Tongue Based HRI

Final Project Write-Up

Jared Leitner – 804436513

Ambarish Kowluri – 704423441

Mayur Bhandary – 904487050

Abstract

There is a need for devices that allow users to control systems discreetly. Our sensing device gives the user the ability to control a system, in this case a simple video game, using their tongue. The project consists of a custom-designed PCB with a mounted pressure sensor, a WiFi communications board, and LiPo battery. These components are packaged together such that they can attach to a mouthpiece that is worn by the user. Our mouthpiece provides hands free control and a discreet means of human robot interaction (HRI). A future goal for our tongue drive system is for it to be integrated into the daily lives of individuals as a wearable technology.

Introduction

Conventional methods for human robot interaction include, but are not limited to, voice control, physical manipulation (joystick, etc.), the use of cameras and image processing, and directly programming the robot to perform a task. One similarity between these means of HRI is that user must actively participate in the interaction for communication to occur. Also, from an outside perspective it is obvious that the user is taking part in controlling a robot. Voice control requires the human to make audio commands, while direct manipulation and image processing require physical movements from the human. In the case of programming, there may not be any real time interaction or control occurring between the human and the system. As the proximity of technology increases, robots will become more integrated into the daily lives of humans. These traditional methods of HRI will remain relevant, however, not feasible in all situations. This project aims to address this problem by providing the human a more discreet means of control.

This discreet control is made possible by this projects novel sensing device which gives the user the ability to give commands using their tongue. The device makes use of a pressure sensor to detect tongue movement and microcontroller with WiFi capabilities to send the sensor readings to a separate device. The final assembly also includes a rechargeable LiPo battery, providing longevity to this device. A simple computer game, Pong, is used to showcase the control capabilities of this mouthpiece. The software for this game was found online and modified in order to receive commands from the mouthpiece itself.

The remainder of this document details the design and implementation of this device, as well as the major problems experienced along the way. It includes the key experiments and takeaways from this project and discusses future designs and applications for the device.

Methods / Design Challenges

This section will serve as documentation of the detailed procedures used to produce this project and the design challenges that were encountered. It will be a resource for our own future reference as well as for those who wish to expand on our project.

InstaMorph Scaffolding

Creating the mouthpiece scaffolding was the first step in this project. The dimensions of the scaffolding determine the physical constraints that the electronics are subject to. The choice of material used to form the plastic is known as InstaMorph. It is a thermally activated plastic that is highly moldable once heated to 150°F. This method was used over 3D printing because of its ease of use and its rapid prototyping capability; 3D printing would not have given us the ability to mold to the shape of the mouth in the same way that InstaMorph does. It is important to note, however, that InstaMorph is not considered “food safe” (Team, InstaMorph). It is up to the user to take the risks involved with the possibility of ingestion. Neither InstaMorph nor Team Carrot will take on any liability related to any adverse effects that may occur due to the use of InstaMorph in the mouth. With that being said, none of the subjects that have created mouthpiece scaffoldings using InstaMorph have reported any adverse side effects.

The first step is to purchase a boil and bite mouth guard commonly used in sports. This will create a protective layer around the teeth which and create a base for the InstaMorph scaffolding.

The next step in creating the InstaMorph scaffolding is to heat the InstaMorph via one of two methods: water or heat gun. The first method is by heating approximately 1 liter of water in a pot to 150°F. Once the water is at the appropriate temperature, place a handful of InstaMorph beads into the pot and allow the beads to change color from white to clear. The beads will coagulate to form a bolus which will be used for molding. A heat gun can be used instead of hot water which is a better method when dealing with circuitry that should not be exposed to water. After heating, carefully take the bolus of InstaMorph out of the pot using metal tongs and place it on a clean surface. The InstaMorph will be completely malleable for only a few minutes. Using a strong, cylindrical, glass tube, roll the InstaMorph into a flat sheet like a pizza (we found that an empty beer bottle works very well for this). Once the InstaMorph is flat it will cool and harden more rapidly than when it was a ball. For this reason, the flat sheet of InstaMorph should be reheated for further molding. Now, there is a flat, clear sheet of InstaMorph that can be placed in the roof of the mouth to create a scaffolding. Carefully cut half a circle out of the InstaMorph sheet using scissors. Make sure that the half circle is large enough to cover the area encompassed by upper mandible. With the boil and bite mouth guard in place, take the half circle and press it up against the roof of the mouth making sure to cover the edges of the mouth guard. Pack the InstaMorph into the roof of the mouth making sure that no area is left exposed. Note: Make sure that the InstaMorph is at a suitable temperature to do this. Team Carrot is not responsible for any burns or related injuries that may occur during this process. Keep the InstaMorph in the mouth until it completely hardens which will take about 10 minutes. Once the InstaMorph hardens, the scaffolding is complete. The following photograph shows the InstaMorph scaffolding formed around a boil and bite mouth guard:

Figure 1: Mouth guard scaffolding

Mouthpiece PCB Design

The 2-layer PCB design for the mouthpiece was scoped out through the ESP01's capabilities. The ESP01 microcontroller is no different from the ESP8266. The microcontroller is rather small and contains similar components to those of the ESP8266 such as the WiFi module. However, there are fewer GPIO ports on the ESP01 to use for the project. The circuit design for the mouthpiece minimizes the ports needed from the ESP01 to keep the overall mouthpiece PCB design small. The PCB is designed such that the ESP01 can be inserted directly into it to provide for a compact product. A small PCB was emphasized to substantially reduce clutter in the mouth and the ability for the tongue to touch the sensor without hassle. The image below shows the top and bottom Eagle CAD designs for the mouthpiece PCB:

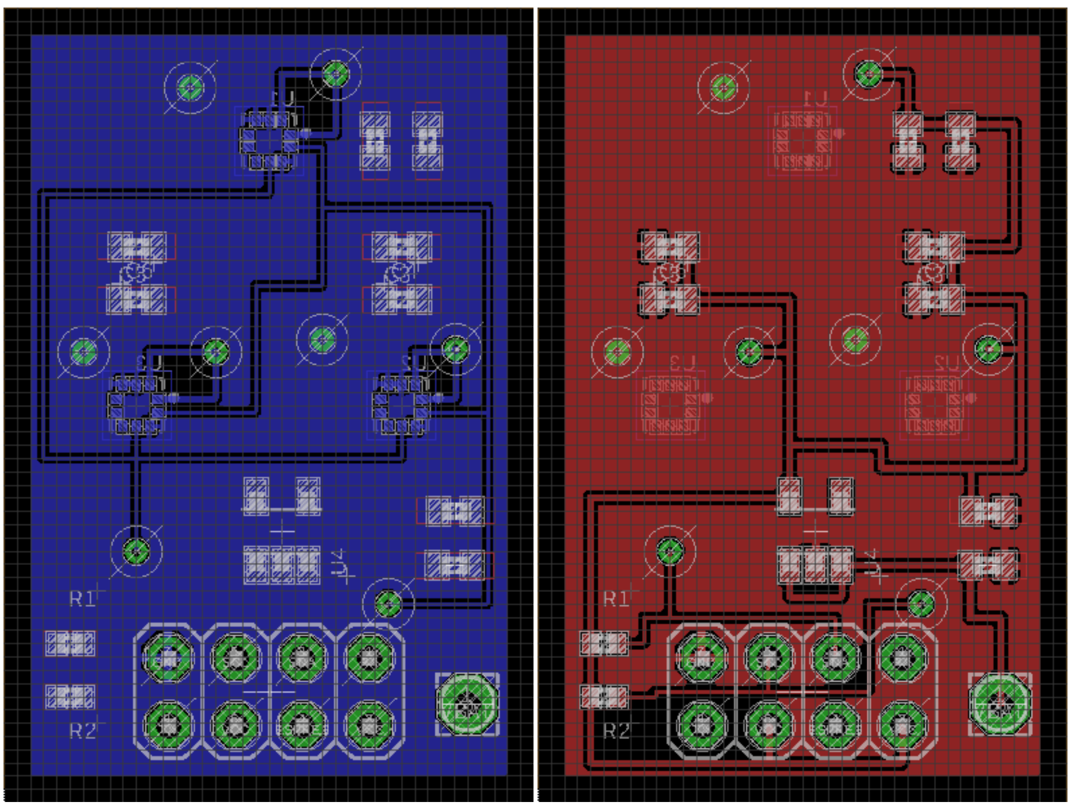

Figure 2: The bottom layer (left) and top layer (right) of the mouthpiece PCB

As can be seen, the mouthpiece PCB design consists of through-holes to accommodate the ESP01. The three LPS22HB atmospheric pressure sensors are aligned symmetrically such that the traces are spread out and can be connected throughout the board using vias. Decoupling capacitors were placed alongside each pressure sensor to reduce noise. These capacitor values were calculated to be 100nF and 0.1uF. In addition, 1k Ohm pull-up resistors were utilized for both the SDA and SCL lines that communicate to the ESP01 microcontroller. A 3.3V linear voltage regulator (TPS73633DB) was implemented to power the ESP01 and power the LPS22HB pressure sensors. These passive components along with the linear voltage regulator are small surface mount components. On the other hand, the battery connector is a through-hole connection. This is because Team Carrot decided on using a rechargeable Lithium Ion Polymer battery. In order for this to be done, the battery's VCC and GND ports would need to be reachable. Nevertheless, the VCC connection from the battery on the bottom right corner of the PCB's top layer can be pulled out using a technique that is described later. The following image displays the ESP01 mounted on the mouthpiece PCB:

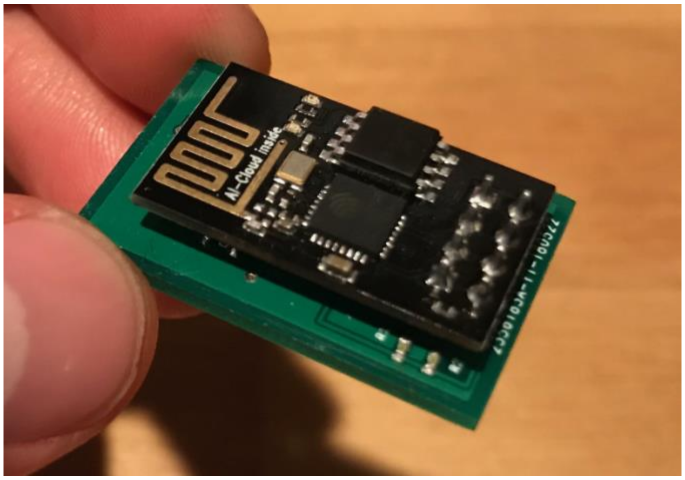

Figure 3: ESP-01 WiFi board (top) attached to mouthpiece PCB (bottom)

The through-hole connections for the ESP01 also serve another purpose. Wires can be pulled from these connections to provide for external connections when the product is finished. This strategy will be shown later. The bottom of the PCB in the image above contains the sealed pressure sensor. The mouthpiece PCB design accounts for the placement of the product in the mouth. In this case the ESP01 and LiPo battery will be stacked on the mouthpiece PCB and the tongue can only touch the bottom of the PCB, which only holds the mounted pressure sensors.

An error that was discovered later within the progress could have been fixed with adjustments to the mouthpiece PCB design. Team Carrot had noticed that the LPS22HB pressure sensors could not be used together because they held the same data address. This meant that retrieving data from them would not be possible and only one sensor reading could be obtained. Thus, the resulting product used just one LPS22HB mount. The resolution for this problem is to simply pull one of the pads on a sensor's mount (SA0) to ground. However, this will only allow at most two LPS22HB pressure sensors to be utilized.

Programming the ESP8266-01

This particular ESP does not contain a micro USB interface to program with. For this reason, there must be a bridge to convert from the USB interface of the computer, and the serial programming protocol of the ESP. An Arduino Uno was used as the bridge in this project, but a TTL to USB converter is also a common method of programming.

In order to use the Arduino as a Bridge, the following connections must be made. Firstly, the most important step in programming the ESP01 is to properly power it. Since the ESP01 can consume up to 170mA of current, the Arduino's 3.3V pin is not sufficient to power it. To power the ESP01 during programming an LM317 adjustable voltage regulator and a 9V battery were used to create the appropriate 3.3V source. If programming fails, it is likely due to inadequate powering capability of the source. Next, make sure that there is a common ground among all the circuits. That is, connect the ground on the Arduino to the ground on the ESP01 to the ground of the voltage regulator circuit. The Arduino reset pin must be grounded to keep code from executing on the Arduino itself. The RX and TX pins on the Arduino must be directly connected to the RX and TX pins respectively on the ESP01. Some sources have suggested that the TX pin should have a logic level converter to convert between the Arduino's 5V logic to the ESP's 3.3V logic, however we have found this to not be the case. The ESP01's Pins are capable of tolerating 5V logic. Next, the CH_PD pin on the ESP01 should be pulled directly to VCC. Finally, the GPIO 0 pin on the ESP01 should grounded to put it into programming mode. This leaves GPIO2 and reset on the ESP01 floating. The reset can be used to reset the code by connecting it to ground. Resetting the code means starting the main loop from its first iteration again. GPIO2 will be connected to the SCL line on the I2C communication bus.

Once the circuit is properly assembled, power the regulator and have the Arduino IDE open with the Arduino plugged into the computer via USB cable. In the Arduino IDE, make sure that the ESP8266 board is properly installed in the board manager using the following tutorial:

https://www.hackster.io/harshmangukiya/how-to-program-esp8266-with-arduino-uno-efb05f

Next, go to tools -> board and select Generic ESP8266 module. Make sure that the port number is set correctly and upload the code. This setup can be used for debugging without the Mouthpiece PCB attached. In this configuration, GPIO0 can be disconnected from ground and connected to the SDA line of the I2C bus and GPIO 2 can be connected to the SCL line of the I2C bus.

Mouthpiece Pressure Sensor

This project implements a novel sensing solution that makes use of the LPS22HB Atmospheric Pressure Sensor. When determining which sensor to use, the main factors that were considered were size, due to the geometric constraints of the human mouth, and sensitivity, in order to increase the resolution of the pressure readings. The dimensions of the LPS22HB (2.0 x 2.0 x 0.76 mm) along with its wide range in pressure readings (260 to 1260 hPa) were the primary factors that lead to its selection as our sensor of choice.

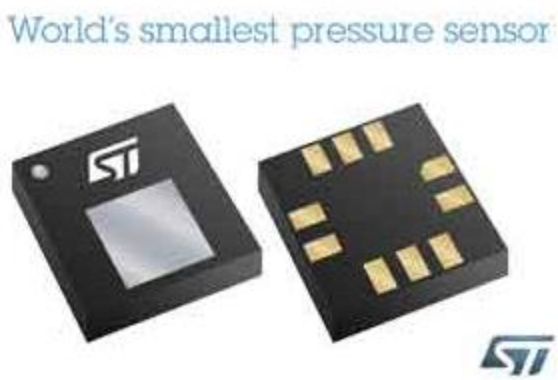

Figure 4: STMicro's LPS22HB Atmospheric Pressure Sensor (STMicro)

The LPS22HB is an atmospheric pressure sensor, meaning it cannot measure any pressures directly applied to it. However, it can still be used for force sensing applications if the sensor is sealed off from its surrounding atmosphere. This sealing creates a new, miniature atmosphere around the sensor. When the user presses on this seal, the pressure inside the seal increases which can be measured by the LPS22HB. Before purchasing this sensor, Team Carrot found a demonstration in which this sensor was used for a force application. This demonstration involved the user drawing lines of varying widths onto a tablet using a stylus. The LPS22HB sensor was housed at the tip of the stylus and measured the pressure the user applied to the tablet. As the user applied greater pressure to the tablet, the width of the line increased. This demonstration was an important proof of concept that showed this sensors ability to be used in force sensing applications.

In order to witness this ability first hand, Team Carrot's LPS22HB development board was sealed in a standard Ziploc bag. This bag was then compressed and the sensor was able to successfully measure the change in pressure. Sealing off the LPS22HB in a clean and compact way was not a trivial task, primarily due to the size of the PCB on which it was mounted. The PCB's size is again limited by the geometric constraints of the human mouth, meaning there is not a lot of room to attach a seal for the pressure sensor. The seal used in the final design came from the rubber membrane found within a keyboard. As shown in the photograph, this rubber piece consists of multiple bubbles that push the keyboard keys back into place once they have been pushed down. These bubbles fit very well over the LPS22HB and were small enough to be attached to the PCB.

The method for sealing involved cutting out one of these bubbles such that there remained a lip around the bubble that allowed for a larger area to be glued down around the sensor. In order to test if this means of sealing would work, one of these was attached to the development board using superglue. Once the glue had set, the bubble could be pressed down and the change in pressure within was successfully measured. On the final product, an FDA approved silicone sealant was used in place of the superglue.

LPS22HB Atmospheric Pressure Surface Mounting

The surface mounting for the LPS22HB is not a trivial task. Team Carrot ordered a stencil in preparation for the pressure sensor mounting. Yet, the solder paste that was used became viscous and could not dry enough to establish a connection between the pads on the PCB and the sensor.

Through internet research, another method was implemented. A microscope is recommended for mounting these small pressure sensors.

The process began by adding a flux medium to the sensor pads on the mouthpiece PCB. Then, solder was placed lightly around the pads. The flux medium allowed the solder to distribute freely throughout the respective pads without making connections between pads. At this point, the solder should have solidified and has to be melted again in order for the sensor to attach to the PCB. Thus, a hot air gun is utilized to accomplish this task. The LPS22HB is placed on the pads and aligned while the hot air gun heats the solder at a high setting. The pressure sensor will slide onto the pads as the solder melts and the flux medium ensures that there are no interconnections among each pad.

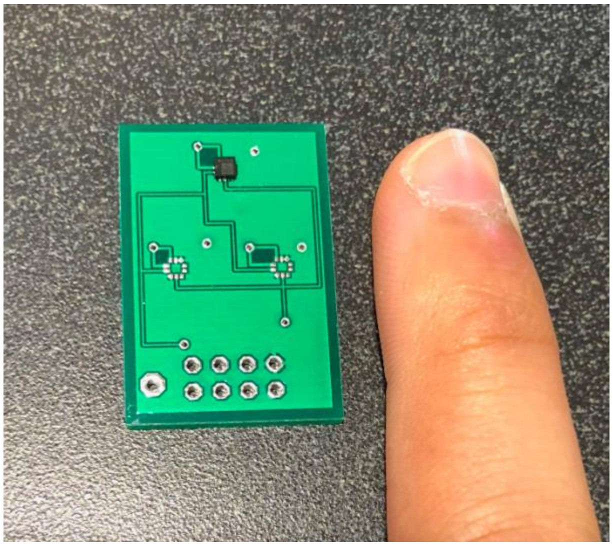

Figure 5: The bottom layer of the PCB with LPS22HB mounted on board

The image above shows the LPS22HB soldered onto the PCB using the method described previously. As can be seen, the pressure sensor with respect to an index finger is substantially smaller, which explains the use of a microscope for this process. If not aligned correctly or if damaged, the LPS22HB can be removed from the PCB similarly. Hover the hot air gun at a high setting over the sensor to melt the solder, and then gently lift sensor off PCB. The solder on the pads can be cleaned using the flux medium again. This strategy for mounting the pressure sensor creates flexibility and leaves room for error.

Battery / On Board Power

In order to make this mouthpiece a discreet device that allows for hands-free control, on board power was a requirement. Team Carrot chose to use a Lithium Ion Polymer battery that was rated at 3.7 V and 150 mAh. This was a high enough voltage to power the 3.3 V regulator and could provide power for a sufficient amount of time before needing to be recharged. The following is a photograph of the battery:

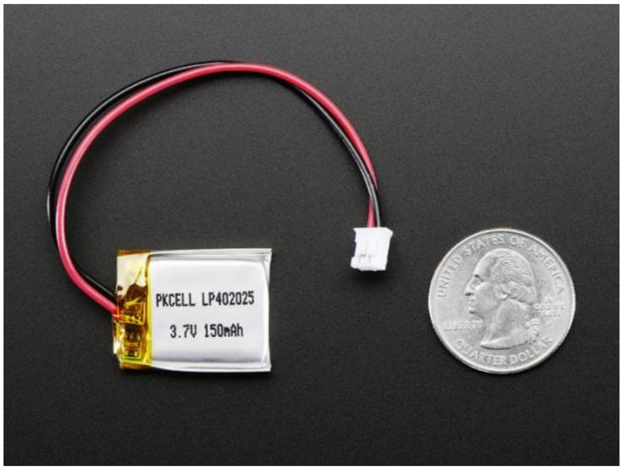

Figure 6: LiPo Battery used to power the mouthpiece (Adafruit Industries)

The physical size of the battery was also an important factor in its selection. As can be seen, the battery is roughly the same size as a quarter, meaning it could fit in the mouth along with the other circuitry. This battery came with a charging board that allowed it to be charged via USB.

Coin batteries were also considered for this project, however, multiple would need to be placed in series in order to produce adequate power. Another reason this option was discarded was because the batteries would need to be stacked and this would take up too much space within the mouthpiece.

Ultimately, the LiPo battery worked well and provided enough power to last the entirety of our class's demonstration session. During this session the mouthpiece was continuously powered for about two hours and did not require a single recharge.

Pin Connectors

Team Carrot's design for the pin connectors encompasses the idea that the Lithium Ion Polymer battery has to be rechargeable and the ESP01 reprogrammable. The method decided involves the use of a bottle cap and female headers. The top of a travel-sized shampoo bottle allowed room for wires to be pulled out from the PCB. This means that the battery can be recharged anytime and the ESP01 can be programmed. The bottle cap that will cover these wires when in the mouth guarantees air-tight sealing.

Figure 7: Headers allowed for external connections to important pins on the PCB

The image above displays the wires fitted into the bottle top. The bottle cap will screw over the headers when the device is in the mouth. The wires are color-coded using heat shrink and the mapping is shown below. This mapping helps keep track of the external and internal connections when opening the device.

Color- External Conn. (Internal Conn.)

Red- VBAT (Battery)
Black- GND (Battery)
Middle Black- GPIO0 (SDA)
White- GND (Board)
Middle White- (RESET)
Orange- TX (TX)
Green- RX (RX)
Blue- (VBAT on Board)

Figure 8: Pin Connector Color-Wire Mapping

In order to power the device, small jumper cables were constructed that connected the battery's positive and negative terminals to those of the PCB.

Overall Assembly

The finished device consists of the LiPo battery, ESP-01 WiFi board, mouthpiece PCB with pressure sensors, and wires that allow for external connections to the battery and PCB. All of these components were stacked together and encased in InstaMorph. The following four photos display the initial configuration of the components and how InstaMorph was added iteratively.

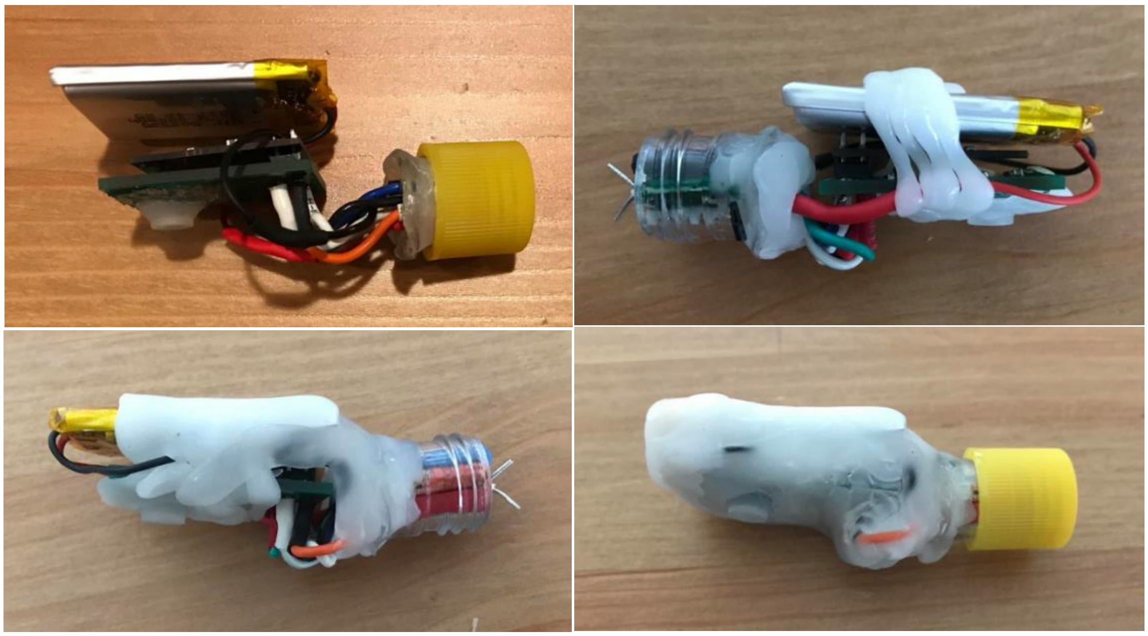

Figure 9: Progression of encasing components in InstaMorph

The top left photograph shows the layout of the components, with the battery on top, the ESP-01 and PCB underneath, and the pull out connections going into the bottle cap. The small white cap at the bottom of the PCB is the keyboard bubble that covers the LPS22HB sensor. The other three photos show how the InstaMorph was added to cover up all the electronics and provide an air-tight seal. It is important to point out the keyboard bubble was not encased in InstaMorph so that it could still be pressed down and change the pressure inside the bubble.

The following photo shows our finished mouthpiece and the LED indicated that it is powered. The physical size of this device was larger than initially thought, so it could not be attached to the mouth guard scaffolding described earlier. Instead, the user would insert the device into their mouth and hold it in place by placing their lips around the yellow bottle cap.

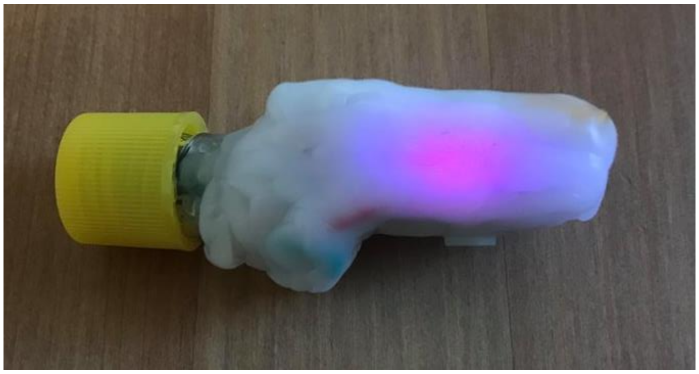

Figure 10: Final mouthpiece with LED indicating power is ON

Using this mouthpiece, the user was able to play a game of Pong by controlling the motion of the on-screen paddle. A mapping from the pressure readings to the on-screen position was established. If the user pressed harder on the sensor, the paddle moved up on the screen. As the user applied less force, the paddle moved down. The following is a screen shot of the game environment:

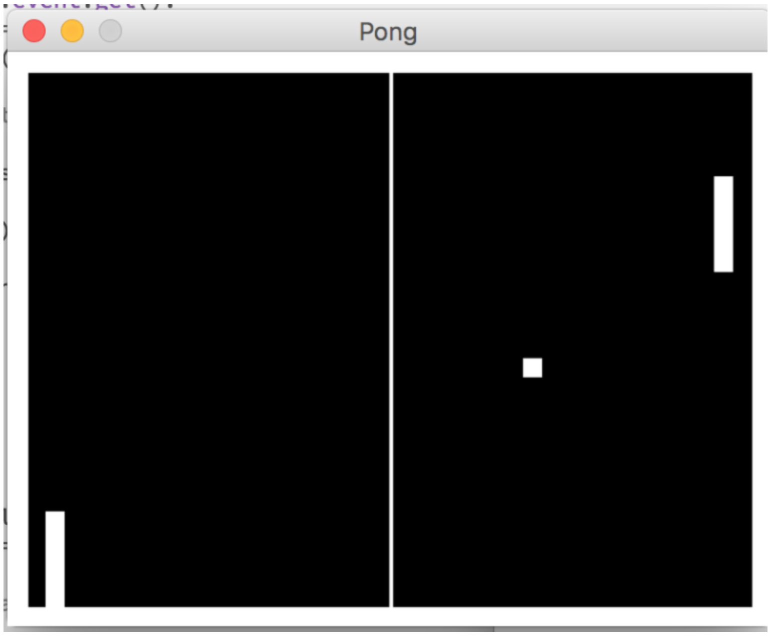

Figure 11: Pong game environment - mouthpiece controlled movement of left paddle (Euclio)

Software and Videos

All of the code and video demonstrations for Team Carrot's mouthpiece can be found at the following GitHub repository: https://github.com/mayurbhandary/TongueControl

Key Experiments and Results

Wireless Communication from Inside Mouth

The purpose of this test was to determine whether or not a WiFi signal could be successfully transmitted from an ESP board inside the mouth. The human body primarily consists of water, and since water is a high attenuator of electromagnetic waves, it was important to determine early on if this form of wireless communication was feasible. Seeing that the goal of this project was to construct a discreet tongue drive system, it was necessary for the communication to be wireless.

The test was carried out by wrapping an ESP8266 board in Saran wrap and placing it inside the mouth. This is not the exact board that is used in the finished device, but it utilizes the same WiFi module making it appropriate to use for this test. A simple program that creates a WiFi network was uploaded to determine whether a connection could be made. The program was run with the ESP inside the mouth of a Team Carrot member and the correct WiFi network appeared on all laptops and phones within the proximity of the board. The following photograph shows a Team Carrot member with the Saran wrapped board in his mouth.

Figure 12: Team Carrot member with Saran wrapped ESP8266 in mouth

A wire coming from the mouth can be seen in the photograph, however, this connection is only there to power the ESP8266 and does not affect the wireless communication. The success of this test allowed Team Carrot to continue down the path of constructing a wireless tongue drive system using the ESP. If this test was not passed, other means of wireless communication, such as

Bluetooth or Zigbee, would have been explored. If these had also been unsuccessful at establishing a connection from inside the mouth, the entire project and problem statement would have had to been altered.

InstaMorph Submersion Test

InstaMorph is the moldable plastic that was mentioned earlier in this document. The purpose of this test was to determine whether this material was waterproof and capable of creating an airtight seal. The inside of the mouth is an unfriendly environment for electronics. Seeing that all of the electronics for this project would be contained within InstaMorph, it had to be determined whether or not the plastic could adequately isolate the electronics from saliva with in the mouth.

The setup for this test was borrowed from the company Aculon, which deals with waterproof coatings for microelectronics. The following circuit, excluding the voltmeter, was used for this test:

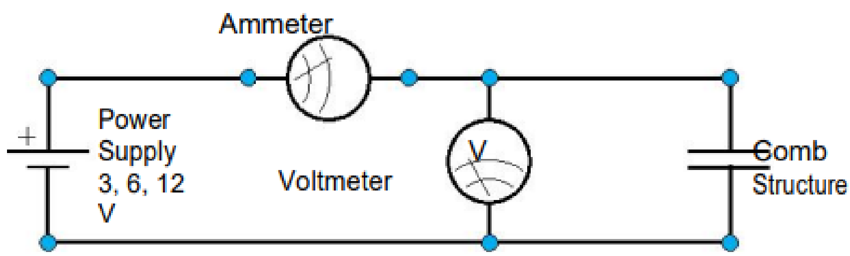

Figure 13: Circuit schematic used for submersion test (Aculon)

The comb structure pictured in the schematic acts as a capacitor, and each of its terminals has multiple overlapping segments. This structure is displayed in the following photograph, and was printed using an OtherMill.

Figure 14: Comb structure printed using an OtherMill

There are no direct connections between the terminals of this structure, hence its capacitor-like nature. This structure was encased in InstaMorph and submerged in a saline solution to determine if liquid would seep in and cause any short circuits. The ammeter would read an increase in current draw from the power source if any shorts did occur. The overlapping branches are put in place to increase the chances of a short occurring between the terminals. It may seem counterintuitive to increase the likelihood of a short circuit occurring, but it makes this test harder to pass and provides more insight into the waterproofing capabilities of InstaMorph.

The comb structure was submerged in the solution both with and without the InstaMorph casing, and the current draw for each case is displayed in the following table:

Test Procedure	Current (mA)
Without InstaMorph Casing	3.54
With InstaMorph Casing (Immediately)	0.09
With InstaMorph Casing (After 5 min.)	0.11

Figure 15: Current draw with and without InstaMorph casing

It is evident from these results that the InstaMorph was effective at preventing any liquid seeping in and causing shorts. With the InstaMorph casing, the current draw was 2.5% that of the draw without the casing. The current draw of comb structure with the InstaMorph casing was recorded after five minutes of complete submersion and only marginally increased by 0.02 mA. These results indicated that InstaMorph could successfully seal off the electronics from the saliva within the mouth. Also, the mouth is not completely filled with saliva, meaning the mouthpiece would not be totally submerged. The fact that InstaMorph could provide an adequate seal when it was fully submerged provided confidence that it could do the same once placed in the mouth.

Atmospheric Pressure Sensors vs Force Sensitive Resistors

Another sensing option that Team Carrot investigated was force sensitive resistors (FSRs). The FSR became a backup option when the ability to seal off the atmospheric pressure sensor such that it could be used for this force application was unknown. Although the atmospheric pressure sensor was the sensor of choice (due to the fact that is could be properly sealed with the keyboard bubble), Team Carrot setup and tested the FSR and the results are worth noting. The following is a side by side comparison of the two sensors:

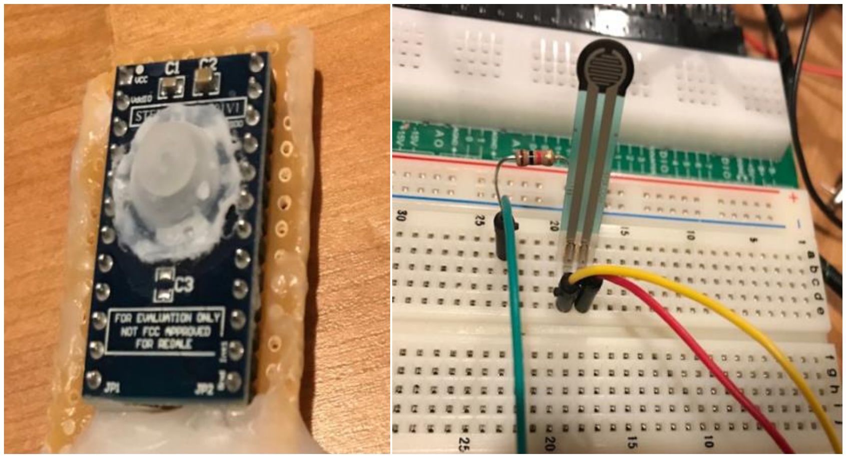

Figure 16: Atmospheric pressure sensor with keyboard bubble (left) vs. force sensitive resistor (right)

The setup for the force sensitive resistor comprised of a simple voltage divider circuit. The FSR was placed in series with a 1k Ohm resistor and an Arduino analog input pin was connected to the resistors' shared node. As the FSR was pressed, its resistance decreased and the voltage at the middle node increased. The following is a screen shot of the Arduino's serial monitor when the FSR was quickly pressed and released.

```
Analog reading = 0
Analog reading = 0
Analog reading = 5
Analog reading = 46
Analog reading = 99
Analog reading = 152
Analog reading = 181
Analog reading = 205
Analog reading = 209
Analog reading = 168
Analog reading = 14
Analog reading = 0
Analog reading = 0
```

Figure 17: Variation in FSR analog readings when it is pushed and released

It is evident from the analog readings that the FSR is sensitive and can provide a large variation in readings. The readings clearly show a spike when the sensor was pressed as they quickly increase and then decrease back to 0 once it was released. This provided Team Carrot with a great backup plan in case the LPS22HB sensor did not work out.

Ultimately, the LPS22HB was successfully sealed off from the surrounding atmosphere and selected as the sensor of choice. The two main reasons for this were that it was smaller in size and that it had its own analog to digital converter (ADC). The microcontroller board (ESP-01) only had digital input pins meaning the LPS22HB could be directly connected without any extra circuitry. If the force sensitive resistors were to be used, a separate ADC would need to be added to the PCB. Although this was possible, the simplicity of the atmospheric pressure sensor connections is was gave it the edge. Team Carrot is confident that FSRs could be used in creating a pressure sensor tongue drive system and believe it is worth attempting.

Future Work

Team Carrot's final mouthpiece provides a sufficient proof of concept that a tongue drive system using pressure sensors is feasible. The successful demonstration of using the mouthpiece to play the video game Pong proves that individuals can use their tongues for controlling other devices or programs. Although this first iteration resulted in a functioning device, there are many further updates that could make the mouthpiece more practical for everyday use. These updates include a decreased physical size, lowered power consumption, altered PCB design, increased number of sensors, and addition of more devices to control.

Physical Size

The majority of the volume of the mouthpiece comes from the battery and the scaffolding. In order to minimize the volume of the mouthpiece, a smaller battery and tighter molding would be required. This could be achieved if the PCB was less power hungry. In order to complete this project within the appropriate scope, we used the ESP-01 to create a wifi network to access sensor data. This board consumes a lot of power when producing the WiFi signal which could be mitigated through the use of Bluetooth Low Energy (BLE). The BLE approach would decrease the range of transmission, but for the applications of this project long range communication is unnecessary. By reducing the power consumption in this manner, a smaller battery may be employed thus reducing the overall form factor by orders of magnitude. In terms of the scaffolding, a custom fitted dental retainer would form a better housing for the internal circuitry. It would be instructive to consult with the dental school at UCLA to move further in this direction. Another size optimization would be to place the microcontroller and sensors on the same pcb instead of making a sandwich between 2 separate PCBs. Furthermore, the use of force resistive sensors may allow for a flatter finish than the atmospheric pressure sensors since they do not require an artificial atmosphere bubble.

Power Consumption

As mentioned earlier, a lower power consumption would drastically decrease the size of the mouthpiece because a large battery would no longer be required. Power consumption was not heavily considered during the first iteration of this project because it was mainly a proof of concept. However, there are multiple ways by which the power consumption could be lowered. Firstly, the use of BLE would reduce power consumption greatly. Second, the microcontroller's sleep mode could be utilized to conserve power while the mouthpiece is not in use. This could be defined as times when the measured pressure is below a certain threshold value and should not be transmitted.

PCB Design

The modified mouthpiece PCB design involves incorporating multiple pressure sensors on board. The SA0 port on one pressure sensor needs to be pulled to ground in order for the other pressure sensor to have its own unique data address during programming. Once the PCB is made, it is difficult to modify the board physically whether it could be through removing solder mask or using wire. The sensors are small so altering the board physically is not a feasible solution. Thus, working with and modifying the PCB design on Eagle CAD should allow more than one LPS22HB to be used. The space between sensors are accommodated based on the bubble that seals each pressure sensor.

Increased Number of Sensors

An increase in the number of pressure sensors allows the device to control different aspects of a system. For the Pong game, Team Carrot decided that one pressure sensor would be needed since the paddle follows one direction and can be defaulted in the other. However, if multiple pressure sensors were used, a more complex game or system could be implemented. This can only happen if the PCB was altered such that more than one pressure sensor could be programmed. For example, a prospective application would be controlling the movement of an electric wheelchair with the tongue drive system. Using more than one sensor can provide a greater means of control such as moving the wheelchair forward, left, and right.

More Devices to Control

Controlling more devices with tongue based HRI allows the use of the tongue to be versatile in terms of robotics. Not only can tongue based HRI control a simple video game, but also it can target larger electromechanical systems such as a wheelchair. The tongue is directly controlled by the cranial nerve, which is often times not affected for those who are paralyzed or disabled. This means that those disabled would be able to use their tongue to control a variety of their devices. However, more complex systems require a more complex tongue drive system. Team Carrot believes that the versatility of the LPS22HB pressure sensor allows for these complex systems to be controlled with the tongue.

Conclusion

Tongue based HRI has many foreseeable uses and is very feasible to implement. Specifically, this project has demonstrated the ability to achieve a device that allows control through a user's tongue discreetly. The form factor of the device was small enough to mostly fit inside of the user's mouth with only a slight protrusion out of the lips. With some refinement, it will be possible to achieve a mouthpiece that is completely concealed in the upper mandible much like a retainer. This project has also demonstrated the efficacy of atmospheric pressure sensors as a means of measuring tongue inputs. As a result, a novel method of sensing has been explored and developed through the completion of this project.

Bibliography

Aculon. “Waterproofing of Printed Circuit Boards and Electronic Components from Water and Salt Water by Using Aculon NanoProof Coatings for Microelectronics.” 2015.

Adafruit Industries. “Lithium Ion Polymer Battery - 3.7v 150mAh.” Adafruit Industries Blog RSS, www.adafruit.com/product/1317.

“Euclio (Andy Russell).” GitHub, github.com/euclio.

STMicro. “LPS22HB adapter board for a standard DIL 24 socket.” 2016.

Team, InstaMorph. InstaMorph, 20 Mar. 2016, www.instamorph.com