University of Waterloo

Faculty of Engineering
Department of Electrical and Computer Engineering

Final Report

Kraeq
(Voice Recognition AR Device)

Group 2022.36

Prepared by
Shuhong Liu (20670463)

Yi Chen (20659492)

Yifan Li (20677030)

Zhiming Lin (20674085)

Consultant: John Zelek

20 March 2022

Abstract

The hearing-impaired people are now receiving more and more attention in society. There are also many products on the market that help the hearing-impaired. Moreover, most hearing-impaired people cannot afford good-quality hearing aids on the market or have a cochlear implant. This project aims to design a cheap, convenient, and flexible visualized tool to provide an alternative for people with hearing problems to make their life easier. The product adopts AR (Augmented Reality) technology that is currently popular in the market and has a promising development prospect. The product has the function of real-time speech-to-text conversion. The microphone converts the recorded sound into text, and a viewfinder presents the text in front of the user's eye as AR images. The mechanical frame of the product is 3D-printed with PC-ABS material. The core-PCB contains an Allwinner H3 processor. In terms of software, the operating system is Linux for the best driver compatibility, and the AR program is written in Python. Furthermore, the device is flexible because it is detachable, so it may be worn with most glasses. The project's main benefit is that it helps people with hearing loss see the words when chatting to others or listening to a speech so that they can easily understand and join the conversation.

Acknowledgements

I want to express my most profound appreciation to all those who assisted us in completing the final year design project and the final report. I give special gratitude to our project consultant, Prof. John Zelek, whose suggestions and encouragement helped us finish our prototype design and construction.

Furthermore, I would also like to acknowledge with much appreciation the crucial role of the lab instructor of Mr. Kim Pop. He provided us with valuable tools and much constructive advice on setting up our circuits. A special thank goes to our online symposium marker, Mr. Andrew Morton, who contributed his time to our final project presentation and gave valuable suggestions. Last but not least, many thanks go to the course instructor, Prof. Sagar Naik, who companied every FYDP group for two semesters and assisted everyone in completing the course.

I have to appreciate all the team members' contributions and efforts to this project. Finally, I would like to thank an upper-year student who gave suggestions on the report structure and formatting.

Table of Contents

List of Figures

List of Tables

1 High-Level Description of Project

1.1 Motivation

One million people, or 15% of the world’s population, experience some form of disability, of whom 25% are hearing impaired people. [1] As technology advances, a growing number of products assisting the hearing-impaired appear on the market. However, a cochlear implant can cost $30,000 to $50,000 on average [2], and a hearing aid with relatively good sound quality costs around $2,000. Since disabled people are commonly experiencing poverty and discrimination, such expensive assistance is unaffordable for them. In addition, AR (Augmented Reality) technology is the most promising visual imaging technology in the market. Therefore, a cheap, convenient, and flexible speech-to-text AR device is essential for the hearing-impaired to improve their life quality.

1.2 Objective

The objective is to design a detachable voice recognition device with built-in AR technology. In terms of software, the device should support real-time speech-to-text translation. The prototype is aimed to be cost-effective, light-weighted, and detachable from eyeglasses. The price of the product should be around CAD$300.

1.3 Block Diagram

The high-level design of the voice recognition AR device is presented in Figure 1. Overall, the design comprises three subsystems: display subsystem, voice recognition subsystem, and execution subsystem.

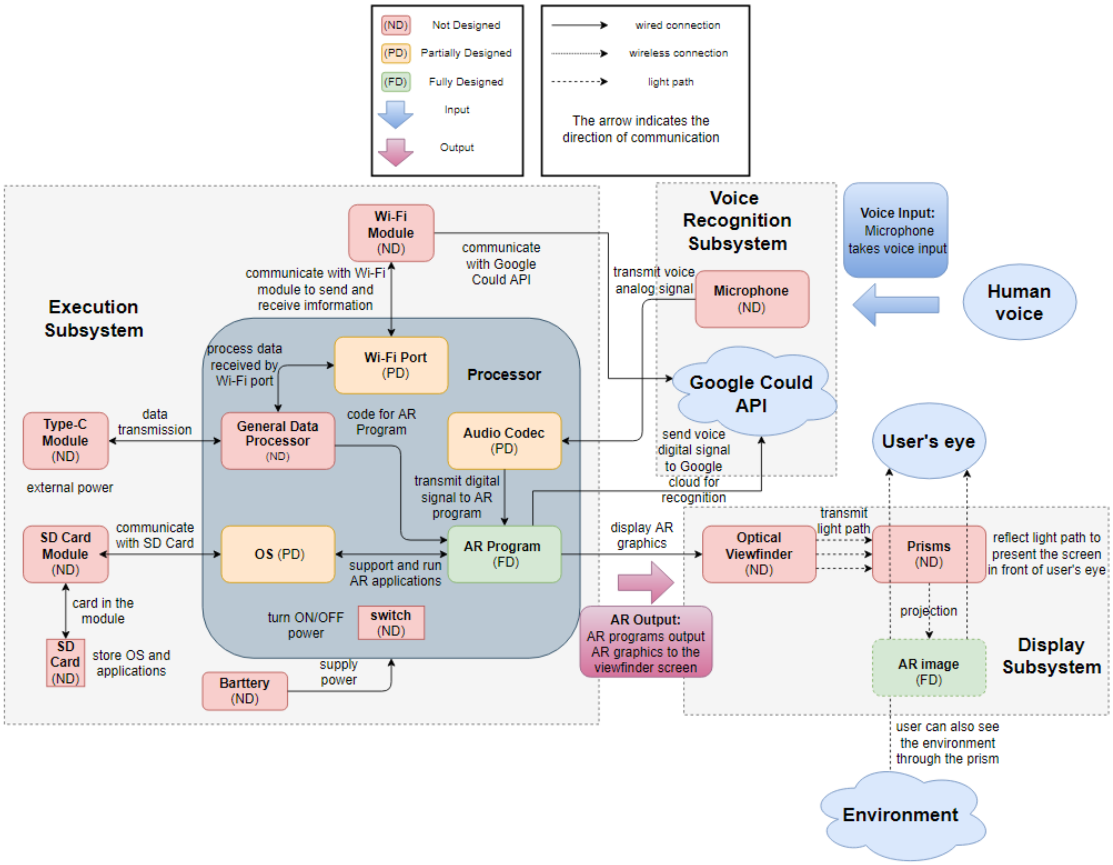

(ND) Not Designed
(PD) Partially Designed
(FD) Fully Designed
Input
Output
wired connection
wireless connection
light path
The arrow indicates the direction of communication
Execution Subsystem
Wi-Fi Module (ND)
communicate with Wi-Fi module to send and receive imformation
communicate with Google Could API
Processor
process data received by Wi-Fi port
Wi-Fi Port (PD)
Type-C Module (ND)
data transmission
General Data Processor (ND)
code for AR Program
Audio Codec (PD)
external power
transmit digital signal to AR program
SD Card Module (ND)
communicate with SD Card
OS (PD)
support and run AR applications
AR Program (FD)
card in the module
turn ON/OFF power
switch (ND)
SD Card (ND)
store OS and applications
Barttery (ND)
supply power
Voice Recognition Subsystem
transmit voice analog signal
Microphone (ND)
Voice Input: Microphone takes voice input
Human voice
Google Could API
send voice digital signal to Google cloud for recognition
User's eye
display AR graphics
transmit light path
Optical Viewfinder (ND)
Prisms (ND)
reflect light path to present the screen in front of user's eye
projection
AR Output: AR programs output AR graphics to the viewfinder screen
AR image (FD)
Display Subsystem
user can also see the environment through the prism
Environment

Figure 1. Block Diagram of detachable voice recognition AR device

1.3.1 Display Subsystem

The display subsystem contains a screen, a convex and a prism. The screen takes the output from the processor through a HDMI cable and displays graphics on the screen. The convex in placed next to the screen. It magnifies the image on the screen for a clearer view. The prism is implemented for imaging. It changes the light path from the viewfinder screen so that the screen can be projected in front of the user's eye. An image of the display subsystem prototype is shown in Figure 2.

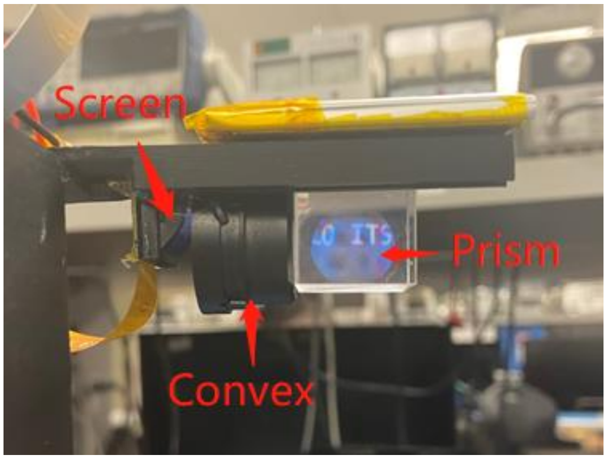


Figure 2. Display subsystem prototype

1.3.2 Execution Subsystem

The execution subsystem consists of two parts, battery and the main PCBs (printed circuit boards). Figure 3 is the battery that we used for assembly. It is a 3.7-V single-cell lithium-ion battery. The battery is connected to the mainboard through a battery module, and it is designed to be rechargeable for long-term use. The battery size is 4cm x 2.5 cm, which satisfies the battery size specification. A switch is added to turn on and off the power.

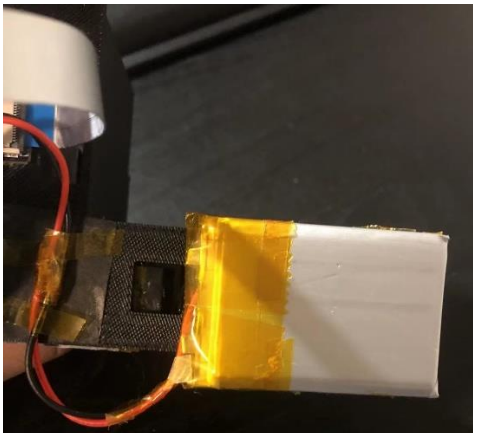
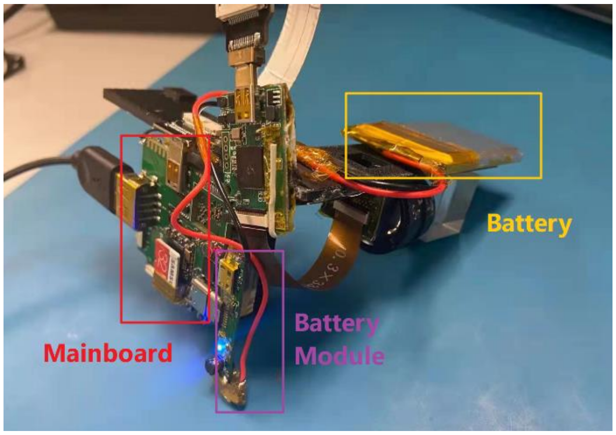


Figure 3. Battery

The mainboard includes a CPU and interfaces used for external connections. We used ubuntu 18.0.4 as our operating system. The processor executes software programs, which are written in Python. The processor is connected to Google Cloud API through W-Fi for voice recognition. A fun is attached to the mainboard to cool down the CPU. Figure 4 shows the connection of the mainboard, battery, and battery module. Figure 5 shows the fan attached to the mainboard.



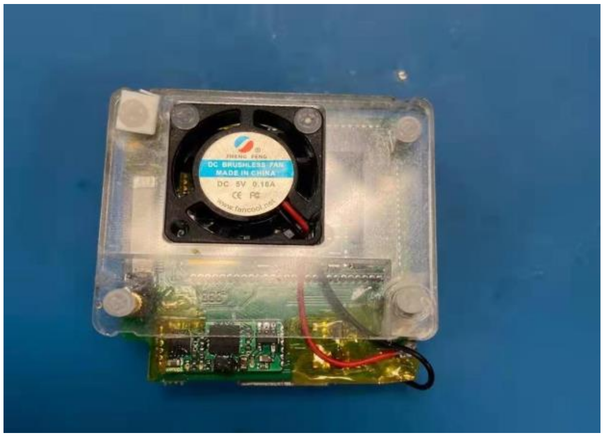


Figure 4. Connection of execution subsystem

Figure 5. Fan on the mainboard

1.3.3 Voice Recognition Subsystem

The voice recognition subsystem takes the input voice. It has a microphone used to record voice and transmit the analog signal to the processor. We use Google Cloud API to accept voice signals from the processor and transmit text data back to the processor.

2 Project Specifications

Project specifications consist of functional specifications and non-functional specifications.

2.1 Functional Specifications

This section elaborates on the functional specifications for the project. They are further broken down into essential and non-essential specifications. Table 1 describes the essential specifications that must be met to consider the satisfaction of the project. Table 2 describes the non-essential specifications that can be met to do the project beyond just satisfactory.

Table 1. Essential Functional Specifications

Subsystem	Specification	Description
Execution Subsystem	Cable Interface	The board must have at least one USB connector to support charging and sharing data with external devices such as computers. USB Type-c port is one of the choices.
	Wi-Fi Interface	The board must have a Wi-Fi module to support a direct wireless connection with Google Cloud API.

	Real-time Voice Recognition Program	The AR program run on the processor must allow the microphone to record the voice and convert it to text in real-time.
	Real-time Screencast Program	Similar to connecting laptops to a monitor, the AR program must allow casting the screen of an external device on the screen in real-time.
Display Subsystem	Image Projection	The light path design must project the screen of the viewfinder to the center of the user's eye so that the user can see the entire screen.
	Transparency	The projected image must be transparent so that the user can also see the scene in front while wearing the AR device.
Voice Recognition Subsystem	Microphone Quality	The device can clearly record the speech if the user talks within one meter

Table 2. Non-essential Functional Specifications

Subsystem	Specification	Description
Overall Prototype	Detachable	The user should easily attach the device onto or remove the device from their glasses.

2.2 Non-functional Specifications

This section elaborates on the non-functional specifications for the AR device. These specifications are not relative to how the system operates; instead, they are characteristics of the system. Table 3 describes the essential non-functional specifications that must be met. Table 4 describes the non-essential specifications that can be met to do the project beyond completion.

Table 3. Essential Non-Functional Specifications

Subsystem	Specification	Description
Overall Prototype	Affordability	The overall cost of the AR device shall be affordable to the public. Each device should cost no more than $300 to produce.
	Weight	The total weight of the AR device containing PCB and Battery should be less than 100 grams to make the device wearable.
Execution Subsystem	PCBs Size	The scale of the processor must not exceed 5 cm x 5 cm
	Battery Rechargeability	The battery must be rechargeable.
	Battery Size	The length and width of the battery must not exceed 5 cm x 3 cm to make glass wearable.
	Battery Capacity	The battery must support the AR device to keep running for at least 5 hours without charging.
Display	Latency	The screen projected into the user's eye must be real-time with negligible latency and without strobe.

Table 4. Non-essential Non-Functional Specifications

Subsystem	Specification	Description
Overall Prototype	Appearance	As a wearable device, the appearance of the AR device should be acceptable or delightful for most people.
	Ergonomics	The AR device should be comfortable to wear.
Execution Subsystem	Battery Fast charging	The time of fully charging the battery should be within 1 hour.

3 Detailed Design

3.1 Mechanical Design

3.1.1 3D Printed Model

The 3D model of the frame is built with SolidWorks as shown in Figure 6. The structure of the model has a few specifically designed structures to maintain stability and safety. Since the frame must be attached to glasses, a cuboid structure with a through groove is designed to fix the frame on the rims.

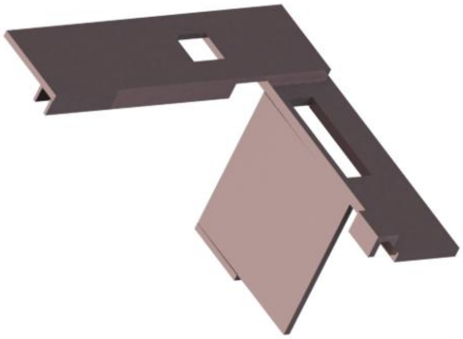


Figure 6. 3D-printed model

Besides, an L-shaped clip is added on the other side of the frame to be attached to the temple. To make the prototype look neat, some spaces are left at the top of the frame for wiring. The last piece is a rectangular separator on the right to prevent PCB from direct contact with the skin. The model's material uses Polycarbonate/acrylonitrile butadiene styrene (PC-ABS), which is a hybrid of PC and ABS. It combines the excellent heat resistance of a PC and the flexibility of ABS, making it one of the most popular 3D printing materials in the market.

3.2 Display Subsystem Design

AR technology essentially overlaps the image generated by the AR device with the real world. As an AR device, the display functionality is critical as it directly affects the quality of the AR images presented for users. To create a good user experience, the display subsystem is required to generate high-quality and low-latency AR images. The display subsystem design of this project consists of three components: a viewfinder, a convex and a prism. The viewfinder displays the software outputs on the screen. The convex is attached next to the screen to magnify the image. The prism adjusts the light emitted from convex and focuses it at a certain place in front of the user's eye. A flow chart of the display subsystem’s working principle is illustrated in Figure 7. The following subsections introduce the detailed design of these three components.

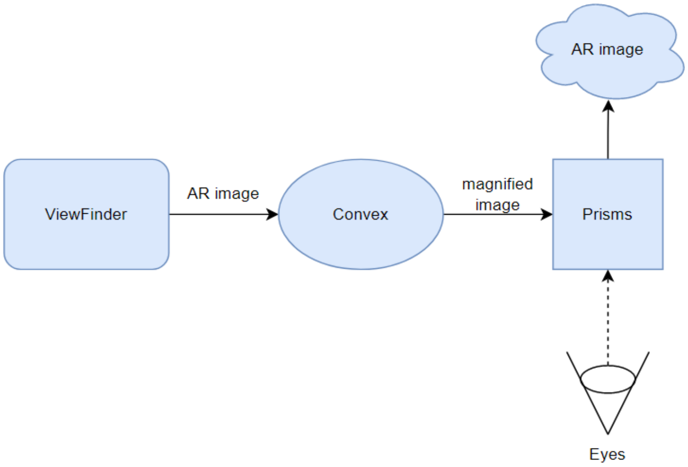


Figure 7. Work priciple of the display subsystem

3.2.1 Viewfinder

The viewfinder contains two modules: a ferroelectric liquid crystal on silicon (FLCOS) microdisplay and a PCB module.

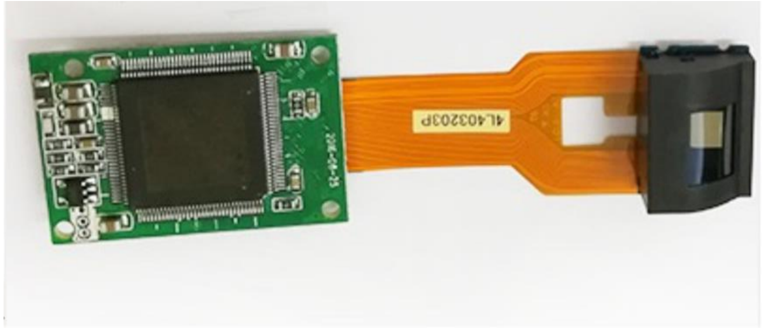


Figure 8. A similar version of the viewfinder

Figure 8 presents a viewfinder with a different version but similar to the one we used. FLCOS microdisplays differ from others in that they use a proprietary ferroelectric liquid crystal (FLC) at a frequency of less than 100 microseconds, while conventional nematic liquid crystals switch at 10 milliseconds. This fast speed allows FLCOS microdisplays to have better and more detailed content. The microdisplay in our prototype supports resolutions up to 1280 x 720p. It has high color contrast, low power consumption and illustrates more detailed pictures. These advantages make it better to present high-quality AR pictures. In addition, the weight of the display is only 5g, which fulfills our requirement for lightweight. The PCB module acts as a bridge between the display and the external circuit. It supports HDMI signal input, so the viewfinder can be directly connected to the motherboard via an HDMI connection.

3.2.2 Convex

A convex lens exists between the viewfinder and the prism. Its function is to enlarge the image output from the viewfinder and transmit it to the prism so that the user can view a clearer text image.

3.2.3 Prism

AR devices can be classified based on appearance, imaging effect, application scenarios, and cost. For this project, we consider three AR optics options: prisms, curved reflections, and waveguides.

3.2.3.1 AR optics options

Table 5 lists the working theory, advantages, and disadvantages of the AR optics options we considered. Based on this table, a comparison is conducted to select the best-fit AR optics for this project.

Table 5. AR Optics Options

	AR optics options	Working Theory	Advantages	Disadvantages
	Prism	Image by projecting onto a prism with a cut reflective surface	- Small and light-weighted - Easy to wear - Low cost	- Narrow angle of view - Limited application
Surface Reflection	Big Curved	Uses specular reflection to an image	- Good imaging quality - Wide angle of view - Immersive experience	- Big size - Inconvenient to wear
	Small Curved	A combination of prism projecting and specular reflection	- Small and light-weighted - Easy to wear - Low cost	- Requires a certain thickness of glass for a reflective mirror
Waveguide	Reflective Waveguide	Light is reflected in the ultra-thin glass to a specific position	- Thin glass - Clear imaging - High color reduction	- High cost - Complex manufacture

	Diffractive Optical Element (DOE)	Diffraction occurs when the light encounters the etched grating	- Small area to image - image with different focal lengths	- Require a certain level of algorithm - High cost
	Holographic Optical Element (HOE)	Light is reflected to image when it encounters a thin film of holographic material	- Low cost - Easy manufactured	- Immature technology

The AR optics performance was evaluated from four main aspects: wearing difficulty, cost, imaging quality, and development difficulty. The product is a wearable and attachable AR device. The difficulty of wearing is the most critical criterion we consider. Therefore, the most prominent weight of 40% is given to the wearing difficulty. Since one of the objectives of this project is to design a cost-effective AR device, the cost is the second important factor to consider. The cost is given a weight of 25%. In terms of the imaging quality and the required technology, we need to balance them. Since the project's functionalities include translation, projection, and teleprompting, imaging quality is not highly required. The imaging quality is weighted 15%, and the development difficulty is weighted 20%.

According to Table 6, the prism has the highest score among all AR optics options. A conclusion has drawn that prism is the most suitable AR optics option for this project.

Table 6. AR Optics Options Decision Matrix

AR Optics Options	wearing difficulty (40%)	Cost (25%)	imaging quality (15%)	development difficulty (20%)	Total
Prism	9	9	5	8	8.2
Big Curved	3	8	8	7	5.8
Small Curved	8	7	7	6	7.2
Reflective Waveguide	7	2	8	4	5.3
DOE	6	2	9	3	4.85
HOE	7	5	8	2	5.65

3.2.3.2 Prism Imaging System

The prism imaging system consists of two main parts: a prism and a concave mirror. As shown in Figure 7, the light emitted from the image on the viewfinder first passes the prism and reaches the concave mirror. The mirror reflects light and forms an intermediate virtual image on the left. The reflected light is reflected again by the prism into the user's eye, which forms the final virtual image in front of the user's eye. The final image in front of the user's eye should be at least 15cm away, and the mini-OLED is placed 3cm away from the concave mirror. The focal length of the concave mirror is calculated as: $f = \frac{u*v}{u+v}$, where u is the object distance and v is the image distance. Therefore, $f = -15cm * 3 - 15 + 3 = 3.75cm$.

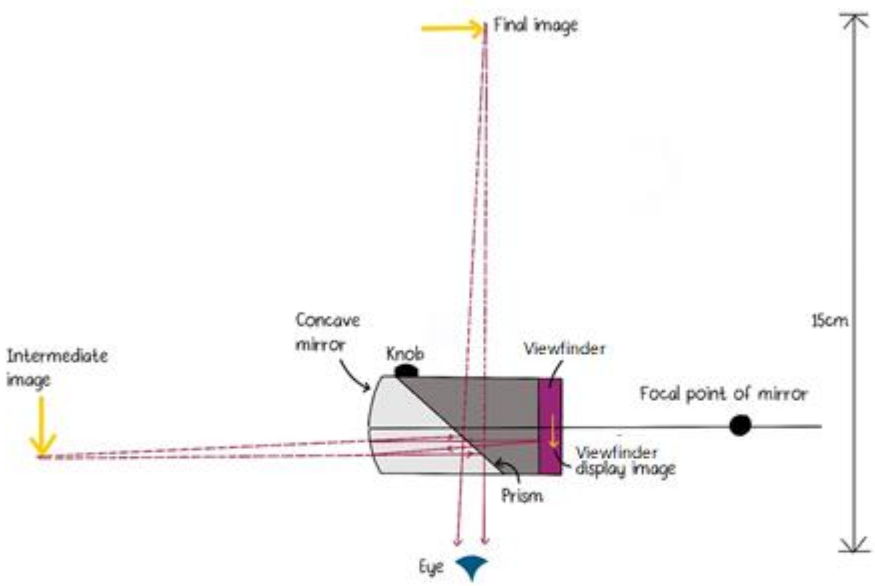


Figure 9. Prism Imaging Light Path

The focal length of the concave mirror should be 3.75cm to present the AR image 15cm away from the user's eye.

As shown previously in Figure 9, a knob is added to adjust the degree of the prism, which changes the position of the AR image to meet different users' requirements.

3.3 Execution Subsystem

3.3.1 Hardware System (PCBs)

The hardware system is responsible for running the Linux system to support the execution of microphone recordings and content display. It is also used to transmit recordings to the cloud for voice recognitions and receive the corresponding results from the cloud. There are 3 main parts in the hardware system: core-PCB, mainboard-PCB, and Power-PCB.

3.3.1.1 Core-PCB

The Core-PCB is the brain of the whole IoT device. It receives the input data and signals from the mainboard-PCB. The processor on Core-PCB, which runs a Linux system file on the TF card, executes the input signals. Once the results are generated, the corresponding output contents for displaying is sent back to the mainboard-PCB.

The Core-PCB contains an Allwinner H3 processor, and the architecture is shown in Figure 10.

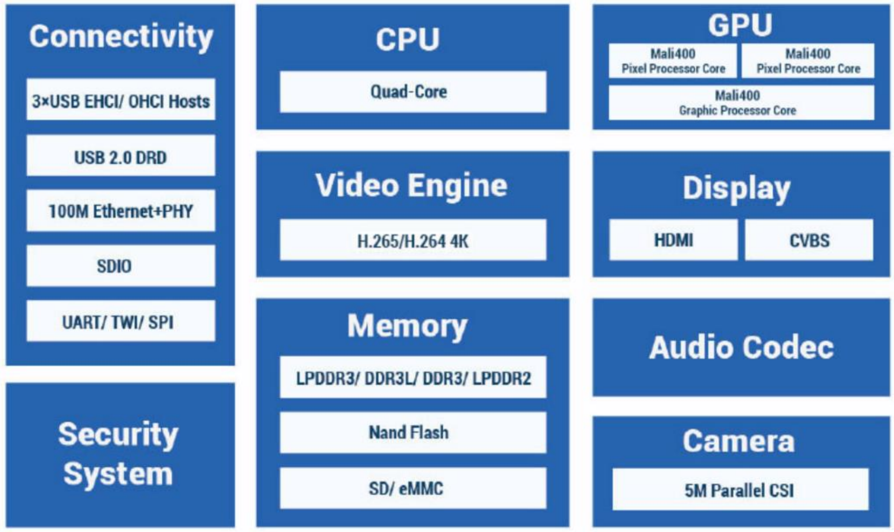


Figure 10. The architecture of an Allwinner H3 Processor [3]

As shown in Figure 10, in addition to the CPU, GPU, and memory required to run the basic Linux system, the processor also contains other specific execution modules, such as connectivity, video engine, display, and camera. These modules are handy for designing functions on the device:

- The connectivity module allows other parts of the device to communicate with the processor by centralizing the I/O signals.
- The video Engine module contributes to the displaying function by accelerating the video decoding and encoding.
- The display module, which makes the display contents stable, can directly send out HDMI outputs signals to the viewfinder module.

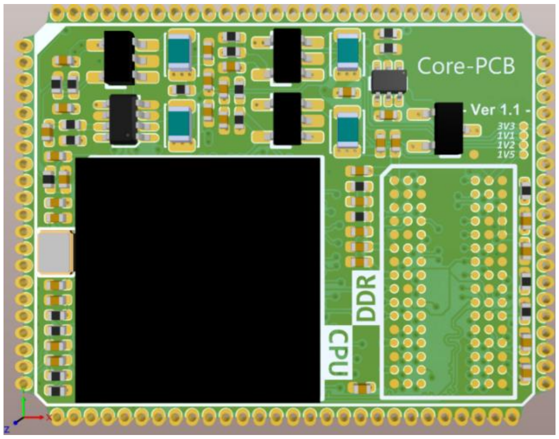


Figure 11. 3D Model of Core-PCB

As illustrated in Figure 11, the Core-PCB contains an Allwinner H3 processor (CPU), a Samsung Double-data Rate (DDR) memory, and some electronic parts working as voltage regulators and analog signals filters. Samsung DDR memory is an independent cache for CPU, allowing Core-PCB to have more robust data processing performance.

3.3.1.2 Mainboard-PCB

The Mainboard-PCB is an intermediate PCB that communicates with Core-PCB and other parts of the IoT device. It transmits the input signal to the Core-PCB and sends the display output from the Core-PCB to the mini-OLED through the Type-C port or Wi-Fi connection. Figure 12 presents the 3D model of the Mainboard-PCB.

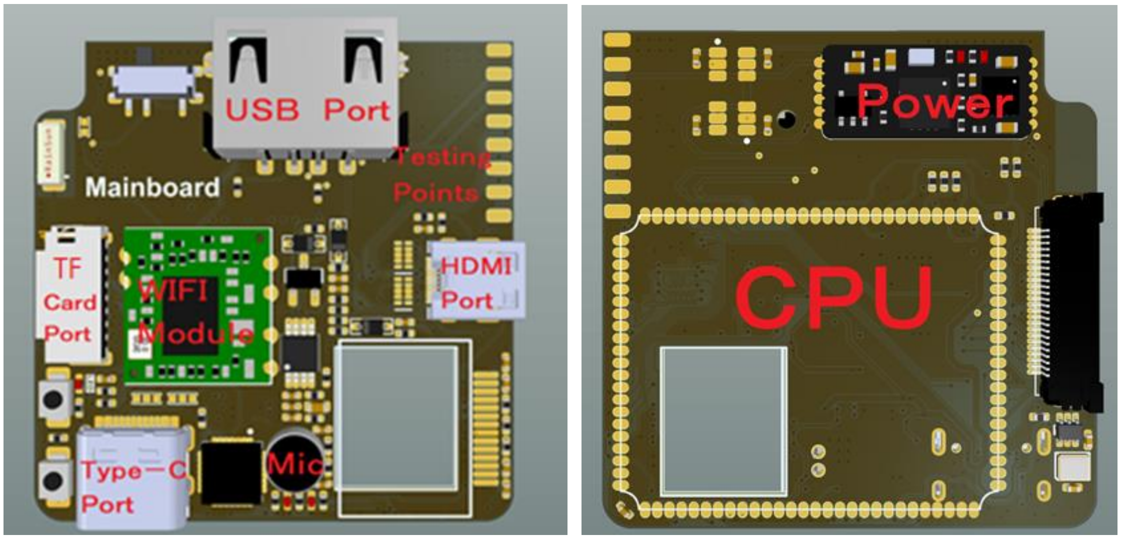


Figure 12. The 3D layout of the front side (left) and backside(right) of Mainboard-PCB

The yellow circles in Figure 12 (right) are stamp holes used to attach Core-PCB on top of Mainboard-PCB. This stamp holes design separates one whole large PCB into two smaller PCBs. It significantly decreases the size of the PCB to 36.017mm * 29.845mm while only adding around 3mm of thickness, that is, the thickness of Core-PCB. Therefore, the feasibility of wearing this IoT device increases significantly by adopting such a design. Moreover, this design makes debugging, testing, and repairing two PCBs easier since it can quickly distinguish whether the execution or transmission part encounters a problem.

In addition, there are a lot of I/O modules on Mainboard-PCB (as shown in Figure 10), which help the Mainboard-PCB communicate with external components.

- A TF card can be inserted into the TF card port on Mainboard-PCB to run the OS while increasing the device's storage. The TF card is chosen in the design rather than an SD card because the maximum memory size of a TF card is sufficient for our storage requirement, and its size is smaller than an SD card.
- There are two buttons on the Mainboard-PCB, allowing users to interact with the system quickly. The user can use the button on the left top to reboot the device and use the left bottom button to pause or resume the ongoing application.
- The USB port can connect external devices, such as the camera and keyboard, to add more functionalities and interactions with the board.
- Wi-Fi module connects the board to the Internet and communicates with the cloud for voice recognitions.
- The mic collects the voice's analog input and sends it to the CPU for processing.
- The Type-C module at the bottom is used for programming and charging purposes.
- HDMI port can connect to the viewfinder to transmit the displaying contents.

3.3.1.3 Power PCB

The small power PCB enables the board to run by connecting to the external battery as the supply power. This power PCB contains a 5V DC-DC boost converter and a 3.3V Buck converter to supply corresponding power to Mainboard-PCB. It also has a charging module so that USB connections can charge the external battery. A lithium-ion battery is used, and its capacitor is based on the hardware system's power consumption. The total average current running on the board is about 1.1 A. Assuming that the battery keeps supplying power for 1 hour, the theoretical capacity of the Lithium-ion battery should be calculated as below:

$$1.1A * 1000 = 1100mA$$

$$1100mA * 1h = 1100mAh$$

Thus, a battery with at least 1100mAh is required for our project. After balancing the weight and running time, we selected a 2000mAh battery as the external power supply.

3.3.2 Operating System (OS)

The external controller and execution system's operating system are selected between the mainstream OS, Android, and Linux. Table 7 shows some differences between the two systems.

Table 7. Linux vs. Android

Linux	Android
The Kernel used in Linux is Monolithic.	Its Kernel type is Linux-based.
It is written mainly in C and assembly language.	It is written in C, C++, and Java.
It is used in personal computers with complex tasks.	It is the most used OS overall on mobile devices for all simple tasks.
Its target system types are embedded systems, mobile devices, personal computers, servers, and supercomputers.	Its target system types are smartphones and tablet computers.

Linux appears to be the superior operating system for the nature of this project. Many existing modules and applications from Linux can be utilized to shorten development cycles. For example, Linux ISO image contains a Network Manager, which can be easily used to enable and set up the Wi-Fi connection; it also has the preinstalled driver for the microphone so that no external software is required. Although Android has similar modules for the above features, it does not guarantee the compatibility of drivers over embedded systems. In addition, Linux is also a developing-friendly operating system to programmers, as its kernel and device tree can be easily modified and built to fit some specific criteria for the embedded devices. Besides, linux offers far more flexibility than Android, and this system has the potential to be further scaled as a powerful wearable server or portable computer because of the powerful computation capabilities of the chip and the linux system. Therefore, Linux is selected as the primary operating system for this project.

3.3.3 AR Program

3.3.3.1 Real-time Speech-to-text Recognition

The application is designed to provide real-time speech-to-text recognition for users. The application records audio streams from the microphone on the mainboard packets the audio into gRPC message, sends the message to the GCP endpoints, returns the extracted text from the audio, and finally displays the recognized sentences to the user.

Figure 13 illustrates precise steps taken by the application to perform real-time speech-to-text recognition.

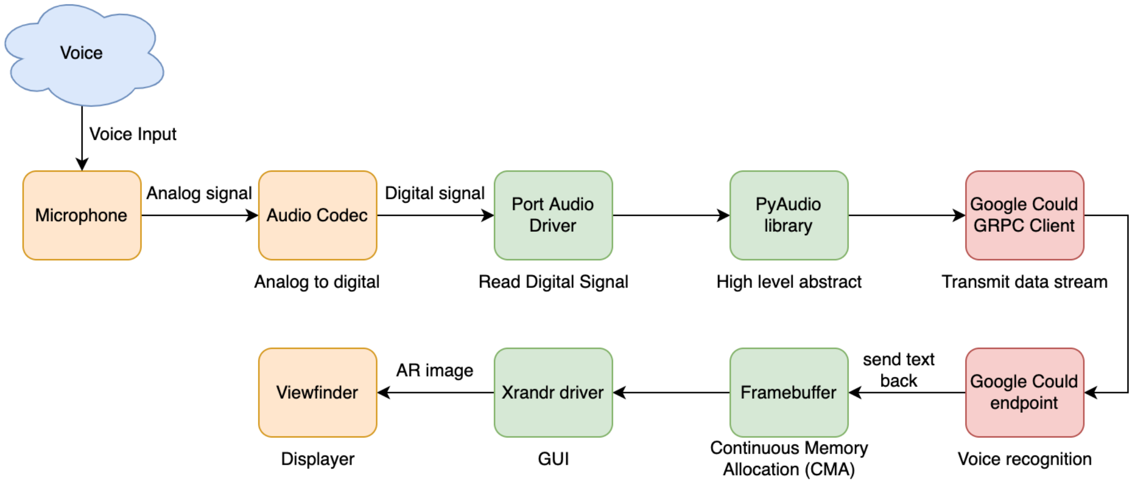
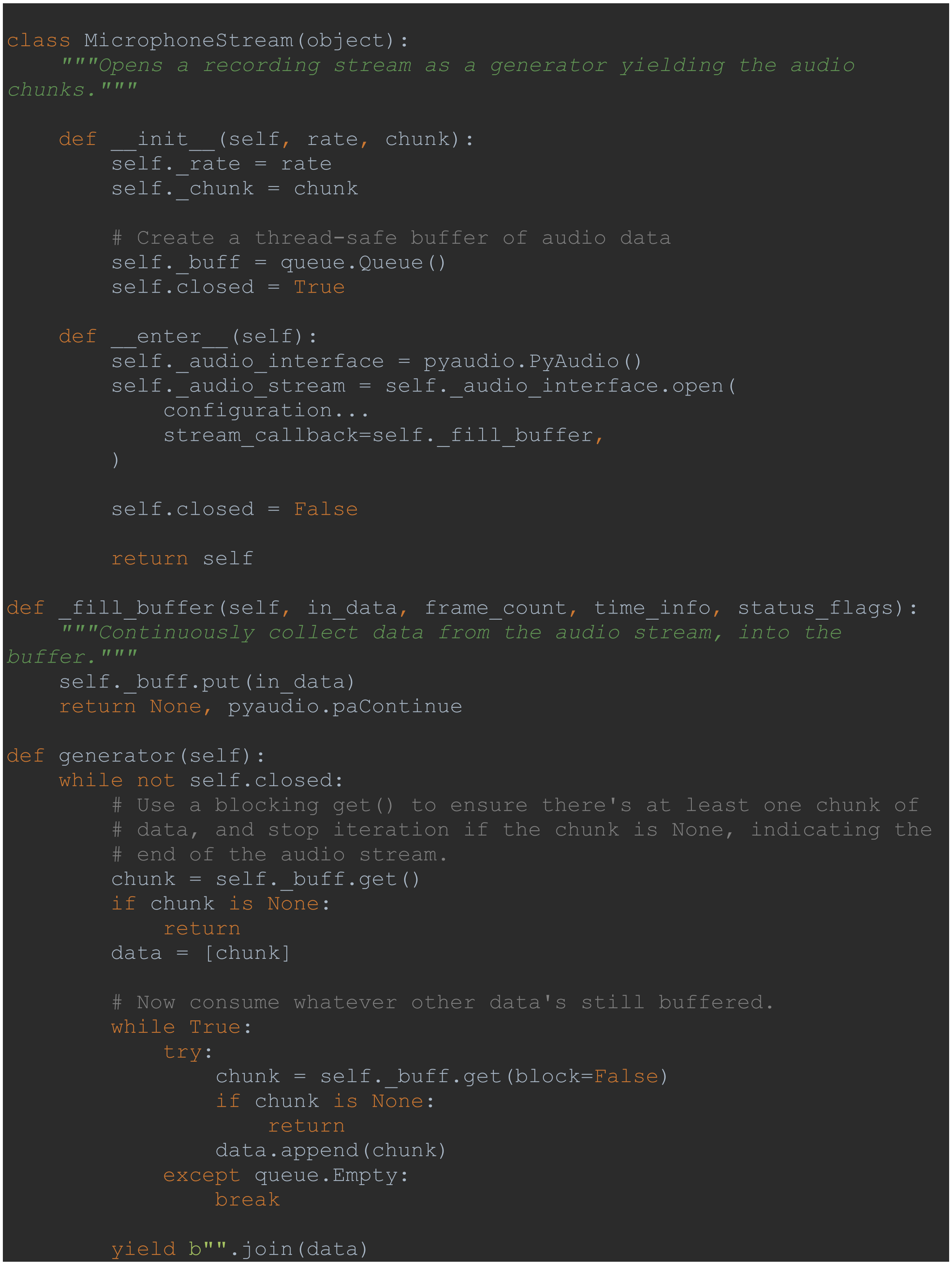


Figure 13. Real-time Speech-to-text Recognition Flow Chart

3.3.3.2 Speech Data Stream

The microphone and audio driver are installed on the system. PortAudio is a C++ audio utility library that can provide the speech data captured by the system. PyAudio, a higher-level interface, provides the functionality of PortAudio for python applications.

PyAudio provides the abstract of the audio client with user-defined chunk size and transmission rate. The data fetched by the system can be loaded to a thread-safe buffer with a specific chuck size by calling the callback function. The chunk size is set to be a length of 100ms speech signal.



```
class MicrophoneStream(object):
    """Opens a recording stream as a generator yielding the audio
chunks."""

    def __init__(self, rate, chunk):
        self._rate = rate
        self._chunk = chunk

        # Create a thread-safe buffer of audio data
        self._buff = queue.Queue()
        self.closed = True

    def __enter__(self):
        self._audio_interface = pyaudio.PyAudio()
        self._audio_stream = self._audio_interface.open(
            configuration...
            stream_callback=self._fill_buffer,
        )

        self.closed = False

        return self

def _fill_buffer(self, in_data, frame_count, time_info, status_flags):
    """Continuously collect data from the audio stream, into the
buffer."""
    self._buff.put(in_data)
    return None, pyaudio.paContinue

def generator(self):
    while not self.closed:
        # Use a blocking get() to ensure there's at least one chunk of
        # data, and stop iteration if the chunk is None, indicating the
        # end of the audio stream.
        chunk = self._buff.get()
        if chunk is None:
            return
        data = [chunk]

        # Now consume whatever other data's still buffered.
        while True:
            try:
                chunk = self._buff.get(block=False)
                if chunk is None:
                    return
                data.append(chunk)
            except queue.Empty:
                break

        yield b"".join(data)
```

3.3.3.3 gRPC Connection

gRPC is a modern open-source high-performance Remote Procedure Call (RPC) framework with pluggable support for load balancing, tracing, health monitoring, and authentication. The application uses gRPC to establish a stable connection to the Google Cloud Platform (GCP) endpoints. The library google.cloud.speech provides the utility functions to create a client for the gRPC connection.

3.3.3.4 Speech Recognition

The speech recognition functionality is provided by the GCP Speech API, which takes the truncated speech stream and returns the recognized sentences via the gRPC session.

3.3.3.5 Display

There are two options for the application display. One uses the classic Linux teletype (TTY), which only requires FrameBuffer to be accessible on the system. The other option is the GUI display which requires X-driver to be preinstalled on Linux.

Table 8. TTY and GUI Comparison

TTY	GUI
Can only display text as a command line on the console	Can show graphical interface
FBterm supports size and rotation	X-driver supports powerful features
Power efficient, low computational cost	Power is dissipated, uses high computational cost

Table 8 shows the trade-off between TTY and GUI display. Considering the speech-text recognition application, which only requires recognized text to be displayed on the screen, TTY is selected as the default method in the application. The configuration, for example, the size of the output text and the rotation of the screen, is hardcoded in a shell script and is executed by default while the system is booting up. This display option helps to save energy and reduces the core temperature during the experiments. To support integration of other applications in the future, the system can switch to GUI display manually.

While switching to GUI display, the speech-to-text application departs two threads. One thread is used to perform real-time speech recognition, and the other thread is used to update GUI components. The communication between two threads, such as fetching and putting, is accomplished using a synchronous queue.

3.3.4 External Connection

There are two types of connections provided in the system, wireless connection, and cable connection. The preinstalled Xrdp driver supports the wireless connection via a Wi-Fi connection. It allows users to connect and login into the system remotely from external devices, such as a laptop. The wireless connection is recommended use for its ease of accessibility. It enhances the availability and freedom of the external controller as users can freely place it as long as it is within the range of the wireless connection. With that being said, the cable connection is still designed and planned to be implemented for developing and charging purposes.

The following two sections discuss the potential technologies to implement the two types of connection and why particular technologies are chosen over others.

3.3.4.1 Cable Interface

The execution system consists of a cable interface. It is used for charging, sharing data, and developing purposes. Table 9 shows the data transfer speed of the latest standards and their corresponding connector types to determine which connector type to use.

Table 9. USB Cable Types, Standards, and Speeds

Standard	As Known As	Connector Types	Max. Data Transfer Speed	Power Capability
USB 3.2 Gen 2	USB 3.1, USB 3.1 Gen 2, SuperSpeed+ SuperSpeed 10Gbps	USB-A, USB-B, USB Micro B, Type-C	10 Gbps	All support the USB Power Delivery (USB PD) Specification, which can offer up to 20 volts of power at 5 amps for a potential of 100 watts
USB 3.2 Gen 2x2	USB 3.2, SuperSpeed 20Gbps	Type-C	20 Gbps	
USB 4	USB4 Gen 2×2, USB4 20Gbps	Type-C	20 Gbps	
USB 4	USB4 Gen 3×2, USB4 40Gbps	Type-C	40 Gbps	

Type-C shows promising potential as it supports various standards and speeds. Furthermore, Type-C is small enough for the execution system and robust enough for a laptop computer. Many new technologies eliminate other connector types and utilize Type-C as the only port for video, network, data transfer, and charging by introducing new protocols (e.g., DisplayPort, MHL, and HDMI). Furthermore, Type-C supports the USB Power Delivery Specification and offers up to 100 watts, far beyond our requirements. It negotiates to supply a lower power depending on the hardware.

Type-C cable is selected as the execution system's cable interface because it supports a fast data transfer rate and has many existing protocols to support different functionalities. Its power capacity is more than enough.

Due to design selection, Type-C was adapted as the power source of the product as it enables high-performance power delivery. Although the performance exceeds the power level that the current system needs, it grants the ability for the system to expand and supports higher power consumption AR applications. On the other hand, USB-A is utilized as an input port for various purposes. With this extra USB-A port, the system can connect to an external controller. Although the current application does not require an external controller, it could be used for developing and debugging purposes when connected to a keyboard or a laptop. Furthermore, it also grants the ability for the system to expand and supports future applications requiring external controllers. Lastly, USB-A was favored over Type-C because USB-A is commonly used universally, and its transfer speed is enough to support most applications.

In conclusion, Type-C and USB-A were incorporated in the system for charging and developing, connecting to external controller purposes.

3.3.4.2 Wireless Interface

There are two mainstream ways to connect the two systems wirelessly: Wi-Fi and Bluetooth. For Wi-Fi, there are two ways to connect. One is for both devices to connect to the same network, which requires the presence of a network. The other is to use Wi-Fi Direct, which allows devices to create their Wi-Fi networks and be connected by the other device without an internet connection.

Table 10. Wi-Fi Direct versus Bluetooth

	Speed (Data Rate)	Power Consumption	Range (neglect obstructions)
Wi-Fi via phone hotspot	30-60Mbps Using 4G LTE	1.5-2 Watts	Up to 10m
Wi-Fi via home/office network	210Mbps-1G Using 5 GHz (802.11ac)	N/A (No studies have shown the exact power consumption on merely connecting to a Wi-Fi)	Up to 125m
Wi-Fi Direct [4]	250Mbps	Up to 20 Watts	Up to 200m
Bluetooth 5.0 [5]	2Mbps/1Mbps/500kbps/125kbps Depending on the distances between devices (e.g., for devices that are further away, we could switch to 125kbps mode to enable low data transfer rate but maximum range)	1 Watt in general or 0.01-0.50 Watt for Low Energy technology. It is implemented using the Lower Power Node feature, where the external controller provides a "store and forward" service to our associated Low Power Node (execution system). This allows our execution system to operate in low power mode by receiving information only when it polls messages from the external controller.	60m-240m

From Table 10, even though Bluetooth provides a better power consumption rate and range, its speed is significantly slower than Wi-Fi Direct. The bandwidth needs to be at least 3Mbps for a clear standard definition to use the real-time screencast program. Bluetooth does not meet this requirement and is therefore eliminated. Moreover, Wi-Fi via phone hotspot or home/office network are eliminated for three reasons: the applications do not need an internet connection; the speed of Wi-Fi Direct is sufficient to accomplish communication between the two systems; Wi-Fi Direct does not increase the cost of the project. Therefore, it is best to choose Wi-Fi Direct for the project.

To iterate and in addition, Wi-Fi Direct provides the following features:

- Peer-to-peer(P2P) connection to quickly locate and interact with nearby devices.
- The network or hotspot is unneeded.

- It supports WPA2 encryption.
- Utilizes broadcast of services which enables easy discovery of nearby devices.

Since Wi-Fi is significantly superior and more commonly used, a Wi-Fi module was included. Although Wi-Fi Direct was not used, the application used the Wi-Fi module to transmit data from the device to google cloud, then fetched back the result from google cloud. Moreover, the benefit of the Wi-Fi module further enhances the compatibility of the device in future development, with a solid board where every standard technology is attached, developers could freely develop applications utilizing various hardware.

Lastly, Bluetooth could be added as an additional criterion that enables the potential for Bluetooth-connected external controllers.

To sum up, the Wi-Fi module was implemented as a crucial wireless interface due to the need of the current application. In addition, it grants many other approaches, such as Wi-Fi Direct, which could be used based on the developer's choice. On the other hand, Bluetooth is temporarily not considered since the Wi-Fi module covers a significant portion of wireless usage. However, it would be a plus to include Bluetooth as part of the system.

3.3.5 Voice Recognition Subsystem

The Voice recognition subsystem requires a microphone and the software to decode audio signals. Transferring captured analog signals to google cloud API was already discussed.

The microphone chosen for the project is a standard electret condenser microphone, model B4012AP422-003, with a sensitivity of -42dB. Generally, a lower sensitivity of a microphone is better for recording distinct, loud sounds, whereas high sensitivity is better for recording ambient sounds. For the application, a lower-sensitivity microphone was chosen. Furthermore, the sensitivity of -42dB falls within the standard range of microphones for recording speaking sounds.

Moreover, the current consumption of the microphone is between 100 to 260 μA, which is significantly low compared to a general microphone with 0.5mA. The microphone was selected such that it fulfills the task of accurately capturing analog signals while drawing as less power as possible to reduce the load of the board. In addition, the microphone costs around 0.33 dollars and 0.20 dollars when purchasing over 1000 units.

In conclusion, this microphone was explicitly selected due to low cost, low power consumption, and enough sensitivity to recording speaking voices.

4 Prototype

4.1 Physical Appearance

The entire prototype attached on a pair of glasses is demonstrated in Figure 14. The product is a detachable device, fulfilling the most significant non-essential functional specification. As briefly introduced in the mechanical design section, the prototype can be easily installed on the glasses by attaching the two fixtures respectively to the rims and the temple. The design of the frame allows the prototype to be stably mounted on the glasses. Furthermore, the device is relatively small and lightweight compared with other AR glasses. The total weight is approximately 90 grams, which is less than the requirement.

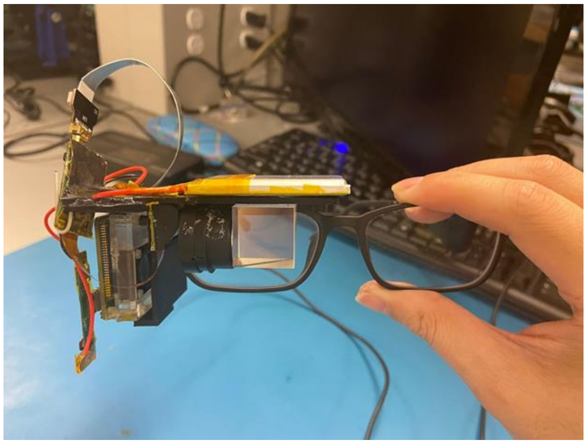


Figure 14. Physical Appearance of the entire prototype

Figure 15 presents the circuitry structure among the three modules: the mainboard, the viewfinder module, and the battery module. The size of the mainboard is required to not exceed 5cm x 5cm and its actual size is approximately 4cm x 4cm. The size of the battery is also within the specification, with the length and width of 4cm and 2.5cm. Being much smaller than expected, the module size makes the product prototype more lightweight than expected.

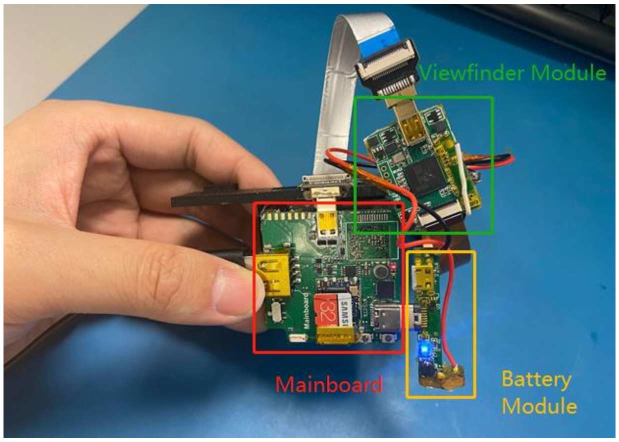


Figure 15. Physical Appearance of the Circuitry

4.2 AR Output Imagery

The display subsystem has two essential functional specifications, which are image transparency and image projection. Figure 16 is an example of the real AR text image that users view while using our product. The projected image in the figure is transparent and the objects behind it can be distinctly seen through the text. During testing, the images barely interfered with the user's observation of the outside world or essential daily activities, which satisfies the requirement for image transparency.

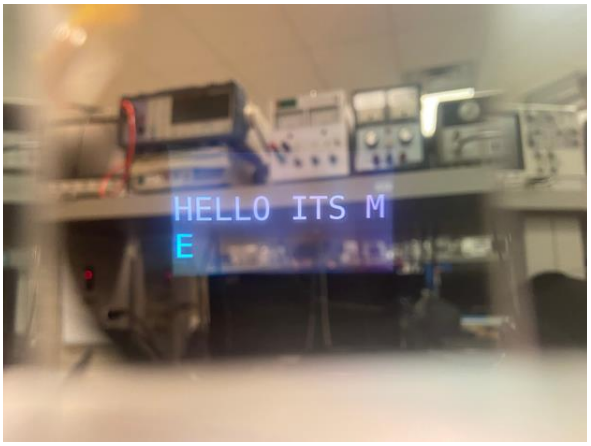


Figure 16. The real AR text image that users view

Figure 17 is a zoomed-out version of AR text image, which better demonstrates the image projection. The center of the figure is the prism of the display subsystem. It is not difficult to see from the figure that the image is projected almost at the center of the prism, which not only proves the accuracy of image projection but also ensures that the user can see the entire screen with no cut-off.

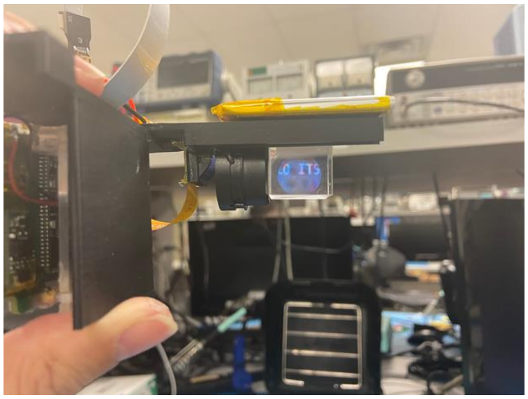


Figure 17. Zoomed out AR text image

4.3 Experimental Data

4.3.1 Recognition Speed

To accurately test and reflect the recognition speed, we did three rounds of testing each with 10 different phrases. The recognition duration is timed from playing the sound to displaying the text image. The results are listed in Table 11.

Table 11. Experimental Data of Recognition Speed

Word	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg (s)**
Round 1 (s)	0.36	0.52	0.39	0.39	0.34	0.34	0.37	0.39	0.30	0.52	0.39
Round 2 (s)	0.45	0.30	0.51	0.47	0.43	0.29	0.37	0.39	0.34	0.41	0.40
Round 3 (s)	0.37	0.40	0.42	0.39	0.45	0.29	0.29	0.50	0.36	0.49	0.40
Avg Speed											0.39

The average speed concluded from the above data is approximately 0.39s. From the user’s point of view, the image is basically synchronized with the input of the sound and this delay is barely noticeable. Therefore, it is safe to say that our product has achieved real-time voice recognition and real-time screencast with almost zero latency.

4.3.2 Recognition Accuracy vs. Distance

Another functional specification that cannot be ignored is the microphone quality. In other words, the device should clearly record and recognize speech within one meter. We conducted 5 rounds of testing with different distances on 5 different sentences. The accuracy is measured as a percentage of accurately recognized words in the sentence. The results are listed in Table 12.

Table 12. Experimental Data of Recognition Accuracy vs. Distance

Sentence	1	2	3	4	5	Avg Accuracy
0.3m	100%	100%	100%	100%	100%	100%
0.6m	98%	97%	99%	96%	98%	98%
0.8m	94%	94%	95%	93%	93%	94%
1.0m	92%	91%	93%	90%	91%	91%
1.2m	89%	87%	90%	89%	89%	89%
Avg Accuracy	95%	94%	95%	94%	94%	94%

It is not difficult to see from the table that speech recognition accuracy is inversely proportional to distance. There is a small drop in accuracy as the distance increases. However, even at 1.2m from the device, the accuracy rate is as high as 91%. During the test, we found that the test error rate of 9% has almost no substantial impact on the output image, and the user can still understand the meaning of the input speech by looking at the text image. Therefore, we believed that the microphone quality of the voice recognition subsystem meets the needs of the product.

4.3.3 Battery Capacity and Charging Speed

The product's battery requirements include capacity and charging speed. We hope that the battery can be fully charged within one hour and can meet the needs of continuous use for more than 5 hours. During the test, it took about 40 minutes to fully charge the battery once, which is within the specification. Additionally, we got nearly 8 hours of continuous use without charging, which is better than the 5-hour requirement.

5 Discussion and Project Timeline

5.1 Evaluation of Final Design

Our project is aimed to design a voice recognition device with built-in AR technology. The AR program supports real-time speech-to-text conversion. The voice recognition subsystem records the voice input and transmits the analog signal to the execution subsystem. The device communicates with Google Cloud API through Wi-Fi and Google's AI technologies are used to accurately convert speech into text. The display subsystem can represent images in front of the user's eye through a viewfinder and a prism. A concave mirror with a specific focal length is chosen to meet the display specifications. The execution subsystem processes input signals and generates output images to the display subsystem. The hardware circuits and components fit the mechanical container. The AR software program supports real-time voice recognition and real-time casting. We have tested our product and the transmission delay can be negligible.

5.2 Use of Advanced Knowledge

This project is closely related to many upper-year courses offered by the university. These courses are ECE358 (Computer Networks), CS484, and ECE298 (Instrumentation and Prototyping Laboratory). ECE358 introduces fundamental design principles and protocols about internet access. We apply knowledge from this course to build Wi-Fi connections between the AR device and external controllers. The image preprocessing uses the knowledge of image normalization and calibration included in CS484. The OCR technology we used in the translation functionality covers the object detection knowledge in this course as well. Although ECE298 is not an upper-year course, the circuit designing and prototyping knowledge we learned was very helpful for our project design.

5.3 Creativity, Novelty, and Elegance

Currently, most of the hearing aids on the market are expensive. Our project provides a cheap and convenient alternative for hearing-impaired people by visualizing words. This unique functionality makes our project stand out and novel. The other creative part of the design is that we embed AR technology into our product to provide a more realistic experience. Its real-time voice recognition and casting feature help the hearing-impaired join the conversation easily. An example of the elegance of this project is its broad applicability. Its lightweight and convenient design allow it to fit on most glasses. In addition, the frame is 3D printed with solid PC-ABS material, which promises the stability and firmness of our prototype when stalled on glasses.

5.4 Quality of Risk Assessment

Back in 498A, we assessed our project hazard that included harmful particles released during 3D printing and the circuit's voltage. Both of them did not produce any safety risk during the prototype construction. However, during the model assembly, we encountered several obstacles that could create safety issues, such as circuit assembly and soldering. Fortunately, we complete our prototype without any accident.

5.5 Student Workload

Table 13 indicates the percentage of overall workload to each group member who has worked on this project. The percentage of each member is relatively even because we separate our tasks clearly. Shuhong was responsible for software development. Yi and Zhiming were responsible for hardware design. Yifan was responsible for mechanical and optical design. All group members contributed to the final prototype assembly and testing.

Table 13. Percentage of workload to each member

Student Name	Percentage
Shuhong Liu	27%
Yi Chen	26%
Yifan Li	25%
Zhiming Lin	22%

References

[1] C. McClain-Nhlapo, "Disability Inclusion," The World Bank Group, 2022. [Online]. Available: https://www.worldbank.org/en/topic/disability#1. [Accessed 20 March 2022].

[2] K. Nunez, "Cochlear Implants: Pros, Cons, and How They Work," Healthline Media, 16 02 2022. [Online]. Available: https://www.healthline.com/health/cochlear-implant#cost. [Accessed 20 03 2022].

[3] "4K OTT Box Total Solution," Allwinner Technology CO., Ltd, [Online]. Available: https://www.allwinnertech.com/index.php?c=product&a=index&id=47. [Accessed 07 10 2021].

[4] W.-F. Alliance, ""Discover Wi-Fi," Wi-Fi Alliance, [Online]. Available: https://www.wi-fi.org/discover-wi-fi/wi-fi-direct. [Accessed 07 06 2021].

[5] I. Bluetooth SIG, "Bluetooth Mesh Glossary of Terms," Bluetooth SIG, Inc., [Online]. Available: https://www.bluetooth.com/learn-about-bluetooth/recent-enhancements/mesh/mesh-glossary/#low_power. [Accessed 07 06 2021].