Audio Augmented Reality Glasses

By

Nikita Vasilyev

Sunny Chen

Evan Chong

Final Report for ECE 445, Senior Design, Spring 2025

TA: Aishee Mondal

07 May 2025

Project No. 14

Abstract

The primary purpose of this project is to reduce search time for identifying unknown objects.

The Audio Augmented Reality Glasses (AARG) are a pair of glasses that are capable of taking pictures of what the wearer is observing and feeding back an audio prompt describing the object. The total time taken by the process averages around 7-8 seconds, which is considerably faster than searching online.

The device uses a push button to prompt an image to be taken and a microcontroller that sends the image to a developed mobile application, which handles the image classification. The application then sends the appropriate audio prompt over Bluetooth to the microcontroller, which outputs the audio to a pair of speakers.

Contents

1. Introduction

1.1 Problem

Have you ever seen an object that piqued your interest, but you didn't have an efficient way of researching what it was? Repeatedly searching online to identify the subject can be a lengthy and tedious task, and this is the problem we seek to address. Our solution is meant to enlighten our users about unknown plants, animals, or objects in any setting they are observing.

Furthermore, an auditory aid can be further applied to helping the visually impaired, giving context to the surroundings that they would otherwise be unable to fathom or see. A multifaceted solution to spatial object recognition problems is what we are trying to accomplish with the AARG glasses.

1.2 Solution

Our project idea stems from the surge of AR prototype glasses being introduced over the past year. We are planning to create our own glasses, but in contrast to those on the market, ours will focus on the audio experience of the user. These glasses will have the explicit capability of capturing images of objects and relaying this information to an application that will process these images in the backend. The application will then send an explanation of the object back to an audio device on the glasses (either a speaker or a bone-conducting device).

The glasses will essentially work as a digital tour guide, with the explanation of the object being auditory rather than visual. The use case we have decided to tackle is a botanical tour guide, but the purpose is to create a platform that other applications can utilize for their objectives. Other objectives include, but are not limited to the following:

- Accessibility for the visually impaired
- Language translation
- Tourism and museum tour guide
- Security and Surveillance

1.3 Visual Aid

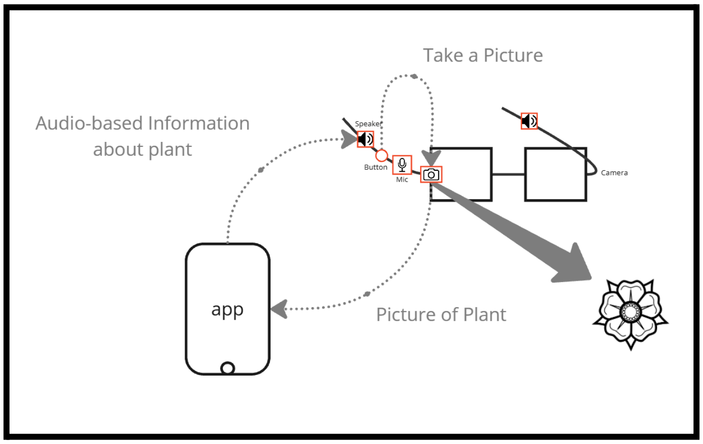


Figure 1: Visual Depiction of Example User Flow

1.4 High-Level Requirements

- **Fast Computation Time**
    - The time taken for the user to obtain audio feedback from the glasses system should not exceed 12 seconds. This value was obtained from experimentation of timing LLMs trying to identify unknown plants, along with references based on the summation of GPT backend processing time with hardware communication relay times. Reach Goal: 6 seconds
    - Conclusion: Out of 40 trials, all 40 of them were below our threshold of 12 seconds (ranging between 5 to 10 seconds). 20% of these trials also met our reach goal.

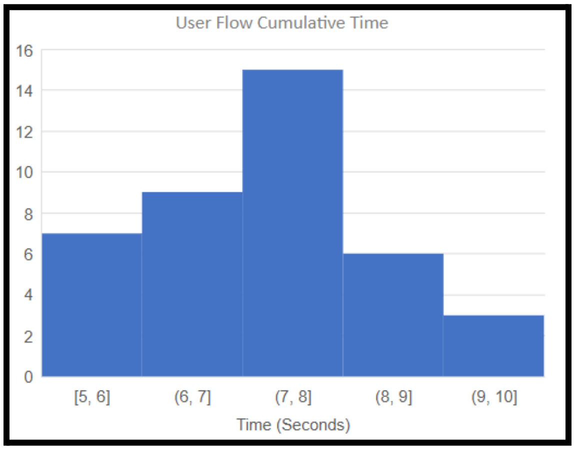


Figure 2: Bar Graph of User Flow Cumulative Times for 40 Trials

- **Accuracy**
    - The final prototype should be able to correctly identify plants 85% of the time. This will be based on our application subsystem, which will use a model to determine the plant's classification. We will test the device on 20 plants and expect to get 17 out of the 20 correct. Reach Goal: 90% accuracy
    - Conclusion: Out of 40 trials, 35 of the tests classified the plants correctly, resulting in an accuracy percentage of 87.5%.

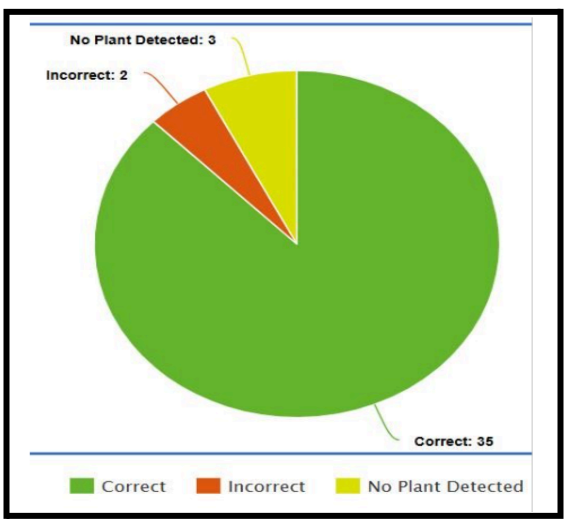


Figure 3: Pie Graph of 40 Plant Classification Trials

- **Final Prototype Weight**
  - The final prototype will weigh no more than 200 grams. We obtained this value from researching similar augmented reality headsets and aim to make our final prototype lighter than headsets on the market, but still lightweight enough to be comfortably worn, similar to commercial glasses, throughout the day. Reach Goal: 100 grams
  - Conclusion: Weighing the final prototype gave us a weight of 136 grams.

Figure 4: Weight of Final Prototype

2 Design

2.1 Block Diagram

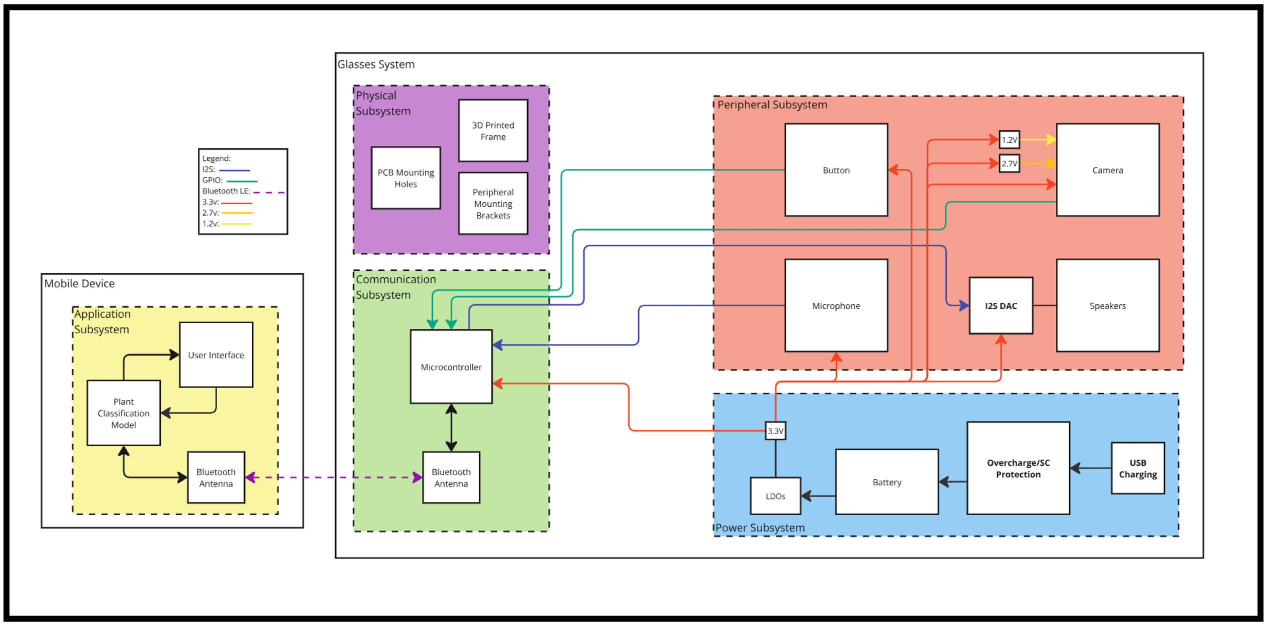


Figure 5: Block Diagram of AARG Subsystems

This block diagram depicts the subsystems present within the glasses and the mobile device. The glasses platform contains the physical, peripheral, communication, and power subsystems. The mobile device platform contains the application subsystem. The peripheral subsystem consists of all devices that interface with the user and the outside world (speaker, microphone, button, camera). The power subsystem is used to power all the sensors and devices onboard the glasses platform. The communication subsystem uses the ESP32 and Bluetooth antenna to facilitate communication between the mobile device and the peripherals. The physical subsystem consists of mounting holes and brackets for all devices alongside the frame in which they will all fit. Within the mobile device, the application subsystem is tasked with using the peripherals to complete an app-driven task.

2.2 Physical Subsystem

2.2.1: Description

The purpose of the physical subsystem is to create an enclosure for all electronic components such that they can fit into a pair of glasses. This will be done by designing and 3D printing a set of glasses with hollowed-out arms and custom dimensions to accommodate the PCBs, peripherals, and battery clearances. The mounting of the components will be facilitated by pre-planned pilot holes for each part. This subsystem largely contributes to the high-level requirement of wearable total weight. This is done by eliminating unnecessary mounting mechanisms and using lightweight plastic to save weight within the enclosure. Additionally, a significant goal of this subsystem is to ensure the wearability of these glasses. The wearability requirement consists of the glasses not containing any abrasive edges that could scratch the user during wear, having the correct dimensions according to the measurements of the average ear size, and the size of the arms/frame.

2.2.2: Requirements and Verification

1. Components are secured firmly in place during normal user operations. There must be no internal movement of components when shaken at a rate of ½ meter per second.
   Result:
   Upon performing the test as described in the R&V table, we were not able to notice any notable change in the dB of the ambient volume in the room. As a result, no rattling or disturbances were detected when shaking the glasses

2. The surface of the glasses does not contain any abrasive textures/sharp edges. The user must not be able to receive any cuts or scratches during normal operation.
   Result:
   When performing the test as described in the R&V table, there were no remnants of the paper towel left on the glasses. Additionally, there was no damage done to the paper towel.

2.2.3: Design Alternatives

There were several iterations of the dimension of the physical design. These design changes were small. Mainly moving mounting holes to accommodate new PCBs or widening the frame to ensure the glasses could comfortably fit on the face.

2.3 Peripheral Subsystem

2.3.1: Description

Our peripheral subsystem consists of a microphone, a camera, a push button, and speakers, whose schematics are shown above. All components in the peripheral subsystem receive a 3.3-volt signal from the power subsystem to act as their operating voltages. For the camera, we chose to use the Arducam OV2640 CCM, which will communicate with the microcontroller via the I2C protocol. The camera has built-in JPEG encoding to aid with our application subsystem and also has a 1600x1200 color resolution. To ensure the camera can accurately communicate with the microcontroller, we will also implement a camera driver with an external clock set at a 24 MHz frequency. This clock frequency is the typical value for our chosen camera and should ensure proper functionality. A more detailed description of the microcontroller is contained in section 2.5.

The push button in our peripheral subsystem works as a way of commanding the camera to take a photo. The push button will be connected between an active high signal and ground; when the button is pressed, it will send an active low signal to the microcontroller, and when it is not pressed, the microcontroller will receive a low signal. The signal from the button will be integrated within the implemented camera driver to perform the desired camera behavior.

The microphone we chose to implement is the ICS 43434. The microphone outputs digital signals, so it communicates easily with the microcontroller via the I2S interface. The microphone will be connected to the general-purpose input/output (GPIO) pins of the microcontroller, which will also have an implemented microphone driver to appropriately handle audio prompts from the user.

The speakers we chose to use for this project are ADS01008MR-LW100-R. These speakers run on the I2S communication protocol and will be connected to the microcontroller's GPIO pins. Another driver for the speakers will be developed and implemented in the microcontroller to ensure adequate handling of audio signals. The microcontroller emits digital audio signals, and the speakers emit audible analog signals; hence, the inclusion of a PCM5100APW digital-to-analog converter (DAC) in the speaker DAC schematic. The DAC acts as a bridge between the microcontroller and the speakers, ensuring the speakers will receive the appropriate analog signals. The DAC is also paired with a network of coupling capacitors to mitigate the effects of any potential DC offset and ensure only the desired AC signals are passed through to the speakers.

The purpose of the peripheral subsystem is to communicate with the user and the outside world. This will be done via the push button that the user physically presses, the microphone that the user speaks into, the camera that takes photos of the outside world, and the speakers that emit audio to the user. These components will be soldered onto the main PCB and encased within the physical subsystem for protection. The pushbutton or microphone will enact an instance of the camera to take a photo, which will then be fed through the application subsystem. The speakers will then receive and project the appropriate audio via the DAC after the application subsystem is finished with its task.

The peripheral subsystem contributes heavily to the general operation of our project. The camera enables the overall glasses system to have a photo to process, and the speakers and push button enable the user to prompt the system and receive the desired information efficiently.

2.3.2 Requirements and Verification

1. Only the push button should initiate the glasses system: no other process should start it, and it should not start randomly.

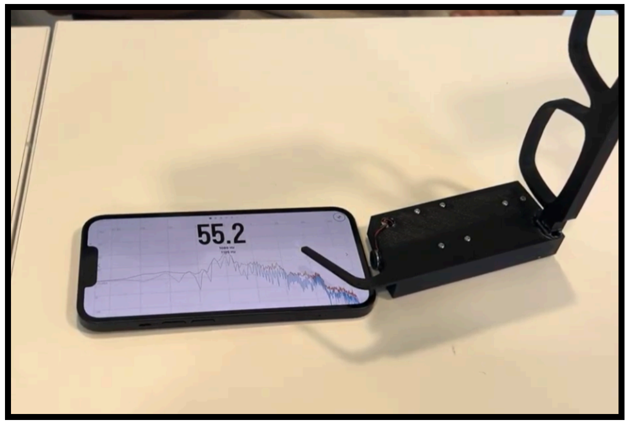
   Result:

   Considering that the user flow is successful in its entirety and no input other than the interact button notification from the ESP32 begins the user flow, it can be logically deduced that this criterion is satisfied.

2. The speakers must output audio that can be comfortably heard by the user. Research would indicate that this range is around 50 dB.

   Result:

   Unlike the desired 50 dB at 1", we were only able to achieve a measurement of 50 dB at ¼ ". This resulted in quieter audio than ideal.



2.3.3: Design Alternatives

One design alternative that we were required to take was to use the Adafruit breakout board for the PCM5100. This was because when receiving the final round PCB with the corrected DAC pinout, we had a faulty component and were unable to test the functionality of the DAC circuit. By the time the replacement component had come in, we had only two days before the demo, which did not allow us enough time to debug the PCB and determine why the DAC circuit on the 4th round PCB was not functional. The result was functionally the same as planned, with the caveat of using a breakout to house the DAC.

2.4 Power Subsystem

2.4.1: Description

The power subsystem is responsible for providing 3.3V of regulated voltage to the microcontroller (ESP32-S3-WROOM-1-N16R8), the microphone (ICS 43434), the DAC (PCM5100APW), the camera (Arducam OV2640 CCM), and the push button. Additional linear voltage regulators are also used to output 2.7V and 1.2V to power analog and logic core components in our camera, respectively. It is also responsible for recharging our Lithium-Ion Polymer battery (PRT-18286), along with providing overcharge, over-discharge, and short-circuit protection for our power circuit.

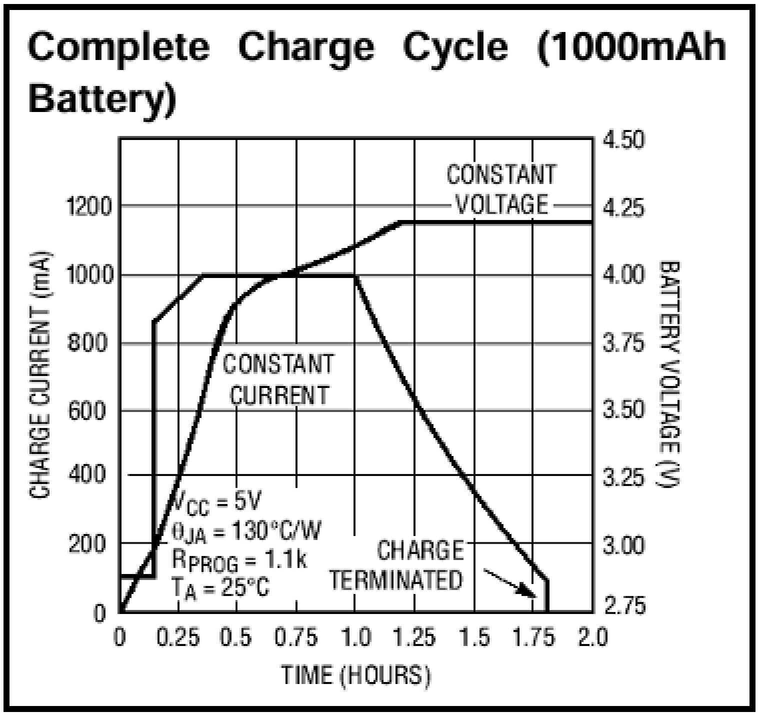


Figure 6: CC/CV charging cycles for a 1000mAh battery from TP4056 datasheet

When charging our battery, it is important to consider the appropriate resistor for the PROG pin of the TP4056, as this will determine the charging current during the CC phase. The specifications of our battery must line up with the specifications of the TP4056 to prevent any form of power malfunction. The formula given by the TP4056 datasheet is:

$$I_{BATTERY} = \frac{V_{PROG}}{R_{PROG}} * 1200,\ V_{PROG} = 1V$$

The standard charge current for the PRT-18286 is 0.5C, at a capacity of 1250mAh (on the datasheet). This means that the recommended charging current is:

$$I_{CHARGE} = 0.5C * 1250mAh = 625mA$$

Factoring this into our initial equation:

$$I_{BATTERY} = \frac{V_{PROG}}{R_{PROG}} * 1200$$

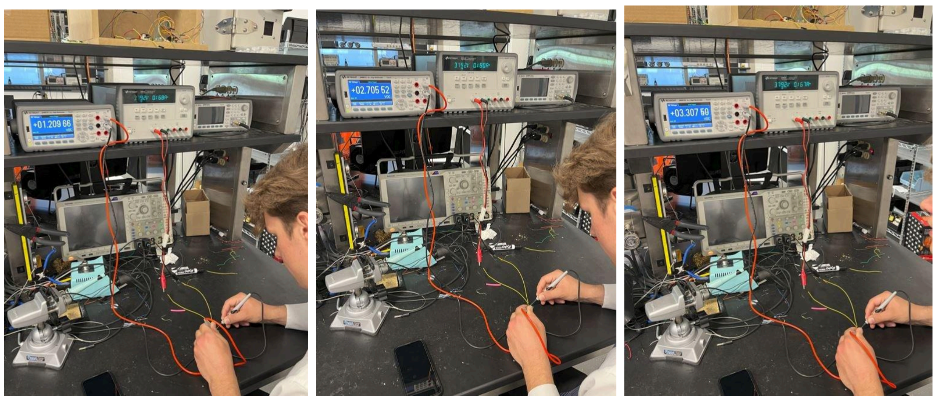
$$625 = \frac{1}{R_{PROG}} * 1200$$

$$R_{PROG} = 1.92k\Omega$$

Rounding this value to 2kΩ, we now have the appropriate resistor for the PROG pin of the TP4056, which will result in 600mA of current being used to charge the battery.

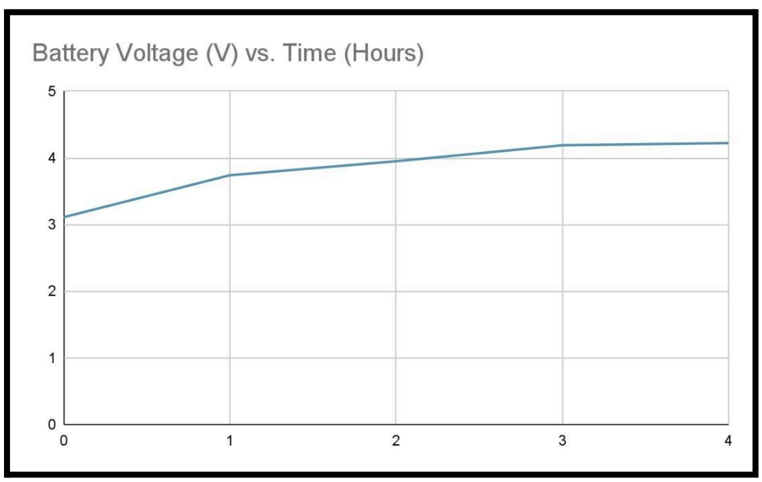
2.4.2: Requirements and Verification

1. Must be able to output 3.3V, 2.7V, and 1.2V to our peripherals and microcontroller, independent of one another. Datasheets detail tolerances of ± 5%.
   Result:



Images above show output voltages of 1.2V, 2.7V, and 3.3V ± 5%, respectively.

2. The battery must be able to be recharged via 5V USB input within 4 hours (based on the PRT-18286 datasheet), and charging should automatically stop when the battery is at maximum capacity.
   Result:



Line graph of Battery Voltages vs. Time when charging via 5V USB input, with 8 data points measured every 30 minutes across a cumulative time of 4 hours. A fully discharged battery starts at 3V, takes 1 hour to reach nominal voltage (3.7V), 2 hours to reach 4.0V, and finishes CC cycle and transitions to CV cycle at 3 hours (4.2V).

2.4.3: Design Alternatives

We originally created a custom PCB modeled after the TP4056 HiLetGo breakout board for our power subsystem, but unfortunately, the EP pin was connected incorrectly and caused the PCB to start smoking upon plugging in a 5V USB-C source. Due to this and time constraints, we resorted to using our backup breakout board instead of our custom power PCB for this subsystem in the end, albeit they are functionally and parts-wise nearly identical except for the PROG pin's connector resistor, as detailed before in the equation.

2.5 Communication Subsystem

2.5.1: Description

Our communication subsystem acts as a bridge between the peripheral and application subsystems. Our microcontroller to handle this task is the ESP32-S3-WROOM-1-N16R8. The GPIO pins of the microcontroller will be wired to the components of the peripheral subsystem (push button, microphone, speaker DAC, and camera). It will communicate with the speaker DAC and microphone via the I2S protocol and with the camera via the I2C protocol. This communication will be enabled by drivers for each respective device. Additionally, the microcontroller will provide a 24 MHz clock signal to the Arducam OV2640 CCM camera to ensure the camera can accurately capture and transfer images. The microcontroller will be powered by a 3.3V signal from the power subsystem and will communicate with the application subsystem via its built-in Bluetooth antenna.

The purpose of this subsystem is to facilitate communication between the peripheral subsystem and the application subsystem, which contributes heavily to the overall functionality of our design. With this subsystem properly constructed, the camera will only take photos at the desired times (when prompted by the microphone or pushbutton), the application subsystem will receive the captured images via Bluetooth, and the speaker DAC will receive the audio prompt sent from the application subsystem.

This subsystem will also help to achieve our high-level requirement of a fast computation time. Using a microcontroller that can run its core at up to 240 MHz can help speed up the process of image and audio transfer between the peripheral and application subsystems.

2.5.2: Requirements and Verification

1. Communication with all devices in the peripheral subsystem.

   Result:

   Images below show the serial console after running programs to capture an image, play audio from the speakers, and record from the microphone.

```
Enter command: img_capture
file, size, time taken: img_1746496455.jpg, 49400, 2.196117877960205 s
Enter command: spk_stream
Audio file info: 1 channels, 1 bytes per sample, 8000 Hz
Total frames: 30497, estimated size: 30497 bytes
Sent spk_stream command
Sending chunk 61/61 - 30497/30497 bytes
Finished sending audio data
Audio streaming completed. Total bytes: 30497, time taken: 4.16s
Enter command: mic_stream
file, size, time taken: mic_1746496481.wav, 160044, 4.648178815841675 s
```

2. Bluetooth connection established with the application subsystem.
   Result:
   The image above shows serial console output when prompted to output whether or not communication has been established between the microcontroller and the application.

```
Connected to ESP32-S3!
```

2.5.3: Design Alternatives

We chose to use this particular version of the ESP32 microcontroller (ESP32-S3-WROOM-1-N1698) primarily due to its built-in Bluetooth antenna. Since we wanted to communicate with a mobile app for image processing, the use of this microcontroller eliminated the need for an external antenna. The ESP32-S3 series supports camera interfaces, which is an essential feature in our project. Additionally, the ESP32 is supported by Arduino IDE, which is a platform some of our group is familiar with, which aided in our development of drivers for our devices from the peripheral subsystem.

2.6 Application Subsystem

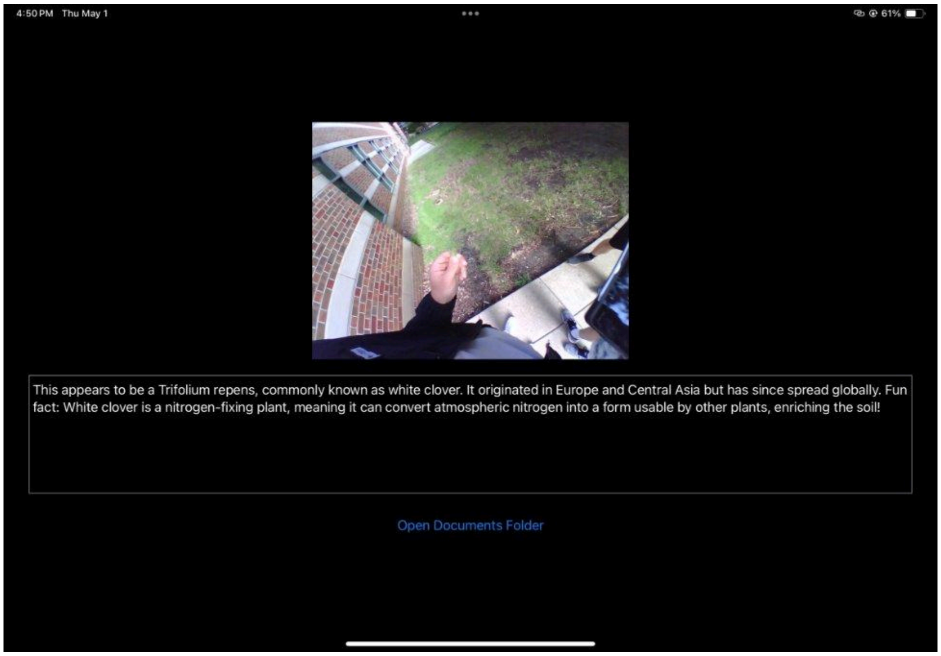
2.6.1: Description

The application subsystem will interface with the communication subsystem to receive and send data for app functionality. The UI/UX of the application will be on an iOS device and developed using Xcode tools and Swift programming. The application will utilize the mobile device's onboard Bluetooth to send and receive data.

Overall, the application itself will receive the image from the communication subsystem, process the image using a plant classification algorithm (the image processing model that we will use will be discussed in our tolerance analysis), generate an audio stream, and then relay this audio stream through a Digital-to-Analog Converter (DAC). This analog signal will finally be played by the speaker on the glasses, giving the user an audible description of the photo initially taken.

2.6.2: Requirements and Verification

1. The application must be able to receive a photo taken by the camera.
   Result:



   The screenshot above shows the rendered photo taken from the camera on the iOS application.

2. The application must be able to send accurate data to the Digital-to-Analog converter.
   Result:
   As shown in a previous screenshot of the firmware, a “spk_stream” command is sent to the ESP32 to let it know that raw PCM data of the description will be sent from the app to the firmware, where it is upsampled to the specifications needed. The correct number of bytes is sent along with the correct raw PCM metadata.

2.6.3: Design Alternatives

The application has a barebones UI/UX, and has a hardcoded prompt being fed to Gemini asking for it to analyze the plant shown in the image. Wider applicability of the iOS app can be accomplished through having the user specify a prompt that is first sent to Gemini, then initiating the user flow (i.e, you could input “What animal is being shown?” instead of asking for plant classification). Also, the “Open Document Folder” button does not work, so that could also be fixed if the user wants to relisten to the description read out from the application.

4. Costs

4.1 Parts

Description	Manufacturer	Quantity	Cost
Arducam OV2640 CCM Camera	Arducam	1	$8.99
ICS-43434 Microphone	Adafruit Industries LLC	1	$4.95
ADS01008MR-LW100-R Speakers	PUI Audio Inc.	2	$4.32
PCM5100APW DAC	Texas Instruments	1	$2.50
ESP32-S3-WROOM-1-N16R8 Microcontroller	Espressif Systems	1	$6.56
B3U-1100P Pushbutton	Omron Electronics Inc-EMC Div	1	$0.85
PRT-18286 Rechargeable Battery	SparkFun Electronics	1	$10.95
DW01A Overcharge Protection IC	EVVO	1	$0.23
FS8205A Dual-MOSFET	EVVO	1	$0.34
USB4135-GF-A USB-C Port	GCT	1	$0.67
IRLML6402TRPBF PMOS	Infineon Technologies	1	$0.38
B2B-PH-K-S Connector	JST Sales America Inc.	2	$0.24
PHR-2 Connector	JST Sales America Inc.	2	$0.20
Passive Components (Resistors, Capacitors, etc.)	YAGEO, E-Shop	50	~$0.10

Parts Total = $47.57

4.2 Labor

To determine the labor cost of producing our project, we will use the average starting salaries of the University of Illinois at Urbana-Champaign electrical and computer engineers. The average starting salary for an electrical engineer is

$88,321 [4] and the average starting salary for a computer engineer is $118,752 [5]. The average number of work days in a year in the United States is 260, with an average work day of 8 hours. This leaves us with approximate hourly wages for electrical and computer engineers of $42.46 and $57.09, respectively. Our group is composed of 2 computer engineers and 1 electrical engineer, giving us an estimated average group hourly rate of $52.21. We anticipate that the project will take around 150 hours to complete (50 hours per person), giving us an estimated labor cost of $19,578. Combining this with the cost of parts, this gives us an estimated total cost of $19,625.57 to create our project.

5. Conclusion

5.1 Accomplishments

Overall, the project achieved the vast majority of its intended functionality. The push button, as intended, started the glasses system by prompting the image capture. After the image capture, the microcontroller fed the image to our developed application, which then identified what was in the image and sent an appropriate audio prompt back to the microcontroller. Additionally, the lithium-ion battery is capable of being recharged via USB-C.

5.2 Uncertainties

The main shortcoming of the project was the speaker volume level. In a completely silent room, the speakers were able to output some slightly audible audio, but overall, it was much too quiet for real-world application. Our requirements and verification aimed to have the speakers output around 50 dB of sound an inch away from the speakers, but we were only able to achieve this metric around ¼ of an inch away.

Another shortcoming in our project was the custom power PCB. The created PCB had a trace coming from the bottom of the TP4056 charging module chip that was supposed to be connected to ground, but was accidentally connected to Vin. We had ordered the PCB on a late schedule, and in the interest of time, instead of attempting to use jumper wires to connect an appropriate trace, we used a breakout board for our battery recharging module.

The last shortcoming of our project was the DAC for the main PCB. Although we achieved the intended functionality of the DAC via a written driver, we needed to use a breakout board to do so. Our third round PCB order had the DAC incorrectly mapped, which we aimed to fix on the fourth round PCB. However, upon arrival of the fourth round PCB, we could not connect the camera to the microcontroller and, resultantly, opted to use our third round main PCB in our final implementation with jumper wires connecting the microcontroller to the DAC breakout board.

5.3 Ethical considerations

A major ethical concern associated with our project is that of privacy due to the inclusion of a camera in our design. The IEEE Code of Ethics compels us to protect the privacy of others [1], which will impact how our application subsystem handles the images taken with the camera. To comply with this legislation, our application will not store any of the images taken with the camera, therefore preserving the privacy of anyone photographed with our design. This will be done by the application automatically, overwriting the last taken picture when a new picture is taken. Additionally, closing and reopening the application will delete the last taken picture, therefore giving the user no access to the most recent image capture.

5.4 Future work

Future work of this project would mainly entail 2 major areas: the speakers and the form factor. Since our speakers' output sounds much quieter than we would have liked, future work would include an op-amp in our implementation to help remedy this. This would amplify the stream sent to the speakers and therefore increase the volume output. As for the form factor, our fourth round PCB was considerably smaller than our third round, leading to a much more compact system and smaller overall design. Not only would this lead to a more desirable product, but it would also potentially push our project further into the reach goal territory of our high-level requirement regarding total weight. Given more time, we would like to debug this further and achieve a more lightweight and compact design.

References

Arducam OV2640 CCM - Camera
https://www.uctronics.com/download/OV2640_DS.pdf

ICS-43434 - Microphone
https://invensense.tdk.com/wp-content/uploads/2016/02/DS-000069-ICS-43434-v1.2.pdf

ADS01008MR-LW100-R - Speaker
https://www.mouser.com/datasheet/2/334/ADS01008MR_LW100_R-1916757.pdf

PCM5100APW - Audio Stereo DAC
https://www.ti.com/lit/ds/symlink/pcm5100a.pdf?ts=1739433553738&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FPCM5100A

ESP32-S3-WROOM-1-N16R8 - Microcontroller
https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf

B3U-1100P - Pushbutton, Ultra-small Tactile Switch (SMT)
https://omronfs.omron.com/en_US/ecb/products/pdf/en-b3u.pdf

PRT-18286 - Rechargeable Battery
https://cdn.sparkfun.com/assets/e/3/3/2/7/-KPL623450-1300mAh-3.7V________________2021-01-21.pdf

TP4056 - Standalone Li-Ion Battery Charger
https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf

DW01A - Overcharge Protection IC
http://www.tp4056.com/d/dw01a-fs.pdf

FS8205A - Dual-MOSFET
https://www.mpu51.com/mcucity/DATA_PDF/FS8205A-DS-12_EN.pdf

USB4135-GF-A - USB-C Port
https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/6501/USB4135%20Product%20Spec.pdf

IRLML6402TRPBF - PMOS
https://www.infineon.com/dgdl/irlml6402pbf.pdf?fileId=5546d462533600a401535668d5c2263c

B2B-PH-K-S - Connector
https://www.jst-mfg.com/product/pdf/eng/ePH.pdf

PHR-2 - Connector
https://www.jst-mfg.com/product/pdf/eng/ePH.pdf

FPC FFC PCB - Converter Board
https://media.digikey.com/pdf/Data%20Sheets/Hirose%20PDFs/FPC,FFC%20Connectors.pdf

5.2 Ethics/Safety

[1] IEEE. “”IEEE Code of Ethics”.” (2020), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 03/06/2025).

[2] Conro. “”How to Waterproof Electronics?”.” (2023), [Online]. Available: https://www.conro.com/Blog/How-to-waterproof-electronics/ (visited on 03/06/2025).

[3] L. Kong, C. Li, J. Jiang, and M. Pecht, “Li-ion battery fire hazards and safety strategies,” Energies, vol. 11, p. 2191, 08 2018. [Online]. Available: https://www.mdpi.com/1996-1073/11/9/2191

[4] G. E. O. of M. and Communications, “Electrical Engineering,” grainger.illinois.edu. https://grainger.illinois.edu/academics/undergraduate/majors-and-minors/electrical-engineering

[5] G. E. O. of M. and Communications, “Computer Engineering,” *grainger.illinois.edu*. https://grainger.illinois.edu/academics/undergraduate/majors-and-minors/computer-engineering

Appendix A: Requirement and Verification Tables

Application Subsystem:

Requirements	Verification	Verification Status (Y or N)
The application must be able to receive a photo taken by the camera.	• The application should show the same picture that was taken by the camera and should be vivid and visible. 10 pictures will be taken by the camera in intervals of 30 seconds to verify that a photo is ready to use for further image processing on the application.	**Y**
The application must have a working image processing algorithm.	• This verification process is very similar to our high-level requirement verification process. • The algorithm should be able to correctly identify plants 85% of the time. This will be based on our application subsystem, which will use a model to determine the plant's classification. We will test the algorithm on 20 plant images and expect to get 17 out of the 20 correct.	**Y**
The application must be able to send accurate data to the Digital-to-Analog converter.	• Run the algorithm through our selected classification model 10 times in intervals of 1 minute. Verify that each time this test is run, audible and crisp audio output is being outputted from the speaker, indicating that the application and DAC interface are working as intended.	**Y**

Table 1: Requirements and Verification for Application Subsystem

Communication Subsystem:

Requirements	Verification	Verification Status (Y or N)
Communication with all devices in the peripheral subsystem (push button, camera, microphone, speaker, DAC)	• Camera ○ After programming, open the serial monitor in Arduino ○ Save the information sent over the	**Y**

	serial bus to the PC ○ If the opened jpeg file is not an image, the test has failed ● Push button ○ Attach a digital multimeter probe to the pin of the microcontroller connected to the push button ○ If the button is pressed and the signal at the pin does not switch to active low, then the test has failed ● Microphone ○ After programming the microcontroller, add statements within the code to print the value of the audio signal received ○ If there is an error printing the value or the value is zero when sound is being made, the test has failed ● Speaker DAC ○ Attach an oscilloscope to the DIN pin of the speaker DAC ○ Monitor the oscilloscope reading when data should be sent to it ○ If the oscilloscope reading does not change, then the test has failed	
Bluetooth Connection established with the application subsystem	● After programming the microcontroller to use Bluetooth low energy add print statements within a polling loop to print to the serial monitor when it establishes Bluetooth connection with the application ● If the serial terminal never prints a connection message, the test has failed	**Y**

Table 2: Requirements and Verification for Communication Subsystem

Power Subsystem:

Requirements	**Verification**	**Verification Status (Y or N)**
Must be able to output 3.3V,	● Power circuit and measure the output	**Y**

2.7V, and 1.2V to our peripherals and microcontroller, independent of one another.	voltage at each of the LDOs using a multimeter at no load. We should expect to see values of 3.3V, 2.7V, and 1.2V ± 5%. ● Power only one regulator at a time to ensure the other outputs are unaffected. Go through all 8 possible combinations of on and off LDOs to make sure voltage readings are the same as above when on. ● Measure voltages when device loads are attached (peripherals and microcontroller). Ensure that we still see the same voltage values as what we saw from the multimeter.	
The battery must be able to be recharged via 5V USB input within 4 hours (based on the PRT-18286 datasheet), and charging should automatically stop when the battery is at maximum capacity.	● Start with the battery discharged to 3.0V, the cut-off voltage for our PRT-18286. Monitor the current delivered to the battery with a multimeter, making sure that the current is 600mA ± 5%. ● Verify that during the CC phase, the voltage of the battery is increasing and stops at 4.2V. ● Once the battery is at 4.2V, verify using a multimeter that the current being fed into the battery is now decreasing during the CV phase. Charging should fully stop once the current drops below 10% of the set charging current (60mA). ● The red LED should be on during the CC/CV phase and on for about 4 hours, while the Green LED should turn on, and the Red LED should turn off once the multimeter reads 0mA.	**Y***

Table 3: Requirements and Verification for PowerSubsystem

Peripheral Subsystem:

Requirements	**Verification**	**Verification Status (Y or N)**
Only the microphone and push button should initiate the glasses system: no other process should start it, and it should not start randomly	● Attach an oscilloscope probe to the pin of the microcontroller that is responsible for activating the camera ● Test active high inputs to the other pins of the microcontroller, coming from the other subsystems	**Y**

	• If anything other than inputs from the microphone/push button cause the oscilloscope reading to change, then the test has failed	
The speakers must output audio that can be comfortably heard by the user. Research would indicate that this range is around 50 dB	• The glasses' speakers will be positioned an inch away from a phone • An audio prompt is to be played from the speakers • If the phone's sound level meter detects any sound level outside of the 45-55 dB range, the test has failed	**N**

Table 4: Requirements and Verification for Peripheral Subsystem

Physical Subsystem:

Requirements	**Verification**	**Verification Status (Y or N)**
Components are secured firmly in place during normal user operations. There must be no internal movement of components when shaken at a rate of ½ meter per second.	• Measure out a distance of ½ meters. • Set up a timer to count down from 10 seconds. • Start the timer and move the glasses in the air left to right, aiming to reach each side at 1-second intervals. During this time, count the number of lengths traveled to verify the final rate. • Once the timer expires, halt the process and ensure that the number of lengths traveled is 10. • The test is successful if there is no audible rattling internal of the glasses throughout the test.	**Y**
The surface of the glasses does not contain any abrasive textures/sharp edges. The user must not be able to receive any cuts or scratches during normal operation.	• Set the device on a flat table. • Using a paper towel, brush up against the outer surface of the glasses. Be sure to brush every surface, including the frame and arms. • Once all surfaces have been brushed two times, inspect the glasses for any signs of debris from the paper towel. • If there is any sign of paper towel on the	**Y**

	glasses, the test has failed.	
The final prototype will weigh no more than 200 grams with all components inside.	● Put a scale on a flat table and zero out the scale. ● Place the glasses on the scale and inspect the shown weight. ● If the shown weight is above 200g, the test has failed.	**Y**

Table 5: Requirements and Verification for Physical Subsystem

Appendix B: Subsystem Schematics and PCB Designs

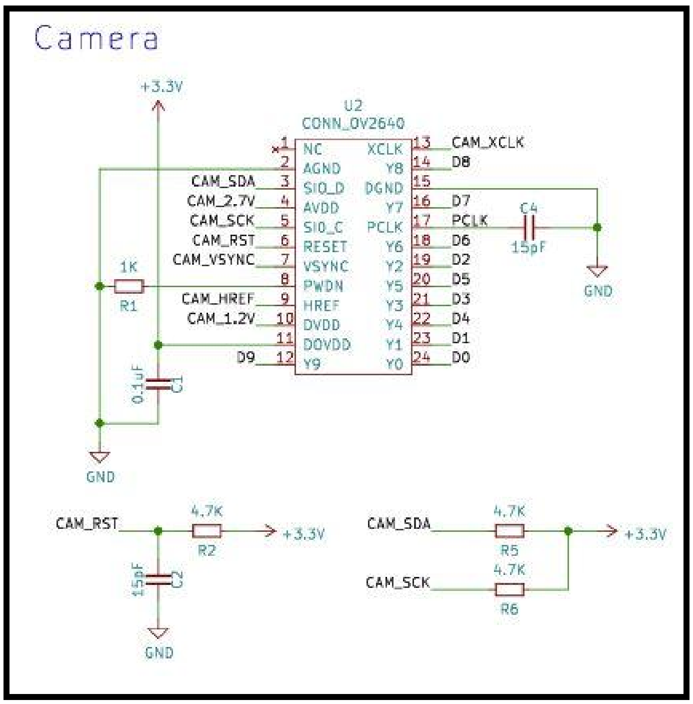


Figure 7: Camera Schematic

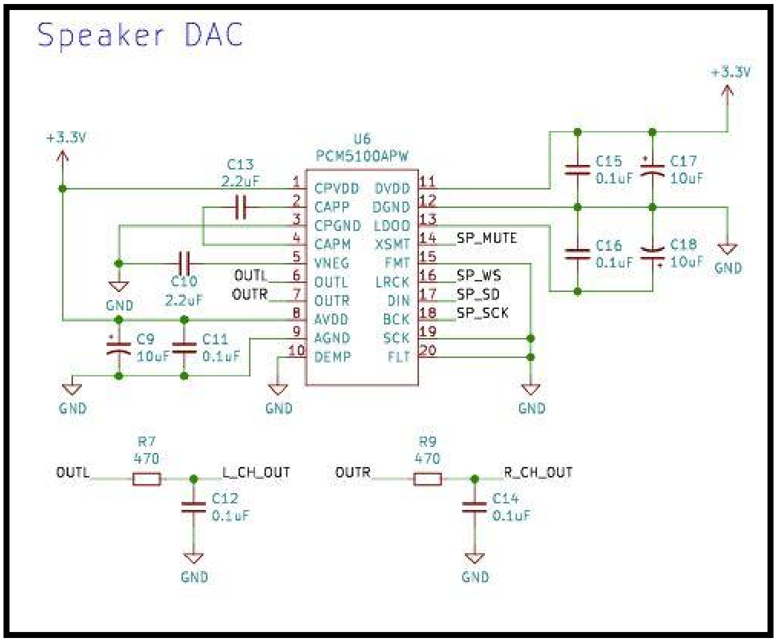


Figure 8: Speaker and DAC Schematic

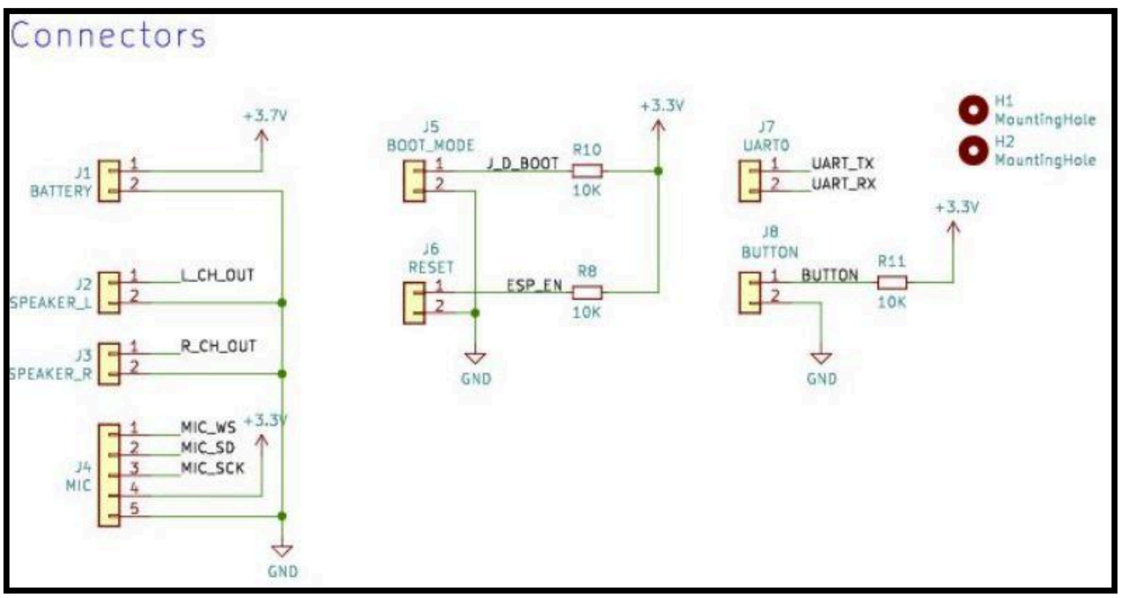


Figure 9: Connectors, Microphone, and Push Button Schematic

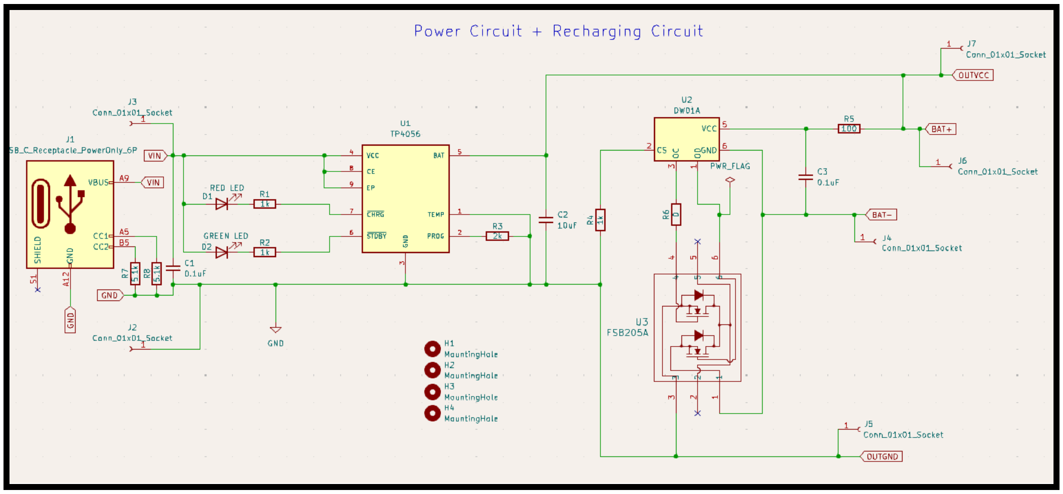


Figure 10: Power Circuit with Recharging Capabilities Schematic, Outputs 3.7V

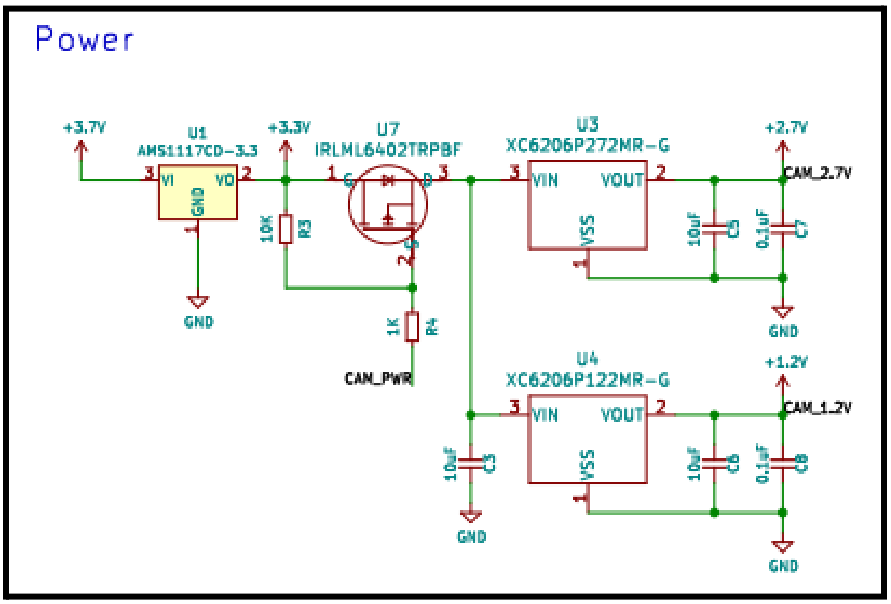


Figure 11: Voltage Regulation Circuit, 3.7V converted to 3.3V, 2.7V, and 1.2V

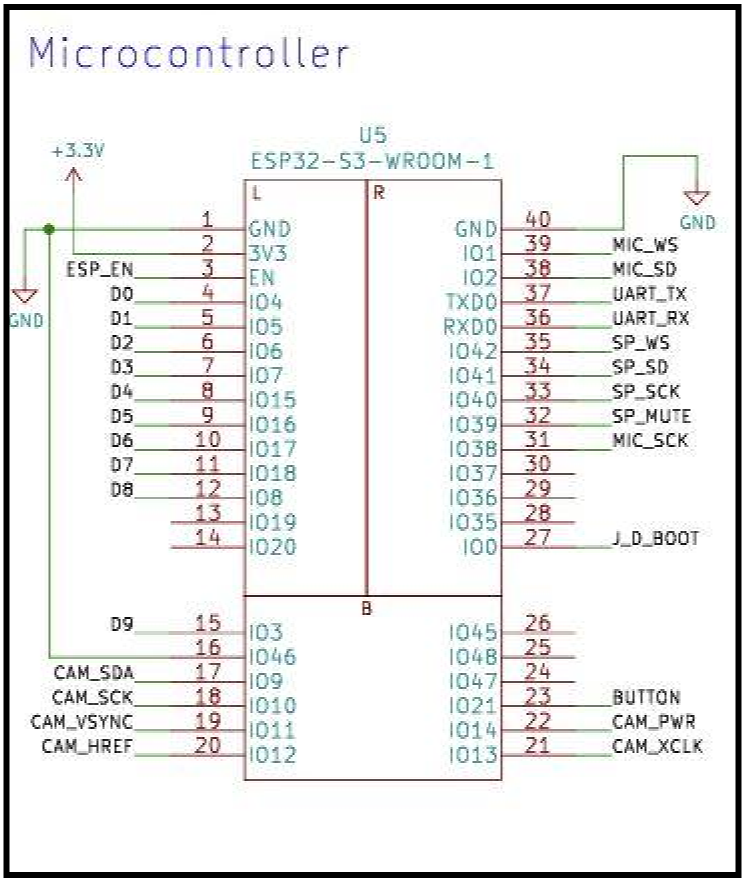


Figure 12: Microcontroller Schematic

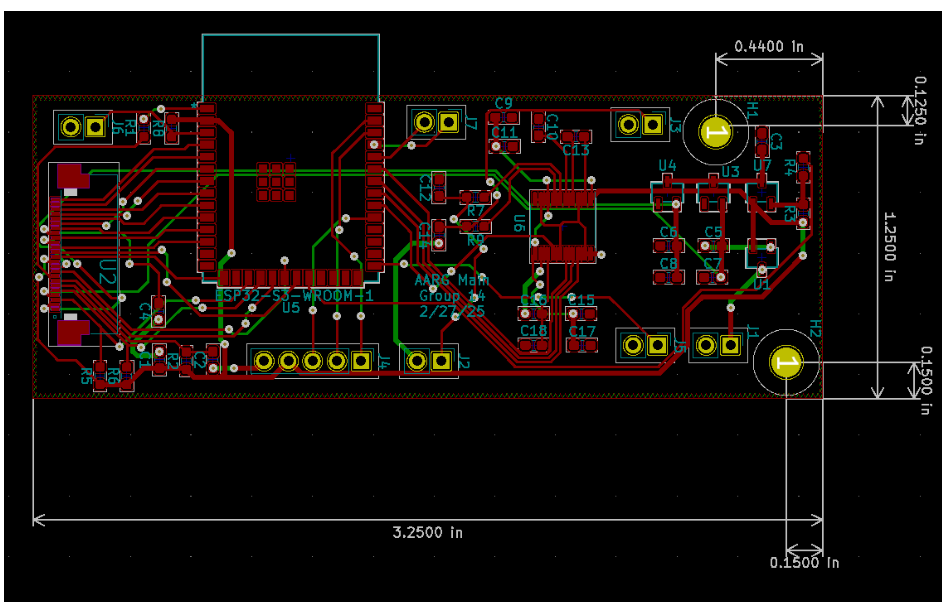


Figure 13: Final Main PCB Design

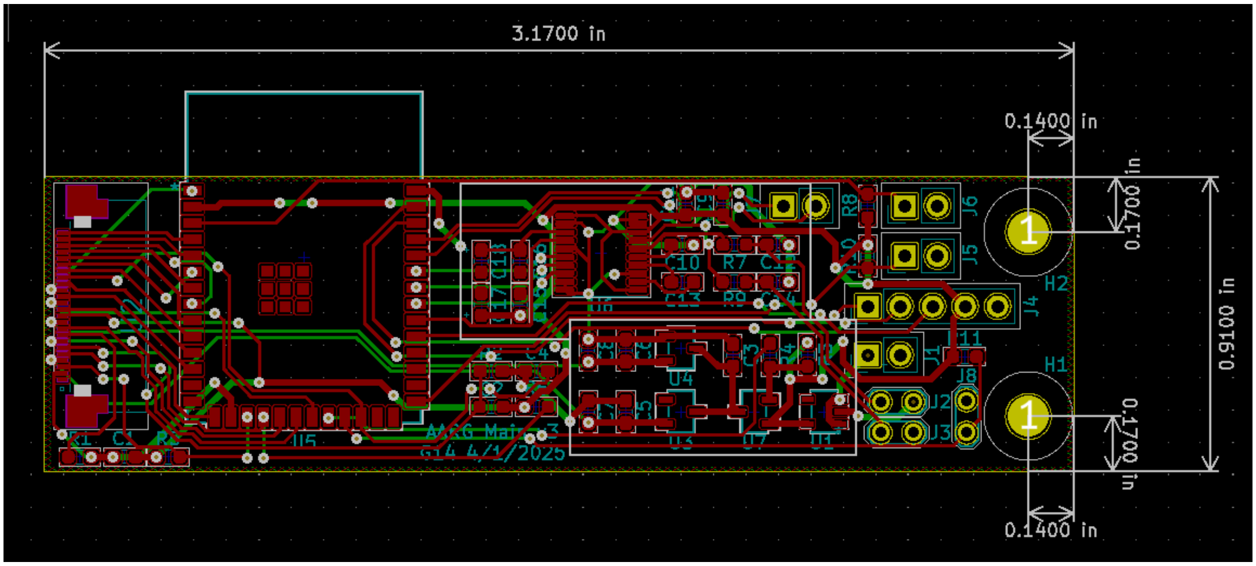


Figure 14: Round 4 Main PCB Design

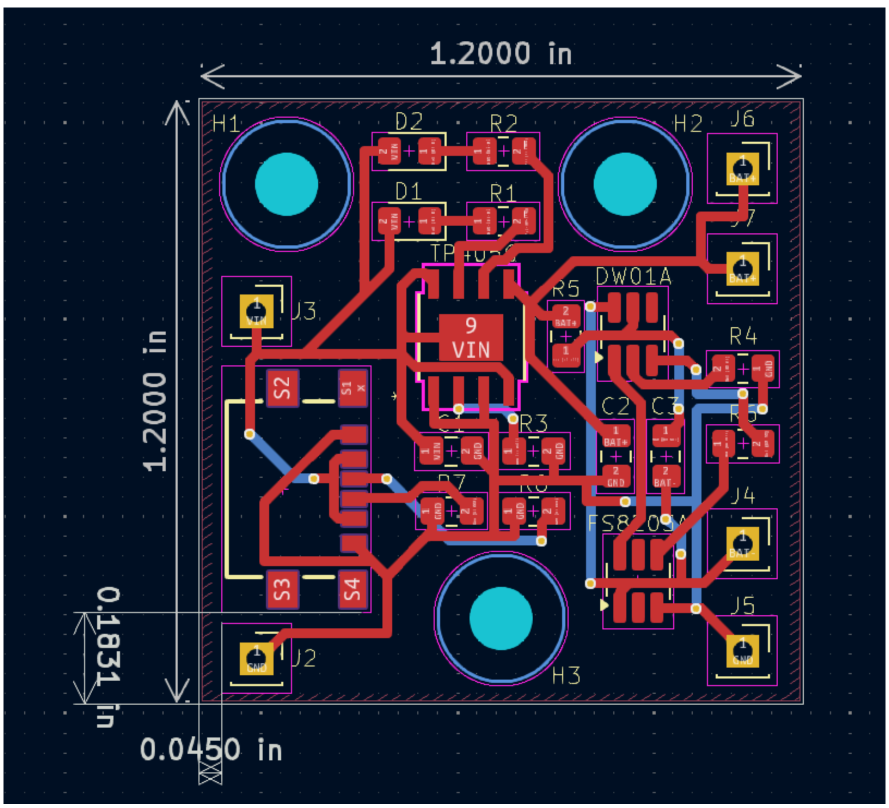


Figure 15: Custom Power PCB Design