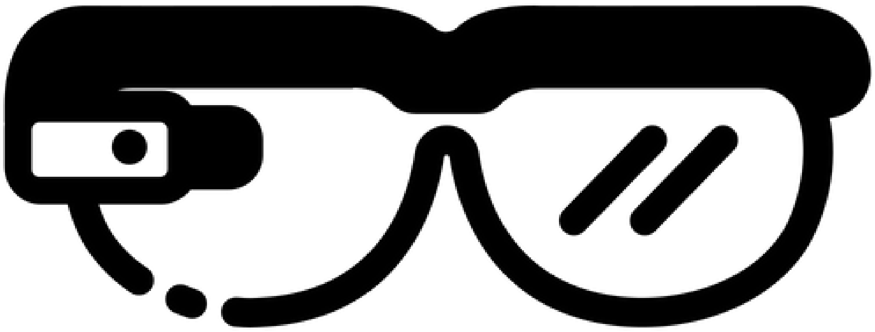
Smart Glasses

The making of an AR Glasses System



By

RG Rämibühl

Matura Paper in Computer Science

Supervisor:

School Year 2022/2023

09.01.2023

Abstract

Augmented Reality (AR) technology has the potential to revolutionise how humans interact with the world around them, by overlaying digital information on top of the physical environment. An upcoming application of Augmented Reality is AR Glasses, more commonly known as "Smart Glasses," which can be worn like normal glasses while providing a seamless, hands-free AR experience. This paper presents the radical potential of Augmented Reality (AR), the technology behind AR, its instances, and the building of a prototype AR Glasses system. This paper focuses on documenting the process of building a prototype AR Glasses System, including the process, considerations as well as challenges in building it. Furthermore, the paper supplements the documentation with explanations of AR technology, its potential and various applications. It concludes with an evaluation of the project and a discussion of the limitations and future directions of AR Glasses technology.

Keywords: Augmented Reality, AR Glasses, Programming, 3D-Printing, Product Design

Table of Contents

1. Introduction

There is no question that Technology is deeply embedded in our lives. Day by day, it steadily makes its way into more aspects of our lives. technology eases our daily life, taking over and automating numerous aspects which would otherwise be tedious. It is estimated that all the electronic devices in the world generate 2.5 quintillion bytes per day[1].One needs to look no further than our own pockets. This device called the "Smartphone" is so integral to our life that we could not imagine a life without it. It is not only the smartphone that we carry everywhere with us. Technology has replaced our analogue watches with smartwatches that can do fitness tracking, receive notifications, and much more. We carry these little devices on our wrists with us. It again makes our lives much simpler, giving us a little vibration when we receive a notification. It helps us keep track of our steps and fitness goals and helps us achieve them. Since the early 2010s, the market has steadily been growing, with big companies like Apple and Google investing in the market and releasing their own products. The global wearable technology market size was accounted at USD 121.7 billion in 2021 and it is expected to reach around USD 392.4 billion by 2030[2].

Yet, there is still one area where technology has not nestled into: Glasses. Smart glasses have been a reality for quite some time, with companies like Google producing "Google Glasses". Although at the time they seemed like the new product which would revolutionise our lives, they did not catch on, due to high production cost, privacy issues, lack of practical use, and poor design. Society was simply not ready for a computer mounted on the head with cameras and displays. There was a plethora of altercations which lead to Google actively discontinuing the product. Since the initial release there have been many different attempts at smart glasses. And although they are elegant solutions, they simply lack the connection to everyday life. They are still seen as cool little gimmicks but lack functionality in key areas. Many companies produce smart glasses with heads-up displays for industry-specific use cases, not for the public to use. Oftentimes these designs are also not very elegant, because they do not have to look appealing, due to not being used by the public.

As a person who has had to wear prescription glasses since the age of eight, I have always found the prospect of smart glasses very intriguing. I take quite an interest in technology and the Internet of Things (IoT). I like not only using smartphones and other gadgets but also

[1] Kevin Bartley, 2020
[2] Precedence Research, No Date

building things and analysing design processes and internal hardware. My yearning for the day a good pair of smart glasses hit the market coupled with my innate interest in technology have led me to attempt to build smart glasses myself.

At this point, I would like to thank all the people who have helped me throughout the process of this Matura project. I would like to especially thank my supervising teacher, Mr. Florian Hotz, who was there throughout the project, giving helpful feedback, constructive criticism, and especially believing in me. I would also like to thank him for supplying me with a 3D-Printer, which enabled me to design my product expertly. My friends and family were also always there to help if I needed their advice or expertise, for which I am thoroughly grateful.

2. Theory

2.1 Augmented Reality

Augmented Reality, commonly abbreviated as AR, is a form of technology which combines aspects of the real world and computer-generated things. It is a form of experience which enhances a multitude of sensory modalities. Although it is most known to enhance the visual sensor, it also encompasses the enhancement of other sensory inputs, such as sound or touch. Its objective is to augment reality with a digital reality to enhance a person's perception of reality. It does not merely encompass the display of data, but also immersive sensory stimuli, which, to the user, seem like a natural part of reality.

There is oftentimes confusion between Virtual Reality (VR) and AR. The difference lies in how much digital content is added. In VR you perceive a completely virtual reality, not connected by any means to the actual reality. It is a digital world, which is unhinged from reality. It does not enhance, but it creates an entirely digital one. AR is a mix between VR and normal reality, is also referred to as Mixed Reality (MR). The most common technological application of AR is called heads-up-display technology (HUD). The HUD exists as a graphical UI atop what you see and delivers vital information.

Thomas P. Caudell of Boeing named this technology “augmented reality” in his paper about the topic[3]. It described adding data to visual input, to enhance the user experience. But AR the way we know it today was already invented in the 1960s by a Harvard Professor, who built a head-mounted display. Since then, slowly but surely AR had started to make its way into different industries with extremely useful help. In 2014, Google was the first to release AR glasses. These were the first real application of AR, which were made for and released to the public. It did not gain much traction, with Google discontinuing the product soon after the release, due to many different problems. They did however continue producing it for industry-specific use cases, where AR has cemented itself as a viable technology. Notable blog CNET rumour the existence of Apple, another big player in the consumer technology branch, looking to release their own pair of AR Lenses[4]. While these are, at the time of writing, still just a rumour, it is an indication that major companies are looking to bring AR technologies to the masses.

[3] Thomas P. Caudell, 1995
[4] Scott Stein, 2022

2.2 AR Glasses

"AR Glasses" is the implementation of AR in wearable glasses. Often, they are also referred to as "Smart Glasses". Using a computer attached to the glasses, a display projects information, and data through the lenses of the glasses, which is then viewable by the user. In most use cases, the data is superimposed onto what the user can already see. This data can also work together with what can be seen by the user, improving the experience for the user. They are minicomputers on the frame of a pair of glasses. Many features on smart glasses are akin to those found on smartwatches, such as time, GPS data, fitness tracking and many more features.

Nowadays, smart glasses are exclusively used in industrial markets, for example in factories where precision and quick information is needed. Its practicality much overweighs its need for elegance. It has cemented itself quite well into the industrial market. After a failed attempt to bring it to the public for everyday consumption, companies like "Nreal" attempt to bring it to ordinary consumers. While it has just begun, there seem to be two choices from which companies should choose their design types.

One has a translucent display that you view in front of you. This one is attached to the front of the glasses, where you look through the display. This display directly overlaps your existing visual input and delivers data atop it. Even though this may be an elegant solution to the space problem, it does not look sleek or integrates seamlessly.

The second design has a display on the side which projects into the glass at an angle, so that it reflects off the glass into your eyes, seeming like it naturally is a part of what you are viewing through the glass. Although this design type could lead to some clunky-looking builds, it makes the projected input seem seamlessly integrated into what you are viewing. This makes it a very compelling design choice. For this reason, this design was preferred and ultimately chosen.

2.3 Concept

The concept of this project is straightforward. The goal is to build smart glasses that look original and not too abstract, so that it does not appear very jarring. It should be able to be worn in public. The functionality of the smart glasses should also be useful, although not too intricate, otherwise it would overwhelm any user. They should work with the touch of a button and scroll through different menus, for simplicity's sake.

3. Resources

In this section of the paper, the entirety of the resources used for the project is presented and explained. This includes the electrical components used in the system. The final product was a system inspired by the tutorial but deviated from it with my own design choices. While all the hardware parts were original, the code was inspired by a tutorial, and developed to the needs of the product designed for this project.

Link to tutorial: https://www.youtube.com/watch?v=FTfMG_iYbMw

3.1 Required Tools

For this project there were not only components required, but also many external tools, which I did not have before. This includes:

1. **3D-Printer**

   For this project I was fortunate enough and am thankful that my supervisor could provide me with one from the school. This enabled me to print and create my own design. To print with a 3D-printer you also require PLA plastic.
2. **Soldering Station and Other Equipment**

   I also needed a soldering station where I could solder the parts. This included a soldering iron, a sufficient amount of soldering wire, soldering flux, a wire stripper, and a general-purpose cutter to cut wires and other things.

3.2 Frame and Design

For the frame of the glasses, the original plan had been to implement a normal pair of sunglasses. It was when I found out about a modelling application called “Fusion 360”, that I decided to design my own glasses. “Fusion 360” is a commercial computer-aided design (CAD) software application by Autodesk. The application allows the design of manufacturable goods and printed circuit boards (PCBs). It is a design software used by professionals to design and manufacture any product from start to finish. I used Fusion 360 to design my glasses to my requirements and print it using a 3D-Printer.

3.3 Hardware Components

The hardware components used in the AR glasses system were inspired by a tutorial available on YouTube, as mentioned above. This tutorial was an inspiration, a sort of guideline for the AR glasses system. The hardware components include a display, a microprocessor, and other components, used to complete the circuit.

3.3.1 Display

The display used for the AR glasses system was a small 0.91-inch, 128 x 32 OLED display. This display was not to directly display information, but to reflect information off a lens so that the experience was not too harsh or overwhelming. This kind of display is unbelievably cheap and widely accessible and used for a multitude of Arduino projects.

3.3.2 Processor

For the CPU, the brain of the system, there were many different contenders. The requirements were simple. A powerful processor with many capabilities in as small a form factor as possible. An immensely popular computer chip for such “Internet of Things” is the Raspberry Pi. It is a small computer with wide areas of application. The drawback lies in the size. For the project, the general size of a Raspberry Pi computer is too large. After researching many videos for tutorials and researching parts across many websites, I settled for the ESP line of processors.

The ESP line of processors is a series of efficient microcontrollers with integrated Wi-Fi and Bluetooth capabilities. It is made by Espressif Systems and is immensely popular for a variety of IoT applications. The processors have several digital and analogue input/output pins, as well as I2C, SPI, and UART, which are communication interfaces. They also have integrated sensors such as an accelerometer, gyroscope, and temperature sensor. For its low price and minuscule size, it was a clear choice for the project. For the project, the newest processor in the line, the ESP32 was chosen.

At first, as per the tutorial, the base ESP32 processor was without any board attached to it. The processor came from AliExpress, as did most of the other parts. Due to some technical fault, the processor that was received was dead on arrival. So, instead of reordering the same processor, I ordered the Sparkfun ESP32 Thing Plus, an ESP32 board with many useful inbuilt features. Although it is a full board, it still is packed in an exceptionally small form factor. In a way, the board was a better choice, due to its built-in charging and connection port, a JST 2.0 connector for a battery, and its voltage regulator. The board also made programming a lot easier, since a lot of documentation and a GitHub repository were available for the product.

3.3.3 Other Components

The other components of the AR glasses system include a TP4056 charging board, a 140mAh LiPo, and a drawing pin, which is the interface for receiving input for the system. The TP4056 USB type C charging board was an elegant method to charge the battery since it offered a newer port, a light indicating whether the battery was empty or full, and a dedicated voltage generator. Even though the Sparkfun board also had this, it was easier to use the TP4056, because I did not need the JST 2.0 port, which was restricting in terms of programming. For the input, I used a simple drawing pin input. The ESP32-Processor can detect charges from anything, like the electrical charge of your finger. So, any conducting surface can be used as input, so long as the threshold for what counts as input is sufficient. For the Glass, a simple thin sheet of plexiglass foil was used, reinforced by a repurposed smartphone protection screen. This was glued to the frame.

3.4 Software Components

As mentioned, the software components of the project, the code with which the AR glasses system works, were based on a GitHub repository of the tutorial, and enhanced to meet the requirements of the project. The written code consists of two parts: One part is the Android application code; the other part is the Arduino IDE code for the ESP32 processor.

3.4.1 Android Studio Code

This explanation of the code will not be too detailed, since it would take up a lot of space. To put it simply, the Android Studio code is the code that feeds data to the ESP32 processor, which in turn displays it. It works on the side of the phone, outputting data to the ESP32. This defines the data which the ESP32 processor can pull and display.

Here is an example code snippet from the MainActivity.java file:

```
public static void onHome() {
    SimpleDateFormat sdf = new SimpleDateFormat("HH:mm",
Locale.getDefault());
    String time = sdf.format(new Date());
    sdf = new SimpleDateFormat("dd MMM", Locale.getDefault());
    String date = sdf.format(new Date());

    mainScreen(time, date, "-1", "");
}
```

This snippet defines the layout of the home screen. It defines the time, followed by the date and the current weather temperature.

3.4.2 Arduino IDE Code

The Arduino IDE code is the code for the ESP32. The language of the ESP32 is C++. This includes powering up, connecting to the phone, retrieving data from it, and displaying said data, which includes fonts, screens, and more. The code is what displays the data in an orderly fashion and makes the glasses work. Since the display reflects onto a lens, the data displayed must be mirrored. Here is a snippet of the IDE code:

```
void screenCall(String from) {
      // Restart offset when three dots reached but give two more updates for larger delay beetween
three dots and back to 1 dot
      if (__Offset >= 6) __Offset=0;

      u8g2.clearBuffer();
      drawSymbol(0, 9, 260, 1);

      setFontSize(9);
      if      (__Offset == 0) drawString(12, 9, "Calling.");
      else if (__Offset == 1) drawString(12, 9, "Calling..");
      else                    drawString(12, 9, "Calling...");
      setFontSize(8); drawString(0, 22, from);

      u8g2.sendBuffer();

      // Increase dots offset
      __Offset++;
    }
```

This snippet shows the display of the call screen. It shows how the calling animation works and how it draws it on the OLED screen.

4. Process

This section will dive deeper into the considerations of each decision and all the decisions in this project, serving as a guide for reproducing the smart glasses. A GitHub repository with all necessary files was created to enable this.

Link to GitHub repository: https://github.com/abhinavb03/SmartGlasses

4.1 Part Selection

After discussing and agreeing what exactly I wanted to do with my supervisor, I began to look for build guides and tutorials on the matter. Upon searching I found an overabundance of tutorials available on said matter. I spent hours researching them, and finally found one which had parts accessible. Following that, I listed the parts and started searching the internet on where they would be accessible. Since I had begun in February 2022, I could order from websites like AliExpress which had long delivery times. A multitude of the parts required in the tutorial were available on AliExpress. In the following chapter, all the products used in the AR glasses system will be listed along with their costs. In addition to the parts, I needed to buy a soldering iron station, wires, and soldering flux. I also had to buy accessories and plastic for the 3D-Printer, which was used to print the case of the glasses. During the entire process I had to continually reorder items that broke, were depleted or not suitable.

4.2 Material List and Production Costs

In this section, all the parts used are listed in a table, along with the link to buy them. The prices may vary, so the price listed next to them is the current price at the date of writing: 27.12.2022.

Product	Use	Picture	Price in CHF	Link
Sparkfun ESP32 Thing Plus	Processor		23.90	https://bit.ly/sparkfunESP32
0.91" OLED Display	Display		1.40	https://bit.ly/oleddisplay
TP4056 LiPo Battery Charging Module	Charging Module		0.45	https://bit.ly/tp4056batterycharger
140mah LiPo Battery	Battery		3.60	https://bit.ly/lipobattery1
Switch	Power Switch		1.55	https://bit.ly/powerslideswitch
Drawing Pin	Input Button		3.95	https://bit.ly/input-Button
Lens	Projection Lens		0.42	https://bit.ly/focallens

The sources of the pictures can be found in the bibliography.

4.3 Building Process

This section contains an in-detail description of the building process of the project. For a day-to-day detailed journal of what I did, see the work journal. In this section, the different thought processes and ways of tackling tasks are emphasized and explained.

After creating the parts list and discussing it with my teacher, I created a rough timeline where I outlined what I would do and when. I also got the 3D printer from my supervisor shortly after I met with him in person. I did underestimate the hours of work I would have to put into the project since all of it was very new to me. Having ordered from AliExpress it took quite a while for all the products to arrive. In the meantime, I started experimenting with the 3D printer, printing different things, and learning how the 3D printer works. I also started learning Fusion 360, which was quite troublesome at first but became easier the more I worked with it.

I had never used a soldering iron or any such electrical equipment in my life before, so when that arrived, I started soldering some wires together and creating small circuits, so that I could learn how the soldering works. I learnt many things related to soldering, such as how to isolate wires, how to intertwine them and solder them, and many more things. I did not comprehend what scale I would be working at, so I did underestimate what fine motor controls the soldering would take. For all the new skill sets I had to learn, no matter how mundane, I always consulted YouTube tutorials, learning some tips and tricks. It was quite helpful for soldering.

After the parts finally arrived, I closely followed the tutorial and re-sketched the circuit which I had to build. After double-checking that everything was properly sketched, I recreated the circuit on a breadboard. I soldered all the parts that had to be soldered directly to the processor or display. It is important to note, that at this stage, I imitated the tutorial with high precision. That meant that I had specially ordered a voltage regulator and had soldered separate capacitors to the voltage regulator. The processor I was using at this point was just the base processor, without anything attached to it. This also meant that for me to upload a program from the Arduino IDE, I had to solder it to a special USB-to-UART device, which would let me connect the processor to the computer. It was when I connected the processor to the USB device that I found out that it was dead. For some unknown reason, it did not work. Instead of losing time, I went online and tried to order the chip locally to avoid long delivery times. There I found the Sparkfun ESP32 Thing Plus. It was a board which had many of the things I had to externally solder built in, saving me a load of trouble. Seeing that it would integrate effortlessly into my project, I enthusiastically ordered it.

Since the board was all-inclusive, I did not need the tutorial. I soldered the display to the processor and the charging module along with the switch to the port. It took many tries to find the right port, but when I found it, I was finished with soldering. To make the soldered connections last, I put hot glue on the connections. I soldered a drawing pin to the board, using it as a button. The ESP32-Processor can detect anything that has an electrical charge, including the charge of a finger. So instead of building a full-fledged button, I just added a drawing pin as a button, decreasing the wires used and space used.
In the final assembly, I had to attach the lens in a way, that it would reflect properly and be visible. Then I noticed, that for every person the focal length is different. This meant that if I adjusted the distance and angle of the lens for myself, meaning that others may not be able to use it. For demonstrative purposes I decided that I would make one focal length for reference. In addition to that, I developed a mechanism for people to configure the lens in a way that they could also see something. This mechanism would allow anyone to use the glasses.

4.4 Coding Process

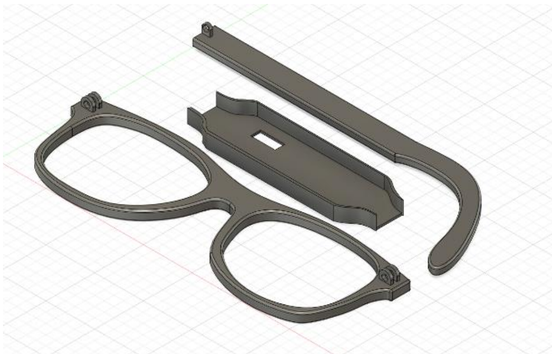
Unlike the building process, the coding process was not very straightforward. Since the code which the tutorial provided was to be modified, I had to analyse it before I could further develop it. Before starting the project, the only coding languages I knew were python and some rudimentary HTML. I had no clue about JavaScript or C++. This was one of the biggest blockades I had during the project.

The first thing I had to do was to learn the languages. I started on the Arduino IDE side, so I started looking through the code, searching up the things I did not understand in C++. Thankfully, I knew the bare minimum due to my prior knowledge of python. After perusing through the code, I found a few syntax errors and corrected them. I also modified the code a tiny bit with the help of Stack Overflow. A big issue in the code was, that the port it had defined as the Battery_GPIO port was non-existent on my board. I tried many of the ports and it did not work, causing the processor to fall into a reboot loop. I finally had no choice but to manually go through all the ports from zero on, till I found the port. In the end, it was GPIO 10 which worked for the port labelled 3v3 on my board. This was the board that delivered power to the display and took power from the TP4056 module.

Another issue was the button. In the tutorial, the person uses a normal drawing pin used to pin objects to walls, as an input button by simply soldering it to a GPIO. It did not seem very credible so I read up on it and found out, that the ESP32 can recognise voltage differences caused by our touch and register them as input. I tried implanting, to no avail. It would not give the output it had to. It once again required a bit more decoding and code changing. It did not work the first time, so I double-checked it with some online tutorials and redid it again. Then it worked.

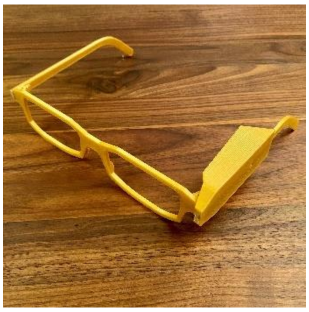
On the other side, the Android Studio code was quite confusing. After working with C++ for a while, JavaScript was quite a change of pace. At first, the code worked bug-free. I deployed the built APK from the studio to the phone. There it worked fine as well. Connecting the phone to the ESP32 worked as well. Now the problem was a different one. It connected and showed some demo screens, but nothing else. This was not something with which I was happy. After digging both in the IDE code as well as in the Arduino code, I found 500 lines deep in a specific file, that a substantial chunk of the important code had for some reason been turned into a comment. So, I had to bulk uncomment the lines of code. This caused a tsunami of errors. I had to go line for line through the code. After countless hours of debugging, I could fix all the errors. Now finally in its full functionality, all the code worked.

4.5 Glasses Design Process



An especially important part of the project for me was the design of the glasses. Throughout the process, I found a new liking for product design. I enjoyed the design very much and took my careful time to polish my designed glasses.

I started by closely analysing the design of my everyday glasses. This was the design I was drawn to when I chose them at the shop, and the one I wanted to try and imitate for the AR glasses. I used Fusion 360 to draw a very closely accurate sketch of the frame of the glasses. I also designed a case for the board and the various other components. For the base aspect ratio, I fixed it in Fusion. For any further stretching and parameter changes, I did it in the 3D-printing app Cura. Like any other product, over time I produced multiple prototypes.



Prototype 1: For the first prototype, I did it with a pre-defined STL file I downloaded from the internet. From the tutorial's GitHub repository, there was a case given by the tutorial, which I printed and attached to the glasses. For this prototype, there was no hinge mechanism or any way of connecting the frame, so I just glued it together.

Prototype 2: This was the first one I designed myself. I played around with the size of the frame since my everyday glasses were a bit on the smaller side. It turned out to be too large. I also played around with the infill of the plastic. It turned out to be too low, so I increased it to 100% for the following prototypes.

Prototype 3: This was the prototype where I added a rudimentary hinge mechanism. It turned out to be fragile, so for the next one, I made it sturdier. I did not change the size for this one, although I did meddle a bit with the thickness of the frame. For the case, I designed a case that would fit all the parts and not look too jarring so that it would attach seamlessly.

Prototype 4: In this prototype, I drastically changed the size of the frame. I matched the size of the frame to my glasses. These looked good, but they did not look quite as polished as I wanted them to be. This was also the first prototype where I added the self-designed case to the print. I played with the height a little but since I was not ready to assemble the other components, I left it at that.

Prototype 5: This prototype was a complete redesign. Although I liked the other design, something inexplicably seemed off with it. The aspect ratio and thickness felt off. I went back to the drawing board and produced a better design. This time it looked clean. It started to shape up into the final prototype.

Prototype 6: This prototype was the final prototype of the frame. I fine-tuned the aspect ratio of the glasses, so it looked more appealing. I had found the perfect sized screw and screw cap. The thickness of the hinge was a little too much, so I adjusted it as well and made it thinner so the screw could be used. It worked out perfectly. For final design of the case, I modified the previous prototype by adding a hole for the switch and the input button. I also adjusted the height of the case, so all the wiring and components fitted snuggly inside.

5. Challenges

As mentioned, this project did not at all go smoothly from start to finish. There were many obstacles. The very first problem was with the processor being dead and not working. This was for the better because I then stumbled upon the ESP32 board which was much more helpful.

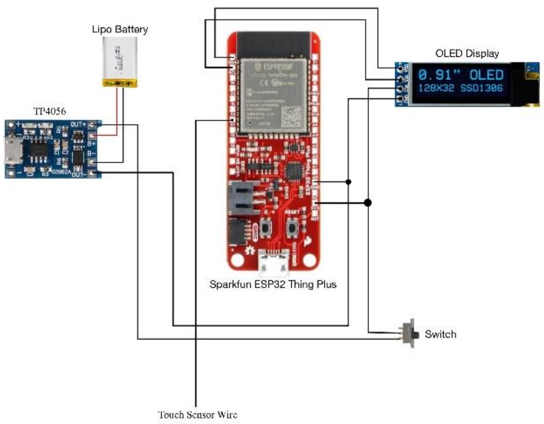
The code took an exceptionally long time to fix the problem with the 0 voltage Battery_GPIO read. I could not figure out why it did this. I had a lot of documentation on the product, but the Arduino IDE did not accept an alphabet as a GPIO port so at the end I had to resort to manually trying all the ports. After that, the button mechanism shown in the tutorial did not work, so I had to implement my pushbutton and change the Sensor threshold for it to work.

In the Arduino IDE code, there were many minor changes I had to make, and there were a few syntax errors which took a while to fix. There were also some errors in Android Studio which I had to fix. At first, the project could not even be emulated because the project did not recognise its manifest file. I had to recreate the manifest file, the file which outlined the parameters of the project, and reformatted it so that the project could work.

The essential part of the code did not work because somewhere in the activity file the important part was commented on. After a few days the code worked for a brief time before failing again, so I headed back to Android Studio, it turns out that there was only one “public static void” function for the home screen, and none for the other ones. I tried going on Stack Overflow, but to no avail. I tried getting help from my parents but to no avail. The functionality of the other screens was flawed and limited.

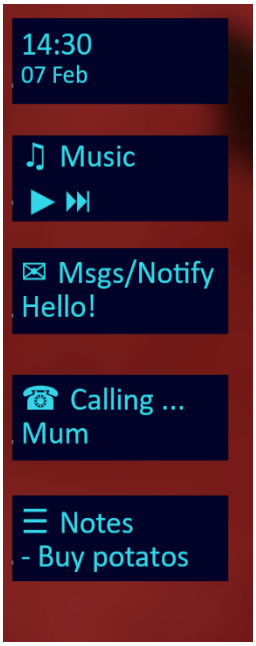
The issue with the lens was also something that persisted. It became painfully clear once I tried fitting the lens in the correct position for it to display, that if I could see it with glasses, I could not see anything without glasses. I researched the problem and found that focal lengths are different for different people. It turns out I could not configure it in a way that anyone could wear and see it. It had to be customised.

6. Final Product



This section presents the feature of the final product, what it can do and how it does it. The features are presented with pictures and so is the final design of the glasses. On the right is the circuit of the final design, showing the parts used. The battery is connected the TP4056 charging module. This is connected to the Ground (GND) and to the switch which then is connected to the 3v3 port. The OLED display is connected to the SCK (Clock) port, the SDA (Data), the 3v3 port, and the GND port. The touch sensor wire is only connected to GPIO 27, one of the capacitive touch ports of the ESP32.

6.1 Features



The final features of the glasses are simple yet useful. Although some additions to the code were foreseen, the complexity of the consequences did not allow for that. The code is taken from the tutorial's GitHub repository and is visible for all to see.

The following picture is also from the GitHub repository, displaying the different screens and features. The controls are simple. To change the screen, double tap. The screen turns off after 8 seconds. A long tap brings up the last screen that was projected. The notes can be changed in the Android app. In the music screen, a single tap either plays or pauses the music, and a double tap plays the next track. When messages roll in you can see them when on the messages screen. When someone is calling you, you can also see that in the call screen. The list screen is a rudimentary feature where you can note things down in a list in the app. The first item on the list will then be displayed on the screen.

6.2 Design

The design of the glasses is unique. Many models seen online are quite bulky, with big extensions protruding from the original frame. Although the computer is not seamlessly integrated into the frame, the end size is sizeably more appealing than other self-made models. The Glass used is a thin sheet plexiglass, reinforced by a repurposed smartphone screen protector to give some thickness and heft. The display projects into a lens which makes the data viewable at a certain distance. The plastic is sanded and smooth, allowing for comfortable wear. The weight is quite moderate, not too heavy to wear. The glasses have a hinge mechanism, allowing them to fold like any other specs. The charging port is in the back with easy access. On the outside, there is a button for input and a switch to turn the computer on and off. The case seen in the final prototype of the glasses is a bit elongated since the original size did not allow for enough room. The button and the switch are hot-glued to the 3D-printed case, and the rest of the components are attached with tape.

Since every person has a different focus length at which they can see things, every person would need a different lens with a different focal length. With the final product, I created a reference length where you can see the things being displayed at a certain length, without having to wear them. With professional AR glasses, such as the Nreal Air, you can adjust the size of the projection which allows for customisation that way.

7. Conclusion and Evaluation

At the beginning of the project, it was my goal to create a pair of AR glasses, which could be functional, all the while looking optically pleasing. I wanted to design glasses myself and make a full-fledged product by myself from start to finish. Looking back, I think I have successfully accomplished my goal. The smart glasses look quite pleasing with an original design and convenient functionality. Of course, the code being pre-made and there being so many supports both online and offline made the whole task easier. However, just like any project, there were some major roadblocks that at times, did make for some frustration, for example, when the code did not even compile because of some missing curly braces. Alleviating such problems took anywhere from 5 minutes to 2 hours, depending on how painful they were. This resulted in frustration and annoyance, but when solved, caused great relief. Undertaking this project showed me how complicated it is to bring a simple idea to fruition and improve upon it. It gave me a new desire to push on until the given problem is overcome. It brought me patience and perseverance.

I must admit that I underestimated the learning curve this project had. Going into the project, I could not fathom how troublesome the coding part could be. The tutorial made it seem as if anyone could do it. It was not truly clear what all skillsets one needed to have. Although it caused some dismay, I am glad I had a chance to learn many things I always have wanted to learn. Throughout the project, I learned how to solder, how to use Fusion 360, how to use a 3D printer, how to program C++, how to program JavaScript, and how to use Adobe InDesign. To learn all these things, I used a variety of online courses such as Khan Academy, YouTube, and Stack Overflow. It was thanks to these resources, that this project was possible. I am happy with the outcome of the product and the project overall, but I had some extra features planned for the glasses. One of them was to implement a speaker and a microphone so that you could take calls directly from the glasses. Another one was to implement a camera so that you could record videos and take pictures from the glasses directly. These were all ideas which I thought about implementing, but due to the time needed, I had to drop them. It would have been undoubtedly preferable if I could have implemented these features, but since my knowledge in C++ and JavaScript was not rife enough for me to implement such features, I could not do it in the project.

What I do want to do, is implement it in the future. For me, this project has also been an awakening. It made me realise how new a market this is, and how there is so little competition.

There are almost no commercially usable AR glasses that do not have a major compromise. It is my intention to further broaden this project into something bigger.

Overall, this Matura paper was an unforgettable experience. From relaxed writing to sleepless nights decoding some evil syntax errors, I got first-hand experience of how it is to build a product from start to finish. At times, my motivation peaked, and I would be in the zone writing and debugging like wild. Other times It felt like there was no hope. There was nothing left to be done but give up. Even when I was at rock bottom, I learned something crucial. Perseverance. This project made me learn the necessary perseverance and time management needed to get things done. It was not easy, taking up such an ambitious project, all the while having school exams and much more to deal with. I could improve my time management even more, but overall, I am quite happy with the way things turned out. This project has given me some important skills and for those, I am incredibly grateful.

8. Bibliography

Caudell, T.P. (1995). *Introduction to augmented and virtual reality*. [online] SPIE. Digital Library. Available at: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/2351/0000/Introduction-to-augmented-and-virtual-reality/10.1117/12.197320.short?SSO=1 [Accessed 13 Dec. 2022].

Stein, S. (2022). *Apple AR, VR Headset Expectations: When Will It Show Up?* [online] CNET. Available at: https://www.cnet.com/tech/computing/apple-ar-vr-headset-rumors-m2-2022/ [Accessed 13 Dec. 2022].

Bartley, K. (2020). *Big Data Statistics: How Much Data Is There in the World?* [online] Rivery. Available at: https://rivery.io/blog/big-data-statistics-how-much-data-is-there-in-the-world/ [Accessed 21 Dec. 2022].

www.precedenceresearch.com. (n.d.). *Wearable Technology Market Size, Trends, Growth, Report 2030*. [online] Available at: https://www.precedenceresearch.com/wearable-technology-market [Accessed 23 Dec. 2022].

Android Developers. (n.d.). *Configure your build*. [online] Available at: https://developer.android.com/studio/build#sourcesets [Accessed 23 Nov. 2022].

Arduino Forum. (2018). *[SOLVED] Measure the voltage on the same battery powering the Arduino.* [online] Available at: https://forum.arduino.cc/t/solved-measure-voltage-on-the-same-battery-powering-the-arduino/500237 [Accessed 15 Nov. 2022].

Arduino Project Hub. (n.d.). *Augmented Glasses*. [online] Available at: https://create.arduino.cc/projecthub/rev_atomic/augmented-glasses-a9db7d?ref=part&ref_id=11332&offset=1 [Accessed 26 Oct. 2022].

docs. Arduino. cc. (n.d.). *A Guide to Arduino & the I2C Protocol (Two Wire) | Arduino Documentation*. [online] Available at: https://docs.arduino.cc/learn/communication/wire [Accessed 19 Oct. 2022].

ESP32-WROOM-32 Datasheet. (n.d.). [online] Available at: https://cdn.sparkfun.com/assets/learn_tutorials/8/5/2/esp32-wroom-32_datasheet_en.pdf [Accessed 15 Oct. 2022].

GitHub. (2022). *SparkFun ESP32 Thing Plus*. [online] Available at: https://github.com/sparkfun/ESP32_Thing_Plus [Accessed 14 Oct. 2022].

Hackster.io. (n.d.). *Arduino Smart glasses*. [online] Available at: https://www.hackster.io/ardineer999/arduino-smart-glasses-a32162 [Accessed 24 Aug. 2022].

Hayes, A. (2022). *Augmented reality*. [online] Investopedia. Available at: https://www.investopedia.com/terms/a/augmented-reality.asp [Accessed 10 Sep. 2022].

HeyTech (2020a). *DIY fully working Smart glasses under 20$! - Arduino/ESP 32 Project*. [online] www.youtube.com. Available at: https://www.youtube.com/watch?v=FTfMG_iYbMw [Accessed 4 Aug. 2022].

HeyTech (2020b). *Programming ESP32 (SMD) with USB TTL/UART - Tutorial*. [online] www.youtube.com. Available at: https://www.youtube.com/watch?v=lH7hVTUL7W8 [Accessed 14 Oct. 2022].

Kraus, O. (n.d.). *u8g2reference · olikraus/u8g2 Wiki*. [online] GitHub. Available at: https://github.com/olikraus/u8g2/wiki/u8g2reference#drawglyph [Accessed 6 Nov. 2022].

Last Minute Engineers. (2022). *ESP32 Pinout Reference*. [online] Available at: https://lastminuteengineers.com/esp32-pinout-reference/ [Accessed 23 Oct. 2022].

learn.sparkfun.com. (n.d.). *ESP32 Thing Plus Hookup Guide - SparkFun Learn*. [online] Available at: https://learn.sparkfun.com/tutorials/esp32-thing-plus-hookup-guide [Accessed 13 Oct. 2022].

Pesce, M. (2019). *Augmented Reality – The Past, The Present and The Future*. [online] The Interaction Design Foundation. Available at: https://www.interaction-design.org/literature/article/augmented-reality-the-past-the-present-and-the-future [Accessed 12 Oct. 2022].

Poetker, B. (2019). A Brief History of Augmented Reality (+Future Trends & Impact). *G2*. Available at: https://www.g2.com/articles/history-of-augmented-reality [Accessed 27 Sep. 2022].

Random Nerd Tutorials. (2019). *ESP32 Pinout Reference: Which GPIO pins should you use? | Random Nerd Tutorials*. [online] Available at: https://randomnerdtutorials.com/esp32-pinout-reference-gpios/ [Accessed 8 Oct. 2022].

Sparkfun. (n.d.). *ESP32 Thing Plus v2.0 Pinout*. [online] Available at: https://cdn.sparkfun.com/assets/learn_tutorials/8/5/2/ESP32ThingPlusV20.pdf [Accessed 13 Nov. 2022].

Stack Overflow. (n.d.). *Android Studio Manifest Missing Error In Gradle*. [online] Available at: https://stackoverflow.com/questions/17096983/android-studio-manifest-missing-error-in-gradle [Accessed 23 Dec. 2022].

Stack Overflow. (n.d.). *java - Android - Missing Application Subclass*. [online] Available at: https://stackoverflow.com/questions/13124098/android-missing-application-subclass [Accessed 20 Dec. 2022].

Stack Overflow. (n.d.). *XML - Android studio can not recognize manifest file*. [online] Available at: https://stackoverflow.com/questions/72229063/android-studio-can-not-recognize-manifest-file [Accessed 21 Nov. 2022].

Wikipedia Contributors (2018). *Augmented Reality*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Augmented_reality [Accessed 13 Sep. 2022].

Wikipedia Contributors (2019a). *I^2C*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/I%C2%B2C [Accessed 3 Nov. 2022].

Wikipedia Contributors (2019b). *Smartglasses*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Smartglasses [Accessed 1 Oct. 2022].

www.electronicwings.com. (n.d.). *Digital GPIO of Arduino | Arduino*. [online] Available at: https://www.electronicwings.com/arduino/digital-gpio-of-arduino [Accessed 16 Oct. 2022].

www.microchip.com. (n.d.). *GPIO Issues with VBat Pin | Microchip*. [online] Available at: https://www.microchip.com/forums/m1103729.aspx [Accessed 16 Nov. 2022].

www.youtube.com. (2019). *Product Showcase: SparkFun ESP32 Thing Plus*. [online] Available at: https://www.youtube.com/watch?v=nvpeqxgSIE0 [Accessed 17 Oct. 2022].

Picture of Sparkfun ESP32 Thing Plus [online] Available at: https://eu.mouser.com/new/sparkfun/sparkfun-thing-plus-xbee3/ [Accessed 18 Dec. 2022]

Picture of 128x32 OLED Display[online] Available at: https://www.elementzonline.com/blog/oled-ssd1306-i2c-display-128x32-interfacing-with-arduino-uno [Accessed 18 Dec. 2022]

Picture of Nreal Air AR Glasses [online] Available at: https://www.amazon.com/Nreal-Micro-OLED-Augmented-iOS-Consoles-Compatible/dp/B0BF5LKP5Q/ [Accessed 18 Dec. 2022]

Picture of TP4056 Charger [online] Available at: http://www.piready.co.za/TP4056-LIPO-CHARGER [Accessed 18 Dec. 2022]

Picture of Lithium Battery [online] Available at https://www.ebay.com/itm/274650266020 [Accessed 18 Dec. 2022]

Picture of Slide Switch [online] Available at: https://uk.farnell.com/c-k-components/s101031ms02q/slide-switch-spst-6a-250v-tht/dp/2931779 [Accessed 18 Dec. 2022]

Picture of Drawing pin [online] Available at: https://www.desertcart.no/products/43136341-mroco-push-pins-thumb-tacks-steel-thumbtacks-sliver-thumb-tack-flat-head-push-pin-office-thumbtack-clear-push-pins-for-cork-board-for-home-school-sharp-steel-points-3-8-inch-head-box-of-600 [Accessed 18 Dec. 2022]

Picture of lens [online] Available at: https://de.aliexpress.com/item/33011072917.html?gatewayAdapt=glo2deu [Accessed 18 Dec. 2022]

Icon of Smart glasses [online] Available at: https://iconscout.com/icon/smart-glasses-50 [Accessed 18 Dec. 2022]