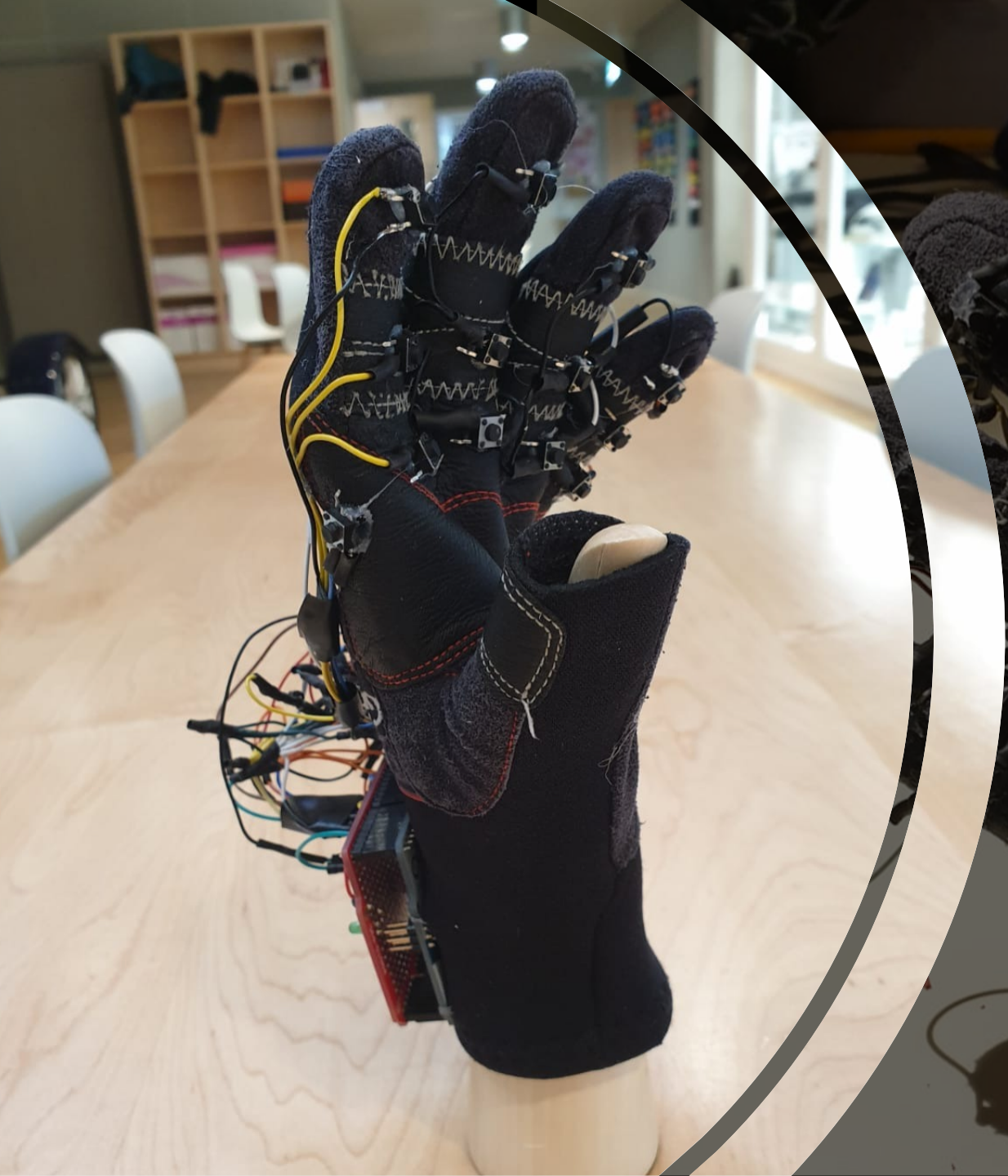


System to Help the Verbally Impaired

Henry Hollingworth

Research supporting the Brief

A mute is a person who does not speak. While mute is often used to describe a person who was deaf from a young age and thus never learnt to speak, it can be extended to encompass all people who cannot speak including those with damaged vocal cords and those with autism spectrum disorders who are non verbal.
Both mute and deaf people often use sign language to communicate and it is these people who have suffered severe hearing loss that make up the majority of the mute community.

Would my product help many people?
The WHO states that 5% of the worlds population have disabling hearing loss which could cause them to also be mute.
It also estimates that unaddressed hearing loss poses an annual global cost of US$ 750 billion. This includes health sector costs (excluding the cost of hearing devices), costs of educational support, loss of productivity, and societal costs.

What would mute people think of this?

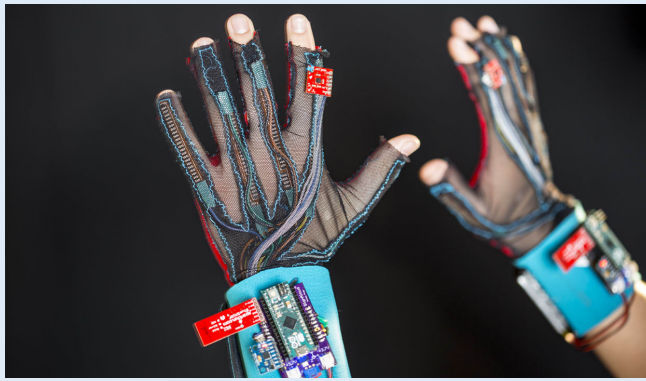
It is essential that mute people do not feel as though they are expected to provide for themselves. There is an extent to which society should provide for people who use sign language, this would include systems which use cameras and which are less invasive. Unfortunately this ideal is not always possible. For convenience devices for personal use such as gloves, phone add-ons and typing translators could be much more useful. Such devices would have the added benefits of being more portable, some of them could even be wearable.

A deaf Reddit user wrote the following concerning the difficulty in directly translating sign language with technology:
“Sign languages are not verbal languages in the proper sense where words are combined in a specific order to make sentences. They're visual languages, more akin to taking meaning from a painting than from a paragraph. The structure of the language itself allows meaning to be expressed in ways that can't be done in spoken languages, and these significant differences would be completely lost in any direct translation device.”
This is important as it may not be possible integrate existing sign languages into my device and I should therefore not focus on this aspect.

What already exists?

Sign language is a primary way of communicating within the deaf community. Sign language is a term which encompasses a plethora of different national and colloquial languages, this means that while guides and translators can be proficient in one language, they may be inept at another.

Gloves which translate sign language are not a novel concept; there have been a number of attempts to produce working prototypes but all of them suffered from problems concerning reliability, being proprietary and inconvenience. If I were to pick that option for development, I would need to consider those factors.



https://www.reddit.com/r/deaf/comments/b3siwt/why_sign_language_gloves_dont_work/
https://www.youtube.com/watch?v=l01sdzJHCCM&feature=emb_err_woyt
https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss
https://www.british-sign.co.uk/what-is-british-sign-language/

Interpreting the Brief

Camera Concept:

This idea encompasses a webcam style camera which a user would sign into and could output text like a normal keyboard.

The design incorporates a sleek, stylish camera design with fast computational power supplied by a Raspberry Pi Zero and outputted with no drivers to a computer via a slightly modified keyboard chip.

The design will contain shiny white plastic and silicone on the curved sides in order to provide an ergonomically satisfactory experience to consumers.

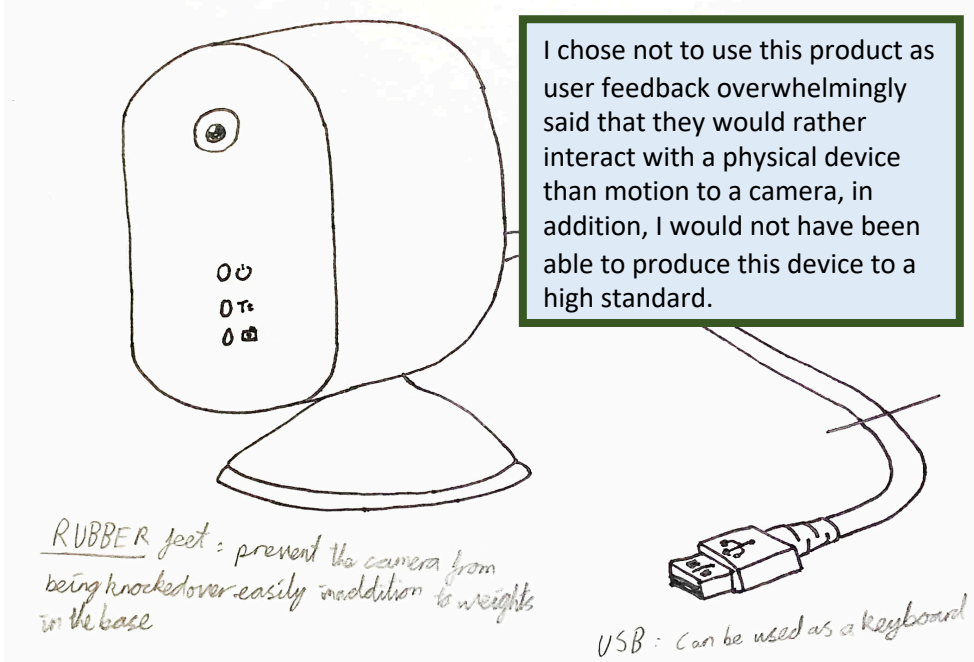
Algorithms will analyse the camera feed and translate sign language into text using machine vision and machine learning. There are many open source frameworks which can facilitate this for example OpenML.

Inspiration:

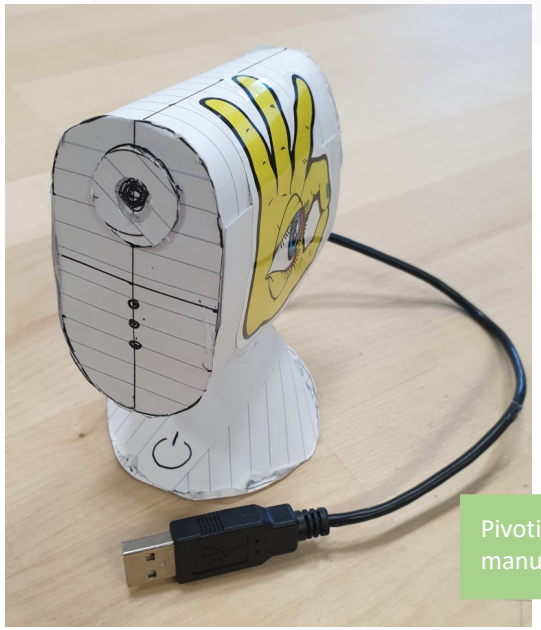
Luxurious camera design has been perfected previously in industries such as security where homeowners sceptical of defacing their homes with traditional boxy cameras were catered for by companies such as Arlo and nest.

This same design will undoubtedly be essential for a camera which is likely to be used on a day to day basis.

Our smart camera builds upon the success of webcam design throughout the ages to create the best product for a potential customer.



I chose not to use this product as user feedback overwhelmingly said that they would rather interact with a physical device than motion to a camera, in addition, I would not have been able to produce this device to a high standard.



Pivoting base allows for better manual positioning

Success Criteria

Quality:

This is ensured by the materials namely textured silicone on the sides of the camera and shiny plastic on the front and stand which will make it comfortable to hold.

Performance:

The built-in algorithms will allow it to function like a normal keyboard

Robust?:

The container will be strong and sealed so there will be no chance of any malfunctions due to damage to circuitry. The materials have been chosen to be wear resistant to mitigate any physical damage.

Usability:

If it is built to specification, this product would essentially be plug-and-play.

Easy to modify?:

Since it will be built on the Raspberry Pi platform any product owner could install their own third-party software based on whichever machine learning I chose to use.

Ease of production:

Programming the Raspberry Pi will be difficult, but once that has been developed it will be easy to mass produce by simple construction.

App concept:

This concept would allow consumers to gesture at their phone and create text. It would incorporate an installable add-on to keyboards on both apple and android phones which would provide a sign language menu.

Since most people have phones this would likely be a very successful and accessible means of solving the brief.

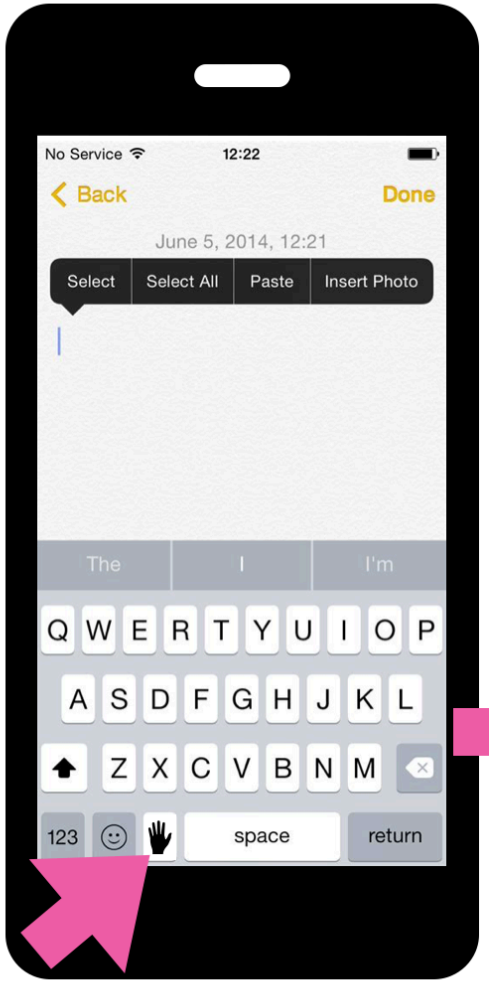
The add on would present the user with a button in place of the microphone button on most keyboards. Once in use, the application will utilise the forwards facing cameras on the phone to translate the sign language.

How it would work:

On a normal apple keyboard and some android keyboards there is a microphone button which provides the facility for many consumers to input text by interpreted speech.

Obviously, this facility is not useful to those who suffer from muteness. This product would provide a useable alternative for them and as such increase accessibility to the internet and other services which require text input. This will help to break down the language, or lack thereof, barrier which exists online and give those suffering from muteness another means of communication.

This revolution in technology would utilise the phones array of cameras and sensors to detect hand movements in much the same way as the google pixel phone can skip songs with gestures



Compared to other solutions, I thought this idea came off as a novelty. Additionally, research revealed that this would not be practical to use as it would require someone or something else to hold the phone and it would probably be quicker to type.

Success Criteria

Quality:
The application would have to meet similar design standards to existing phone interfaces such that it would not take away from the user experience.

Performance:
Modern phones are easily capable of the necessary computation.

Robust?:
As there is no physical form of the application, there is no chance that it could be broken, and continuous updates will serve to future proof it.

Usability:
The simple idea of gesturing at a camera can be explained and demonstrated in a short tutorial commonplace in many phone applications.

Easy to modify?:
Due to the necessary integrity of a keyboard application, and in order to prevent hackers from taking advantage, the application would not be possible to modify, however the code could be made available for perusal online.

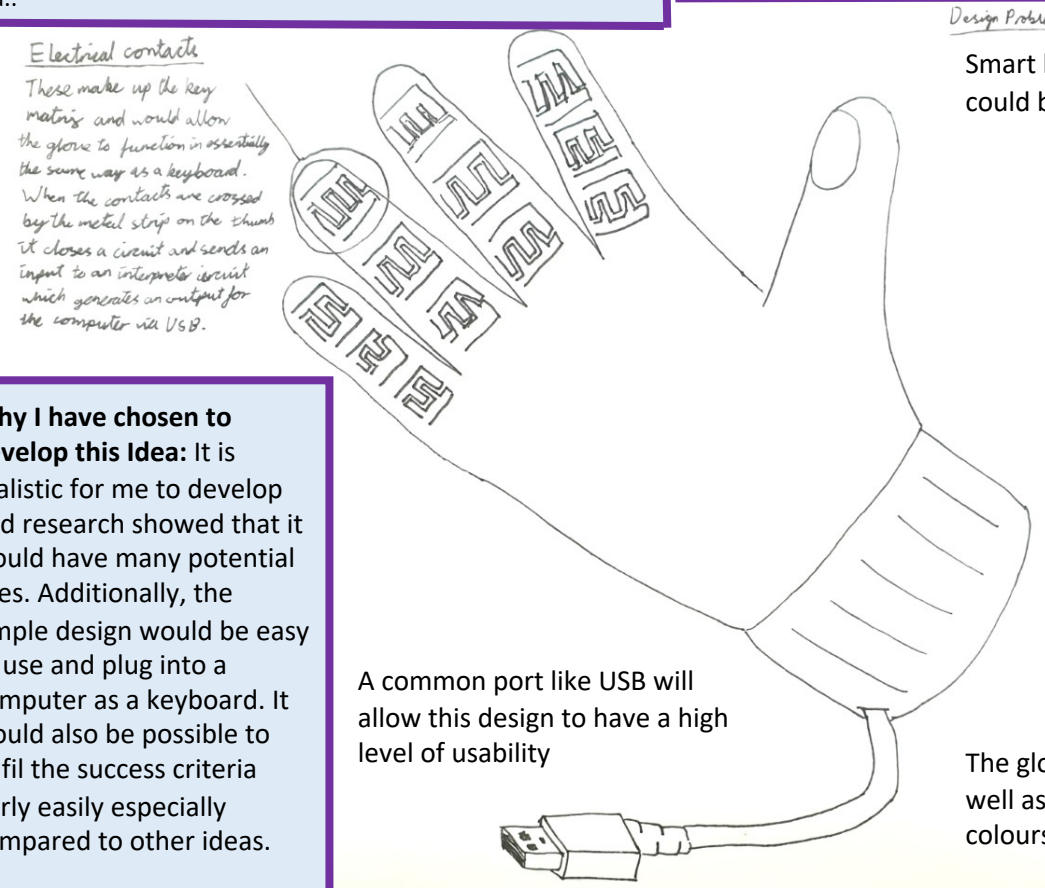
Ease of production:
Once the code is written, it can be replicated indefinitely

Glove concept:

A wearable device would interpret hand gestures and output in the same manner as a keyboard. It would be easy to use out of the box, would be comfortable and also stylish. Not only could gloves like this be used for muteness but they may also have applications in the world of smart glasses and other similar keyboard-less smart devices. Manual text input has always been an important part of how we interact with computers and this device would allow it to continue as such in a new era..

How it could be used:
In order to interact with a smart device, many tech companies are turning to voice control and as such deaf/mute people are struggling in the modern technological revolution. These gloves could serve as an equivalent for microphones and allow easier communication.

A valuable feature:
Due to the nature of sign languages as poorly understood and private, it is true that many mutes don't like everyone being to know what they are talking about. This design would allow them to not only use a different typed language to spoken language but it would also be harder to translate without permission than other designs.



A common port like USB will allow this design to have a high level of usability

Design Problem

Smart keyboard-less glasses could be controlled by these.

The gloves could be stylish as well as practical in a number of colours and cuts.

Why I have chosen to develop this Idea: It is realistic for me to develop and research showed that it would have many potential uses. Additionally, the simple design would be easy to use and plug into a computer as a keyboard. It would also be possible to fulfil the success criteria fairly easily especially compared to other ideas.

Success Criteria

Quality:
The device will be made from quality materials similar to the fashion industry and the electrodes will be made of a corrosion resistant metal.

Performance:
The glove will be almost exclusively hardware and will run as quickly as any normal keyboard.

Robust?:
The glove will be made of high-quality materials and although it will not be able to be used in the garden, most situations will be suitable.

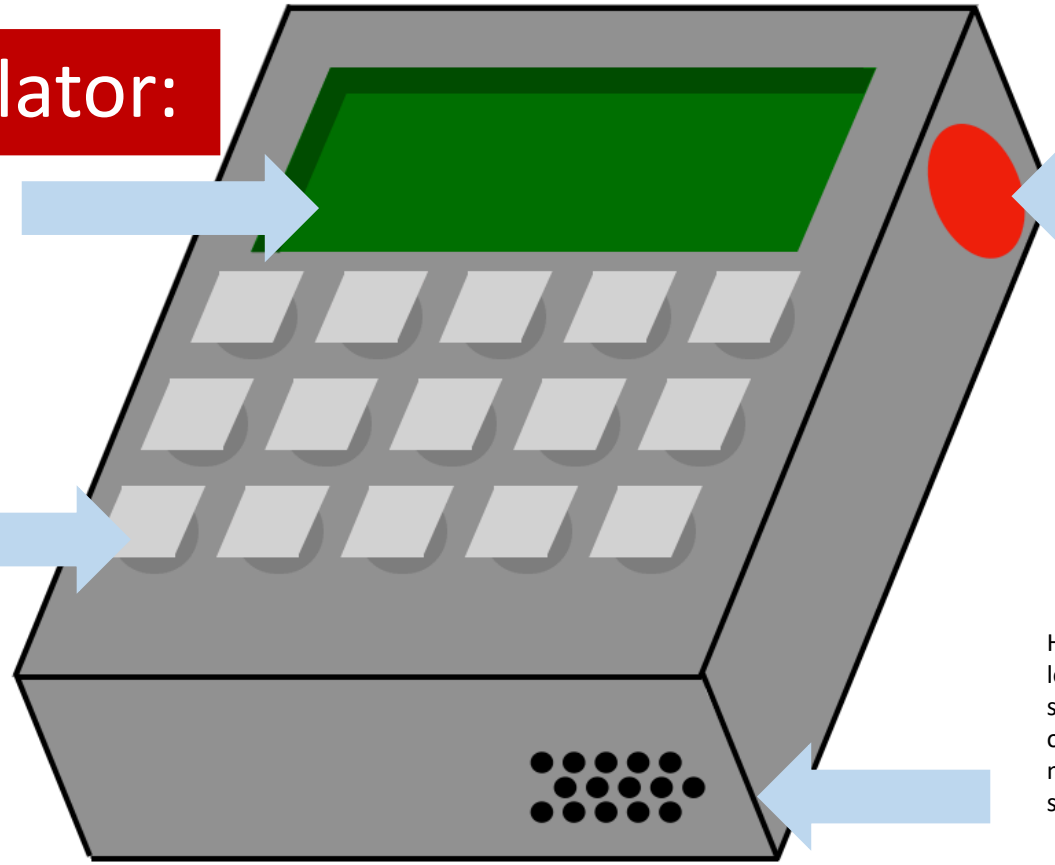
Usability:
After learning to touchtype with the device, it should be no more difficut than a standard keyboard to use and operate.

Easy to modify?:
This device has many possibilities for uses outside of simply helping the mutes. For example, the problem of smart glasses without keyboards is significant and it is key to the development of future technology that a solution is found.

Ease of production:
The device will be fairly difficult to produce and if the glove must be built from scratch that will also decrease the ease of production.

Keyboard translator:

The LDC screen would show the desired sentence, phrase or exclamation to give the user an opportunity to proofread it before adding it to their conversation.



Button to press when you want the displayed phrase to be spoken by the speech synthesiser.

I decided not to use this idea as it presented no benefits over using text to speech apps on a phone and would be unlikely to fulfil the success criteria.

The keyboard allows users to enter text into the device which then appears on screen.

High quality speakers which loudly project the synthesised speech at up to 65 decibles in order to be heard by the other members of a conversation the same as if the mute were talking.

This concept would be used by typing into the keyboard and pressing the button on the side to turn the text into speech
A portable computer which can be carried and used to communicate with those who do not understand sign language. This will also have the ability to store a number of commonly used phrases which will mean it is quicker and easier to use in every day life for example when buying groceries or asking for directions. All in all, this sort of minimalistic device will vastly improve the standard of living for mutes.

Important aspects of design:
While designing an object like this which will be used frequently it Is important to consider the conditions it will be used in. This is the type of item which will be used every day and as such will have to withstand the wear and tear of handbag or pocket life.
To address this, the main case will be made out of metal with hard–wearing plastic buttons which can be used to type the desired message. The display will be an LCD to minimise the cost of production and to also be durable and easily replaceable. The speaker will be enclosed behind holes pictured in the diagram, a mesh should be placed behind the holes to prevent dust and grit from entering and eroding the cone of the speaker.

Success Criteria

Quality:
The device will be made from quality hard wearing materials to survive everyday life and the outer case will be made of a corrosion resistant metal.
Performance:
The device requires minimal processing power and the computer powering it will be more than powerful enough to surfice.
Robust?:
The device will be made of high-quality materials and will be hard wearing and useable everywhere.

Usability:
For seasoned phone users, this minimalistic design will be trivial to understand and master
Easy to modify?:
Due to this being a specialised device for mutes, I can't see many reasons to modify it. In theory, it would be possible to add third-party software to the computer running the device.
Ease of production:
The device could easily be mass produced but building just one prototype would be difficult due to the complexity of the design

Hacking a Keyboard

As I am basically creating a keyboard, I decided to see what was inside one. A keyboard is made up of a keyboard matrix in which the keys are placed. This matrix is then fed into a microcontroller which interprets the keystrokes through pins and converts it into a USB output.

In order to create my own glove-based keyboard, it is essential to mimic the microcontroller's USB hardware output. This presents two options and their associated processes:

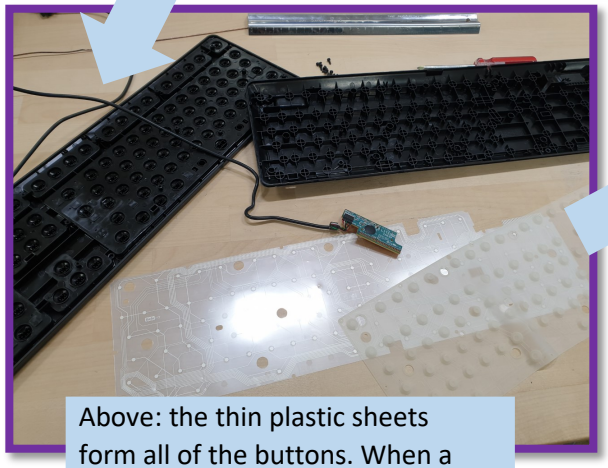
Using the existing microcontroller:

- Find a way to attach my own wires to the microcontroller as it only has connection pads.
- Find a way to map out the key configuration of the microcontroller.
- Find a way to attach the controller to the glove.

Using my own microcontroller:

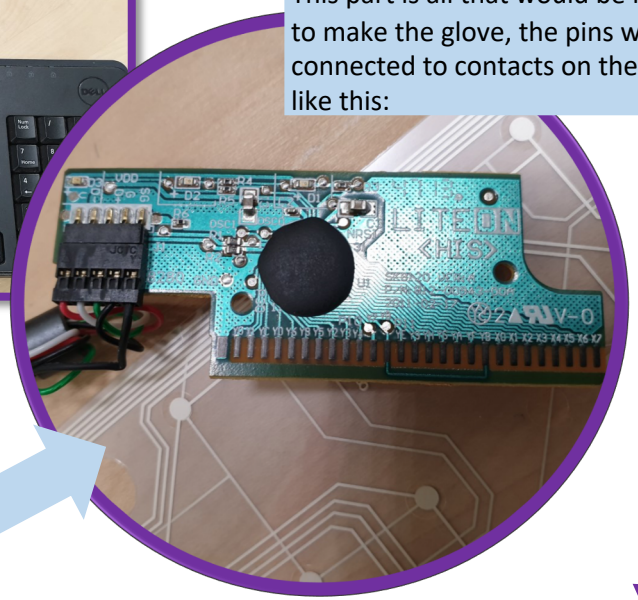
- Find a controller with USB hardware mimicking capability
- Programming the microcontroller
- Find a way to connect enough buttons (26+) to the controller.

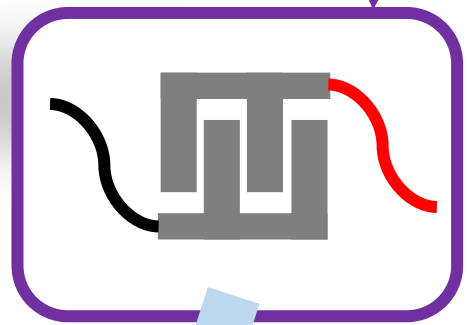
As I have now got a microcontroller from the keyboard I took apart, thus I will pursue the first . The main practical challenge with this approach is to attach the new buttons to the microcontroller, If this problem cannot be overcome, all other development on this approach will be futile.



Above: the thin plastic sheets form all of the buttons. When a key is pressed, the two sheets touch together at a point which outputs a key press.

This part is all that would be needed to make the glove, the pins will be connected to contacts on the glove like this:





The buttons (or contacts) will be attached with one wire on the x pin of the controller, and one on the y pin. This creates an 8*19 matrix or 152 different key combinations all of which I will need to try to get a complete idea of which keys go where.

Building a breakout board

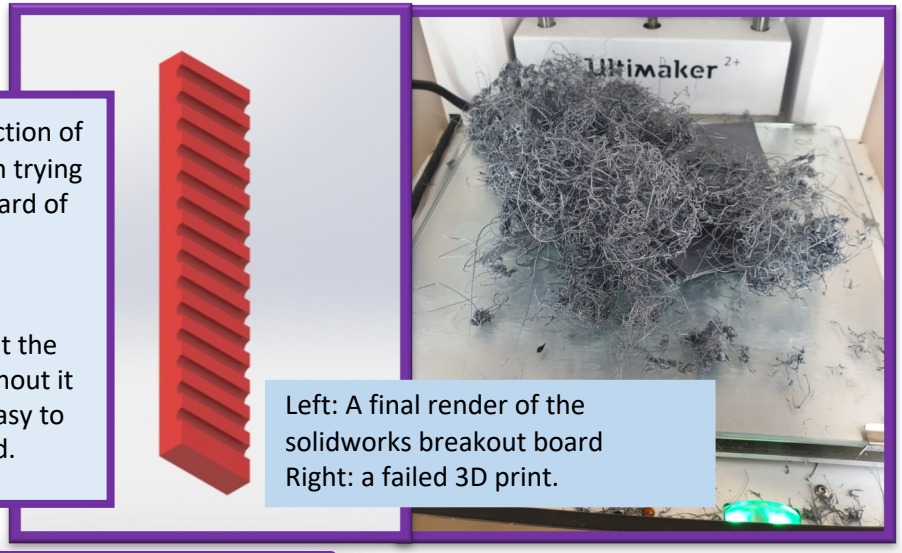
In order to attach the circuit board to the glove, I needed a way of attaching cables. Since the keyboard makers didn't want people to do this, they made the electrodes out of graphite so that they would be very hard to solder to. I tried and failed to do this. Modifying the interpreter circuit would also prove difficult as it was very small and complicated. As such a breakout board of two types occurred to me.

The first type would be plastic and 3d printed. I designed such a device in Solidworks (a 3D modelling software) and after a failed print, I managed to get a useable model. Unfortunately, it was almost impossible to get the fiddly wires to fit the board and I had to conclude that this means of attachment was not feasible.

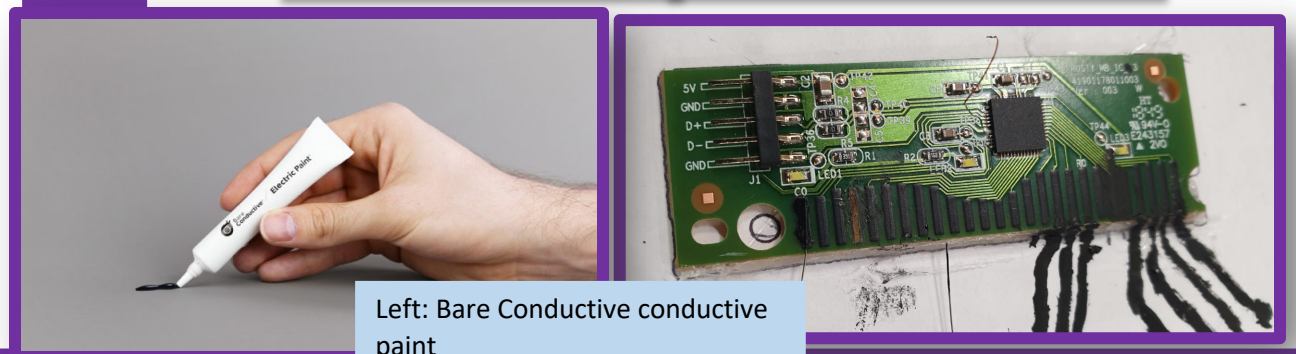
The second approach was using electrical paint to create a printed breakout board. I discovered a company called 'Bare Conductive' which produce conductive paint and so I tried to use some of that to make connectors which could simply be clamped down on the microcontroller with the existing metal bar. Unfortunately, the electrical paint proved to be far too flaky and would come off during the unmasking of the painted connectors and also when exposed to any bending.

Throughout the construction of the breakout board, I am trying to maintain a high standard of quality and robustness.

I made the circuit board housing in order to meat the robustness target as without it would have been very easy to damage the circuit board.

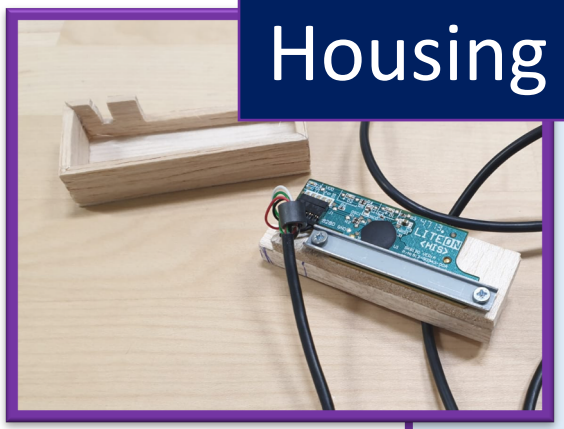


Left: A final render of the solidworks breakout board
Right: a failed 3D print.

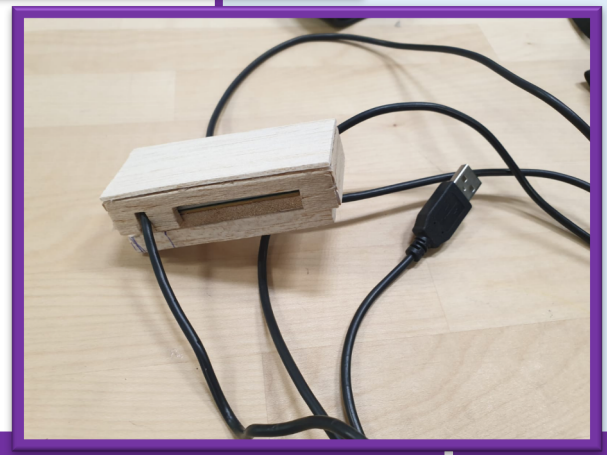


Left: Bare Conductive conductive paint
Right: An attempt to create a painted breakout board

Housing the circuit board



To house the board and help attach it to the glove, I created a prototype balsa wood container for it. This incorporated a hole through which a USB cable could pass, a hole for wires or breakout boards to be able to access the pins and was designed to be a friction fit with the bottom half of the container.



Had I continued the development of the Keyboard microcontroller, I would have taken measurements from this design and 3D printed it.

A potential user said, "It looks fine and I'm sure it'll be even better when 3D printed. It would be better if you could make it smaller"

Development

Using an Arduino

Due to the difficulty which I experienced in hacking the keyboard controller, I decided that the only way I could progress was by completely scrapping the previous iterations and development using a keyboard controller and instead using an Arduino Micro controller instead.

Below is a potential evolution of which microcontrollers I would use in the glove. The Arduino Uno is ideal for initial prototyping as it is widely available and the serial communication is far easier to debug. The similarity of the Serial and Keyboard functions will make it easy to move the prototype code to the final design.

I found a website for DIY keyboard creators which was selling the 'Pro Micro'; an Arduino compatible product with a built in 32u4 chip. This can be used with the 'Keyboard' library to emulate the action of a normal keyboard. The pro micro is based on the Arduino Leonardo which is principally used for this purpose.

Arduino Leonardo

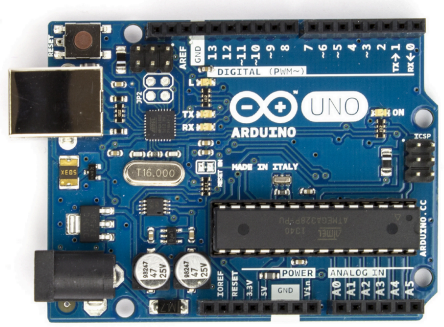
The Arduino LEONARDO is an integrated USB HID Arduino board. Ideal for projects requiring the board to behave (act) as a USB human interface devices.

A description of the Leonardo on the Arduino website.

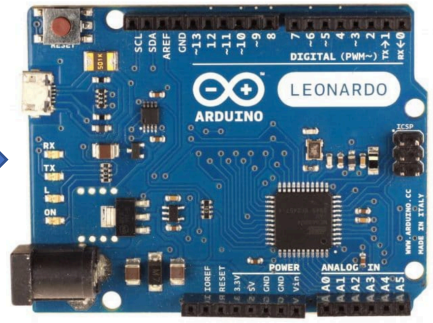
https://www.diykeyboards.com/parts

$7.99

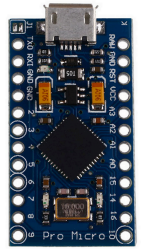
Pro Micro



The Arduino Uno that I used for initial prototyping could only output to the Arduino COM port.



The Arduino Leonardo can be used as a USB hardware device and can therefore act as a normal keyboard albeit in a large form factor.



The Pro Micro retains the utility of the Arduino Leonardo but utilises a smaller form factor.

Using an Arduino

The keyboard functions in fig. 1 are available on all Arduinos using an ATmega32u4 chip which has built in USB interfacing (such as the Arduino Leonardo in fig. 2). As the aim of the project is to build a keyboard, I will use them extensively to mimic key presses. The important functions are the begin, print and write functions. In order to fulfil the "Performance" section of the success criteria, it is essential that my code is efficient.

Functions

Keyboard.begin()
Keyboard.end()
Keyboard.press()
Keyboard.print()
Keyboard.println()
Keyboard.release()
Keyboard.releaseAll()
Keyboard.write()

Fig. 1

Another feature I developed was the ability to type entire words with a single button click, this allowed for faster typing as some common words make up a large proportion of our language

```
else if((digitalRead(12)==LOW) && (digitalRead(8)==LOW)){
  Keyboard.print("the");
digitalWrite(2, HIGH);
delay(200);
digitalWrite(2, LOW);
}
```

Keyboard.begin()

This function initiates the Arduino's ability to mimic a keyboard. After this function has been run, the host computer will no longer recognise the Arduino and will assume that it is a keyboard. While his is a good thing for the project, it becomes very difficult to program the Arduino once it this function has been initiated. As such it is necessary to build in a failsafe.

Keyboard.end()

The end function is imperative as it tells the Arduino to stop mimicking the keyboard. This allows the Arduino to receive new code (as it cannot be communicated with while in keyboard mode). It is this function that the failsafe will be built around.

The Failsafe

Using the INPUT_PULLUP class for pin 2 of the Arduino, I can create a functionality where the keyboard code will only run if pin 2 is grounded (fig. 2). Having done this, when I want to upload new code to the Arduino Leonardo, I must simply pull the pin out of the header which will then trigger a function in the code to end the keyboard functionality.

Keyboard.print()

This function prints characters by simulating key presses. The great thing about the print function over the press function is that it is not limited to pressing one key at a time and could allow future development such as being able to use the glove to type entire sentences with one movement. Ultimately, the open source nature of Arduino allows for third parties to write their own code and modify the glove.

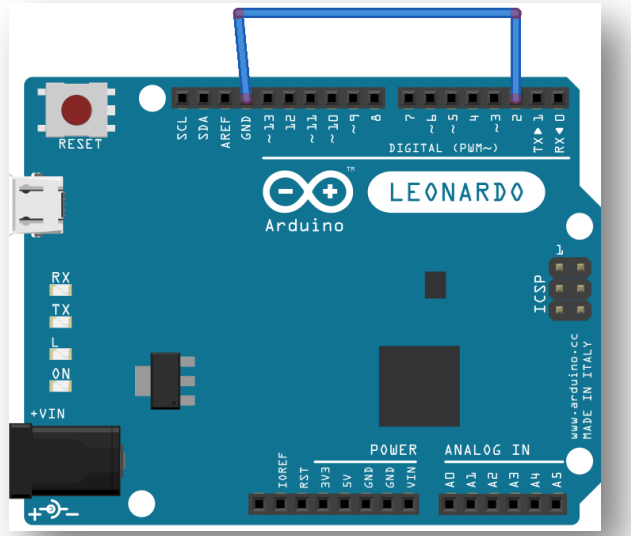


Fig. 2

Development between first and final prototype

To start with my code used the “Serial.print()”function which allowed limited communication with a tethered computer, As I was using an Arduino UNO at this stage, it could not emulate a computer

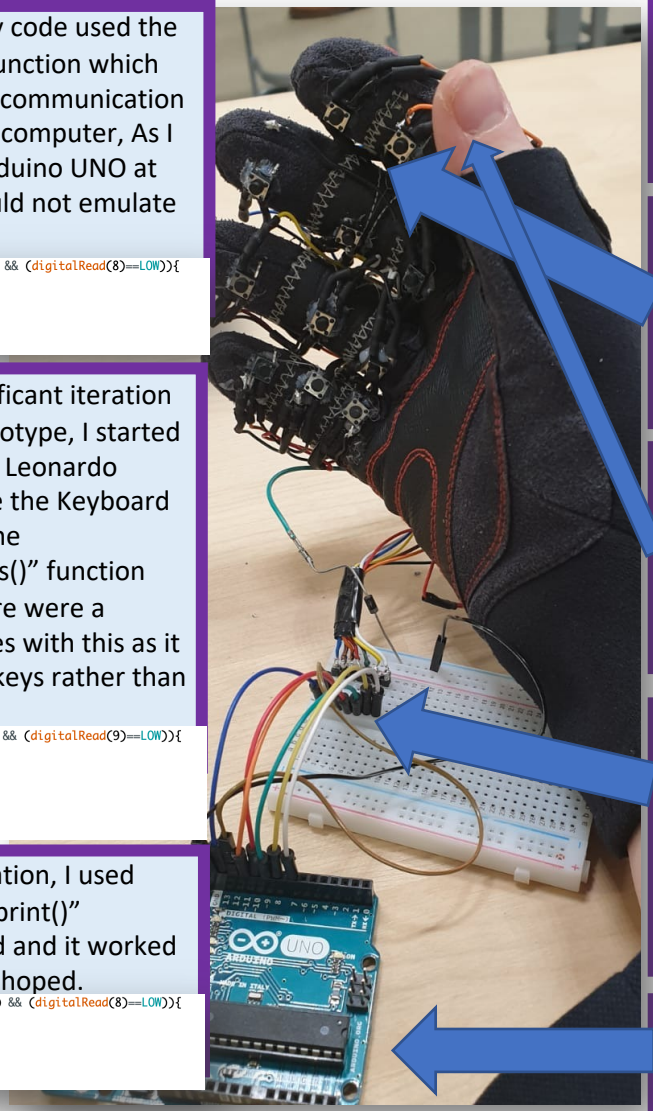
```
else if((digitalRead(11)==LOW) && (digitalRead(8)==LOW)){
  Serial.println("h");
  digitalWrite(2, HIGH);
  delay(200);
  digitalWrite(2, LOW);
  }
```

In the first significant iteration of the final prototype, I started using a Arduino Leonardo which could use the Keyboard library. I used the “Keyboard.press()” function initially but there were a number of issues with this as it held down the keys rather than typing them

```
else if((digitalRead(12)==LOW) && (digitalRead(9)==LOW)){
  Keyboard.press("k");
  digitalWrite(2, HIGH);
  delay(200);
  Keyboard.release("k");
  digitalWrite(2, LOW);
}
```

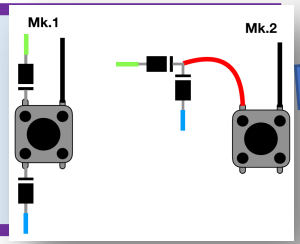
In the final iteration, I used the “Keyboard.print()” function instead and it worked exactly as I had hoped.

```
else if((digitalRead(11)==LOW) && (digitalRead(8)==LOW)){
  Keyboard.print("h");
  digitalWrite(2, HIGH);
  delay(200);
  digitalWrite(2, LOW);
  }
```



Once I had a fairly firm idea of what I needed to do to build my product, I decided to build a first prototype to inform my final realisation. This would allow me to see what worked in practice and what did not. This page will serve to document the changes that I made between the first and final prototype

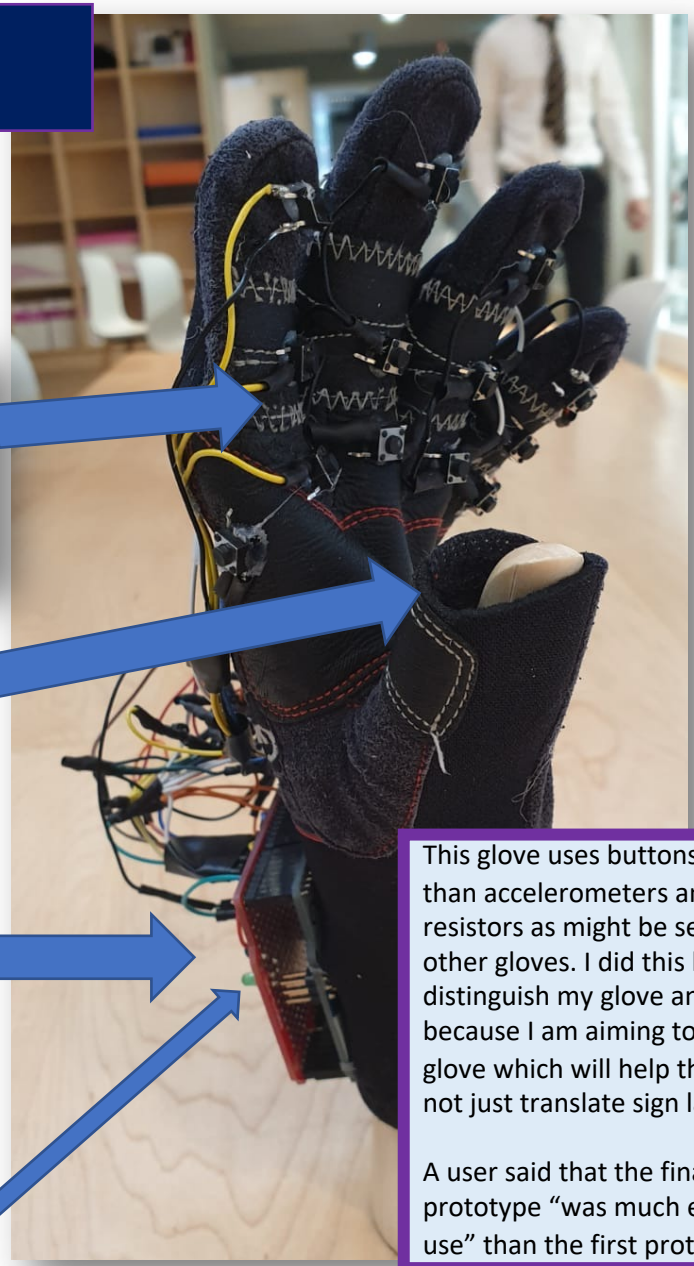
In the first prototype many of the wires connected to the buttons broke off. In the second prototype I decided to move the location of the diodes further down to the back of the glove to reduce stress on the connections which worked well.



In the first prototype, the glove had add a top on the thumb. Unfortunately this made it very hard to press some of the buttons. Having removed this, it became much easier to type and I incorporated this feature in the realisation iteration.

I connected my initial prototype to the Arduino using a solderless breadboard. However, when I wanted to to attach it permanently to an Arduino for my realisation, it was necessary to create a soldered Arduino shield. Additionally, in order to register a button press, I added a vibration motor to provide haptic feedback when a key was registered.

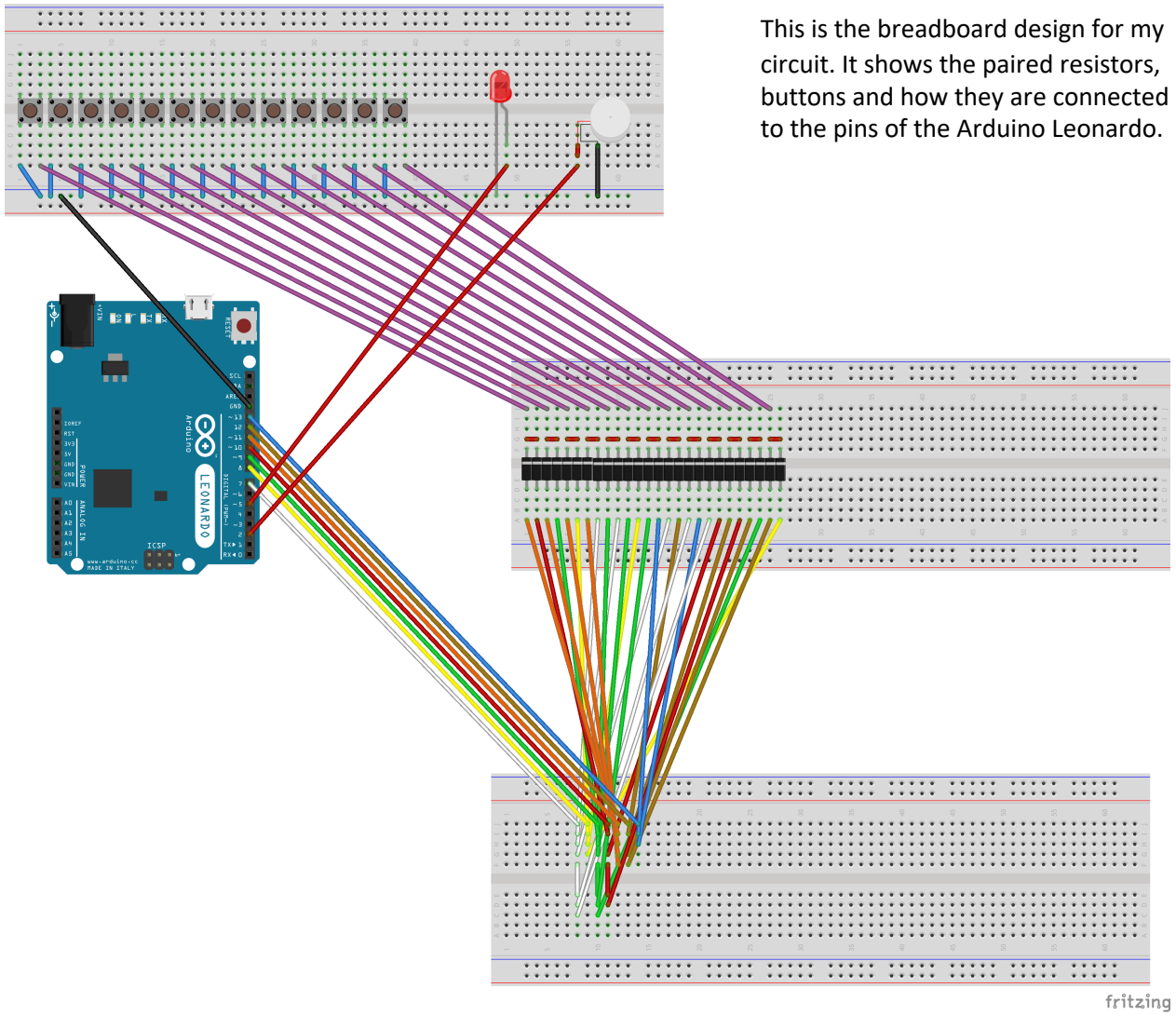
In order to make it more obvious whether my code was running or not, I added an LED status indicator to the glove.



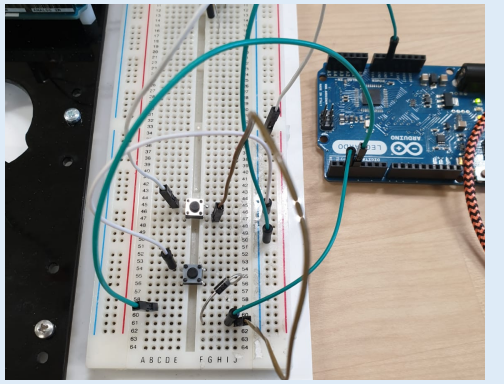
This glove uses buttons rather than accelerometers and flex resistors as might be seen in other gloves. I did this both to distinguish my glove and because I am aiming to create a glove which will help the mute not just translate sign language.

A user said that the final prototype “was much easier to use” than the first prototype.

Realisation skill 1: Designing a circuit

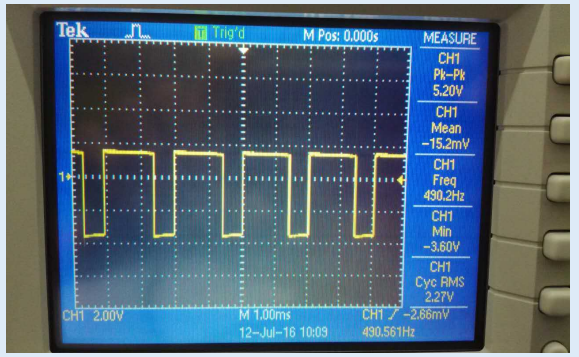


This is the breadboard design for my circuit. It shows the paired resistors, buttons and how they are connected to the pins of the Arduino Leonardo.



The majority of the circuit will be comprised of buttons thus I prototyped it on a plug in breadboards. I worked out the correct polarities of the diodes in this manner and realised that by using diodes to control the flow of current, I would be able to minimise the number of pins used in the construction.

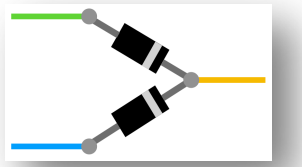
Two additions were made at the last minute. One of which was a DC vibration motor, the other of which was an LED in order to show the status of the project (ie : on if the project is working). The LED was controlled using pulse width modulation on pin 5 in order to prevent the use of a resistor.



Pulse width modulation produces a digital wave part of which is that 0 volts and part of which is at 3.30V direct current. The amount that is on and the amount that is off determines the effective voltage of the PWM pin. The motor works perfectly well at 3.30V direct current and therefore does not require any further complexity to function

Realisation skill 1: Designing a circuit

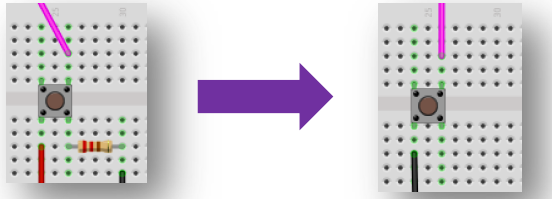
Shown on the right is the schematic of the glove that I built. As you can see, there are 13 buttons and 26 diodes. Each diode is paired with another and they then connect to the button by one pin, but are connected to two different Arduino digital pins.

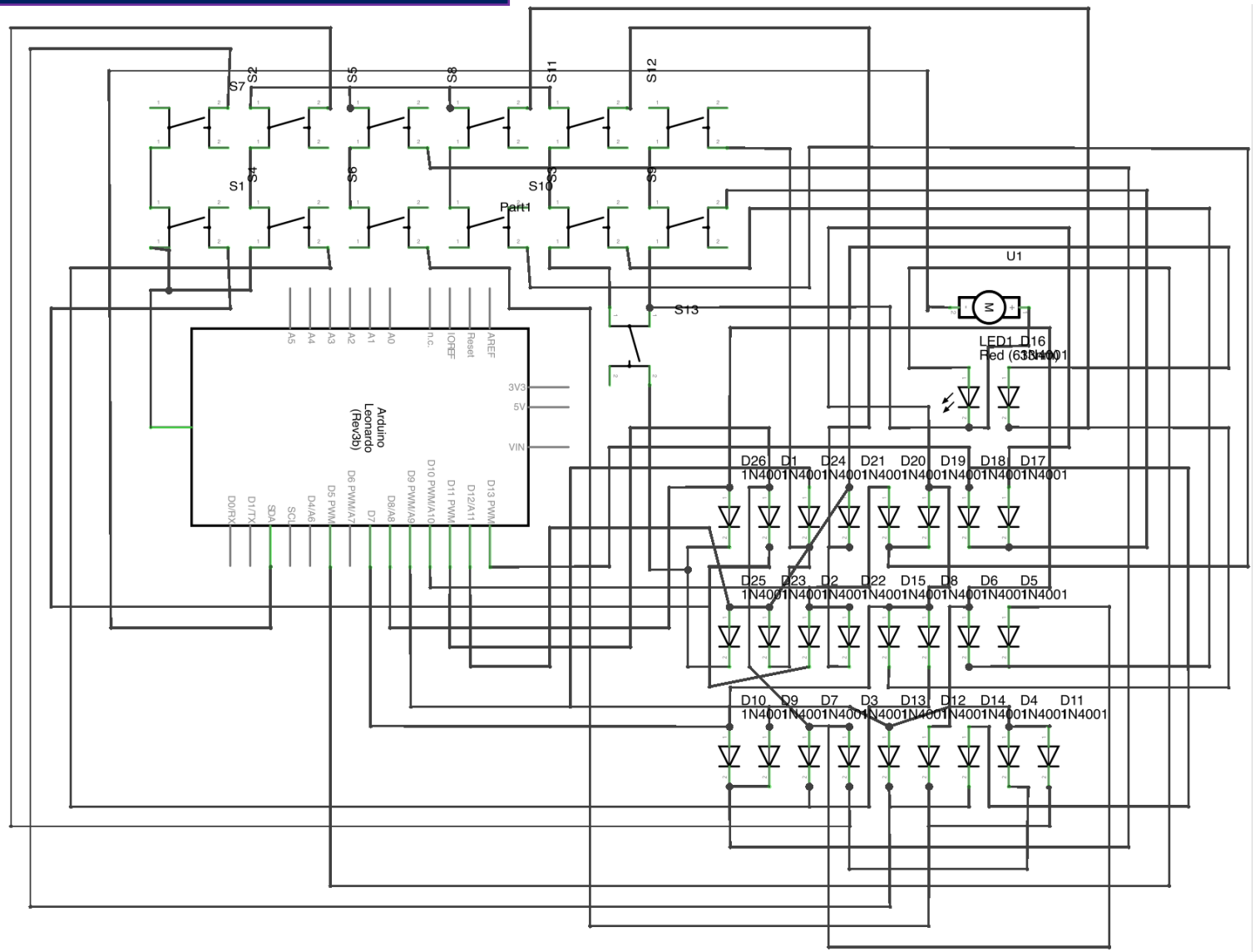


I decided when designing my circuit that it would need to use soldered connections since using a solderless prototyping board would have resulted in the large number of wires being pulled out by mistake. Some soldered connections were made on the Arduino HAT (Hardware Attached on Top) but most were made between components and then covered with shrink wrap.

```
pinMode(10, INPUT_PULLUP);
```

Here is the reduction which can be made in the circuit due to the INPUT_PULLUP class. This is possible as the PULLUP class simply checks to see if the pin is grounded or not.



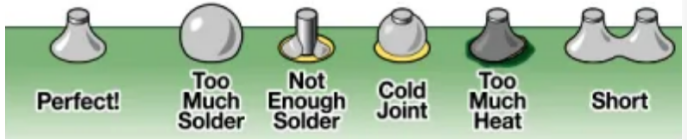


Realisation skill 2: Assembling the product

Soldering

The key skill that I used in the construction of the gloves was soldering. I constructed a soldered Arduino shield (fig. 1) and then soldered all 26 diodes to the wires and covered in shrink-wrap. Additionally, all of the buttons were soldered to ensure a robust connection which would not break under the strain of being bent when the glove is closed into a fist.

Fig. 2



Overall my soldered connections were fairly good. Using the infographic (fig. 2), I determined that most of my joints were perfect. While I was still in the process of soldering the shield, I shorted a few connections but with the careful use of solder wick (fig 3), I managed to fix them all and make them perfect.

Fig. 3

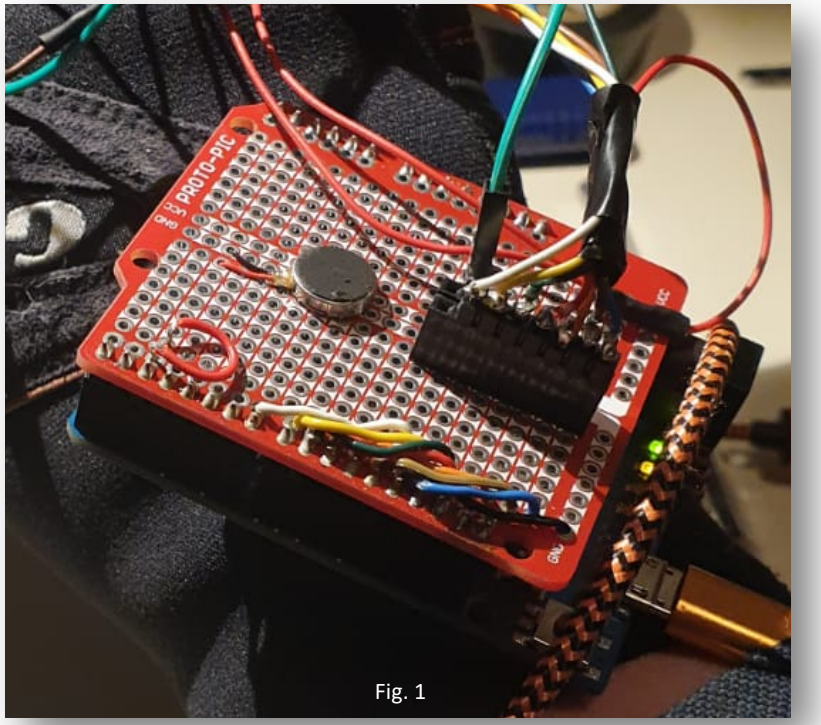


Fig. 1

The biggest setback which I experienced in the realisation process was when some of the wire connectors from the buttons on the first prototype began to fall off (fig. 4). I solved this by removing some of the heat-shrink to the buttons (as this was causing unnecessary stress to them) and in the second prototype, moved the location of the diodes to the back of the hand so that the button connectors had more redundancy (fig. 5).

In order to attach the electronics to the glove , I used hot glue shrink wrap and cotton thread. These were easy to use, quick and reversible ways to attach the electronics and as such, the correct methods to use, especially when working with a textile.

Fig. 4

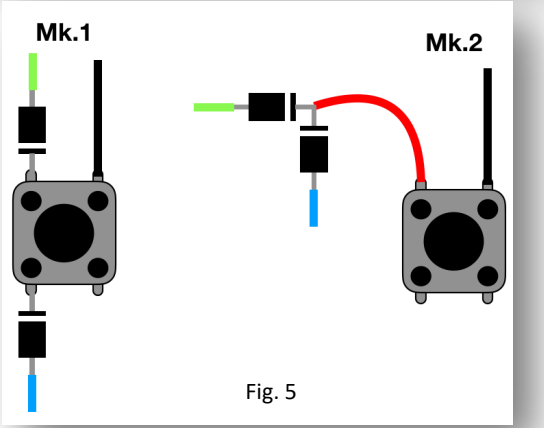


Fig. 5

Realisation skill 3: Programming an Arduino

To receive all the button inputs, I am using pullup pins. These mean that I don't need to use the complicated button circuit and drastically reduces the amount of soldering required.

```
pinMode(10, INPUT_PULLUP);
```

When the pin is grounded the readout is LOW otherwise the Arduino reads it as HIGH. I used these values in my code so that when any two pins were grounded by one button press, a key will be pressed, the vibration motor will be turned on for 200 milliseconds and then turned off.

	7	8	9	10	11	12	13
7			o	a	e	y	i
8			n		h	u	
9					l	k	r
10						m	
11			N/A			" "	
12							
13							

```
else if((digitalRead(11)==LOW) && (digitalRead(8)==LOW)){
  Serial.println("h");
  digitalWrite(2, HIGH);
  delay(200);
  digitalWrite(2, LOW);
  }
```

The difference between the prototype and final code. Note the addition of the keyboard functions.

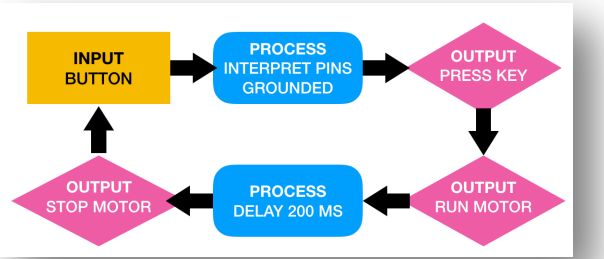
```
else if((digitalRead(11)==LOW) && (digitalRead(8)==LOW)){
  Keyboard.print("h");
  digitalWrite(2, HIGH);
  delay(200);
  digitalWrite(2, LOW);
  }
```

Key matrix

In order to program which keys need to go where, it was first necessary to decide what I wanted to be able to type. The full product would have two gloves and all available keys, but as this is a prototype, I decided I wanted to be able to write "I am Henry". As such, here is the key matrix with pin numbers on the side.

Pseudocode

The following shows the Inputs, Processes and Outputs of the looping code in a visual form.



Evaluation: Success Criteria

Quality:
The glove is made from high quality, hard-wearing materials such as neoprene and leather. The circuit is soldered rather than using weaker solderless connections. All joints are shrink-wrapped so as as to prevent unwanted contact and short circuits. The Arduino is interfaced via a soldered shield which makes the connections to the Arduino secure. This criteria is fulfilled to an extent as the majority of the glove however due to a lack of cable management, it leaves something to be desired.

Performance:
Due to the low latency of the Arduino Leonardo and the efficiency of my code, the glove can act almost as quickly as a normal hardware keyboard. This was shown in the testing section of the evaluation. In reality, the limiting factor of this glove will never be the device but rather the users reaction time and ability to hit all of the buttons. Especially on the pinkie finger, the buttons can be slow to use.

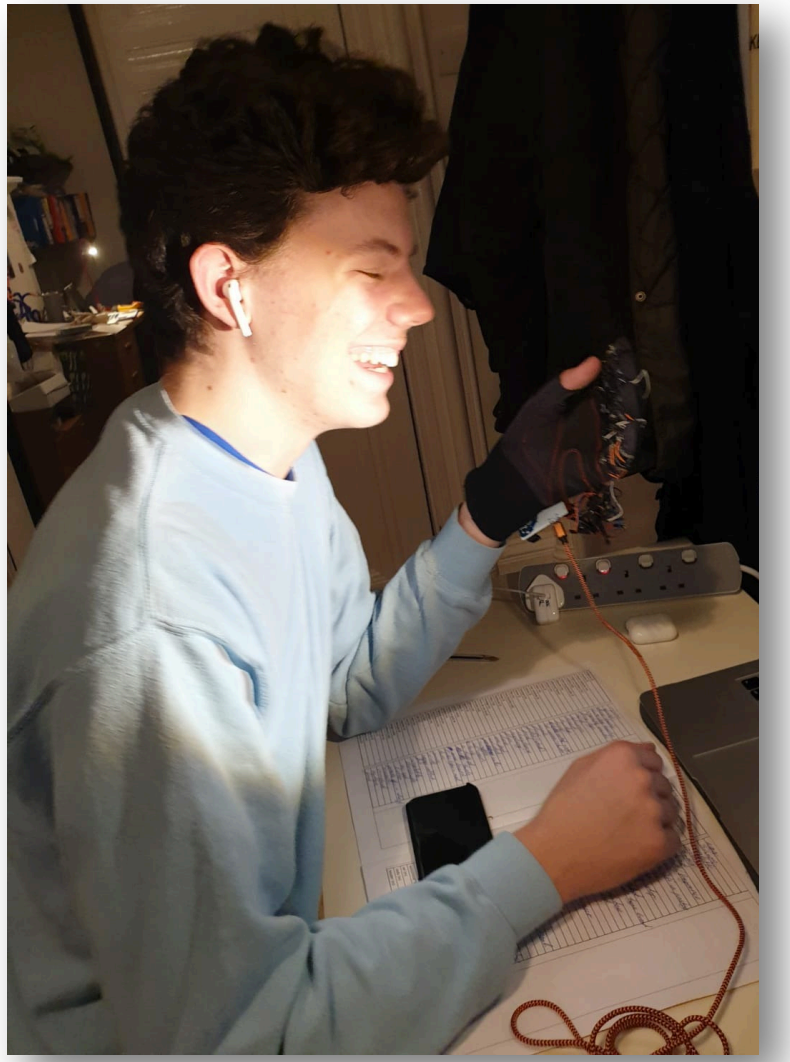
Robust?:
While the glove itself is very hard-wearing and fits this success criteria, the buttons I used in the glove had the proficiency to break. More specifically, the tiny metal legs on the buttons kept breaking off.
I believe that part of the reason for this was the use of shrink wrap to insulate the connections and prevent short circuiting. This resulted in more rigid connections which put more stress on the legs. As such the product does not meet this criteria as it would not withstand regular repeated use.

Usability:
Having learnt the key layout , it becomes very easy to type using the glove keyboard. The inclusion of a status LED as well as haptic feedback both in the form of clickable buttons and a vibrating motor give the user a very satisfying button click experience. It is true that some users have difficulty reaching the buttons on the pinkie finger of the glove.
The glove is actually surprisingly quick to type with and this was said several times by users in the testing and feedback processes

Easy to modify?:
The Arduino used to control the keyboard is an open source hardware and software solution. What this means is is there anyone in the world can you use the internet to find documentation concerning the development on the Leonardo board. This would allow for a variety of programs remapping or increasing and functionality of the glove.
This absolutely fulfils the criteria as it would allow for future developers to incorporate the glove into their own interface systems such as but not limited to smart glasses, watches, and keyboard less computer concepts.

Ease of production:
My realisation prototype could hypothetically be miniaturized and incorporated into a printed circuit board. This would allow for ease of production as most of the assembly could be mechanized and then the only manual assembly process would be attaching the buttons to the glove. The prototype was quite labour intensive as it was necessary to solder every single joint at an odd angle free hand.
There are several companies which offer easy to use solutions for hobbyist's PCBs and as such it would be possible to mass produce with relative ease.

Evaluation: User Feedback



Throughout my user feedback, the main positives which I heard were:

- It's easy to use
- It's very useful
- It allows greater accessibility

The main criticisms and improvements which were suggested were:

- It isn't very comfortable
- It shouldn't fall upon the desired user to provide translation services.

"My only concern is that the deaf/mute people would feel put out by this, surely it shouldn't fall on them them to provide their own accessibility, it should be provided by society."

"Its actually really easy to use! I was surprised as it seems like a complex device. The glove needs to have more keys so that you can type full sentences but the concept is sound."

"I think this is a very useful product and be very helpful to those who need it."

"It's a great idea and is really easy to use but is somewhat uncomfortable especially when worn for a long period of time."

Having used the glove, another participant said, concerning the potential uses of the product:

"I think this glove is really great. It allows people to fully integrate into society, a lot of the problems with sign language and other speech toolkits is that the responsibility of accessibility falls on the host. With this glove, it's a lot easier for mute people to function as normal in society; they don't have to ask anyone to help, they can be self sufficient"

Evaluation: Testing and Trialing

Fig. 1

hello my name is henry

Fig. 3

During testing, the glove was successfully used on a variety of computers, all normal operating systems were trialled (MacOS, Windows, Linux (Ubuntu)). All operating systems were equally easy to use the glove on and could be operated simply as shown on the flow chart in fig. 2. The glove could then be used as shown in fig. 1 to type into any word processing application on the computer used. The results of this are shown in fig. 3. As you can see, the glove works in exactly the same manner as a traditional keyboard would.

There were a few drawbacks and discrepancies to the design, firstly, there is a limit to what can be typed using this prototype as it only has 13 keys, this could be improved by either building a second glove to increase the design to 26 keys + extra keys which could be added to the glove or by modifying the code so that less common characters could be accessed using a shift key of sorts.

Fig. 2

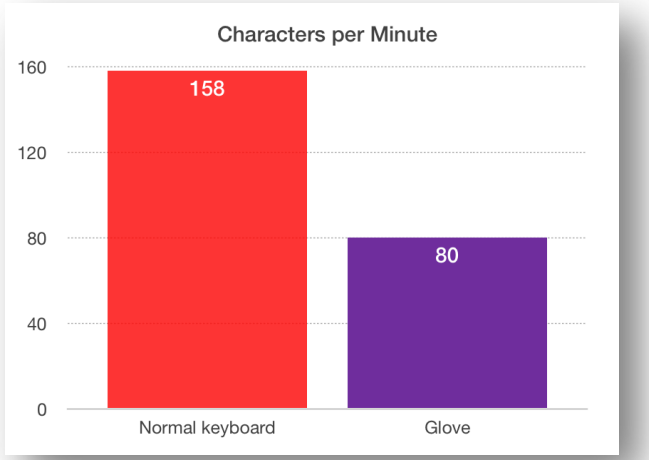
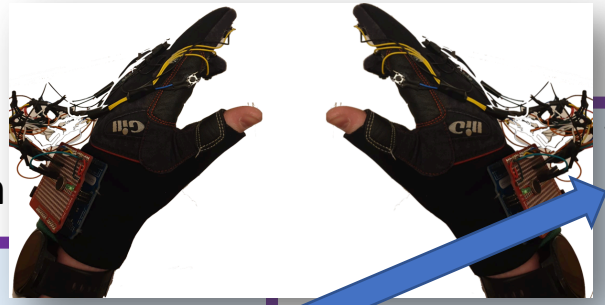


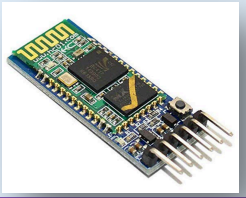
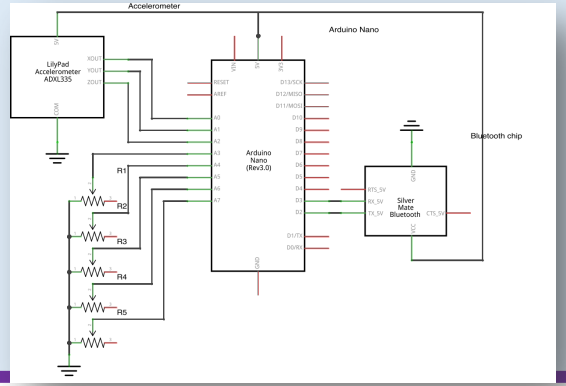
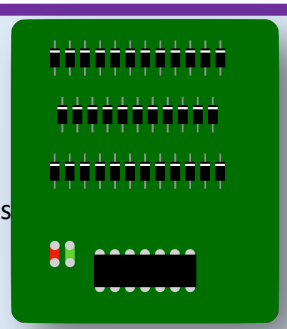
Fig. 4

Speed testing:
The glove was never going to be as fast to type with as a normal keyboard but during a speed test compared to one, it proved to be just over half as fast (fig.4) which was surprising as I had initially thought it would be considerably slower. If, in future a predictive text technology was made available to speed up the process of typing, much like those available on phone keyboards, I think the glove could rival that of a normal desktop mechanical keyboard.

Does it work:
As it is, the glove is not fully functional, this is because there are not enough keys on the glove. This is not really an issue as the glove is only a proof of concept and it is certainly successful in this aspect. The glove could be made more successful in tests by making a second glove, adding more keys and implementing two key characters.

Evaluation: Potential Improvements



Issue	Solution	Result
There are not enough keys on the keyboard to comprise every letter symbol and short response required.	This could be improved by having two gloves and adding extra keys to the existing glove.	This would result in a pair of gloves which allow the user the same range of options as normal keyboard wood which will create a more streamlined user experience.
As the glove is tethered to a computer via a USB cable, it is rather unwieldly and limits its useful applications.	Integrating a Bluetooth transmitter into the design would mitigate this and would allow complete freedom of movement. 	Having integrated Bluetooth into the device, it will become far easier to use and will no longer be subject to criticism concerning the use of a USB cable.
The keyboard does not actually use sign language and it would be a lot easier for deaf/mute people to use it it if it did.	This could be solved by more sophisticated algorithms or a future glove design which uses accelerometers and flex resistors rather than buttons	This would result in a glove which is far easier to use and more natural to use. A possible schematic for this glove is shown below 
The electronics on the back of the glove are bulky and restrict movement.	The electronics could be minimised by using a smaller microcontroller such as an attiny85 and by using a custom PCB shown on the right which would contain all of the diodes as well as smaller wires sewn into the glove 	This improvement would result in a far more comfortable use of the glove. It would also make the glove more aesthetically satisfactory and decrease the overall weight making it far easier to carry around.