

LAB NOTEBOOK

Notebook Entry: January 24th 2024

Objectives:

- Write a web post for project so scope out what projects I can do
- Brainstorm ideas on what a project could entail with the project constraints

Work Session Record:

Wrote the following web post for our chosen project

Create an insole to track where as you hike you experience the most pounding. I'd use the same sort of idea as existing ski boot tackers but adjust it to hiking. Would use a conductive sheet to measure where there is most pressure. Also could integrate an accelerometer and other sensors to summarize a hike.

1/26

approved for RFA

Notebook Entry: January 29th 2024

Objectives:

1. To formulate a comprehensive project proposal outlining the development of an insertable insole with integrated sensors for hikers.
2. Identify the key components, objectives, and criteria for success of the project.
3. Discuss and define the technical specifications and functionalities of the proposed solution.

Work Session Record:

During today's work session, the focus was on drafting sections of the project proposal for a specialized product aimed at enhancing hiking safety through real-time foot stress monitoring. We delved into the problem statement, acknowledging the necessity for a solution that can mitigate injury risks by tracking various factors affecting foot dynamics during hikes, such as gait, terrain, and fitness levels.

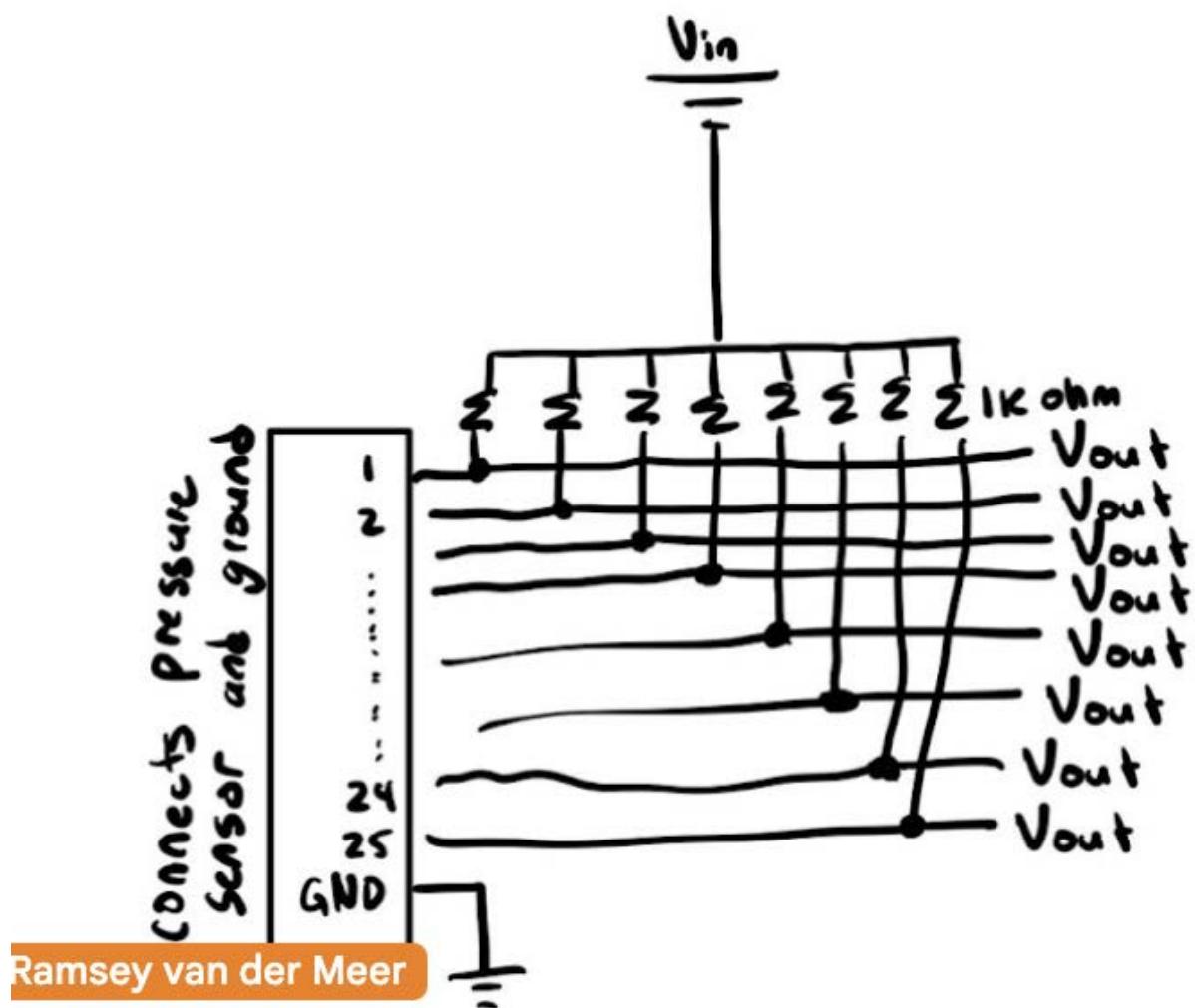
Our proposed solution centers around the development of an insertable insole equipped with an array of sensors including accelerometers, gyroscopes, and pressure sensors. These sensors are crucial for monitoring foot movements and pressures, providing valuable data for injury prevention. We emphasized the importance of selecting durable, comfortable materials for the insole and ensuring seamless wireless connectivity for real-time feedback.

In detailing the solution components, we discussed integrating status LEDs for visual indicators, a user-friendly smartphone interface, and physical controls on the insole for convenient data management. Additionally, we outlined plans for powering the device using a lithium-ion battery with sufficient capacity to support extended hikes.

To measure the success of our project, we established key metrics including sensor accuracy, insole comfort and durability, efficient data communication, battery life, and user interface usability. Furthermore, we highlighted the potential for data analytics and stressed the significance of professional interpretation of the collected data for actionable insights.

This work session laid the groundwork for a comprehensive project proposal, addressing both the technical aspects of the solution and the criteria for its success, while also considering potential future developments and the broader implications of the project.

Diagrams and Figures:



Conclusion:

This session focused on drafting a detailed project proposal for a sensor-integrated hiking insole. The team outlined the problem, proposed a solution with specific components, and defined the success criteria for the project. The proposal includes considerations for sensor accuracy, user comfort, data communication, and power management, aiming to enhance the hiking experience and reduce injury risks.

Notebook Entry: January 30th 2024

Objectives:

- To update the project proposal by incorporating feedback and new ideas.
- Add details on data storage on the device and the process for offloading data after a hike.
- Introduce considerations for filtering noise through Digital Signal Processing (DSP).
- Clarify the scope of advice and recommendations provided by the project in relation to medical expertise.

Work Session Record:

New Sections on Design Doc:

- **On-Device Data Storage and Offloading:** Added a new section to the proposal detailing the functionality for storing data directly on the device during hikes. This would allow users to save battery on their smartphones and offload data to a smartphone or computer after the hike. Discussed potential memory requirements and data compression techniques to maximize storage efficiency.
- **Noise Filtering through DSP:** Outlined the challenges of filtering out noise from sensor data, such as irregular impacts or external pressures that do not contribute to meaningful analysis. Included a brief overview of DSP techniques that could be employed to clean the data, such as filtering algorithms and thresholding.

Revisions to Existing Sections:

- Revised the **Solution Components** and **Criterion for Success** sections to reflect the new focus on data storage and noise filtering.
- Enhanced the **User Interface** section to emphasize the importance of visualizing data in an accessible manner for users, with the option for exporting data for professional analysis by a podiatrist.

Clarification on Medical Advice:

- Added a disclaimer clarifying that while the project aims to provide useful feedback on foot stress and hiking technique, it does not offer medical advice. Stressed the importance of consulting with a professional for a thorough analysis of the data.
- Suggested including references to existing studies on foot pressure and hiking injuries as a basis for any general recommendations made by the app.

Conclusion:

In this session, the project proposal was significantly enhanced to address feedback and incorporate new ideas. The addition of on-device data storage and offloading mechanisms, along with considerations for noise filtering, strengthens the proposal. By clarifying the project's stance on medical advice, we ensure that the project remains focused on providing value to hikers through data analysis and visualization, while respecting the boundaries of our expertise. This revised proposal lays a solid foundation for developing a tool that enhances hiking safety and enjoyment.

Notebook Entry: January 31st 2024

Project Approved

Notebook Entry: February 2, 2024

Objectives:

- Initialize lab notebooks
- Look into design ideas for the insole system
- Look into how to make the ESP 32 PCB

Work Session Record:

Created GitHub to contain lab notebooks, PCB design files, Web application, and any miscellaneous resources needed for our project. I followed the guide listed on the course wiki to format our file structure. Additionally, I looked at a few online resources for existing components that we could consider using on our final design.

works cited

- <https://www.amazon.com/Pressure-Sensing-Intelligent-Accessory-Pressure-Sensitive/dp/B09RWYRV5T>
- https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf

Notebook Entry: February 15, 2024

Objectives:

- add to design doc
- flush out sections to add more detail to the devices inner workings

Work Session Record:

Software Design Section for Smart Insole Project:

- Detailed aspects of "Data Collection, Aggregation, and Transmission" and "Web Interface".

Data Collection, Aggregation, and Transmission:

- Described the software flow, starting from Bluetooth pairing mode and transitioning into main operational phases.
- The software records sensor data during hikes and writes this information to storage.
- It sends data packets to a connected device via a web interface for real-time tracking or backup.

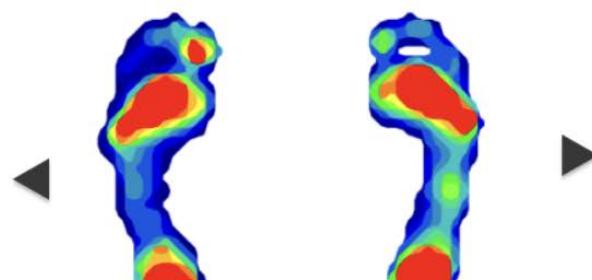
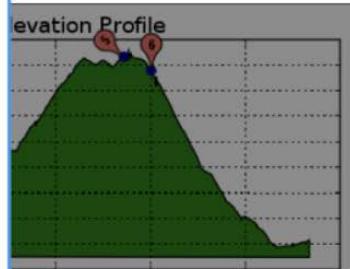
Web Interface:

- Outlined the development of a web interface for offloading data from the ESP32 device.
- Detailed the use of BLEServer, BLEService, and BLECharacteristic modules for Bluetooth services.
- The interface will utilize the Web Bluetooth API for connecting to the ESP32, enabling data transfer and interaction with device services and characteristics.
- Plans include displaying hiking statistics through charts and graphs using Next.js and various NPM modules.

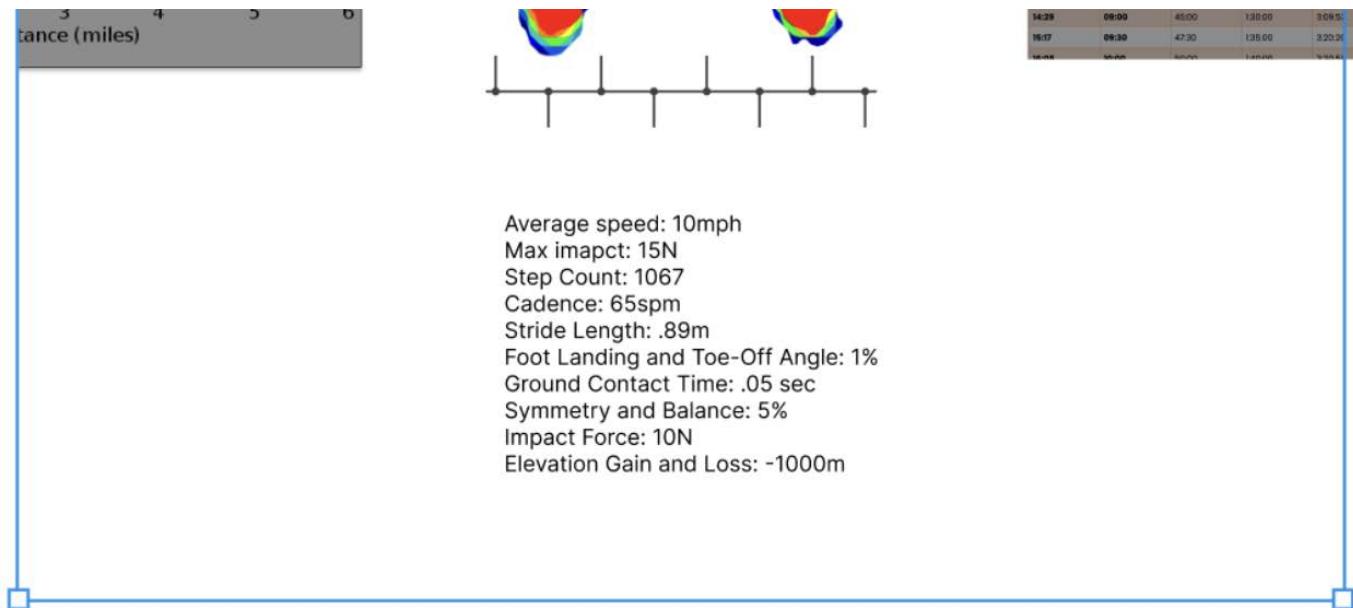
Diagrams and Figures:



Desktop - 1



Time	Altitude (meters per minute)	Speed (meters per minute)	Slope	Vertical Speed
4:50	03:00	15.00	30.00	103.8
5:18	03:00	17.00	35.00	113.5
5:26	04:00	20.00	40.00	124.24
7:14	04:30	22.30	45.00	134.56
8:03	05:00	25.00	50.00	145.29
8:52	06:30	27.30	55.00	156.02
9:40	06:00	30.00	60.00	166.31
10:28	06:30	32.30	65.00	177.08
11:16	07:00	35.00	70.00	187.41
12:54	07:30	37.30	75.00	198.14
13:53	08:00	40.00	80.00	208.4
13:40	08:30	42.30	85.00	219.2



Conclusion:

- Successfully documented the software design aspects of the Smart Insole project, focusing on efficient data handling and user interaction through a web interface.

Notebook Entry - February 17th

Objectives:

- Illustrate the wiring diagram and the physical design for clarity and implementation guidance.
- Outline the electronic principle of the voltage divider applied to the pressure sensors.
- Create diagrams for physical design of smart insole system
- Create a blow up of the Controller device and label what function it will perform
- Update the design document with the new diagrams and technical explanations.

Work Session Record:

1. Diagrams Created and Integrated:

- Completed a wiring diagram for the pressure sensors within the insole, highlighting the use of a voltage divider circuit to measure variations in pressure .
- Developed a visual representation of the physical design of the smart insole, detailing the placement of pressure sensors, wiring, and connectivity to the external sensor box.
- Illustrated a detailed view of the external sensor box, indicating the functions of various buttons, LEDs, and the USB-C charging port.

2. Technical Concept Elaborated:

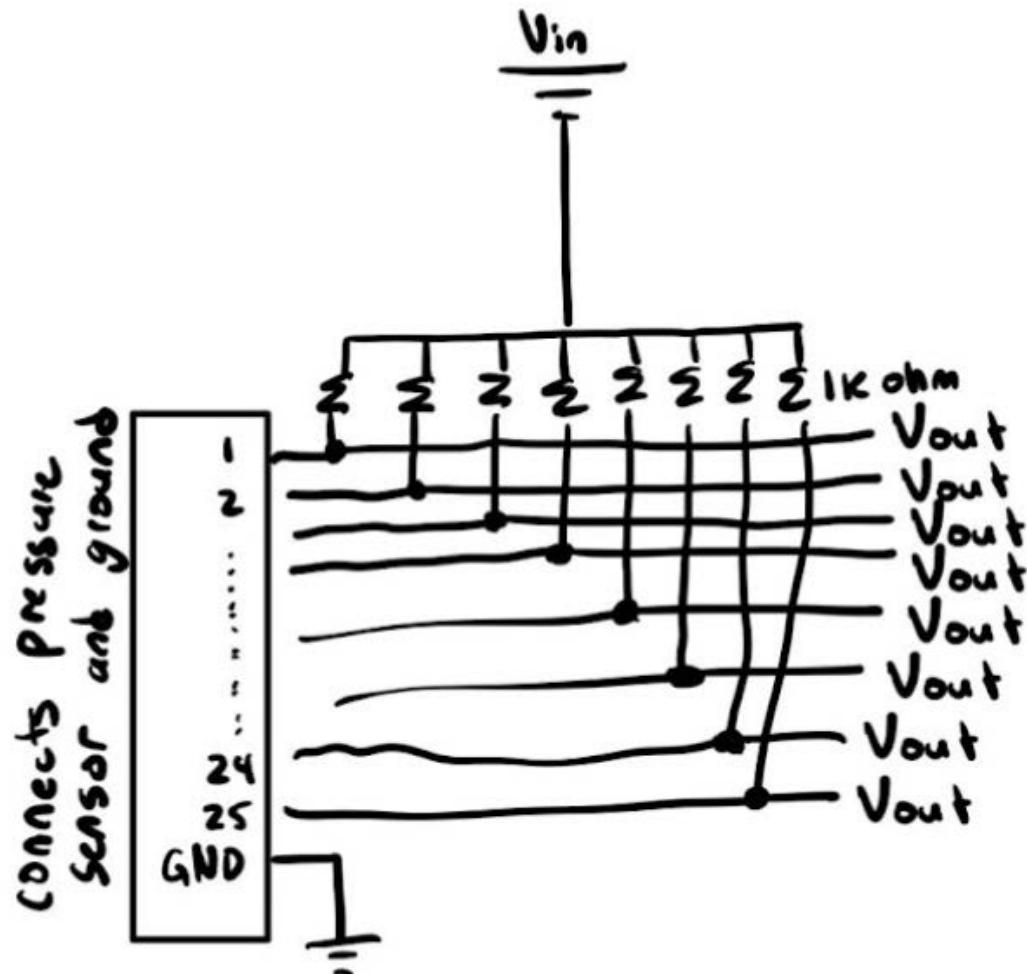
- Explained the electrical concept of a voltage divider in the context of pressure sensors. As the resistance of a pressure sensor increases with applied pressure, the voltage across it changes. This change in voltage is used as an analog signal representing pressure levels.
- Elaborated on how this principle is key to the function of our smart insoles, allowing for accurate pressure readings which are then transmitted to a web interface via Bluetooth.

3. Design Document Update:

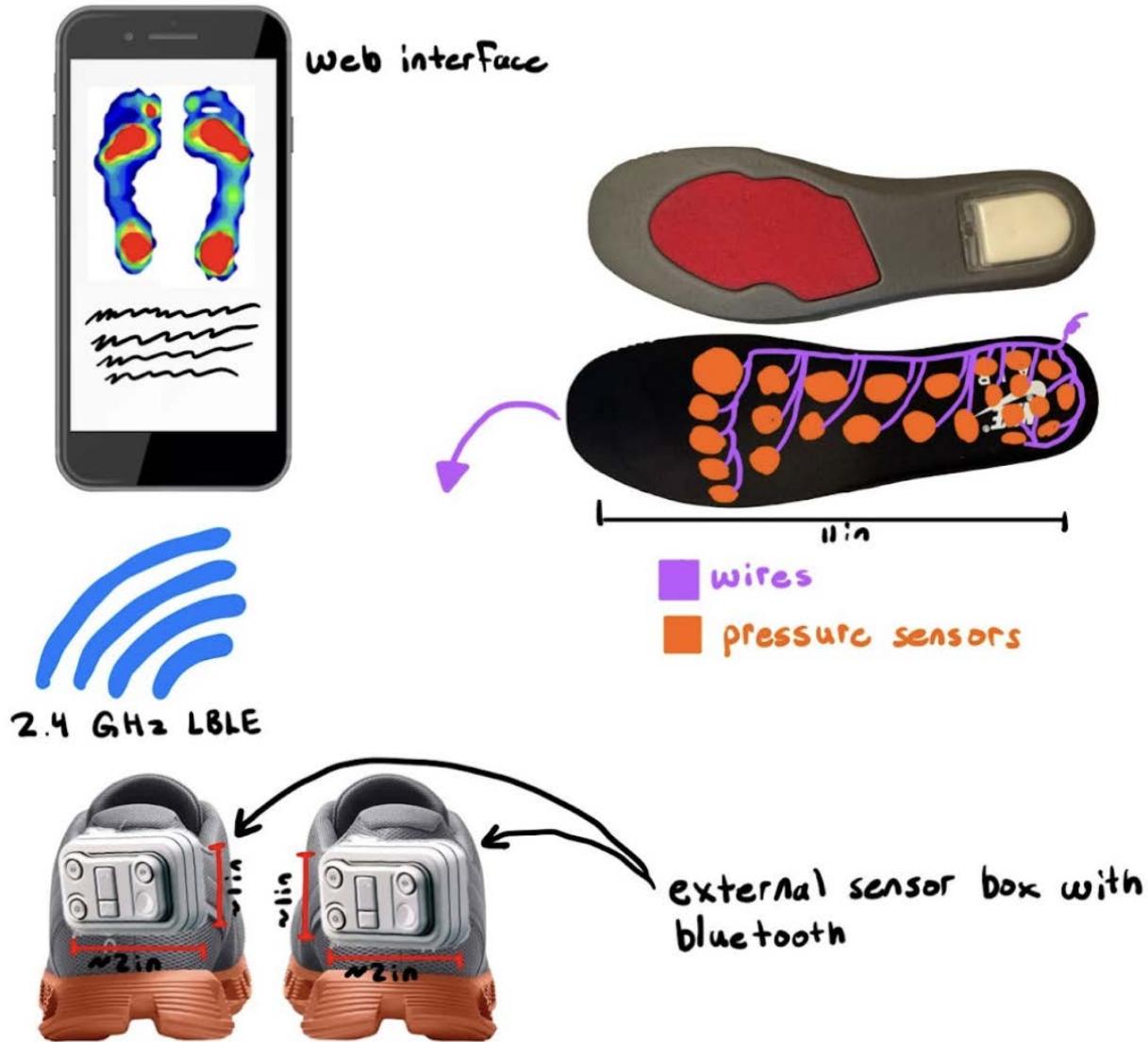
- Added the new diagrams to our design document
- Included annotations in the diagrams for better understanding and implementation.

Diagrams and Figures:

- **Pressure Sensor Wiring Diagram:** A schematic showing the arrangement of resistors and connections to the microcontroller to form a voltage divider network.



- **Physical Design Illustration:** A graphical layout of the insole, displaying sensor positions, wiring paths, and connectivity to external components.



- **External Sensor Box Design:** A blow-up view of the external sensor box with labeled buttons and



- Start hike
- end hike
- Status LED's
- Bluetooth pairing buttons + LED
- Power Button
- Misc. Buttons
- USB C charging port

ports.

Equations:

- Voltage Divider Equation: $V_{out} = V_{in} * (R2 / (R1 + R2))$
- Where V_{out} is the voltage across the pressure sensor ($R2$), V_{in} is the input voltage, $R1$ is the fixed resistor, and $R2$ is the variable resistor (pressure sensor).

Notebook Entry: February 19, 2024

Objectives:

- Integrate the microcontroller into the sensing subsystem.
- Add control buttons to the status subsystem.
- Make the wire legend distinct for each subsystem.
- Update the requirements and verifications table.
- Calculate storage needs for data collection and finalize storage solutions.

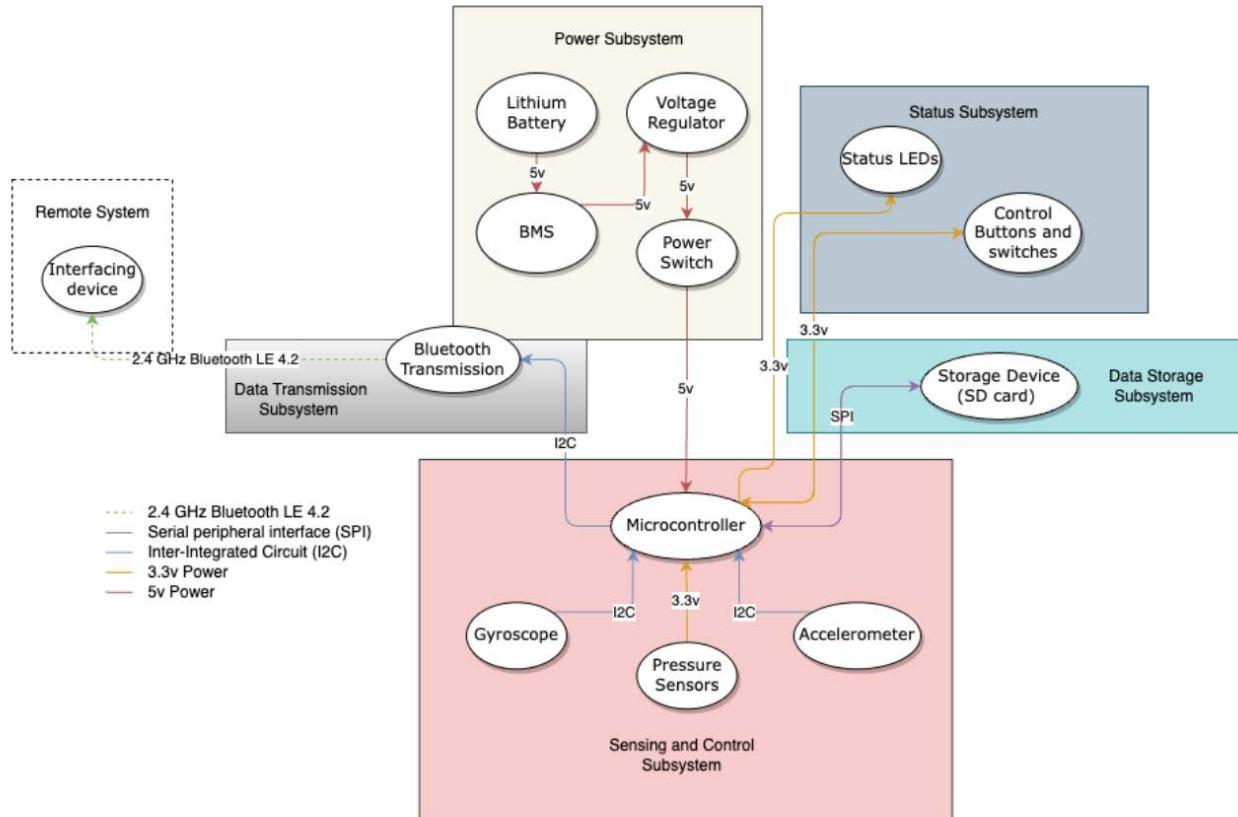
Work Session Record:

- Edited the block diagram to include the ESP32 microcontroller within the sensing subsystem for enhanced data processing and control.
- Incorporated control buttons (start/end hike, Bluetooth connectivity) into the status subsystem, allowing user interaction directly from the device.
- Revised the wire legend to ensure clear distinction among the subsystems, facilitating easier debugging and maintenance.

- Updated the requirements and verification table to reflect the changes made to the system design and functionality.
- Performed a crucial calculation for storage requirements based on sensor data collection rates:
- $4 \text{ bytes per float} \times \sim 54 \text{ sensors} \times 5 \text{ Hz} \times 60 \text{ sec/min} \times 60 \text{ min/hour} \times 6 \text{ hours} \times 4 \text{ hikes} = 93.312 \text{ Megabytes}$. This calculation underscored the need for an SD card due to the ESP32 having only 512 KB of onboard memory.

Equations, Diagrams, and Figures:

- Updated block diagrams now feature the microcontroller as part of the sensing subsystem and illustrate the new connections for the control buttons within the status subsystem.



Conclusion:

The session successfully integrated essential components and updated the design documentation to reflect these changes. The calculated storage requirements led to the decision to use an SD card, ensuring sufficient data storage capacity for the project's needs.

Date: February 26, 2024

Objectives:

- Present our Smart Insole design document to the project group.
- Gather feedback for further refinement and improvement.

Work Session Record:

Today, we presented our Smart Insole design document, covering the concept, physical and software design, power requirements, and safety considerations. The device, aimed at hikers to monitor foot pressure and movement, integrates pressure sensors, accelerometers, and gyroscopes for data collection, which is then transmitted to a smartphone app for analysis and feedback.

Feedback Received:

- Incorporate thermal calculations to ensure device components operate within safe temperature limits.
- Perform power draw calculations to accurately predict device longevity during use, considering the power consumption of sensors, microcontroller, and other components.

Work Session Record:

- Reviewed the design document's sections on hardware design, focusing on the microcontroller, sensors, and power subsystem to identify areas lacking thermal and power draw considerations.
- Discussed strategies for thermal analysis, including modeling component heat generation and dissipation.
- Outlined a plan to calculate power consumption based on the operating voltage and current of each component, aiming to estimate battery life accurately.

Next Steps:

1. Conduct thermal analysis for each critical component to ensure temperature remains within operational limits.
2. Calculate total power consumption under typical usage scenarios to estimate battery life and identify potential improvements for energy efficiency.
3. Revise the design document to include these calculations and any necessary design adjustments identified through this analysis.

Notebook Entry - February 29, 2024

Objectives:

- Advance understanding and skills in PCB design for ESP32 using KiCad software.
- Learn how to incorporate various components into the PCB design.
- Understand the use of LDOs for board protection.
- Adapt an existing design to fit the requirements of the smart insole project.

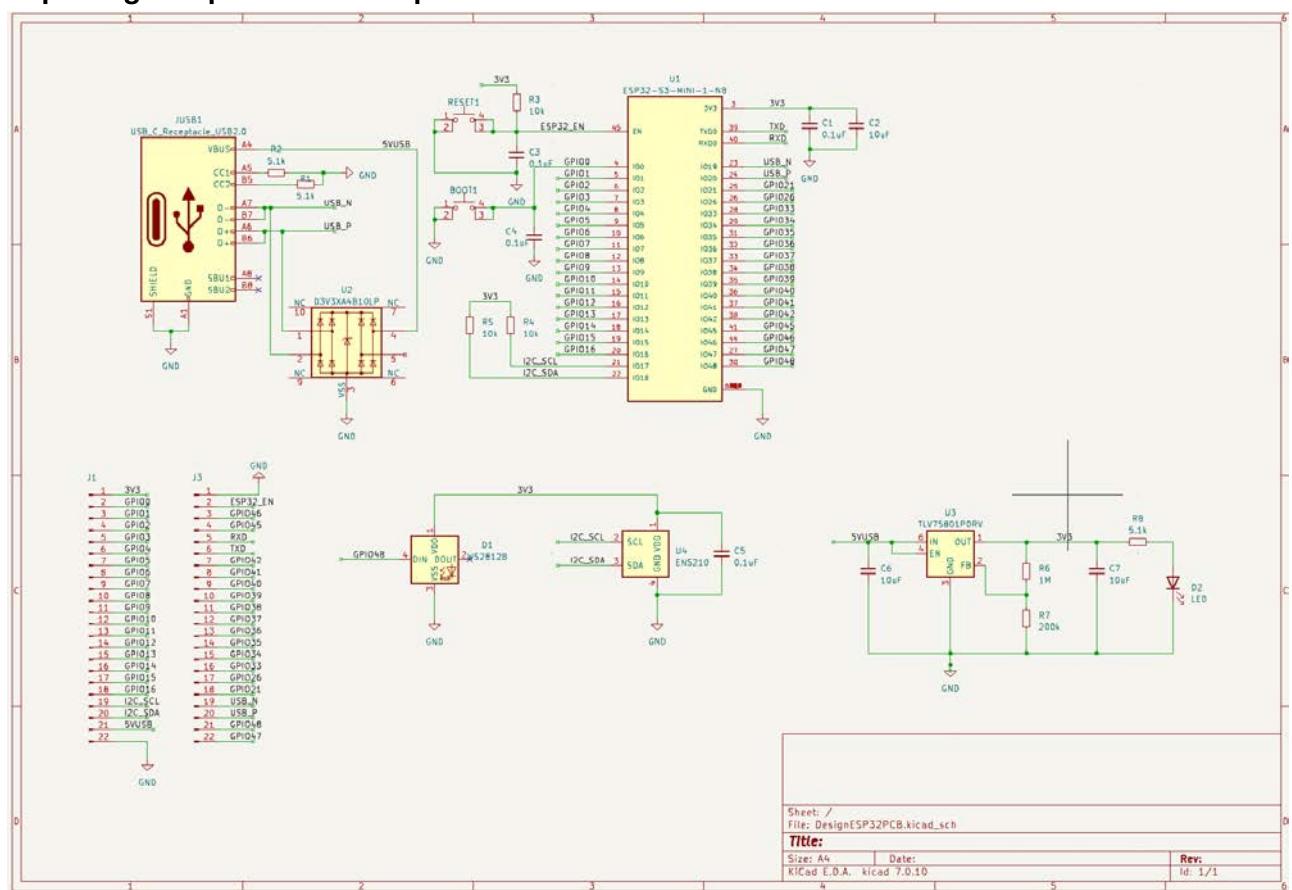
Work Session Record:

During this session, I embarked on a comprehensive tutorial that spanned approximately 4 hours, dedicated to creating a custom PCB for an ESP32 S3 chip. This tutorial was instrumental in guiding me through the process of reading humidity levels using an ENS210 module and subsequently activating an RGB light based on the humidity readings. Given that the humidity sensor's functionality was not pertinent to our smart insole project, I modified the design to exclude this component.

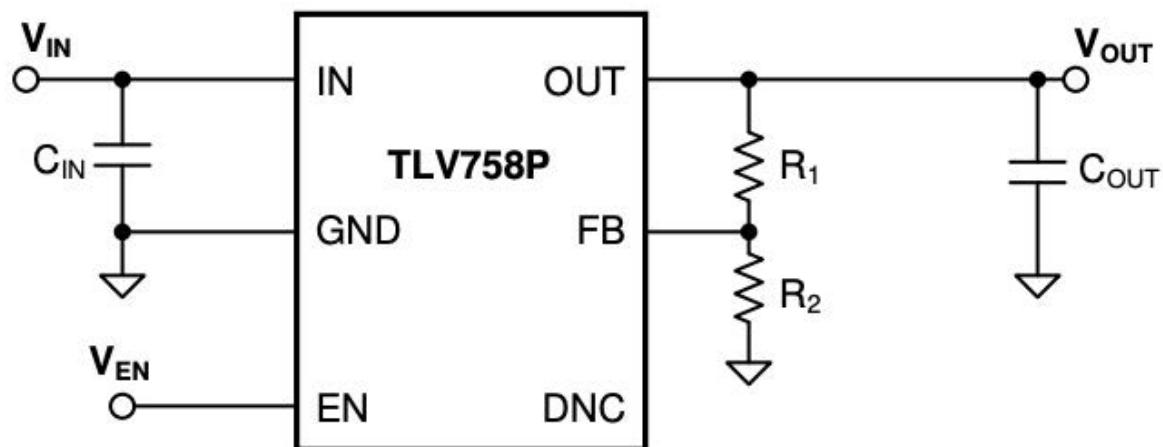
Throughout the tutorial, I acquired valuable skills and knowledge in several key areas:

figures

- Importing Footprints and Components:** schematic for the initial PCB



- Understanding and Implementing LDOs:** A significant portion of the tutorial was dedicated to Low Dropout Regulators (LDOs). I learned how LDOs are crucial for protecting the PCB by stabilizing voltage levels. Moreover, I gained insights into calculating thermal limits for LDOs, ensuring that the board operates within safe temperature ranges.



Typical Application

resources

- <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>

Equations:

The equation used for calculating the thermal limit was: $\Delta T = P_{diss} \times R_{\theta JA}$ where P_{diss} is the power dissipation and $R_{\theta JA}$ is the junction-to-ambient thermal resistance.

Conclusion:

Today's session was highly productive, significantly advancing my proficiency in PCB design for the ESP32 chip using KiCad. By following the detailed tutorial, I not only adapted an existing design for our project's needs but also deepened my understanding of crucial components and calculations needed for a successful PCB design. Moving forward, I will continue to refine the design and explore additional functionalities that could enhance the smart insole project.

Notebook Entry: March 1st

Objectives:

- Address missing footprints for buttons, resistors, capacitors, and pin headers in the PCB design from February 29.
- Ensure the ability to work asynchronously on individual devices without issues related to KiCad's library imports and paths.

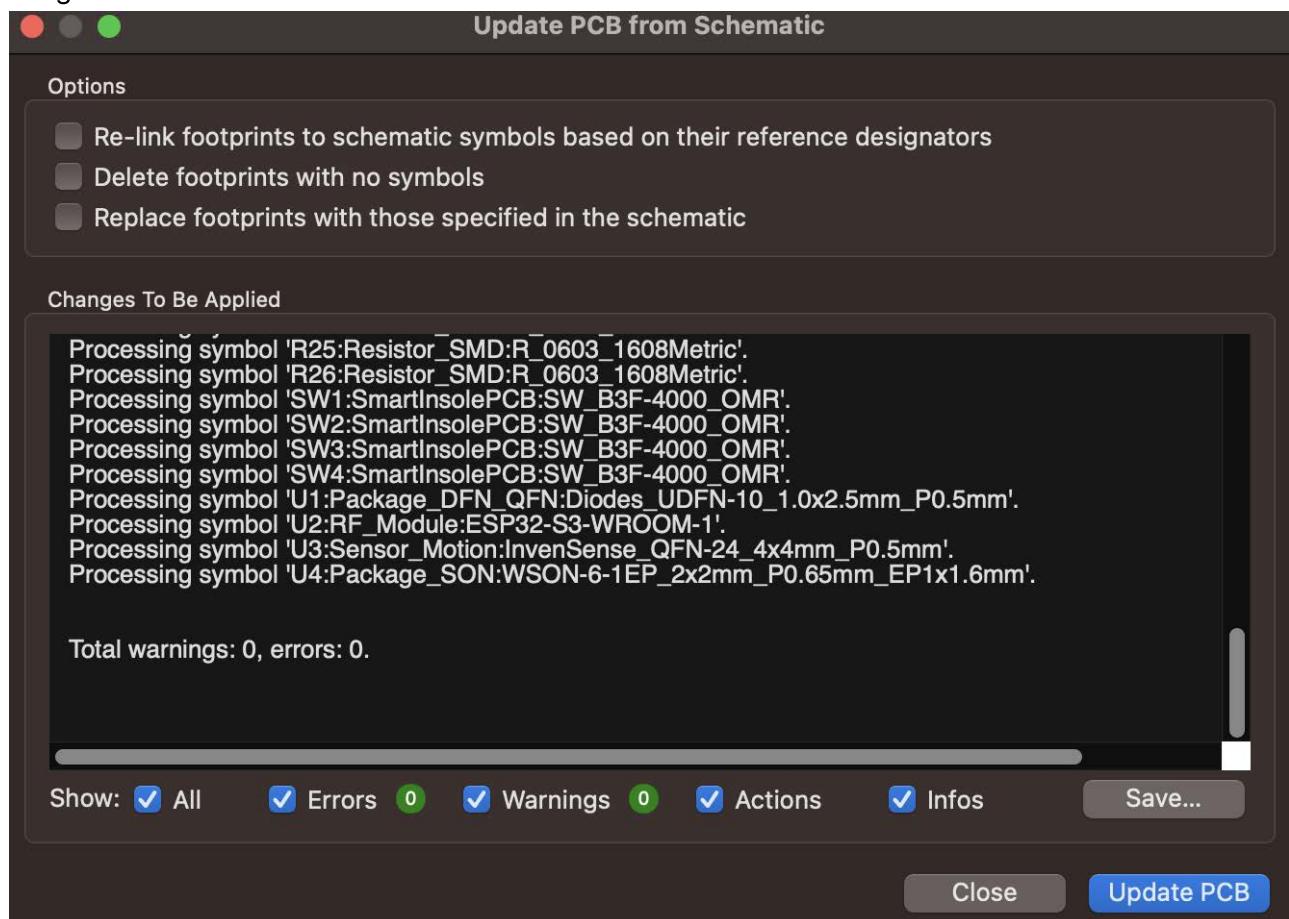
Work Session Record:

During this session, our primary goal was to rectify an oversight from the initial PCB design created on February 29. We discovered that the design was lacking essential footprints. This was relatively straightforward and we just had to use a large enough SMD component.

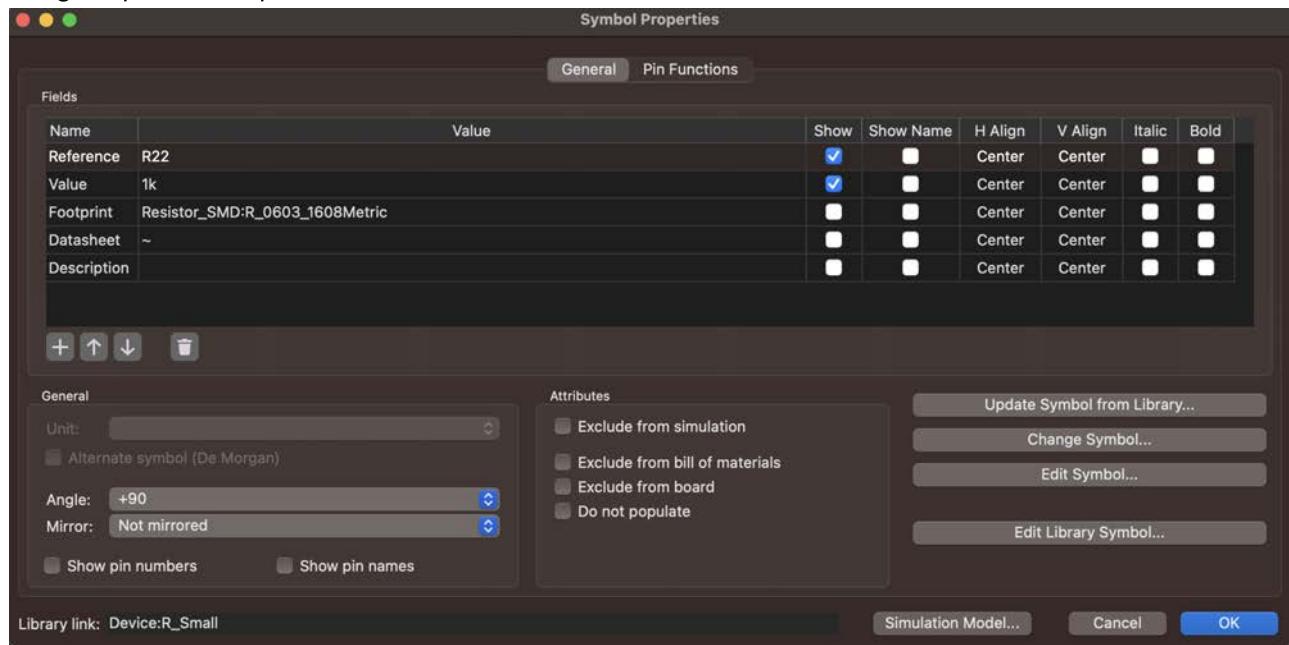
However, we encountered significant challenges related to KiCad's library management system, which hampered our ability to work efficiently across different devices. As we attempted to work asynchronously, moving the project files from one device to another led to multiple instances of library import errors and path misconfigurations. This resulted in the unlinking of previously linked footprints, causing a critical issue where updating the PCB from the schematic would fail due to these broken links. We were able to relink footprints every time we opened the project after an update had been pushed to our GitHub, but this was a very hacky solution.

Diagrams and Figures:

- Image: errors



- Image: Update Footprint



In the future, we will need to come up with a more permanent solution to migrating from device to device.

Notebook Entry: March 4th

Objectives:

- Implement debouncing for buttons in the PCB design, following the guidelines from the CAD assignment on the course wiki.

- Collaborate with the team to finalize the initial PCB schematic design.
- Attend a PCB design review session for expert advice on voltage regulation and battery considerations.

Work Session Record:

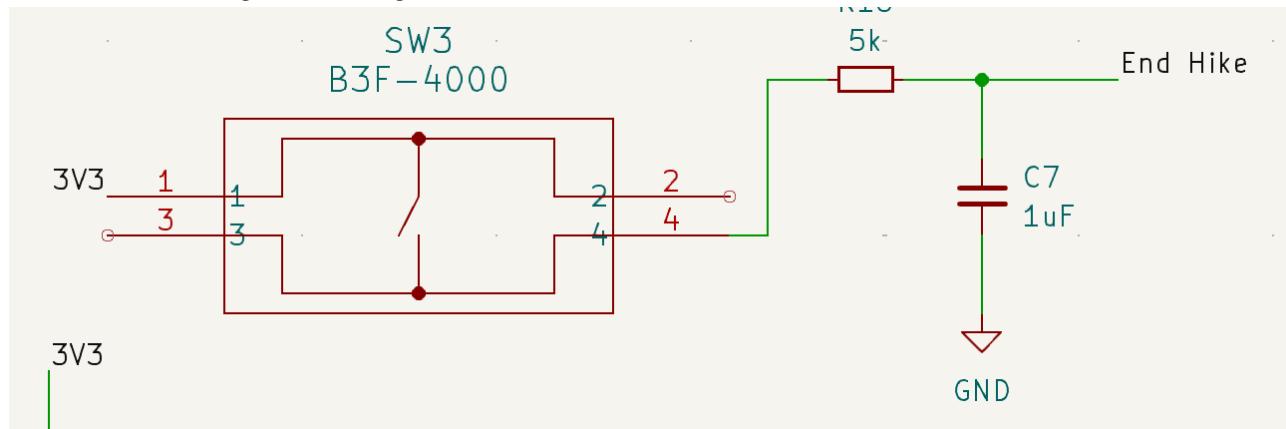
This session's agenda was primarily focused on refining our PCB design, with specific attention to debouncing the buttons, a critical step as outlined in our CAD assignment available on the course wiki. The process involved the integration of hardware debouncing mechanisms into our design to ensure reliable button operation, preventing the erroneous signals that can result from mechanical switch bounce.

Simultaneously, we worked as a team to put the finishing touches on our initial PCB schematic design. This involved a comprehensive review of all components and connections to ensure that our design was robust and free of errors. The collaborative effort was crucial in identifying and rectifying minor issues that had previously gone unnoticed, thereby strengthening the overall design.

A significant portion of the session was also dedicated to a PCB design review session, where we received guidance from a TA. One of the insights gained was the necessity of regulating voltage with our Low Dropout Regulator (LDO). The TA highlighted a potential issue with our power supply strategy, particularly concerning the use of a lithium-ion battery with a voltage range of 3.7 to 3.0 volts and an LDO with a low cutoff voltage of 3.43 volts when outputting 3.3 volts. This advice prompted us to reevaluate our power management approach to ensure that our device could operate reliably across the full range of the battery's voltage without encountering power instability due to the LDO's cutoff threshold.

Diagrams and Figures:

- Button Debouncing Circuit Diagram



This session was pivotal in advancing our PCB design project, bringing us closer to a finalized schematic that meets both our functional requirements and the technical specifications outlined in our course. The insights from the PCB design review session were particularly helpful, underscoring the importance of power management and component selection in ensuring the reliability and longevity of our design. Future sessions will be dedicated to adjusting our power supply strategy and continuing to refine our schematic based on the feedback received.

Notebook Entry: March 5

Objectives:

- Conduct routing and placement of SMD components.
- Address and correct DRC violation errors.

Work Session Record:

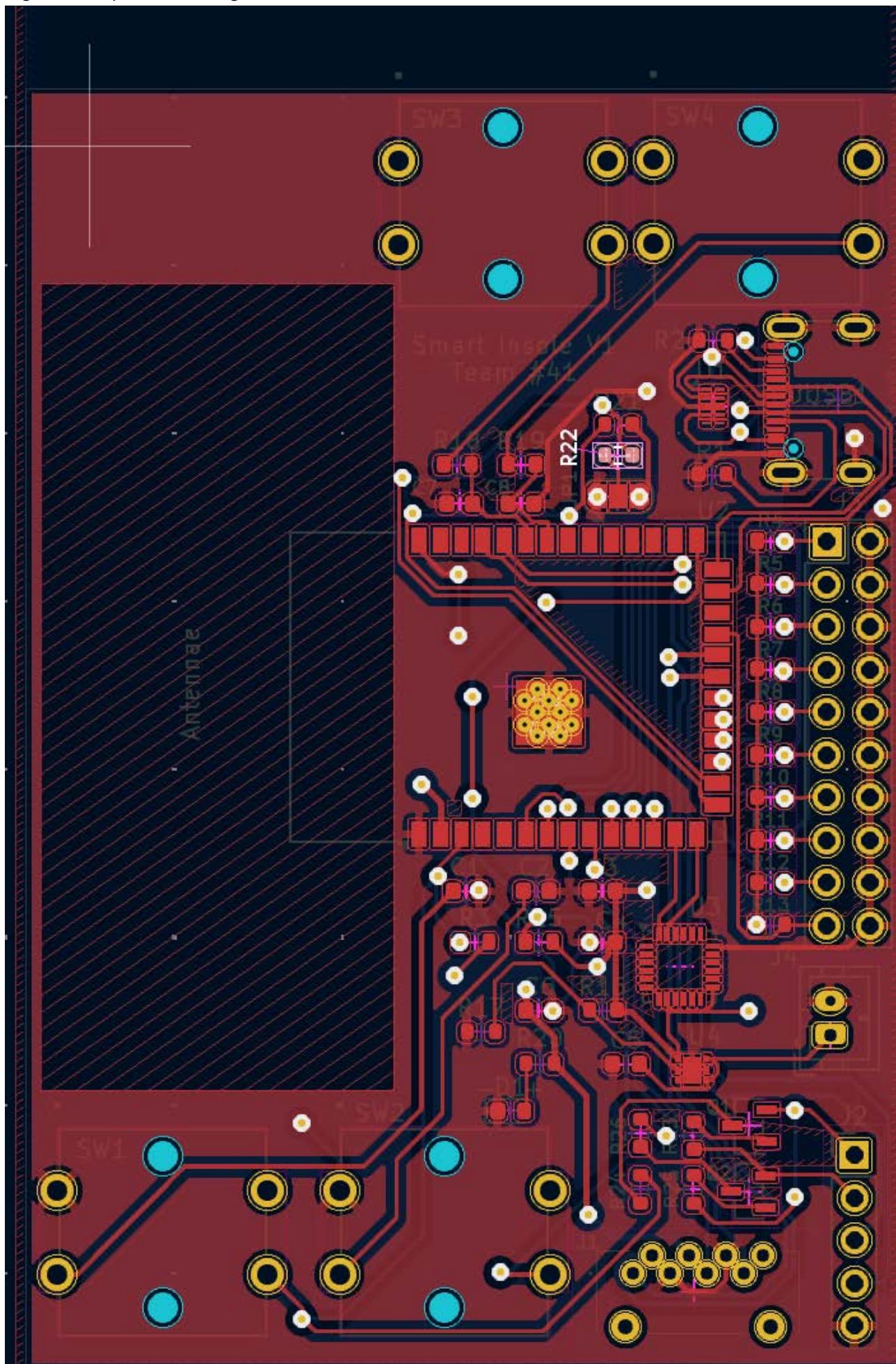
In previous work sessions, I followed an online tutorial intended for an ESP32 mini chip. However, this approach proved to be somewhat problematic as the tutorial's specifications did not align with the ESP32 microcontroller models we had access to in our lab. This discrepancy required significant modifications to the circuitry, particularly in the method through which we'd be interfacing with the board, along with a few minor adjustments to ensure compatibility with our hardware.

In hindsight, a more thorough preliminary research phase could have saved considerable time and effort. I later discovered that our course website offers example PCB files specifically tailored to the lab's hardware, which would have been an ideal starting point for our project. This realization underscored the value of consulting available resources and seeking guidance from teaching assistants before delving into design work.

The bulk of the session was consumed by the routing and placement of Surface-Mount Device (SMD) components on our PCB. The goal was to optimize the layout to prevent overlapping connections among components, a task that required careful planning and spatial awareness.

Figures:

- Figure: Completed Routing of PCB



Through the design and debugging process, I have developed a solid understanding of basic CAD mechanics, which I anticipate applying to future projects.

Notebook Entry: March 6

Objectives:

- Conduct routing and placement of Surface-Mount Device (SMD) components on our PCB.
- Optimize component layout to prevent overlap and ensure efficient use of space.
- Address and resolve Design Rule Check (DRC) violation errors, specifically focusing on the min angular width violation.

Work Session Record:

The focus of this session was on the placement and routing of SMD components for our PCB project. The challenge lay in strategically organizing the components to avoid any overlap of connections, which required a careful consideration of each component's footprint and the available board space. This task was critical in ensuring that our design was not only functional but also adhered to best practices in PCB layout design.

Upon completing what we initially believed to be the final design, I took the lead in addressing the DRC violation errors that emerged. DRC is a crucial step in PCB design, as it helps identify and rectify issues that could lead to manufacturing defects or functional failures.

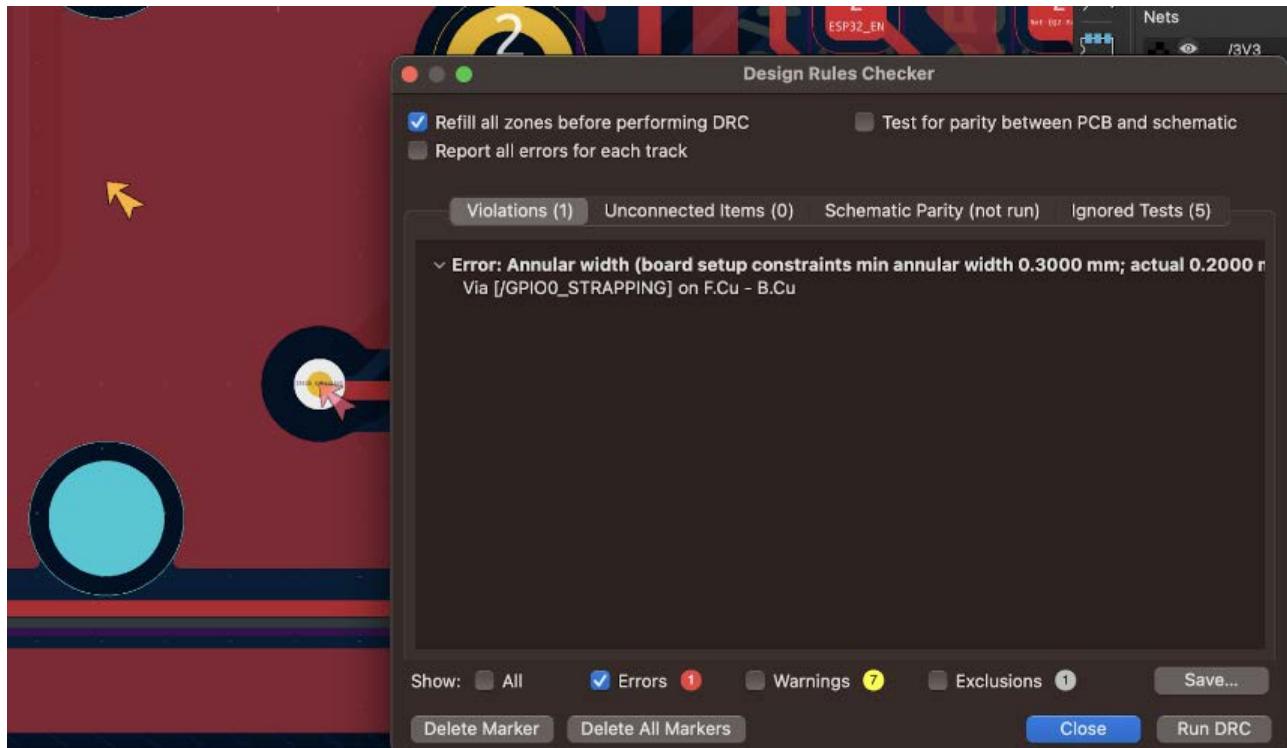
One of the most prevalent issues we encountered was the min angular width violation. This particular error proved to be a significant obstacle due to the small size of the pins on the components we had chosen. The violation was primarily because the pin dimensions fell below the minimum width allowed by our design rules, which could potentially lead to problems in the manufacturing process.

To mitigate this issue, we decided to increase the diameter of the ground pads associated with the problematic components. This adjustment was carefully calculated to ensure that it would not introduce new layout challenges, while also resolving the min angular width violation. The process involved detailed analysis and modification of the PCB design to accommodate the larger pad sizes without compromising the overall layout or the functionality of the board.

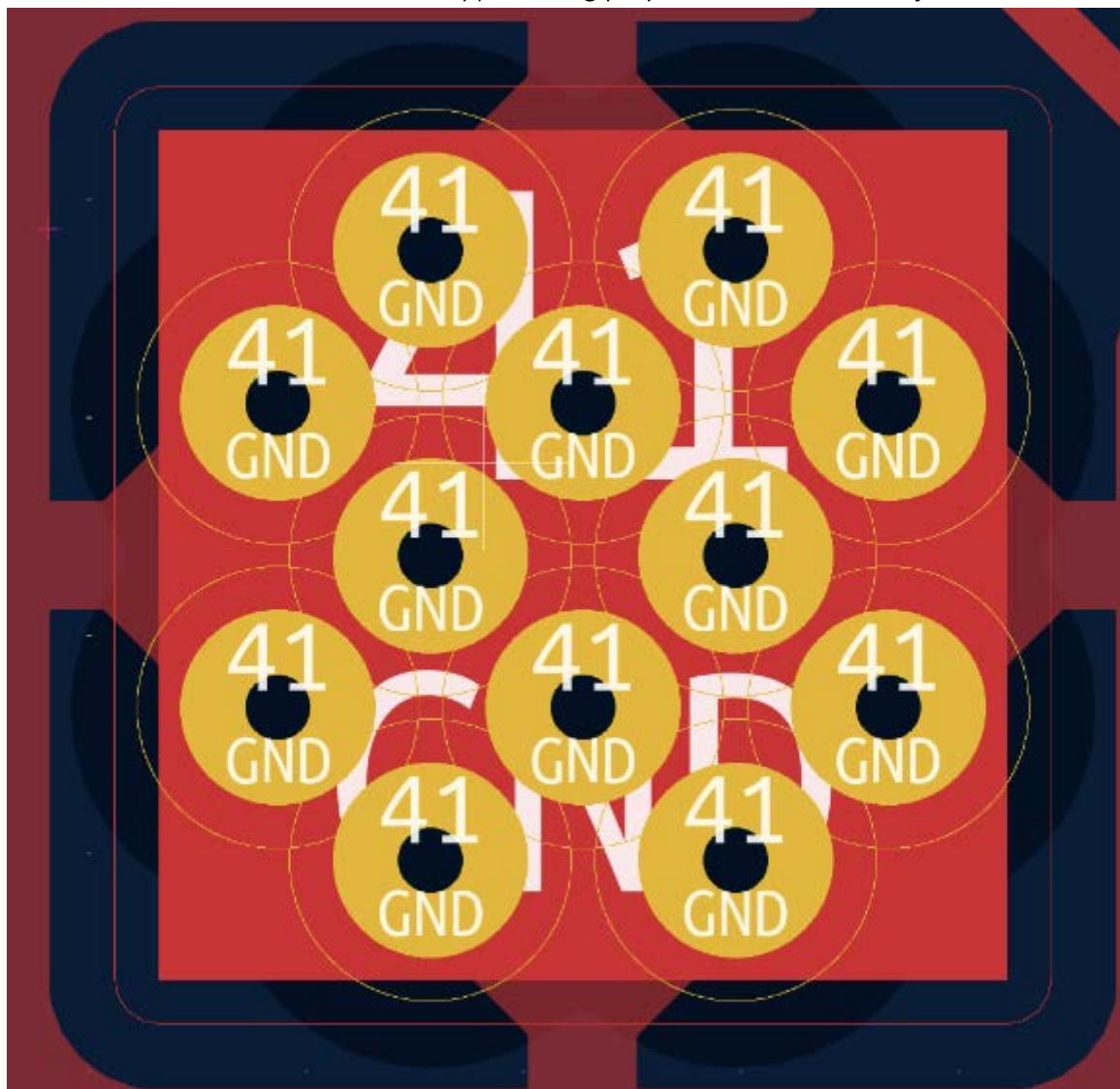
Figures:

- **Figure 1: DRC Violation Errors** - Highlighting the most common issue we encountered, the minimum angular width violation. This error was particularly challenging due to the small pin sizes of the

components we selected, necessitating an increase in the diameter of the ground pads for resolution.



- **Figure 2: Ground Pins for the ESP32 Chip** - Demonstrates the adjustments made to ground pad sizes to accommodate the ESP32 chip, ensuring proper fit and connectivity.



This session was geared towards advancing our PCB design towards finalization. By addressing the DRC violation errors, specifically the challenging min angular width violation, we moved closer to ensuring that our design would be manufacturable and function as intended. In this session, we ended up reducing our total DRC violation errors from ~200 to 43 errors.

Notebook Entry: March 18

Objectives:

- Correct programming and boot issues on PCB identified by Jason at office hours

Work Session Record:

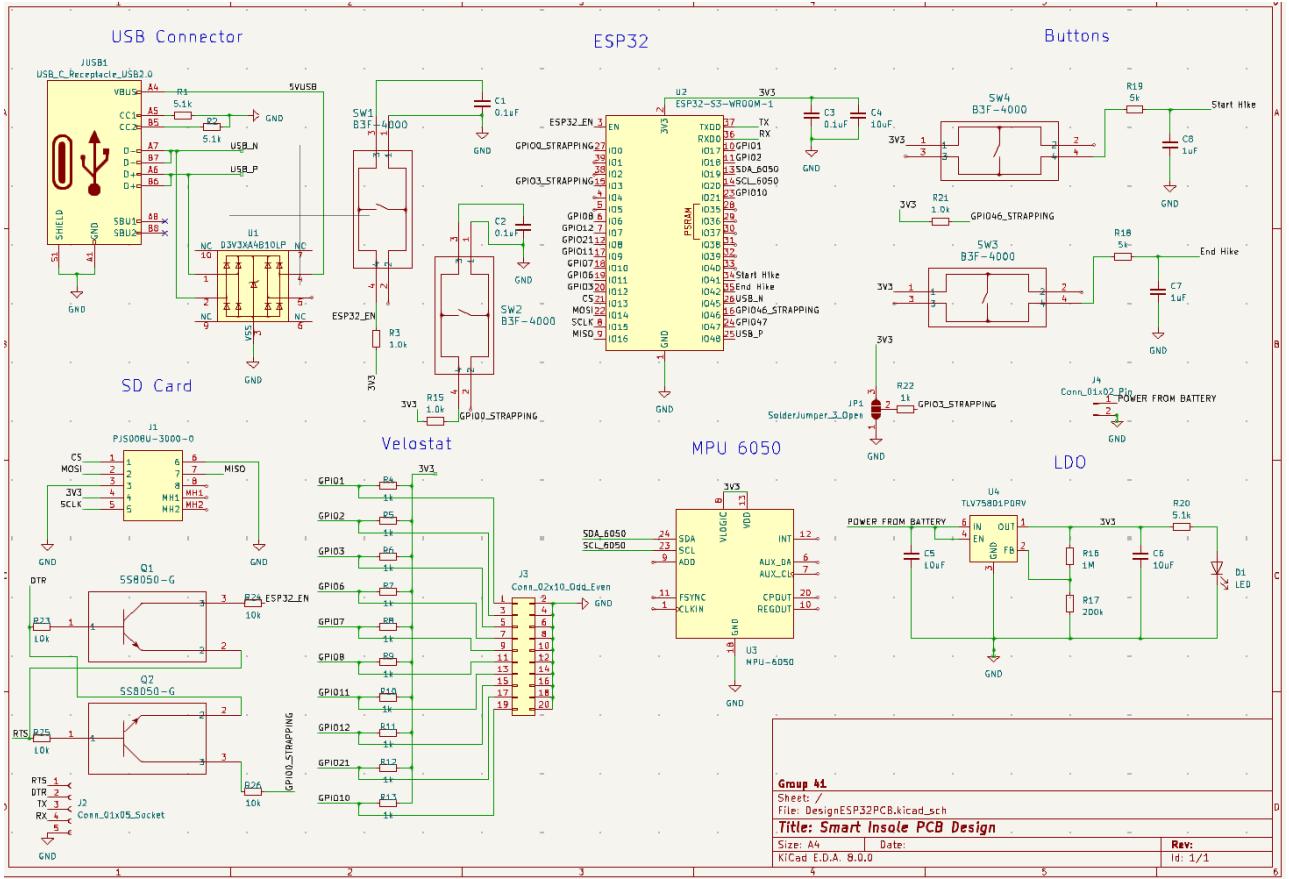
In previous work sessions, I followed an online tutorial intended for an ESP32 mini chip. However, this approach proved to be somewhat problematic as the tutorial's specifications did not align with the ESP32 microcontroller models we had access to in our lab. This discrepancy required significant modifications to the circuitry, particularly in the method through which we'd be programming the board, along with a few minor adjustments to ensure compatibility with our hardware. I had thought previously to just use the USB connection we had to program and interface with the ESP 32 chip. However Jason brought it to our attention in office hours that this would be a very finicky method and we should instead use UART headers in order to program and test our PCB design.

In hindsight, a more thorough preliminary research phase could have saved considerable time and effort. I later discovered that our course website offers example PCB files specifically tailored to the lab's hardware, which would have been an ideal starting point for our project. This realization underscored the value of consulting available resources and seeking guidance from teaching assistants before delving into design work.

The bulk of the session was consumed by the reworking of our schematic to include the required ESP 32 programming circuitry.

Figures:

- Figure: Completed Routing of PCB



Through the design and debugging process, I have developed a solid understanding of basic CAD mechanics, which I anticipate applying to future projects.

Notebook Entry: March 19 2024

Objectives:

- Conduct routing and placement of Surface-Mount Device (SMD) components on our PCB.
- Optimize component layout to prevent overlap and ensure efficient use of space.
- Address and resolve Design Rule Check (DRC) violation errors, specifically focusing on the min angular width violation.

Work Session Record:

The focus of this session was on the placement and routing of SMD components for our PCB project. The challenge lay in strategically organizing the components to avoid any overlap of connections, which required a careful consideration of each component's footprint and the available board space. This task was critical in ensuring that our design was not only functional but also adhered to best practices in PCB layout design. These changes were needed as we updated our PCB to reflect changes to the programming header.

I took the lead in addressing the DRC violation errors that emerged. DRC is a crucial step in PCB design, as it helps identify and rectify issues that could lead to manufacturing defects or functional failures.

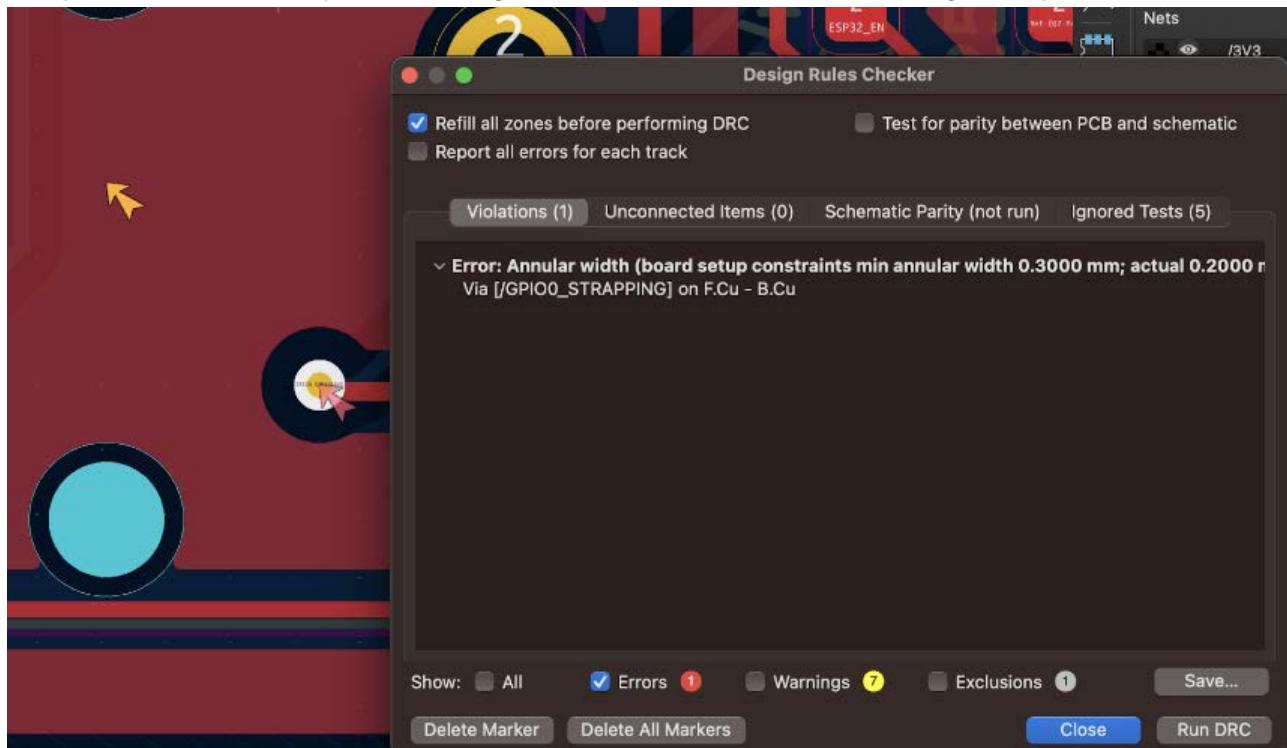
One of the most prevalent issues we encountered was the min angular width violation. This particular error proved to be a significant obstacle due to the small size of the pins on the components we had chosen. The violation was primarily because the pin dimensions fell below the minimum width allowed by our design.

rules, which could potentially lead to problems in the manufacturing process. This problem was similar to that of March 6th but we were needing to complete and fix these errors in order to place our PCB order by 4:45 Tuesday

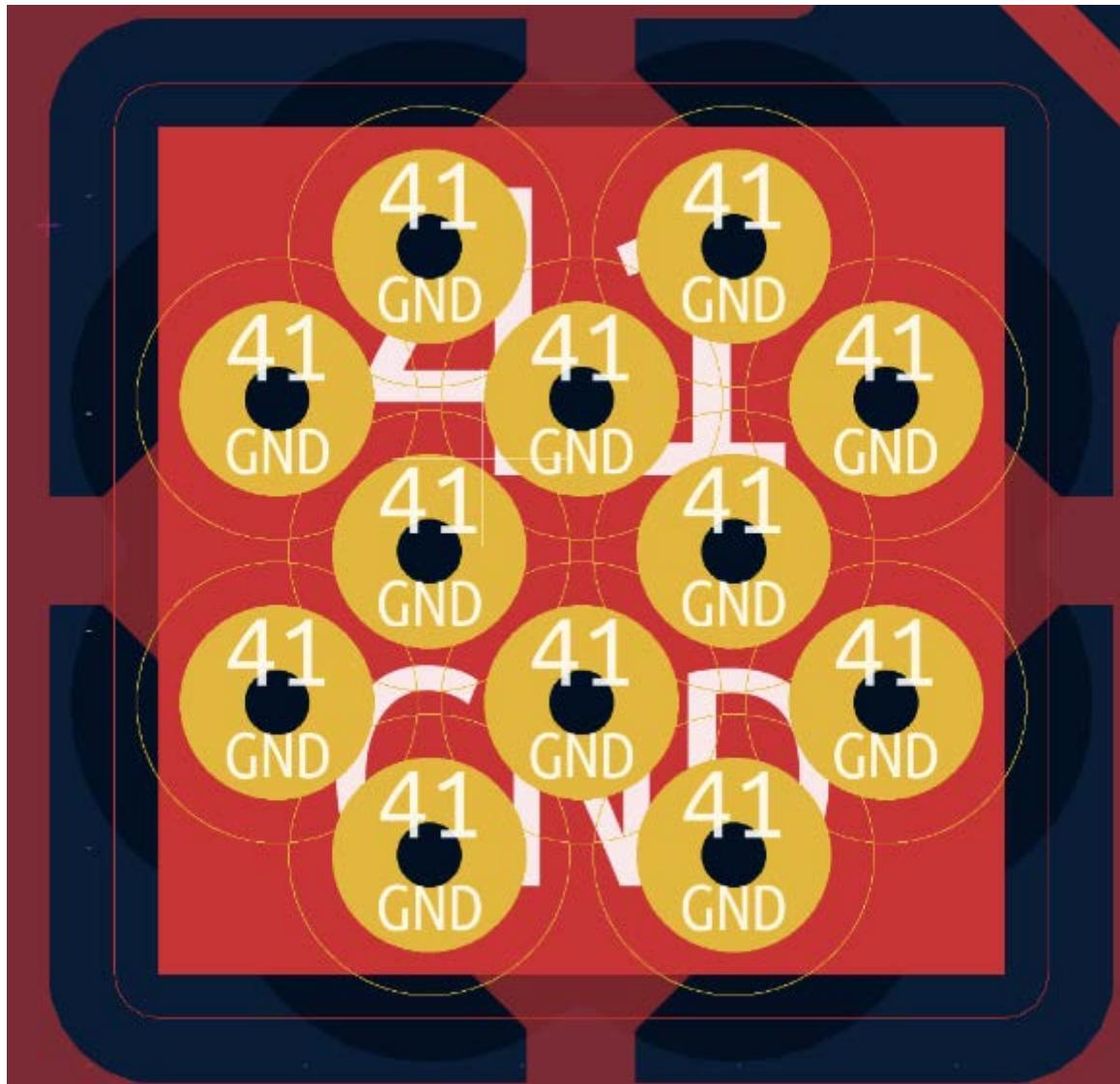
Upon finishing our DRC checks I took the lead in working to generate the Gerber files and uploading them to PCB for a final DRC check. This ultimately was error free which was super nice as we didn't have to fix anything last minute. However it did take a lot longer than expected to get the gerber files reviewed on PCB. ~1 hour vs the estimated 15 minutes.

Figures:

- **Figure 1: DRC Violation Errors** - Highlighting the most common issue we encountered, the minimum angular width violation. This error was particularly challenging due to the small pin sizes of the components we selected, necessitating an increase in the diameter of the ground pads for resolution.



- **Figure 2: Ground Pins for the ESP32 Chip** - Demonstrates the adjustments made to ground pad sizes to accommodate the ESP32 chip, ensuring proper fit and connectivity.



This session was geared towards advancing our PCB design towards finalization. By addressing the DRC violation errors, specifically the challenging min angular width violation, we moved closer to ensuring that our design would be manufacturable and function as intended. In this session, we ended up reducing our total DRC violation errors from ~200 to 43 errors.

Notebook Entry: March 25 2024

Objectives:

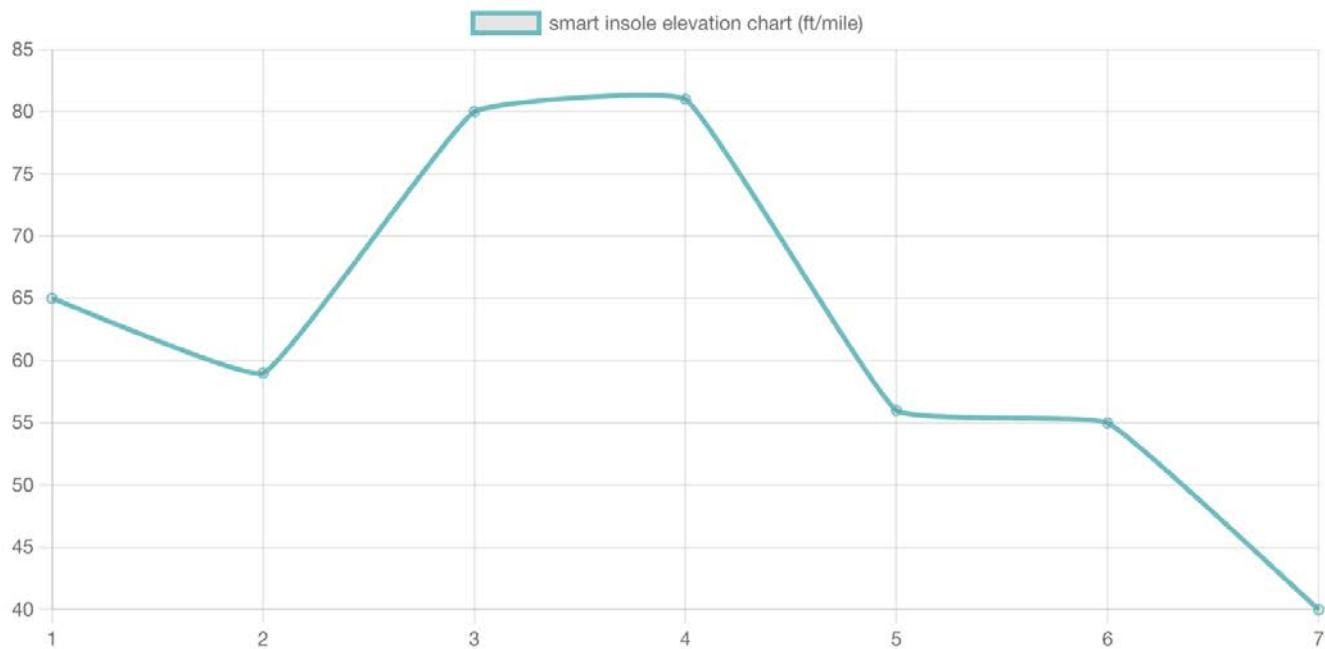
- create web app
- test out data display
- learn how to manipulate data within web app

Work Session Record:

looked into several web development frameworks for our Smart Insole project and determined that plain HTML lacked the advanced features needed for Bluetooth connectivity and dynamic data visualization, while Next.js was more complex than required due to us not needing a backend. React emerged as the best choice due to its balance of functionality and ease of use, along with its support for NPM modules. I then initialized a new React application, removing unneeded boilerplate to tailor it to our project's requirements.

Additionally, I explored the Chart.js module, finding it versatile for our needs to display data from the Smart Insole, making it the selected tool for our data visualization efforts.

Figures:



Notebook Entry: March 31 2024

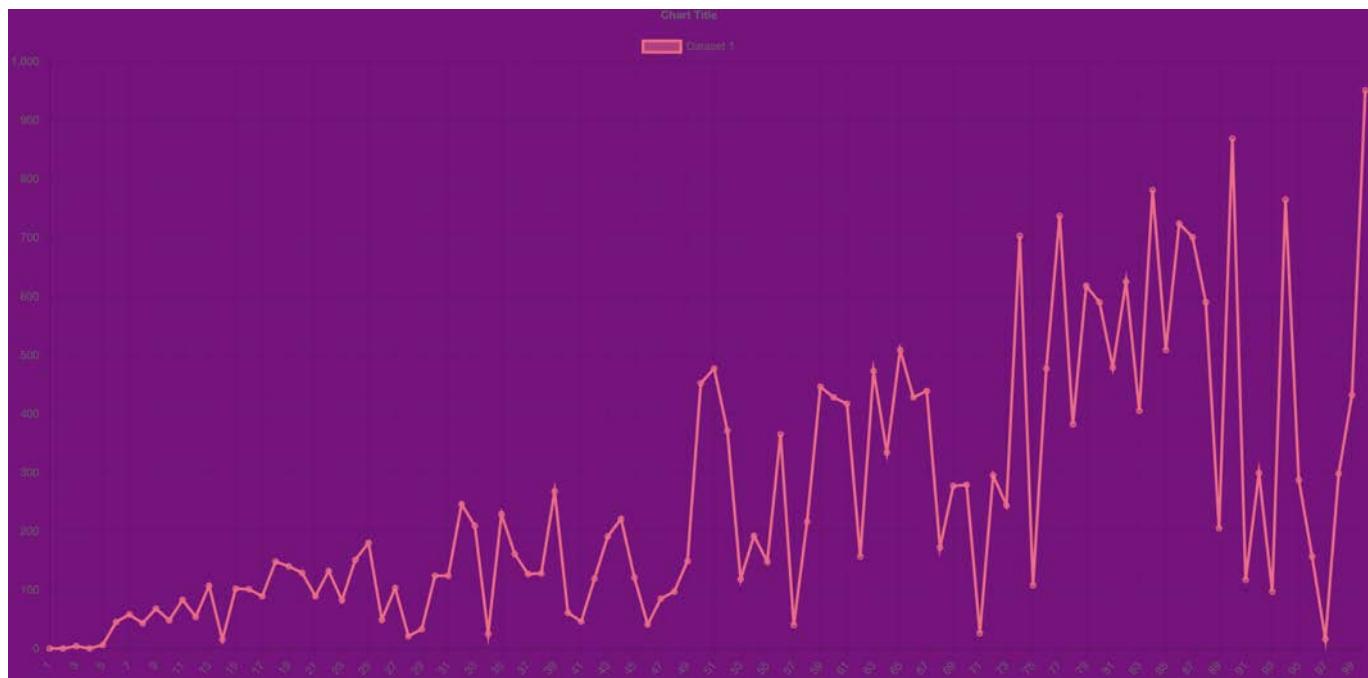
Objectives:

- work on adding chart.js to react app
- allow it to ingest data in list format that we would send it
- for now display random data

Work Session Record:

Worked with react to install and integrate the chart.js NPM module. This module allows us to use charts like line, pie, bar, bubble, scatter, radar, and more. We will use this to display user data in an easily understandable way. I installed the module using `npm install chart.js react-chartjs-2`. This was a point of contention as I spent a lot of time trying to use chart.js which doesn't integrate with react well. After which I googled how to add this in react and there were some helpful links. From here it was easy to integrate the module and display random data on a line chart which will be most useful in displaying user data. This will be used for elevation gain charts, distance charts, step charts, average speed charts, and many more.

Figures:



Works Cited:

- <https://blog.logrocket.com/using-chart-js-react/>

Notebook Entry: April 2 2024

Objectives:

- work with dev board to transfer data from esp32 to web interface
- transmit multiple types of data
- transmit data to the dev board to control it
- send acknowledge packets to device to indicate successful data transfer

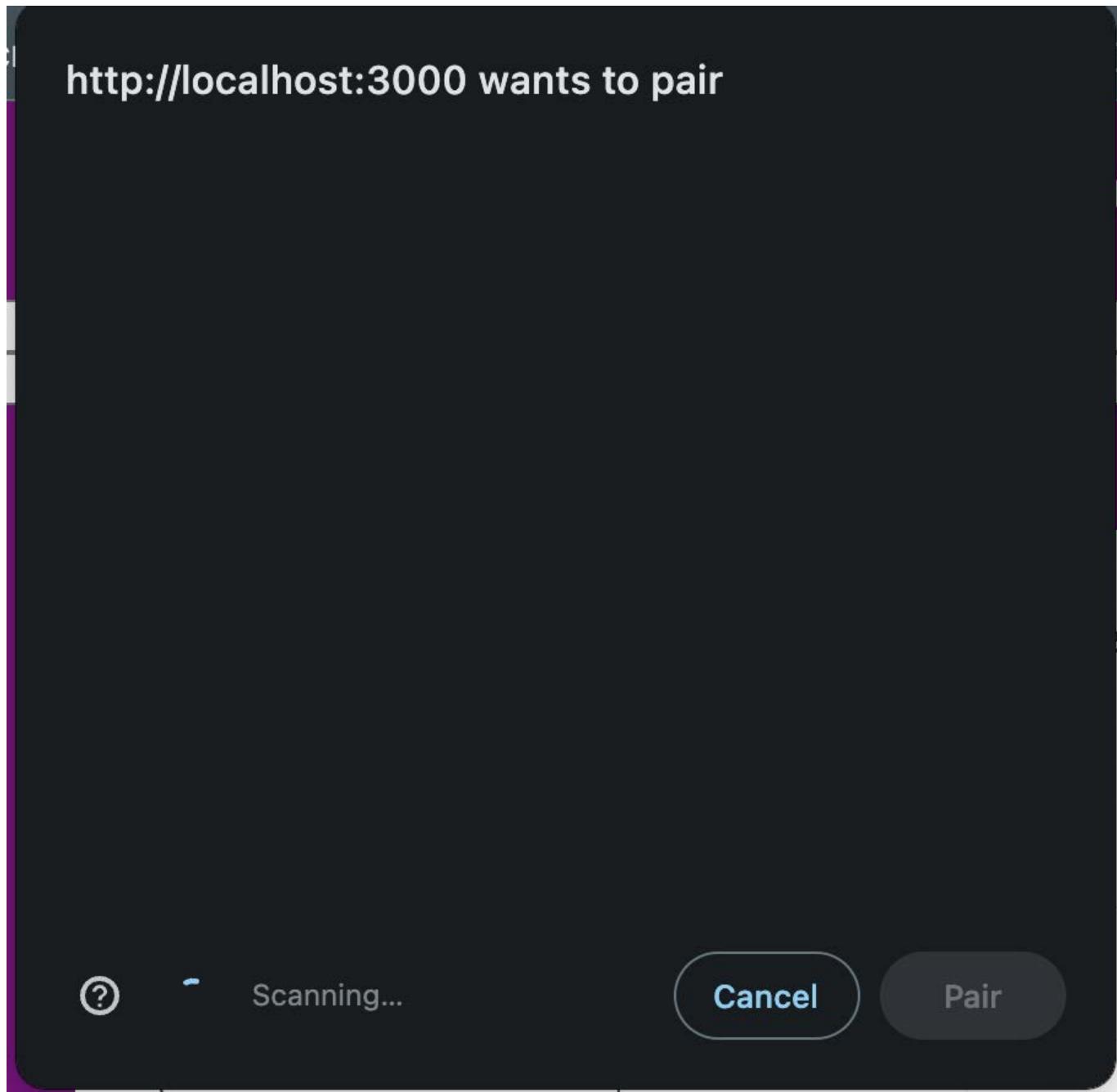
Work Session Record:

Worked with Tony to write arduino code to set up a bluetooth server that our web interface could connect to and communicate with. I was in charge of the react side while Tony worked on setting up his drivers to program the ESP 32 dev board with the Arduino IDE. We stumbled upon a super helpful resource which detailed arduino and html code which would allow up to receive an incrementing number upon connection of a device to the ESP 32. This was done with HTMl and needed to be highly adapted to our React interface. For example the tag selection doesn't work with React and instead we use onclick={} properties. Another thing we needed to correct was the dynamic updates and to do this we had to use the useState react hook. This eventually worked and was able to connect with the programmed esp32 from Tony. I then edited the code to accept more than just integer values.

Code

- `const [variable, setVariable] = useState("")`
- `onclick={function}`

Figures:



A screenshot of a mobile application interface. At the top, the text "Fetched Value" is displayed. Below it, the text "Last reading:" is shown. Underneath that, the text "Control GPIO 2" is displayed. There is a toggle switch with the options "ON" and "OFF". At the bottom, the text "Last value sent:" is shown.

Works Cited:

- <https://randomnerdtutorials.com/esp32-web-bluetooth/>

Notebook Entry: April 4 2024

Objectives:

- work building the ESP 32 PCB

- *test the ESP32 with programming and our web interface*

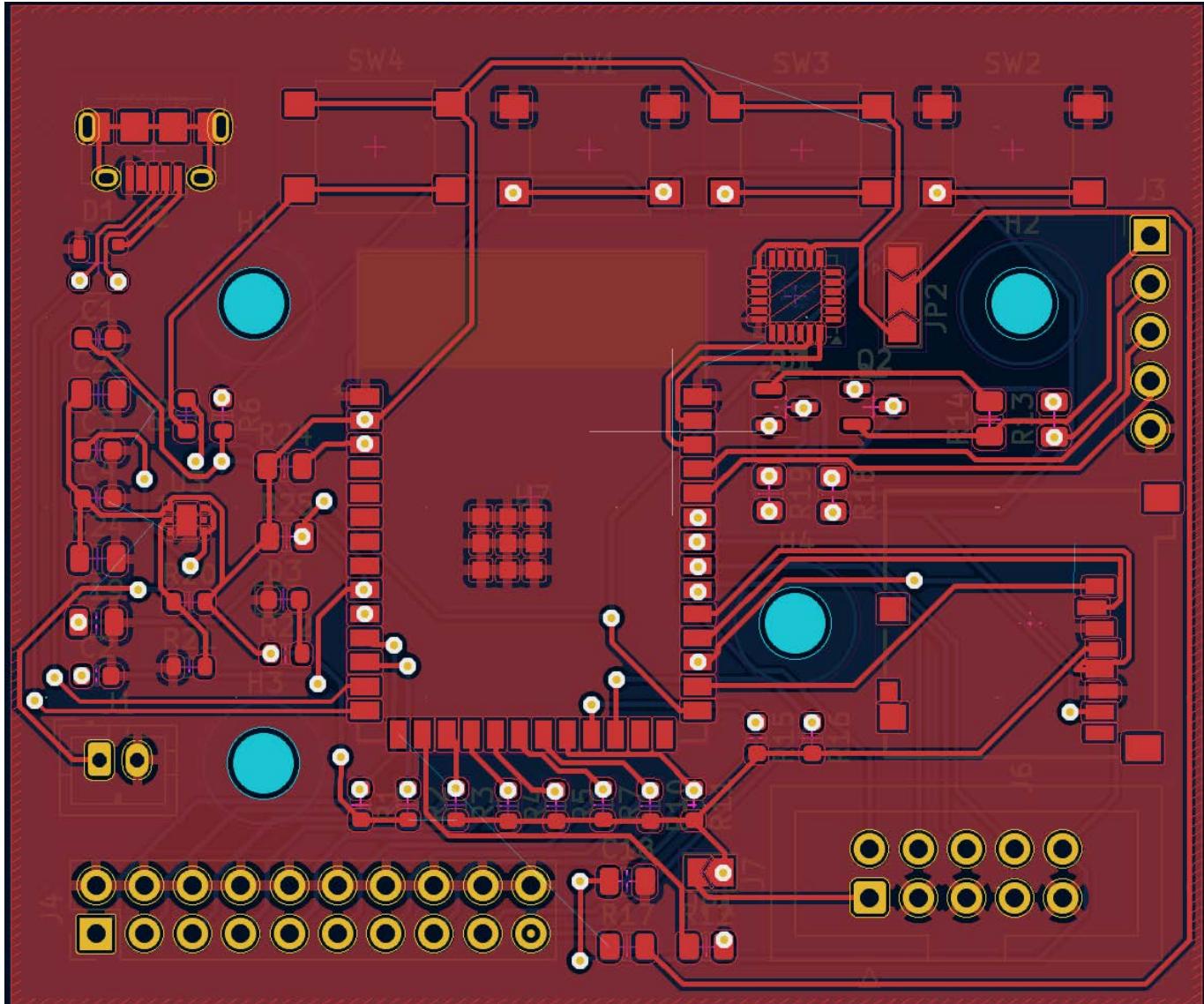
Work Session Record:

Worked with Alyssa to solder on our SMD components with the help of solder paste and a solder mask. This was a learning process as we first used too much solder and it bridged the pads. However next we used too little and the component didn't connect to the pads. After which we used a tad too much making some pads bridged and some not. At this point we looked into programming and realized that the pins supposed to connect to the programming pins were not mapped right. D+ was supposed to be mapped to GPIO 19 and D- was supposed to be on GPIO20. After further analysis into our board connections we realized that most of the GPIO had special functions and couldn't be used for general purpose. Other areas we messed up were our strapping.

- Chip boot mode – GPIO and GPIO46
- VDD_SPI voltage – GPIO45
- ROM messages printing – GPIO46
- JTAG signal source – GPIO3

We mapped these pins willy nilly or just didn't have them and didn't realize that they couldn't be remapped. As such we decided to not continue with the PCB creation. Instead we moved on to redoing our PCB to account for the mismapped pins. Another thing we realized was wrong was that we had to map out the voltage divider to the Analog to Digital compatible pins in order to read our voltage outputs reliably. As such we took this time to redo our PCB and get an initial version to look over and adjust after.

Figures:



Notebook Entry: April 5 2024

Objectives:

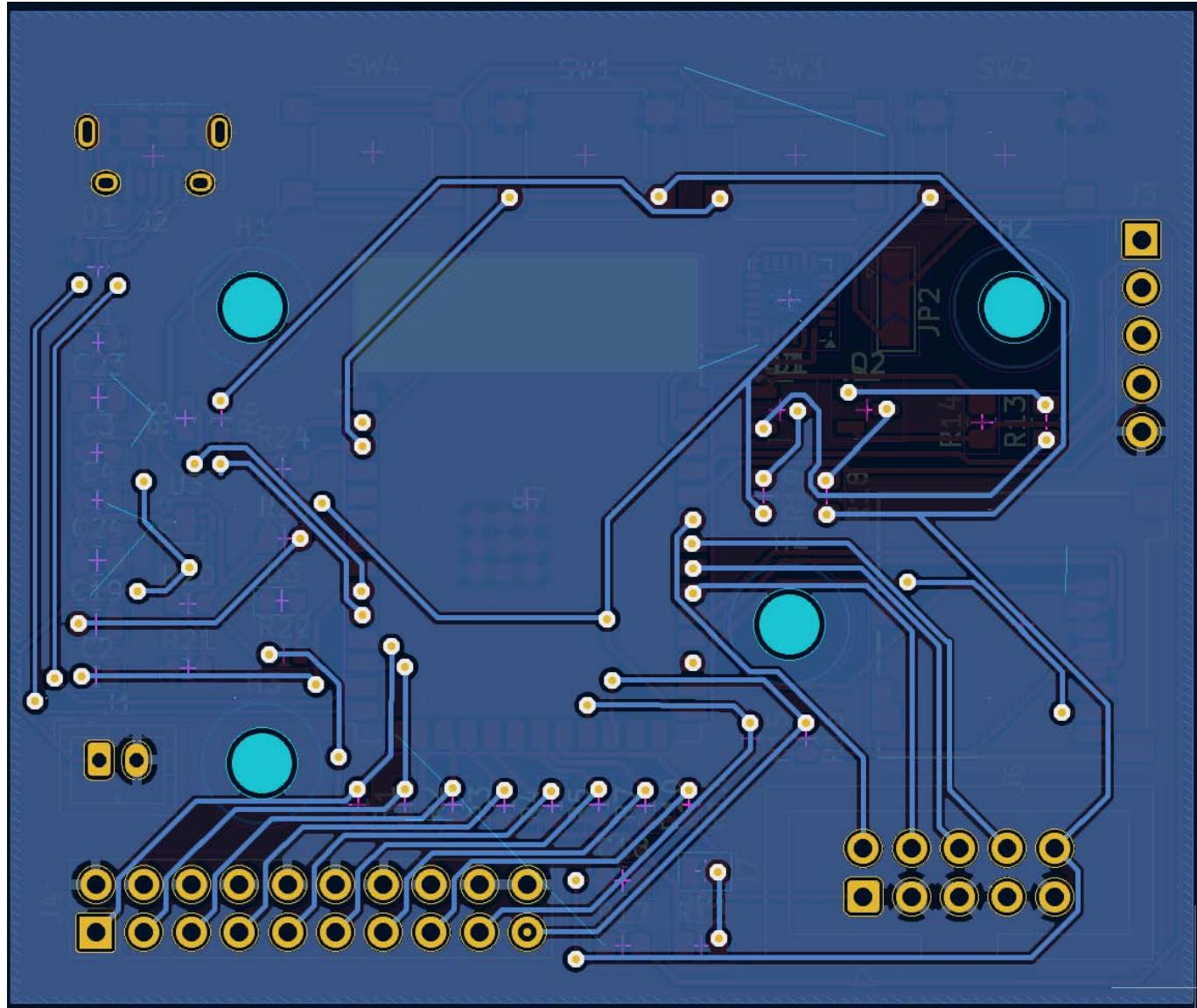
- Correct DRC errors on Initial PCB from yesterday
 - Work on making sure connections are right on our PCB
 - Work with Alyssa to prune board of unnecessary components

Work Session Record:

Worked to fix DRC errors when making the board. These included the ones detailed earlier where we were having min angular width errors on footprints of some of our components including the Micro USB connector, the SD card holder, and a few other such components. Throughout this process we worked to add in vias for circuit protection. Additionally we had to connect isolated islands or reconnect components such that we didn't end up with these islands. This process was quite long but can be summed up by saying googled and corrected errors on our PCB in Kicad.

After I worked on correcting the DRC errors Alyssa wanted to make the PCB smaller. As such we discussed components to remove and decided on removing mounting holes and buttons on the board to reduce board size so it is easier to mount to a shoe. This reduced the size quite a bit but caused Alyssa to take over on correcting the new DRC errors.

Figures:



Notebook Entry: April 9 2024

Objectives:

- Get a PCB made and correct connection errors
- Get TA help to look over and verify our PCB board
- send PCB gerber files out for PCBWay printing

Work Session Record:

We needed to remake a bunch of connections again as we realized that we needed to remap channel 2 of our analog to digital pins to channel 1. This was because when the bluetooth module is active we couldn't read from these pins as the bluetooth module would be using them. Additionally I had to conduct thermal analysis knowing the bluetooth module would use 300 mA of power which our LDO would drop 1.7 V of power. This means that .51 Watts of power would be dissipated by the LDO. This seems to be ok but may need a heatsink on top to reduce this power draw.

We realized that the bluetooth module when on would deactivate channel 2 for Analog to Digital compatible pins and as such had to account for that by remapping pins. However we were unable to remap all pins and

as such had to think of a different solution. I thought to add another button to activate bluetooth pairing or deactivate it while a hiker wanted to offload data. This would shut off the data recording and would allow for us to validate that we wouldn't use channel 2.

Alongside this I fixed many DRC errors and also remapped wires. Additionally I placed an updated SD card footprint on the board as our original one was throwing an error which neither the TA, Google nor I could figure out. Jason at his office hour looked over our board and realized we needed to move our micro usb to the edge which I did.

Some errors I fixed in office hours were:

- foot print expected though hole got SMD
- Same error on example ESP 32 board and Jason confirmed this was expected
- min spoke count expected 2 got 1 when copper ground sheets only connect by one direction to ground pins.
- Same error on example ESP 32 board and Jason confirmed this was expected
- min angular width of default footprints for micro USB
- Same error on example ESP 32 board and Jason confirmed this was expected
- rear solder mask bridges items with different widths on the solder jumper for the strapping pins
- Same error on example ESP 32 board and Jason confirmed this was expected

Figures:

Design Rules Checker

Refill all zones before performing DRC Test for parity between PCB and schematic
 Report all errors for each track

Violations (14) Unconnected Items (0) Schematic Parity (not run) Ignored Tests (3)

PTH pad 6 [unconnected-(J2-Shield-Pad6)] of J2

✓ **Error: Annular width (board setup constraints min annular width 0.3000 mm; actual 0.1950 mm)** PTH pad 6 [unconnected-(J2-Shield-Pad6)] of J2

✓ **Warning: Copper connection too narrow (board setup constraints minimum connection width 0.1000 mm; actual 0.0500 mm)** Zone [GND] on F.Cu

✓ **Error: Footprint component type doesn't match footprint pads (expected 'Through hole'; actual Footprint J2)**

✓ **Error: Thermal relief connection to zone incomplete (layer F.Cu; zone min spoke count 2; actual 1)** Zone [GND] on F.Cu
Pad 3 [GND] of J6 on F.Cu

✓ **Error: Thermal relief connection to zone incomplete (layer F.Cu; zone min spoke count 2; actual 1)** Zone [GND] on F.Cu
Pad 6 [GND] of J6 on F.Cu

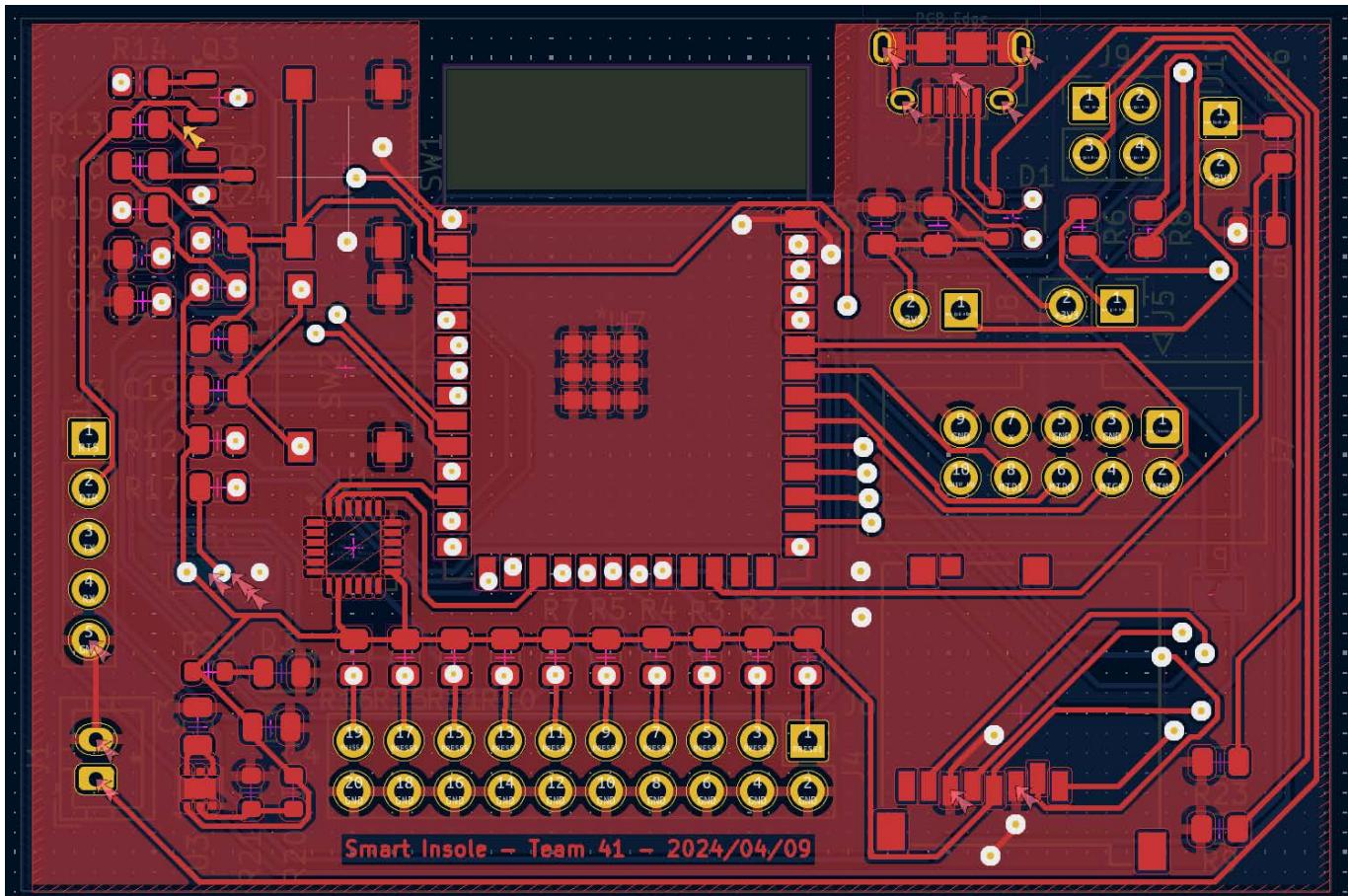
✓ **Error: Thermal relief connection to zone incomplete (layer B.Cu; zone min spoke count 2; actual 1)** Zone [GND] on B.Cu
PTH pad 5 [GND] of J3

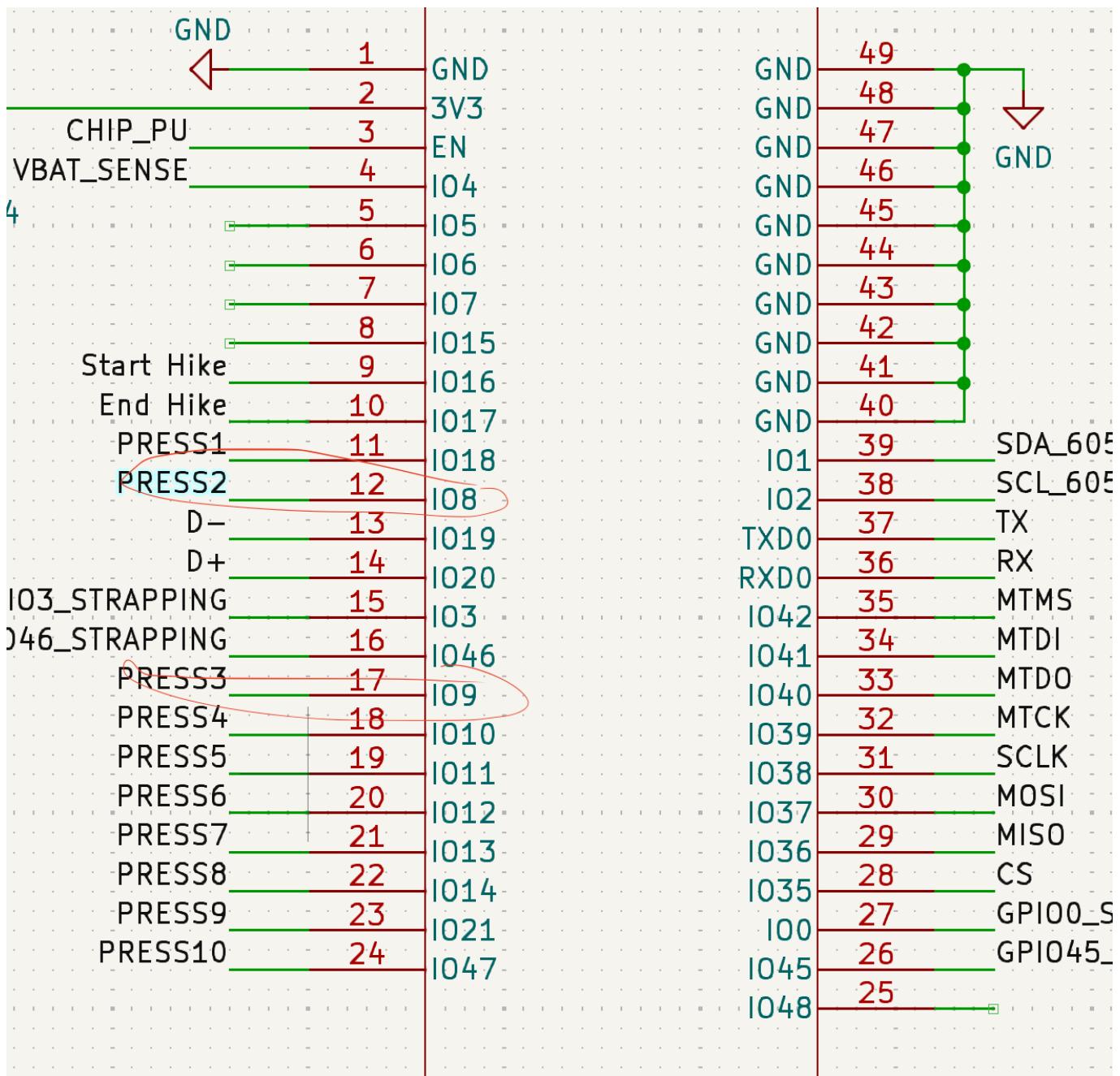
✓ **Error: Rear solder mask aperture bridges items with different nets**
Pad [<no net>] of JP2 on B.Mask

Show: All Errors (13) Warnings (1) Exclusions Save...

Delete Marker Delete All Markers Close Run DRC

 aperture





Notebook Entry: April 10 2024

Objectives:

- work on finding parts to order for final PCB
- take part numbers on Kicad and find parts from verified vendors

Work Session Record:

This work session I worked with Alyssa to match parts found on our schematic in Kicad with parts I could find on DigiKey, Mouser, Amazon, and the ECE supply shop. This process gave me a chance to introspect on the needs we had and give our connections a last once over in terms of making sure things were connected to the right ESP 32 pins. Additionally this process was needed as some vendors were out of some of the parts and we had to find alternatives from other vendors. This wasn't too hard as most parts were available on other sites.

Figures:

Notebook Entry: April 13 2024

Objectives:

- work on micro SD card read
- solder pin headers
- find pin connections for SPI

Work Session Record:

I wanted to prototype the file read/write/edit operations on the SD card with the ESP 32 S3 dev board. The first step was to get the SD module in a state which would allow us to read. I chose the SD card module as it used the least components between the pins and the pinouts. To get the board working I had to solder on a pin header to attach wires to and from the ESP 32 dev board. I found a few tutorials online on how to hook up the SD card module to the ESP32. However it was a different version of the chip/board and the pinouts had different protocols. This forced me to try and remap the pins to the different ones. After browsing I found a reddit post with the following pinout "MOSI (FSPID) aligns to GPIO11, MISO (FSPIQ) aligns with GPIO13, and SCK (FSPIKLER) aligns with GPIO12". This all online code I tried ended up working with the Arduino DS card and SPI modules.

Figures:

Sources:

- https://www.reddit.com/r/esp32/comments/w2e2gl/help_regarding_the_esp32s3_pins_for_spi/
- <https://forum.arduino.cc/t/esp32-s2-sd-card-interfacing/969091/7>
- <https://www.programmingelectronics.com/esp-32-sd-card-test-and-hardware-setup/#:~:text=The%20micro%20SD%20card%20reader,can%20talk%20to%20each%20other.>
- <https://www.lucadentella.it/en/2021/01/13/esp32-arduino-sd-card-con-pin-custom/>

Notebook Entry: April 14 2024

Objectives:

- work to adjust code to work with non-default pins

Work Session Record:

Our PCB design on Kicad mapped the pins to different places than the pins I had used yesterday. As such I physically changed the pin to the pin out found on the Kicad. This caused the operations to break and I had to find a custom way to assign pins for SPI communication. This caused a lot of frustration as no matter what I tried I was not able to get it to work.

Figures:

Notebook Entry: April 16 2024

Objectives:

- finalize part order
- order components

Work Session Record:

I worked this morning on going through the part order process. This was a long process that took a while to add everything to the cart. I also had to order from each vendor separately and take into account if things were in stock or not as the lead time would not allow for us to get these parts in time. Additionally, I had to compile a list of parts needed from the electronics services shop.

Figures:

n	Order Number	Requested By	Request Routed	Vendor	Items	Total Amount	Date Requested	Status	Action																																																												
1	172110	ramseyv2	Electrical and Computer Engineering	Digi-Key Corporation	<table border="1"> <thead> <tr> <th>Catalog/Part #</th> <th>Description</th> <th>Qty</th> <th>Recv'd</th> <th>Price</th> <th>Ext Price</th> </tr> </thead> <tbody> <tr> <td>SS8050-G</td> <td>BJT NPN Transistor</td> <td>8</td> <td>8</td> <td>\$0.27</td> <td>\$2.16</td> </tr> </tbody> </table>	Catalog/Part #	Description	Qty	Recv'd	Price	Ext Price	SS8050-G	BJT NPN Transistor	8	8	\$0.27	\$2.16	\$2.16	4/17/2024 10:19:25 AM	Picked up	[View][Clone]																																																
Catalog/Part #	Description	Qty	Recv'd	Price	Ext Price																																																																
SS8050-G	BJT NPN Transistor	8	8	\$0.27	\$2.16																																																																
2	172039	ramseyv2	Electrical and Computer Engineering	Digi-Key Corporation	<table border="1"> <thead> <tr> <th>Catalog/Part #</th> <th>Description</th> <th>Qty</th> <th>Recv'd</th> <th>Price</th> <th>Ext Price</th> </tr> </thead> <tbody> <tr> <td>1528-1857-ND</td> <td>Battery 3.7v 200mAh</td> <td>1</td> <td>1</td> <td>\$12.50</td> <td>\$12.50</td> </tr> <tr> <td>B2B-PH-K-S</td> <td>JST connector</td> <td>4</td> <td>4</td> <td>\$0.17</td> <td>\$0.68</td> </tr> <tr> <td>TLV75801PDRVT</td> <td>LDO</td> <td>2</td> <td>2</td> <td>\$1.01</td> <td>\$2.02</td> </tr> <tr> <td>1597-106990290-ND</td> <td>Battery Charger</td> <td>1</td> <td>1</td> <td>\$4.75</td> <td>\$4.75</td> </tr> <tr> <td>SP0503BAHTG</td> <td>Diode array</td> <td>4</td> <td>4</td> <td>\$0.78</td> <td>\$3.12</td> </tr> <tr> <td>TLV75801PDRVR</td> <td>LDO</td> <td>12</td> <td>12</td> <td>\$0.29</td> <td>\$3.49</td> </tr> <tr> <td>B35-1000</td> <td>Button</td> <td>12</td> <td>12</td> <td>\$0.65</td> <td>\$7.76</td> </tr> <tr> <td>MEM2052-00-195-00-A</td> <td>SD card adapter</td> <td>4</td> <td>4</td> <td>\$1.09</td> <td>\$4.36</td> </tr> <tr> <td>MPU-6050</td> <td>MPU</td> <td>2</td> <td>2</td> <td>\$7.85</td> <td>\$15.70</td> </tr> </tbody> </table>	Catalog/Part #	Description	Qty	Recv'd	Price	Ext Price	1528-1857-ND	Battery 3.7v 200mAh	1	1	\$12.50	\$12.50	B2B-PH-K-S	JST connector	4	4	\$0.17	\$0.68	TLV75801PDRVT	LDO	2	2	\$1.01	\$2.02	1597-106990290-ND	Battery Charger	1	1	\$4.75	\$4.75	SP0503BAHTG	Diode array	4	4	\$0.78	\$3.12	TLV75801PDRVR	LDO	12	12	\$0.29	\$3.49	B35-1000	Button	12	12	\$0.65	\$7.76	MEM2052-00-195-00-A	SD card adapter	4	4	\$1.09	\$4.36	MPU-6050	MPU	2	2	\$7.85	\$15.70	\$54.39	4/16/2024 6:20:51 PM	Picked up	[View][Clone]
Catalog/Part #	Description	Qty	Recv'd	Price	Ext Price																																																																
1528-1857-ND	Battery 3.7v 200mAh	1	1	\$12.50	\$12.50																																																																
B2B-PH-K-S	JST connector	4	4	\$0.17	\$0.68																																																																
TLV75801PDRVT	LDO	2	2	\$1.01	\$2.02																																																																
1597-106990290-ND	Battery Charger	1	1	\$4.75	\$4.75																																																																
SP0503BAHTG	Diode array	4	4	\$0.78	\$3.12																																																																
TLV75801PDRVR	LDO	12	12	\$0.29	\$3.49																																																																
B35-1000	Button	12	12	\$0.65	\$7.76																																																																
MEM2052-00-195-00-A	SD card adapter	4	4	\$1.09	\$4.36																																																																
MPU-6050	MPU	2	2	\$7.85	\$15.70																																																																
3	172067	ramseyv2	Electrical and Computer Engineering	Mouser Electronics	<table border="1"> <thead> <tr> <th>Catalog/Part #</th> <th>Description</th> <th>Qty</th> <th>Recv'd</th> <th>Price</th> <th>Ext Price</th> </tr> </thead> <tbody> <tr> <td>RS73G2ARTTD2003F</td> <td>200k ohm 0805 SMD</td> <td>20</td> <td>20</td> <td>\$0.43</td> <td>\$8.64</td> </tr> <tr> <td>SS8050-G</td> <td>Item B/O til July remove from order per Ramsey BJT NPN Transistor</td> <td>0</td> <td></td> <td>\$0.00</td> <td>\$0.00</td> </tr> <tr> <td>560112120003</td> <td>47 ohm 0805 SMD</td> <td>50</td> <td>50</td> <td>\$0.08</td> <td>\$3.95</td> </tr> </tbody> </table>	Catalog/Part #	Description	Qty	Recv'd	Price	Ext Price	RS73G2ARTTD2003F	200k ohm 0805 SMD	20	20	\$0.43	\$8.64	SS8050-G	Item B/O til July remove from order per Ramsey BJT NPN Transistor	0		\$0.00	\$0.00	560112120003	47 ohm 0805 SMD	50	50	\$0.08	\$3.95	\$12.59	4/16/2024 3:27:59 PM	Picked up	[View][Clone]																																				
Catalog/Part #	Description	Qty	Recv'd	Price	Ext Price																																																																
RS73G2ARTTD2003F	200k ohm 0805 SMD	20	20	\$0.43	\$8.64																																																																
SS8050-G	Item B/O til July remove from order per Ramsey BJT NPN Transistor	0		\$0.00	\$0.00																																																																
560112120003	47 ohm 0805 SMD	50	50	\$0.08	\$3.95																																																																

Notebook Entry: April 17 2024

Objectives:

- work to correct part orders

Work Session Record:

The NPN transistors I had ordered on Mouser sadly went out of stock overnight and now had a lead time of ~2 months. This was unacceptable as we wouldn't get our parts on time and would not be able to build our PCB. As such we canceled this order and made a new order on digi Key which did have the part in stock. I kept in constant communication with the purchasing office to make sure the parts were able to arrive on time.

Figures:

Order Status Tracker						
	Request		Purchase		Delivery	
	Requested	Approved	Processed	Purchased	Received	Picked-Up
Status	Completed	Pending				
Cancelled	4/16/2024 3:28:16 PM Van Der Meer, Ramsey Hendrik		Cancelled Request			
Date	4/16/2024					
Time	1:36:56 PM					
Name	Van Der Meer, Ramsey Hendrik					

Notebook Entry: April 18 2024

Objectives:

- Demo
- print enclosure

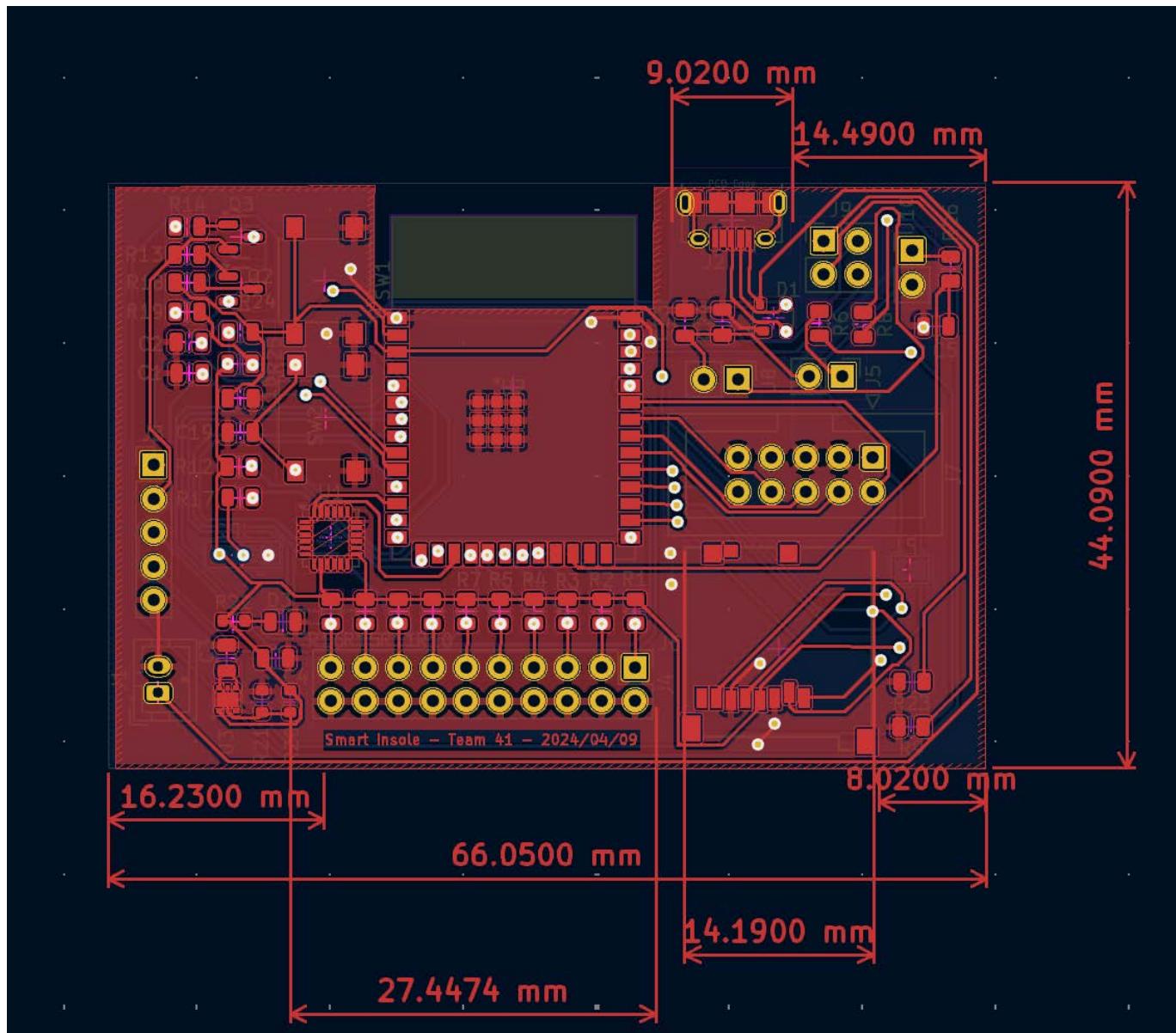
Work Session Record:

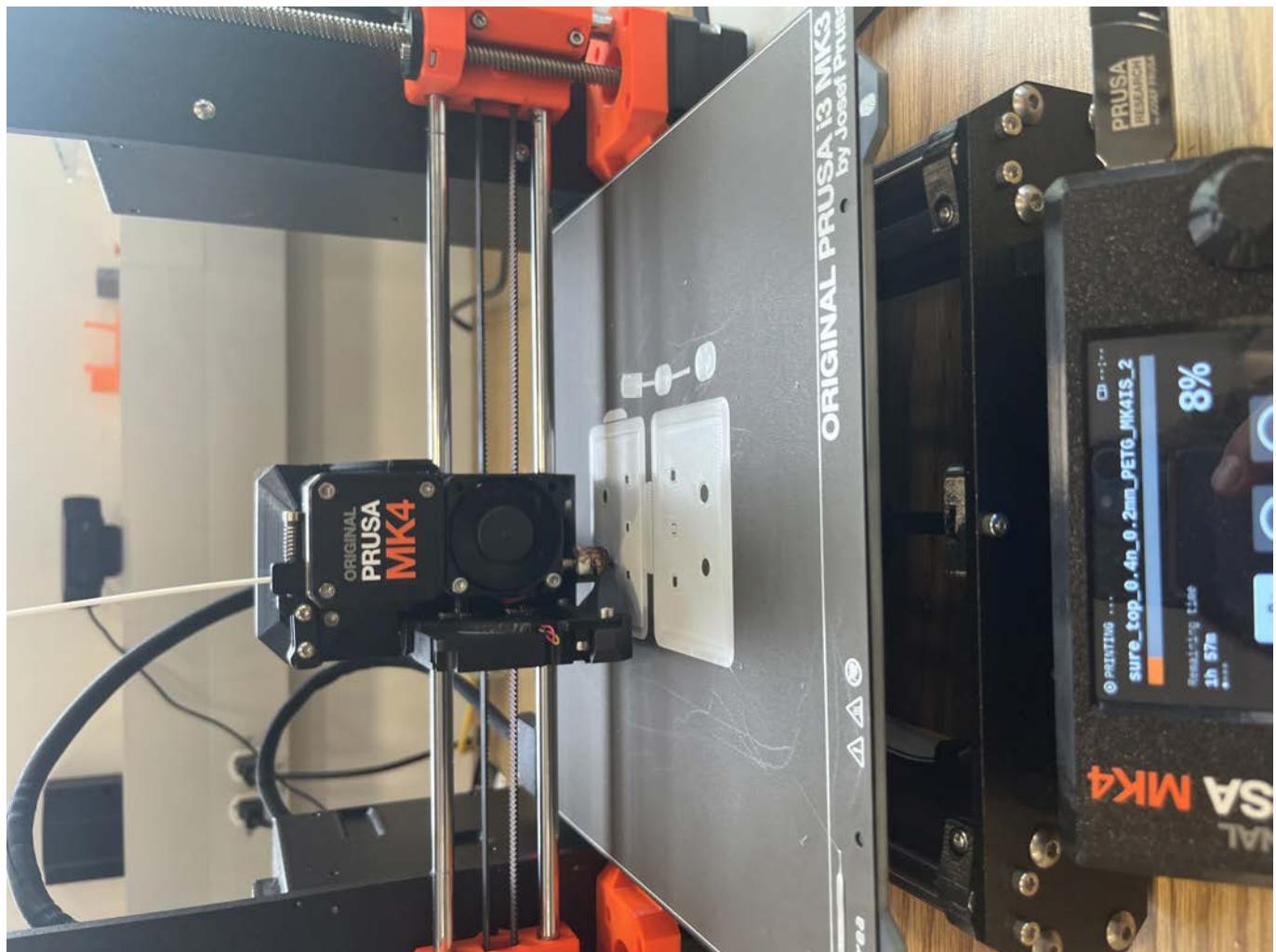
This day we demoed our bread boarded smart insole device to our TA and got feedback on what we should prioritize and what our estimated point total would be. This was a lot lower than we wanted and caused us to redouble our efforts towards creating a working PCB. I also worked to create a printed enclosure for our PCB which could be mounted with velcro or with a clip to the inside of the shoe. I worked to design the part in Fusion 360 where I was somewhat comfortable with 3D modeling. I used the dimensions on Kicad to create an enclosure that would fit our PCB. This was an important process as one of our high level requirements was to make a modular design which would not impede a hiker. I tried to design this enclosure to be as compact as possible while still maintaining functionality. This required holes for LEDs, buttons, and wires to get through. Additionally I consulted the relevant footprint datasheets for the dimensions of the components which would allow us to fit in the pcb and the components.

Figures:

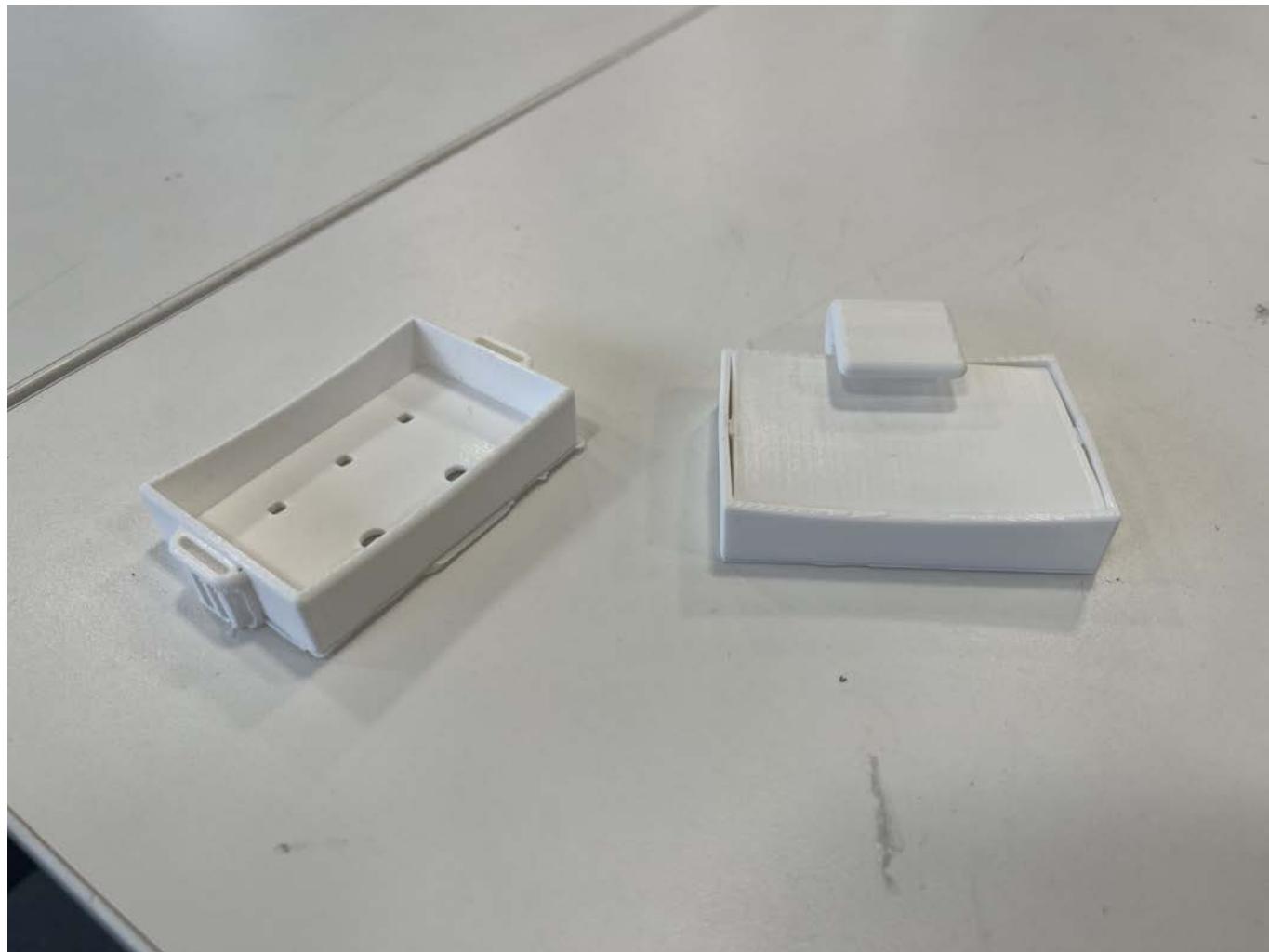












Sources:

- <https://www.sparkfun.com/datasheets/Components/LED/COM-09590-YSL-R531R3D-D2.pdf>
- https://omronfs.omron.com/en_US/ecb/products/pdf/en-b3s.pdf

Notebook Entry: April 19 2024

Objectives:

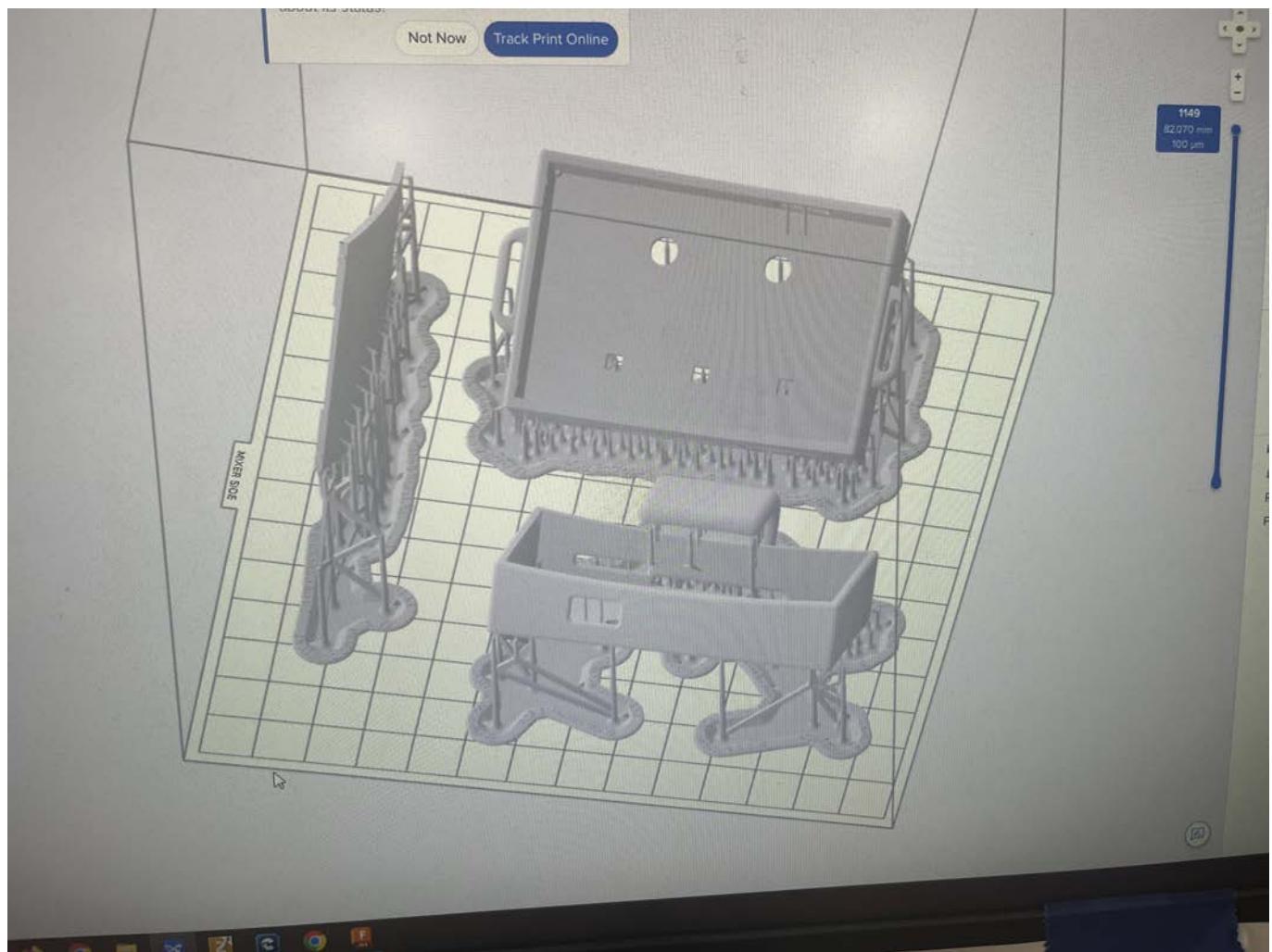
- make adjustments to tolerances
- add holes for SD card and Micro USB
- print updated enclosure

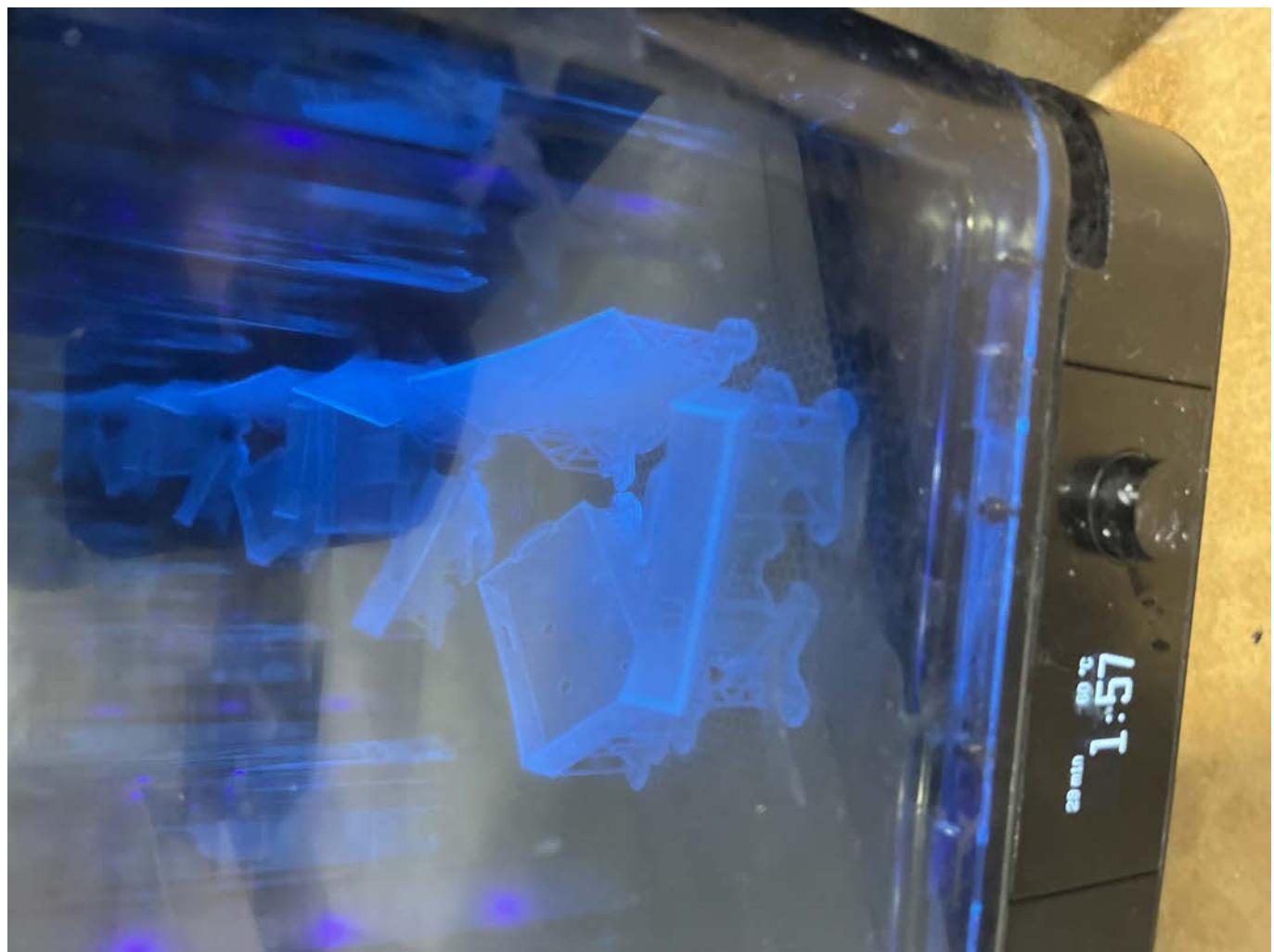
Work Session Record:

Worked with fusion360 to again remake the 3D printed enclosure which would fit on a user's shoe. We realized once inserting the PCB into the enclosure that the IO would not be able to be accessed. This would not be ideal as users would need to be able to access the SD card and be able to charge our device. As such I worked to add these in again using the kicad measurements. Additionally I consulted the relevant footprint datasheets for the dimensions of the components which would allow us to fit in the pcb and the components. Additionally the PCB barely fit into the enclosure and as such I scaled up the model by factor 1.01 to account for the expansion and tolerances which may occur when printing with FDM. These I thought would be a final print so I tried using resin as it would be much higher quality and allow for more accurate parts. This mentality and thought process however was a hindrance as it forced me to wait a while for the

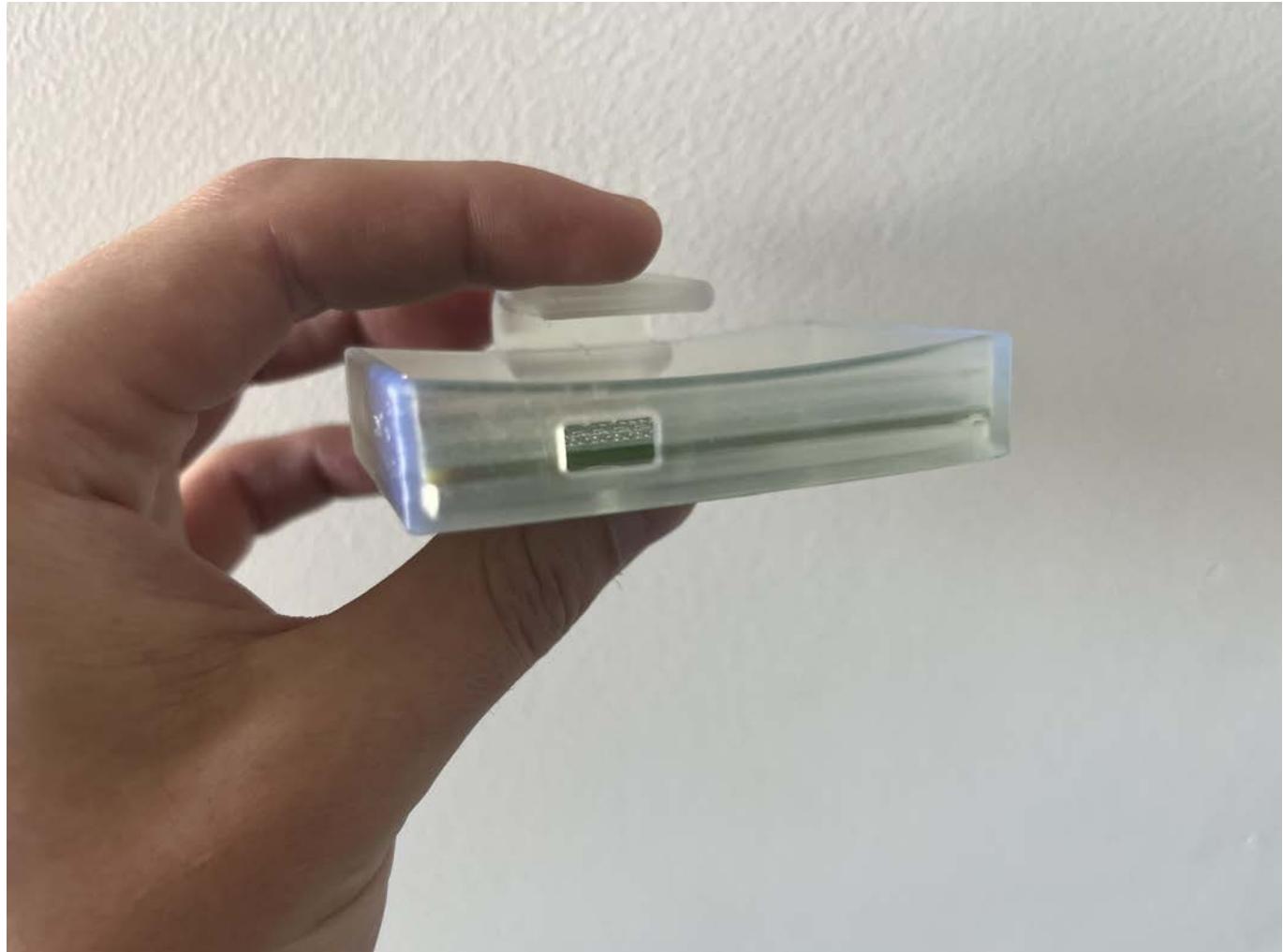
resin printers to print. Effectively 3x my print time on parts. I also had to use labs which were only open from 4-10pm and 5-11pm. This truly limited the time I'd be allowed to work and as such slowed down development. The resin printing process required a custom slicer for each lab. After the print I'd have to take care not to get resin on me as it is toxic. Additionally I would have to do a 15 minute wash cycle preceded by a 20 minute cure cycle so the resin can harden. However there was an issue with the resin printer as there was a massive hole in one of the enclosures.

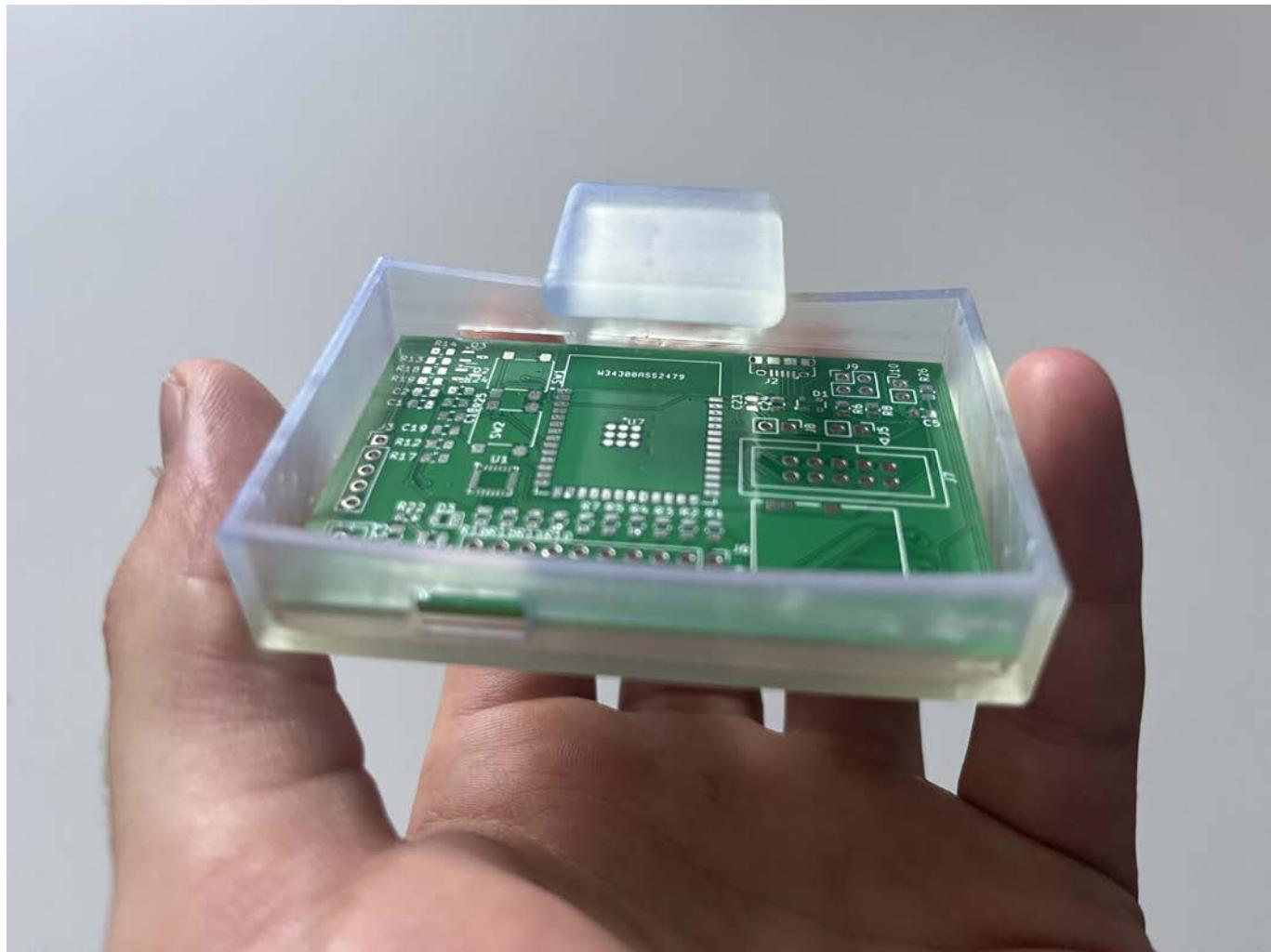
Figures:











Sources:

- https://au.mouser.com/datasheet/2/448/Yamaichi_Electronics_08162018_PJS008U-3000-0_RevD-1391577.pdf
- https://cdn.amphenol-cs.com/media/wysiwyg/files/documentation/datasheet/inputoutput/io_usb_micro.pdf

Notebook Entry: April 22 2024

Objectives:

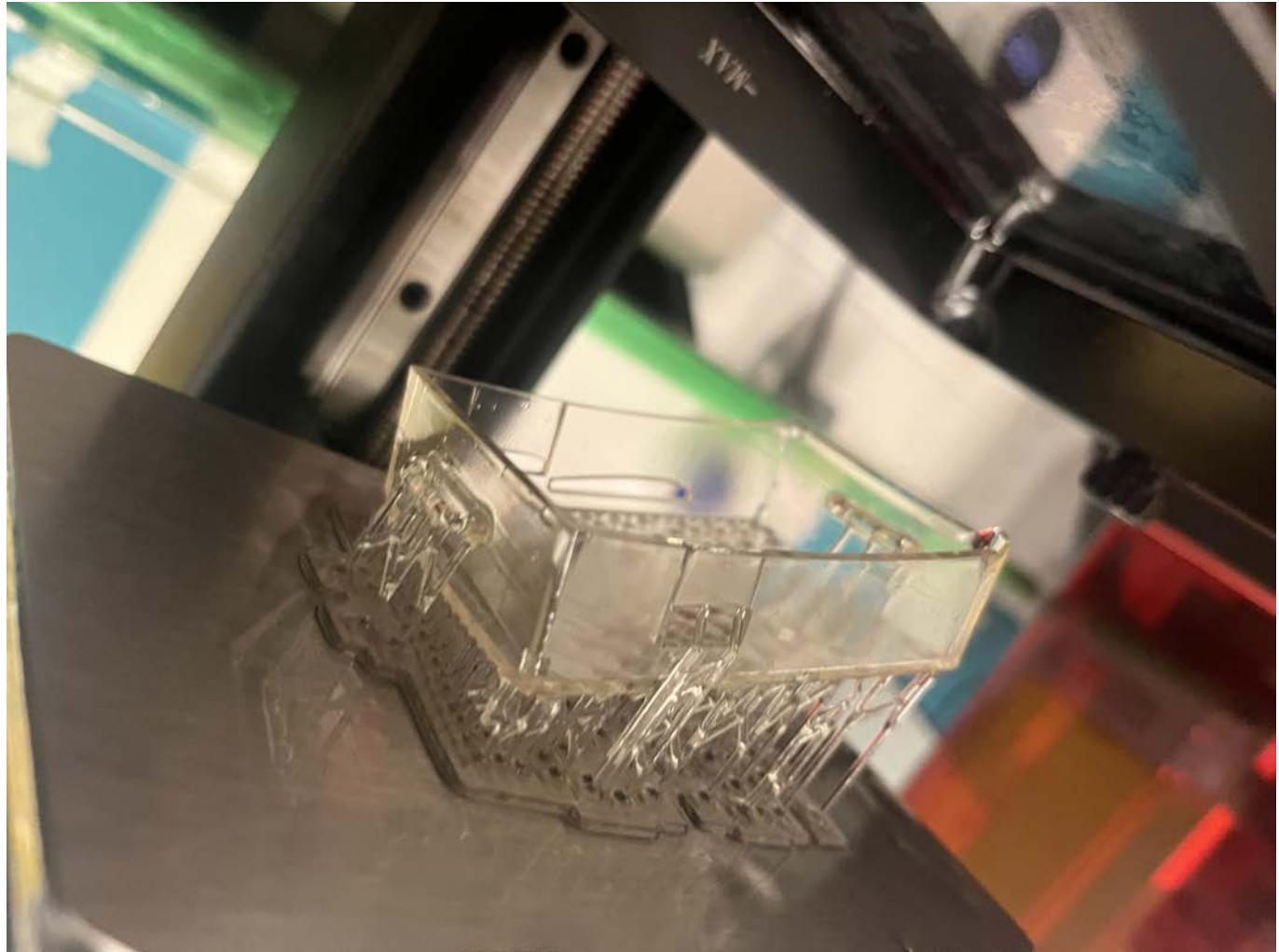
- make adjustments to cad since PCB has thickness

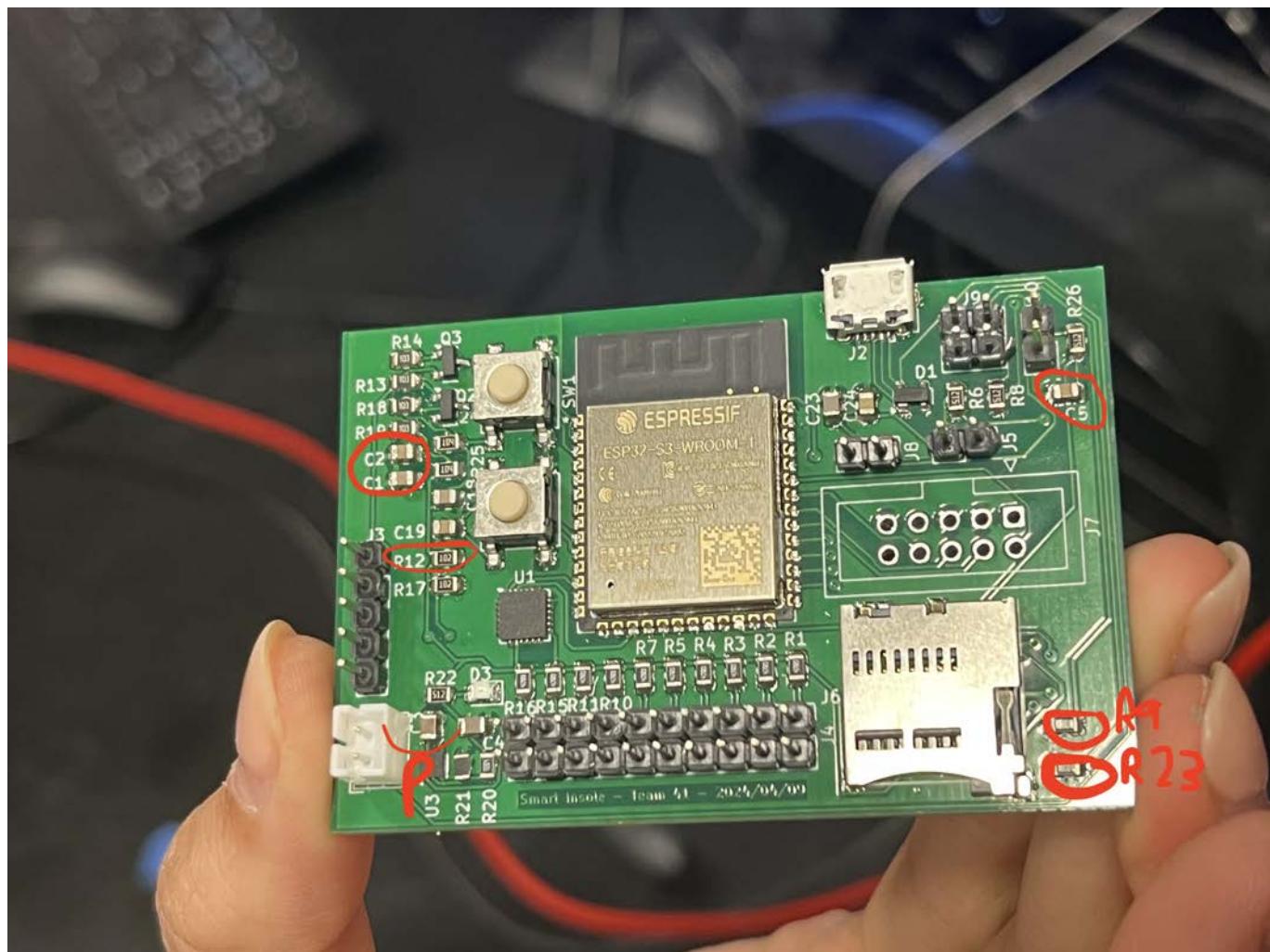
Work Session Record:

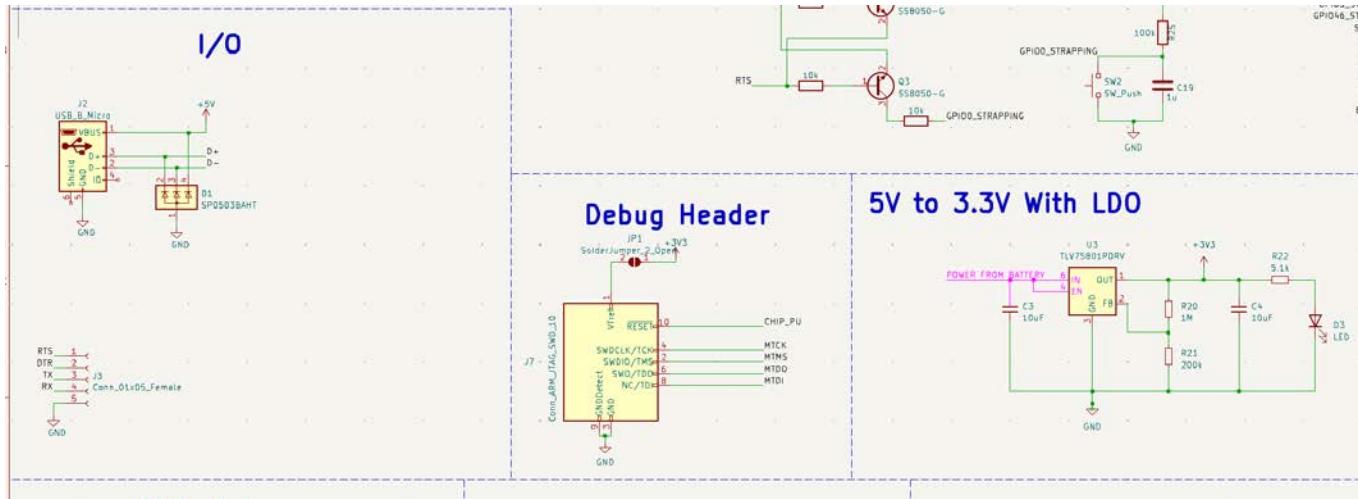
Worked to adjust the 3D fusion model to account for the PCB and its ~2 millimeters of physical thickness. This I forgot about when designing the holes through which the SD card and the micro USB would fit though when placed in the housing. As such I moved all holes on the side up by 2 millimeters. I then reprinted on a resin printer and went through the whole curing and wash cycle process again. As shown in the figure the print turned out well. I then brought this to the lab where Alyssa and I worked on SMD soldering parts to our PCB. This process was long as we not only had to find each component and place it on really small solder pads but bake it in the oven as when we tried the heat gun it was a little inaccurate and too hard to use precisely. I could not imaging using anything smaller than a 805 component as even these were extremely tiny and hard to work with. Overall we got the PCB soldered. We then moved on to try programming it.

However we were unable to get it to show up on any COM port on our laptop. We then examined our connections and realized that our board was not getting power. This we realized was since USB did not feed into our LDO which supplies the regulated 3.3v power to our ESP32 board. As such we were forced to solder on a wire connecting the USB 5v power (which also joins with the 3.7-3.3v Battery) to the input on our LDO. This is depicted in the component_markup figure at label P. where we connected these 2 pads. Next we tried connecting over COM with the ESP32 and it showed up. However we were unable to program it and got an error message. We thought it wasn't in the right mode and were proved correct when examining the strippain pin out requirements of the ESP32. We realized that we were supposed to leave IO46 floating and to do so not populate a resistor. As such we unsoldered resistor R9 shown in component_markup. After this we were able to program it successfully. However in order to get serial data to show up we used ESP32S3 BOX instead of ESP32S2 Dev Board to program our board. We don't know why this was the case but we rolled with it. We then tested SD card operations and it didn't work. We tested pressure sensing and it worked. We tested gyro functionality and it worked. After this we called it a night.

Figures:







Strapping Pin #	Function	ESP32-S3 Module Default	Example Board Implementation
0	A value of 0 will put the chip into "Download Mode" which enables the flashing of the program memory. A value of 1 will boot from the SPI memory IC inside the module that contains the program memory.	Pulled up internally	Operate with button or programmer circuit
3	Switches the source for JTAG signals	Floating	Set with solder jumper but can be left alone.
45	Sets the voltage for the internal SPI memory IC	Pulled up or down internally	Pin is left unconnected to prevent the SPI memory voltage from changing.
46	Sets the boot mode and where the ESP32-S3 ROM sends messages.	Pulled down internally	A pull-up resistor is in the schematic and labelled as DNP to ensure the functionality of this pin is made available to the designer. In order to program normally, do not populate the pull-up resistor on the board. This will allow the internal pull-down to function and set the pin to 0.

sources:

- <https://www.pololu.com/blog/403/new-product-breakout-board-for-microsd-card>
- <https://gct.co/files/drawings/mem2052.pdf>
- https://courses.engr.illinois.edu/ece445/wiki/#esp32_example/index

Notebook Entry: April 22 2024

Objectives:

- DAY
- make adjustments to cad since PCB now is a bit bigger due to connecting wire
- identify why SD card operations aren't working
- NIGHT
- work on PCB
- make esp 32 chip work
- debug circuit
- check connections

- program esp 32
- test all components
- test sd card
- test IMU
- test pressure

Work Session Record:

Since we now routed a wire around the PCB we had to reprint the model a bit bigger. I also adjusted it so that there would be space for a battery in the top of the enclosure. Additionally I moved down the wire hole so it didn't overlap with the PCB. I also redesigned the closing system to have a little less precision. I then started a resin print in siebel. After we met up in the lab to try and debug our PCB.

Figures:



component_markup

Resources:

- <https://electronics.stackexchange.com/questions/278852/sd-card-detect-in-spi-mode>
- <https://forum.arduino.cc/t/how-can-i-connect-an-sd-card-to-arduino-directly/902446/15>
- <https://goughlui.com/2014/01/03/project-read-collect-decode-sd-card-csd-register-data/>
- https://www.reddit.com/r/arduino/comments/15w4ctf/how_to_connect_an_sd_card_to_esp32_s3/

Notebook Entry: April 23 2024

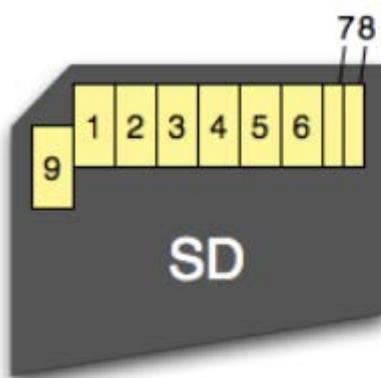
Objectives:

- fix sd
- fix IMU (help alyssa)
- add buttons
- fix capacitors causing long discharge time of capacitor
- add LED
- create presentation

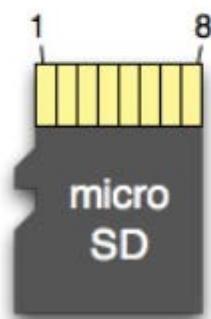
Work Session Record:

Found out that SD card pins weren't mapped right which caused the problem where it wouldn't mount. This was caused as we mapped the pins for SPI protocol according to the pin mappings for an SD card. This should have been easy to fix as we could have changed these pin mappings in software. However since one of the main pins was connected to ground it would have been a lot more challenging. One approach we tried was to splice off a section of the ground plane connected to this ground pin and solder a wire from this large pad to the right place. However this was a little too delicate of a process for us as there were a bunch of wires all around this plane which caused us to have to work a lot more carefully. As such we gave up on pursuing this endeavor. If we were to do another PCB order we would definitely be able to have resolved this connection error on our PCB. I also helped Alyssa to try and fix the IMU by soldering on floating components to connect to the pins which were needed to be powered. Ultimately this took too long and as such we had given up on this process as well. Something we did accomplish was the integration of the LED and the buttons which would show if bluetooth was working, if the sensors were recording and a combination. Some issues we had here was that there was a 100k ohm resistor in series with our LED which caused it to not turn on. We then soldered on a 1k ohm which allowed it to turn off but it was super dim. One change we didn't get to was to convert this 1k ohm to a 47k ohm allowing the LED to be a lot brighter. In terms of the button we had an issue where the button when pressed would discharge and then on the charging a capacitor resistor combination would have a super high RC circuit charge time. This would cause our software to think the button was constantly pressed. We tested this by console logging the values read by our analog to digital pin reading the voltage of the button press. As such we replaced these components to have a much shorter RC. Finally in this work session we created our final demo presentation slides which helped to guide our audience during our demo.

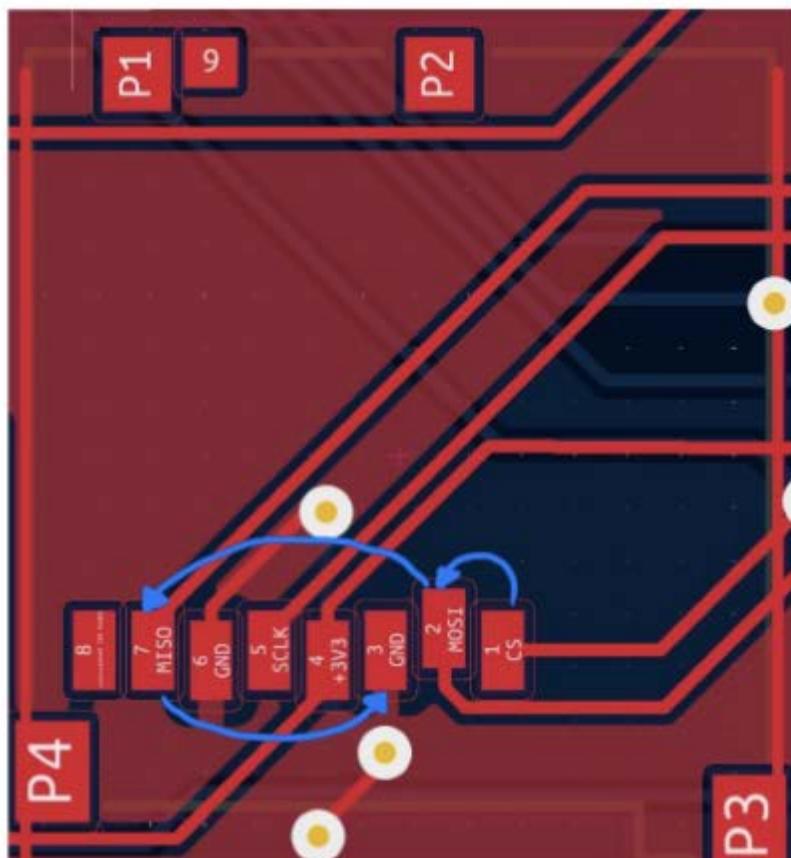
Figures:



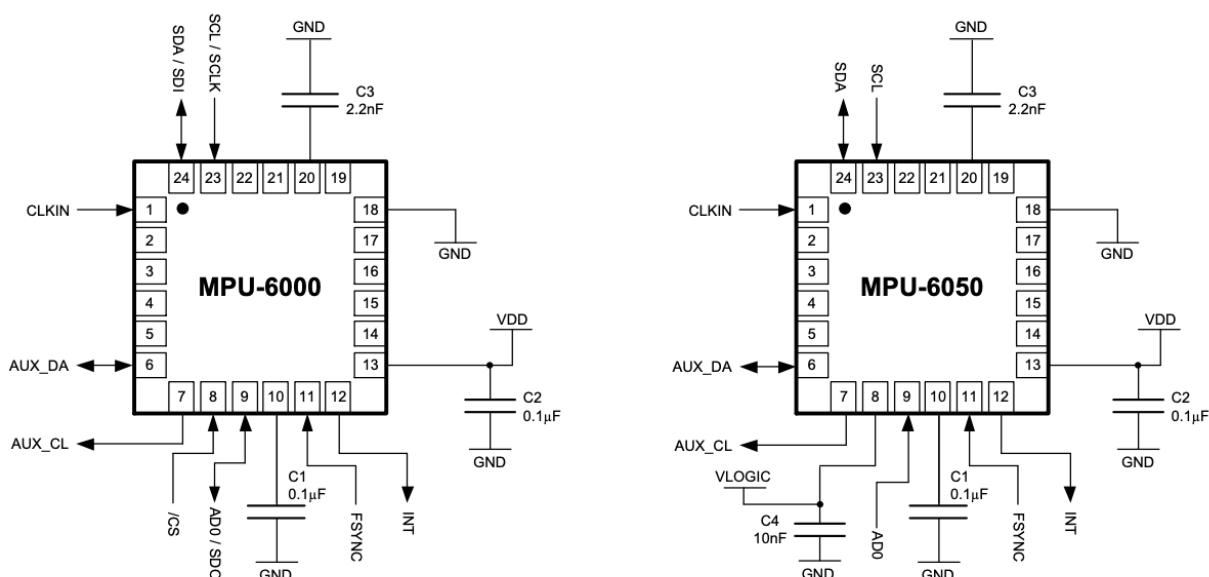
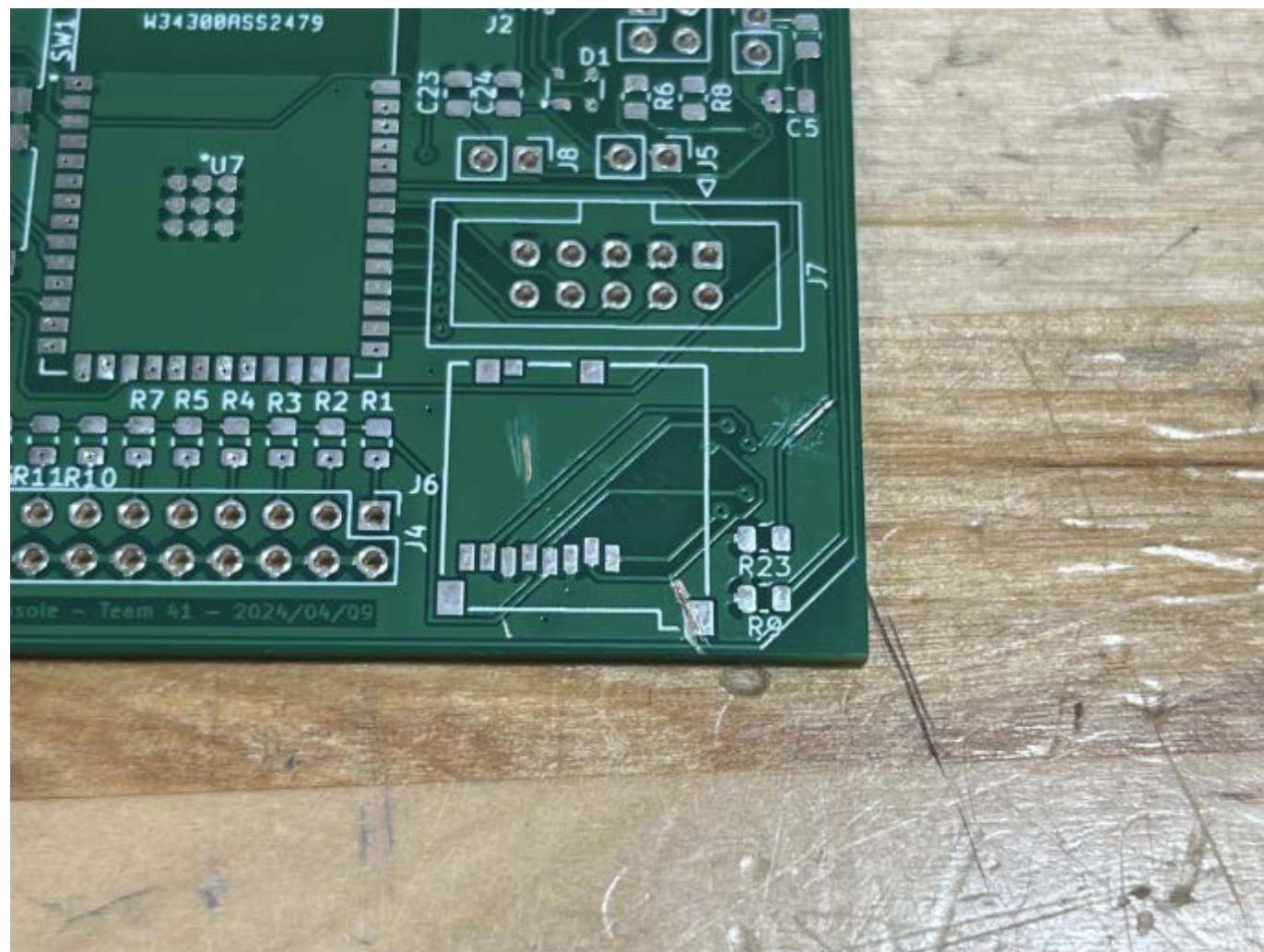
Pin	SD	SPI
1	CD/DAT3	CS
2	CMD	DI
3	VSS1	VSS1
4	VDD	VDD
5	CLK	SCLK
6	VSS2	VSS2
7	DAT0	DO
8	DAT1	X
9	DAT2	X



Pin	SD	SPI
1	DAT2	X
2	CD/DAT3	CS
3	CMD	DI
4	VDD	VDD
5	CLK	SCLK
6	VSS	VSS
7	DAT0	DO
8	DAT1	X







Typical Operating Circuits

Resources:

- <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>

Notebook Entry: April 24 2024

Objectives:

- create presentation demo
- polish up long discharge of button
- move PCB to casing

Work Session Record:

We got the discharge figured out but it was still a bit finicky. As such we added a delay in software by about 5 seconds where we wouldn't read if the button was pressed so that we wouldn't get any false readings. We also worked to polish up the demo presentation as I attended a peer review demo where the group was asked a bunch of hard questions like temperature, power delivery, and to show the output of their power circuit. As such we had to modify our presentation to include these calculations and readings. However when we did go into the demo we never got asked this. I guess it was asked to try and allow the group that I watched to earn back points. We then did our demo in front of 2 A's and 1 professor. This was fun to showcase what we had been working really hard on all semester. Overall I think it went really well as our demo went without hiccup except for our insole being a little bent causing the insole to have 1 sensor be perpetually high.

equations

Calculation for max ambient temperature our device can be in

- $TA(MAX) = 125^{\circ}\text{C} - 80.3^{\circ}\text{C}/\text{W} \times (5 \text{ V} - 3.3 \text{ V}) \times (0.25 \text{ A}) = 90.87^{\circ}\text{C}$
- $TA(MAX) = 125^{\circ}\text{C} - 80.3^{\circ}\text{C}/\text{W} \times (5 \text{ V} - 3.3 \text{ V}) \times (0.5 \text{ A}) = 56.74^{\circ}\text{C}$
- $TA(MAX) = 125^{\circ}\text{C} - 80.3^{\circ}\text{C}/\text{W} \times (5 \text{ V} - 3.3 \text{ V}) \times (0.6 \text{ A}) = 43.09^{\circ}\text{C}$

Calculation for max time on battery power with 2000 mAh battery as Gyro + SD not measured with PCB but with breadboard

- Gyro: 1.73v over 1k ohm - .0017 A
- SD : 0.159v over 1k ohm - 0.000159 A
- **PCB Power Draw - 400mA → 2000mAh / 400mA = ~5 hours**

Figures:

Resources:

- https://courses.engr.illinois.edu/ece445/wiki/#esp32_example/index

Notebook Entry: April 25 2024

Objectives:

- create mock demo presentation

Work Session Record:

We wanted to get feedback on a presentation for our final as we wanted to be as prepared as possible. As such we created a mock presentation where we showcased what we had done in a susince presentation. Slides we included were Overall, Recommendations for further work, What We Learned, SD Card Read PCB

Error, MPU PCB Error, PCB Capabilities/Errors, Ethics, Max Time On Battery Power, Max Ambient Temperature, Wearable/Modular Design, Web Interface, Insole design, Block Diagram, High Level Requirements, and Background.

Figures:

Resources:

Notebook Entry: April 26 2024

Objectives:

- Do mock presentation
- Get feedback on mock presentation

Work Session Record:

We went through our mock presentation where we showcased what we had done in presentation form. Some feedback we got was that we needed a demo, to focus on positioning, reduce wordiness of slides, and focus on removing filler words.

Figures:

Date: 04.20 Course:

Team Name: 41



Just In Time Workshop Feedback Form

Great job on your presentation! The following feedback is intended to help you polish your presentation. Please consider filling out a feedback survey on your experience. Scan the QR code up above!

Slide Design	Collaboration and Cohesion	Delivery
<p>use bullet points instead of full sentences.</p> <ul style="list-style-type: none"> * Reduce the amount of text on your slides high lvl Re-order your slides to tell a clear story Change the font style and/or size to be legible and consistent * Consider using UIUC approved slides * Add slide numbers and/or an agenda slide <p>Additional notes:</p> <ul style="list-style-type: none"> block diagram is a good size! you can read it. consider circling or highlighting some of the hard to see picc. I like the break down of website but hard to read up grad pic. - consider two slides check spelling: power and "teperature" & "SD" vs "SD" in could read slide myu pcb error is a great slide too. 	<p>req. emails</p> <ul style="list-style-type: none"> * Add verbal transitions between speakers (e.g., "Now, I'll pass it to Anna.") * Physically position yourselves to stand on the sides of the screen in an even and balanced way Consider actively listening to your fellow teammates while they speak to look engaged Practice with your teammates to find a common rate and pitch of presenting <p>Additional notes:</p>	<p>* Stand strong with your feet shoulder width apart, your spine straight, and your shoulders back</p> <p>* Identify and use one resting hand gesture that is not crossing your arms or putting your hands in your pockets, play w/ clothes jewelry</p> <p>Make eye contact with the audience at least 50-70% of your speaking time</p> <p>Practice without your notes so you can adopt an extemporaneous style of speaking</p> <p>Additional notes:</p> <ul style="list-style-type: none"> speak to the back of the room. vocal filter: "m" "uh" great job explaining and coming back to audience. I like the addition of the ethic slide.

Resources:

Notebook Entry: April 28 2024

Objectives:

- create final demo presentation

- create final paper
- add equations to final paper
- add images to final paper

Work Session Record:

Created the final demo presentation and made the following adjustments to our mock presentation:

- less words on ethic slide
- switched to ECE template like dictated in assignment details
- Added demo video slide
- split up real time and upload slides for web interface
- added design changes slides

Figures:

Resources: