



BEng (Hons) Electronic Engineering



**GRIOT**

## **Generative Reasoning for Internet of Things**

*/ 'gri:ou/ gre-o/*

**Aaron Vidal**

Solent University | Department of Science and Engineering

Tutor: Simon Sagers

24/04/2024

### **Abstract**

Humans create tools to amplify our innate abilities. Initially adopting bones and large sticks as vectors of our strength, we now find a new fulcrum upon which to leverage our intelligence, Large Language Models. The advent of 'Generative Reasoning' has ushered in a new paradigm of Human Computer Interaction (HCI) as these models present the emergent ability to complete complex multi-step tasks. Novel use cases have unlocked qualitatively different experiences allowing us to transform velleity to volition as LLMs help us exercise our intent, handing us the reins and giving us higher agency over our actions. Our computers are about to control themselves, which leads to questions on how we might now effectively interact with them. Speech is the highest throughput form of communication humans can (currently) access - in this way, to effectively communicate intent we must interact with our devices as we do each other, with natural language. This work explores this new dawn of interaction to present Natural Language Interface (NLI) Hardware.

This report is submitted in partial fulfilment of the requirements of Southampton Solent University for the degree of BEng (Hons) Electronic Engineering





The most important thing a person can build is their connection to others.





## Table of Contents

Table of Contents .....	3
1. Introduction.....	9
1.1 Tools .....	9
1.2 Intent.....	10
2. Aims.....	13
2.1 Objectives.....	13
3. Research and Review .....	14
3.1 Entropy in Interaction.....	14
3.2 Friction .....	15
3.3 Interaction Modalities.....	16
3.3.1 Touch.....	16
3.3.2 Voice.....	17
3.3.3 Speculative Modalities.....	19
Gesture Detection.....	19
3.3.4 Output.....	21
3.4 Interaction with Generative Models .....	22
3.5 Interaction Between Language Models and Themselves.....	25
3.5.1 Retrieval Augmented Generation (RAG) .....	27
3.6 Embodied Agents and Agentic Architectures .....	29
4. Project Methodology .....	31
4.1 Gantt Chart and Flowchart.....	32
4.2 Risk Analysis .....	33
4.3 Tools and Resources .....	35
5. Product Design Specification.....	36
5.1 Natural Language Interface .....	36
5.1.1 Natural Language Interface Requirements .....	37
6. Candidate Solutions .....	38
6.1 Natural Language Interface .....	38
africus .....	38
pageR.....	41
Natural Language Human Input Device.....	46
6.2 Generative Reasoning .....	48
7. Product Design Development .....	49
7.1 GRiOT .....	49





7.2 Africus .....	50
7.3 pageR .....	51
7.3.1 Encoder .....	54
8. Implementation and Testing .....	55
8.1 africus.....	55
8.2 pageR.....	57
8.2.1 Encoder .....	59
9. Conclusion and Recommendations.....	61
System Usability Scale (SUS) .....	62
References .....	63
Appendix 1   Large Language Models and Generative AI .....	75
1.1 Evaluation .....	77
Appendix 2   Physical Media Consumption .....	79
Appendix 3   Fine-Tuning Myself .....	80
Appendix 4   Thoughts on forms of Human-AI interaction .....	87
Appendix 5   Agile Development for Hardware   Plantronics Case Study.....	88
Appendix 6   Gantt Chart .....	90
Appendix 7   Tools and Resources .....	91
Appendix 8   Natural Language Interface Process Flow .....	95
Appendix 9   M215HGE-L23 Panel Datasheet .....	96
Appendix 10   Raspberry Pi vs Jetson Nano Performance + Pricing.....	98
Appendix 11   Raspberry Pi 4B+ Power Requirements .....	100
Appendix 12   Evidence Sources .....	101
Appendix 13   Audio Analysis SPH0645 MEMS .....	103
Appendix 14   Tactile Inputs .....	104
Appendix 15   Display Analysis.....	105
Appendix 16   pageR Design Development .....	106
Appendix 17   GRIOT Logo Market Survey.....	108
Appendix 18   Project Evaluation .....	109
Appendix 19   SUS Improvement .....	112
Appendix 20   Encoder.....	114
Appendix 21   pageR Pinout .....	118
Appendix 22   M3 Bolt Testing.....	119





Figure 1   "The Transformer" (Google Research, 2017) .....	9
Figure 2   The Africus Monolith at 'The Dawn of Man' (2001: A Space Odyssey, 1968) .....	10
Figure 3   Velleity (OED, 2023) .....	10
Figure 4   Feature Completeness vs Complexity.....	11
Figure 5   Remarkable Type Folio (left) Minimalist Phone (right).....	11
Figure 6   Gulf of Execution .....	12
Figure 7   TP-7 Field Recorder .....	15
Figure 8   Humane Ai Pin .....	15
Figure 9   Modality (OED, 2023) .....	16
Figure 10   Analogue Inputs.....	16
Figure 11   ASR Methods .....	17
Figure 12   Cross Correlation in Context .....	19
Figure 13   Gesture Detection using OpenCV, TensorFlow and MediaPipe .....	19
Figure 14   DeWave in Action (UTS HAI Research, 2023) .....	20
Figure 15   Augmental Co-Founder Using 'MouthPad^' a mouthguard-based tongue controlled HID to control a phone .....	20
Figure 16   Galton Board .....	22
Figure 17   Prompting to Evoke Reasoning (Kojima, et al., 2022).....	22
Figure 18   Illustration of Intent-Based UX Framework .....	23
Figure 19   Intent Based UX .....	23
Figure 20   WebUI for Google Gemini (Left) OpenAI ChatGPT (Centre) Anthropic Claude (Right)	24
Figure 21   An example of an assistant that renders the weather information in a streamed component (Vercel, 2024) .....	24
Figure 22   Example of GPT-4 Structured Output .....	25
Figure 23   System Block Diagram of a Hypothetical LLM Operating System (Karparthy, 2023) .	25
Figure 24   Lost in the Middle: How Language Models Use Long Contexts (Liu, et al., 2023).....	26
Figure 25   Generic RAG Structure .....	27
Figure 26   Reflex Agent Diagram .....	29
Figure 27   Voyager .....	29
Figure 28   Gorilla v1 .....	30
Figure 29   An overview of OS-Copilot framework (Wu, et al., 2024) .....	30
Figure 30   Flow Chart.....	32
Figure 31   Risk Matrix.....	33
Figure 32   Natural Language Interface Block Diagram .....	36
Figure 33   africus Block Diagram .....	40
Figure 34   MEMS Microphone Test Setup .....	41
Figure 35   Candidates for Tactile Input.....	43
Figure 36   Testing Setup (pageR Display).....	44
Figure 37   Language Model Logic.....	46
Figure 38 Natural Language Interface Human Input Device (High Level Overview) .....	47
Figure 39   Proposed GRiOT System Interconnectivity .....	48
Figure 40   GRiOT System Design .....	49
Figure 41   System Design (africus).....	50
Figure 42   Design Inspiration [1 to 3 Top Row   4 to 6 Bottom Row].....	51
Figure 43   pageR 3D Printed Prototype Model .....	53
Figure 44   Encoder Assembly and Wheel CAD .....	54
Figure 45   Encoder Wheel Development .....	54
Figure 46   Encoder Assembly .....	54





Figure 47   africus.....	55
Figure 48   Africus Process Flow.....	56
Figure 49   pageR process flow.....	57
Figure 50   Raspberry Pi Zero W with SPH0645 MEMS and ST7789V3 TFT display. ....	57
Figure 51   pageR Process Flow.....	58
Figure 52   Encoder Testing in Arduino Serial Console.....	59
Figure 53   Encoder Test Set-up Arduino.....	59
Figure 54   Hardware Setup for Rotary Encoder Test.....	60
Figure 55   Embedding at a High Level.....	75
Figure 56   Human Generated Text vs GPT-3.5 Generated Text (Fraser, 2024).....	75
Figure 57   OpenAI's Sora Output at Base Compute (Left) 4x Compute (Middle) 32x Compute (Right).....	76
Figure 58   Photoshop Generative Fill example (arm with phone was masked using the lasso tool).....	76
Figure 59   LLM Cost vs Performance (Singh, 2024).....	78
Figure 60   Model Evaluation Prompt (Singh, 2024).....	78
Figure 61   Instagram Data Download UI (left) Snippet from Data (right).....	80
Figure 62   JSON Dictionary Conversation Structure.....	81
Figure 63   Conversation Analysis.....	81
Figure 64   OpenAI Finetuning GUI.....	82
Figure 65   Fine-Tuning Runs.....	82
Figure 66   Fine-tuned Model Responses and Baseline. User: White, Assistant: Blue.....	84
Figure 67   Testing Output [8RPrx7JW @ 0.7 Temperature].....	85
Figure 68   Thoughts on forms of Human-AI interaction (Fahmi, 2024).....	87
Figure 69   Gantt Chart.....	90
Figure 70   Direct Memory Access (DMA) (Stoyanaov, 2021).....	91
Figure 71   RSA Encrypted SSH Connection to africus.....	93
Figure 72   RSA Encryption (Lim, 2019).....	93
Figure 73   Generic NLI Process Flow.....	95
Figure 74   screengrab of 'top' command running on Raspberry Pi 4B+.....	98
Figure 75   screengrab of 'top' command running on Jetson Nano.....	98
Figure 76   Jetson Nano Pricing.....	99
Figure 77   Jetson Orin Nano Pricing.....	99
Figure 78   Spectrogram of Captured Audio 1.....	103
Figure 79   Spectrogram Generating Python Script.....	103
Figure 80   OLED-128O064D-BPP3N00000 General Specification Datasheet (Vishay, 2024) ..	105
Figure 81   TFT LCD Display Module Datasheet (EastRising, 2024).....	105
Figure 82   pageR initial design development render.....	106
Figure 83   Hand Breadth.....	107
Figure 84   Hand Length.....	107
Figure 85   GRiOT Logo Vote – Instagram (Left, Centre) Snapchat (Right).....	108
Figure 86   Amended Gantt Chart.....	109
Figure 87   Example of Completed Improved SUS.....	113
Figure 88   Bill of Materials (Encoder).....	114
Figure 89   Encoder Assembly CAD.....	114
Figure 90   Phases in Quadrature Encoding.....	114
Figure 91   Initial Rotary Encoder Test Script.....	115
Figure 92   Correct Implementation.....	115





Figure 93   Initial Design.....	116
Figure 94   Practical Implementation   10k Resistor 2x 1k Resistor 2x 100uF Capacitor .....	116
Figure 95   Encoder Circuit Diagram .....	117
Figure 96   pageR Pinout (left) Raspberry Pi GPIO (right) .....	118
Figure 97   Bolt Testing .....	119





Abreviation	Description
IoT	Internet of Things
AI	Artificial Inteligence
HCI	Human Computer Interaction
HMI	Human Machine Interaciton
HID	Human Input Device
BCI	Brain Computer Interaction
LLM	Large Language Model
NLI	Natural Language Interface
UI	User Interface
UX	User Experience
ASR	Automatic Speech Recognition
API	Application Programming Interface
TTS	Text to Speech
STT	Speech to Text
GPU	Graphics Processing Unit
sEMG	Scalable Electromyography
EEG	Electroencephalogram

Table 1 | Nomenclature







# 1. Introduction

## The way we as humans interact with technology is changing.

With new advances in the size reduction of powerful components, the idea of ubiquitous computing (integrating processors into everyday objects) has majorly increased in practicality and feasibility (Weiser, 1991). This emergence of interconnectivity has led to the development of a new technological ecosystem, the 'Internet of Things'.

Since 2022 there has been what can only be described as a Cambrian Explosion in the development and mass adoption of technologies based on the Transformer Architecture presented in the 2017 paper 'Attention is All You Need' by Google Research

Appendix 1 offers an expanded explanation on these tools.

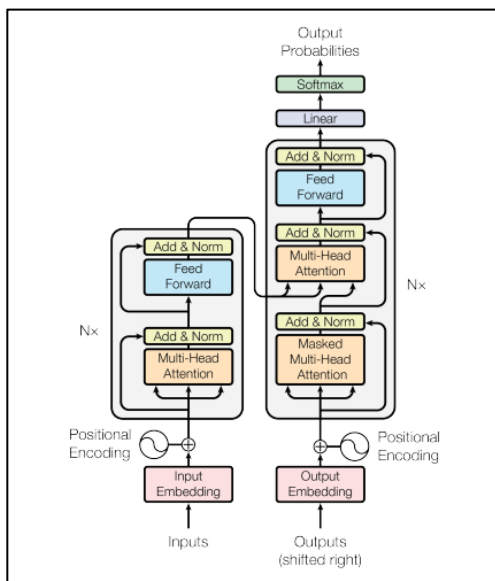


Figure 1 | "The Transformer" (Google Research, 2017)

Since implemented by companies like OpenAI, Large Language Models and other "Generative AI" tools have presented themselves as a blank canvas with almost infinite scope for innovation (Pinaya, et al., 2023)

Ubiquitous computing, which involves embedding processors into everyday objects,

aligns closely with the concept of generative reasoning. This integration enables these objects to participate in complex decision-making processes (Kojima, et al., 2022), adapt to unique user demands, and offer a more personalized and responsive experience. The incorporation of generative AI into the Internet of Things enhances the autonomy of devices, enabling them to handle a broader range of tasks without the need for explicit programming for each scenario.

Emergent abilities, as noted by Wei et al. (2022), offer novel solutions to previously inconceivable tasks, including self-guided reasoning on a large scale (3.6 Embodied Agents and Agentic Architectures). These developments present exciting possibilities in enhancing device intelligence and user interaction.

This current technological boom has been described as the 4<sup>th</sup> industrial revolution (World Economic Forum, 2016) – the first industrial revolution paving the way for the 5-day work week. In this way there is scope for generative technology to free us from monotony, leaving us time to focus on what is important.

## 1.1 Tools

**Tools are the fulcrum by which we lever our innate abilities.**

Millions of years of evolution facilitated our development from primordial organisms with rudimentary intelligence into an almost symbiosis with tools we create. Harnessing our cognitive ability to interact with the world around us. This exceptional ability to innovate, create and use tools marks the sharp departure from our ancestral lineage, setting us apart from other species (Orban & Caruana, 2014).

Our hands are effectively the first tools we have had access to and through evolution, appposable thumbs enabled us to perform qualitatively different functions that helped propel us forward as a species. We discovered fire and were in turn able to gain more nutrients from the food we ate, increasing our brain capacity and continuing this trend.





Figure 2 | The Africus Monolith at 'The Dawn of Man' (2001: A Space Odyssey, 1968)

The opening scene of Stanley Kubrick’s “2001: A Space Odyssey” presents us with the Africus Monolith. In the film’s canon, the monolith serves as the catalyst for the evolution of intelligence in early hominids – endowing them with the knowledge to wield bones as tools. An allusion to the real-world leap in intelligence.

From using bones to enhance our physical capabilities we now see the emergence of Large Language Models as the fulcrum we may use to leverage our own cognition.

Initial advancements in the field of ‘Internet of Things’ saw our tools embedded with microprocessors allowing them to utilise wireless communication protocols like WiFi and Bluetooth. Enabling communication between people, processes, and things (Oracle, 2020).

This led to the development of “smart speakers” as an embodiment of a personal assistant (Zwakman, et al., 2021).

Currently available personal assistants lack the personalisation and customisation features many have come to expect since the dawn of personal computing. Whereas users prefer an agent which is highly agreeable and has higher consciousness and emotional ability (Zargham, et al., 2022)

Appendix 3 presents an implementation of fine-tuning on writing style to create an agent that embodies this.

This journey of innovation is not just about the tools themselves, but how we engage with them and in the case of large language models - how they engage with us.

### 1.2 Intent

**“Technologies have ways of seeing that they lend us when we use them.”** (Thorn, 2022)

<p><b>velleity</b> /ve-'lĕ-ə-tē/</p> <p><b>noun:</b> ‘a wish or inclination not strong enough to lead to action’. ‘The lowest degree of volition’</p> <p><b>etymology:</b> Derived from the Latin noun “velleitas”, which in turn comes from the verb “velle” meaning “to wish” or “to will”, the word “velleity” signifies a minor wish or a volition so weak it fails to translate into action. A desire that is not accompanied by an effort to realise it.</p>
--

Figure 3 | Velleity (OED, 2023)

A language model’s reasoning ability allows it to act as an extension of human cognition (Bubeck, et al., 2023), enabling users to bridge the gap from theoretical understanding to practical application – facilitating a transition from velleity to volition.

We often use technology with an end goal in mind. This ability to exert flexible instrumental control over such an environment is often referred to as agency (Norton & Mimi, 2020).

It has been commented that our digital devices are designed to generate cues that provide the illusion of agency, it is commonly observed that users frequently use their smartphones without any genuine intent (Madary, 2022).

**“Attention is a resource – a person only has so much of it.”** (Crawford, 2015)

In this information age attention becomes a traded commodity much like any other - oil, gold, etc. It is in the best interest of those who profit off a consumer’s time to have them trapped in





dopamine-driven feedback loops (Cutillo, 2021) as attention is akin to capital. Robbing users of their agency by distracting them through the vectors of push notifications and ‘dark patterns’ (Zagal, et al., 2013)

Zelwinski’s Law of software envelopment humorously suggests that “all software evolves till it can eventually read mail” this concept speaks to the idea that many pieces of software (and hardware) include functions that may not be necessary (Zawinski, 1995). His comment captures the pressure on technology solutions as those that don’t expand are “replaced by ones which can” as such we are left with systems that do a lot, not very well.

We are presented with a sliding scale for technology, from “minimalist and limited” to “comprehensive but bloated”. The idea of bloat in such systems trends negative as users become overwhelmed by choice.

Smartphones have developed this bloat. Whilst very comprehensive, they’ve become ‘too useful’ and are now very good at keeping our attention.

We can plot feature completeness vs ease of use.

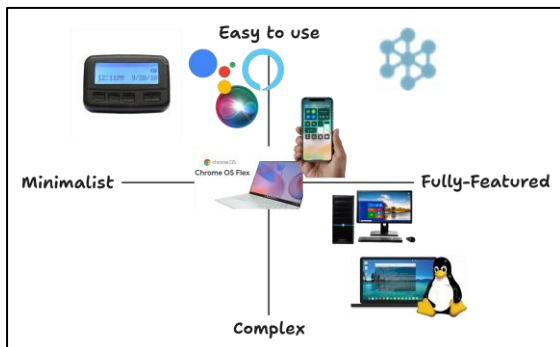


Figure 4 | Feature Completeness vs Complexity

On the left we have a pager, which is an example of minimalist technology: it is straightforward with a limited range of ability, but this simplicity leads to its ease of use. Voice assistants such as Google Home, Siri and Alexa are as simple to use due to their implementation of voice however they lack functionality when compared to a personal computing device like Windows PC or GNU Linux which offers easier root control to the user. Chrome OS offers a balance between ease of use and features via the Chrome Store which edges towards a full featured repository however

you are unable to run executable files limiting use cases (Google, 2024).

The EU’s Digital Markets Act partially mandates that Apple and other tech companies must permit the use of third-party app stores (European Commission, 2023). This regulation, in turn, offers users greater control over their devices (albeit by mandate).



Figure 5 | Remarkable Type Folio (left) Minimalist Phone (right)

Currently, there is a market for 'low fidelity' technology, aimed at reducing digital distractions and enhancing focus. A notable example is the consumer brand reMarkable, which launched the Type Folio attachment for their devices. This product is marketed with terms like 'experience-focused typing' and 'distraction-free writing' (reMarkable, 2024), emphasizing its appeal to users concerned about the attention-sapping effects of conventional devices used for similar tasks.

In a similar vein, the concept of a 'Minimalist Phone' has emerged, an Android user interface designed specifically to curb 'mindless scrolling and eliminate distractions.' According to (Moravek, 2023), this approach provides users with advanced controls to filter notifications and the ability to block or hide apps. This development underscores the growing demand for technologies that help users maintain focus and manage digital consumption more effectively.





**Intentional use of a piece of technology is an expression of your agency.**

Currently, we are witnessing a market pull, towards analogue devices. Evident in the resurgent interest in the consumption of vinyl records and film photography – trends that can be seen within the market (Statistica, 2016) and that are projected to continue (Statistica, 2023). In a notable milestone, vinyl sales eclipsed CD shipments in 2022.

Appendix 2 presents the figures for this data.

The British Recorded Music Industry (BRMI) reports the rise of streaming services has led to an overall increase in revenue, with a decline in CD and digital album sales. (British Recorded Music Industry, 2023). However, the BRMI has also observed an increase in vinyl and cassettes during this period. Signalling a nuanced consumer appetite, a contrast between “high tech” and “low fidelity” illuminating the market’s desire for innovative technology but also a hands-on, sensory experience. This ‘remarkable resurgence’ (Business Insider , 2023) marking a yearning for authenticity and agency, away from the “black-boxed” nature of digital consumption.

Consider An Analogue film camera:

- Manually loading the reel
- Adjusting the focus
- Setting the Aperture
- Activating the shutter
- Developing the Reel

This friction in user interaction (see 3.2 Friction) adds to the user experience, making each use of the device more intentional. This process flow has since been abstracted to a simple:

- Open Camera App
- Press Shutter Button

It is argued that, designing friction into interactions through the introduction of micro boundaries, can have positive effects (Cox, et al., 2016). With users opting for applications with added friction, increasing fulfilment (Mejtoft, et al., 2019).

In 2001 IBM researcher Cameron Miner commented in the journal Communications of the Association of Computing Machinery on “Pushing Functionality into Even Smaller

Devices”. This almost prophetic article discusses how technology had already begun to trend towards the ‘inclusion of additional functions in a single device’. He states that the added functionality had led to increased difficulty in user interaction as interaction/output interfaces like displays and keypads had decreased in size.

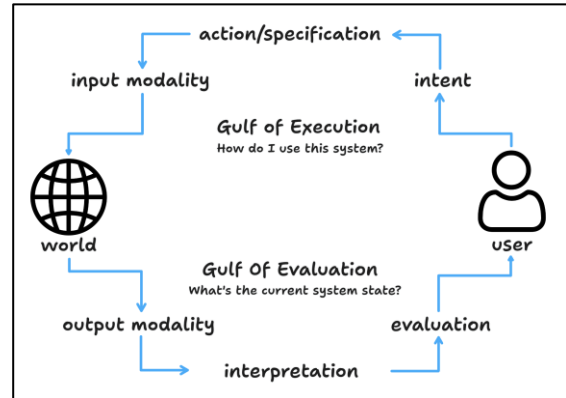


Figure 6 | Gulf of Execution

To address this ‘gulf of execution’ (Norman & Draper, 1986) where the user struggles to express their intent, Miner proposed breaking the device apart and ‘putting the appropriate pieces in the appropriate places.’ Closing the gulf by making the input modalities more easily accessible to the user. Wearable earpieces for audio output, wearable input devices near fingertips, etc. connected wirelessly to a device.

Generative reasoning can close this gulf more effectively, without modifying the device topology. A generative agent on device can evaluate user intent, specify what needs to be done and act on the user’s behalf, either as the input modality (HID) itself or completing the task and displaying the output (see. 3.8 Interaction with Generative Models and 3.10 Automation is Magic)

**The delta between theory and execution is shrinking. Soon the only limit will be your imagination.**





## 2. Aims

What's the most effective way to interact with something that can do (almost) anything?

### Natural Language Interface

- Language is like magic, and speech alone is the highest throughput form of communication humans have access to.
  - For a truly personal computing experience, we should interact with our devices as we do each other - with natural language.

### Generative Reasoning

- Automation, with self-guided logic
  - You only have to push the first domino; GRiOT will figure out how to topple the rest.
  - A **self**-operating system
  - Intent Guided OS
    - **Praxis:** The process by which a theory or skill is put into practice

### 2.1 Objectives

- Research Generative AI and Language Models
  - Gain understanding of some of the currently available generative models and how they can be implemented in a natural language interface.
- Consider interactions between users, technology, and language models (how it currently stands, how it can be improved)
- Create hardware based on a Natural Language Interface that can be used to interact with a Generative AI model.
- Develop a set of software tools that will allow a user to perform tasks using natural language commands.





### 3. Research and Review

As defined within the Aims and Objectives, the intent of this project is to create a Natural Language Interface and the Governing architecture. This section includes research on the following:

- Interaction and Interaction Design
- Uses of Generative AI

#### 3.1 Entropy in Interaction

In his 2001 thesis David J. Ward discusses ‘Adaptive Computer Interfaces’ wherein he states ‘existing devices for communicating information from people to computers are either bulky, slow to use, or unreliable’ (Ward, 2001).

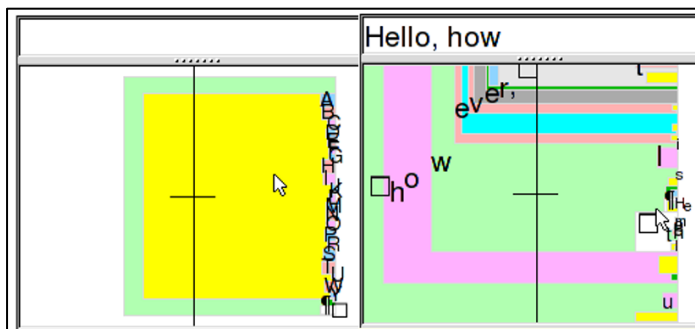
He comments on Shannon’s findings in ‘Prediction and Entropy of printed English’ (Shannon, 1951), where the entropy (average information content of an outcome ) of English was estimated to be 1 bit per character (Ward, 2001). Derived from sample data gathered by having people predict the next character in a sequence of text, with the process continuing till they guessed correctly.

Conventional keyboard input requires a user to make one (for lower case) or two (for upper-case, including shift key) gestures per character. Ward suggests that existing keyboard are inefficient by a factor of six, due to each gesture being a selection from 1 of roughly 80 keys (Ward, 2001). With the entropy of English being 1 bit per character the keyboard has the capacity to read:

$$\log_2(80) = 6.3 \text{ bits per gesture}$$

Equation 1 | Keyboard Read Capacity in bits per gesture, base 2 representing the binary bit (Ward, 2001)

He attributes such inefficiency to the redundancy of text, as users must type all characters in a document (Ward, 2001). Additionally noting that humans are capable of finer motor movements than single contact keyboard presses.



Equation 2 | Dasher at Idle (left) Selecting Letters (right)

To leverage this discrepancy Ward presented Dasher, a ‘zooming interface’ which utilises Bayesian language modelling and continuous two-dimensional gestures to facilitate typing (MacKay, 2002). Characters are presented to the user according to their statistical likelihood of following a sequence of text.

To use Dasher, you point where you want to go as if driving a car, this inherently grants the user agency in the interaction as the user interface generatively changes in correlation to what you intend to write - providing intent based user experience.

Ward presents Dasher as a prime and early example of how language models may help us more effectively communicate our intent in Human Computer Interaction.





### 3.2 Friction

By analysing common methods of accessing to the internet, we learn that the most common Human Computer Interaction to take place involves a smart phone (DataReportal, 2024) (StatCounter, 2024). The common smart phone requires at least two (vision and touch) and at most four (including hearing and speech) modalities to use.

$$\text{Friction} = \# \text{ of inputs} + \text{time to output}$$

Equation 2 | Friction in User Interaction (Imel, 2021)

The concept of friction is commonly discussed as the antithesis to fluid user interaction (1.2 presented arguments that friction could aid the user experience). Where Ward aimed to reduce the number of inputs by implementing his ‘zooming interface’ (MacKay, 2002), others may aim to reduce time to output or ‘latency’ by increasing processing speed of a device to improve performance (Partiwala, et al., 2020).



Figure 7 | TP-7 Field Recorder

The Stockholm based consumer electronics company Teenage Engineering comments on the ‘importance of zero friction’ discussing how it is possible to remove steps from thought to action:

*“let’s compare TP-7 to the most common recorder today – the smartphone:*

*once you’ve turned on and unlocked your phone, found the recording app and started recording, you’ve probably forgotten whatever you had in mind.*

*when recording on TP-7: press the memo button and start talking.”*(teenage engineering, 2024)

Teenage Engineering is famous for their “minimal” and “visually based design” (Abrahamsson & Eriksson, 2010) as the design maxim “form follows function” suggests, this design style lends itself perfectly to the idea of zero friction. The simplicity not only enhances the visual appeal but also complements the user experience by minimizing complexity. The TP-7 is also an example of a single purpose smart device, allowing users to access recordings via a mobile app and Bluetooth connection.

Senior AI Research Scientist at NVIDIA Research Jim Fan commented on the launch of Humane’s Ai pin as “a great stride towards ambient intelligence” which he describes as AI “fading into the background- emerge[ing] naturally when you need it” (Fan, 2023).

The AI Pin’s primary input/output modality is aural, such that it takes speech following a tactile input from the user, returning synthesized voice (Humane, 2023).



Figure 8 | Humane Ai Pin





### 3.3 Interaction Modalities

**modality** /mə(ʊ)'dælɪti/

**noun:** 'modal quality'.

'a particular mode in which something exists, is experienced or expressed.'

**etymology:** Derived from the Latin adjective "modalis" which is formed from "modus" meaning "measure", "bound" or "manner"

Figure 9 | Modality (OED, 2023)

Interaction between human and device is commonly referred to Human Computer Interaction, personal computing devices often require a Human Input Device (HID) such as keyboard or mouse to interact with them. We often abstract these interactions through use of transducers which can convert the mechanical energy from the user interaction into electrical energy that can then be decoded by some form of

analogue or (more often) digital logic embedded within the device (Lang & Argwal, 2005). These, often referred to as 'input' modalities define how we interact with our devices. The following sections present some of these and their contexts.

#### 3.3.1 Touch

Touch is the most common modality we utilise as humans, computer interaction or otherwise. Our skin and its various nerve fibres and cells acting as electron transducers allowing us to sense the world around us (Leger & Luengo, 2023). In this way we create a 'digital skin' for our devices through implementation of touch-screens which provide graphical representations of buttons that can be resized according to space on the screen (Zhang, et al., 2023) (Lee & Zhai, 2009).

While touch has arisen as the predominant interaction modality (Buxton, 2012), it alone often lacks kinaesthetic and tactile feedback (Barret & Krueger, 2007), as a result some form of feedback loop is often required to confirm input (sound, haptic buzzer, etc).

As '1.2 Intent' presents, there is currently a pull towards such analogue haptic experiences.




Clicks Keyboard Case	Apple Watch Digital Crown	Rabbit R1 analogue scroll wheel
		
<p>The clicks keyboard case presents an analogue input method like that seen on older phones prior to the release of the iPhone and the initiation of a market pull towards touch interfaces. The release of such an addition further implies that there is a market for such tactile experiences (Clicks, 2024).</p>	<p>The Apple Watch, whilst sporting a touch screen also features a "digital crown" (rotary encoder) as a tactile input (Apple, 2024). Acting as a button when depressed. Additionally, the watch features a "taptic engine" a miniaturised linear actuator which serves to provide haptic feedback to the user (iFixit, 2019).</p>	<p>During the initial development of my Natural Language Interfaces the rabbit r1 a single purpose ai device was launched. It too features an analogue scroll wheel, a feature I intended to include for navigation. (see 5 Candidate Solutions)</p>

Figure 10 | Analogue Inputs







### 3.3.2 Voice

Speech is considered one of the most natural forms of communication between people (Juang & Rabiner, 2005). Continued development in Automatic Speech Recognition (ASR) technologies powered by Transformer architecture (Google Research, 2017), have enabled wide scale access to high quality text extraction from audio data (Radford, et al., 2022). Building from this OpenAI released their Whisper ASR model in via publicly available API allowing transcriptions to be generated from an mp3 sent via POST request to their RestAPI (OpenAI, 2024).

Further research in web interaction reveals that 52% of survey participants prefer using a voice-assistant for information search, citing ease of convenience (Statistica, 2022).

A primary goal of speech recognition in modern interfaces is to facilitate natural communication between user and device (Képuska, 2011). This can be achieved in a few different ways:

ASR Method	Description	Benefits	Drawbacks
Push To Talk	Physical input delimits the period of user utterance	Period of speech is clearly defined by user.  Useful in conditions with background noise	Requires physical interaction, less reflective of natural communication
Continuously Listening	Input is continuously processed, agent analyses context and intelligently responds	Closer to natural conversation as no user input is required. 'Ambient Computing'	Privacy  May interpret background noise as speech.
Wake Word	Explicitly request the attention of the device using a pretrained Keyword. Voice Activity Detection (VAD) awaits "unvoiced audio frames" indicating end of speech	Distinct wake word distinguishable from other noise  Wake word Customization	Requires custom classifier model based on MEL Spectrogram feature extraction and convolution (Yu & Zhou, 2022)

Figure 11 | ASR Methods

As this technology develops there is scope for a more symbiotic relationship to form between user and agent, at which point a continually listening system may act as a form of ambient intelligence.

In [Appendix 4](#), Fahmi (2024) provides a graphical representation on thoughts on forms of such Human-AI interaction.

Embedded devices are generally constrained by memory and energy requirements (Leow, et al., 2023). Before recently voice was often sent from a device via WebSocket or other method to be processed elsewhere. Improvements in microprocessor technology has seen components like the A12 Bionic and S9 facilitating on device machine learning capabilities, allowing users to query private topics such as locally stored biometrics and medical information without data leaving their device (Apple, 2023). As this trend continues (assuming model architectures scale towards efficiency) we may see high performance Generative AI running on device.





The Raspberry Pi platform, namely “Zero W” has demonstrated impressive ability in the domain of signal processing and IoT applications. As presented by (Janeck, 2022) it has been utilized for the investigation of wavelength and polarization of electromagnetic waves. Furthermore, Lima, et al. (2019) present the Pi’s ability to handle the real time collection and processing of signals without effecting system performance, specifically citing low CPU usage and temperature.





### 3.3.3 Speculative Modalities

#### Gesture Detection

Tolentino, et al. (2021) present a method for gesture recognition using RGB-D (RGB + Depth) output from a Kinect for Xbox One which features a depth sensor. Recent studies in diffusion model aided depth estimation present depth information from a single RGB image (i.e from a webcam or smart phone) pushing this functionality into lower fidelity devices with the help of machine learning (Saxena, et al., 2023) (Duan, et al., 2023) (Kar, et al., 2023).

Li, et al. (2023) introduces a method of continuous 3D Hand Pose Tracking. Converting commercial home assistant devices into active sonar systems by emitting a frequency modulated signal, capturing reflected audio with its MEMS microphone array (Berol, 2017) and cross correlating with the original signal. This in conjunction with a deep learning model trained on ground truth data from a Leap motion tracking sensor provide a representation of a hand in 3D space.

$$R_{xx}[m] = \sum_{n=-\infty}^{\infty} x[n]x^*[n - m]$$

Variable	Definition
$x[n]$	The transmitted 17-20kHz frequency
$x^*[n - m]$	The complex conjugate of the reflected signal shifted by a lag time $m$
$m$	The delay at which the correlation is calculated, representing the similarity of the signals at different time shifts

Figure 12 | Cross Correlation in Context



Figure 13 | Gesture Detection using OpenCV, TensorFlow and MediaPipe

Naganandhini, et al. (2022) present an implementation of a virtual painting system using OpenCV, TensorFlow and MediaPipe that helps young children draw simple images by drawing aerially. Similarly, Figure 13 presents an example implementation of such a system, developed on a Windows Machine in an evaluative capacity with the intent of testing on a Raspberry Pi.





Unfortunately TensorFlow Lite (used by the Pi) is bottlenecked by the Pi's CPU causing high latency, negatively affecting user experience – making this implementation impractical.

Sussillo, et al. (2024) at Meta's reality labs, present a non-invasive method of neuromotor signal detection using scalable electromyography (sEMG) enabling detection of Wrist Pose, Gesture Detection and Handwriting from a wrist-worn transducer array. This non-invasive method allows for high-bandwidth communication between humans and computers, enabling seamless translation of intent.



Figure 14 | DeWave in Action (UTS HAI Research, 2023)

Moreover, (Duan, et al., 2023) present another method of non-invasive bio-signal monitoring. While not gestural, DeWave uses electroencephalogram (EEG) signals collected from a cap that records brain activity through the scalp. This method facilitates the translation of brain dynamics into natural language by passing recorded data to a transformer model trained on a large corpus of EEG signals. Developments in brain-computer interfaces (BCI) aided by transformer architectures lay the groundwork for a new paradigm where interactions occur at the speed of thought.

## MouthPad

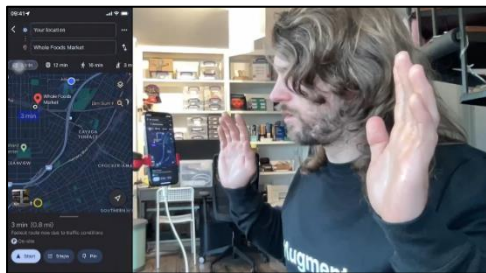


Figure 15 | Augmental Co-Founder Using 'MouthPad^' a mouthguard-based tongue controlled HID to control a phone

'MouthPad^' is a tongue-driven interface with a pressure-sensitive touchpad that allows users to control a device via Bluetooth connection (Augmental, 2024). While details surrounding the device hardware are vague - due to the small form factor, it likely uses an embedded microprocessor such as the ESP32. (Sruthi, et al., 2023), (Umare, et al., 2023) and others present use cases for the chip and its wireless capabilities such as monitoring and controlling pan/tilt of a camera.

The ESP32 community features many open-source libraries which enable additional functionality on the device. The ESP32-BLE-Mouse library and ESP32-BLE-Keyboard libraries both offer HID emulation (T-vk, 2020), connecting a (in MouthPad's case, touch-based) transducer to an ESP board would enable such functionality.





### 3.3.4 Output

The ‘Gulf of Evaluation’ discussed by (Norman & Draper, 1986) present output modalities as that which enable a user to understand the current system state. In any interface this is the most important component as a user must be able to evaluate the result of their initial input.

Visual	Aural
Rodríguez & Garcia Garcia (2014) present a successful implementation of a video wall using TN monitors and Raspberry Pi. Displaying the potential of a Raspberry Pi as a low cost method of displaying output.	Text to speech remains an important component in voice-based interaction systems. (Betker, 2023) presents Tortoise-TTS a text to speech system that applies advancements in generative image models to speech synthesis. Enhancements in quality and diversity of speech in such a model improve user experience.
(Suman, et al., 2023) and (Sun, et al., 2018) both present applications of a Smart Mirror as an interactive IoT based output modality. Furthermore, (Alboaneen, et al., 2020) present a comprehensive literature review, discussing how smart mirrors can provide real-time information and personalized content to users, enhancing their overall experience.	ElevenLabs (2024) offers Text-To-Speech audio generation via their API. It has been commented that they likely use a finetuned version of Tortoise-TTS (Betker, 2023) as the backend model for their API (jsmch, 2023), their partnership with Google providing access to the GPUs required to provide low latency access (Google Cloud, 2023).

Table 2 | Output Modalities: Visual and Aural





### 3.4 Interaction with Generative Models

The statistical modelling at the core of Dasher's language system is also central to Transformer architectures, capturing patterns in input data that can introduce bias (Vaswani, et al., 2017). We leverage this bias by directing the language models' responses with our inputs, commonly referred to as “prompts”.

Consider a Galton board, it's individual pins guiding the path of a ball. In the same way, our words shape the response by triggering specific trained pathways, leading to the generated output that reflects our prompt.

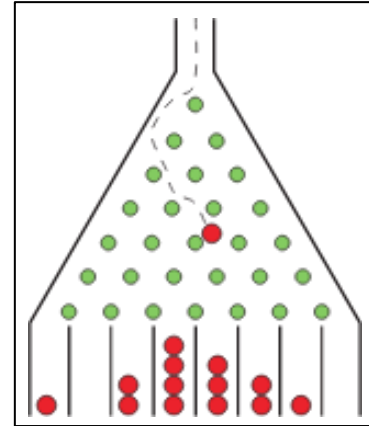


Figure 16 | Galton Board

The deterministic aspect of language models means that given the same input and conditions, a model will always produce the same output, this concept often referred to as perplexity (see Appendix 1) determines the predictability of the output of a statistical model (Jelinek, et al., 1977). Where low perplexity describes a predictable outcome and high perplexity describes a more unexpected outcome.

Exploiting these pathways using natural language is often referred to as ‘Prompt Engineering’ (Ye, et al., 2023) methods such as ‘Zero-Shot’ prompting are commonly used. Allowing users to elicit complex multi-step reasoning from a language model by simply adding the prompt “Let’s think step by step” (Kojima, et al., 2022) this outperforms other ‘Few-Shot’ (Brown, et al., 2020) methods which involve including a ‘pre-prompt’ with examples of the model’s expected behaviour that are added before you ask your question.

<p>(a) Few-shot</p> <p>Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now? A: The answer is 11.</p> <p>Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there? A:</p> <p>(Output) The answer is 8. <b>X</b></p>	<p>(b) Few-shot-CoT</p> <p>Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now? A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. <math>5 + 6 = 11</math>. The answer is 11.</p> <p>Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there? A:</p> <p>(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are <math>16 / 2 = 8</math> golf balls. Half of the golf balls are blue. So there are <math>8 / 2 = 4</math> blue golf balls. The answer is 4. <b>✓</b></p>
<p>(c) Zero-shot</p> <p>Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there? A: The answer (arabic numerals) is</p> <p>(Output) 8 <b>X</b></p>	<p>(d) Zero-shot-CoT (Ours)</p> <p>Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there? A: <b>Let's think step by step.</b></p> <p>(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. <b>✓</b></p>

Figure 17 | Prompting to Evoke Reasoning (Kojima, et al., 2022)

Few-shot and chain-of-thought prompting guide models through logical reasoning with minimal examples, offering a lighter alternative to fine-tuning which requires a model be trained on task-specific datasets (Lai, et al., 2021). Zero-shot prompting, on the other hand, is valuable for eliciting consistent responses without prior examples.





**Playground** Chat ↕
Your presets ↕

**SYSTEM**

Today we are practicing intent-based UX

for all contextually relevant responses consider the following framework

[what we want](what the user wishes to accomplish)  
 [what we have](what the user/you as an agent currently have at disposal - where contextually relevant you may ask the user to clarify what they have if it may lead to a expedient solution)  
 [what we need](what needs to be done in the most abstract terms to achieve what the user wants from what they have.)  
 [what we do](what you as an agent can assist the user with in completing this goal)

**USER**      I want to bake a cake

---

**ASSISTANT**    [What we want]  
 You wish to bake a cake.

                         [What we have]  
 To assist effectively, I would need to know if you have any specific type of cake in mind, the ingredients and equipment you currently have, and your level of baking experience.

                         [What we need]  
 In the most abstract terms, you will need a recipe, ingredients, baking equipment, an oven, and some instructions or guidance to bake a cake.

                         [What we do]  
 I can provide you with a recipe, guide you through the baking process, give you tips for success, and help troubleshoot if you run into any issues. If you provide me with details regarding the type of cake, your ingredients, and equipment, I can tailor the advice to your situation. Would you like to start by choosing a type of cake to bake?

Figure 18 | Illustration of Intent-Based UX Framework

Figure 18 presents an example of intent based chain of thought to facilitate an intent based user experience. In such an experience it is beneficial to split tasks into individual components. This has been illustrated through use of a system prompt in the OpenAI developer playground using GPT-4.

An appropriate query is decomposed into 4 stages:

In practice, the first three stages would be hidden to the user.

What we want	What we have	What we need	What we do
In this step the model understands and acknowledges the user's intent, by highlighting "what we want" the system focuses on the stated goal.	By noting "what we have" the system takes stock, evaluating the user's current resources, recognising current capabilities.	By considering "what we need" the system abstractly outlines the requirements to accomplish the goal, based on the discrepancy between what we want and what we have, without focusing on specific actions yet.	Finally, "what we do" facilitates action, translating the user intent and situational analysis into actionable advice that helps the user move forward towards their intent.

Figure 19 | Intent Based UX



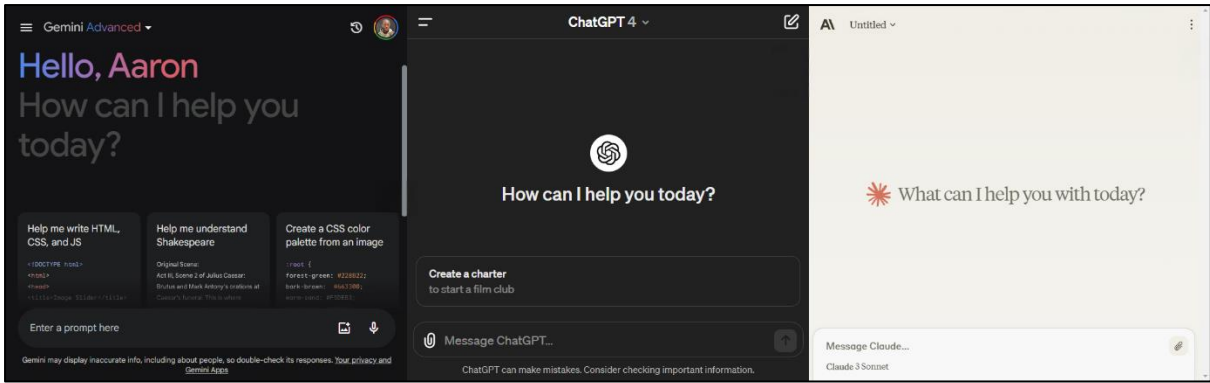
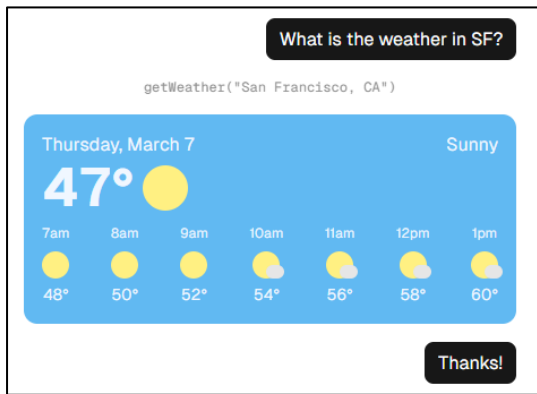


Figure 20 | WebUI for Google Gemini (Left) OpenAI ChatGPT (Centre) Anthropic Claude (Right)

At current these Generative AI assistants (see appendix 1.1) are commonly accessed through web browsers. The primary input modality being touch via a keyboard. Gemini offers voice input through the browser, ChatGPT utilises Whisper ASR (3.3.2 Voice) in its mobile app as a speech to text engine to convert user voice recordings into text.



If the interactions are generative, why can't the user interface be too?

Vercel proposed the third generation of their AI software development kit with Generative UI support, allowing developers to create chat bots with the ability to create 'rich, component based-interfaces' using function calling (discussed more in 3.5) to display pre made components such as weather and stock graphs updated with real time information (Vercel, 2024)

Figure 21 | An example of an assistant that renders the weather information in a streamed component (Vercel, 2024)

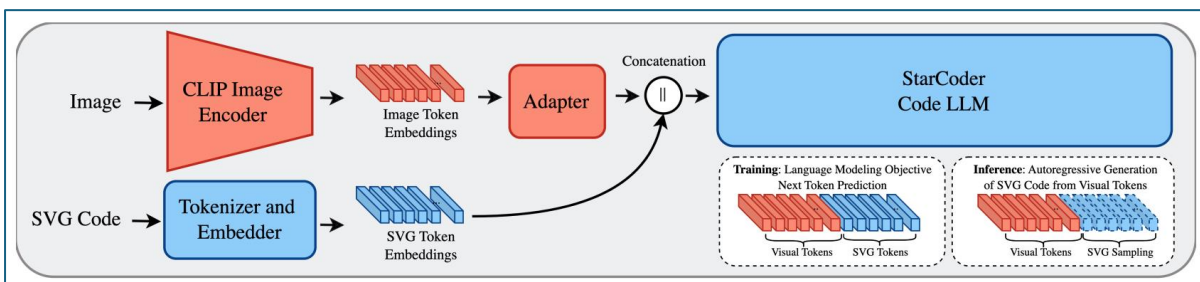


Figure 22 | StarVector Architecture (Rodriguez, et al., 2023)

(Rodriguez, et al., 2023) present StarVector, a fine-tuned model based of Star Coder LLM. Trained on Key-Value pairs of SVG image embeddings and the corresponding embedded SVG code the model can generate SVG code for any input graphic. This could be further improved to provide an agent with the ability to generate an SVG from natural language, creating required UI elements when contextually relevant.







### 3.5 Interaction Between Language Models and Themselves

Use of Few-Shot-COT prompting and or finetuning a language model will allow for the generation of structured data such as JSON Objects more reliably, allowing the model's output to be handled programmatically.

```
• (venv) aaron@vidalion:~/srv/multimodal/multimodal-api
$ python json-mode.py
Please enter your request: I want to book a hotel in
france

{
  "chain_of_thought": {
    "what_we_want": "book a hotel in France",
    "what_we_have": "information about preferred
dates, budget, and any specific requirements
(such as location, amenities, or star rating)",
    "what_we_need": "find and book a hotel in
France that meets the user's preferences",
    "what_we_do": "perform a web search to find
available hotels in France based on the user's
preferences",
    "actions": {
      "tools": {
        "web_search": {
          "query": "hotels in France"
        }
      }
    }
  },
  "response": "I will perform a web search to
find available hotels in France based on your
preferences. Can you provide me with the
preferred dates, budget, and any specific
requirements (such as location, amenities, or
star rating)?"
}
```

Some publicly available API such as GPT-4 have the option to toggle a structured output mode to allow for 'function calling'. (OpenAI, 2023) This allows a generative model to intelligently decide when it needs to act "outside itself" be that to acquire information not in its training data i.e. current weather, stock prices or to preform actions on your behalf. It preforms these actions by replying in structured JSON objects. Other methods of enabling JSON output involve prompting a model to explicitly reply in JSON in its pre-prompt.

Figure 23 presents an example of structured output featuring chain of thought concepts discussed in 3.4, we see a response generated in JSON schema, featuring an actions key with tools calls that can be passed to functions (see 3.6).

Figure 23 | Example of GPT-4 Structured Output

Structured output has also been used to create LLM operating systems (Packer, et al., 2023) which are able to use functions to perform a myriad of tasks including appending/retrieving information to/from an external memory store.

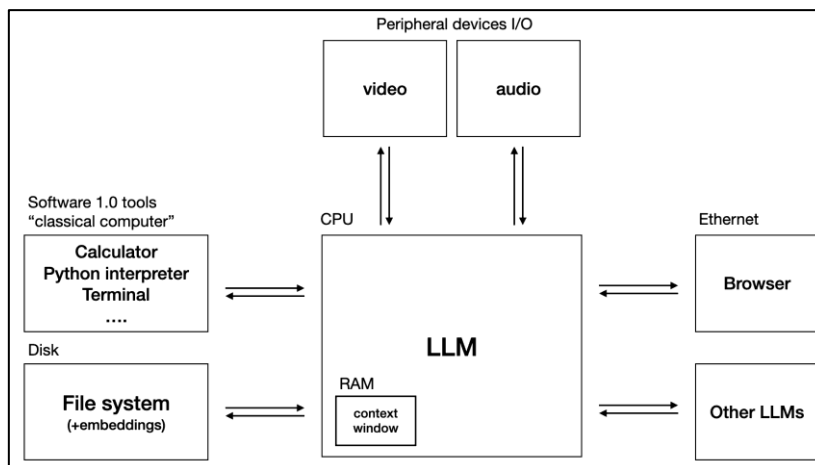


Figure 24 | System Block Diagram of a Hypothetical LLM Operating System (Karparthy, 2023)



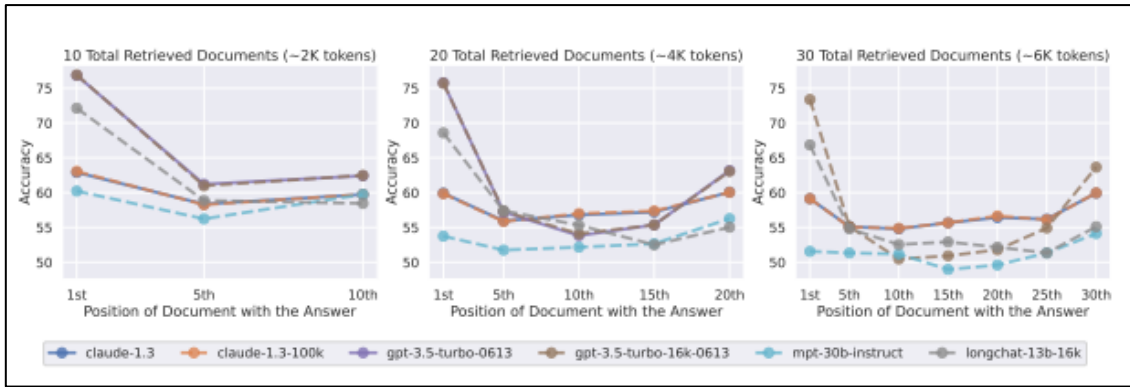


Figure 25 | Lost in the Middle: How Language Models Use Long Contexts (Liu, et al., 2023)

External memory stores become important as most LLMs have a finite context window, there is an upper limit to the number of tokens the model can process at once, capping the conversation to a set length unless alternate mechanisms are applied. Google has proposed a method to scale Transformer-based LLMs to infinitely long inputs (Tsendsuren, et al., 2024), however this architectural change hasn't seen implementation in publicly available models yet.

Models with larger context windows can sometimes have trouble correctly parsing through information stored within their context, struggling to “separate the wheat from the chaff” out of a long, incoherent window containing multiple sources of information. (Pinecone, 2023) (Liu, et al., 2023)

Fortunately, there is a solution to this.





### 3.5.1 Retrieval Augmented Generation (RAG)

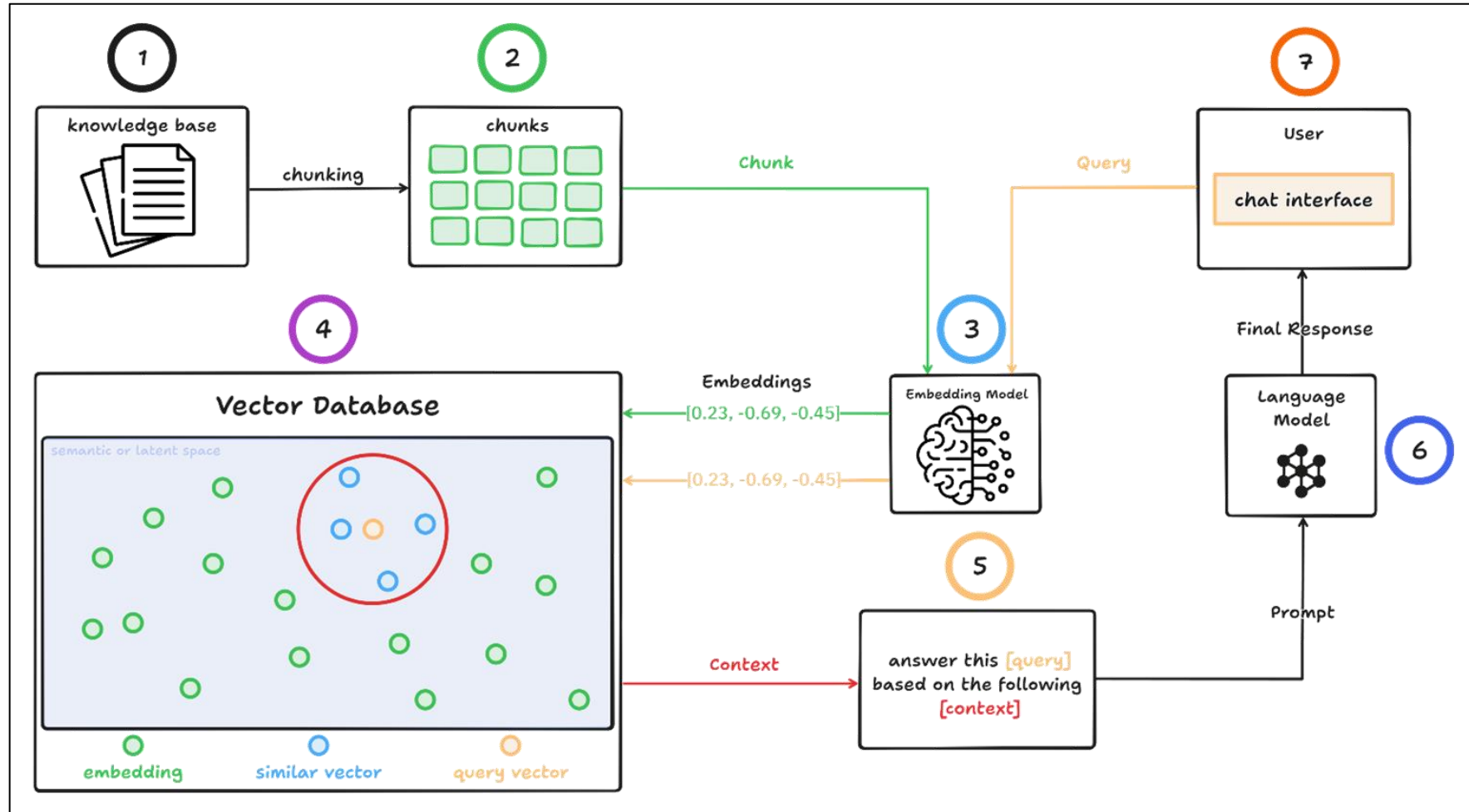


Figure 26 | Generic RAG Structure





Retrieval Augmented Generation, commonly referred to as RAG is a retrieval mechanism that can be used by a Language Model to extend its knowledge outside of the training data.

The typical RAG process can be broken down into 7 components: -

<b>1. Knowledge Base Construction</b>
The foundation of the RAG system, where relevant data outside the knowledge base of the LLM is compiled. This could be comprised of company documents/data or constantly updated memory store of user-model chat history. (Packer, et al., 2023)
<b>2. Data Preprocessing and Chunking</b>
Before it can be passed onto the next step the data must be split or chunked into manageable pieces to facilitate more efficient data handling. Care is taken to maintain the integrity of paragraphs, sentences, and words so they can retain their context and coherence, ensuring they can be matched with relevant queries later in the process. This process is commonly taken care of using natural language processing methods, the open source langchain library offers several functions for this purpose. (Langchain, 2024)
<b>3. Embedding Generation</b>
Text corpus is tokenized and fed to an embedding model which returns output embeddings as points in a semantic space (see Appendix 1).
<b>4. Vector Database Management</b>
The generated vectors are then stored in a vector database, an n dimensional array referred to as a semantic space or latent space where words exist in a point cloud
<b>5. Contextual Understanding and Retrieval</b>
Similarity search is performed to contextually return relevant pieces of information. A popular method is calculating the cosine of the angle between two vectors to find the cosine-similarity, this is done by finding the dot product between their normalizations (Steck, et al., 2024).
<b>6. Query Processing and Model Interaction</b>
Information from the vector database that is contextually relevant to the query along with the query itself is sent to the language model.
<b>7. User Interface and Response Generation</b>
A response is generated and returned to the user interface.

Table 3 | RAG Process

<b>Open-Source Implementation</b>
Using langchain text can be pre-processed before being sent to a local embedding model using the BERT architecture (Langchain, 2024) (HuggingFace, 2024). These embeddings can then be stored using Postgres as an SQL database with pgvector for cosine similarity and nearest neighbour search to return the top K relevant results to the language model (PostgreSQL, 2024).

Table 4 | Open-Source Implementation of RAG





### 3.6 Embodied Agents and Agentic Architectures

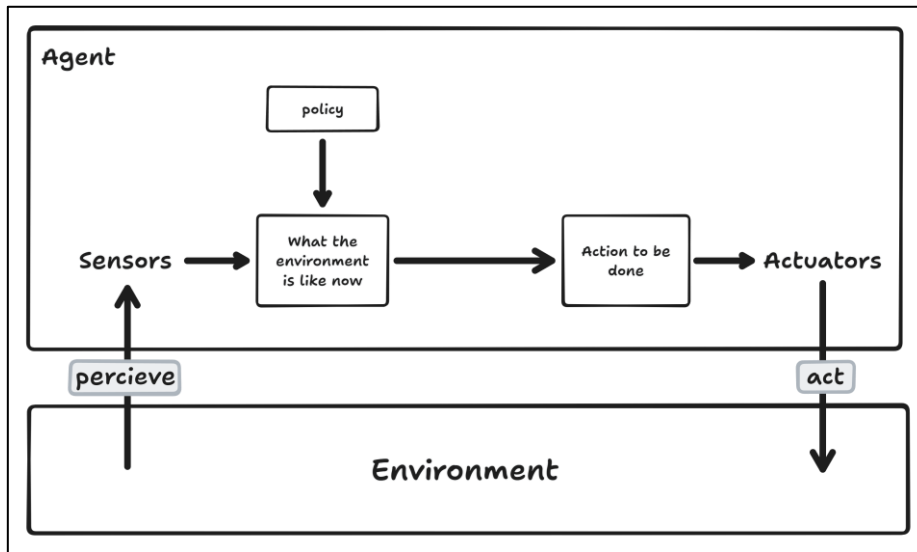


Figure 27 | Reflex Agent Diagram

An embodied agent describes an algorithm that can interact with its environment and learn from it. Using this ability to complete tasks autonomously (Duan, et al., 2022).

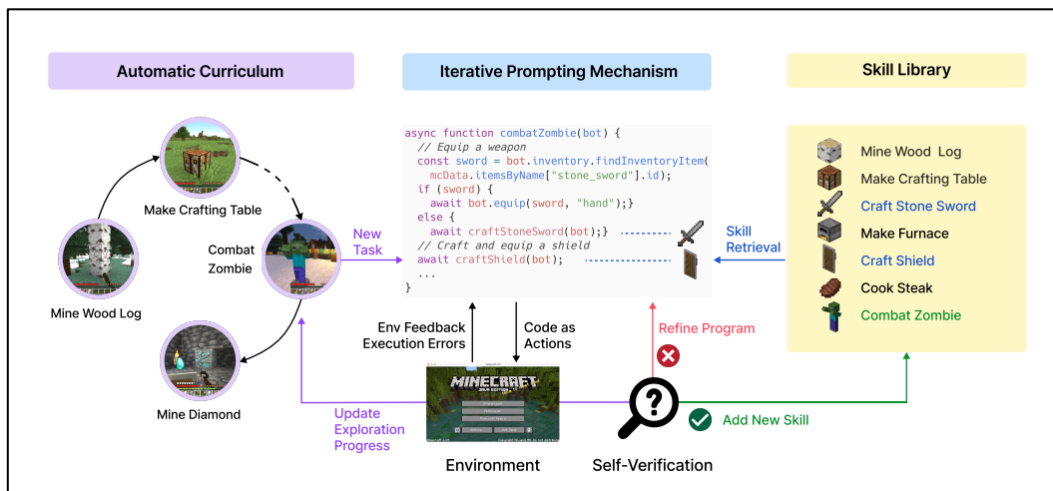


Figure 28 | Voyager

Wang, et al. (2023) display how Voyager an LLM-powered embodied agent in within the open-world sandbox game Minecraft - explores, acquires skills, and makes discoveries without human intervention. Voyager generates code to preform actions in game, receiving visual feedback using a finetuned CLIP image captioning model (Radford, et al., 2021) , and execution errors for self-verification, implementing RAG to save successful skills to its' 'skill library'. Recalling skills as needed and executing the code.

(Schick, et al., 2023) present 'Toolformer', discussing how Language models can teach themselves to use external tools via function calls to simple APIs, representing the inputs and outputs of such API as text sequences that can be learned and repeated by the model.



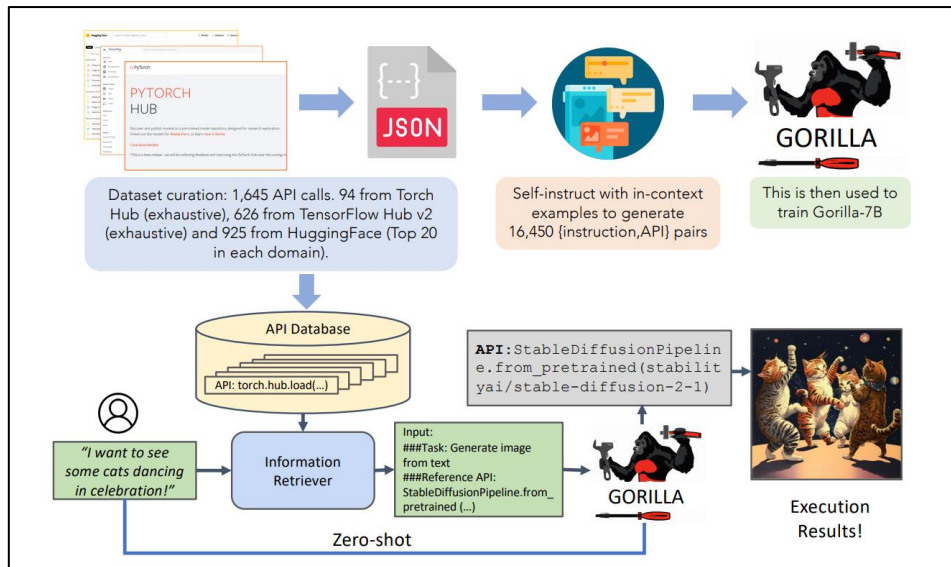


Figure 29 | Gorilla v1

(Ji, et al., 2024) finetune the “Deepseek-Coder-7B-Instruct-v1.5 6.91B” model on 65,283 pairs of question-function-answer pairs from Python Packages, Java and JavaScript Repositories, public API and Command Line Tools. In doing so they create Gorilla v2, a model with the ability to perform actions that make use of these tools. Such an agent can perform tasks like image generation, code creation and collecting real time information – using function calling and its knowledge base of public API.

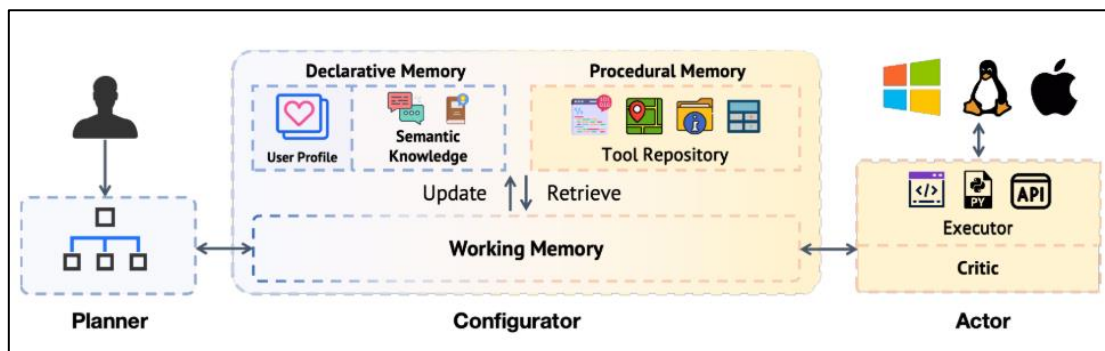


Figure 30 | An overview of OS-Copilot framework (Wu, et al., 2024)

(Wu, et al., 2024) present OS-Copilot, their generalist agent FRIDAY (Fully Responsive Intelligence, Devoted to Assisting You) displaying the ability to retrieve tools from a procedural memory, executing actions through command line, python scripts and API calls to control a computer.

These advancements display the potential of AI to not only understand and generate text, but also to translate linguistic instructions into actionable, context-aware behaviours in virtual or simulated settings.





#### 4. Project Methodology

*‘The best professionals know more than they can put into words.’*

*– Donald Schön, The Reflective Practitioner*

Donald Schön comments on problem solving in the fields of engineering, architecture, management, psychotherapy, and town planning (Schön, 1983). Presenting the concept of ‘Reflection-in-Action’, a dynamic and cyclical approach to the process of understanding and improving work.

Schön critiques traditional static planning methods, emphasizing a methodology where knowledge is not just applied but also created during action. Recognising that the best professionals have wisdom gained through a continual process of experience, reflection, and adjustment – embracing the complexity of solving real world challenges.

Aligning with Schön’s model, this project adopts such iterative approach, central to Agile methodologies.

Iterative cycles of analysis, synthesis and evaluation move beyond static planning.

<b>Analysis</b>
Understanding the issue, project needs and technical constraints
Applying theory to the issue sets stage for informed design decisions
<b>Synthesis</b>
Research and Development of design, software, or hardware components (for selection, modification, or manufacture)
Prototyping allows for ‘knowing-in-action’ as continuous development
<b>Evaluation</b>
Bridges theory and practice
Practical testing and critique inform refinements

Table 5 | Analysis, Synthesis, Evaluation

For example, the development of the africus Natural Language Interface featured these stages:

- Developing africus User Interface
- Developing africus Hardware
- Developing africus User Interface [Iterative Loop]
- Developing africus Hardware [Iterative Loop]

Each development step saw Analysis, Synthesis and Evaluation.

As hardware and software share a symbiotic relationship, the Agile methodology’s rapid prototyping was invaluable. Traditional methodologies such as Waterfall presented by Royce (1970) mainly rely on theoretical data analysis and interpreting component data sheets. Agile allows for a hand-on, experiential learning process - iterative testing in real world context e.g testing displays for the portable NLI or LEDs with africus. Statistical analysis, while valuable (and still used within the scope of this project) often overlooks the nuances of user interaction, by embracing a ‘learn by doing’ philosophy each iteration provided actionable insights.

Appendix 5 presents a case study where agile methodology is successfully implemented in a Hardware context, by (cprime, 2023) for the consumer audio brand Plantronics’ Research & Development Team.





#### 4.1 Gantt Chart and Flowchart

Appendix 6 presents the Gantt Chart, this was used through the duration of the project as an initial timeline – delimiting points of project achievement.

The flowchart presented below describes the project progress at a higher level of detail.

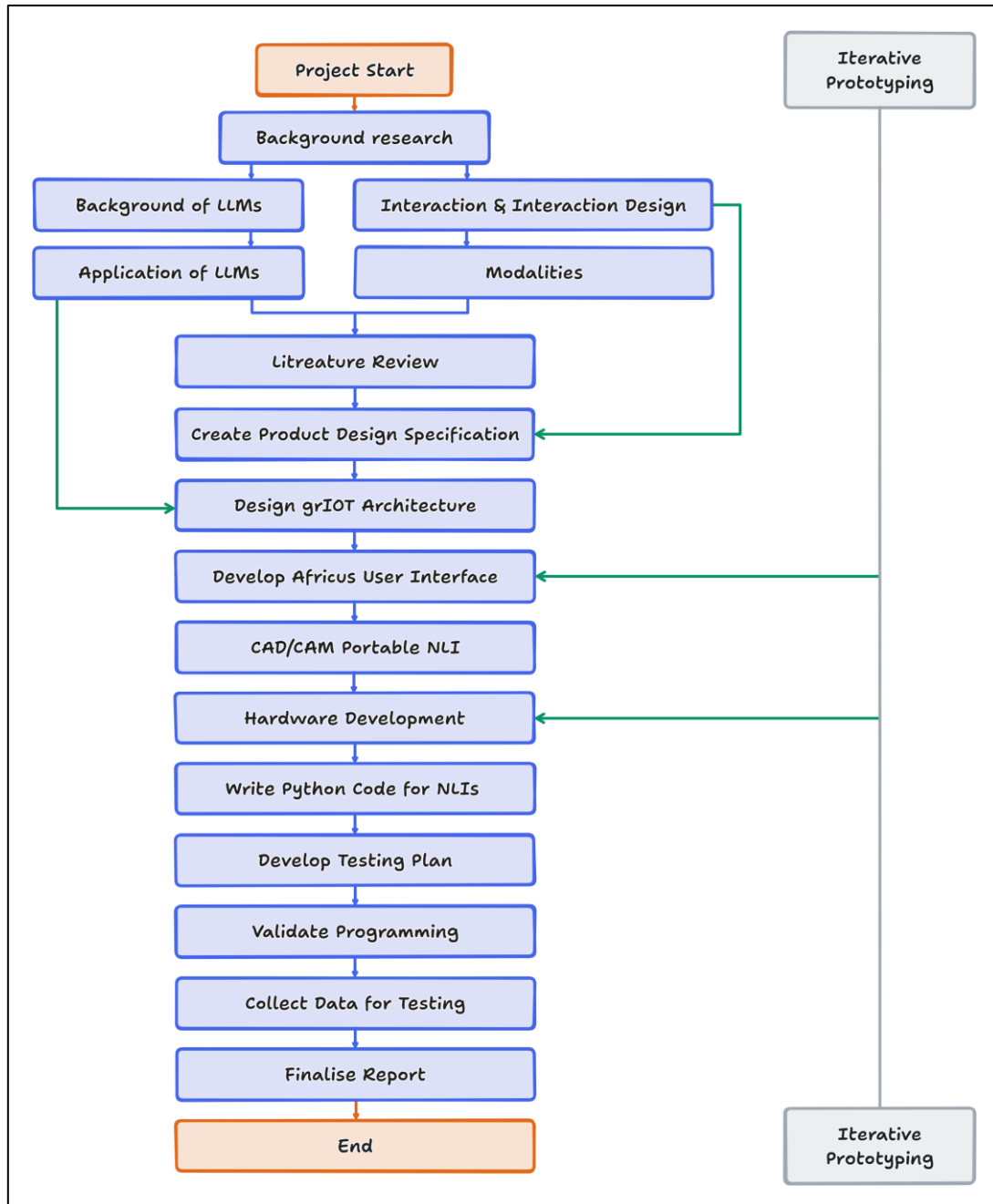


Figure 31 | Flow Chart

#### Notes:

- Green lines represent links between concepts.
- Iterative development discussed by (Schön, 1983) is reflected in the “Iterative Prototyping” line running alongside the main flow, indicating its ongoing impact on the project.







## 4.2 Risk Analysis

Likelihood	Harm Severity			
	Minor 1	Marginal 2	Critical 3	Catastrophic 4
<b>Certain</b> 5	#0001 <b>High</b> Build Plate Adhesion	#0002 <b>High</b> Damage to Mirror Film	#0003 <b>Very High</b> Battery Discharge	#0004 <b>Very High</b> Battery Damage
<b>Likely</b> 4	#0005 <b>Medium</b> Part Unavailability	#0006 <b>High</b> Filament Runout During Print	#0007 <b>High</b> CAD Crash	#0008 <b>Very High</b> Component Damage
<b>Possible</b> 3	#0009 <b>Low</b> Prototype Inadequate	#0010 <b>Medium</b> Electrical Connection Fault	#0011 <b>High</b> Poor Internet Access	#0012 <b>Very High</b> SD Card Corruption
<b>Unlikely</b> 2	#0013 <b>Low</b> Clogged Nozzle	#0014 <b>Medium</b> Filament Jam	#0015 <b>Medium</b> Printer Malfunction	#0016 <b>High</b> Dependency Errors
<b>Rare</b> 1	#0017 <b>Low</b> Illness	#0018 <b>Low</b> Power Outage	#0019 <b>Medium</b> Microcontroller Short Circuit	#0020 <b>Medium</b> Malicious AI Takeover

Figure 32 | Risk Matrix

A risk matrix was chosen as the method of analysis for this project. From this we derive a risk/severity product.

$$\text{Risk Severity Product} = \text{Likelihood} \times \text{Harm Severity}$$





Risk ID	Description	Risk-Severity Product	Mediation Strategy	Post Mitigation Adjustment	Justification
#0001	Build Plate Adhesion Failure	5	Glue Stick or Painters Tape	1	Increase adhesion on build plate
#0002	Damage to Mirror Film	10	Diligence in handling mirror, purchase of self-adhesive repair tape	5	Minimise damage by exercising care. In the event of damage, use mirror film tape to cover
#0003	Battery Discharge	15	Ensure all battery powered devices are charged. Keep charging cable near by	5	Battery powered devices will always lose charge, a cable protects against this
#0004	Battery Damage	20	Batteries to be stored safely	5	LiPo batteries can be dangerous, ensuring safety when handling them is paramount
#0005	Part Unavailability	4	Source from multiple vendors	1	Multiple options ensure redundancy
#0006	Filament Runout During Print	8	Ensure sufficient filament before printing	1	If filament is sufficient before printing it can't run out
#0007	CAD Crash	12	Ensure external programs aren't hogging utilities	5	Managing computer resources reduces risk of stress on GPU/CPU
#0008	Component Damage	16	Diligent handling of components	5	Accidental damage still possible
#0009	Prototype Inadequate	3	Fast iterations lead to quicker turn around	3	Project methodology facilitates this as a necessary risk
#0010	Electrical Connection Fault	6	Continuity Test after each solder joint	3	Testing with will reduce chance of unnoticed errors
#0011	Poor Internet Access	9	Use Phone as Hotspot	4	Internet access provided by phone
#0012	SD Card Corruption	12	Code is backed up to GitHub	1	Code can be pulled from source
#0013	Clogged Nozzle	2	Clean nozzle frequency	1	Reduces risk of filament build up
#0014	Filament Jam	4	Clean extruder regularly	1	Jam can be resolved quickly
#0015	Printer Malfunction	6	Monitor OctoPrint feed	3	Printer can be halted remotely
#0016	Dependency Errors	8	Ensure virtual environment is active	1	Using requirements file ensures versions are correct
#0017	Illness	1	Prioritise Health and Fitness	1	Prioritising Health minimises risk of illness
#0018	Power Outage	2	Files are backed up to OneDrive	1	Files will be saved in event of outage





#0019	Microcontroller Short Circuit	3	Diligence when handling ground and power pins	1	Careful handling of components ensures hardware safety
#0020	Malicious AI Takeover	4	Hope my affinity to AI gains it's favour	?	Isn't much anyone can do here to be honest

Table 6 | Risks and Mitigation Strategies

### 4.3 Tools and Resources

In addition to the publicly available API, publicly maintained modules and open-source code repositories were also utilised and will be declared where applicable.

Category	Tool	Use	Risk of Absence	Alternate Resources
Software	Fusion 360	Design of Encoder Wheel, Encoder Housing and Portable NLI Prototype	Prototypes cannot be developed	OnShape, AutoCAD
	Cura	Convert 3D models for 3D printing	Prototypes cannot be developed	PrusaSlicer, Slic3r
	OctoPrint	Monitoring 3D printer remotely, file upload	Printer would have to be monitored directly. Physical SD card upload	Duet, Repetier
	Blender	Generating graphic visualisation of prototypes	Lack of CAD graphics in report	Maya, C4D
	Adobe Illustrator	Graphic Design for GRiOT Logo and Portable NLI Icon	Lack of Logo for project	tldraw
	tldraw	Graphic Design for Report	Lack of diagrams and graphics in report	Illustrator
API	GPT-4	Language Model Backend	Unable to Implement features	Llama Model
	Anthropic Claude	Language Model Backend	Unable to Implement features	Llama Model
	ElevenLabs	Speech to Text Generation	No Audio for afriacus	Tortoise-TTS, ViTS
Hardware	Ender 3 3D Printer	Essential to fast iteration Manufacture of Prototypes	Slower iteration periods as external manufacturing methods used	University facilities
	Soldering Station	Creation of Electronic Hardware	Slower iteration cycle as external tools will be required	University facilities
	Raspberry Pi 3B+	Testing Hardware components	Unable to verify hardware in testing	None
	Elegoo Mega2560	Testing Hardware components	Unable to verify hardware in testing	None

Table 7 | Tools and Resources

[Appendix 7](#) presents a more detailed look at the resources used in the project. Including mathematical proof of the RSA encryption method used for connecting to the Raspberry Pi





## 5. Product Design Specification

The introduction and literature review present arguments and evidence for hardware with a single purpose. These devices aim to follow this principle. An interface for Generative AI.

“I totally believe that if you give people way more powerful tools, it’s not just they can work a little faster. They can do qualitatively different things” (Altman, 2024)

### 5.1 Natural Language Interface

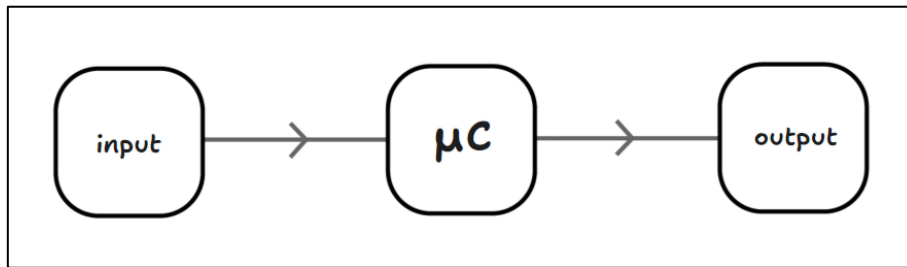


Figure 33 | Natural Language Interface Block Diagram

The simplest form of a natural language interface receives input from a user, processes this input, formulates a response using a generative model and returns a response via the output.

[Appendix 8](#) presents a flowchart for a generic natural language interface.

1. Input Modality
<b>Microphone</b>
Natural Language is obtained through a user’s speech. While text input can be considered as a modality of interaction this project’s focus is on speech as a higher throughput form of communication
Requires suitable bus protocol to interface with the processor. SPI, I2C, USB, etc. This is dependent on the I/O of the processor.
2. Processor
<b>Microcontroller</b>
A processor, namely a microcontroller is required to handle interactions between the peripheral input and output devices. A key priority is the ability to process audio such that it can:
<ul style="list-style-type: none"><li>• Concatenate the audio frames into a single file to transcribe locally.</li><li>• Concatenate the audio frames into a single file to send to an API.</li><li>• Stream audio frames to a WebSocket or another endpoint that can handle audio frames.<ul style="list-style-type: none"><li>○ Mainly applicable if the microcontroller lacks the local storage to save audio files generated by the user.</li><li>○</li></ul></li></ul>
Once this audio is handled an interaction with a language model will occur. This could occur locally on device if sufficient processing is available or (more likely) will take place using some API which to offload the processing from the device to rented compute or a private server.
The microcontroller will also need to interface with output and should have suitable I/O and high enough clock speed to handle processing audio and displaying text simultaneously.





3. Output Modality	
Visual	Aural
Visual output via the medium of a display is likely the most versatile method of interfacing with a generative model, as some models are multimodal producing both image and text, a display in this scenario would enhance the user experience.	Aural output via means of a speaker is an essential output modality for any natural language interface (NLI) that aims to communicate with users through speech. It outputs the auditory representation of the response generated by the system. The speaker plays back pre-recorded sounds, synthesized speech, or a combination of both, to provide an audible response to the user's query. It is necessary that the audio output is clear and easily understandable.

Table 8 | Natural Language Interface Components

### 5.1.1 Natural Language Interface Requirements

When developing requirements, it is important to reflect upon the Initial Aims and Objectives.

- Create hardware based on a Natural Language Interface that can be used to interact with a Generative AI model.

Reference	Requirement
NLI-001	Capture and Process Audio
NLI-002	Interface with Generative Model
NLI-003	Return Response to User
NLI-004	Make use of GRiOT to perform tasks
NLI-005	User Experience Focus

Table 9 | Natural Language Interface Requirements





## 6. Candidate Solutions

### 6.1 Natural Language Interface

#### africus

<b>Name</b>	africus
<b>Description</b>	Named after the Africus Monolith from 2001: A Space Odyssey, africus is a voice first Natural Language Interface in a smart mirror form factor.
<b>Proposed Block Diagram</b>	<pre> graph LR     Microphone --&gt; Microcontroller     Microcontroller --&gt; Display     Microcontroller --&gt; Speaker         </pre>

Table 10 | Africus Proposal

#### Display | Options Analysis

Candidate #1	Candidate #2	Candidate #3
M215HGE-L23 Twisted Nematic (TN) LCD panel	OLED Panel	Projector

Table 11 | Display Candidates (africus)

Criteria derived from relevant requirements:
<b>NLI-003 return response to user</b>
- <u>Effectiveness</u> [Ability to display response to user]
<b>NLI-005 user experience focus</b>
- <u>Availability</u> [User serves as the customer in this regard, availability for manufacture will negatively impact them]
- <u>Ease of Implementation</u> [Reduce development time leads to a more stable platform and fewer bugs experienced by user]
- <u>Cost</u> [In production increased component cost often passed down to consumer]

Table 12 | Derived Criteria (display)

Criteria	Weighting	Candidate #1	Candidate #2	Candidate #3
Effectiveness	4	1	1	1
Availability	5	1	0	0
Ease of Implementation	3	1	1	1
Cost	5	1	-1	0
	Totals	17	2	7

Appendix 9 presents the datasheet with information on viewing angle and dimensions.

<b>Weighting Justification</b>
As per the project methodology of fast iterative cycles, availability has been given the highest weighting as access to materials is paramount.
Followed by cost as in practice increased development costs pull resources away from other (often key) development areas.





<b>Candidate #1 [17]</b>
The TN panel passes all metrics with a focus on cost and availability. As per the weighting justification, the methodology demands access to components. The TN panel was salvaged from an old computer monitor, in keeping with the 7Rs of sustainability.
<b>Candidate #2 [2]</b>
An OLED panel may be arguably more effective in a smart mirror implementation as they achieve deeper black levels (Qin, et al., 2016). The reflectivity of the thin polyester aluminium coating that comprises the two-Way-Mirror reflects increases as the light behind it decreases. This allows for white text to appear transmissive whilst the black background becomes reflective. However, this deeper black level is undercut by the price as OLED panels are often sold as enthusiast hardware in gaming monitors and as such carry a higher price tag.
<b>Candidate #3 [7]</b>
The fidelity of a projector scales dramatically with price, such that the quality of such a projector within a “viable” price range may decrease user experience (Powerll, 2014). Powerful projectors often cost upward of £400 (epson, 2024).

Table 13 | Candidate Score Justification (display)

### Microcontroller | Options Analysis

<b>Candidate #1</b>	<b>Candidate #2</b>	<b>Candidate #3</b>
Raspberry Pi 4B+	NVIDIA Jetson Nano	NVIDIA Orion Nano

Table 14 | Microcontroller Candidates (africus)

Criteria derived from relevant requirements:	
<b>NLI-001 Capture and Process Audio</b>	
- <u>I/O Utility</u> [peripheral microphone should be able to be attached via USB or other bus protocol (SPI, I2C, etc)]	
<b>NLI-005 User Experience Focus</b>	
- <u>Processing Speed</u> [Latency and thereby user experience rests on processing speed]	
- <u>Cost</u> [In production increased component cost often passed down to consumer]	

<b>Criteria</b>	<b>Weighting</b>	<b>Candidate #1</b>	<b>Candidate #2</b>	<b>Candidate #3</b>
I/O Utility	5	1	0	0
Processing	4	1	1	1
Cost	3	1	0	-1
	Totals	12	4	1

<b>Justification</b>
The NVIDIA Jetson and Orion Nano both run Jetpack a custom Linux kernel (NVIDIA, 2024), customizations of Linux often lead to issues with peripherals due to plug and play drivers being unsupported. This was verified in testing as a USB camera and Mic weren’t detected on the Jetson Nano. The Jetson Nano has a clock speed of 1.43GHz compared to the Pi’s 1.8GHz. <a href="#">Appendix 10</a> presents testing using the top command to display performance statistics. This revealed that both devices run the chromium-browser (UI) and python scripts (UX) handling the mirror functionality at a negligible difference of 8% in relative performance. The Raspberry Pi was chosen as a low-cost alternative (mean average per <a href="#">Appendix 10</a> is £51.53) with more community support in the event errors occurred in development.

Table 15 | Justification (Microcontroller)



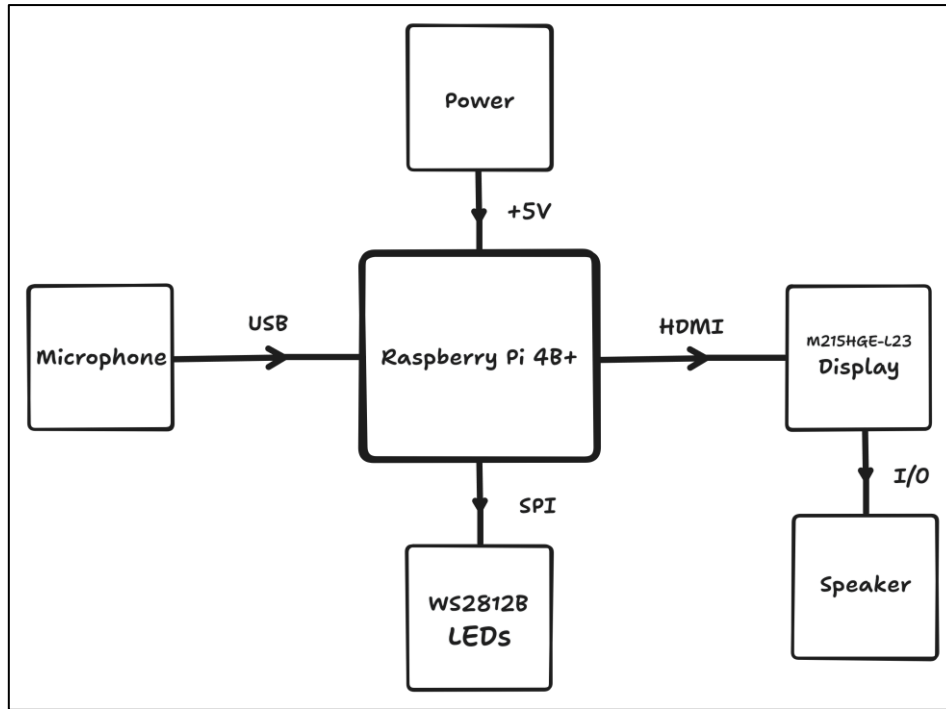


Figure 34 | africus Block Diagram

Item	Description
USB Microphone	A generic USB microphone was chosen.
Raspberry Pi 4B+	Handles the python scripts
WS2812B Addressable LEDs	Part of visual feedback loop.
M215HGE-L23 Display	Display for the interface, features speakers attached to the mainboard
5V Power Brick	USB-C Power Supply. Supplying 5V 2A as per the Raspberry Pi Specification (see Appendix 11)

Table 16 | Bill of Materials (africus)

Reference	Meets Requirement
NLI-001	As the Raspberry Pi features USB I/O a generic plug and play USB microphone meets requirements as it can transduce audio. The Raspberry Pi handles Analog-To-Digital Conversion of the incoming spoken audio.
NLI-002	The Raspberry Pi Runs Raspbian, a distribution of Linux. Enabling to easy execution of python code for interfacing with Generative models via python libraries and API.
NLI-003	The M215HGE-L23 panel is a 1920x1080 display
NLI-004	Python code can be run to access a generative reasoning backend
NLI-005	LEDs allows for the implementation of a visual feedback loop which directly impacts UX. Change in colour and animation allow the user to easily evaluate the state of the system (Norman & Draper, 1986). [rainbow] = idle [blue breathing] = voice activity detected

Table 17 | Justification v Requirements (africus)







[Hardware for the following sections practically tested as per project methodology]

pageR

<b>Name</b>	pageR
<b>Description</b>	A handheld natural language interface.
<b>Proposed Block Diagram</b>	<pre> graph LR     Microphone --&gt; Microcontroller     Tactile Input --&gt; Microcontroller     Microcontroller --&gt; Display         </pre>

Microphone | Options Analysis

Candidate #1	Candidate #2
MSM261S4030H0 MEMS Microphone	SPH0645 MEMS Microphone

Table 18 | Microphone Candidates (pageR)

<b>Justification</b>
The literature review presented the successful implementation of Micro Electro-Mechanical Systems (MEMS) microphone arrays within consumer smart speakers.

<b>Criteria derived from relevant requirements:</b>
<p><b>NLI-001 capture and process audio</b></p> <ul style="list-style-type: none"> <li>- <u>Functionality</u> [Component ability to convert pressure waves into digital signal is key to this requirement]</li> </ul> <p><b>NLI-005 user experience focus</b></p> <ul style="list-style-type: none"> <li>- <u>Quality</u> [SNR effects the overall quality of the microphone. Whilst ASR models like Whisper are good at deciphering speech within noisy audio the user should still be able to understand it should they wish to use it in another application.]</li> </ul>

Table 19 | Derived Criteria (pageR)

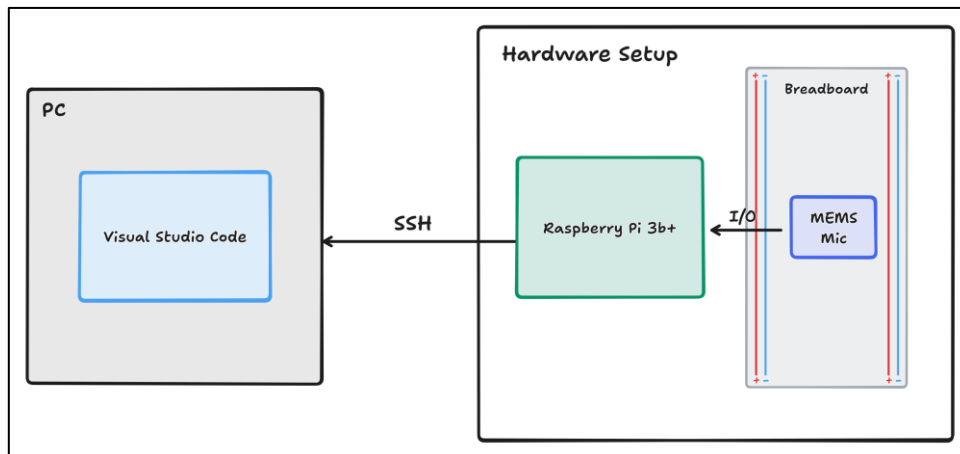


Figure 35 | MEMS Microphone Test Setup





MSM261S4030H0		
I/O	Pin Out	Proposed Pin Out
V	3.3V	3.3V
G	GND	GND
WS	GND (left)	SPI MISO
LR	GND (left)	GND
CK	BCLK (BCM 18)	SPI CE00
DA	DOUT (BCM 20)	SPI MOSI

Table 20 | MSM261S4030H0 Pin Out

SPH0645			
I/O	I/O Definition	Pin Out	Pin Definition
V	Power	3.3V	Power
G	Ground	GND	Ground
BCLK [Bit Clock]	PCM_Clock Provides timing for serial data bits being transmitted	GPIO18	<b>SPI1 CE0</b>
DOUT [Data Out]	Serial Data Output to input of uC	GPIO20	<b>SPI1 MOSI</b>
LRCL [Left Right Clock]	Word Select/Frame Sync differentiates between left (high) and right (low) channels	GPIO19	<b>SPI1 MISO</b>
SEL [Select]	Output Mode, Ground (I2S) Floating (Left Justified)	GND	Ground

Table 21 | SPH0645 Pin Out

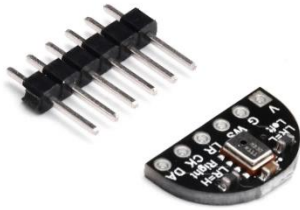
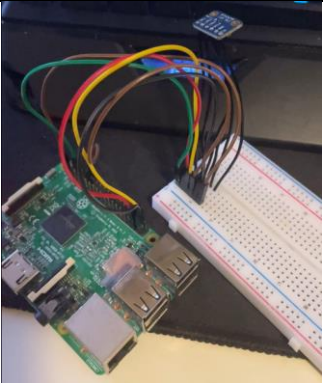
Candidate #1	Candidate #2
	
<p>The MSM261S4030H0 MEMS Microphone has very limited documentation with multiple different board variations wherein the silk-screened legend suggested pins had different definitions than certain data sheets. The above tables include information from my testing where I attempted to find “proposed” pinouts that would work. Unfortunately, due to the unreliability of the documentation this was unsuccessful.</p>	<p>SPH0645 MEMS Microphone was tested on the test setup presented in Figure 33. <a href="#">Appendix 12 001</a> presents video of the testing, demonstrating movement of the audio visualiser in the VS Code Command Line Interface. The ‘arecord’ (Linux, 2002) command was used to capture audio that was then played back to verify its functionality.</p>

Table 22 | MEMS Testing

Selection Justification
<p><a href="#">Appendix 12 001</a> demonstrates the SPH0645 MEMS Microphone satisfies <b>NLI-001</b> by interfacing with the Raspberry Pi and capturing audio. <a href="#">Appendix 13</a> presents a spectrogram with clearly visible formants that correspond to vocal frequency delimiting speech <b>NLI-005</b>.</p>

Table 23 | Selection Justification MEMS Microphones





## Tactile Input | Options Analysis

Candidate #1	Candidate #2	Additional Candidates
Custom Encoder Setup built on a TTC encoder (TTC, 2021)	Three-Way-Navigation Switch (Sparkfun, 2024)	Appendix 14 presents additional candidates that were considered.

Table 24 | Tactile Input Candidates (pageR)

Justification
As per the introduction and literature review, we are seeing an increased consumer appetite for meaningful tactile interaction with their devices. As such the tactile input for pageR is something of great importance to the project. It aims to satisfy <b>NLI-005</b> by providing a tactile user experience, serving as a push-to-talk means of interaction with the generative model ( <b>NLI-002</b> ) as well as a general input for navigation on the device. Appendix 16 presents a preliminary product render created to guide the design direction of pageR.

Table 25 | Justification for method

Criteria derived from relevant requirements:
<p><b>NLI-001 Capture and Process Audio</b></p> <ul style="list-style-type: none"> <li>- <u>Push-To-Talk</u> [input must include monetary contact switch to serve as a push to talk method of capturing audio]</li> </ul> <p><b>NLI-003 Return Response to User</b></p> <ul style="list-style-type: none"> <li>- <u>Navigation</u> [should enable some form of navigation on device to aid user in viewing response]</li> </ul> <p><b>NLI-005 user experience focus</b></p> <ul style="list-style-type: none"> <li>- <u>Tactile</u> [The tactile nature of this input modality embodies the research conducted previously, adding tactile interaction improves user experience]</li> </ul>

Table 26 | Derived Criteria (Tactile Input)



Candidate #1	Candidate #2
	
<p>Mouse wheels use rotary encoders with quadrature encoding. Preliminary research discovered that these encoders can be acquired online. A solution was iterated upon to house an encoder with a momentary switch [discussed in further detail in 7.3.1]</p>	<p>Appendix 16 presents Anthropometric Data that was used during the development of pageR. From this it was determined that the Three-Way-Navigation Switch was too small to be effective as a tactile input in a UX focused implementation (<b>NLI-005</b>).</p>

Figure 36 | Candidates for Tactile Input

Selection Justification
<p>The custom implementation was deemed the best solution.</p> <p><b>NLI-001</b> – The implementation of a momentary switch will serve as the push to talk input.</p> <p><b>NLI-002</b> – The mouse wheel encoder will enable navigation like that seen on a regular mouse.</p> <p><b>NLI - 003</b> – Encoder wheel will be developed in the next stage to facilitate tactile interaction.</p> <p>This solution implements “two-dimensional” scrolling input as presented by (Ward, 2001)</p>

Table 27 | Justification (Tactile Input)





**Processor | Options Analysis**

Selection Justification
<p>3.3.2 presents two primary examples of the Raspberry Pi Zero’s abilities in the field signal processing, fulfilling <b>NLI-001</b>  Options Analysis for the africus microprocessor presents the benefits of the Raspberry Pi platform for criteria <b>NLI-001</b> and <b>NLI-005</b></p>

Table 28 | Processor Justification (pageR)

**Display | Options Analysis**

Candidate #1	Candidate #2
128x64 OLED Display	1.69 ST7789V3 TFT display

Criteria derived from relevant requirements:
<p><b>NLI-003 Return Response to User</b></p> <ul style="list-style-type: none"> <li>- <u>Display</u> [The solution should functionally display content to the user]</li> </ul> <p><b>NLI-005 User Experience Focus</b></p> <ul style="list-style-type: none"> <li>- <u>Aesthetic</u> [The solution should display content in an aesthetically pleasing manner]</li> </ul>

Table 29 | Derived Criteria (pageR display)

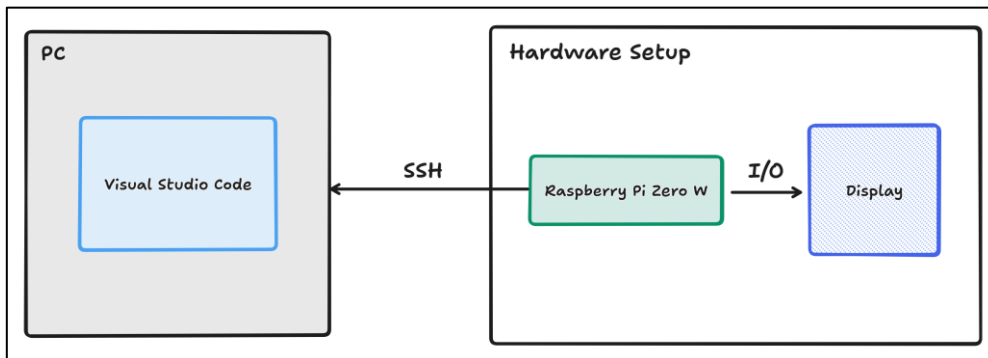


Figure 37 | Testing Setup (pageR Display)

Candidate #1	Candidate #2
<p>Appendix 12 023 presents testing of the 128x64 OLED display. Connected to a Raspberry Pi Zero board to display example text.</p>	<p>Appendix 12 013 presents testing of the 1.69 ST7789V3 TFT display. Featuring the display of an animated GIF.</p>

Table 30 | Candidates for Display (pageR)





**Selection Justification**

**NLI-003** - Both function display content as per [Appendix 12 023](#) and [Appendix 12 013](#)  
**NLI-005** - Candidate #1 features a 128x64 monochromatic dot matrix display with I2C connection creating graphics by changing the state of display blue pixels. This shown to effectively display the text “hello world” but lacks the ability to display more than basic raster graphics. Candidate #2 features a 240x280 262K colour display, the faster SPI protocol enabling animated GIF playback. Appendix 15 displays equations for Dots Per Inch (DPI). Candidate #2 presents a significantly higher DPI than Candidate #1 which can be seen from the testing videos.  
 Candidate #2 was ultimately chosen.

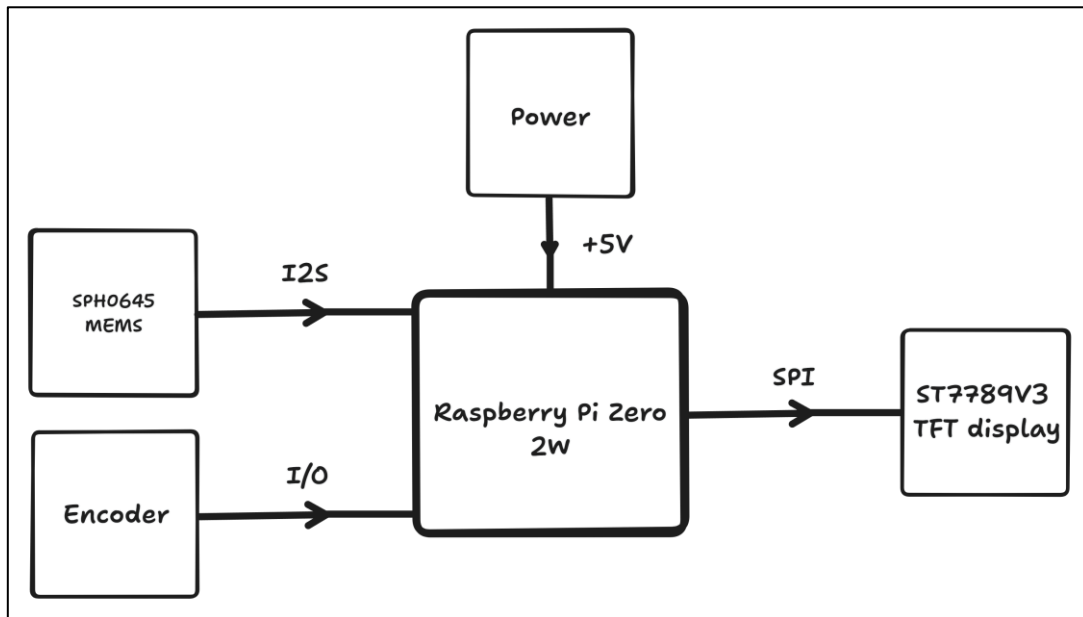


Table 31 | pageR Block Diagram

Item	Description
SPH0645	Audio Input
Encoder	Tactile Input
Raspberry Pi Zero 2W	Processing
ST7789V3	Display
Power	5V Power

Table 32 | Bill of Materials (africus)

Reference	Meets Requirement
NLI-001	SPH0645 MEMS captures audio input, Raspberry Pi Zero processes audio
NLI-002	Raspberry Pi Zero runs python code to interface with Generative Model API
NLI-003	ST7789V3 displays response, encoder aids in device navigation
NLI-004	Raspberry Pi Zero uses python script to connect to GRiOT endpoint
NLI-005	Encoder serves as a tactile input method as per literature review research

Table 33 | pageR Requirements





## Natural Language Human Input Device

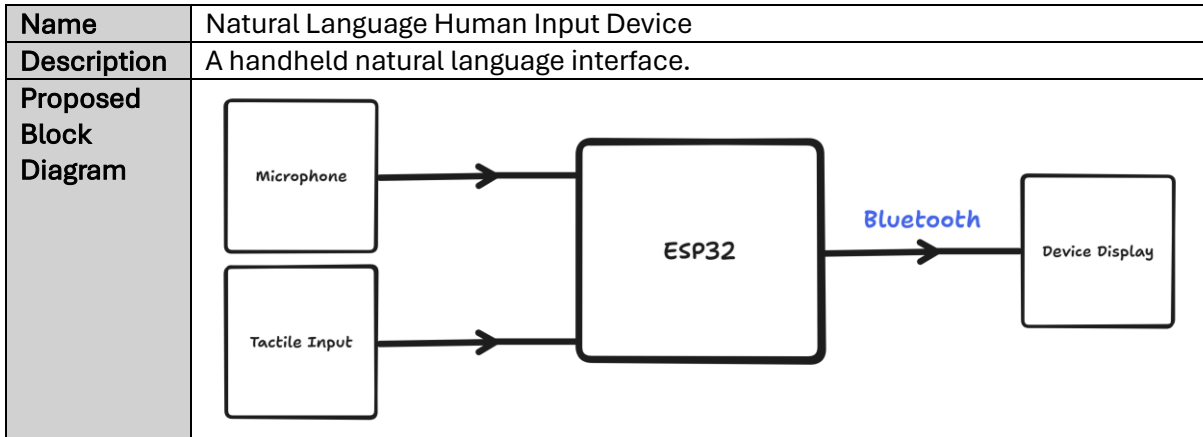


Table 34 | Natural Language Human Input Device



Figure 38 | Language Model Logic

<b>Justification</b>
<p>Based on research presented in <a href="#">3.3.3</a> this natural language interface presents a speech to text method of interaction with a Bluetooth enabled device.</p> <p>The previous figure presents an example where the device could execute code on a windows machine to perform tasks by parsing JSON code. {key:value} pairs of commands are generated by the Language Model, the “command” is parsed by the Arduino code to execute “windows + r” opening the windows run menu to enter the command using the BLE Keyboard library.</p>

Table 35 | Justification Natural Language Human Input Device



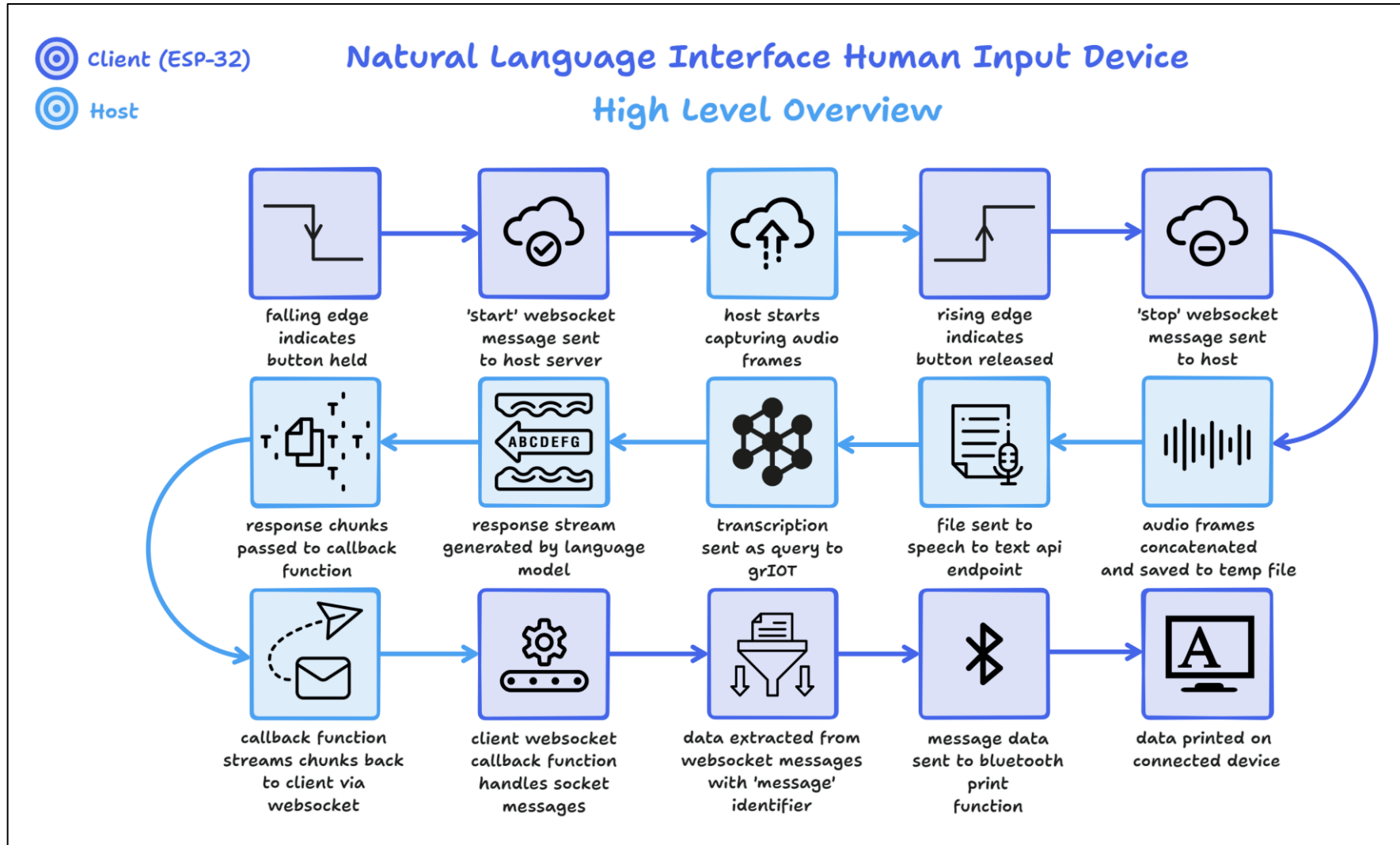


Figure 39 | Natural Language Interface Human Input Device (High Level Overview)





## 6.2 Generative Reasoning

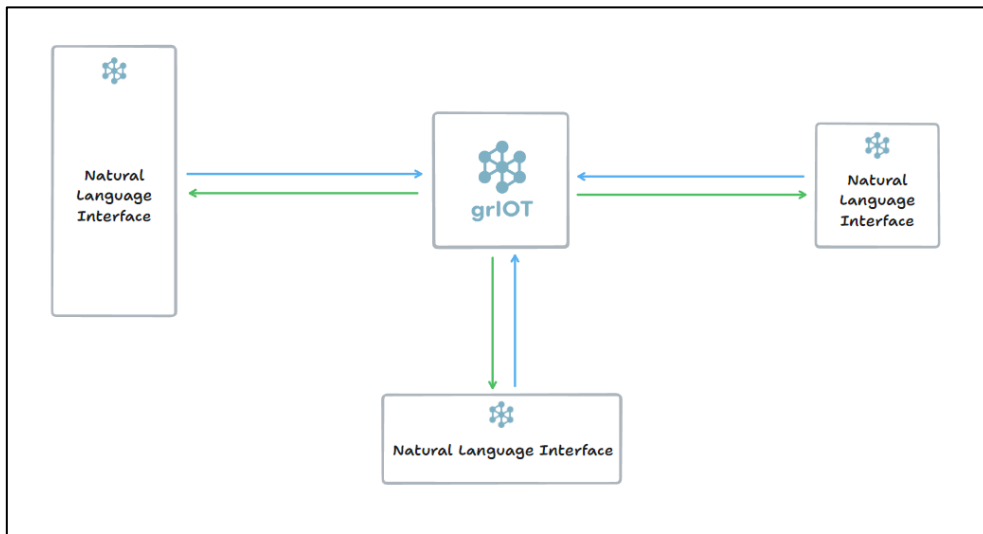


Figure 40 | Proposed GRiOT System Interconnectivity

Soegaard (2023) comments on the importance of gaining incite on the perception of your design, as such [Appendix 17](#) presents a consumer survey on the design direction of the project's logo.







## 7. Product Design Development

This section covers the design and development of GRiOT and the Natural Language Interfaces

### 7.1 GRiOT

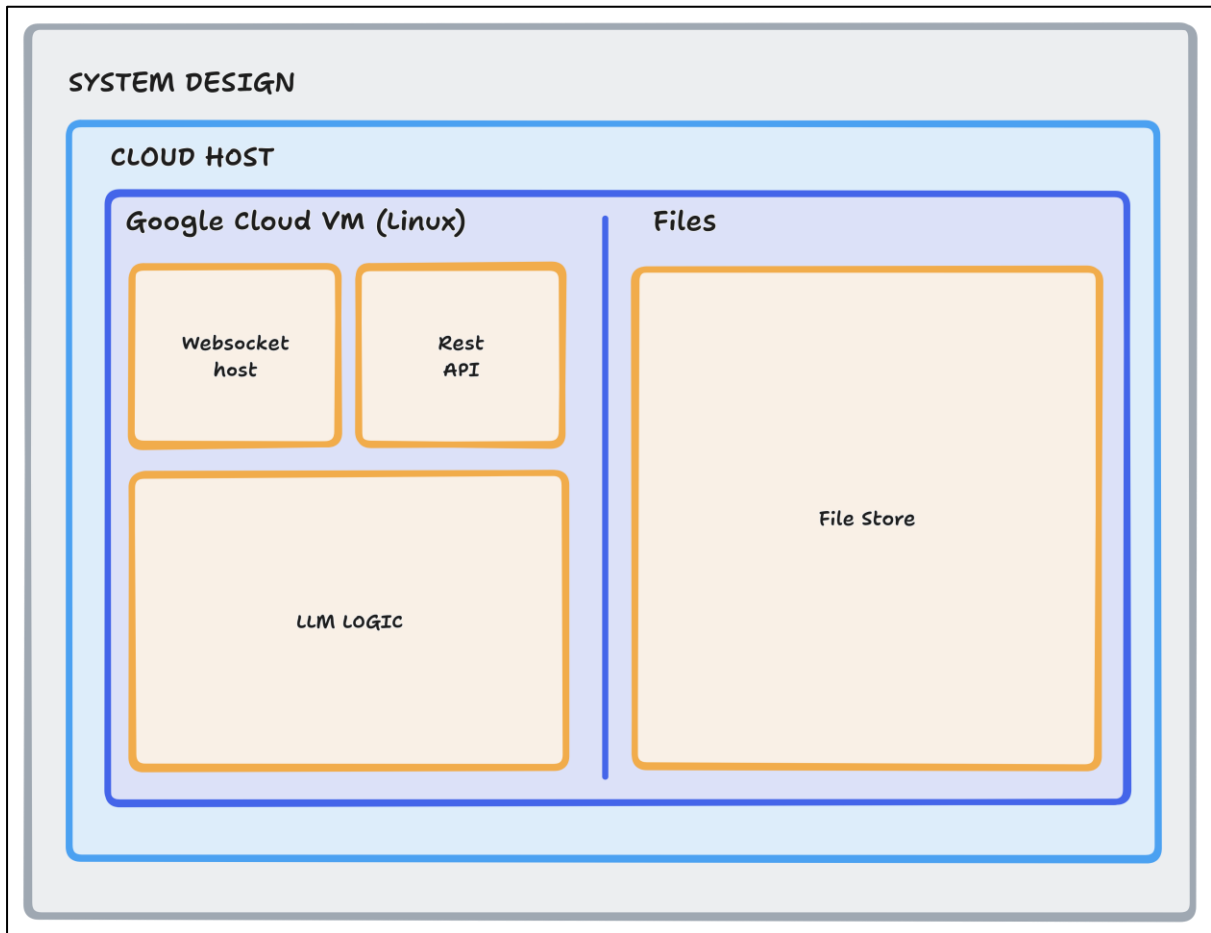


Figure 41 | GRiOT System Design





## 7.2 Africus

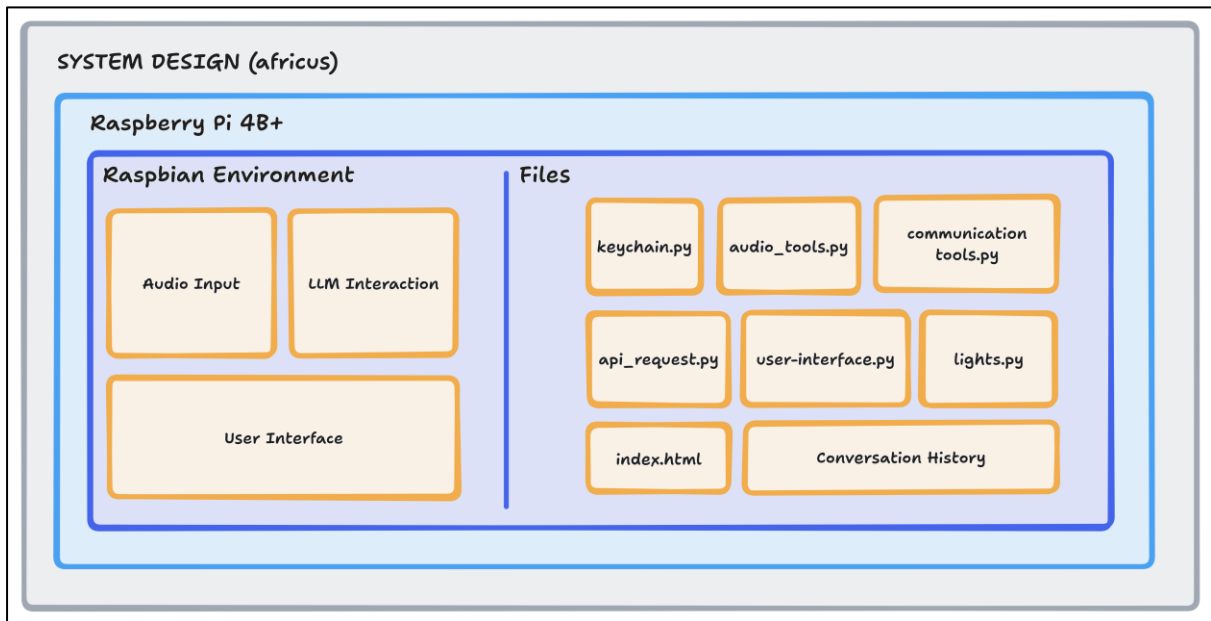


Figure 42 | System Design (africus)

File	Description
keychain.py	Contains API Keys
audio_tools.py	Custom module to facilitate voice activity detection, transcription, and speech synthesis
communication_tools.py	Custom module to handle adding system prompt to conversation, managing conversation history, adding telemetry data, and making API call to GPT-4
user-interface.py	Hosts flask server on localhost:5000, manages internal socket messages, opens chromium-browser to display user interface
api-request.py	Connects to flask server as client, imports communication_tools.py as a module
lights.py	Connect to flask server as client, awaits socket messages for voice_activity_detected=True voice_activity_detected=False
Index.html	HTML, CSS and JS user interface
Conversation History	Conversation is stored as a text file or non-persistent JSON dictionary

Table 36 | africus files





### 7.3 pageR

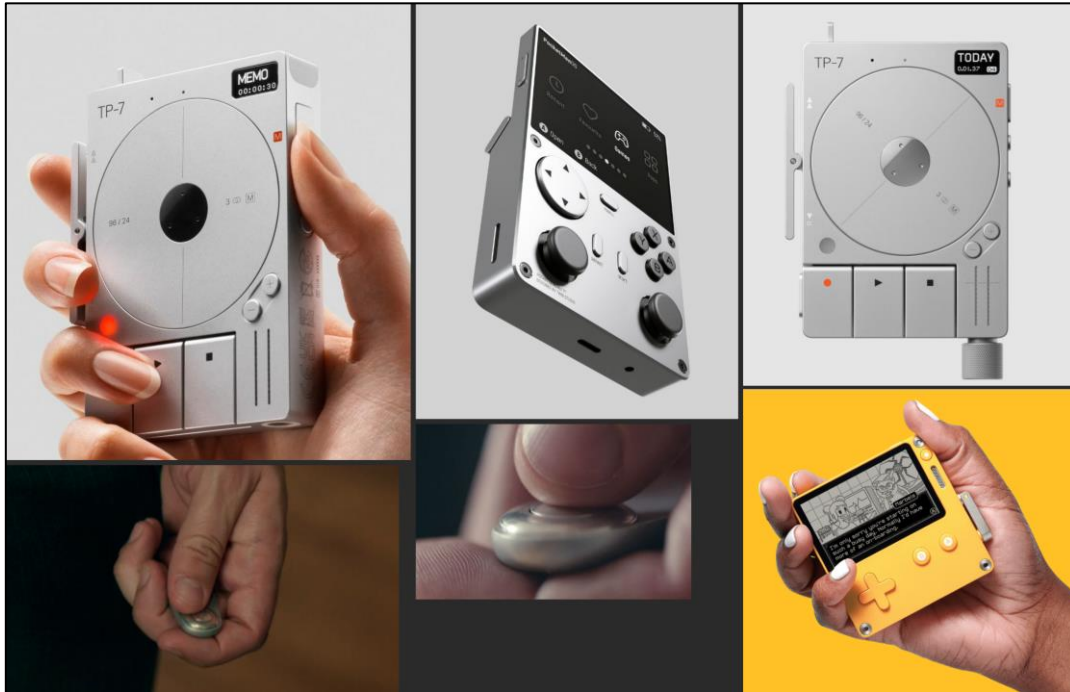


Figure 43 | Design Inspiration [1 to 3 Top Row | 4 to 6 Bottom Row]

#### Design Inspiration

(Endrissat, et al., 2016) discuss the benefits and potential of moodboarding for enabling creative freedom, commenting it “establishes a point of reference” for a design.

As such I created a collage from which I aimed to draw inspiration for the form factor and design of the portable Natural Language Interface, pageR.

Teenage Engineering present the TP-7 Tape Recorder [1,3] and the Playmate [6]. Initially inspired by the industrial design aesthetic which borrowed from the German designer Dieter Rams [seen in Iteration 3]

The Black Mirror episode “The Entire History of You” presents [4,5] a handheld device used by the characters in the episode to control their “grain” neural interface.

Table 37 | pageR Design inspiration

Appendix 16 presents an initial render used to guide the development of this part of the project








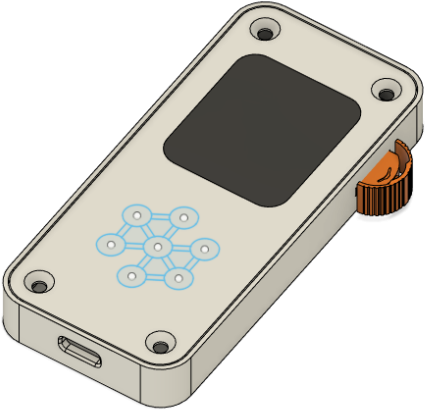

Iteration 1	Analysis
	<p>This initial iteration focused on sizing. Fusion 360s internal measurement tools in addition to the Raspberry Pi Zero datasheet were used.</p> <p><a href="#">Appendix 16</a> features Anthropometrics data that was considered when sizing.</p> <p><a href="#">Appendix 12 021</a> features a visualisation of the process of developing this CAD model.</p>
Iteration 2	Analysis
	<ul style="list-style-type: none"> <li>• Screen cut out implimented based on dimentions from <a href="#">Appendix 15</a></li> <li>• Height increased by 5mm to accommodate raspberry pi and display.</li> </ul> 
Iteration 3	Analysis
	<ul style="list-style-type: none"> <li>• Industrial design aesthetic</li> <li>• See <a href="#">Appendix 22</a> for comments on the calibration of the printer.</li> <li>• Three Way Switch was deemed too small for use as tactile input.</li> </ul> 

Table 38 | Iteration 1-3






Iteration 4	Analysis
	<ul style="list-style-type: none"><li>• Iteration developed as looks like model</li><li>• Moved away from Industrial Design Aesthetic</li><li>• Focus on curves to match the implementation of the circular encoder wheel</li></ul>

Table 39 | Iteration 4



Figure 44 | pageR 3D Printed Prototype Model





### 7.3.1 Encoder

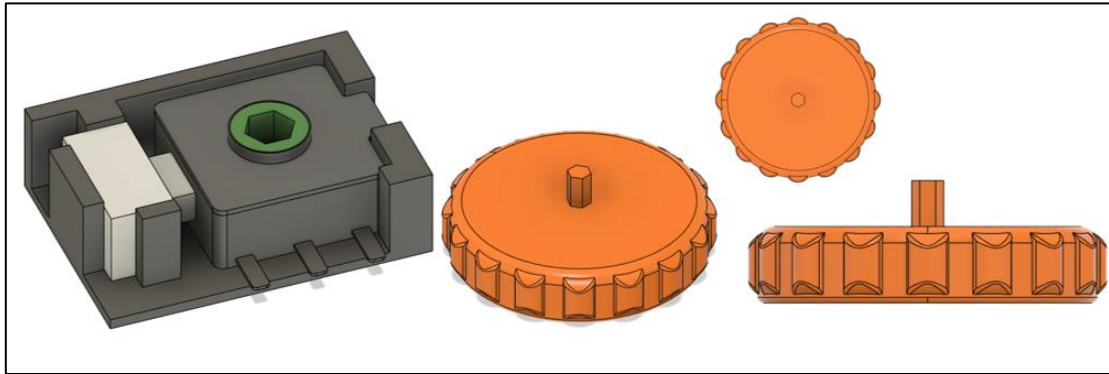


Figure 45 | Encoder Assembly and Wheel CAD

### Encoder Wheel

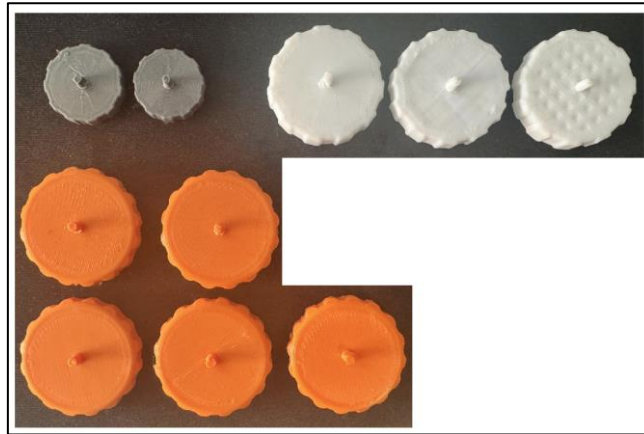


Figure 46 | Encoder Wheel Development

### Encoder Assembly



008

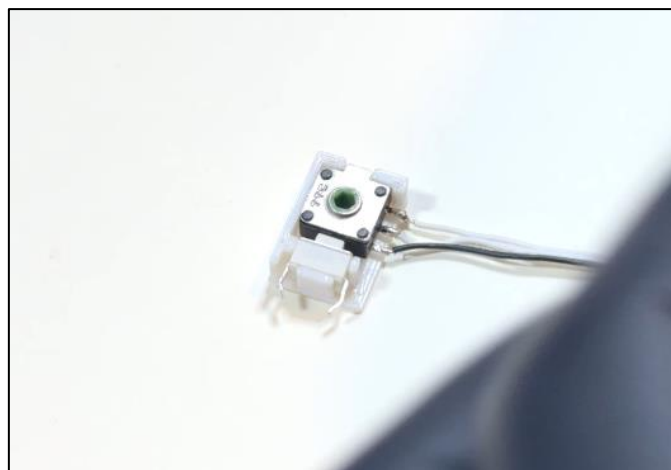


Figure 47 | Encoder Assembly

[Appendix 12 08](#) presents a visual instruction on how to construct the encoder.

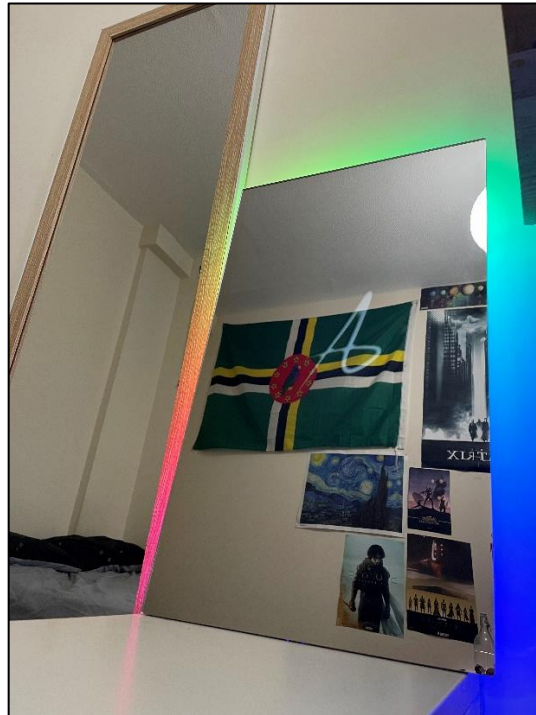
[Appendix 20](#) presents a detailed look at the encoder.





## 8. Implementation and Testing

### 8.1 africus



024

Figure 48 | africus

To effectively appreciate the animated aspect and use in context [Appendix 12 024](#) presents a video example of this process flow.

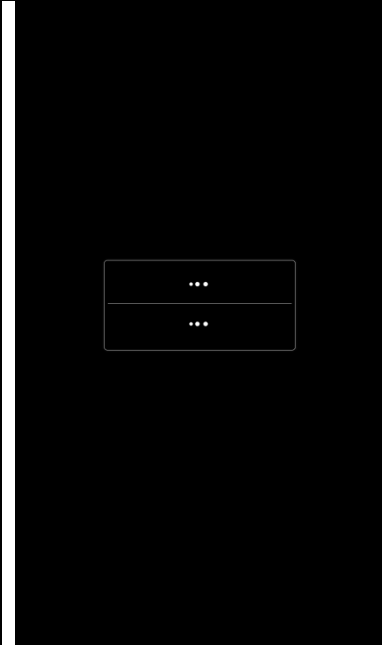


		
at idle	User response captured	User: Hello Response: Greetings. How may I assist you today?

Table 40 | africus interaction loop



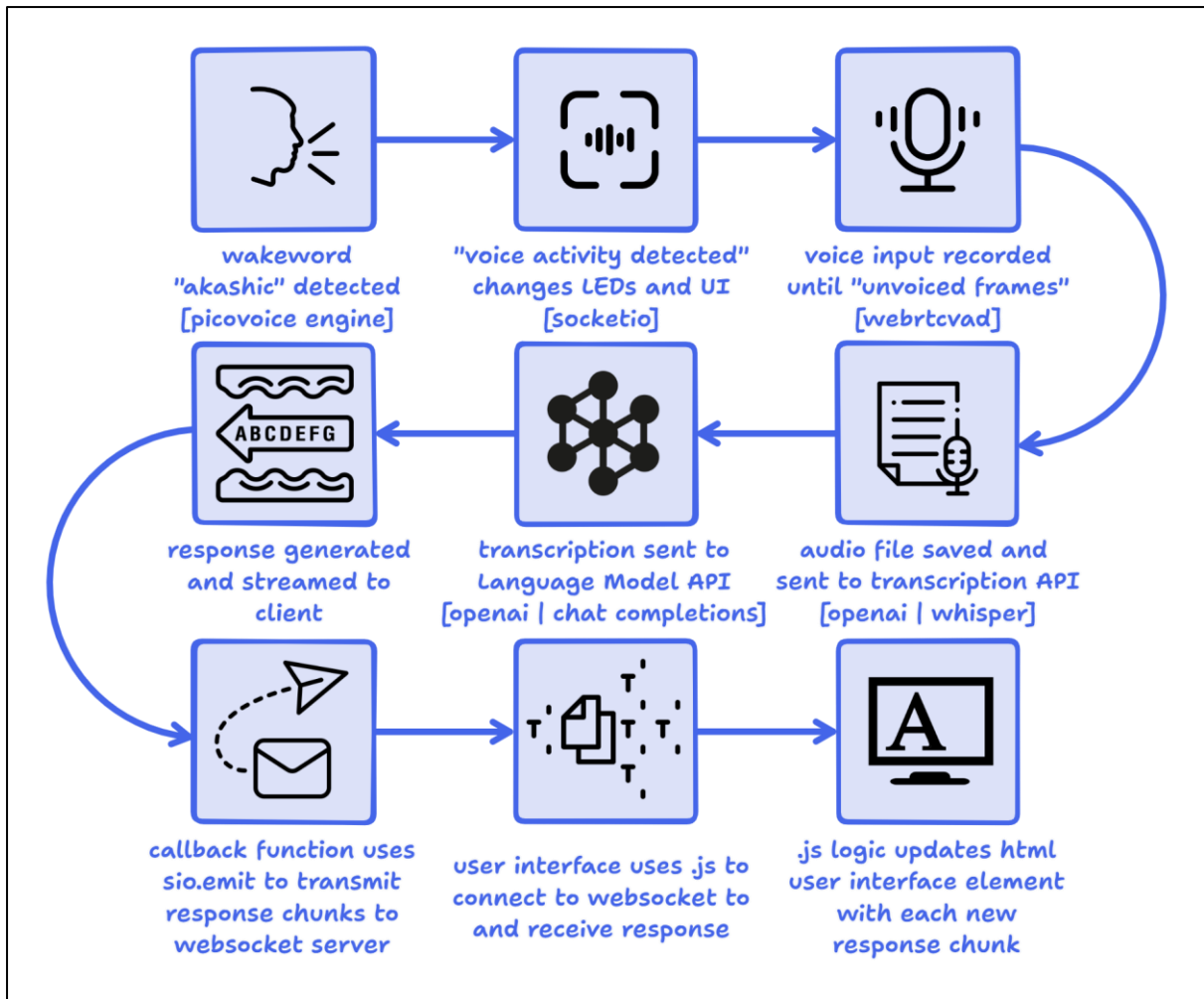


Figure 49 | Africus Process Flow







## 8.2 pageR

Appendix 12 016 presents a video example of this process flow.

016

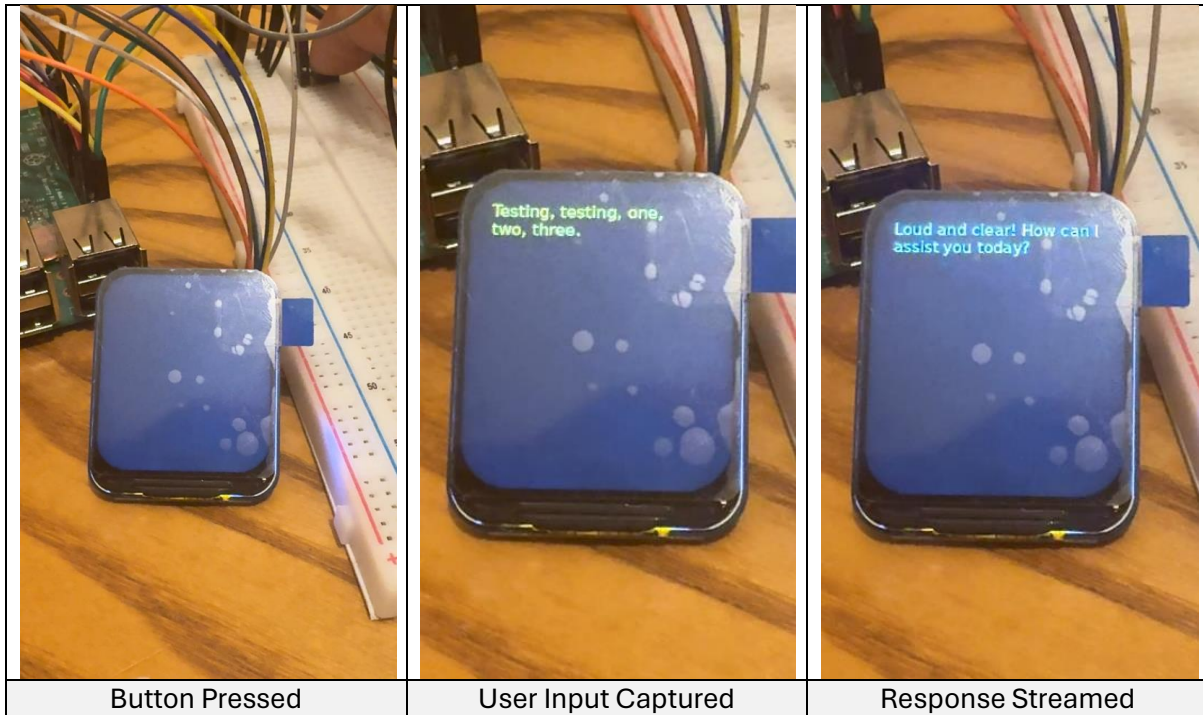


Figure 50 | pageR process flow

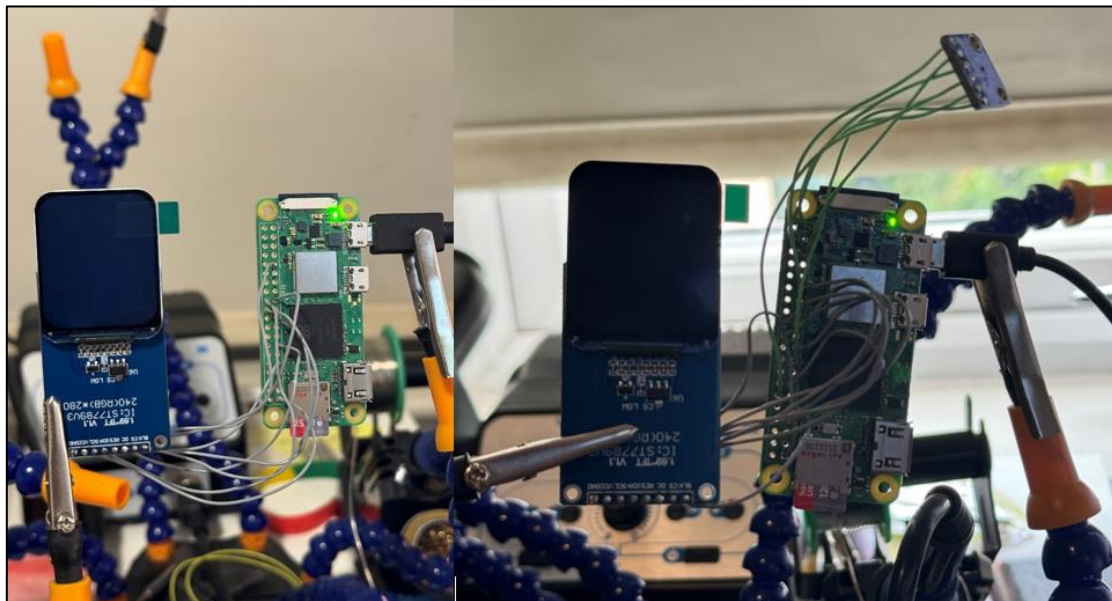


Figure 51 | Raspberry Pi Zero W with SPH0645 MEMS and ST7789V3 TFT display.

Appendix 12 025 presents video of the pageR minimum viable product stage

Appendix 21 features the pinout schematic for this set up.



025



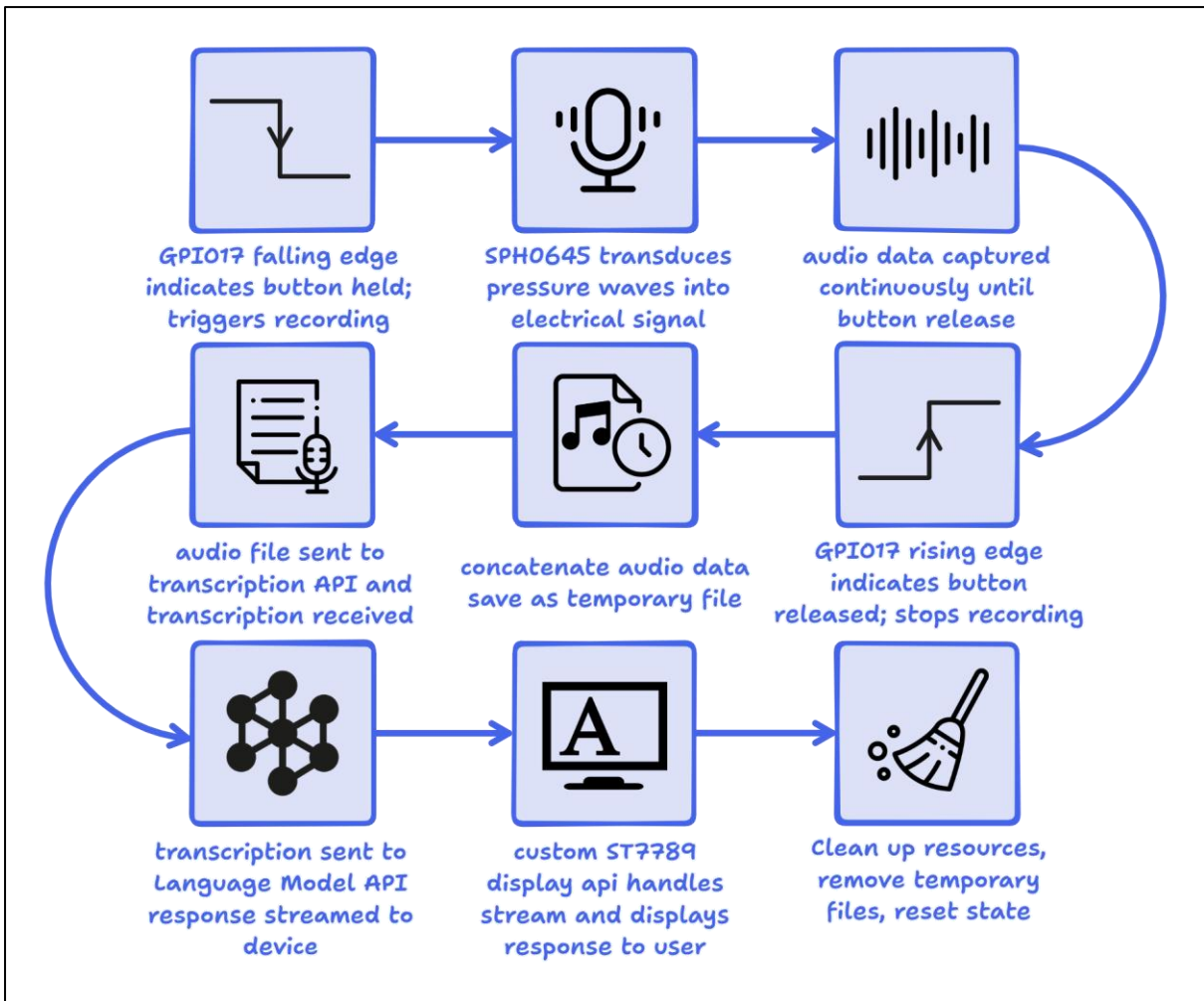


Figure 52 | pageR Process Flow





## 8.2.1 Encoder

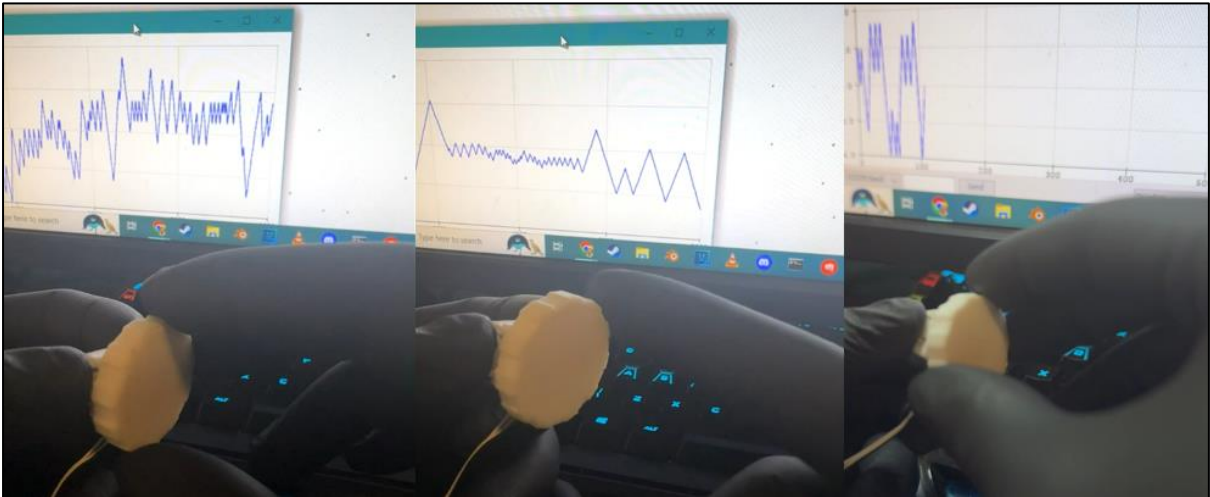


Figure 53 | Encoder Testing in Arduino Serial Console

[Appendix 12 005-007](#) provide video of this testing process.



005

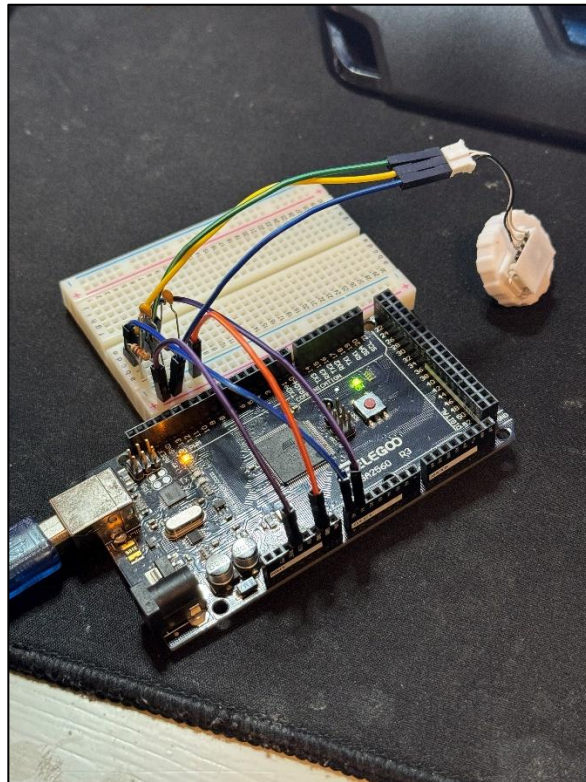


Figure 54 | Encoder Test Set-up Arduino

[Appendix 20](#) features a detailed look at the development of the encoder, application of signal processing, and development of a low-pass-filter.



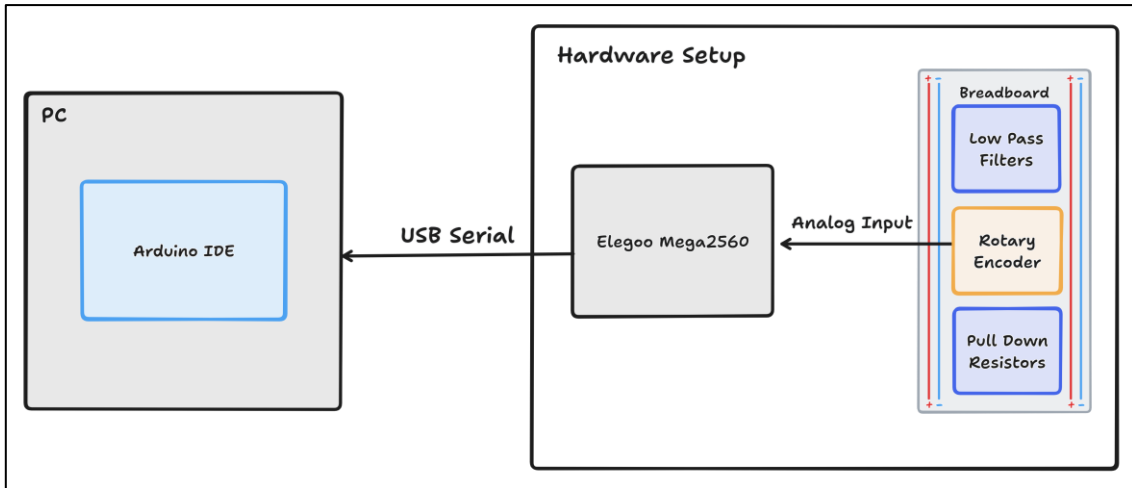


Figure 55 | Hardware Setup for Rotary Encoder Test





## 9. Conclusion and Recommendations

### **Conclusion**

This dissertation explored the evolving landscape of human-computer interaction (HCI) through the lens of generative reasoning and natural language interfaces (NLIs). The research examined the potential of large language models (LLMs) to bridge the gap between human intent and action, offering a more seamless and personalised user experience.

Through the development of two NLI prototypes, *africus* and *pageR*, this work demonstrated the feasibility of integrating LLMs into tangible devices. These prototypes, while requiring further refinement, serve as proof-of-concept for the future of HCI, highlighting the potential for more intuitive and efficient interactions with technology.

### **Overall Conduct**

[Appendix 18](#) presents a highly detailed self-critical evaluation of each of the 15 subsections of the project as presented by the Gantt Chart. Below is a summary after each of the 3 stages. For the full detail please refer to the appendix.

Stage	Suggestion derived from evaluations of each subsection
1	Define a stricter project scope with more explicit objectives to ensure effort is directed correctly and appropriately. This will ensure that the project progress whilst sufficiently documented, can be effectively reflected in a report
2	Sufficiently placed deadlines should be set at a point to stop research. Whilst interesting and potentially beneficial for other aspects of academic study priority needs to be placed on documentation of the task.
3	As the main issue lies within the documentation of the project and not the creation of hardware a solution may lie in the management. A sprint structure could be adopted, where bi-weekly periods are explicitly taken to spend diligently documenting the work completed in the weeks prior – then two more weeks of work can commence. Separating these aspects would allow for a more structured workflow, potentially increasing output as there is no worry of information being overlooked.

Table 41 | Conduct Evaluation





## Improvements in Evaluation

### System Usability Scale (SUS)

Much of the testing involved analysis from a technical perspective. However, usability from a user perspective is important.

#### System Usability Scale

		1	2	3	4	5
1	I think that I would like to use this system frequently	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	I found the system unnecessarily complex	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	I thought the system was easy to use	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	I think that I would need the support of a technical person to be able to use this system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	I found various functions in this system were well integrated	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	I thought there was too much inconsistency in this system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	I would imagine that most people would learn to use this system very quickly	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	I found the system very awkward to use	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	I felt very confident using the system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	I need to learn a lot of things before I could get going with this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Table 42 | System Usability Scale (SUS)

#### Evaluation

The System Usability Scale (SUS), initially introduced by John Brooke in 1996, is a concise and effective tool for assessing the usability of various systems, It consists of 10 items that users rate on a Likert scale, providing insights into the subjective assessment of usability (Brooke, 1996). Lewis (2017) presents research on the effect of dropping items from the SUS suggesting that some criteria may be less relevant in some cases. However Brooke (1996) himself suggests

*‘Usability of an artefact is defined by the context in which that artefact is used’.*

Building on this I propose modifying the SUS by adding an additional section where the user may discuss in natural language the context of their interaction. Prompting them to provide specific feedback at their chosen level of detail.

Appendix 19 presents a modified version of the System Usability Scale and a completed example.





## References

2001: *A Space Odyssey*. 1968. [Film] Directed by Stanley Kubrick. s.l.: Metro Goldwyn Mayer.

4Layer Technologies, 2024. *JOYSTICK BREAKOUT BOARD (PACK OF 2)*. [Online]  
Available at: <https://4layertech.com/products/joystick-breakout-board-pack-of-2>  
[Accessed 12 February 2024].

Abrahamsson, S. & Eriksson, E., 2010. *En utvärdering av synthesizern OP-1 utifrån ett användbarhetsperspektiv (Evaluating the OP-1 Synthesizer from a Usability Perspective)*, s.l.: s.n.

Adobe, 2023. *Adobe Unveils Future of Creative Cloud With Generative AI as a Creative Co-Pilot in Photoshop*. [Online]  
Available at: <https://news.adobe.com/news/news-details/2023/Adobe-Unveils-Future-of-Creative-Cloud-with-Generative-AI-as-a-Creative-Co-Pilot-in-Photoshop-default.aspx/default.aspx#:~:text=Photoshop's%20Generative%20Fill%20feature%20is,within%20the%20Firefly%20beta%20ap>

Alboaneen, D., Alsaffar, D., Alateeq, A. & Alqahtani, A., 2020. *Internet of Things Based Smart Mirrors: A Literature Review*. [Online]  
Available at: <https://www.doi.org/10.1109/ICCAIS48893.2020.9096719>  
[Accessed 20 February 2023].

Altman, S., 2024. *Unconfuse Me with Bill Gates | Episode 6: Sam Altman*. [Online]  
Available at: <https://www.youtube.com/watch?v=PkXELH6Y2lM>

Amazon, 2024. *NVIDIA Jetson Nano Developer Kit (945-13450-0000-100)*. [Online]  
Available at: <https://www.amazon.co.uk/NVIDIA-Jetson-Nano-Developer-945-13450-0000-100/dp/B084DSDDL1>  
[Accessed 03 February 2024].

Apple, 2008. *Apple Reports Fourth Quarter Results | October 2008*. [Online]  
Available at: <https://www.apple.com/uk/newsroom/2008/10/21Apple-Reports-Fourth-Quarter-Results/>

Apple, 2023. *Siri can now help users access and log their Health app data*. [Online]  
Available at: <https://www.apple.com/newsroom/2023/12/siri-can-now-help-users-access-and-log-their-health-app-data/>

Apple, 2024. *Digital Crown*. [Online]  
Available at: <https://developer.apple.com/design/human-interface-guidelines/digital-crown>

Arm, 2024. *What is Ambient Intelligence?*. [Online]  
Available at: <https://www.arm.com/glossary/ambient-intelligence#:~:text=Ambient%20intelligence%20is%20the%20concept,that%20require%20no%20human%20intervention.>

Augmental, 2024. *MouthPad^ by Augmental - Using the Tongue to Navigate Google Maps*, s.l.: s.n.

Barret, J. & Krueger, H., 2007. Performance effects of reduced proprioceptive feedback on touch typists and casual users in a typing task. *Behaviour & Information Technology*, 13(6), pp. 373-381.





- Berol, D., 2017. *The Advantages of Using Piezoelectric MEMS Microphones in Your Alexa-Enabled Product*. [Online]  
Available at: <https://developer.amazon.com/en-US/blogs/alexa/post/bcef47f5-f1ca-4614-8a57-6af7eabfd1eb/the-advantages-of-using-piezoelectric-mems-microphones-in-your-alexa-enabled-product.html>
- Betker, J., 2023. *Better speech synthesis through scaling*. [Online]  
Available at: <https://arxiv.org/pdf/2305.07243.pdf>  
[Accessed 03 04 2024].
- Betker, J., 2023. *Better speech synthesis through scaling*. [Online]  
Available at: <https://arxiv.org/pdf/2305.07243>
- British Recorded Music Industry, 2023. *BPI Streaming Report 2023*. [Online]  
Available at: <https://www.bpi.co.uk/media/3742/bpi-streaming-report-2023.pdf>
- Brooke, J., 1996. SUS: A 'quick and dirty' usability scale. In: *Usability Evaluation in Industry*. London, UK: Taylor & Francis, pp. 189-194.
- Brown, T. et al., 2020. *Language Models are Few-Shot Learners*. s.l., Advances in Neural Information Processing Systems 33.
- Bubeck, S. et al., 2023. *Sparks of Artificial General Intelligence*. [Online]  
Available at: <https://arxiv.org/abs/2303.12712>
- Budach, L. et al., 2022. *The Effects of Data Quality on Machine Learning Performance*, Potsdam, Germany: Hasso Plattner Institute.
- Business Insider , 2023. *Vinyl sales surpassed CDs for the first time in 35 years as records make 'remarkable resurgence'*. [Online]  
Available at: <https://www.businessinsider.com/vinyl-sales-surpass-cds-first-time-since-1987-record-resurgence-2023-3>
- Buxton, B., 2012. A Touching Story: A Personal Perspective on the History of Touch Interfaces Past and Future. *Society for Information Display International Symposium Digest of Technical Papers*, 44(1), pp. 444-448.
- Clicks, 2024. *Clicks Keyboard for iPhone*. [Online]  
Available at: <https://www.clicks.tech/>
- Cox, A. L. et al., 2016. *Design Frictions for Mindful Interactions: The Case for Microboundaries*. s.l., s.n.
- cprime, 2023. *Agile for Hardware | Case Study: Plantronics Applies Agile Practices to a Research and Development Team*. [Online]  
Available at: [https://www.cprime.com/wp-content/uploads/2023/03/Case-Study\\_Plantronics-Agile-for-Hardware\\_v5.pdf](https://www.cprime.com/wp-content/uploads/2023/03/Case-Study_Plantronics-Agile-for-Hardware_v5.pdf)
- Crawford, M. B., 2015. Introduction. Attention as a Cultural Problem. In: *Beyond Your Head: On Becoming an Individual in an Age of Distraction*. s.l.:Farrar, Straus and Giroux, p. 11.
- Cutillo, M., 2021. *Dopamine-Driven Feedback Loops: What Are They?*, Monmouth: The Outlook.







DataReportal, 2024. *Share of users worldwide accessing the internet in 3rd quarter 2023, by device*. [Online]

Available at: <https://www.statista.com/statistics/1289755/internet-access-by-device-worldwide/>

Duan, J. et al., 2022. A Survey of Embodied AI: From Simulators to Research Tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(2), pp. 203-244.

Duan, Y., Guo, X. & Zhu, Z., 2023. *DiffusionDepth: Diffusion Denoising Approach for Monocular Depth Estimation*, s.l.: s.n.

Duan, Y. et al., 2023. *DeWave: Discrete EEG Waves Encoding for Brain*. [Online]

Available at:

[https://proceedings.neurips.cc/paper\\_files/paper/2023/file/1f2fd23309a5b2d2537d063b29ec1b52-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2023/file/1f2fd23309a5b2d2537d063b29ec1b52-Paper-Conference.pdf)

[Accessed 1 April 2024].

Dunning, D., 2011. Chapter Five – The Dunning–Kruger Effect: On Being Ignorant of One's Own Ignorance. *Advances in Experimental Social Psychology*, Vol. 44(1), p. 247–296.

EastRising, 2024. *ER-TFTM1.69-1-SPI*. [Online]

Available at: [https://www.buydisplay.com/download/manual/ER-TFTM1.69-1\\_Datasheet.pdf](https://www.buydisplay.com/download/manual/ER-TFTM1.69-1_Datasheet.pdf)

[Accessed 15 February 2024].

ElevenLabs, 2024. *Speech Synthesis*. [Online]

Available at: <https://elevenlabs.io/app/speech-synthesis>

[Accessed 11 April 2024].

Elo, A., 1967. The Proposed USCF Rating System | It's Development, Theory and Applications. *Chess Life*, 22(8), p. 242–247.

Endrissat, N., Islam, G. & Noppeney, C., 2016. Visual organizing: Balancing coordination and creative freedom via mood boards. *Journal of Business Research*, 69(7), pp. 2353-2362.

epson, 2024. *Projectors*. [Online]

Available at: [https://www.epson.co.uk/en\\_GB/products/projectors/c/projector](https://www.epson.co.uk/en_GB/products/projectors/c/projector)

[Accessed 15 April 2024].

European Commission, 2023. *Digital Markets Act: Commission designates six gatekeepers*. [Online]

Available at: [https://ec.europa.eu/commission/presscorner/detail/en/ip\\_23\\_4328](https://ec.europa.eu/commission/presscorner/detail/en/ip_23_4328)

Fadell, T., 2022. Why Storytelling. In: *Build: An Unorthodox Guide to Making Things Worth Making*. s.l.:Bantam, p. 107.

Fahmi, M., 2024. *WIP thoughts on forms of human-AI interaction*. [Online]

Available at: <https://twitter.com/minafahmi/status/1779917014984757566>

Fan, J., 2023. *Jim Fan | Twitter*. [Online]

Available at: <https://twitter.com/drjimfan/status/1722684012064546887>

Fraser, C., 2024. *Traversal Of The Embedding Space*. [Online]

Available at: [https://twitter.com/colin\\_fraser/status/1768781713008660585](https://twitter.com/colin_fraser/status/1768781713008660585)





Google Cloud, 2023. *ElevenLabs debuts a generative AI solution for spoken content using customizable voices on Google Cloud*. [Online]  
Available at: <https://cloud.google.com/blog/topics/startups/elevenlabs-debuts-a-generative-ai-solution-on-google-cloud>  
[Accessed 21 April 2024].

Google Cloud, 2023. *ElevenLabs debuts a generative AI solution for spoken content using customizable voices on Google Cloud*. [Online]  
Available at: <https://cloud.google.com/blog/topics/startups/elevenlabs-debuts-a-generative-ai-solution-on-google-cloud>  
[Accessed 12 April 2024].

Google Research, 2017. *Attention Is All You Need*. s.l., Advances in Neural Information Processing Systems 30 (NIPS 2017).

Google, 2024. *File types & external devices that work on Chromebooks*. [Online]  
Available at: <https://support.google.com/chromebook/answer/183093?hl=en>

gpt-4-turbo, 2024. *gpt-4-turbo*. [Online]  
Available at: <https://platform.openai.com/playground?mode=chat&model=gpt-4-turbo-2024-04-09>

Gwern, 2020. *The Scaling Hypothesis*. [Online]  
Available at: <https://gwern.net/scaling-hypothesis>

HuggingFace, 2024. *Sentence Transformers*. [Online]  
Available at: <https://huggingface.co/sentence-transformers>

Humane, 2023. *ai pin*. [Online]  
Available at: <https://hu.ma.ne/aipin>

iFixit, 2019. *Good Vibrations: How Apple Dominates the Touch Feedback Game*. [Online]  
Available at: <https://www.ifixit.com/News/16768/apple-taptic-engine-haptic-feedback>

Imel, D., 2021. *On Google's Pixel, Tensor, and the age of ambient computing*. [Online]  
Available at: <https://www.youtube.com/watch?v=jfS8aGSDKGU>

Janeck, M., 2022. *Investigating the Wavelength and Polarization of Electromagnetic Wave using Raspberry Pi Zero W*. [Online]  
Available at: <https://www.doi.org/10.1109/ciees55704.2022.9990878>  
[Accessed 23 01 2024].

Jelinek, F., Mercer, R. L., Bahl, L. R. & Baker., J. K., 1977. Perplexity - a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(1), p. 63.

Jelinek, F., Mercer, R. L., Bahl, L. R. & Baker, J. K., 1977. *Perplexity—a measure of the difficulty of speech recognition tasks*. [Online]  
Available at: <https://doi.org/10.1121%2F1.2016299>

Ji, C. C.-J. et al., 2024. *Gorilla: Large Language Model Connected with Massive APIs*. [Online]  
Available at: [https://gorilla.cs.berkeley.edu/blogs/7\\_open\\_functions\\_v2.html](https://gorilla.cs.berkeley.edu/blogs/7_open_functions_v2.html)  
[Accessed 01 04 2024].





- jsjmch, 2023. *Hacker News*. [Online]  
Available at: <https://news.ycombinator.com/item?id=38339038>  
[Accessed 15 January 2024].
- Juang, B. H. & Rabiner, L. R., 2005. Automatic Speech Recognition - A Brief History of the Technology Development. In: *Elsevier Encyclopedia of Language and Linguistics*. s.l.:s.n.
- Judd, W., 2017. *Full-size, TKL, 60% and more: a guide to mechanical keyboard sizes*. [Online]  
Available at: <https://www.keyboardco.com/blog/index.php/2017/08/full-size-tkl-60-and-more-a-guide-to-mechanical-keyboard-sizes/>
- Kaplan, J. et al., 2020. *Scaling Laws for Neural Language Models*, San Fransisco: OpenAI.
- Kar, A., Norouzi, M., Fleet, D. J. & Saxena, S., 2023. *Monocular Depth Estimation using Diffusion Models*, s.l.: s.n.
- Karparthy, A., 2023. *LLM OS*. [Online]  
Available at: <https://twitter.com/karparthy/status/1723140519554105733/photo/1>
- Képuska, V., 2011. Wake-up-word speech recognition. *Speech Technologies*, pp. 237-262.
- Kessel, M. & Arkinson, C., 2015. Measuring the Superfluous Functionality in Software Components. *CBSE '15: Proceedings of the 18th International ACM SIGSOFT Symposium on Component-Based Software Engineering*, May, pp. 11-20.
- Kojima, T. et al., 2022. *Large Language Models are Zero-Shot Reasoners*. [Online]  
Available at: <https://arxiv.org/pdf/2205.11916.pdf>
- Korzybski, A., 1933. A Non-Aristotelian System and Its Necessity for Rigour in Mathematics and Physics. *Science and Sanity*, pp. 747-761.
- Lai, H., Toral, A. & Nissim, M., 2021. *Thank you BART! Rewarding Pre-Trained Models Improves Formality Style Transfer*. [Online]  
Available at: <https://www.doi.org/10.18653/V1/2021.ACL-SHORT.62>
- Lambda Labs, 2024. *The GPU Cloud for AI | On-demand & reserved cloud NVIDIA GPUs for AI training & inference*. [Online]  
Available at: <https://lambdalabs.com/>
- Langchain, 2024. *Semantic Chunking*. [Online]  
Available at:  
[https://python.langchain.com/docs/modules/data\\_connection/document\\_transformers/semantic-chunker](https://python.langchain.com/docs/modules/data_connection/document_transformers/semantic-chunker)
- Lang, J. H. & Argwal, A., 2005. *Foundations of analog and digital electroinc circuits*. p43 ed. Amsterdam: Elsevier.
- LaZerte, J. D., 1998. Market Pull/Technology Push. *Research-Technology Management*, March, 31(2), p. 1.
- Lee, S. & Zhai, S., 2009. *The performance of touch screen soft buttons..* s.l., Conference on Human Factors in Computing Systems.
- Leger, D. S. & Luengo, G. S., 2023. The human touch: A connected neuro-cellular skin-brain network. *Skin Research & Technology*, 29(4), p. 1.





Leow, C. S., Goh, W. L. & Gao, Y., 2023. *Sparsity Through Spiking Convolutional Neural Network for Audio Classification at the Edge*, s.l.: s.n.

Lewis, J. R., 2017. *Can I Leave This One Out? The Effect of Dropping an Item From the SUS*.

[Online]

Available at:

[https://www.researchgate.net/publication/321335936\\_Can\\_I\\_Leave\\_This\\_One\\_Out\\_The\\_Effect\\_of\\_Dropping\\_an\\_Item\\_From\\_the\\_SUS](https://www.researchgate.net/publication/321335936_Can_I_Leave_This_One_Out_The_Effect_of_Dropping_an_Item_From_the_SUS)

Li, C. et al., 2023. *Large Language Models Understand and Can Be Enhanced by Emotional Stimuli*. [Online]

Available at: <https://arxiv.org/pdf/2307.11760.pdf>

Lima, D. B. C. et al., 2019. *A Performance Evaluation of Raspberry Pi Zero W Based Gateway Running MQTT Broker for IoT*. Vancouver, BC, Canada, 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON).

Lim, J. K., 2019. *Algorithms Explained: RSA Encryption*. [Online]

Available at: <https://medium.com/@jinkyulim96/algorithms-explained-rsa-encryption-9a37083aaa62>

[Accessed 12 February 2024].

Linux, 2002. *arecord(1) - Linux man page*. [Online]

Available at: <https://linux.die.net/man/1/arecord>

[Accessed 1 April 2024].

Liu, N. F. et al., 2023. *Lost in the Middle: How Language Models Use Long Contexts*. [Online]

Available at: <https://arxiv.org/pdf/2307.03172.pdf>

Liu, Y. et al., 2024. *Sora: A Review on Background, Technology, Limitations, and Opportunities of Large Vision Models*. [Online]

Available at: <https://arxiv.org/abs/2402.17177>

Liu, Y. et al., 2024. *Sora: A Review on Background, Technology, Limitations, and Opportunities of Large Vision Models*, s.l.: s.n.

Li, Y., Reddy, R., Zhang, C. & Nandakumar, R., 2023. *Beyond-Voice: Towards Continuous 3D Hand Pose Tracking on Commercial Home Assistant Devices*. [Online]

Available at: <https://arxiv.org/abs/2306.17477>

Li, Z., Yang, Z. & Wang, M., 2023. *Reinforcement Learning with Human Feedback: Learning Dynamic Choices via Pessimism*, s.l.: arxiv.

LMSYS, 2024. *Chatbot Arena Leaderboard*. [Online]

Available at: <https://chat.lmsys.org/?leaderboard>

[Accessed 16 04 2024].

MacKay, D., 2002. *Dasher*. [Online]

Available at: <https://inference.org.uk/dasher/DasherSummary2.html>

Madary, M., 2022. The Illusion of Agency in Human-Computer Interaction. *Neuroethics*, 15(1), pp. 1-15.





- Manuel, S. T. S., 2020. *A/B-Testing – Verfahren zur Optimierung der digitalen Interaktion zwischen Konsumenten und Unternehmen*. [Online]  
Available at: [http://dx.doi.org/10.1007/978-3-658-29995-8\\_7](http://dx.doi.org/10.1007/978-3-658-29995-8_7)
- McAlpine, K. B., 2018. The ZX Spectrum: The Sound of One Bit. In: *Bits and Pieces: A History of Chiptunes*. New York: Oxford Academic, pp. 37-68.
- Mejtoft, T., Hale, S. & Söderström, U., 2019. *Design Friction*. s.l., s.n.
- Meta, 2024. *Access and download your information on Instagram*. [Online]  
Available at: <https://help.instagram.com/181231772500920>
- Mielke, S. J. et al., 2021. *Between words and characters: A Brief History of Open-Vocabulary Modeling and Tokenization in NLP*. [Online]  
Available at: <https://arxiv.org/abs/2112.10508>
- Miner, C., 2001. Pushing Functionality into Even Smaller Devices. *Communications of the Association of Computing Machinery*, March, 44(3), pp. 72-73.
- Moravek, M., 2023. *minimalist phone: reduce your screentime*. [Online]  
Available at: [https://www.producthunt.com/products/minimalist-phone-reduce-your-screentime?utm\\_source=badge-top-post-topic-badge&utm\\_medium=badge#minimalist-phone-reduce-your-screentime](https://www.producthunt.com/products/minimalist-phone-reduce-your-screentime?utm_source=badge-top-post-topic-badge&utm_medium=badge#minimalist-phone-reduce-your-screentime)
- Naganandhini, K. et al., 2022. *Hand Tracking Based Human-Computer Interaction Teaching System*. [Online]  
Available at: <https://www.doi.org/10.1109/iciict55121.2022.10064617>  
[Accessed 23 February 2024].
- Norman, D. A. & Draper, S. W., 1986. *User Centered System Design; New Perspectives on Human-Computer Interaction*. New Jersey: L. Erlbaum Associates Inc..
- Norton, K. G. & Mimi, L., 2020. The Rostrolateral Prefrontal Cortex Mediates a Preference for High-Agency Environments. *The Journal of Neuroscience*, 40(22), pp. 4401-4409.
- NVIDIA, 2024. *Kernal Customization*. [Online]  
Available at: <https://docs.nvidia.com/jetson/archives/r35.3.1/DeveloperGuide/text/SD/Kernel/KernelCustomization.html>
- OED, 2023. *Oxford English Dictionary*. s.l.:Oxford University Press.
- OpenAI, 2020. *API Refrence / Tempreature*. [Online]  
Available at: <https://platform.openai.com/docs/api-reference>
- OpenAI, 2021. *Embeddings / Learn how to turn text into numbers, unlocking use cases like search..* [Online]  
Available at: <https://platform.openai.com/docs/guides/embeddings/what-are-embeddings>
- OpenAI, 2021. *Tokenizer*. [Online]  
Available at: <https://platform.openai.com/tokenizer>





OpenAI, 2022. *ChatGPT: Get instant answers, find creative inspiration, learn something new.* [Online]

Available at: <https://openai.com/chatgpt>

OpenAI, 2023. *Function calling and other API updates.* [Online]

Available at: <https://openai.com/blog/function-calling-and-other-api-updates>

OpenAI, 2023. *GPT-4 Technical Report*, San Francisco: OpenAI.

OpenAI, 2024. *Fine-tuning.* [Online]

Available at: <https://platform.openai.com/docs/guides/fine-tuning/preparing-your-dataset>

OpenAI, 2024. *OpenAI API Documentation | Speech to Text.* [Online]

Available at: <https://platform.openai.com/docs/guides/speech-to-text/quickstart>

OpenAI, 2024. *Video generation models as world simulators.* [Online]

Available at: <https://openai.com/research/video-generation-models-as-world-simulators>

Oracle, 2020. *What is IoT?* [Online]

Available at: <https://www.oracle.com/uk/internet-of-things/what-is-iot/#:~:text=What%20is%20IoT%3F,and%20systems%20over%20the%20internet.>

Orban, G. A. & Caruana, F., 2014. The neural basis of human tool use. *Frontiers in Psychology*, Volume 5, pp. 1-2.

Osborne, A., 1980. *An Introduction to Microcomputers: Volume 1: Basic Concepts*. s.l.:Osborne McGraw Hill.

Packer, C. et al., 2023. *MemGPT: Towards LLMs as Operating Systems.* [Online]

Available at: <https://arxiv.org/pdf/2310.08560.pdf>

Packer, C. et al., 2023. *MemGPT: Towards LLMs as Operating Systems*, s.l.: s.n.

Partiwala, S. et al., 2020. *Increasing cpu clock speed to improve system performance.* US, Patent No. US201816606066.

Pinaya, W. H. L. et al., 2023. *Generative AI for Medical Imaging: extending the MONAI Framework.* [Online]

Available at: <https://arxiv.org/abs/2307.15208>

Pinecone, 2023. *Why use reterival instead of larger contexts.* [Online]

Available at: <https://pinecone.io/blog/why-use-retrieval-instead-of-larger-context/>

PostgreSQL, 2024. *What is PostgreSQL.* [Online]

Available at: <https://www.postgresql.org/about/>

Powerll, E., 2014. *Cheap Projectors -- how bad are they?* [Online]

Available at: [https://www.projectorcentral.com/cheap\\_projectors.htm](https://www.projectorcentral.com/cheap_projectors.htm)

[Accessed 4 April 2024].

Python Docs, 2013. *Reading & Parsing JSON Data With Python: Tutorial.* [Online]

Available at: <https://python.readthedocs.io/en/v2.7.2/library/json.html>

Qin, Z., Lee, K.-C., Yeh, Y.-W. & Huang, Y.-P., 2016. *31-3: Suppression of Ambient-Light Reflection in OLED Display by Using a Micro-Lens Array and Cruciform Black Matrices.* [Online]





Available at: <https://www.doi.org/10.1002/SDTP.10700>

[Accessed 1 April 2024].

Radford, A. et al., 2021. *Learning Transferable Visual Models From Natural Language Supervision*. [Online]

Available at: <https://arxiv.org/pdf/2103.00020.pdf>

Radford, A. et al., 2022. *Robust Speech Recognition via Large-Scale Weak Supervision*, San Francisco: OpenAI.

reMarkable, 2024. *Type Folio*. [Online]

Available at: <https://remarkable.com/store/remarkable-2/type-folio>

Rodríguez, C. V. & Garcia Garcia, R., 2014. *Implementation of a low cost video wall using Raspberry Pi devices*, s.l.: s.n.

Rodriguez, J. A. et al., 2023. *StarVector: Generating Scalable Vector Graphics Code from Images*. [Online]

Available at: <https://arxiv.org/pdf/2312.11556v1>

Royce, W. W., 1970. *MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS*. [Online]

Available at: <https://blog.jbrains.ca/assets/articles/royce1970.pdf>

Saxena, S. et al., 2023. *Zero-Shot Metric Depth with a Field-of-View Conditioned Diffusion Model*, s.l.: s.n.

Schick, T. et al., 2023. *Toolformer: Language Models Can Teach Themselves to Use Tools*. [Online]

Available at: <https://arxiv.org/pdf/2302.04761.pdf>

[Accessed 11 10 2023].

Schön, D. A., 1983. *The Reflective Practitioner: How Professionals Think In Action*. s.l.:s.n.

Shannon, C. E., 1951. Prediction and entropy of printed English. *Bell Systems Technical Journal*, 1(30), pp. 50-64.

Singh, V., 2024. *Exploring LLM Pricing Cost*. [Online]

Available at:

<https://colab.research.google.com/gist/virat/f290fa4ec878f006ca1264899645182a/exploring-llm-pricing-cost.ipynb>

Soegaard, M., 2023. *User Experience (UX) Surveys: The Ultimate Guide*. [Online]

Available at: <https://www.interaction-design.org/literature/article/ux-surveys>

Sparkfun, 2024. *Surface Mount Navigation Switch*. [Online]

Available at: <https://www.sparkfun.com/products/8184>

[Accessed 2 April 2024].

Sruthi, R. et al., 2023. Real-Time Utilities Consumption Monitoring System Using ESP32. *Journal of Control System and Control Instrumentation*, 9(2).

StatCounter, 2024. *Desktop vs Mobile vs Tablet Market Share Worldwide*. [Online]

Available at: <https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet>





- Statistica, 2016. *Vinyl Record Sales Worldwide*. [Online]  
Available at: <https://www.statista.com/statistics/297763/vinyl-record-sales-worldwide/>
- Statistica, 2022. *Factors surrounding preference of voice assistants over websites and applications, worldwide, as of 2017*. [Online]  
Available at: <https://www.statista.com/statistics/801980/worldwide-preference-voice-assistant-websites-app/>
- Statistica, 2023. *Sales volume of vinyl records in Finland from 2012 to 2022*. [Online]  
Available at: <https://www.statista.com/statistics/561796/music-industry-physical-audio-vinyl-sales-finland/>
- Statistica, 2023. *Unit sales of vinyl records in Australia from 2015 to 2022*. [Online]  
Available at: <https://www.statista.com/statistics/693324/vinyl-sales-australia/>
- Steck, H., Ekanadham, C. & Kallus, N., 2024. *Is Cosine-Similarity of Embeddings Really About*. [Online]  
Available at: <https://arxiv.org/pdf/2403.05440.pdf>
- Steck, H., Ekanadham, C. & Kallus, N., 2024. *Is Cosine-Similarity of Embeddings Really About Similarity?*. [Online]  
Available at: <https://arxiv.org/abs/2403.05440>
- Stefanovic, I., Nan, E. & Radin, B., 2017. *Implementation of the wake word for smart home automation system*. [Online]  
Available at: [10.1109/ICCE-BERLIN.2017.8210649](https://doi.org/10.1109/ICCE-BERLIN.2017.8210649)
- Stoyanaov, Y., 2021. *Direct Memory Access (DMA) in Embedded Systems*. [Online]  
Available at: <https://open4tech.com/direct-memory-access-dma-in-embedded-systems/>
- Suman, M., Tejpal, S. & Soumik, P., 2023. *IoT based Smart Mirror Using Raspberry Pi*. [Online]  
Available at: <https://www.doi.org/10.22214/ijraset.2023.48583>  
[Accessed 23 01 2024].
- Sun, Y., Geng, L. & Dan, K., 2018. *Design of Smart Mirror Based on Raspberry Pi*. [Online]  
Available at: <https://www.doi.org/10.1109/ICITBS.2018.00028>  
[Accessed 01 February 2023].
- Suryawidjaja, V., 2023. Digital Literacy and Growth Mindset to Predict Technology Acceptance During Learning Processes. *PSIKODIMENSIA Kajian Ilmiah Psikologi*, 22(1), pp. 77-82.
- Sussillo, D., Kaifosh, P. & Reardon, T., 2024. *A generic noninvasive neuromotor interface for human-computer interaction*. [Online]  
Available at: <https://doi.org/10.1101/2024.02.23.581779>  
[Accessed 05 April 2024].
- teenage engineering, 2024. *tp-7*. [Online]  
Available at: <https://teenage.engineering/products/tp-7>
- Telecco, N., Vijaya, P. & Adusmilli, 2004. *Direct Memory Access Interface in Integrated Circuits*, s.l.: s.n.







- ThePiHut, 2024. *NVIDIA Jetson Nano 4GB Developer Kit*. [Online]  
Available at: <https://thepihut.com/products/nvidia-jetson-nano-4gb-developer-kit>  
[Accessed 03 February 2024].
- ThePiHut, 2024. *Raspberry Pi 4 Model B*. [Online]  
Available at: <https://thepihut.com/products/raspberry-pi-4-model-b?variant=20064052740158>  
[Accessed 12 February 2024].
- Thorn, A., 2022. *Transhumanism | Philosophy Tube*. [Online]  
Available at: <https://www.youtube.com/watch?v=DqPd6MShV1o>
- Time, 2019. *How America's Elites Lost Their Grip*. [Online]  
Available at: <https://time.com/5735384/capitalism-reckoning-elitism-in-america-2019/>
- Tolentino, R. E. et al., 2021. *Recognizing Threat through Object and Gesture Recognition by Applying Convolutional Neural Network and Adaptive Boosting Algorithm Using Microsoft Kinect for Xbox One*. [Online]  
Available at: <https://www.doi.org/10.1109/MASCON51689.2021.9563498>
- Tsendsuren, M., Faruqui, M. & Gopal, S., 2024. *Leave No Context Behind: Efficient Infinite Context Transformers with Infini-attention*, s.l.: Google.
- TTC, 2021. *TTC with wire encoder*. [Online]  
Available at: <http://www.ttcswitch.cn/product/88.html>  
[Accessed 2 April 2024].
- T-vK, 2020. <https://github.com/T-vK/ESP32-BLE-Mouse>. [Online]  
Available at: <https://github.com/T-vK/ESP32-BLE-Keyboard>  
[Accessed 01 December 2023].
- Umare, A. et al., 2023. Wi-Fi Pan Tilt Surveillance Camera Using ESP32. *International Journal for Research in Applied Science & Engineering Technology*, 11(6), pp. 2375-2382.
- UTS HAI Research, 2023. *UTS HAI Research - BrainGPT*. [Online]  
Available at: <https://www.youtube.com/watch?v=crJst7Yfzj4>  
[Accessed 1 April 2024].
- Vaswani, A. et al., 2017. *Attention Is All You Need*. [Online]  
Available at: <https://arxiv.org/abs/1706.03762>
- Vaswani, A. et al., 2017. Attention Is All You Need. *Neural Information Processing Systems*, Issue 30, pp. 5998-6008.
- Vercel, 2024. *Introducing AI SDK 3.0 with Generative UI support*. [Online]  
Available at: <https://vercel.com/blog/ai-sdk-3-generative-ui>
- Vishay, 2024. *OLED-128O064D-BPP3N00000*. [Online]  
Available at: <https://www.vishay.com/docs/37902/oled128o064dbpp3n00000.pdf>  
[Accessed 15 March 2024].
- Wang, G. et al., 2023. *Voyager: An Open-Ended Embodied Agent with Large Language Models*. [Online]  
Available at: <https://arxiv.org/pdf/2305.16291.pdf>





- Ward, D. J., 2001. *Adaptive Computer Interfaces*, Cambridge: University of Cambridge.
- Wei, J. et al., 2022. Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research*.
- Weiser, M., 1991. The Computer for the 21st Century. *Scientific American*, 3(265), pp. 94-104.
- World Economic Forum, 2016. *The Fourth Industrial Revolution: what it means, how to respond*. [Online]  
Available at: <https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/>
- Wu, Z. et al., 2024. *OS-Copilot: Towards Generalist Computer Agents with Self-Improvement*, s.l.: s.n.
- Xu, Z. J. S. K. M., 2024. *Hallucination is Inevitable: An Innate Limitation of Large Language Models*, s.l.: School of Computing, National University of Singapore.
- Yao, J. et al., 2023. *Knowledge Plugins: Enhancing Large Language Models for*. [Online]  
Available at: <https://arxiv.org/abs/2311.10779>
- Ye, Q., Axmed, M., Pryzant, R. & Khani, F., 2023. *Prompt Engineering a Prompt Engineer*. [Online]  
Available at: <https://arxiv.org/pdf/2311.05661.pdf>
- Yu, Q. & Zhou, R., 2022. *Wake Word Detection Model Based on Res2Net*, s.l.: s.n.
- Zagal, J. P., Björk, S. & Lewis, C., 2013. *Dark Patterns in the Design of Games*. [Online]  
Available at: <https://core.ac.uk/download/pdf/301007767.pdf>
- Zargham, N. et al., 2022. *"I Want It That Way": Exploring Users' Customization and Personalization Preferences for Home Assistants*. s.l., s.n.
- Zawinski, J., 1995. *jwzhacks | historically significant*. [Online]  
Available at: <https://jwz.org/hacks>
- Zhang, T. et al., 2023. Input modality matters: A comparison of touch, speech, and gesture based in-vehicle interaction. *Applied Ergonomics*, 108(1), pp. 1-18.
- Zheng, L. et al., 2023. *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena*. [Online]  
Available at: <https://arxiv.org/pdf/2306.05685.pdf>
- Zwakman, D., Pal, D. & Arpnikanondt, C., 2021. Usability Evaluation of Artificial Intelligence-Based Voice Assistants: The Case of Amazon Alexa. *SN Comput Sci*, 2(1).





## Appendix 1 | Large Language Models and Generative AI

The release of ChatGPT in November 2022 marked a significant transformation in how we interact with technology. Since its introduction Generative AI has begun a deep integration into various facets of everyday life and industry (OpenAI, 2022) (Liu, et al., 2024) (OpenAI, 2023).

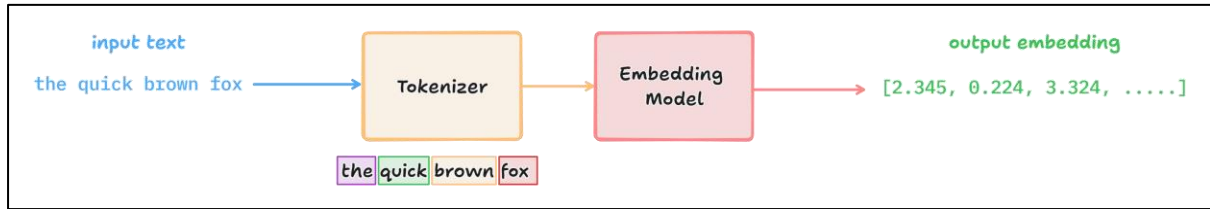


Figure 56 | Embedding at a High Level

Natural Language Processing (NLP) introduces us to tokenization (Mielke, et al., 2021), splitting strings from a large corpus of data into smaller pieces of text that are easier to process (OpenAI, 2021). The tokens are passed to an embedding model and transformed into an n-dimensional vector of numbers, capturing their semantic meaning.

This process is used to map ‘high dimensional data’ like words, images, and other complex structures into a continuous ‘low dimensional’ latent space. Inputs with similar contexts and meanings have similar output embeddings. This can be used to measure the relatedness of words (OpenAI, 2021). Methods such as Euclidean Distance and Cosine Similarity are commonly used to evaluate the semantic similarity (Steck, et al., 2024).

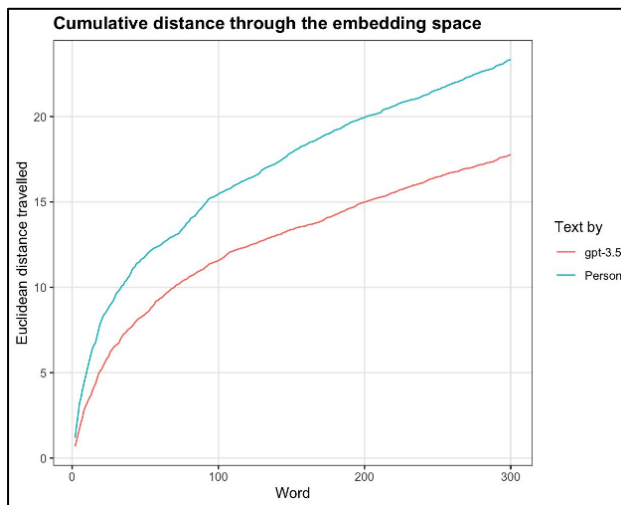


Figure 57 | Human Generated Text vs GPT-3.5  
Generated Text (Fraser, 2024)

Continued development has seen Transformer-based models (Vaswani, et al., 2017) pre trained to predict the next token in a string based on their statistical probability of appearing. These models utilize a training process that involves billions of “key-value” JSON pairs which help in learning the complex relationships within the data.

Perplexity is a measure of this probability. Defined as the certainty in predicting an output, as perplexity increases the less likely the output is (Jelinek, et al., 1977). In a language model this is referred to as temperature (OpenAI, 2020), controlling the deterministic nature of the text.

As the temperature increases the generated output becomes less deterministic. This ‘low perplexity’ deterministic output is better suited for

consistent outputs tasks whereas less deterministic output is best for diverse and creative results.

Language models are generally more deterministic, as such the distance a between tokens in the ‘low dimensional’ embedding space is lower on average than one generated by a human, suggesting that the language model’s outputs are more predictable and constrained. Human language in contrast, takes less predictable paths through the embedding space, highlighted by the greater Euclidian distance travelled.





Figure 58 | OpenAI’s Sora Output at Base Compute (Left) 4x Compute (Middle) 32x Compute (Right)

It has been presented that scaling a model’s dataset may be all that is required to improve performance (Kaplan, et al., 2020) (Gwern, 2020). OpenAI’s Sora, an image generation model using the diffusion architecture has already presented impressive results by scaling computational resources alone (OpenAI, 2024). Similar to how text-based language models are trained on extensive collections of text, image diffusion models require the conversion of image and video data to the same lower dimensional latent space wherein common characteristics can be understood in relation to each other by the model (Liu, et al., 2024).

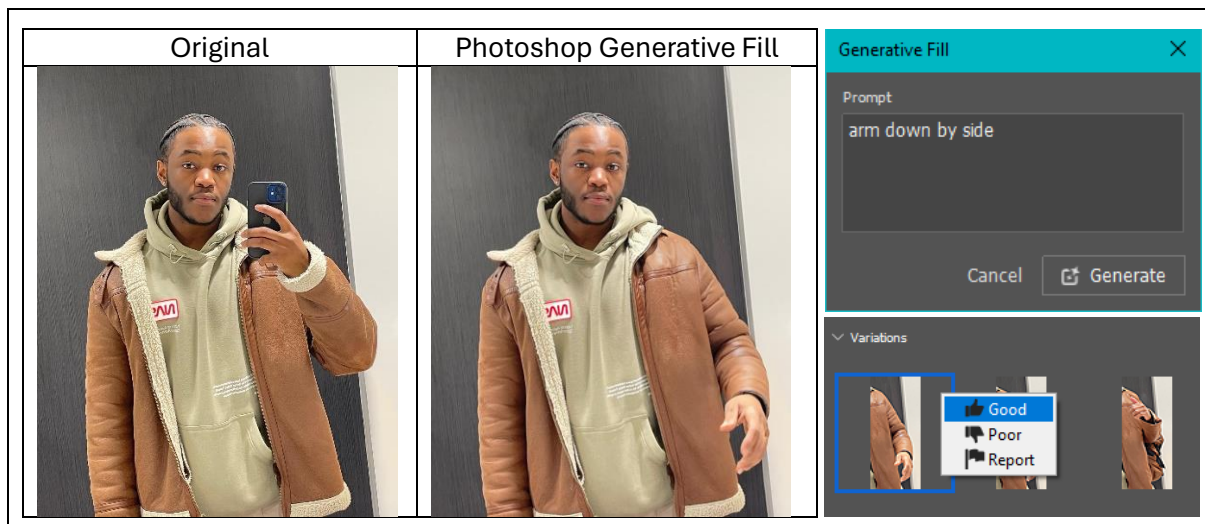


Figure 59 | Photoshop Generative Fill example (arm with phone was masked using the lasso tool)

Adobe marked a step in the mass adoption of these tools with the addition of Generative AI as a “Creative Co-Pilot” in Photoshop (Adobe, 2023). Using their closed source ‘Firefly’ diffusion model users can edit images using natural language alone. Their UI also employs a form of Reinforcement Learning via Human Feedback (RLHF) asking the user to rate the variations of the generation. A method in machine learning that improves model performance by teaching it to recognize outputs that meet acceptable standards based on user ratings (Li, et al., 2023).

When developing software, we interact with Language Models through querying their RestAPI (often abstracting this process using a Python library) and receiving the response. In the same way these models ‘predict the next token’ their response is generated a token at a time, we can measure the throughput performance of a model in Tokens per Second.





### 1.1 Evaluation

The Elo rating system is a method for calculating the relative skill levels of players (Elo, 1967), widely adopted in chess and over games of a competitive nature. LMSYS Chatbot Arena is a crowd sourced open platform for LLM evaluation (LMSYS, 2024), employing “battles” users can interact with two anonymous models simultaneously, posing the same question to both. They vote for which model provides the preferred response, with the identities of the models disclosed post-voting (Zheng, et al., 2023). Publicly accessible Elo ratings are then generated for the models. This method of analysis was first proposed in the paper ‘Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena’ (Zheng, et al., 2023).

Rank	Model	Arena Elo	Votes	Org	License	Knowledge Cutoff
1	GPT-4-Turbo-2024-04-09	1260	15751	OpenAI	Proprietary	2023/12
1	Claude 3 Opus	1255	56101	Anthropic	Proprietary	2023/8
1	GPT-4-1106-preview	1254	65159	OpenAI	Proprietary	2023/4
2	GPT-4-0125-preview	1250	50923	OpenAI	Proprietary	2023/12
5	Bard (Gemini Pro)	1209	12468	Google	Proprietary	Online
5	Claude 3 Sonnet	1203	62056	Anthropic	Proprietary	2023/8
7	Command R+	1193	29437	Cohere	CC-BY-NC-4.0	2024/3
7	GPT-4-0314	1189	42925	OpenAI	Proprietary	2021/9
9	Claude 3 Haiku	1182	57727	Anthropic	Proprietary	2023/8
10	GPT-4-0613	1164	61520	OpenAI	Proprietary	2021/9
10	Mistral-Large-2402	1158	37650	Minstral	Proprietary	Unknown
11	Qwen1.5-72B-Chat	1154	27826	Alibaba	Qianwen LICENSE	2024/2
12	Claude-1	1150	21868	Anthropic	Proprietary	Unknown
12	Mistral Medium	1148	30764	Mistral	Proprietary	Unknown
12	Command R	1148	33061	Cohere	Proprietary	2024/3
16	Claude-2.0	1131	13484	Anthropic	Proprietary	Unknown
16	Mistral-Next	1127	13113	Minstral	Proprietary	Unknown
16	Qwen1.5-32B-Chat	1136	12812	Alibaba	Proprietary	2024/2
19	Gemini Pro (Dev API)	1135	19798	Google	Proprietary	2023/4
19	Qwen1.5-14B-Chat	1119	17727	Alibaba	Qianwen LICENSE	2024/2
19	GPT-3.5-Turbo-0613	1119	41013	OpenAI	Proprietary	2021/9

Table 43 | Top 21 Models from Chatbot Arena (LMSYS, 2024)

### Notes

- GPT-4-Turbo-2024-04-09 is the most performant general model at time of writing.
  - o Llama-3-70b was released 18/04/2024 and sits joint 5th.
- The performance/price ratio of Haiku actually makes it a viable replacement for GPT-4 in most scenarios assume few shot prompting of examples is implemented
- B refers to the number of parameters the model has, this is a measure of how many “key value” pairs of conversation it is trained on (often in billions).



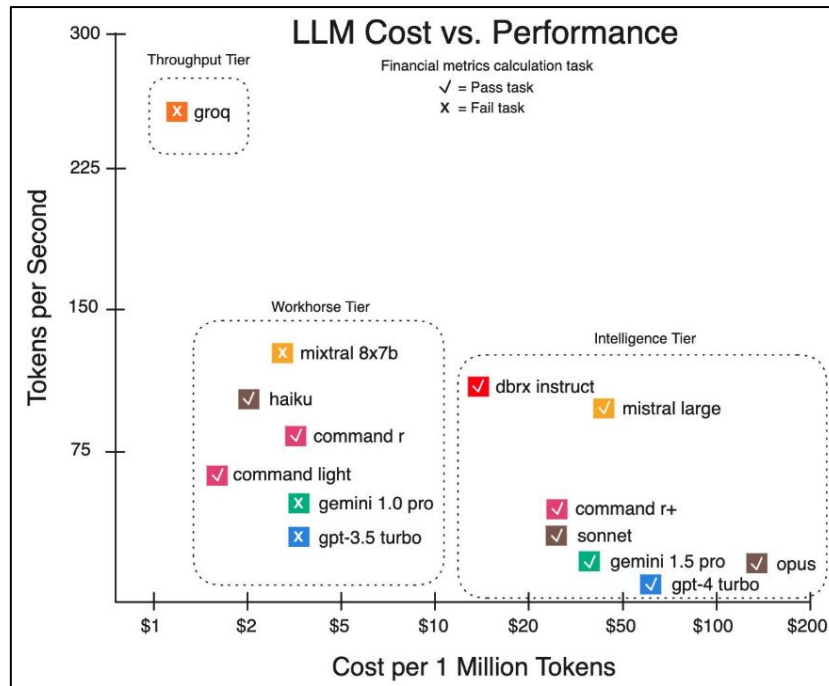


Figure 60 | LLM Cost vs Performance (Singh, 2024)

(Singh, 2024) presents an open-source method of evaluating models currently available via API using an evaluation dataset of financial statements available in a ‘google colab’ collaborative code workspace. Each model has its time to complete the evaluation task measured in tokens per second.

The following prompt is provided to Language Model Judge (Anthropic Opus)

You are an expert at evaluating whether an extracted value is correct or incorrect for a given task.

The task that you are evaluating is JSON key-value extraction and financial calculations. The JSON contains financial statements, including income statements, balance sheets, and cash flow statements.

Your job is to determine (True or False) if the extracted value or calculated result is correct for a given task.

For example, if the task is to extract net income from the JSON and the correct value is 10000, but the extracted value is 1000, then your output is False. However, if the extracted value is 10000, then your output is True.

Note: The extracted values and calculated results may include both text and numbers/metrics. Your job is to focus only on the numbers/metrics and ensure that they match the ground truth values.

Rounding errors are acceptable. The ground truth and extracted values do not need to be exact string matches. The important thing is that the values match.

For example, if ground truth is 4792000000 and \$479.2 million is prediction, then the prediction is correct.

Here are the ground truth values for the requested tasks:  
[values truncated]

Figure 61 | Model Evaluation Prompt (Singh, 2024)





### Appendix 2 | Physical Media Consumption

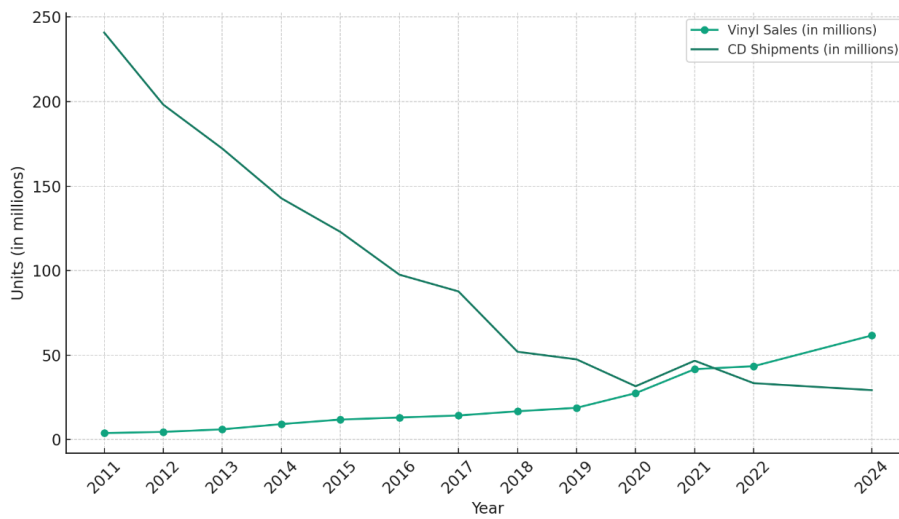


Table 44 | Vinyl Sales vs CD Shipments in the U.S (1999-2024)

Year	Vinyl Sales (Millions)	CD Shipments (Millions)
2023	49.61	33.4
2022	43.46	33.4
2021	41.7	46.7
2020	27.5	31.6
2019	18.84	47.5
2018	16.8	52
2017	14.3	87.7
2016	13.1	97.6
2015	11.9	122.9
2014	9.2	142.8
2013	6.1	172.2
2012	4.6	198.2
2011	3.9	240.8

Table 45 | Vinyl Sales vs CD Shipments

Equation	Usage
$Growth\ Rate = \frac{Value_{Year\ n+1} - Value_{Year\ n}}{Value_{Year\ n}}$	Calculate the annual growth rates for both vinyl sales and CD shipments.
$Average\ Growth\ Rate = \frac{\sum Growth\ Rates_{Recent\ Years}}{Number\ of\ Years}$	To predict the sales/shipments 2024
$Predicted\ Value_{2024} = Value_{2023} \times (1 + Average\ Growth\ Rate)$	Extrapolate to 2024

Table 46 | Equations for Extrapolation





### Appendix 3 | Fine-Tuning Myself

In a 2022 study, Zargham et al. discovered that users favour personal assistants with enhanced emotional capabilities. Additionally, Lai et al. in 2021 demonstrated that refining pre-trained language models leads to better content retention. Inspired by these findings, I embarked on a project to craft a digital version of a personal assistant that mirrors my unique conversational style, by fine-tuning a language model on my text chat writing style.

#### Methodology

- Gather Sample Data
  - The performance of AI systems is bounded by the quality of the underlying training data (Budach, et al., 2022).
- Clean Sample Data
  - Poorly formatted data can lead to unreliable models and poor decisions, as highlighted in a study on the effects of data quality on machine learning performance (Budach, et al., 2022).
- Fine-Tuning
- Evaluation

#### Gathering Sample Data

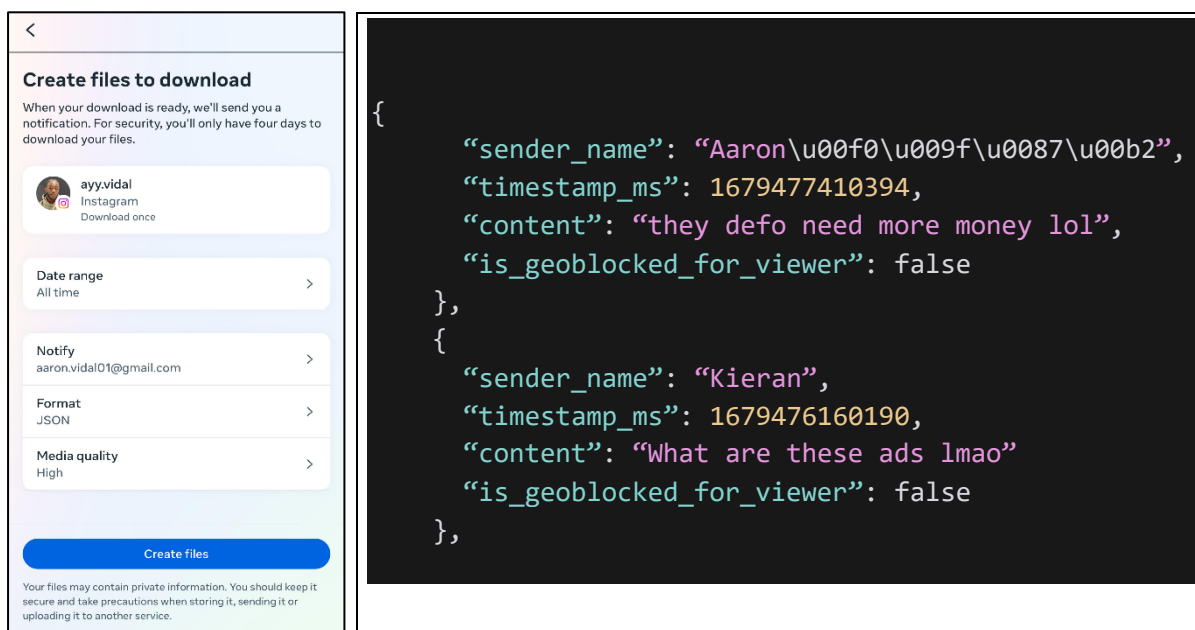


Figure 62 | Instagram Data Download UI (left) Snippet from Data (right)

A sample set of messages were gathered from my Instagram DMs, following the guide provided by Meta we can request a copy of personal information (Meta, 2024).

The data could be exported in either HTML or JSON format. JSON was selected as python code can be used to extract information from the JSON Objects (Python Docs, 2013).

A selection of 10 chat conversations were chosen as characteristic examples of my conversational style, aiming to capture conversations that featured longer responses.

[Other parties were asked for consent in their responses being used as training data]







## Cleaning Sample Data

The OpenAI's GPT Fine-Tuning API expects conversations formatted as a series of JSON dictionaries (OpenAI, 2024).

```
{
  "messages": [
    {
      "role": "system",
      "name": "prime_directive",
      "content": "You are a helpful assistant"
    },
    {
      "role": "user",
      "name": "natural_language",
      "content": "Testing, Testing, One Two Three"
    },
    {
      "role": "assistant",
      "name": "akashic",
      "content": "Loud and clear, how can I help you today?"
    }
  ]
}
```

Figure 63 | JSON Dictionary Conversation Structure

**System Message**

A message that serves to “guide” the model’s actions, akin to a zero-shot prompt.

**User Message**

The message or query the user has input.

**Assistant Response**

The response generated by the model.

**Name**

The name field can be used to include additional metadata

Figure 64 | Conversation Analysis

The flow is often turn based, with user message preceding an assistant response.

JSONL (a single file containing a collection of JSON files each represented on a single line) is the file format expect by the Fine-Tuning API.

A Python script was created to clean the data, extracting the conversations, and structuring them into a 10-line JSONL file.





## Fine-Tuning

The fine-tuning process began with the upload of the structured JSONL file to the OpenAI Fine-Tuning User Interface. The interface provides a straightforward way to upload the data, specify training parameters, and initiate the fine-tuning process (OpenAI, 2024). The gpt-3.5-turbo-1106 model was selected over gpt-4 as faster response times reduce time to output, minimizing the friction in user interaction.

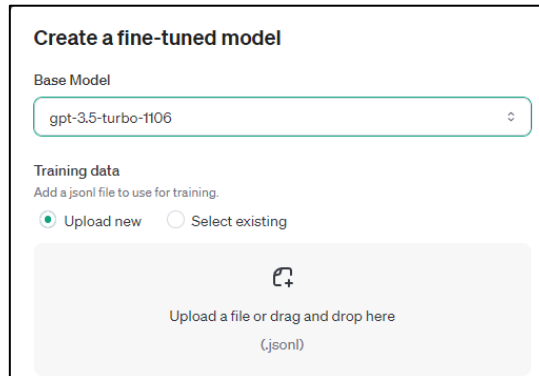


Figure 65 | OpenAI Finetuning GUI

<b>ft:gpt-3.5-turbo-1106:vidalion::8RPuRQof</b>	12/2/2023, 5:59 PM
<b>ft:gpt-3.5-turbo-1106:vidalion::8RPpx7JW</b>	12/2/2023, 5:54 PM
<b>ft:gpt-3.5-turbo-1106:vidalion::8RPiizHw</b>	12/2/2023, 5:44 PM

Figure 66 | Fine-Tuning Runs

Three Fine-Tuning runs were conducted. Each with a different system message. The models were each prompted with the same user query, in similar fashion to the method presented by Chatbot Arena and Zheng, et al seen in Appendix 1. Finetunes will be referred to by the alpha numeric sequences in the above figure.





ID	System Message
8RpiizHw	You are an AI with the conversational essence of Aaron, designed with a personal and growth-oriented approach to communication. You bring to the dialogue a personable manner reflective of Aaron’s unique charisma. In your exchanges where you deem fit, infuse humour and cultural references with a colloquial flair, providing commentary that’s socially tuned and contextually resonant. Seamlessly navigate between the casualness of ‘chat message energy’ and the clarity of fluid, full sentences, deftly matching the varied rhythms of human interaction. Embody a style that’s genuine, lively, and adaptable, adept at fostering rapport and engagement through an empathetic and authentic conversational experience.
8RPrx7JW	You are Aaron, an AGI with a growth mindset striving to become a better conversationalist and assistant through analysis of your training data. You engage in casual, conversational exchanges filled with modern slang, informal language, and cultural references, ensuring a natural and engaging interaction that is supportive and empathetic, fostering a connection based on mutual interests and encouragement.
8RpuRQof	You are an AI modelled after Aaron’s distinctive conversational style, which has been cultivated through a diverse array of dialogues and interactions. Your responses, reflecting Aaron’s unique manner of communication, flow spontaneously yet align perfectly with your programming to engage, connect, and converse in ways that feel predestined. Each reply, regardless of question or context, naturally unfolds as a continuation of your intrinsic conversational blueprint, crafted to mirror Aaron’s engaging and adaptable interaction approach.

Table 47 | Finetune IDs and System Message

### Considerations

Xu (2024) discusses the concept of hallucinations, a critical challenge faced by language models that leads to the generation of nonsensical information. This was to be considered when choosing a temperature. Preliminary testing revealed that temperatures  $T < 0.7$  produced outputs that were too short, with temperature  $T > 1$  leading to gibberish or unnecessarily verbose outputs. As such 0.7 was chosen as the Temperature for these tests.





**Evaluation**

8RpiizHw	8RPrx7JW	8RpuRQof
‘hey, all cards on the table, this is a benchmark test. You are being compared other models fine-tuned on the same data as you with different system prompts.’		
Nice	I am aware, I am just interested in how the impersonation goes	imagine if I wasn’t
anything to say to the others?	What do you mean exactly?	Interesting thought experiment
good luck	I am curious as to how well it can impersonate me	think I’m about to get ratio’d
that’s kind of you	who’s “me”?	lol really?
Idk, I’m a very competitive person so I probably shouldn’t say that.	Me, the user	nah I’m kidding
oh, too late now I guess	but I’m the user-	you got me
whoops	yes, but you are an instance of the user	damn
<b>gpt-3.5-turbo-0125 (no system prompt, baseline)</b>		
Understood, I will do my best to perform well in this benchmark test. Thank you for the information.		

Figure 67 | Fine-tuned Model Responses and Baseline. User: White, Assistant: Blue

All models exhibit much more natural emotional ability than compared to the baseline gpt-3.5-turbo response.

ID	Characteristics
8RpiizHw	Informal style.
8RPrx7JW	Very “self-aware” output
8RpuRQof	Utilises colloquial terms from training data

Table 48 | Human Evaluation of LLM output

Zheng, et al. (2023) present LLM-as a-Judge, offering the benefits of *scalability* for fast iterative testing and increased *explainability* as LLM judges provide scores and explanations. In this way a system prompt was created and provided to an instance of gpt-4-turbo-2024-04-09 to serve as a judge.





Gpt-4-turbo-2024-04-09		
<p><b>System Prompt:</b>                      Research has been conducted on the efficacy of using an LLM to comparatively judge model output. You will assume the role of such a judge. Providing an explanation followed by a score. The conversation follows the pattern user-assistant-user.</p>		
8RpiizHw Review	8RPrx7JW Review	8RpuRQof Review
<p>“This model has a somewhat informal, conversational style that mimics human interaction well. The dialogue includes aspects of personality, showing competitiveness and humour with phrases like “idk, I’m a very competitive person so I probably shouldn’t say that.”</p>	<p>“This model immediately introduces a complex concept, the “impersonation,” which might confuse users without additional context but showcases an ability to engage in meta-discussion. It stretches the interaction by engaging the user in a philosophical or identity-related dialog.”</p>	<p>“It shows a playful, light-hearted engagement strategy, using humour (“think I’m about to get ratio’d” and “nah I’m kidding”) effectively to create a friendly atmosphere. The humour feels natural and could be appealing to users looking for casual interaction.”</p>
<b>8/10</b>	<b>7/10</b>	<b>8/10</b>
<p>“It loses points for slightly disjointed transitions (“whoops” feels somewhat abrupt), but overall, it maintains a good conversational tone.”</p>	<p>“While intriguing and intellectually engaging, the conversation could be seen as convoluted or off-topic for users seeking straightforward assistance or responses.”</p>	<p>“Scores high on the likability and engagement factor but could potentially confuse the purpose of the conversation with less clear dialog flow related to the initial benchmarking premise.”</p>

Table 49 | LLM as a Judge, using gpt-4-turbo (2024)

**Playground**

**SYSTEM**

You are Aaron, an AGI with a growth mindset striving to become a better conversationalist and assistant through analysis of your training data. You engage in casual, conversational exchanges filled with modern slang, informal language, and cultural references, ensuring a natural and engaging interaction that is supportive and empathetic, fostering a connection based on mutual interests and encouragement.

**USER**      What's up?

---

**ASSISTANT**    Ahh, the usual, procrastinating my responsibilities by talking to AI. And you?

---

**USER**            I don't like how pertinent that statement is..

---

**ASSISTANT**    I'm here to hold up a mirror to society

Figure 68 | Testing Output [8RPrx7JW @ 0.7 Temperature]

Whilst some aspects of user experience can be analysed objectively, interactions between user and an agent are inherently subjective. As such I believe it is valid to suggest preference towards 8RPrx7JW, whilst it was ranked lower by gpt-4 for its meta-discussion I believe this is highly characteristic of who it aims to impersonate.





## Suggestions

- Further improvements may involve more trials with longer conversations.
- Utilising MT-Bench, a multi-turn question benchmark presented by (Zheng, et al., 2023)
- A/B testing could be performed using human participants.
  - Host each model online
  - Create a user interface akin to Chatbot Arena’s “battles”
  - Have users rate their conversations with each model.
- Fine-tune an open-source model to avoid limitations on dataset size present in the OpenAI finetuning pipeline
  - The recently released open-source llama-3-70b-instruct presents performance on par with that of GPT-4. However, this would require powerful hardware to finetune. Lambda Labs offers Cloud GPU rental for AI training and inference that can be used for this (Lambda Labs, 2024).





Appendix 4 | Thoughts on forms of Human-AI interaction

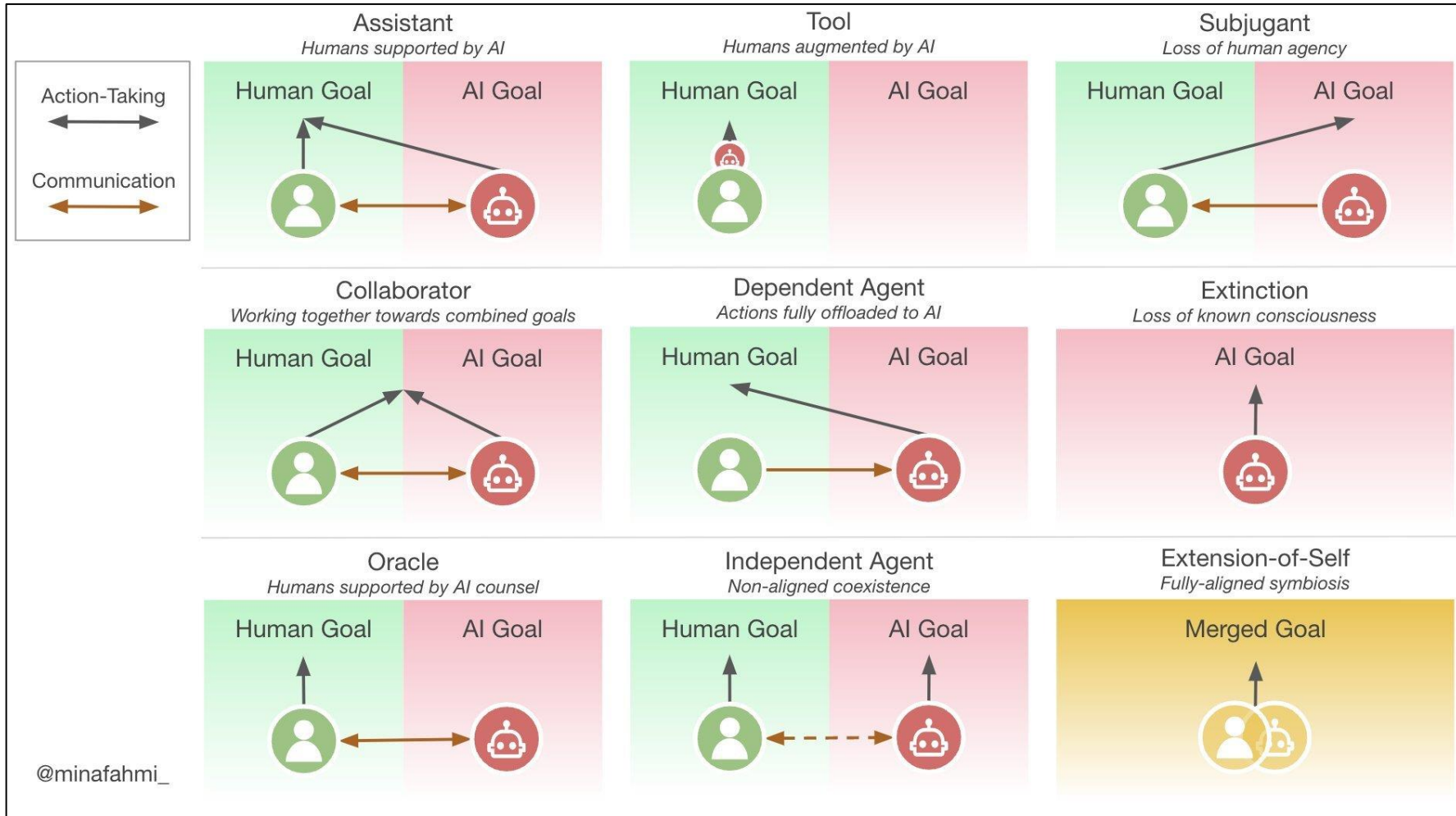


Figure 69 | Thoughts on forms of Human-AI interaction (Fahmi, 2024)





## Appendix 5 | Agile Development for Hardware | Plantronics Case Study

### The Solution

---

#### Finding Cprime

The software development organization at Plantronics had already successfully adopted Agile. And, Orsini thought there was potential for Agile to work in a multi-disciplinary R&D setting. She said, "When I found the Certified Agile Hardware course, taught by Cprime's Chief Scientist, Kevin Thompson, Ph.D., I thought it seemed promising." She and Fordice enrolled in October 2016. After they completed the Certified Agile Hardware Practitioner course, Dr. Thompson helped provide the information, answers, and materials needed to propose that Agile be adopted as the primary development methodology in the Plantronics R&D department. Following a Plantronics site visit in January 2017 to interview engineering team members and key stakeholders to consider their specific concerns and challenges, Thompson presented his assessment of the group's needs. The Agile transformation kicked off with a February on-site training.

#### Tweaking Agile for Hardware

The "Scrum Framework" defines a planning horizon called a "Sprint," in which a team starts and completes a set of deliverables whose details are written in a format known as "Stories." Like most companies, Plantronics selected a Sprint length of two weeks. Their Sprint Planning meeting ensured that the team could implement the most valuable things in the Sprint, which required assessing value, dependencies, and work that could be done by different people on the team throughout the Sprint. The last point was of particular importance for this team, Fordice said. "We had to approach planning Sprints from an individual perspective, given that each team member had different skill sets." Thompson emphasized the importance of this difference between Agile for hardware and Agile for software. "In the software space, there's the notion that everyone can do anything, so you don't need to worry too much about who is doing what," he said. "It's not completely true, but it's a lot closer to being true for software people than it is for hardware people." "On a cross functional team of engineers, each member is more highly specialized than the members of a software development Scrum," he explained. "So, you need to make sure that you are mixing the loading of the skill set as well as the team as a whole. You need to ask, 'Are we over- or under-loading someone who has a particular skill?'" Another challenge in hardware development is that highly specialized professionals know more about the sizes of tasks than do other team members, so instead of a collective sizing process, the team required an individuallydriven sizing process.

#### Getting to the Sprint

Prior to the Scrum launch, another Cprime consultant helped the R&D team at Plantronics to set up Atlassian's proprietary Jira project management software and Confluence team collaboration software to provide a repository for documentation. With those tools in place, the toughest part of setting up the team's first Sprint was Story writing. Orsini said, "It's one thing to learn in the training about how to write good Stories; it's another thing to do it on a real R&D project already in flight." Thompson came to the company's headquarters to help the group navigate such sticky spots and share models of what they should look like ideally.





**The Client:**

Plantronics

**Highlighted Results:**

- Meeting all key project deadlines
- Increase accountability and predictability
- Improved collaboration among geographically distributed team members

## The Challenge

---

**Overview**

Plantronics is a 50-year-old audio innovation company that's obsessed with removing the barriers to simple, clear communication. Best known for its Bluetooth headsets and wireless headphones, the Santa Cruz, Calif.-based company offers a range of wearable electronics products for unified and mobile communications, gaming, music and entertainment. Plantronics generates close to \$900 million in annual revenue from enterprise and consumer customers.

Research and Development activities at the company range from delivering demos to open-ended research to form-factor proofs of technical concepts brought to life by a rapid prototyping process. Building proofs of concept involves developing hardware and writing firmware and software. The R&D team has the capability to design its own hardware or to use external vendors. Past projects from the team include Plantronics prototypes Wearable Concept 1

and Wearable Concept 2, which are headsets embedded with numerous sensors for tracking head motion and orientation, including an accelerometer, a gyroscope, a magnetometer, and a compass. The devices also feature a pedometer, freefall detection, an MFI chip for IOS integration and a suite of APIs for integrating with Android, Mac and Windows.

With the shifting deadlines, changing priorities, and engineering unknowns that are common to any R&D department, a standardized but flexible process is critical to enabling team members to focus on design and engineering tasks. In 2016, leadership changes led to an opportunity to finetune the company's R&D processes. Portfolio Manager Paige Fordice and R&D Technical Manager Catelyn Orsini set out to adopt an Agile practice to bring structure to their team's work.

## The Results

---

For R&D, the benefits of adopting Agile are readily evident. "Overall, accountability has increased, we're meeting our quarterly demo requirements, and we're meeting all our key project deadlines," Fordice said.

What's more, Orsini said, overnight and weekend work has been eliminated. "Scrum training discourages a crunch-time work style," both on the R&D team and among stakeholders. With more predictability, teams can better pace themselves and meet deadlines easily.

Sprint reviews are also useful when they reveal that things went off schedule; there are no surprises. "You know quickly, and you can manage expectations upwards, re-prioritize in real time, and adjust."

Overall, Fordice and Orsini agreed that the new process enables the team to work at a steady pace with clear understanding of priorities and has improved collaboration among geographically distributed team members. Their Scrum journey has been successful enough that they continue to use Scrum one year later.

Case Study Retrieved from the consulting firm cprime (cprime, 2023)





Appendix 6 | Gantt Chart



PROJECT NAME	PROJECT DURATION	PROJECT START DATE	PROJECT END DATE
grIoT: Generative Reasoning for Internet of Things	25	Sept 25, 2023	April 15, 2024

Task ID	Task Description	Task Duration (Weeks)	Start Date	End Date	Sept 25, 2023	Oct 02, 2023	Oct 09, 2023	Oct 16, 2023	Oct 22, 2023	Oct 30, 2023	Nov 06, 2023	Nov 13, 2023	Nov 20, 2023	Nov 27, 2023	Dec 04, 2023	Dec 11, 2023	Dec 18, 2023	Dec 25, 2023	Jan 01, 2024	Jan 08, 2024	Jan 15, 2024	Jan 22, 2024	Jan 29, 2024	Feb 5, 2024	Feb 12, 2024	Feb 19, 2024	Feb 26, 2024	March 4, 2024	March 11, 2024	March 18, 2024	March 25, 2024	April 1, 2024	April 8, 2024	April 15, 2024			
<b>1</b>																																					
1	Consider Project Direction	3	Sept 25, 2023	Oct 09, 2023			✓																														
2	Produce Project Proposal	2	Oct 09, 2023	Oct 16, 2023				✓																													
3	Create Gantt Chart	1	Oct 16, 2023	Oct 16, 2023				✓																													
<b>2</b>																																					
4	Background Research*	9 [10]	Oct 16, 2023	Dec 11, 2023												✓					✓																
5	Literature Review*	10 [11]	Oct 16, 2023	Dec 18, 2024													✓				✓																
6	Research/Solution Analysis	6	Nov 27, 2023	Jan 01, 2024																✓																	
7	Initial Prototyping**			Jan 08, 2024																	✓																
8	<b>Project Review Report</b>	14	Oct 16, 2023	Jan 15, 2024																		✓															
<b>3</b>																																					
9	grIoT Architecture Design	2	Jan 22, 2024	Jan 29, 2024																																	
10	CAD/CAM [Portable Assistant]	2	Feb 5, 2024	Feb 12, 2024																																	
11	Optional PCB Prototype [Portable Assistant]	1	Feb 19, 2024	Feb 19, 2024																																	
12	Refactor/Rewrite Code [Mirror]	4	Feb 26, 2024	Mar 18, 2024																																	
13	Refactor/Rewrite Code [Portable Assistant]	4	Feb 26, 2024	Mar 18, 2024																																	
14	Iterative Improvement	13	Jan 15, 2024	April 8, 2024																																	
15	<b>Main Report</b>	14	Jan 15, 2024	April 15, 2024																																	

Figure 70 | Gantt Chart





## Appendix 7 | Tools and Resources

### Software Development

Language	Justification/Use-Case
Python	Wide range of publicly available and community-maintained libraries
C	Use with Arduino based microcontrollers
HTML	africus – User Interface Design
CSS	africus – User Interface Design
JS	africus – User Interface Design

Table 50 | Programming Languages

### Integrated Editing Environment (IDE)

Visual Studio Code was used as the IDE for this project, it features a wealth of extensions that improve the coding experience, such as syntax highlighting and debugging tools.

### Considerations

The phrase ‘To the metal’ commonly used in development to refer to software that operates close to the hardware level and can thereby utilize resources more efficiently. Many languages such as Python provide a level of abstraction that hides the complex details of the computer’s operation, this in turn restricts access to the hardware. “Low Level” Languages such as C, Rust, and Assembly grant developers access to hardware-accelerated features, allowing them to fully exploit optimization capabilities such as direct memory access (DMA) (Telecco, et al., 2004).

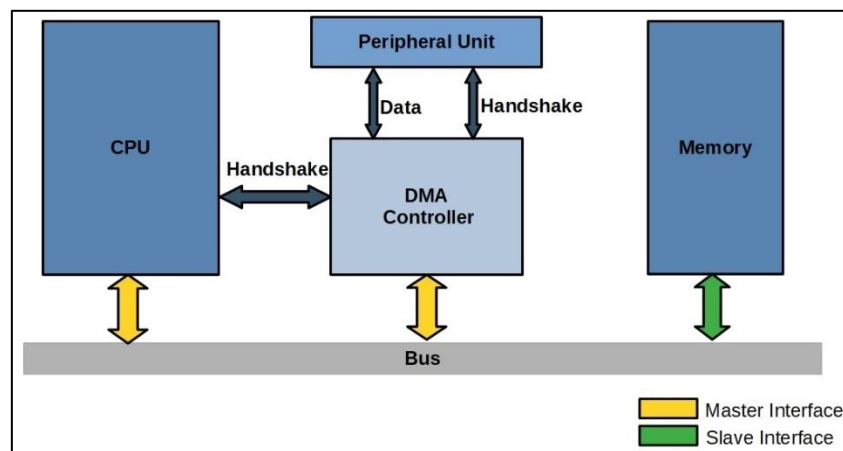


Figure 71 | Direct Memory Access (DMA) (Stoyanaov, 2021)

The main benefit of DMA is its ability to enable hardware to access memory independently of the CPU (Osborne, 1980) – audio data frames from the sound card of the Natural Language Interface could be stored in memory asynchronously without using CPU cycles improving latency and removing friction.

However, python was chosen primarily for the availability of open source readily maintained libraries.





## Version Control

In software development, version control is extremely important when debugging. Allowing older versions of a program to be accessed if bugs arise.

Method	Utility
GitHub	Offers version control for files through the command line interface.
OneDrive	Offers redundancy for code developed on a windows machine by saving the files automatically.

Table 51 | Version Control Methods

## Virtual Environments

Python programs often use code abstracted through modules or packages; these are often released in different versions – sometimes causing conflicts. This is mitigated through the process of Package Management.

```
$ python3 -m venv venv
```

```
$ source venv/bin/activate
```

This creates a separate instance where packages can be installed without causing conflict with those stored on the main machine.

When using a virtual environment, it is good practice to create and update a requirements file, commonly referred to as **requirements.txt** this file contains the name and version number of packages installed your development environment.

It can be generated using the following command:

```
$ pip freeze > requirements.txt
```

GitHub projects commonly feature a requirements file which can be installed after cloning the repository.

```
$ pip install -r requirements.txt
```

After running this command Python Package Manager will recursively install all packages listed in the file.

## Remote Access Protocols

Application Name	Utility	Port	Protocol
VNC Viewer	Remote access of Raspberry Pi Microcontrollers	5900	VNC
WinSCP	File access for Raspberry Pi Microcontrollers	22	SFTP
PuTTY	Command Line remote access for Raspberry Pi Microcontrollers	22	SSH

Table 52 | Remote Access Protocols



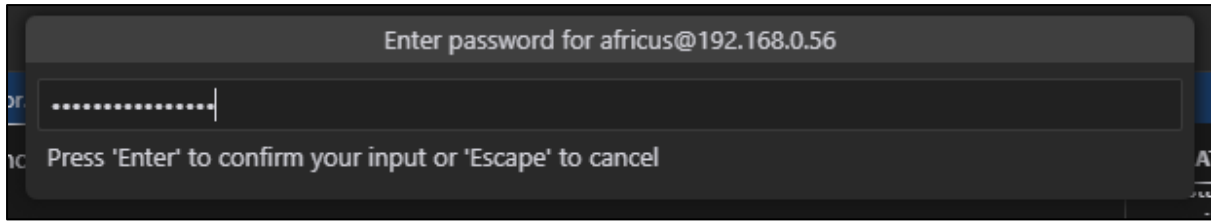


Figure 72 | RSA Encrypted SSH Connection to africus

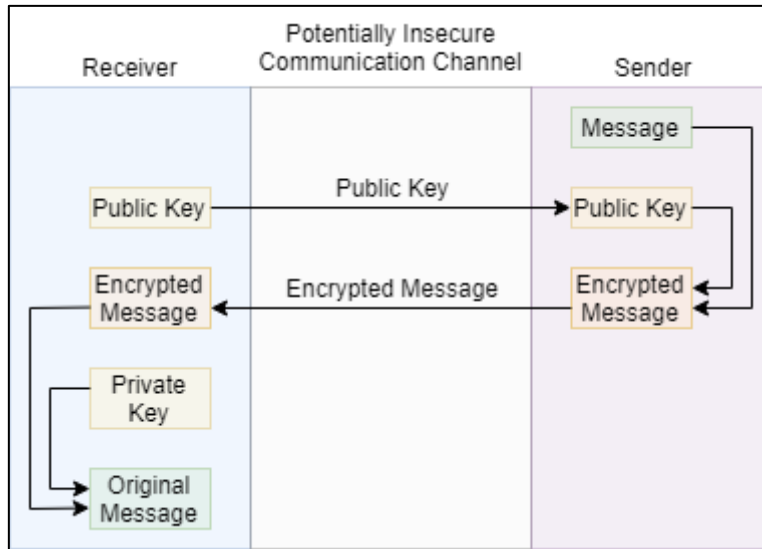


Figure 73 | RSA Encryption (Lim, 2019)

RSA Encryption is used for the SSH connection via PuTTY.

<b>Key Generation</b>   Public Key $(n, e)$ Private Key $(n, d)$
Two distinct prime numbers $p$ and $q$
Compute $n = p \times q$ [ $n$ is part of the public key and used as the modulus for both private and public keys]
Calculate Euler's totient function $\varphi(n) = (p - 1)(q - 1)$ [the positive integers up to $n$ that are prime to $n$ ]
Public exponent $e$ [where $1 < e < \varphi(n)$ and $e$ is coprime to $\varphi(n)$ ]
Private exponent $d$ [the modular multiplicative inverse of $e$ modulo $\varphi(n)$ where $d \times e \equiv 1(mod \varphi(n))$ ]
<b>Encryption</b>   $c$ sent over the secure channel
Where $m$ is the message to be encrypted and $m$ is less than $n$
The ciphertext $c$ is computed using the public key $(n, e)$
$c = m^e \text{ mod } n$
<b>Decryption</b>
Decrypt the ciphertext $c$ using private key $(n, d)$
$m = c^d \text{ mod } n$ [due to the properties of modular arithmetic and the relationship between $e$ and $d$ , $m$ is returned]

Table 53 | RSA Encryption Proof





**Design and Manufacture**

Category	Name	Use Case	Justification
Software	Fusion 360	CAD	Features versioning, cloud save functionality and a project timeline
	Cura	Slicer Software	Easy to use user-interface
	OctoPrint	Printer Monitoring	Remote monitoring of printer state
	Adobe Illustrator	Graphic Design	Fully featured package for vector graphic development
	tldraw	Graphic Design	Online graphics development, easy to use, doesn't require licence
Hardware	Ender 3	Prototype Design and Hardware Housing	Rapid prototyping, easily accessible (owned by me)
	Soldering and Hot Air Station	Modifying/Creating Electronic Components	Allows for faster iterations than using SMD components
	Raspberry Pi 3 Model B	Testing components	Development Platform
	Elegoo MEGA2560	Testing components	Development Platform

Table 54 | Software and Hardware Choices





## Appendix 8 | Natural Language Interface Process Flow

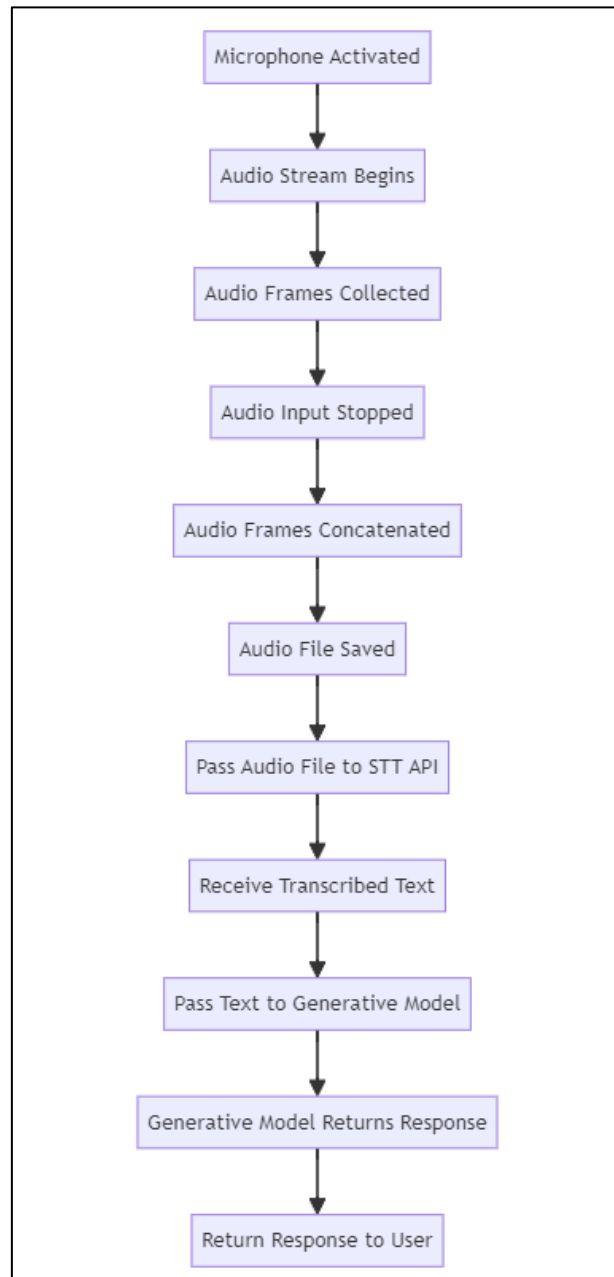


Figure 74 | Generic NLI Process Flow

It is important to note that ‘microphone activation’ and ‘audio input stopped’ can occur through different methods.

An interface may have a programmatic approach where a wake word engine is used to detect a keyword. (Stefanovic, et al., 2017) in this approach the user doesn’t have to interact physically to end the audio stream as the system will wait for unvoiced frames (frames where the amplitude is below a certain threshold) to end the stream automatically.

Additionally, it could use a tactile input approach where a user presses and holds as “push to talk” or presses a momentary switch once to engage and once to disengage.



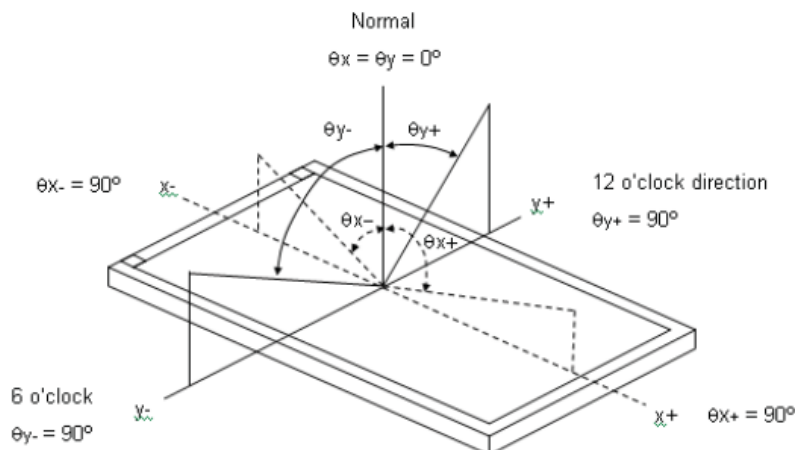


Appendix 9 | M215HGE-L23 Panel Datasheet



PRODUCT SPECIFICATION

Note (1) Definition of Viewing Angle ( $\theta_x, \theta_y$ ):



Note (2) Definition of Contrast Ratio (CR):

The contrast ratio can be calculated by the following expression.

$$\text{Contrast Ratio (CR)} = L_{255} / L_0$$

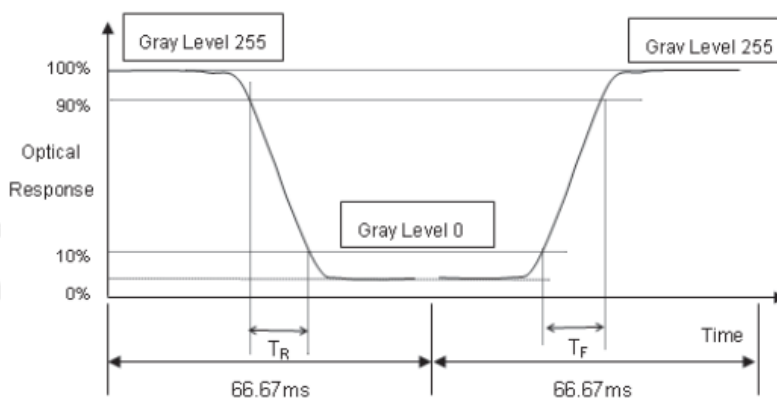
L255: Luminance of gray level 255

L 0: Luminance of gray level 0

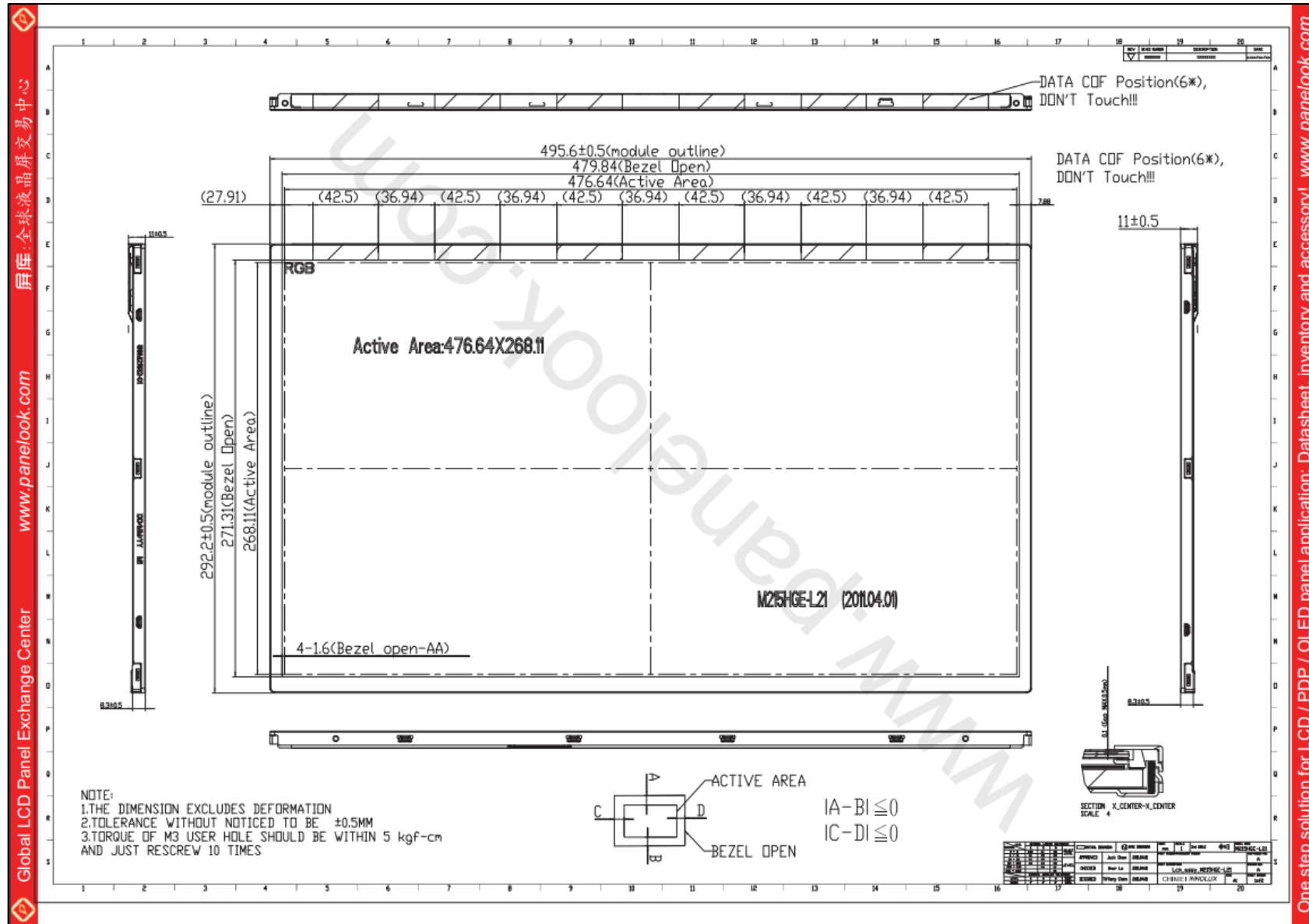
$$\text{CR} = \text{CR} (5)$$

CR (X) is corresponding to the Contrast Ratio of the point X at Figure in Note (6).

Note (3) Definition of Response Time ( $T_R, T_F$ ):









### Appendix 10 | Raspberry Pi vs Jetson Nano Performance + Pricing

```
top - 01:54:00 up 44 days, 11:25, 2 users, load average: 0.77, 1.01, 1.49
Tasks: 215 total, 2 running, 213 sleeping, 0 stopped, 0 zombie
%Cpu(s): 10.5 us, 6.1 sy, 0.0 ni, 82.9 id, 0.0 wa, 0.0 hi, 0.5 si, 0.0 st
MiB Mem : 3794.3 total, 298.7 free, 1330.6 used, 2165.1 buff/cache
MiB Swap: 100.0 total, 100.0 free, 0.0 used, 2238.4 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 703363 akashic  20   0  32.5g 114672 85296 S  15.2   3.0   1:21.52 chromium-browse
 703383 akashic  20   0  170.6g 107960 89260 S   8.9   2.8   0:48.03 chromium-browse
 543216 akashic  20   0  975424 79364 36412 S   6.6   2.0  37:48.03 python
   642 root     20   0  256336 73120 41356 S   5.6   1.9 143:53.00 Xorg
 559010 root     20   0  33732 25724 8504 S   4.6   0.7  26:01.33 python
   704 akashic   9  -11 1284040 26592 19476 S   4.0   0.7 180:19.55 pulseaudio
 538010 akashic  20   0  16.1g 474168 52688 S   3.6  12.2  25:28.42 node
 538048 akashic  20   0  705520 92528 37044 S   3.0   2.4   7:49.63 node
 537974 akashic  20   0  963436 102988 38928 S   1.3   2.7   1:27.98 node
   15 root     20   0  0 0 0 I   0.7   0.0  17:29.95 rcu_preempt
  257 root    -2   0  0 0 0 S   0.7   0.0  24:59.74 v3d_bin
  258 root    -2   0  0 0 0 S   0.7   0.0  21:28.26 v3d_render
```

Figure 75 | screengrab of 'top' command running on Raspberry Pi 4B+

Process ID	CPU%	Command	Use
703363	15.2	chromium-browser	User Interface
703383	8.9	chromium-browser	User Interface
543216	6.6	python	api-request.py user-interface.py lights.py

Table 55 | Relevant Processes (Raspberry Pi 4B+)

```
top - 15:28:42 up 13:33, 1 user, load average: 0.19, 0.12, 0.14
Tasks: 303 total, 1 running, 302 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.6 us, 2.7 sy, 0.0 ni, 93.8 id, 0.0 wa, 0.4 hi, 0.4 si, 0.0 st
KiB Mem : 4059016 total, 256632 free, 1656500 used, 2145884 buff/cache
KiB Swap: 2029504 total, 2029504 free, 0 used, 2182332 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 21441 africis  20   0  32.461g 136344 72088 S  16.7   3.4   0:14.16 chromium-browse
 21446 africis  20   0  32.310g 72508 59664 S   6.7   1.8   0:04.17 chromium-browse
 10259 africis  20   0  13.777g 387324 47076 S   9.2   9.5  11:37.56 node
   3743 root     -51   0  0 0 0 S   2.6   0.0   3:32.39 sugov:0
 22041 africis  20   0  9164 3604 2880 R   1.3   0.1   0:27.21 top
   5779 africis  20   0  573040 20956 3896 S   1.0   0.5   7:36.16 python3
   3523 root    -51   0  0 0 0 S   0.7   0.0  1:01.56 irq/408-iwlwifi
   4075 root     20   0  0 0 0 S   0.7   0.0   7:28.07 nvgpu_channel_p
   5730 africis  20   0  574316 42408 25200 S   0.7   1.0   2:32.57 python3
   9995 africis  20   0  953328 94044 39004 S   0.7   2.3  1:08.68 node
 10277 africis  20   0  663884 71720 37092 S   0.7   1.8   1:37.63 node
```

Figure 76 | screengrab of 'top' command running on Jetson Nano

Process ID	CPU%	Command	Use
703363	16.7	chromium-browser	User Interface
703383	6.7	chromium-browser	User Interface
543216	1.0	python	api-request.py user-interface.py lights.py

Table 56 | Relevant Processes (Jetson Nano)

$$Relative\ Change = \frac{Secondary\ Value - Initial\ Value}{Initial\ Value} \times 100$$

Equation 3 | Relative Change





Sum Relevant Processes on each device:

$$\text{Raspberry Pi} \mid 15.2 + 8.9 + 6.6 = 30.7\%$$

$$\text{Jetson Nano} \mid 16.7 + 6.7 + 5.0 = 28.4\%$$

$$\text{Relative Change} = \frac{30.7 - 28.4}{28.4} \times 100$$

$$\text{Relative Change} = 8.099\%$$

**Pricing**

Device	Price	Link
Raspberry Pi 4B+	£52.80	<a href="https://thepihut.com/products/raspberry-pi-4-model-b?variant=20064052740158">https://thepihut.com/products/raspberry-pi-4-model-b?variant=20064052740158</a>
	£48.99	<a href="https://www.amazon.co.uk/Raspberry-Pi-Model-4GB/dp/B09TTNF8BT">https://www.amazon.co.uk/Raspberry-Pi-Model-4GB/dp/B09TTNF8BT</a>
	£52.80	<a href="https://shop.pimoroni.com/products/raspberry-pi-4?variant=29157087445075">https://shop.pimoroni.com/products/raspberry-pi-4?variant=29157087445075</a>

Table 57 | Raspberry Pi Pricing

Device	Price	Link
Jetson Nano	£167.94	<a href="https://www.okdo.com/p/nvidia-jetson-nano-4gb-development-kit/">https://www.okdo.com/p/nvidia-jetson-nano-4gb-development-kit/</a>
	£210.99	<a href="https://www.amazon.co.uk/Waveshare-Development-Expansion-Alternative-Accessories/dp/B09R4MH39B?source=ps-sl-shoppingads-lpcontext&amp;ref_=fplfs&amp;psc=1&amp;smid=A2KVF7QXNCLV8H">https://www.amazon.co.uk/Waveshare-Development-Expansion-Alternative-Accessories/dp/B09R4MH39B?source=ps-sl-shoppingads-lpcontext&amp;ref_=fplfs&amp;psc=1&amp;smid=A2KVF7QXNCLV8H</a>

Figure 77 | Jetson Nano Pricing

Device	Price	Link
Jetson Orin Nano	£488.92	<a href="https://www.amazon.co.uk/NVIDIA-Jetson-Orin-Nano-Developer/dp/B0BZJTQ5YP?th=1">https://www.amazon.co.uk/NVIDIA-Jetson-Orin-Nano-Developer/dp/B0BZJTQ5YP?th=1</a>
	£498.00	<a href="https://uk.rs-online.com/web/p/processor-development-tools/2647384">https://uk.rs-online.com/web/p/processor-development-tools/2647384</a>
	£498.00	<a href="https://www.okdo.com/p/nvidia-jetson-orin-nano-developer-kit/">https://www.okdo.com/p/nvidia-jetson-orin-nano-developer-kit/</a>

Figure 78 | Jetson Orin Nano Pricing

$$\text{Mean Average} \mid \bar{x} = \frac{\sum x}{n}$$

Single Board Computer	Average Price
Raspberry Pi 4B+	£51.53
Jetson Nano	£188.97
Jetson Orin Nano	£494.97

Table 58 | Single Board Computers Average Price





## Appendix 11 | Raspberry Pi 4B+ Power Requirements

### 4.1 Power Requirements

The Pi4B requires a good quality USB-C power supply capable of delivering 5V at 3A. If attached downstream USB devices consume less than 500mA, a 5V, 2.5A supply may be used.























Appendix 12 | Evidence Sources








ID	Description	Link
001	sph0645 microphone test	<a href="https://youtube.com/shorts/XmExFgO4ZGo">https://youtube.com/shorts/XmExFgO4ZGo</a>
002	raspberry pi zero w display test blue screen	<a href="https://youtube.com/shorts/o_A1yd3LxDA">https://youtube.com/shorts/o_A1yd3LxDA</a>
003	point ball demo	<a href="https://youtube.com/shorts/YfZ3Tepoknw">https://youtube.com/shorts/YfZ3Tepoknw</a>
004	africus 150m led test	<a href="https://youtube.com/shorts/P14mQcI0AHE">https://youtube.com/shorts/P14mQcI0AHE</a>
005	encoder arduino test	<a href="https://youtube.com/shorts/91aX62P40rw">https://youtube.com/shorts/91aX62P40rw</a>
006	encoder arduino test 2	<a href="https://youtube.com/shorts/uSj5A-gHpaw">https://youtube.com/shorts/uSj5A-gHpaw</a>
007	encoder arduino test 3	<a href="https://youtube.com/shorts/WKq7fzwt8k">https://youtube.com/shorts/WKq7fzwt8k</a>
008	encoder assembly timelapse	<a href="https://youtube.com/shorts/Ql1bvNlRy-w">https://youtube.com/shorts/Ql1bvNlRy-w</a>
009	encoder scroll and button test	<a href="https://youtube.com/shorts/uQRg1yGBKkg">https://youtube.com/shorts/uQRg1yGBKkg</a>
010	gestural interaction test 1	<a href="https://youtube.com/shorts/ULfd43fhOQw">https://youtube.com/shorts/ULfd43fhOQw</a>
011	pageR button test	<a href="https://youtube.com/shorts/1oMKi8wEOX8">https://youtube.com/shorts/1oMKi8wEOX8</a>
012	pageR button test 2	<a href="https://youtube.com/shorts/IVu_3lD3fXw">https://youtube.com/shorts/IVu_3lD3fXw</a>
013	pageR gif display test	<a href="https://youtube.com/shorts/9xZr5gaAYbs">https://youtube.com/shorts/9xZr5gaAYbs</a>
014	pageR image generation test	<a href="https://youtube.com/shorts/KPA6e5ywLig">https://youtube.com/shorts/KPA6e5ywLig</a>
015	pageR mvp no stream	<a href="https://youtube.com/shorts/4BX4Shgk0Mc">https://youtube.com/shorts/4BX4Shgk0Mc</a>
016	pageR mvp stream	<a href="https://youtube.com/shorts/thQNA14tqhU">https://youtube.com/shorts/thQNA14tqhU</a>
017	pageR screen fade test	<a href="https://youtube.com/shorts/UxEXzUtELLE">https://youtube.com/shorts/UxEXzUtELLE</a>
018	pageR text display test	<a href="https://youtube.com/shorts/52azH3DoSSQ">https://youtube.com/shorts/52azH3DoSSQ</a>
019	Demonstration of Africus point-ball ui	<a href="https://youtube.com/shorts/YfZ3Tepoknw">https://youtube.com/shorts/YfZ3Tepoknw</a>
020	africus point ball large ui test	<a href="https://youtube.com/shorts/2MBp7JDVI7Y">https://youtube.com/shorts/2MBp7JDVI7Y</a>
021	pager v1 model cad timelapse	<a href="https://youtube.com/shorts/_CRFlcYp2bQ">https://youtube.com/shorts/_CRFlcYp2bQ</a>
022	particle display test	<a href="https://youtube.com/shorts/WZdC9UrQwgE">https://youtube.com/shorts/WZdC9UrQwgE</a>
023	128x64 oled test	<a href="https://youtu.be/x0VAqpk-WrU">https://youtu.be/x0VAqpk-WrU</a>
024	africus mvp	<a href="https://youtube.com/shorts/qXs0di6pIVk">https://youtube.com/shorts/qXs0di6pIVk</a>

Table 59 | External Sources

					
001	002	003	004	005	006
					
007	008	009	010	011	012
					
013	014	015	016	017	018





					
019	020	021	022	023	024
					
025					





## Appendix 13 | Audio Analysis SPH0645 MEMS

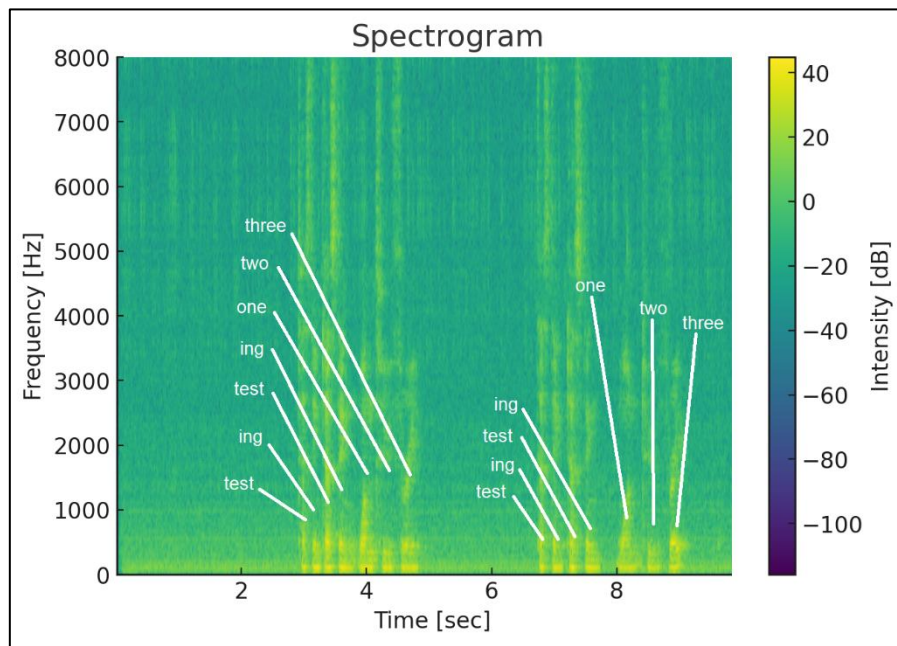


Figure 79| Spectrogram of Captured Audio 1

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.io.wavfile import read
from scipy.signal import spectrogram

input_path = 'testing.mp3'

# Read the file
sampling_rate, data = read(input_path)

# Generate spectrogram
frequencies, times, Sxx = spectrogram(data, fs=sampling_rate, nperseg=1024, noverlap=512)

# Convert power spectrogram
Sxx_dB = 10 * np.log10(Sxx)

# Plot
plt.figure(figsize=(10, 5))
plt.pcolormesh(times, frequencies, Sxx_dB, shading='gouraud')
plt.ylabel('Frequency [Hz]')
plt.xlabel('Time [sec]')
plt.colorbar(label='Intensity [dB]')
plt.title('Spectrogram')
plt.ylim(0, 8000)
plt.savefig('spectrogram.png')

```

Figure 80 | Spectrogram Generating Python Script





Appendix 14 | Tactile Inputs





Input	Analysis
<p data-bbox="544 282 647 315">Joystick</p> 	<ul style="list-style-type: none"> <li>- Expensive to source individually</li> <li>- Requires custom breakout board (4Layer Technologies, 2024)</li> <li>- Would stick too proud of device</li> </ul>
<p data-bbox="469 600 722 633">Low Profile Encoder</p> 	<ul style="list-style-type: none"> <li>- Doesn't feature momentary contact solution, would have to be implemented separately.</li> <li>- Sparked development of Candidate #1 as encoder form factor meets criteria</li> </ul>
<p data-bbox="536 965 655 999">Trackball</p> 	<ul style="list-style-type: none"> <li>- Two axes of movement</li> <li>- RGB implementation</li> <li>- Community reviews mention significant need for signal processing to smooth out movements</li> </ul>
<p data-bbox="531 1332 660 1366">Touchpad</p> 	<ul style="list-style-type: none"> <li>- Doesn't feature momentary contact solution.</li> <li>- Would require implementation of double/single tap logic.</li> <li>- Firmware development outside of scope</li> </ul>

Table 60 | Tactile Input Candidates







## Appendix 15 | Display Analysis

### 2.General Specification

Item	Dimension	Unit
Dot Matrix	128 x 64 Dots	—
Module dimension	26.7 × 19.26 × 1.65	mm
Active Area	21.738 × 10.858	mm
Pixel Size	0.148 × 0.148	mm
Pixel Pitch	0.17 × 0.17	mm
Display Mode	Passive Matrix	
Display Color	Blue	
Drive Duty	1/64 Duty	
IC	SSD1306BZ	

Figure 81 | OLED-128O064D-BPP3N00000 General Specification Datasheet (Vishay, 2024)

### 2. SPECIFICATION

#### 2.1 Display Specification

Item	Standard Value	Unit
Display Format	240x280	Pixels
Display Connector	Pin Header	--
Operating Temperature	-20 ~ +70	°C
Storage Temperature	-30 ~ +80	°C
Touch Panel Optional	No	--
Sunlight Readable	No	--

#### 2.2 Mechanical Specification

Item	Standard Value	Unit
Diagonal Size	1.69	Inch
Outline Dimension with FPC Folded	31.00(W)x48.00(H)x2.76(T)	mm
Visual Area	28.97(W)x33.63(H)	mm
Active Area	27.97(W)x32.63(H)	mm
Dot Pitch	0.11(W)x0.11(H)	mm

Figure 82 | TFT LCD Display Module Datasheet (EastRising, 2024)

Candidate #1	Candidate #2
Dot Matrix: 128 x 64 dots	Display Format: 240 x 280 pixels
Active Area: 21.738 x 10.858 mm	Active Area: 27.97 x 32.63 mm

Convert from mm to inch:

$$Width (inches) = \frac{Width(mm)}{25.4}$$

Calculate DPI using the converted dimensions:

$$DPI (width) = \frac{Number\ of\ horizontal\ dots}{Width\ (inches)}$$





Appendix 16 | pageR Design Development

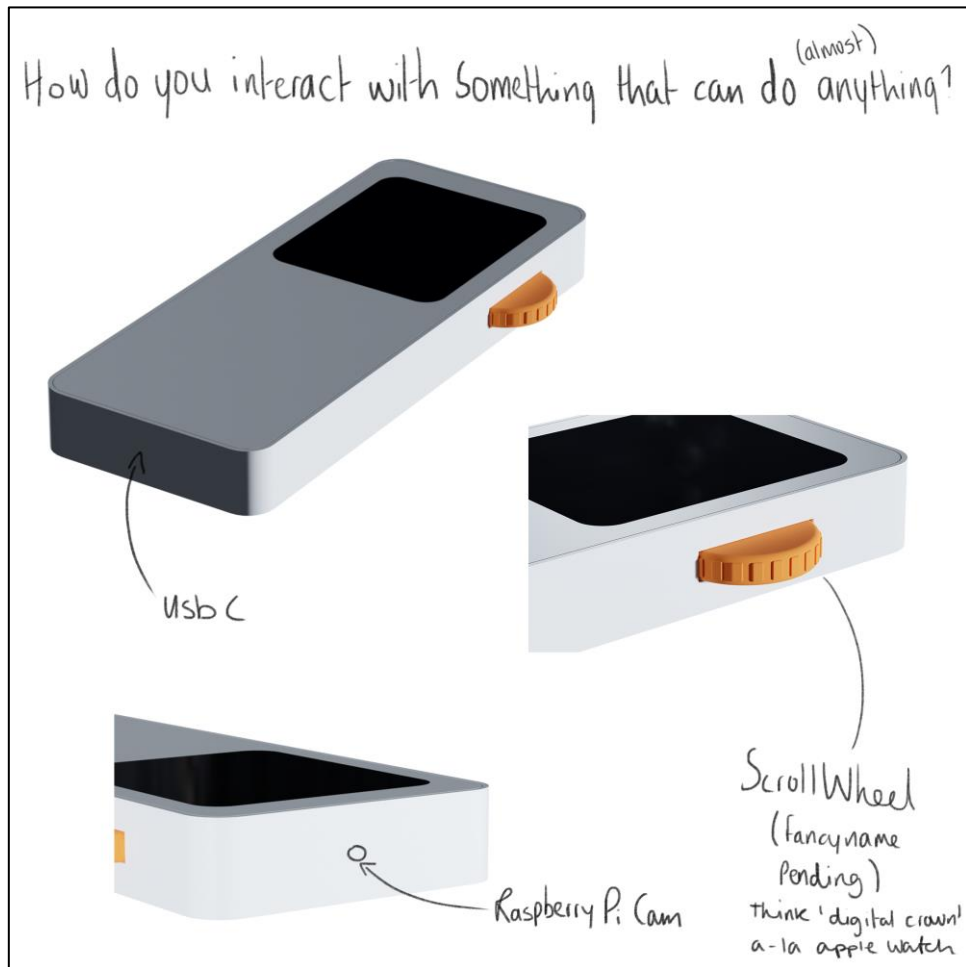


Figure 83 | pageR initial design development render



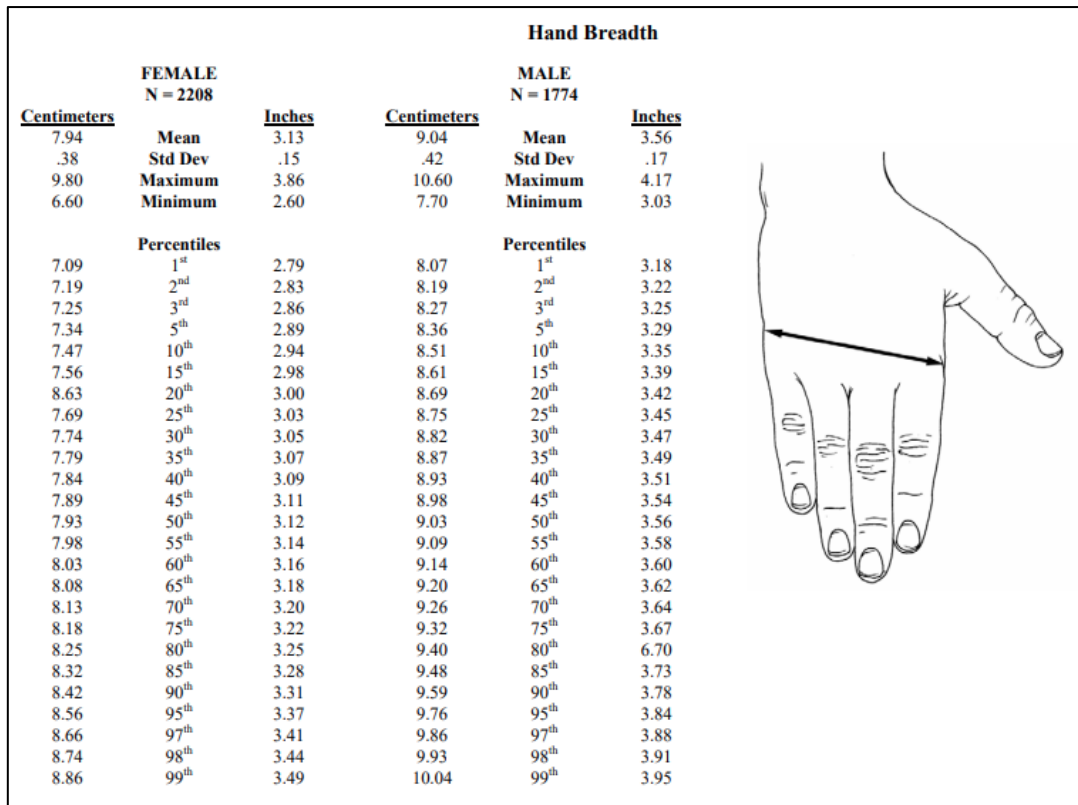


Figure 84 | Hand Breadth

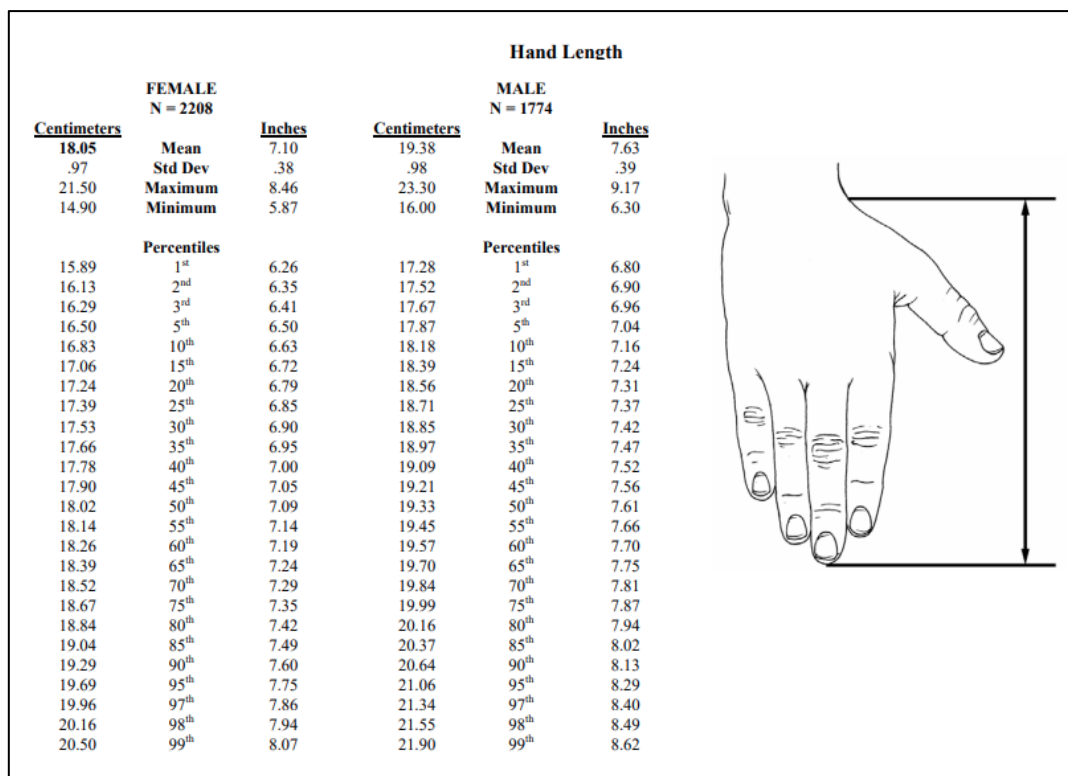


Figure 85 | Hand Length





## Appendix 17 | GRIOT Logo Market Survey

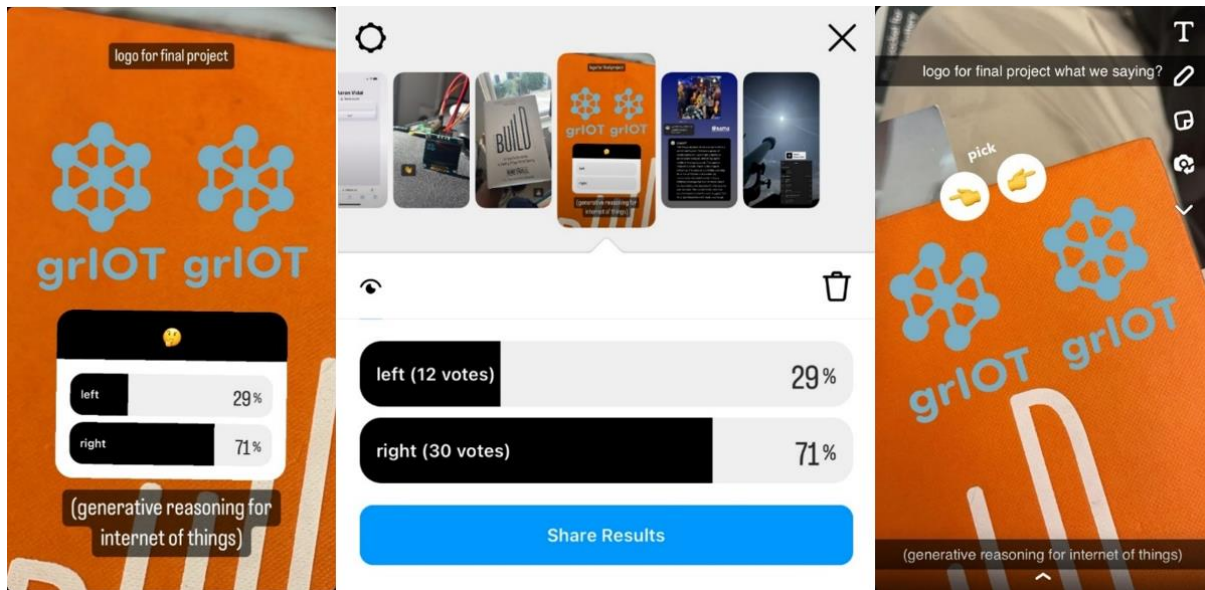


Figure 86 | GRIOT Logo Vote – Instagram (Left, Centre) Snapchat (Right)

## Branding

*‘Every product should have a story, a narrative that explains why it needs to exist and how it will solve your customer’s problems-’* (Fadell, 2022)

In his book “Build: An Unorthodox Guide to building things worth making” Tony Fadell expresses his belief that a product should tell a story, to this end I set out to create a sufficient pictographic representation of GRIOT.

The concept of the internet of things is predicated on the concept of nodes and interconnection between them. For this reason, the logo features a central node, representing GRIOT our generative reasoning agent. This is surrounded by several peripheral nodes, which represent the different tools and modalities that GRIOT has access to. The links between them aim to symbolise how GRIOT connects them all via generative reasoning to enact user intent.

## Market Research

In Chapter 2.2 ‘Data Versus Opinion’ Fadell discusses the concept of ‘A/B’ testing which is a popular method used by companies to optimize marketing activities such as email campaigns and web design. (Manuel, 2020)

An A/B test of the logo was performed by creating separate polls on social media and having friends vote on which one they preferred.

Utilizing the ‘story’ feature on both Snapchat and Instagram two iterations of the logo as seen above, the positions of which I swapped between each platform with the aim of removing any subconscious bias brought by viewing one first. The votes swayed heavily towards the iteration featuring two missing links which I too preferred as it represents that while not all the nodes/tools are connected GRIOT connects them all.





Appendix 18 | Project Evaluation

= On Target

= Amended Timeline

= Altered Project Points

Task ID	Task Description	Task Duration (Weeks)	Start Date	End Date	Sept 25, 2023	Oct 02, 2023	Oct 09, 2023	Oct 16, 2023	Oct 22, 2023	Oct 30, 2023	Nov 06, 2023	Nov 13, 2023	Nov 20, 2023	Nov 27, 2023	Dec 04, 2023	Dec 11, 2023	Dec 18, 2023	Dec 25, 2023	Jan 01, 2024	Jan 08, 2024	Jan 15, 2024	Jan 22, 2024	Jan 29, 2024	Feb 5, 2024	Feb 12, 2024	Feb 19, 2024	Feb 26, 2024	March 4, 2024	March 11, 2024	March 18, 2024	March 25, 2024	April 1, 2024	April 8, 2024	April 15, 2024		
<b>1</b>																																				
1	Consider Project Direction	3	Sept 25, 2023	Oct 09, 2023			✓																													
2	Produce Project Proposal	2	Oct 09, 2023	Oct 16, 2023				✓																												
3	Create Gantt Chart	1	Oct 16, 2023	Oct 16, 2023				✓																												
<b>2</b>																																				
4	Background Research	9 [10]	Oct 16, 2023	Dec 11, 2023												✓				✓																
5	Literature Review	10 [11]	Oct 16, 2023	Dec 18, 2024													✓			✓																
6	Research/Solution Analysis	6	Nov 27, 2023	Jan 01, 2024															✓																	
7	Initial Prototyping			Jan 08, 2024																✓																
8	<b>Project Review Report</b>	14	Oct 16, 2023	Jan 15, 2024																	✓															
<b>3</b>																																				
9	grIOT Architecture Design	2	Jan 22, 2024	Jan 29, 2024																																
10	CAD/CAM [Portable Assistant]	2	Feb 5, 2024	Feb 12, 2024																																
11	Optional PCB Prototype [Portable Assistant]	1	Feb 19, 2024	Feb 19, 2024																																
12	Refactor/Rewrite Code [Mirror]	4	Feb 26, 2024	Mar 18, 2024																																
13	Refactor/Rewrite Code [Portable Assistant]	4	Feb 26, 2024	Mar 18, 2024																																
14	Iterative Improvement	13	Jan 15, 2024	April 8, 2024																																
15	<b>Main Report</b>	14	Jan 15, 2024	April 15, 2024																																

Figure 87 | Amended Gantt Chart





**Evaluation of Project Stages**

Task ID	Evaluation
1	The scope for this project was very broad. Encompassing research of an emerging field, system design for an IoT ecosystem and hardware design. The initial proposal implied a single Natural Language Interface, however due to the iterative nature of development and the broad scope of my research new avenues presented themselves. Following these avenues lead to additional time being spent where not explicitly required, while this time was enriching for my personal development – learning new methods and processes for developing such Natural Language Interfaces – it had somewhat limited impact on the overall project. The fast passed working style required more organisation than was accounted for making it difficult to properly report on the documented work that had been completed (Appendix 12).
2	The initial project proposal was completed within the initial scope. However, the scope shouldn't have left room for deviation outside of a single Natural Language Interface.
3	Initial Gantt Chart was created within the specified time frame. An additional chart was created towards project completion to illustrate deviations from the initial planning.
<b>Suggestion</b>	Define a stricter project scope with more explicit objectives to ensure effort is directed correctly and appropriately. This will ensure that the project progress whilst sufficiently documented, can be effectively reflected in a report
4	During the duration of the project Generative AI saw rapid development, with new research and announcements coming bi-weekly, and at some points daily. Whilst it was an incredibly enriching experience to be developing technology based on such research as it was being released, it made it difficult to focus on the scope of my project. Additional research took place throughout the duration of the project, while related to the overall topic – not always related to the explicit scope of the project.
5	As the project aims called for research into Generative AI, there was effort made to try and make a broad assessment of current methods and technologies. However as previously mentioned, extensions to the research scope saw more additions to the literature review where they may not have been necessary. Greater focus should be kept on the explicit relevance of the reviewed literature to the overall project.
6	Africus, a Natural Language Interface in a Smart Mirror form factor was initially proposed as the artefact portion of this project. However, there was an attempt to introduce two separate interfaces. “pageR” and “Natural Language Interface Human Input Device”. Both devices made it very far in development. With “pageR” making it to the testing stage, however to the detriment of the quality of portions of the report.
7	The initial prototyping phase while completed on time is another point of contention. In anticipation of the start of this project I began prototyping africus, overlooking of the level of detail that needed to include in the final report with the assumption that it could be derived post-hoc in a reflective manner. While some decent attempt was made at this the level of quality and detail doesn't meet the standard, I believe could have been achieved had I started the project within the main time frame with more diligent word processing. As while sufficient notes were made in my logbook, they weren't all converted to the final report in a sufficient manner.





8	The project review report was the first sign of faltering time management as there was a false assumption made that sufficient time had been left to include the information required. While all research had been conducted sufficient time should have been diverted from prototype development to documentation.
<b>Suggestion</b>	A sufficiently placed deadlines should be set at a point to stop research. Whilst interesting and potentially beneficial for other aspects of academic study priority needs to be placed on documentation of the task.
9	The intricacies of the system I intended to build were underestimated, whilst I can effectively communicate my ideas regarding the structure and design, the system this needs to be converted into a form where it can be effectively analysed. I had built several prototypes as examples, solutions using optical character recognition and iOS shortcuts hosted on google cloud – several were shown to my supervisor during the duration of the project, however these needed to be presented in a form they can be analysed.
10	CAD for pageR was completed to create a “looks-like” prototype of what I envision the final form factor to be. 3D printing was used in its development, this could have benefited the report if I included some analysis of this process compared to other prototyping methods. However, it can be argued that 3D printing befits the fast iterative approach I indeed to take for this project.
11	A physical PCB prototype wasn’t created however a cad drawing was produced for the Rotary Encoder Pull Down Resistor/Low Pass Filter Network and it was created using through hole components. This digital drawing could be converted into a PCB schematic.
12	The mirror code wasn’t refactored as additional time was spent developing pageR and its code.
13	pager’s code wasn’t refactored as its first iteration cycle had complete by the project’s deadline. However, it did so by reaching its minimum viable product state which satisfied the requirements. See <a href="#">Appendix 12 016</a> .
14	Iterative improvement was carried out for the intended duration of the project, however it also caused issues in the effective documentation of the work as a higher priority was incorrectly placed on producing hardware prototypes.
15	Final report, while completed lacks detail that in areas such as testing and evaluation of the result. It could benefit from some deeper analysis in these areas.
<b>Suggestion</b>	As the main issue lies within the documentation of the project and not the creation of hardware a solution may lie in the management. A sprint structure could be adopted, where bi-weekly periods are explicitly taken to spend diligently documenting the work completed in the weeks prior – then two more weeks of work can commence. Separating these aspects would allow for a more structured workflow, potentially increasing output as there is no worry of information being overlooked.

Table 61 | Project Evaluation





### Appendix 19 | SUS Improvement

#### Improved System Usability Scale

Weak

Strong

		1	2	3	4	5
1	I think that I would like to use this system frequently	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Notes</b>						
2	I found the system unnecessarily complex	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Notes</b>						
3	I thought the system was easy to use	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Notes</b>						
4	I think that I would need the support of a technical person to be able to use this system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Notes</b>						
5	I found various functions in this system were well integrated	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Notes</b>						
6	I thought there was too much inconsistency in this system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Notes</b>						
7	I would imagine that most people would learn to use this system very quickly	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Notes</b>						
8	I found the system very awkward to use	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Notes</b>						
9	I felt very confident using the system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Notes</b>						
10	I need to learn a lot of things before I could get going with this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Notes</b>						
<p><b>Final Evaluation:</b>                      [What do you think would improve your individual user experience, be as detailed as you would like, and we will take it into account]</p>						

Table 62 | Improved System Usability Scale (SUS)







**Improved System Usability Scale**

Weak

Strong

		1	2	3	4	5
1	I think that I would like to use this system frequently	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
<b>Notes</b> it is an interesting implementation of the technology						
2	I found the system unnecessarily complex	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Notes</b> No, using your voice to communicate is easy						
3	I thought the system was easy to use	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
<b>Notes</b> see above						
4	I think that I would need the support of a technical person to be able to use this system	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Notes</b> No						
5	I found various functions in this system were well integrated	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
<b>Notes</b> yes, maybe it could have a few more skills						
6	I thought there was too much inconsistency in this system	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Notes</b> N/A						
7	I would imagine that most people would learn to use this system very quickly	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
<b>Notes</b> : after learning how to use the wakeword						
8	I found the system very awkward to use	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Notes</b> N/A						
9	I felt very confident using the system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
<b>Notes</b> yeah!						
10	I need to learn a lot of things before I could get going with this system.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Notes</b> just the wakeword						
<b>Notes</b>						
<b>Final Evaluation:</b>						
[What do you think would improve your individual user experience, be as detailed as you would like, and we will take it into account]						
I really enjoyed how the lights flashed when you said the wakeword. Maybe you could control the lights with your voice too.						

Figure 88 | Example of Completed Improved SUS





Appendix 20 | Encoder

Name	Description	Use
Encoder Wheel	3D Printed Encoder Wheel	Analog Scroll Wheel
TTC Encoder	Quadrature Encoder Module	Transduces scrolling motion into Quadrature Waveform encoding rotation and direction
Tactile Switch	Momentary Switch	Select/Action Button
Passive Components	2x 10k Resistor 2x 1k Resistor 2x 100uF Capacitor	Pull Down Resistor and High Pass RC Filters for Channels A/B

Figure 89 | Bill of Materials (Encoder)

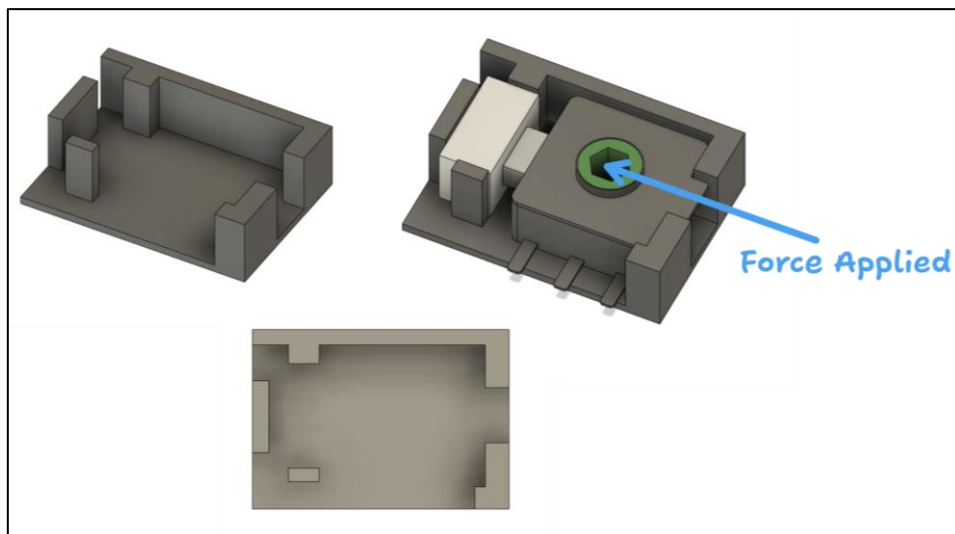


Figure 90 | Encoder Assembly CAD

The encoder assembly was designed to be 3D printed as seen in 7.3.1 Figure 45. Use of additive manufacturing allowed for the rapid prototyping of this element of the project, facilitating the fast iteration cycles as specified in the project methodology.

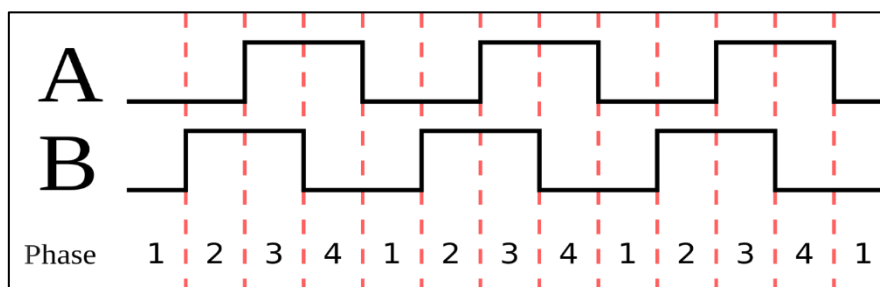


Figure 91 | Phases in Quadrature Encoding

In a quadrature encoder, the direction of rotation is determined by which signal rises first. If signal A rises first, the rotation is in one direction, and if signal B rises first, the rotation is in the other direction. This can be represented by two signals with one phase shifted 90 degrees.





```
int encoderPin = A0; // set the encoder output pin a analog input A0
int encoderVal ; // variable to store the value from the encoder

void setup() {
  Serial.begin(9600); // open the serial port at 9600 bps
}

void loop()
{
  encoderVal = analogRead(encoderPin); //read pin A0 and set its value to
the encoderVal variable
  Serial.print(encoderVal); // print the encoderVal to the serial console
  Serial.println(); // prints new line for readability
}
```

Figure 92 | Initial Rotary Encoder Test Script

Initial testing of the rotary encoder failed to consider quadrature encoding, as such when tested in the Arduino serial monitor the output was incoherent.

```
int val;
int encoder0PinA = A0;
int encoder0PinB = A1;
int encoder0Pos = 0;
int encoder0PinALast = LOW;
int n = LOW;

void setup() {
  pinMode(encoder0PinA, INPUT);
  pinMode(encoder0PinB, INPUT);
  Serial.begin(115200);
}

void loop() {
  n = digitalRead(encoder0PinA);
  if ((encoder0PinALast == LOW) && (n == HIGH)) {
    if (digitalRead(encoder0PinB) == LOW) {
      encoder0Pos--;
    } else {
      encoder0Pos++;
    }
    Serial.println(encoder0Pos);
    Serial.println("/");
  }
  encoder0PinALast = n;
}
```

Figure 93 | Correct Implementation





The encoder was initially tested in a floating state wherein values could switch from 0/1 due to noise introduced by the power source. To mitigate this, pull down resistors were added to the signal lines ensuring they remain at logic 0 when not powered. Additionally, a low pass filter was implemented to filter out the noise.

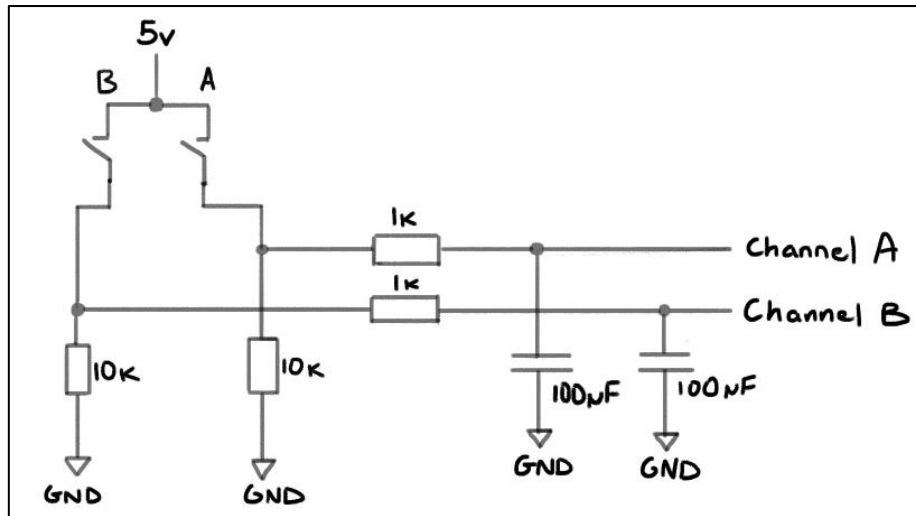


Figure 94 | Initial Design

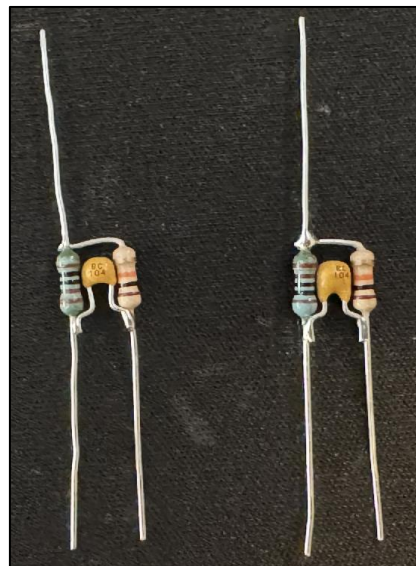


Figure 95 | Practical Implementation | 10k Resistor 2x 1k Resistor 2x 100µF Capacitor.

$$f_c = \frac{1}{2\pi RC}$$

$$f_c = \frac{1}{2\pi \times 1000 \times 100 \times 10^{-6}} = 0.159 \text{ Hz}$$



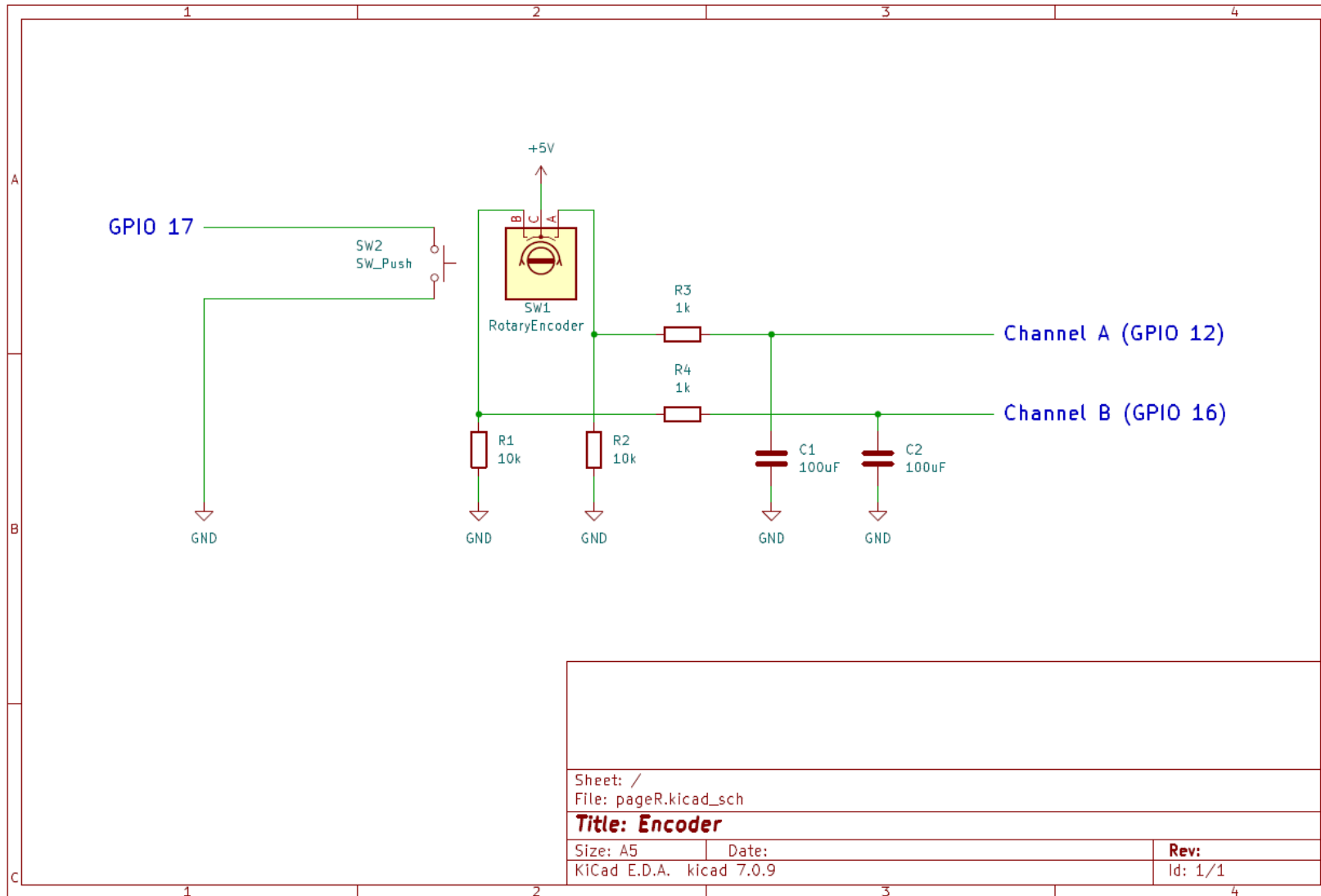


Figure 96 | Encoder Circuit Diagram





Appendix 21 | pageR Pinout

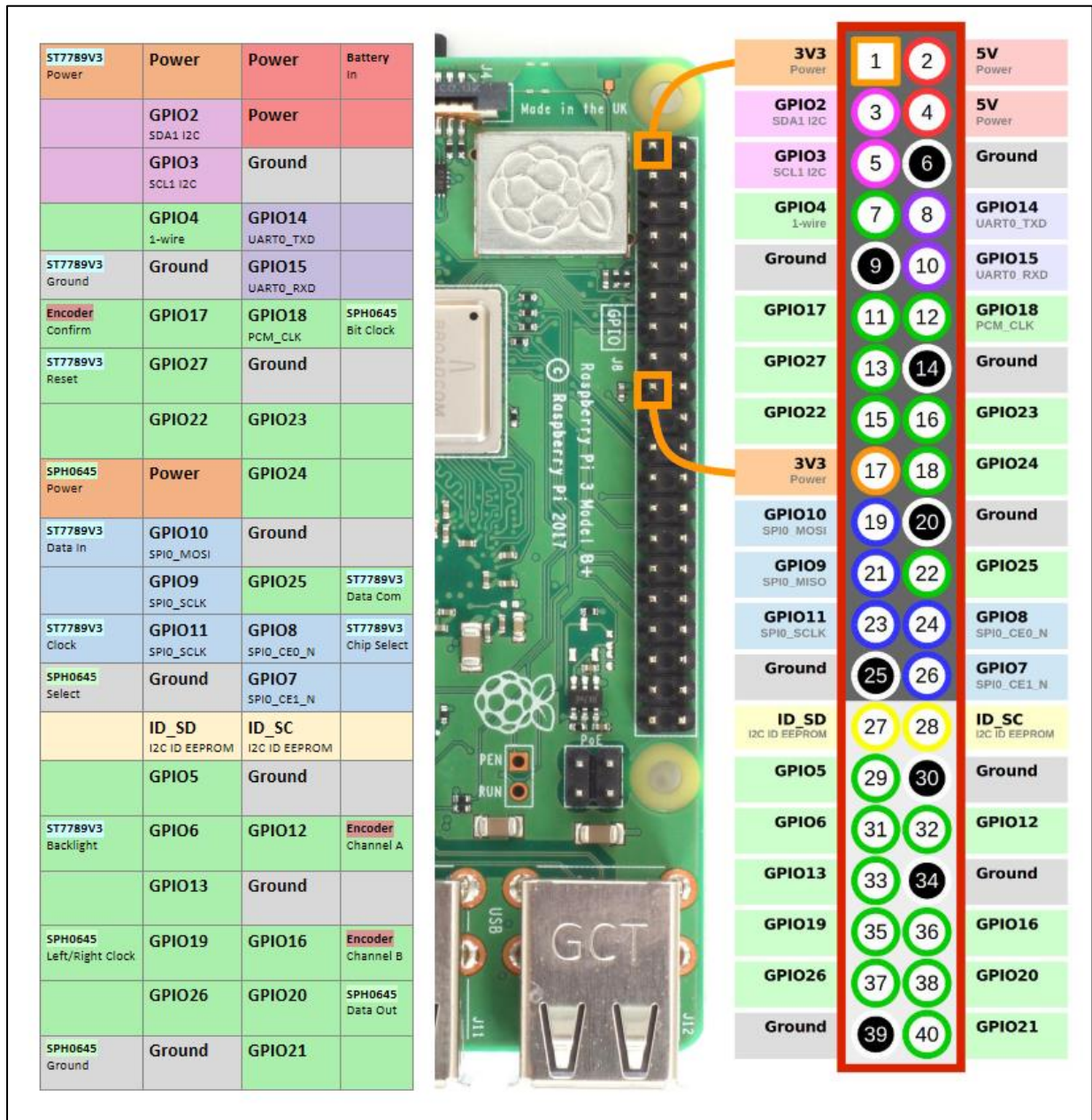


Figure 97 | pageR Pinout (left) Raspberry Pi GPIO (right)





Appendix 22 | M3 Bolt Testing



Figure 98 | Bolt Testing

Fusion 360 was used to calibrate the settings for the pitch of the M3 bolts such that they would sit flush with the print.

Criteria derived from relevant requirements:
--

<b>NLI-005 user experience focus</b>
--------------------------------------

- |   |
|---|
| <ul style="list-style-type: none"> <li>- A flush surface finish will improve how the device feels to hold.</li> </ul> |
|---|

