mouth. The second a haptic feedback device that will alert when the proximity sensor detects. The third is for data processing & wireless connectivity for recording and storing historical data.

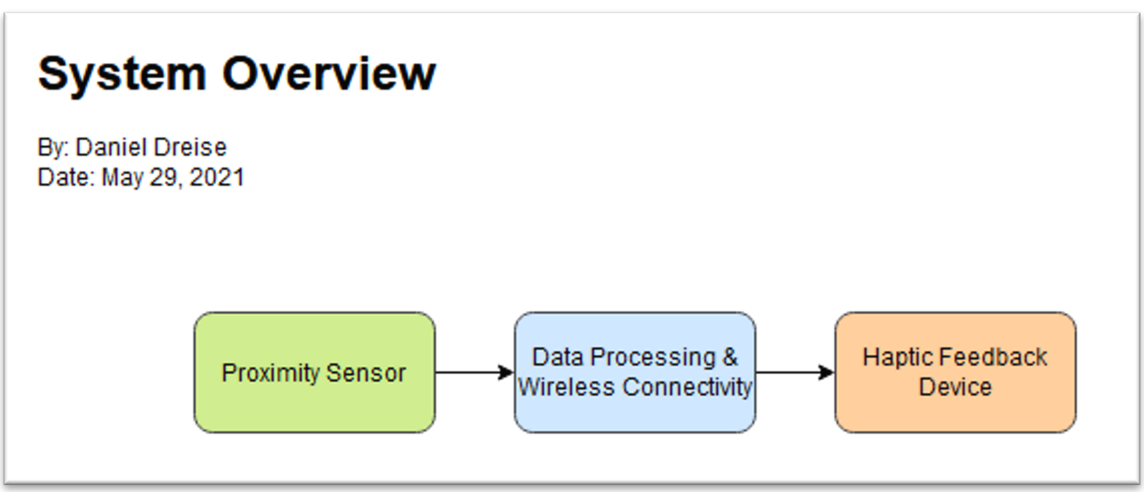


FIGURE 1: BASIC SYSTEM OVERVIEW

Existing Solutions vs Proposed Solution (20 pts)

From our research, there are not many solutions on the market that are taking the same approach as we plan to. Most common solutions are the ones listed above such as keeping nails trimmed, bad-tasting nail polish, or wearing gloves. Some solutions we found which are wearable devices are a shock bracelet called Pavlok and a vibration bracelet called Keen.

Pavlok

Pavlok is a wearable bracelet device that uses aversion therapy as a solution for nail biting. Aversion therapy is pavlovian conditioning; the user begins to associate the habit they want to stop with a negative stimulus [1]. For Pavlok, this is done in the form of electrical stimulus. The device costs $29 USD and includes the following hardware and software:

Hardware

- Wearable bracelet device (silicone)
- Module with rechargeable battery and Bluetooth
- Micro USB charging cable

Software

- App - downloadable on any smartphone
- 5-Day audio habit change course
- Remote control of Pavlok
- Set automated shock options
    - Alarm clock
        - Vibration or electrical impulses
        - SensorCore™ technology to know when you are awake

Pavlok has the following features:

- Activity/sleep tracking - An accelerometer tracks your steps, activity, and sleeping patterns.
- Real-time haptic feedback - Vibration sound, and LEDs act as behavioral triggers and notifiers.
- Human detection - Knows when you're wearing it.
- Hand Detection - When you bite your nails or touch your face.
- Low energy Bluetooth 4.0 - Converse battery and connects seamlessly.
- Snap circuit - Delivers up to 340 V of electric current.

There have been a couple versions of the Pavlok bracelet since its first production launch.

Pavlok 2 [2]

- Automatically track gestures, actions, and behaviors you do throughout the day
- 100x faster than original Pavlok processor (Nordic NRF52 Arm Core Processor)
- Able to learn and detect new gestures and motions (SensorCore® technology)
- Significantly improved connectivity with a redesigned Bluetooth antenna
- 75% thinner band for ease of movement (stainless steel)
- Fully adjustable for any fit and easy on-off (magnetic closure)
- More breathable band (Smooth mesh)
- Step counter
- Machine learning gesture detection (nail biting, smoking, chewing/eating, hair picking / trichotillomania, scratching, "tells" / touching your face, your own programmable gestures)
- App is now a platform

- Improved Pavlok 1 alarm clock and sleep features

Pavlok 3 [3]

- $49 USD
- Core functionalities: Maintain mindfulness, retain focus, break bad habits, wake up easily
- Battery life = 7+ days
- Water resistant
- 3 programmable buttons
- Can be added on existing apple, smart or analog watch
- Snap, vibration, and chime notifications
- Improved Bluetooth pairing

Specs

Bluetooth subsystem	Bluetooth 5.2 - Nordic Semiconductor nRF52832–QFAA 91725
Datarates	1 Mbps, 2 Mbps
Receive Sensitivity	–96 dBm at 1 Mbps, –89 dBm at 2 Mbps +4 dBm maximum
Transmit Power	GFSK, modulation index = 0.5
Modulation	2MHz
Channel Spacing	–20 dBc max
Adjacent channel power	2402 MHz to 2480 MHz (40 channels)
Operating Frequency Range	+/–250 kHz for 1 Mbps, +/– 500 kHz for 2 Mbps
Frequency Deviation	Johanson Technologies 2450AT18A100E ceramic chip antenna, 0.5 dBi typical peak gain over the 2400–2500 MHz band
Antenna	32.768 kHz, 32.000 MHz
Internal Clocks Power	Internal 3.7V nominal lithium polymer battery, 70 mA–hr capacity. Charged via USB-C cable at 5V +/– 0.5V with a maximum current of 70 mA.

FIGURE 2: PAVLOK 3 SPECIFICATIONS

Keen

Keen is marketed as a “habit-tracking, habit-breaking, awareness bracelet for trichotillomania (hair-pulling) and dermatillomania (skin-picking)” which can also be used for nail biting or other habits with distinguishable gestures [4]. Keen is a wearable bracelet device that records your gesture when you are performing your habit and vibrates to make you aware of when you are perform those gestures. Keen uses Bluetooth to connect between the bracelet and the app.

Hardware

- Wearable bracelet device

- USB charging cable

Software

- HabitAware Keen app – available on iOS and Android
- Train, track, and adjust gesture detection settings.
    - Record yourself performing the gesture.
    - Recognizes your movement by wrist angle, position, motion and speed.
    - Up to 4 gestures

Comparative Analysis

Pavlok, Keen and our solution are all wearable devices that use haptic feedback. The approach each solution has taken involves some form of negative stimulus that is applied when the habit is happening/about to happen and acts as a deterrence. Pavlok uses Bluetooth as it's form of communication between the app and the bracelet which is likely the same form of communication we will be using as well between the haptic feedback module and the proximity sensor module.

Proposed Solution

Where our solution differs from Pavlok is the main type of stimulus. The primary stimulus of Pavlok is an electrical shock whereas with our solution the stimulus is vibration. While Pavlok is used to manually shock yourself whenever you notice yourself biting your nails, our solution aims to have this process be completely autonomous. This benefits those who are often unaware when they are biting their nails and thus should have a more significant affect.

Although Keen also uses a vibration to deter the user from nail biting, the user is required to train device. This means they need to perform the action. One consideration here is that what if you bite your nails on the other hand? You would then need 2 Keen devices to completely prevent nail-biting. Our solution does not require this training and individuals only need 1 device to get the full benefit. In Keen's defense, it is not marketed directly as an anti-nail-biting solution but can be used as such if desired. Another note is when the user wants to perform an action such as eating. Our solution will have a clear enable/disable button. It is unclear if Keen also has this function, although there is a button to cancel the current gesture. That implies the vibration already started and would be frustrating for some users.

Comparison of Different Mounting Methods

Wrist

Based on initial research, a bracelet solution seemed to be the approach that was already taken by other solutions. The implementation worked by tracking the movement of the hand/arm, recording this gesture, and then providing a shock or vibration when the user repeated that gesture. The two devices found that used a bracelet design were the Pavlok [5] [6] [7] and Keen [8].

Earpiece

The earpiece was another idea that would mount around the ear (like a wireless headset). The sensor would be extended from the earpiece along the jawbone until it was in an acceptable position. There would either be a haptic actuator mounted behind the ear, or it would utilize sound through a headphone.

Glasses

Having the device mount to the frame of a pair of glasses was the earliest idea we had. The rough idea looked something like this:

Necklace

After further research and tips from our professor, we determined a necklace type design would be a viable solution to our design problem. The necklace, placed on the front side of the necklace, would allow us to provide a clear line of sight from the ultrasonic sensor to the mouth. Also, by placing most components at the front, we can make the necklace front heavy. This allows us to place the actuator at the back, due to the weight, it will rest snugly against the back of the neck. This should improve the noticeability of the vibration and make it more annoying.

The necklace design is optimal for our use case due to its flexibility and potential. In the design section you will find a simple model of the possible necklace design.

High-Level Design Conclusion

Based on the previous design ideas, we eventually landed on having the sensor mounted around the neck based on the following comparison:

TABLE 1: HIGH-LEVEL DESIGN COMPARISON

	Pros	Cons
Wrist	• Unobtrusive design • Doesn't require line-of-sight • Comfortable form factor	• Only detects gesture on a single arm • Need two bracelets to detect both hands
Earpiece	• Unobtrusive design. • More inconspicuous (hiding behind the ear). • Not hindered by clothing, can be worn any time.	• Difficult to design comfortable form factor. • Difficult to position sensor where the cheek would not interfere with the sensor.
Glasses	• Unobtrusive design. • No need to design how the device would mount to the head because it's already using the glasses frame. • Easy line-of-sight for sensor .	• Limited to people with glasses. • Limited to certain types of glasses • Possibly difficult to mount on/off when not wanting to use.
Necklace	• Easy to wear • Not as limited in size constraints • Simple form factor • Better line-of-sight for sensor	• Could potentially shift positions during normal movement • Can't use with bulky clothing (i.e., winter jackets)

Rationale

The impact that this could have on chronic nail-biters freedom from their addiction/habit. Since chronic nail-biting is not seen as a major public issue or addiction, it generally flies under the radar for new products being produced or research being done. There are some solutions that work for some but not for all and they are typically extremely restrictive and/or hindering. Personal experience has shown that none of the existing solutions have worked. In addition, most of the time I don't know I'm biting my nails because I start without realizing. Solutions that I have personally attempted:

- Wearing gloves
  - Very restrictive when working with hands.
  - Need to remove whenever using a touch screen (even the best "touchpad" gloves do not work well).
  - Need to remove when doing any actions requiring fine finger movement (many things with electronics)
  - Typing is more difficult because you can't feel the keyboard as well, slowing productivity
- Bad tasting "nail polish"
  - Taste is not bad enough, will bite through it and endure
  - The "nail polish" is more like an oil, and will get on EVERYTHING you touch, spreading it to other people. Needed to stop using it when I had a child or else the child would get a really bad taste if he ever touched anything I touched.
- Keeping nails/cuticles manicured
  - Difficult to maintain a daily regular schedule
  - Takes a long time to properly maintain every day
  - Barely preventative
- Keeping nails trimmed short
  - They're already as short (and shorter) than they can go.
- Replacing habit with good habit (like a stress ball)
  - Annoying to carry around a stress ball everywhere you go since nail-biting happens at anytime
- Identify triggers
  - Most of the time happens sporadically
  - Most of the time don’t' even realize that I start biting
- Keep mouth busy
  - Chewing on anything hard for extended periods of time is damaging
- Focus on one finger at a time
  - Works for the first one or two, then stops working.

Preliminary Design

As stated earlier in the system overview, the system consists of three main elements to satisfy the requirements of the project. The first is that it needs some way of detecting when a hand is getting closer to the mouth. The second, some form of device that can provide sensory feedback to the user. Finally, the data processing between the two which will implement a state machine that will determine when the tactile feedback device turns on or off depending on if a hand is detected close to the mouth by the sensor.

Data Processor (Bluetooth SoC microcontroller)

A micro-controller was deemed necessary for the project's success due to its ability to receive sensor data from the distance sensor and in response trigger the actuator either on or off depending on if the distance is within certain parameters. In addition to the previous attributes, the micro-controller could also double as a Low Energy Bluetooth module to be able to communicate with a mobile device to log information (i.e., how often the sensor is triggered and/or for how long).

The micro-controller was chosen once we had a better idea of what we were going to do for the power supply, actuator, and sensor so that we could choose a micro-controller that would be best suited for these peripherals.

Following were the requirements for the micro-controller:

- Powered by either USB (5V) or Battery
    - This was deemed necessary because this project is designed to be a wearable device and the usage of a battery as a power supply is necessary. Since our device would be powering an actuator, micro-controller, ultra-sonic sensor, and possibly BLE, it was assumed that a simple coin battery would be insufficient, and therefore a rechargeable battery was necessary. To keep recharging simple, we decided to implement the ability to charge from USB, a very common method for charging small wearable devices.
- I2C communication
    - The CHIRP101 ultrasonic sensor communicates with a host device (the micro-controller) via an I2C interface. Therefore, it is necessary for the micro-controller to have this ability.
- PWM (obsolete)
    - Initially, we assumed that the actuator that we would use would require a sinusoidal signal. Hence, we initially looked for a micro-controller with this capability. However, we ended up choosing an actuator that only required direct current (DC) and therefore did not need PWM.
- Low Energy Bluetooth
    - Since we wanted the ability to communicate with a mobile phone via Bluetooth to log information, it was deemed necessary to have some form of a Bluetooth module on our device.
- Buck Voltage Regulator (Bonus)
    - Since the CHIRP101 requires 1.8V, it would be a bonus to have a micro-controller with a voltage regulator built into it so an external circuit would not be necessary. Note: It should be an Low-dropout (LDO) regulator to minimize noise.

There were a few alternative design ideas for this part of the project. These are listed below:

Alt. Idea 1: "Hack" an existing Bluetooth device

The idea here was to find an existing Bluetooth device (probably a headset) that we could reverse-engineer for our purposes.

Pros:

- This would reduce the time spent on configuring Bluetooth connectivity since it was new field for most of the team.
- It would also mean that we don't have to design the charging capability because it would already exist in the device.
- The structure and form-factor of the device would already exist, so not time spent working on that either.

Cons:

- In this case, we would likely use an auditory signal for indicating that the user is about to bite their nails instead of a vibrator one. However, I do not believe an auditory signal would work as well as a tactile one.
- How to integrate the ultra-sonic sensor that requires I2C? It would be very difficult to find an existing Bluetooth device that has exposed pins for an I2C, let alone finding a device that provides that level of information.

Alt. Idea 2: Prefab micro-controller board

This idea involves using a prefabricated board with all necessary peripherals. Some examples could be a Raspberry Pi Pico or the Bluefruit NRF52 Feather board.

Pros:

- Board is prefabricated so no need for hardware design.

Cons:

- On the larger side for a wearable device. Borderline unusable.
- Many peripherals on the board are not used and are a waste of space.
- Expensive option if project was to be scaled up (production).
- Difficult to find prefab board with all necessary components.

Alt. Idea 3: System-on-a-Chip on Custom Board

This idea would be to use a Bluetooth module with an embedded micro-controller and then design a custom board with necessary peripheral circuitry. Fortunately, there are micro-controller System-on-a-Chip (SoC) that are built into a Bluetooth module, which gives you a micro-controller and Bluetooth capabilities (antenna included) in one module. Without the use of this model, our design would require a host micro-controller, a Bluetooth controller, and a Bluetooth antenna. This SoC package brings all those components into one module.

Pros:

- Custom form-factor. Can choose how big or small and the shape of the device.
- Better customization in terms of what is on the board. Can design for efficiency.
- More freedom in choosing components.

Cons:

- More time spent designing hardware and software.
- Manufacturing and assembly would be necessary.

- More time spent debugging hardware, rather than spending it on software functionality.

Micro-controller Alternative Ideas Conclusion

Due to the comparison between the different ideas, our team decided to go with alternative idea number three, which consists of designing a custom board with a SoC Bluetooth module, but instead of completing the entire fabrication by the end of the term, we would demo the functionality of our design using prefabricated demo boards.

Of the SoCs available, we decided to use the **MDBT50Q-1MV2 from Raytac Corporation that utilizes the nRF52840 Bluetooth chip from Nordic**, because of its availability and because it covers all requirements (including a bonus LDO voltage regulator).

Sensor

One key area researched was the type of detection method that would be used in our device. The possible sensor technologies investigated were Infrared, Magnetic, Ultrasonic, Capacitive, and Inductive.

TABLE 2: DETECTION METHOD COMPARISON

Detection Method	Advantages	Disadvantages
Infrared	• Low power requirements. • Detect motion/proximity with or without light. • Strong noise immunity. • Don't require contact with the object being detected.	• Require line of sight → OK for our use case. • Limited range → Advantage in our use case. • Impacted by environmental conditions (eg. fog) → Unlikely to be an issue in our use case. • Transmission data rate is slow → Negligible for our use case.
Magnetic	• 70mm range should be enough for near-face detection. • Small package size	• Requires a magnetic ring the user can wear → ring might be attracted to other magnetic items in the users life.

Ultrasonic	• Transparent object detectable
	  ◦ Since ultrasonic waves can reflect off a glass or liquid surface and return to the sensor head, even transparent targets can be detected.
	• Resistant to mist and dirt
	  ◦ Detection is not affected by accumulation of dust or dirt.
	• Complex shaped objects detectable
	  ◦ Presence detection is stable even for targets such as mesh trays or springs.
		• Sensing accuracy affected by soft materials
		  ◦ Objects covered in a very soft fabric absorb more sound waves making it hard for the sensor to see the target.
		• Sensing accuracy affected by changes in temperature of 5-10 degrees or more
		  ◦ Although this is true, we have a variety of temperature compensated sensors available that either calibrate upon start-up or before every range reading

	- Not affected by color or transparency of objects - Ultrasonic sensors reflect sound off objects, so the color or transparency have no effect on the sensor's reading. - Can be used in dark environments - Unlike proximity sensors using light or cameras, dark environments have no effect on an ultrasonic sensor's detection ability. - Tend to consume lower current/power - Multiple interface options for pairing with a microcontroller, etc.	depending on the sensor model. During this time is when the sensor will calibrate with any change in temperature, voltage, etc. This dramatically decreases this problem. - Have a limited detection range - At the moment, our longest range sensors have a maximum range of 10 meters, now our cargo sensor detects up to 16.5m. While this is a disadvantage in certain applications, our sensors have great mid-range capabilities and are still suited for many applications. - Low resolution and slow refresh rate, making it not suitable for detection of fast-moving targets
Capacitive	Can sense any type of material.	Typical sizes are much too bulky for our application.
Inductive	Senses ferrous materials well.	Like the capacitive sensors, the typical sizes are much too bulky for our application.

The detection method, based on the pros and cons of each, was decided to be ultrasonic. In addition, a micro electro-mechanical system (MEMS) ultrasonic sensor was chosen based on its size and low power requirements.

The CH101 ultrasonic sensor from TDK InvenSense was eventually chosen [9].

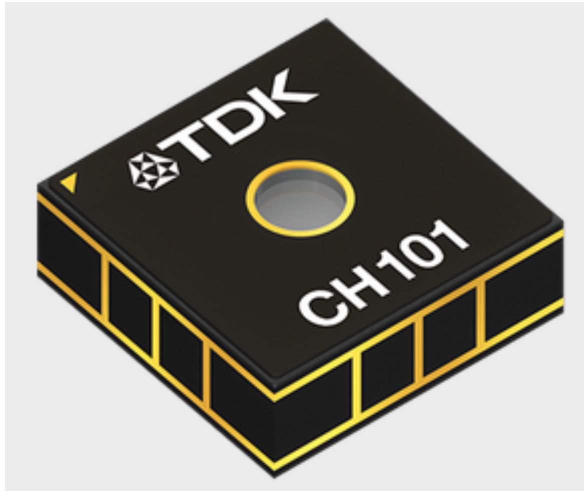


FIGURE 3: CH101 ULTRASONIC SENSOR FROM TDK INVENSENSE [9]

Based on:

- Ultra-low power
- Works in any lighting condition
- Detects objects of any colour and optical transparency
- Customizable field of view (FoV)
- Small form-factor
- Low-power SoC firmware for ultrasonic processing a wide range of usage cases

Actuator (Vibrator)

The actuator is important in order notify the user when they are biting their nails. Since this is an embedded product that will be run on battery power, there are a few considerations for the actuator:

- Low power requirements
- Small footprint
- Low cost

Through some research, we looked at multiple vibration actuator types:

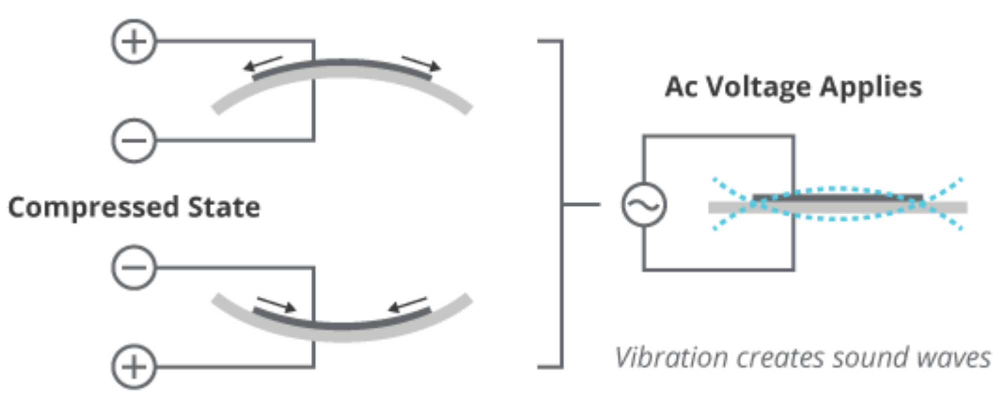


FIGURE 4: PIEZO BUZZER DIAGRAM [10]

Piezo vibrators work by being supplied AC voltage, causing them to flex in opposite directions, creating a vibration.

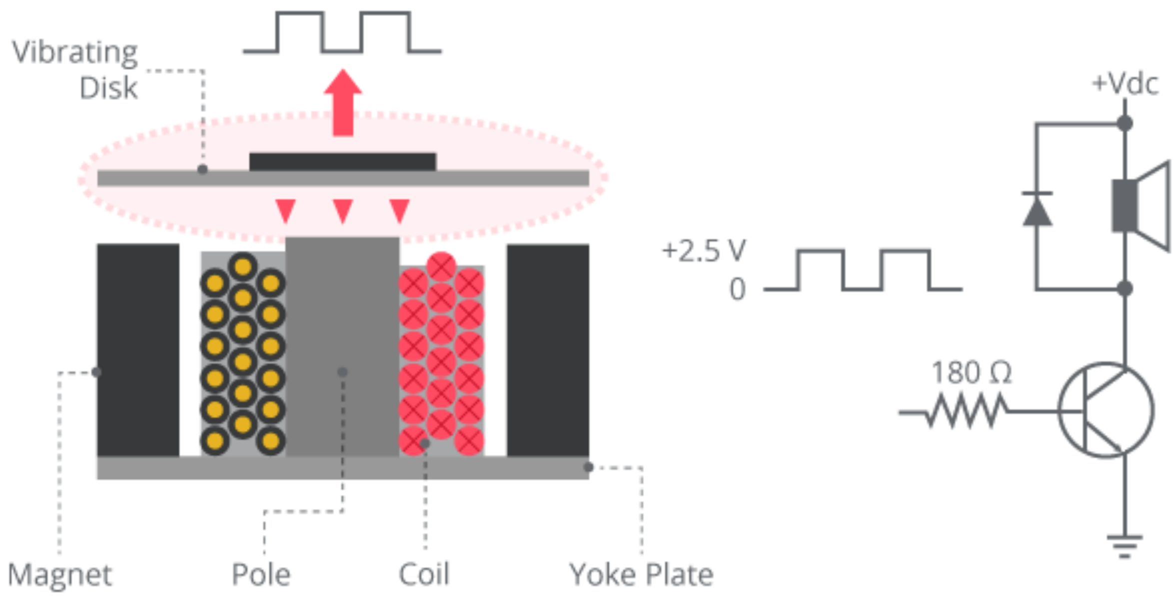


FIGURE 5: MAGNETIC BUZZER [10]

The magnetic buzzer uses a coil to attract a vibrating disk. This requires a PWM signal to attract and de-attract the disk, causing a vibration.

FIGURE 6: COIN VIBRATION BUZZER [11]

The coin vibration motor is essentially a coreless/brushless DC motor. When supplied a DC voltage, it turns on and vibrates. The model in the figure above is a coin model but this type of actuator can also come in a cylinder/bar model where a misbalanced bar is used to get a vibration.

Throughout this research, we've decided to move forward with the coin vibration motor. Here are some of the following reasons:

- Low voltage requirements (3V3)
- Simplicity (only requires constant DC source to use)
- Footprint (smaller than a quarter)

For the above reasons we went with the coin vibration motor [11].

Final System Design

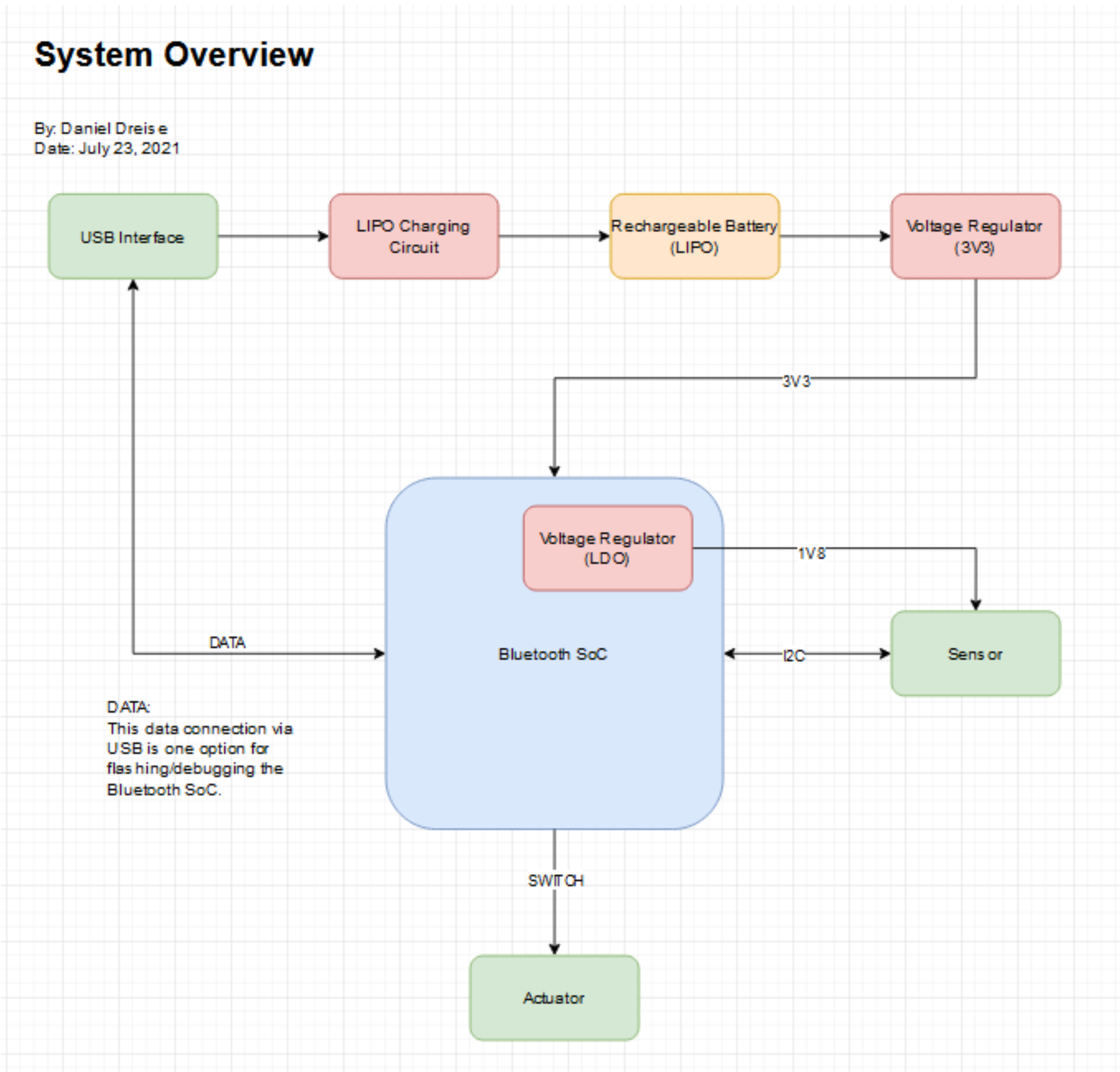


FIGURE 7: SYSTEM DESIGN

Detailed Design

Actuator - Coin Model Vibration Motor

The vibration motor chosen has the following features [11]:

- 12000 RPM (200Hz)
- 10mmx3mm
- 3V DC
- 80mA current requirement
- Self-adhesive
- 0.7 ounces
- $14.99 for 20 pieces

Since this component requires constant DC power to turn on, it's much simpler for us to implement. Unfortunately, we can't power it with a GPIO port of the microcontroller. The GPIO ports are unable to supply the required 80mA. Fortunately, the battery operates at 3V7 and there is a 3V3 pin available on the microcontroller that can supply the required current. If available, we will use the 3V3 pin, if unavailable, we will tap directly into the battery supply pins. A BJT transistor will be used to switch the supply to the actuator on and off. A BJT is chosen to provide the necessary current amplification. One key consideration is that the BJT must be rated for logic level use. This is not a power implementation, and we are operating in the mA and low voltage range. A capacitor and diode are put in parallel with the actuator to absorb, and protect against, voltage spikes.

Schematics

A hierarchical design method was chosen to portray the different circuits, which includes a top level that provides the interface between different modules.

Top Level

Figure 8 shows the top-level design for the project. It consists of four modules: power supply, processor (microcontroller), ultrasonic sensor, and actuator.

Added to this schematic is the pull-up resistors R1 and R2. These pull-ups are necessary for I2C communication, as the internal pull-ups in the microcontroller are insufficient.

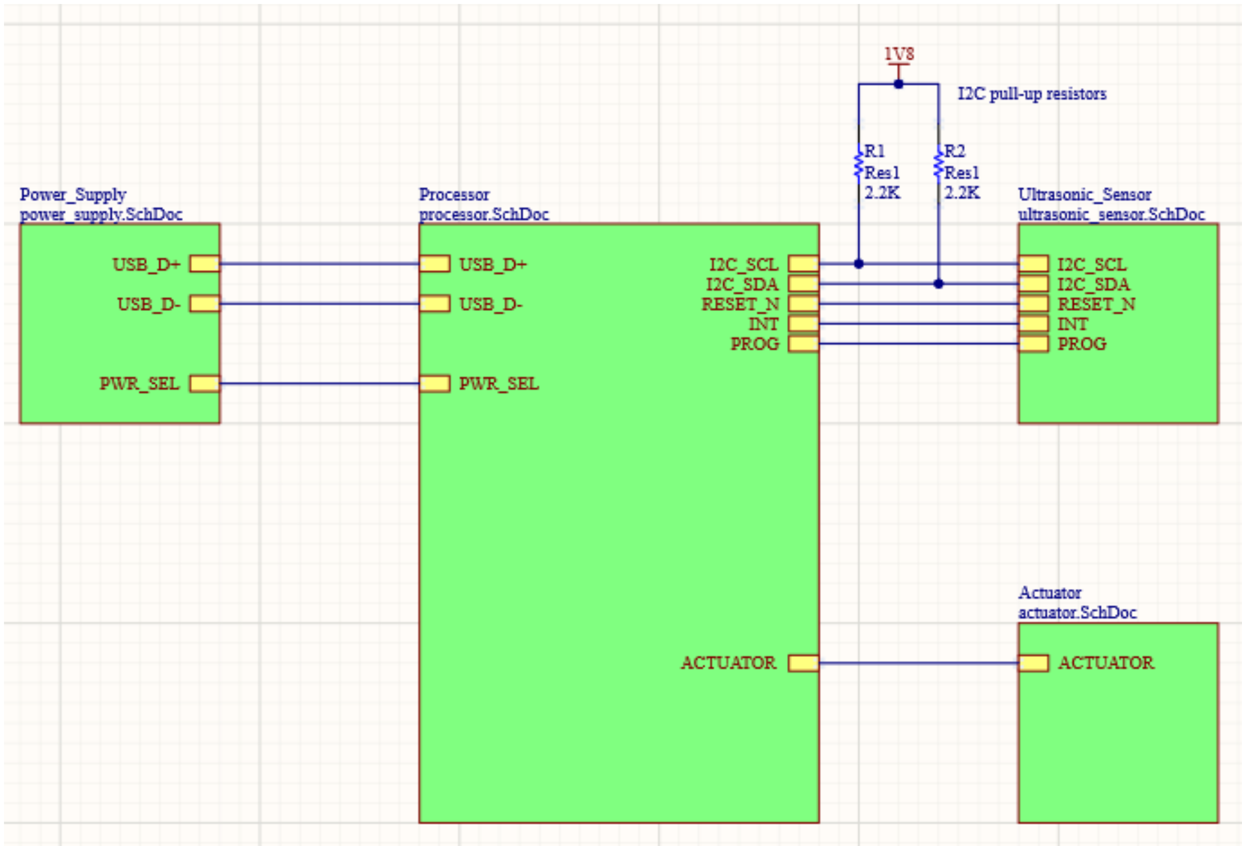


FIGURE 8: TOP LEVEL DESIGN FOR SCHEMATIC

Microcontroller (Processor)

Figure 9 shows the schematic for the MDBT50Q Bluetooth SoC module (M1). Moving in a counterclockwise direction around the microcontroller:

TABLE 3: MICROCONTROLLER PINOUT

M1	MDBT50Q Bluetooth SoC module
SW1	Button for toggling ON/OFF the functionality of the device
I2C_SCL	I2C clock for ultrasonic sensor
I2C_SDA	I2C data for ultrasonic sensor
RESET_N	Reset for ultrasonic sensor
INT	General purpose pin for ultrasonic sensor
PROG	For programming the ultrasonic sensor if custom firmware needed
PWR_SEL	Connected to voltage divider on battery bus for determining power supply
VDD	Output to 1.8V. This is connected to the internal LDO voltage regulator.
VDDH	Input from battery supply (>3.6V)

VUSB	Input from USB supply (5V)
USB_D+	Data line for USB
USB_D-	Data line for USB
ACTUATOR	Actuator switch for turning on/off
P3	SWD 3-pin header for flashing/debugging

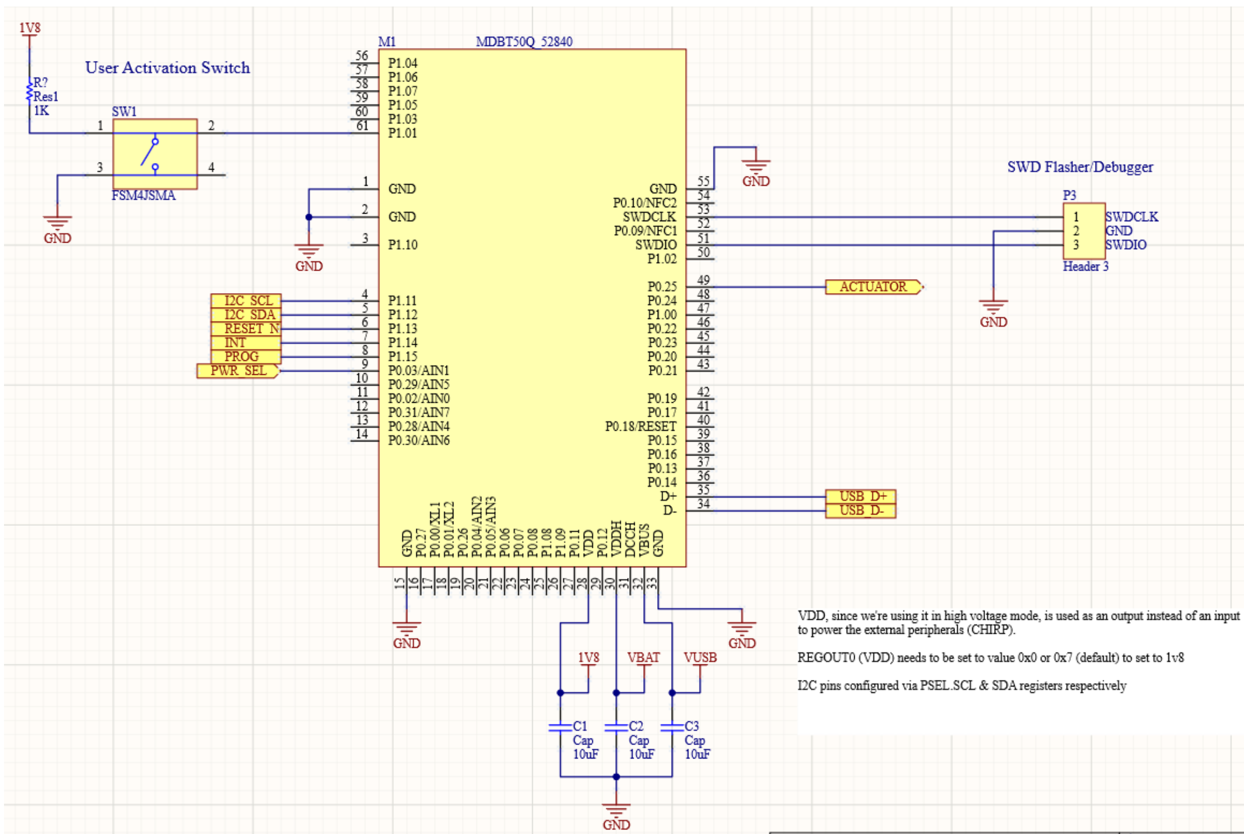


FIGURE 9: MICROCONTROLLER CIRCUIT

Power Supply

USB Connection

P1 is a micro USB connection that when connected to a host device, grants a 5V supply to the board in addition to communication with the microcontroller via D- and D+.

Nordic (the internal microcontroller) says that the USB series terminator resistors are built into the SoC, but the reference schematic from Raytac (the Bluetooth module that encapsulates the Nordic microcontroller) has them included. This design included them, and it has been noted that this could cause issues.

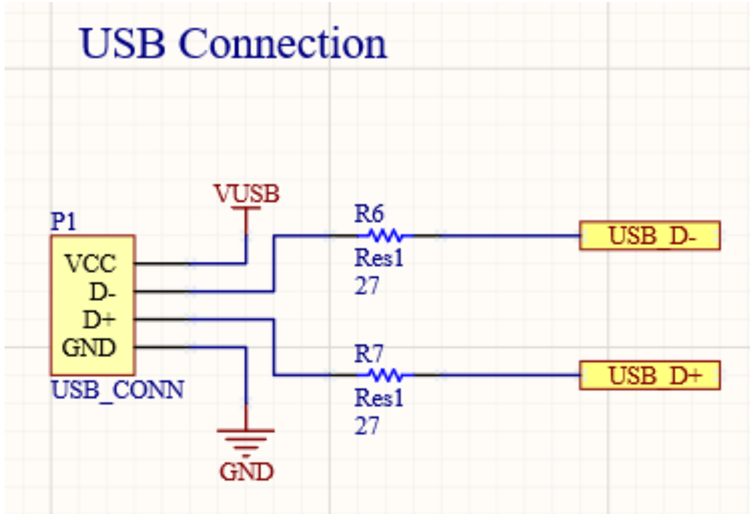


FIGURE 10: POWER SUPPLY - USB CONNECTION CIRCUIT

Battery Connection

Figure 11 shows the battery connection. It has a JST 2-pin connector that allows the device to be connected to a 3.3 V rechargeable battery. The rechargeable battery is the MIKROE-4471 that operates at 3.7 V with 250mAh of power. This can be increased if the amount of battery is insufficient. Capacitor C6 is used to minimize voltage fluctuations.

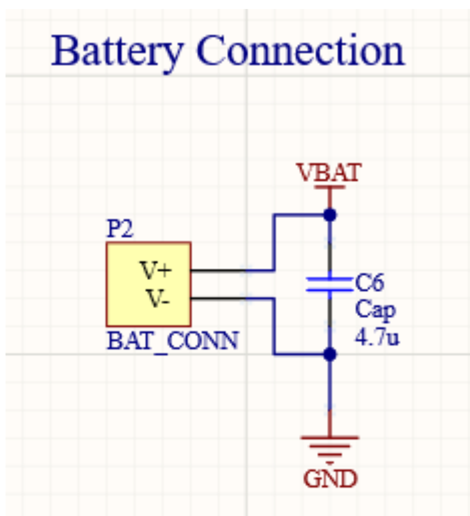


FIGURE 11: POWER SUPPLY - BATTERY CONNECTION CIRCUIT

LIPO Charging

Figure 12 shows the Lithium Polymer charging circuit for recharging the battery from the USB power supply. It consists of the MCP73831T single-cell lithium polymer charge management controller, which allows for safe and efficient battery charging. The LED is added so the user can be notified when the battery is being charged.

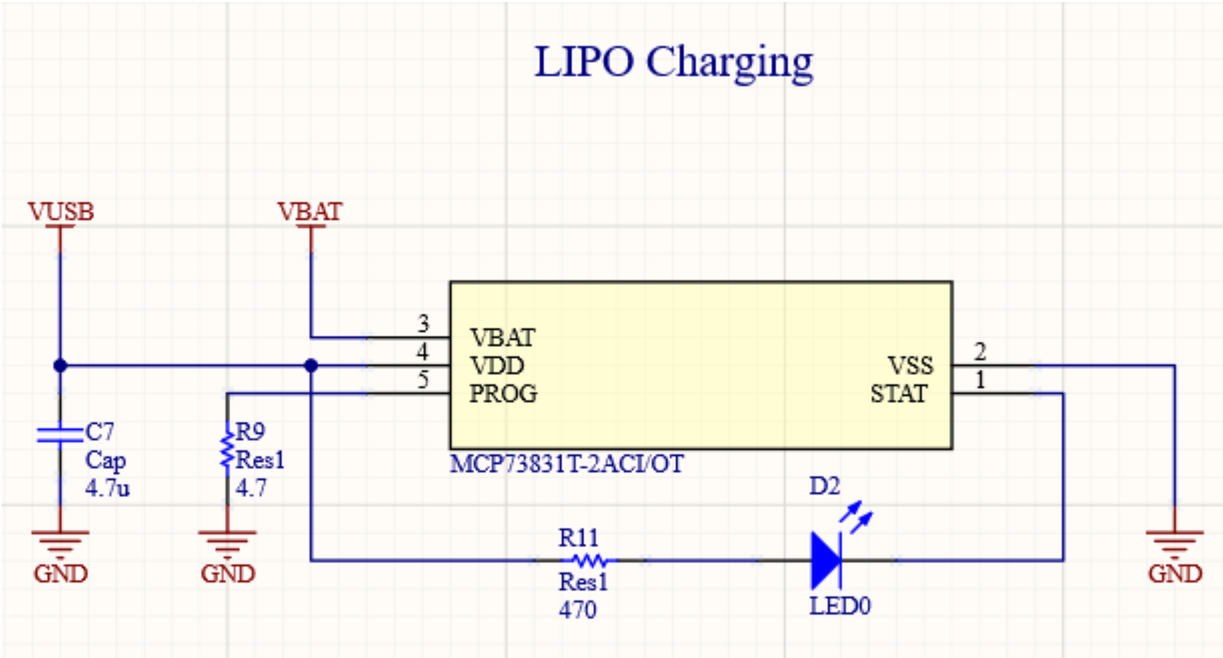


FIGURE 12: POWER SUPPLY - LIPO CHARGING CIRCUIT

Dynamic Power Selection

Figure 13 is a circuit that enables the microcontroller to detect whether both power supplies (USB and battery) are connected at once. Essentially if both are connected, the USB power source will override the battery power source. PWR_SEL is connected to an analog pin on the microcontroller.

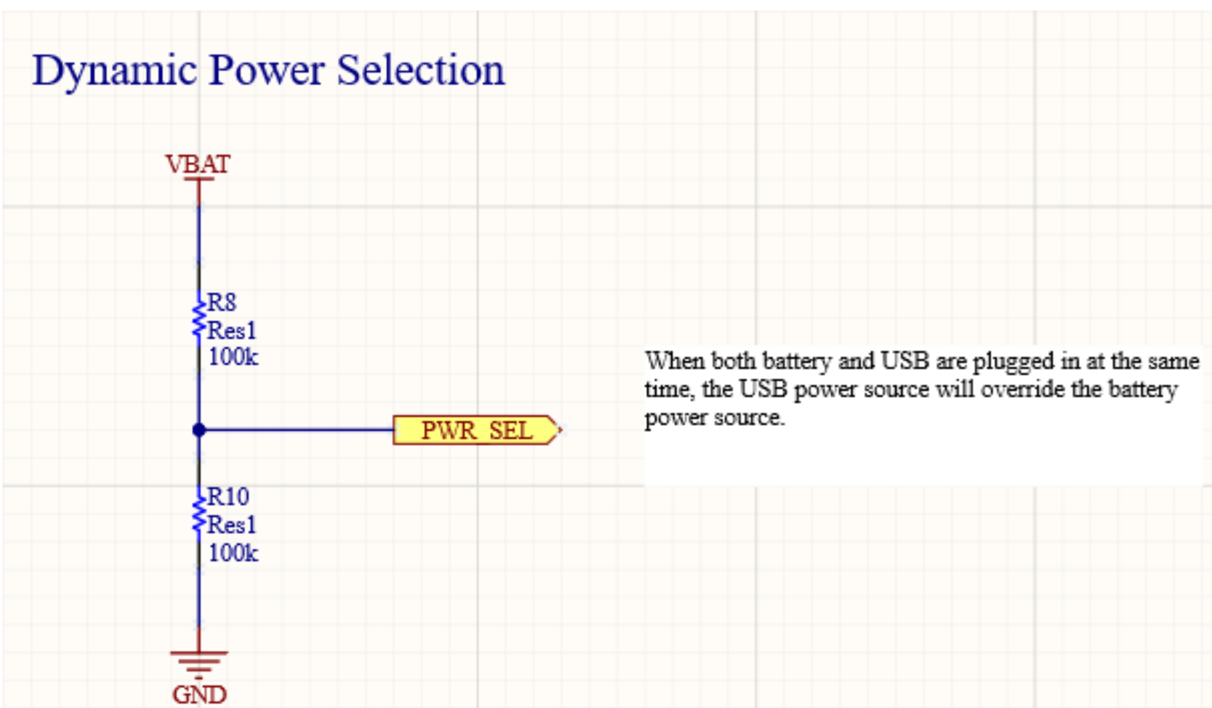


FIGURE 13: POWER SUPPLY - DYNAMIC POWER SELECTOR CIRCUIT.

Actuator

The vibration motor chosen has the following features [11]:

- 12000 RPM (200Hz)
- 10mmx3mm
- 3V DC
- 80mA current requirement
- Self-adhesive
- 0.7 ounces
- $14.99 for 20 pieces

Figure 14 shows the switching circuit for the actuator. Since the GPIO ports on the microcontroller are unable to supply the necessary amount of current to turn on the actuator, a switch is implemented. ACTUATOR is connected to a GPIO on the microcontroller. When current is added to the base of the BJT transistor (Q1), it turns the actuator on and vibrates the buzzer. D1 is a flyback diode which allows for the buzzer to dissipate extra power safely without being directed to the microcontroller. C5 is a capacitor which helps regulate voltage fluctuations.

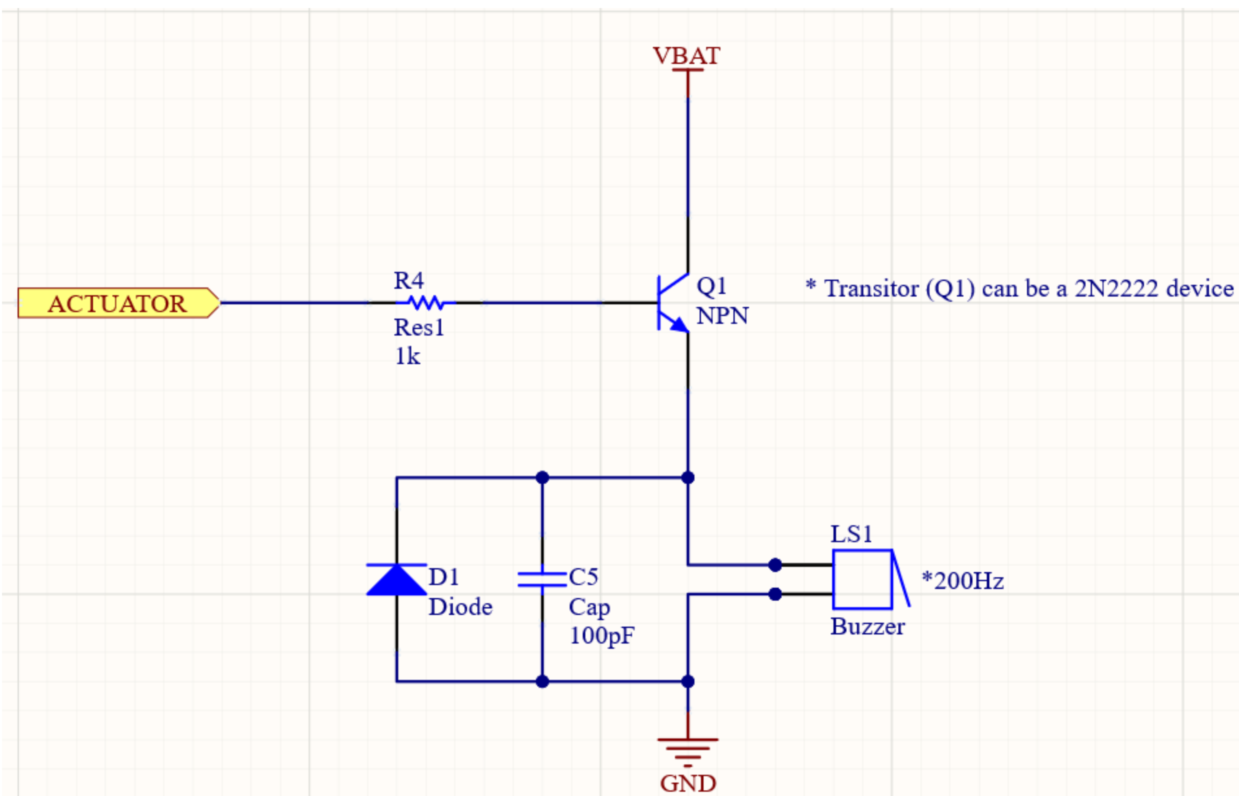


FIGURE 14: ACTUATOR CIRCUIT.

Ultrasonic Sensor

Figure 18 is the MEMS ultrasonic sensor circuit. This circuit is designed in accordance with the CH101 datasheet suggested by the makers TDK Chirp Microsystems.

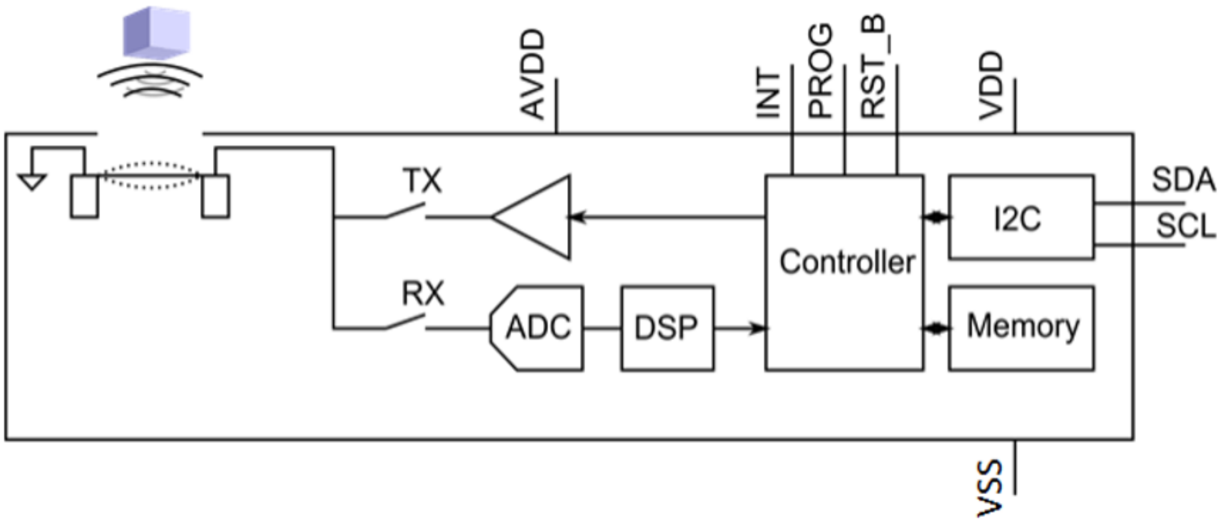


FIGURE 15: SIMPLIFIED BLOCK DIAGRAM OF CH101

PIN	NAME	DESCRIPTION
1	INT	Interrupt output. Can be switched to input for triggering and calibration functions
2	SCL	SCL Input. I²C clock input. This pin must be pulled up externally.
3	SDA	SDA Input/Output. I²C data I/O. This pin must be pulled up externally.
4	PROG	Program Enable. Cannot be floating.
5	VSS	Power return.
6	VDD	Digital Logic Supply. Connect to externally regulated 1.8V supply. Suggest common connection to AVDD. If not connected locally to AVDD, bypass with a 0.1µF capacitor as close as possible to VDD I/O pad.
7	AVDD	Analog Power Supply. Connect to externally regulated supply. Bypass with a 0.1µF capacitor as close as possible to AVDD I/O pad.
8	RESET_N	Active-low reset. Cannot be floating.

FIGURE 16: CH101 PIN DESCRIPTIONS

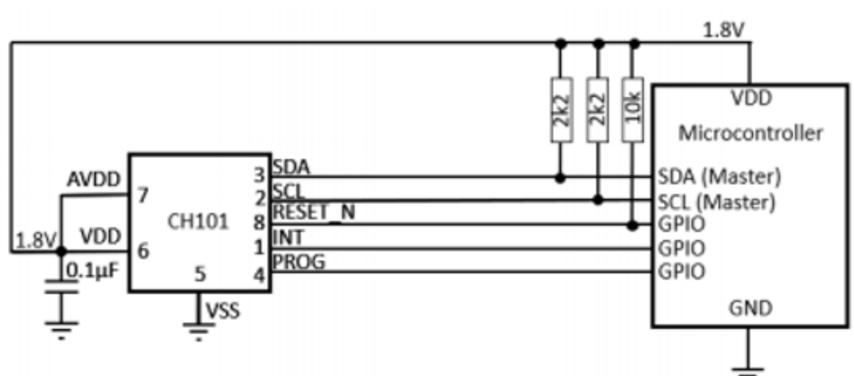


FIGURE 17: CH101 TYPICAL OPERATING CIRCUIT

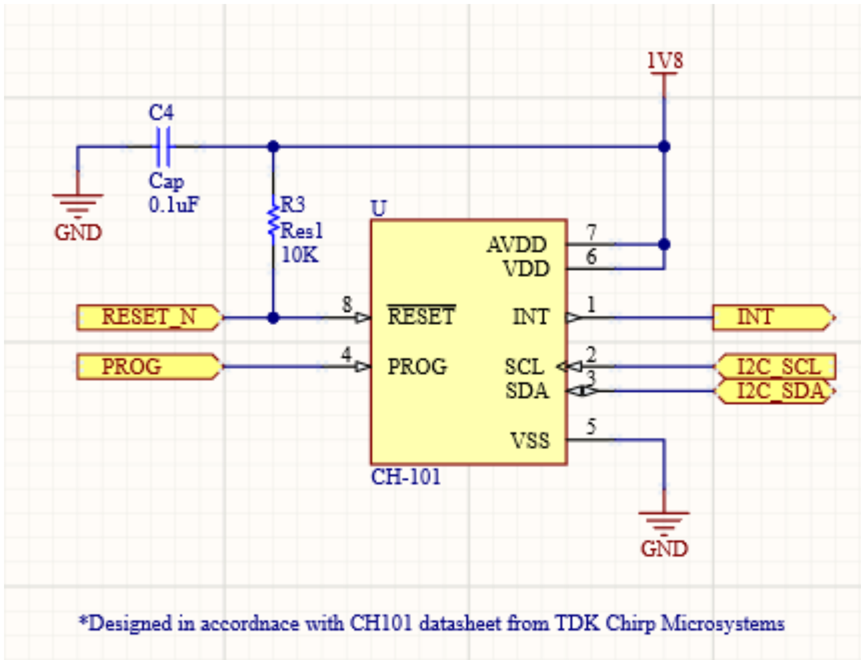


FIGURE 18: FINAL MEMS ULTRASONIC SENSOR CIRCUIT.

State Machine

The anti-nail-biting solution we have come up with has a relatively simple state machine. On power, the device will check for the hand near the mouth. If it's there, the vibration actuator turns on; if it's not, the vibration actuator turns off. This cycle repeats indefinitely until the device is turned off.

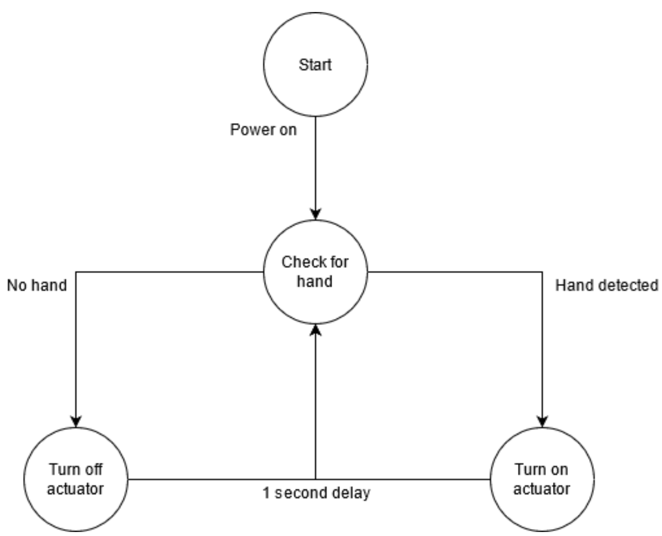


FIGURE 19: ANTI-NAIL-BITING DEVICE STATE MACHINE

Software Design

The overall software design has 4 components. The ultrasonic sensor, the actuator, the push button, and the logic on the micro-controller.

The ultrasonic sensor design is based around the CH101 sensor. The CH101 sensor is packaged with a breakout board featuring its own micro-controller which can be programmed with custom firmware. We interact with the sensor mainly through an I2C interface. Other GPIO pins are used for things like interrupt handling, sensor resets, and sensor programming. The CH101 sensor uses a library called "SonicLib" to handle the sensor interaction. To maintain compatibility with many different micro-controller boards, each user must implement a set number of functions to properly provide support to their chosen development board. These functions essentially hook into the hardware of the specific development board. Such as I2C pin placements, which GPIO pins are connected to what sensor IOs, etc.

In the event we were not able to implement the CH101 sensor in time, we had a simple backup ultrasonic sensor. To interface with this sensor, we need to pull a trig pin high for 10us, then proceed to read the given result on an echo pin. The result is given in time for the wave to hit the object and bounce back. Therefore, we need to divide by 2 and multiply by the speed of sound to get the distance from the object. See the below figure.

```
void loop() {
  // Clears the trigPin condition
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin HIGH (ACTIVE) for 10 microseconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and back)
  // Displays the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  if(distance < MIN_DIST)
  {
    digitalWrite(VIB_PIN, HIGH);
  }
  else
  {
    digitalWrite(VIB_PIN, LOW);
  }

  delay(100);
```

FIGURE 20: BACKUP SENSOR IMPLEMENTATION

The vibration actuator is relatively simple to interface with on the software side. Since it is essentially a DC motor, we only need to toggle it on or off. No PWM or fancy AC signals are required. The below figure demonstrates this function. Near the end of this document are demonstrations that also showcase these designs.

```
if(distance < MIN_DIST)
{
  digitalWrite(VIB_PIN, HIGH);
}
else
{
  digitalWrite(VIB_PIN, LOW);
}
```

FIGURE 21: ACTUATOR SOFTWARE IMPLEMENTATION

The push button is built-in to the micro-controller and therefore relatively simple to access. It exists on pin 7. Simply setting this pin as input and reading its state provides information on whether its being pressed or not. One thing to consider is the debounce of this switch is high. Therefore, it is worth considering adding a dedicated switch to the circuit to allow more flexibility with decreasing debounce. See the below figure for how the button is read:

```
button = digitalRead(7);
```

FIGURE 22: PUSH-BUTTON SOFTWARE DESIGN

The micro-controller wraps all these components nicely together and allows for them to interact. For our design we went with the Adafruit Feather Express NRF52840 board. Using Arduino to program it, we were able to design a codebase that allowed these components to interact with each other. The basic design is that each component is initialized and prepared for runtime. Once the sensor detects an object within range, if the push button hasn't disabled the actuator, the actuator will be activated and start vibrating. Once the object leaves the range, the actuator turns off. Arduino provides many libraries that make initialization of most components simple, with the CH101 being the exception. You can find our codebase at the following link. There are two branches other than main. One is our implementation for the backup sensor and the other is our work in progress implementation of the CH101 sensor: https://github.com/koolohms/anti-nail-biting/tree/main

Necklace Mount

The necklace mount allows the user to wear the device as a necklace so that the sensor is pointing below the chin to detect if the hand gets close to the mouth.

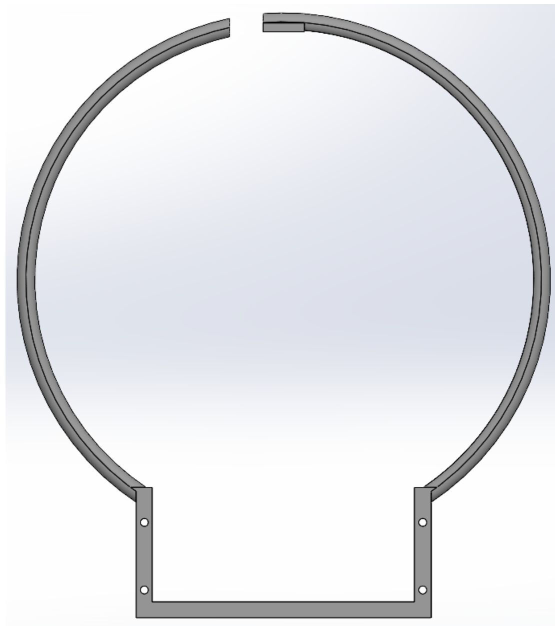


FIGURE 23: NECKLACE MOUNT TOP VIEW

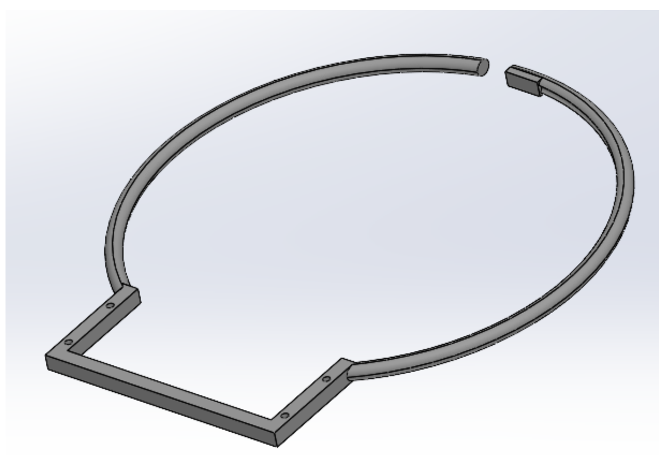


FIGURE 24: NECKLACE MOUNT ISOMETRIC VIEW

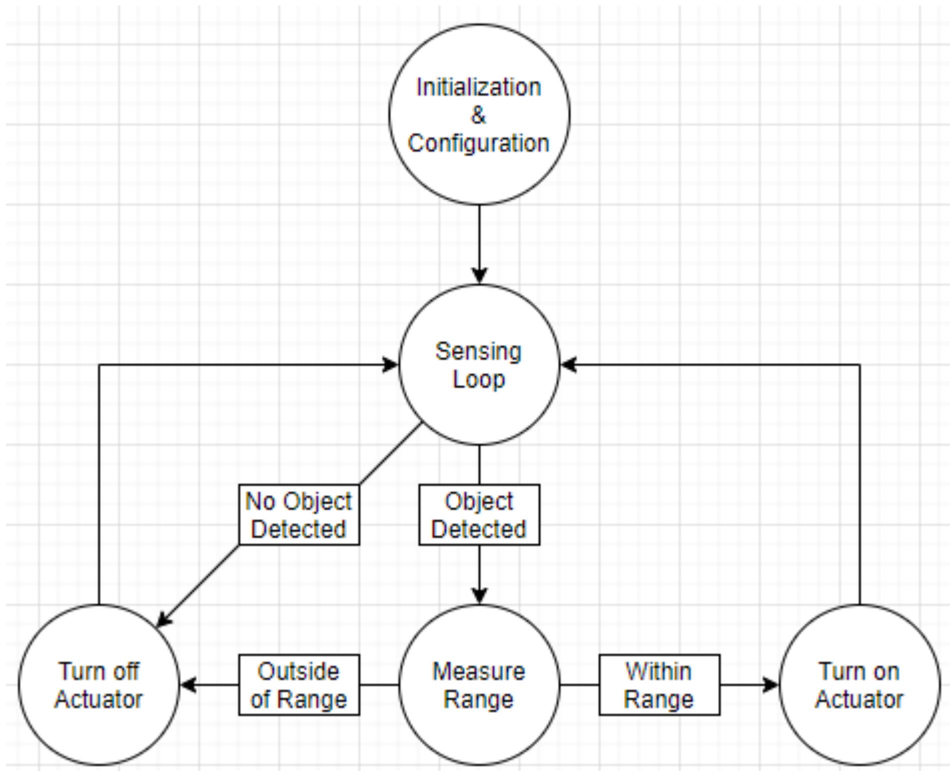
Implementation

Software – Initialization, Configuration & Sensing [12]

Overall Application Flow

1. Initialize hardware (*chbsp_board_init()*)
2. Initialize SonicLib structures for each sensor (*ch_init()*)
3. Register callback routine(s) – see below

4. Program and start all sensors (*ch_group_start()*)
5. Configure sensors (*ch_set_config()* etc.)
6. Enter endless loop to perform sensing:
   a) Callback functions are called by SonicLib/BSP to notify the application of events
   b) Use *ch_get_range()* etc. to get measurement data



Hardware Initialization

- Definition: *void chbsp_board_init (ch_group_t *grp_ptr)*
- Executes the required hardware initialization sequence for the board being used
- Includes clock, memory, and processor setup as well as any special handling that is needed
- Called at the beginning of an application, as the first operation.
- Also initializes the following fields within the ch_group_t sensor group descriptor

ch_group_t Field	DESCRIPTION
num_ports	Number of possible sensor ports on the board Usually the same as *CHIRP_MAX_DEVICES* in **chirp_board_config.h**. Accessible later using *ch_get_num_ports()*.
num_i2c_buses	Number of I²C buses used by sensors on the board. Usually the same as *CHIRP_NUM_I2C_BUSES* in **chirp_board_config.h**.
rtc_cal_pulse_ms	Length (duration) of the pulse sent on the INT line to each sensor during calibration of the real-time clock, in milliseconds. Accessible later using *ch_get_rtc_cal_pulselength()*.

Software Initialization

- Definition: *uint8_t ch_init (ch_dev_t *dev_ptr, ch_group_t *grp_ptr, uint8_t dev_num, ch_fw_init_func_t fw_init_func)*

- Used to initialize various Chirp SonicLib structures before using a sensor
- The ch_dev_t device descriptor is the primary data structure used to manage a sensor, and its address will subsequently be used as a handle to identify the sensor when calling most API functions
- The dev_ptr parameter is the address of the ch_dev_t descriptor structure that will be initialized and then used to identify and manage this sensor.
- The grp_ptr parameter is the address of a ch_group_t structure describing the sensor group that will include the new sensor.
- Both the ch_dev_t structure and the ch_group_t structure must have already been allocated before this function is called.
- dev_num is a simple index value that uniquely identifies a sensor within a group. Each possible sensor (i.e. each physical port on the board that could have a Chirp sensor attached) has a number, starting with zero (0). The device number is constant - it remains associated with a specific port even if no sensor is actually attached. Often, the dev_num value is used by an application as an index into arrays containing per-sensor information (e.g. data read from the sensors).
- The Chirp sensor is fully re-programmable, and the specific features and capabilities can be modified by using different sensor firmware images.
- The fw_init_func parameter is the address (name) of the sensor firmware initialization routine that should be used to program the sensor and prepare it for operation.
- The selection of this routine name is the only required change in the application when switching from one sensor firmware image to another.
- ch_init() only performs internal initialization of data structures, etc. It does not actually interact with the physical sensor device(s)

Register Callback Routines

I/O Interrupt Callback

- Format: void io_int_callback_name (ch_group_t *grp_ptr, uint8_t io_index)
  - The io_int_callback _name is the name of the callback routine in your application. The name may be anything you choose.
  - The grp_ptr parameter is a pointer to the sensor group structure for the interrupting device.
  - The io_index parameter is the device index number within the sensor group.
  - Together, the grp_ptr and io_index parameters uniquely identify the interrupting device.
  - The address of the corresponding device descriptor (ch_dev_t structure) can be determined by passing these values to the ch_get_dev_ptr() function.
- To register: void ch_io_int_callback_set (ch_group_t *grp_ptr, ch_io_int_callback_t callback_func_ptr)
  - The grp_ptr parameter is a pointer to the sensor group structure for the interrupting device.
  - The callback_func_ptr parameter is the address (name) of your callback routine.

I/O Complete Callback

- Format: void io_complete_callback_name (ch_group_t *grp_ptr)

- The io_complete_callback _name is the name of the callback routine in your application. The name can be anything you choose.
   - The grp_ptr parameter is a pointer to the sensor group structure for the device.
- To register: void ch_io_complete_callback_set (ch_group_t *grp_ptr,
  ch_io_complete_callback_t callback_func_ptr)
   - The grp_ptr parameter is a pointer to the sensor group structure for the device whose I/Q data is being read.
   - The callback_func_ptr parameter is the address (name) of your callback routine.

Sensor Initialization

- Definition: *uint8_t ch_group_start (ch_group_t *grp_ptr)*
- Performs the actual discovery, programming, and initialization sequence for all sensors within a sensor group.
- Each sensor must have previously been added to the group by calling ch_init().
- In brief, this function does the following for each sensor:
  1. Probe the possible sensor ports using I2C bus and each sensor's PROG line, to discover if sensor is connected.
  2. Reset sensor and put in known state.
  3. Program sensor with firmware (version specified during ch_init()).
  4. Assign unique I2C address to sensor (specified by board support package, see chbsp_i2c_get_info()).
  5. Start sensor execution.
  6. Wait for sensor to lock (complete initialization, including self-test).
  7. Send timed pulse on INT line to calibrate sensor Real-Time Clock (RTC).

Configuration

Settings include the overall operating mode for the sensor, the maximum range it will measure, internal sample interval (for devices in free-running mode), static target rejection, and object detection thresholds.

Sensor Operating Mode

- Definition: *uint8_t ch_set_mode (ch_dev_t *dev_ptr, ch_mode_t mode)*
- CH_MODE_TRIGGERED_TX_RX – Hardware-Triggered Transmit/Receive Mode
   - The most typical mode for a single sensor is hardware-triggered transmit/receive (Tx/Rx).
   - The sensor’s measurement cycle can be initiated by using a hardware trigger, in which the remote host device asserts and then releases the INT line.
   - The sensor will generate an ultrasonic pulse when it is triggered by the INT line from the host.
   - The sensor then listens for a response (echo) for an amount of time based on the maximum range setting of the device.
   - When the measurement cycle is complete, the sensor will notify the host by asserting the INT line.

Maximum Range

- Definition: *uint8_t ch_set_max_range (ch_dev_t *dev_ptr, uint16_t max_range)*
- The max_range value is the one-way distance to a detected object, in millimeters.

Internal Sample Interval

- Definition: *uint8_t ch_set_sample_interval (ch_dev_t *dev_ptr, uint16_t interval_ms)*
- The sensor will use its internal clock to wake and perform a measurement every interval_ms milliseconds.
- The sample interval setting has no effect on sensors using one of the triggered modes.

Static Target Rejection

- Definition: *uint8_t ch_set_static_range (ch_dev_t *dev_ptr, uint16_t num_samples)*

Setting Multiple Configuration Values

- Definition: *uint8_t ch_set_config (ch_dev_t *dev_ptr, ch_config_t *config_ptr)*

Sensing

Overall cycle of measurements becomes a repeating sequence:

1. Start (trigger) a new measurement cycle. The sensor will emit an ultrasound pulse.
2. Wait while the sensor listens for a receive ultrasonic pulse (either reflected or from a different sensor).
3. The sensor will indicate data-ready by asserting its INT line.
4. The I/O Interrupt Callback routine (registered by ch_io_int_callback_set()) is called by SonicLib.
5. The callback routine calls ch_get_range() or other functions to read sensor data. (The callback function may simply set a flag causing the data read to be done in the regular application loop.)

Lifelong learning

This project created many opportunities for growth. The project we chose presented many different challenges that we needed to solve. The sensor for instance needed to be low power and have a small footprint. The sensor investigation taught us a lot about what is available in terms of sensors and their limitations. The design of the product went through many iterations as we attempted to consider what would provide good comfort while also still being effective. Eventually we settled on the necklace design, suggested by our prof, Ali Tehrani, but only after careful consideration of the alternatives. This helped teach critical thinking skills and tested our creative minds. Overall, this project provided many instances of lifelong learning that will benefit us in the future.

Necessary Knowledge

- Knowledge and experience of the Atmel Studio 7.0 IDE
- Knowledge of embedded systems
- Knowledge of power electronics
- Knowledge of I2C communication
- Knowledge of micro-controllers

- Knowledge of schematic and PCB design
- Knowledge of Bluetooth communications
- Embedded controller development & flashing
- Solder ability & quality

Engineering Tools

- Reading and understanding schematics
- Proficiency in the C programming language
- Altium Designer (20.2.7)
- NI Multisim
- Solidworks
- Arduino IDE
- Soldering station

Testing Methodology

Unit Testing

Module	Item	Test	Pass Criteria
Micro-controller	Switch	1. Attach button switch to GPIO 2. Run program 3. Press and release button	GPIO can detect when input voltage changes from high to low.
	I2C	1. Initialize I2C 2. Run I2C demo program	I2C configured to see devices on I2C bus
	Actuator	1. Put multimeter on GPIO pin 2. Trigger function to turn on GPIO pin	GPIO pin goes HIGH when active, LOW when not.
	Voltage Regulator	1. Confirm REGOUT0 set to value 0x0 or 0x7 2. Check value of VDD pin	VDD pin successfully reads 1.8 V
Ultrasonic	Detecting	1. Confirm if sensor is reading correct distances	Sensor is reading correct distances
	Min range	1. Confirm if sensor can read below distance between neck and chin.	Sensor can accurately read distances from neck to chin.
	Max range	1. Confirm if sensor can detect objects past face.	Sensor can detect objects past face
	Field of View	1. Check field of view in vertical direction 2. Check field of view in horizontal direction	1. Vertical should be as minimal as possible 2. Horizontal is not as necessary
Actuator	Vibrating	1. Input power into actuator	1. Actuator vibrates as desired.

Integration Testing

Interface	Item	Test	Pass Criteria
Microcontroller <> Ultrasonic sensor	I2C	1. Run demo program to detect CH101 on the i2c bus. 2. Send basic message to CH101 and check for reception.	1. CH101 can be detected on the bus. 2. Return message received from CH101
	Power	1. Put multimeter on power rail from uC to sensor	1. Confirm it is ~1.8 V
	Reset	Send LOW voltage from GPIO to CH101	Confirm that the CH101 resets
Microcontroller <> Actuator		Send HIGH voltage to GPIO connected to actuator.	Confirm that the actuator turns ON
Microcontroller <> Power Supply	Power Select	1. Plug in USB cable 2. Plug in battery cable 3. Check status of battery.	Confirm no power is being drawn from battery using a multimeter.

System Testing

Requirement	Item	Test	Pass Criteria
Functionality	Actuator <> Sensor	1. Place object within FoV of sensor 2. Check status of actuator	Actuator is vibrating while object is detected.
Usability	Hand detection	1. Have user wear the device 2. Move hands toward face 3. Check status of actuator	Actuator is vibrating while the object is detected.
	Deterrence	1. Repeat previous test. 2. Confirm that user is deterred from biting their nails.	User is deterred from biting their nails.

Data Collection

Method

The most important part of hand detection is the Field-of-View (FoV). The field of view needs to be narrow enough to not get out-of-bounds hand movements, but wide enough to detect a hand as it approaches the mouth.

Therefore, the method here is to retrieve:

- Max/min values in horizontal plane (x-direction)
- Max/min values in the vertical plane (y-direction)

- Max/min values in the distance plane (z-direction)
- Max/min values in x-y plane when z-distance is 25%, 50%, 75%, and 100% of max.

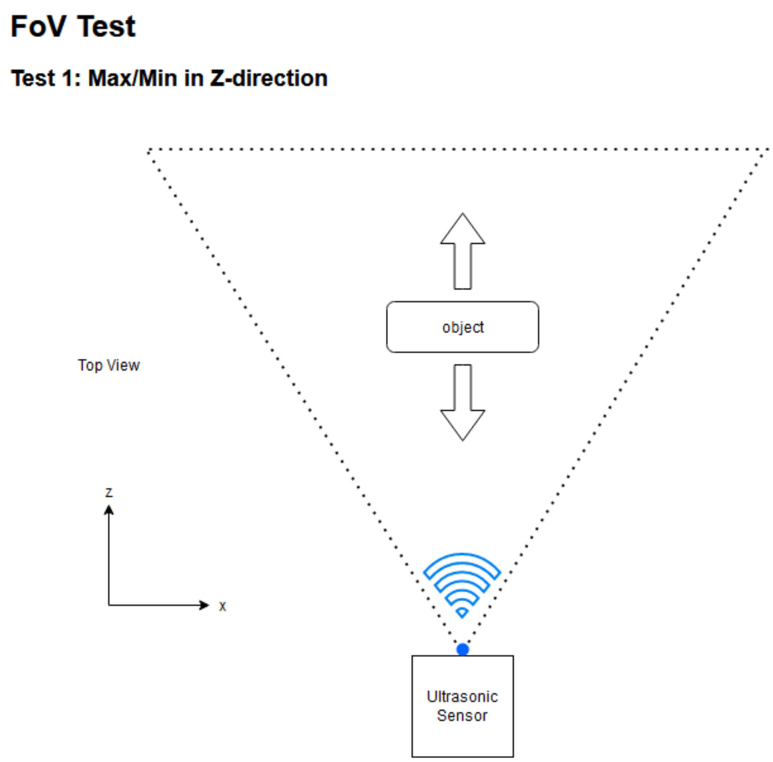


FIGURE 25: FOR TESTING THE Z-DIRECTION

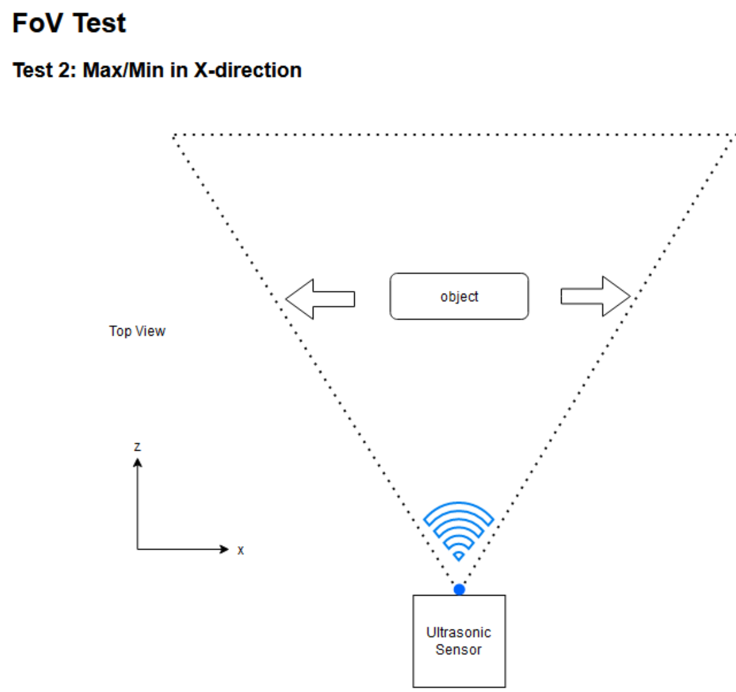


FIGURE 26: FOR TESTING THE X-DIRECTION

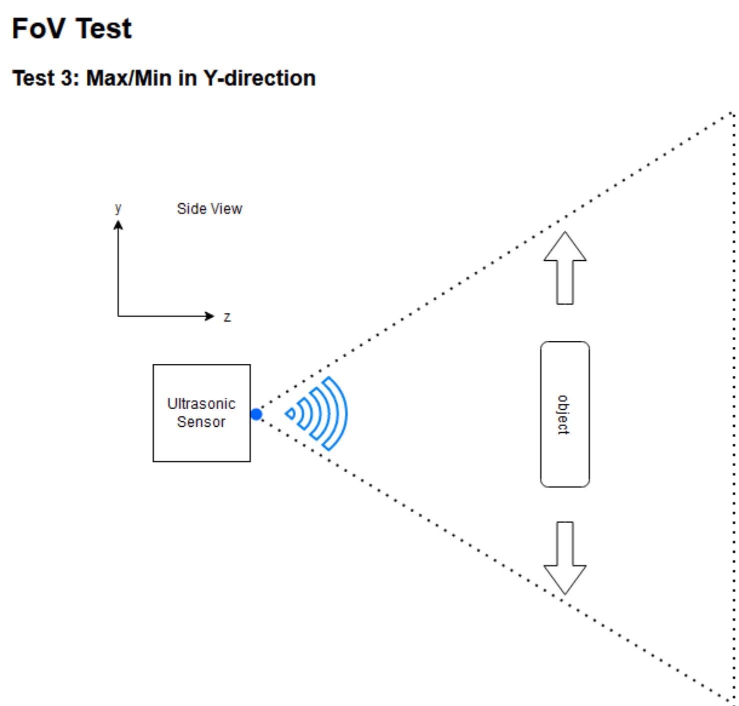


FIGURE 27: FOR TESTING THE Y-DIRECTION

Results

The following image shows how the test setup is set. The farthest horizontal line represents 50cm. Then going inward, we have 37.5cm, 25cm, and 12.5cm. This setup is used to test the max and min of the x axis (left and right). The ultrasonic sensor will then be rotated to test the max and the min of the y axis (up and down).

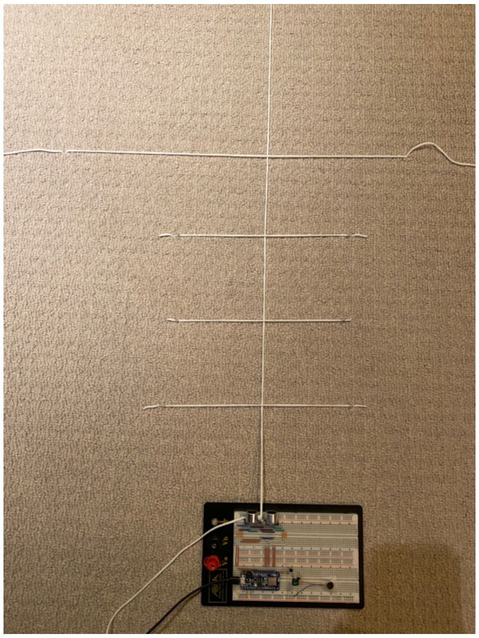


FIGURE 28: X-AXIS TESTING OF ULTRASONIC RANGE

The following image shows the results for the max x at 20 degrees. The ultrasonic sensor was able to read my hand at a 20-degree angle from the center. This angle was confirmed at each horizontal line.

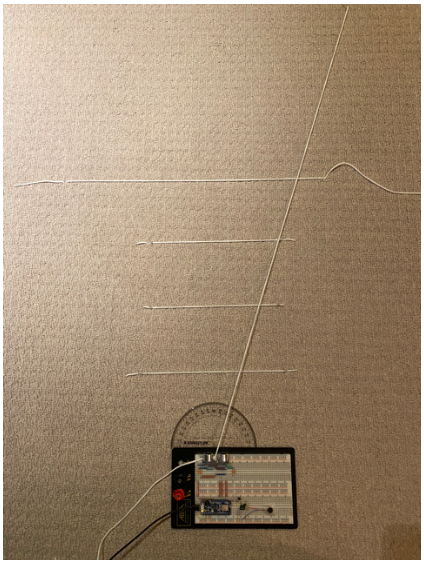


FIGURE 29: MAX X-AXIS ULTRASONIC RANGE (20 DEGREES)

The following image shows the results for the min x at -15 degrees from the center. The test was repeated and confirmed at each horizontal line. It's worth noting that this side of the x-axis is more narrow than the right side.

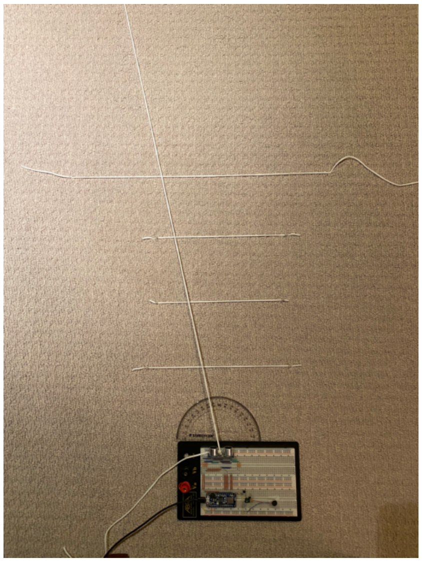


FIGURE 30: MIN X-AXIS OF ULTRASONIC RANGE (15 DEGREES)

The following image shows the results for a max y of 15 degrees. The y axis seems to have a narrower field of view as compared to the x axis. Again, the test was repeated at each horizontal line.

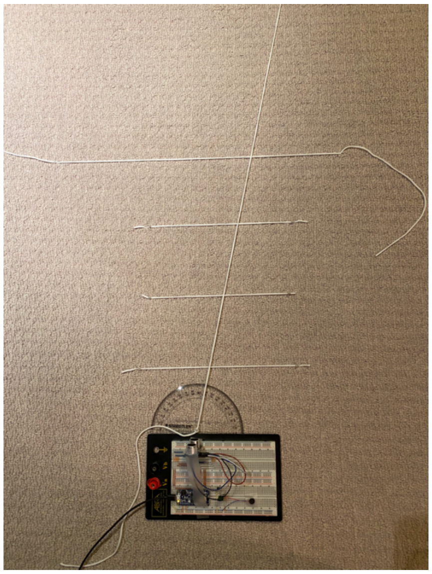


FIGURE 31: MAX Y-AXIS OF ULTRASONIC RANGE (15 DEGREES)

The following image shows the results for a min y of 10 degrees. This side of the y-axis is like the left side of the x-axis, being about 5 degrees smaller than the right side. The test was repeated for each horizontal line.

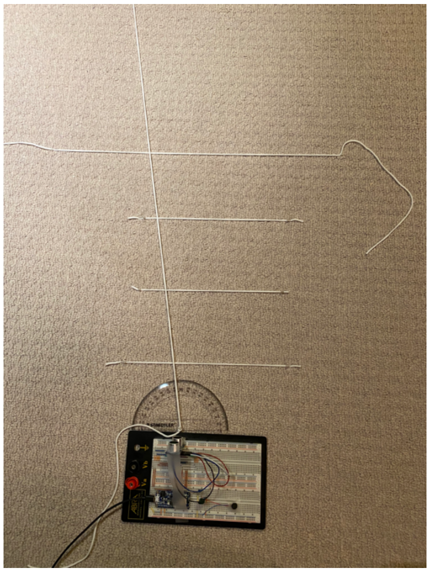


FIGURE 32: MIN Y-AXIS OF ULTRASONIC RANGE (10 DEGREES)

TABLE 4: RESULTS OF DATA COLLECTION TESTING.

Test	Ultrasonic reading (distance measured, angle) (cm, degrees)	Actual (string distance measured) (cm)
max z-direction	50	N/A
min z-direction	1	0
z=0.25 max; max x-direction	(9,20)	(8)
z=0.25 max; min x-direction	(-12,15)	(8)
z=0.25 max; max y-direction	(6,15)	(8)
z=0.25 max; min y-direction	(-6,10)	(8)
z=0.5 max; max x-direction	(22,20)	(20.5)
z=0.5 max; min x-direction	(-26,15)	(20.5)
z=0.5 max; max y-direction	(19,15)	(20.5)
z=0.5 max; min y-direction	(-19, 10)	(20.5)
z=0.75 max; max x-direction	(34,20)	(33)
z=0.75 max; min x-direction	(-34,15)	(33)
z=0.75 max; max y-direction	(34, 15)	(33)
z=0.75 max; min y-direction	(-35, 10)	(33)
z=max; max x-direction	(48, 20)	(45.5)
z=max; min x-direction	(-49,15)	(45.5)
z=max; max y-direction	(43, 15)	(45.5)
z=max; min y-direction	(-49, 10)	(45.5)

Data Analysis

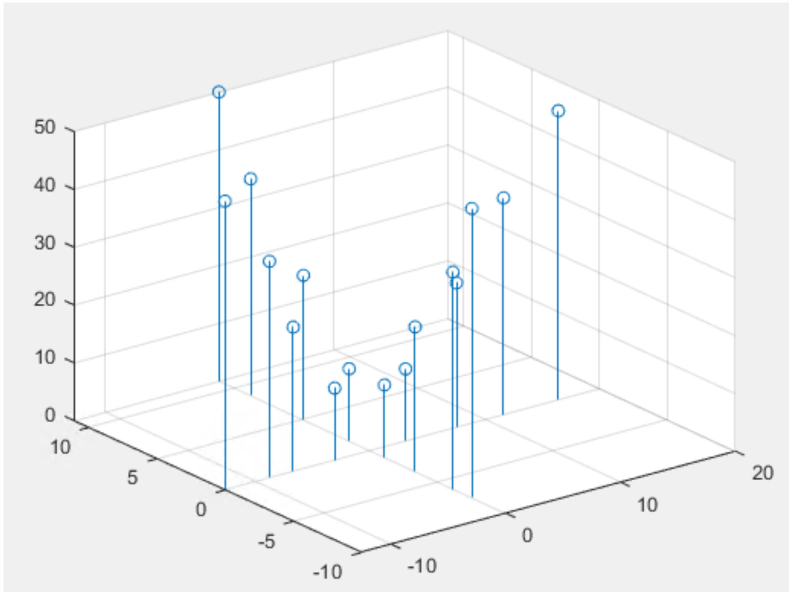


FIGURE 33: VISUAL REPRESENTATION OF TEST DATA.

Figure 33 shows a visual representation of the data, where the ultrasonic sensor is on the bottom facing up, and the points on the graph represent the maximum it can read in each direction. The result is a cone like structure originating from the bottom as the tip and then expanding upwards.

Validation of Data

The information is useful because it shows what kind of angle we need to position the ultrasonic sensor at in order to avoid extraneous objects, like the chin or another person.

Further data could be collected while actually mounted on the neck mount to see where it is currently detecting.

Verification of Data

The data verifies the usefulness of the device by showing that it can measure the hand when it is a certain distance from the mouth and activate the actuator.

Demos

I2C Interfacing Test

A simple I2C scanner program was used to verify the I2C connection with the CH101 ultrasonic sensor. The below image shows the output and we're able to see that both I2C addresses are shown. It's worth noting that each CH101 sensor has one I2C address for firmware programming, and another for application communication. This test passes the micro-controller I2C test.

FIGURE 34: CH101 I2C SCAN TEST

Actuator Demo

The following video leads to a demo of the vibration actuator during the early stages of development. The micro-controller pulls a GPIO pin high, turning on the actuator and then after a delay, pulls that same GPIO pin low, disabling the actuator: https://youtu.be/nz00poK9Utg.

Push Button Demo

The following video leads to a demo of the vibration actuator being toggled by a push button. This demo provides assurance that the vibration actuator can be manually enabled and disabled by the user. This will be useful in instances where the user has their hand or other objects near their mouth, such as eating: https://youtu.be/CGn6KtGBMgg.

Backup Ultrasonic Interface Demo

The following video features a close-up demo of the backup ultrasonic sensor interfacing with the micro-controller to enable and disable the actuator based on how close an object is to the sensor. The proof-of-concept video below shows a more real-world example. However, it's zoomed out, so this demo provides a closer example: https://youtu.be/CGn6KtGBMgg.

Proof-of-Concept Demo

The following video link leads to a proof-of-concept demo for this project. Due to difficulties with the CH101 sensor, a backup sensor is used in its place. With continued progress, the backup sensor would eventually be replaced by the CH101 sensor. You can find the demo here: https://studio.youtube.com/video/BoO6Q6cZCRk/edit.

Conclusion

Overall, this project was a success. We were able to create a proof-of-concept demo that showed how this product is intended to work. Alongside that, we were able to learn and grow many skills during the term. A few of those being our ability to work as a team, our creativity, and many different technical skills.

This project had many challenges. The main one would be not being able to meet up with team members for collaboration. This meant we had to clearly divide up certain parts of the project. This led to difficulties when it came time to integrate these different parts together which caused delays. Therefore, in the future for projects with this limitation it would be worth considering how things will operate in the future so that you can save time and maintain deadlines.

Throughout this project we maintained resourceful and were able to develop a proof-of-concept that supports this project as well as complete the designs for a prototype. Therefore, despite the challenges, this project was a success.

References

[1] "The Science," Pavlok, [Online]. Available: https://pavlok.com/science/. [Accessed 31 05 2021].

[2] M. Sethi, "Pavlok 2 - Change Your Habits with Electric Shock," Indiegogo, 22 May 2017. [Online]. Available: https://www.indiegogo.com/projects/pavlok-2-change-your-habits-with-electric-shock#/. [Accessed 2 June 2021].

[3] M. Sethi, "PAVLOK 3: A mindfulness coach on your wrist," Indiegogo, 27 October 2020. [Online]. Available: https://www.indiegogo.com/projects/pavlok-3-a-mindfulness-coach-on-your-wrist#/. [Accessed 2 June 2021].

[4] "How to Setup your HabitAware Keen2 Bracelet," HabitAware, [Online]. Available: https://habitaware.com/pages/setup-keen2. [Accessed 2 June 2021].

[5] "The Science," Pavlok, [Online]. Available: https://pavlok.com/science/. [Accessed 31 May 2021].

[6] M. Sethi, "Pavlok 2 - Change Your Habits with Electric Shock," Indiegogo, 22 May 2017. [Online]. Available: https://www.indiegogo.com/projects/pavlok-2-change-your-habits-with-electric-shock#/. [Accessed 2 June 2021].

[7] M. Sethi, "PAVLOK 3: A mindfulness coach on your wrist," Indiegogo, 27 October 2020. [Online]. Available: https://www.indiegogo.com/projects/pavlok-3-a-mindfulness-coach-on-your-wrist#/. [Accessed 2 June 2021].

[8] "How to Setup your HabitAware Keen2 Bracelet," HabitAware, [Online]. Available: https://habitaware.com/pahes/setup-keen2. [Accessed 2 June 2021].

[9] TDK InvenSense, "CH101: Ultra-low Power Integrated MEMS Ultrasonic Time-of-Flight Range Sensor with Millimeter Accuracy," TDK InvenSense, [Online]. Available: https://invensense.tdk.com/products/ch101/. [Accessed 03 August 2021].

[10] Gigi, "Introduction to Buzzers: Piezo and Magnetic buzzers," Seeed Studio, January 2021. [Online]. Available: https://www.seeedstudio.com/blog/2020/12/22/introduction-to-buzzers-piezo-and-magnetic-buzzers/. [Accessed 23 July 2021].

[11] T. Store, "tatoko 20PCS 10mmx3mm Mini Vibration Motors DC 3V 12000rpm Flat Coin Button-Type Micro DC Vibrating Motor for Mobile Cell Phone Pager Tablet Household Appliances," Amazon, [Online]. Available: https://www.amazon.com/tatoko-Vibration-Button-Type-Vibrating-Appliances/dp/B07Q1ZV4MJ/ref=sr_1_2?dchild=1&keywords=cell+phone+motor&qid=1624408035&sr=8-2. [Accessed 23 July 2021].

[12] C. Microsystems, "AN-000175-SonicLib-Programmers-Guide-v1.0.pdf," 2020. [Online]. Available: https://invensense.tdk.com/download-pdf/an-000175-chirp-microsystems-soniclib-programmers-guide/. [Accessed 22 07 2021].

[13] tatoko store, "Amazon," [Online]. Available: https://www.amazon.com/tatoko-Vibration-Button-Type-Vibrating-Appliances/dp/B07Q1ZV4MJ/ref=sr_1_2?dchild=1&keywords=cell+phone+motor&qid=1624408035&sr=8-2. [Accessed 4 August 2021].