Microbial Pill Sensor

DESIGN DOCUMENT

sdmay25-17
Dr. Meng Lu
Cade Kuennen
Rakesh Penmetsa
Wes Ryley
Alexander Upah
sdmay25-17@iastate.edu
sdmay25-17.sd.ece.iastate.edu
12/06/2024 v1.0

Executive Summary

Excessive nitrate concentration in bodies of water, often a result of agricultural runoff, poses significant environmental hazards, endangering both aquatic life and the lives of those consuming the water. Current methodologies for identifying nitrate concentration in agricultural runoff are intensive, expensive, and inefficient, often requiring manual sampling and off-site analysis. By providing more efficient, cost-effective, and autonomous nitrate detection, the microbial pill sensor aims to provide continuous monitoring of nitrate concentrations for both environmentalists and farmers. Improved access to environmental conditions allows users to make more informed decisions regarding the health of waterbodies and fertilization processes.

The microbial pill sensor uses a novel biosensing detection mechanism, relying on bioengineered microbes to identify the presence of nitrate in the environment. The microbes used by the microbial pill sensor have been genetically engineered to express **green-fluorescent protein** (GFP) due to the presence of nitrate. The concentration of nitrate in the environment can be determined through the fluorescent response of the expressed GFP, which is initiated, measured, and transmitted via the electronic components of the microbial pill sensor.

The microbial pill sensor is a cylindrical capsule with a radius of 15mm and a height of 47mm, broken into distinct modules. The bioengineered microbes are contained within a housing chamber, and the solution will flow in through a selective membrane located at the top of the capsule. The optical detection PCB and the microcontroller PCB are situated directly below the housing. A 488 nm wavelength LED is required to excite the expressed GFP, resulting in the emission of 532 nm wavelength light. The emitted light is captured by a photodetector, resulting in a photocurrent proportional to the presence of nitrate in the environment. A microcontroller is responsible for the activation of the LED at the desired interval and the transmission of the photocurrent value to an external device via low-power Bluetooth. An external GUI application processes the measurement and displays recorded nitrate concentrations for evaluation by the user.

The microbial pill sensor implementation currently consists of a functional photodetection PCB encompassed in the housing, along with optical emission and excitation filters, which is connected to an exterior ESP32-C3-Dev-Kit to enable MCU functionality. Complications in the design of the MCU PCB make the current MCU PCB design nonfunctional. The photodetection system is capable of a 52.02 mV / mW sensitivity with a ~100 μW limit of detection within the housing for 500 nm light. Poor selection of the excitation LED prohibits the device from properly exciting the necessary fluorescent response from GFP-expressing microbes. Bluetooth low-energy and Serial transmission of photogenerated voltage values have been demonstrated, with an external GUI application capable of receiving, displaying and saving the measured values to a CSV file for analysis by the user.

Future development of the microbial pill sensor will require proper development of the MCU PCB. Updates to the housing design will enable detection of fluorescent response by increasing the light intensity reaching the photodetection PCB through the implementation of lensing. Functional environment testing is required to characterize the lifetime, power consumption, transmission range, and environmental impact of the device.

Learning Summary

Development Standards & Practices Used

Circuit, Hardware, and Software Practices

Part decompositions of existing developments to meet requirements

PUN and PDN in circuit design

Use of meaningful commenting in code

Engineering Standards

IEEE 802 Nendica Report: Flexible Factory IoT: Use Cases and Communication Requirements for Wired and Wireless Bridged Network

IEEE Standard for User Interface Elements in Power Control of Electronic Devices Employed in Office/Consumer Environments: IEEE 1621-2004

IEEE Standard for a Real-Time Operating System (RTOS) for Small-Scale Embedded Systems: IEEE 2050-2018

IPC-221 Standards in PCB Design

Summary of Requirements

F.1 House specified bio-detection microbe in housing chamber and maintain the micorbes for the desired length of operation

F.2 Activate LED component at specified interval to disperse light uniformly in housing chamber

F.3 The dispersed wavelength of light emitted from the LED produces fluorescent emission via GFP protein expressed by bio-detection microbe upon the presence of the desired analyte in the chamber

F.4 Photodetection component produces photocurrent proportional to the concentration of analyte via fluorescent emission from microbe

F.5 Filter membrane allows the solution containing the desired analyte to flow into the detection chamber.

F.6 Recorded voltages are wirelessly transmitted to an external device for processing via Bluetooth Low-Energy connection or a wired Serial connection.

F.7 A graphical user interface will display recorded voltages which can be converted to analyte concentration on external device via processing of recorded voltage values

P.1 The microbial pill sensor should be no larger than $\mathbf{10\ x\ 10\ x\ 5\ } mm^3$

P.2 Properly contained housing chamber that can be removed to replace specific microbe

P.3 Properly contained compartment for holding batteries that can be replaced or recharged when needed.

E.1 The Housing will be composed of environmentally safe materials

E.2 The microbe housing container does not allow other bacteria to enter the chamber due to risk of conjugation

E.3 The microbe housing container does not allow microbe to exit the housing chamber due to risk of uncontrolled mutations

E.4 The battery housing component will prevent environmental contamination resulting from degradation of batteries during the product lifecycle

U.1 The external GUI will display voltage recordings.

U.2 The external GUI will record voltages over a specified interval as a function of time to monitor the time evolution of the system.

U.3 The external GUI will transfer recorded data into a CSV file in order to manage testing data.

Applicable Courses from Iowa State University Curriculum

- EE 2300: Electronic Circuits and Systems
- CPRE 2880: Embedded Systems 1: Introduction
- EE 2850: Problem Solving Methods and Tools for Electrical Engineering
- EE 3300: Integrated Electronics
- EE 3320: Semiconductor Materials and Devices
- EE 3330: Electronic Systems Design
- EE 4140: Microwave Engineering
- EE 4500: Biosensors

New Skills/Knowledge acquired that was not taught in courses

- 3D Cad Design
- 3D Printing
- Creating a functional Graphical User Interface with Python language and Libraries
- Programming MCU dev kit in Arduino IDE
- Understanding of genetic memory circuits in bioengineered bacteria

Table of Contents

List of Figures and Tables

Definitions

Bluetooth Low Energy (BLE):

A wireless personal area network technology providing considerably reduced power consumption and cost while maintaining a similar communication range as classic Bluetooth. It is independent of classic Bluetooth Basic protocol but uses the same 2.5 GHz radio frequencies.

Green-Fluorescent Protein (GFP):

A protein that exhibits green fluorescence when exposed to blue light. The specific version of GFP employed by microbial pill sensors has an excitation wavelength of 488 nm and an emission wavelength of 532 nm.

Universally Unique Identifier (UUID): 16 (short) or 128 (long) bit number that uniquely identifies an object for BLE services.

1. Introduction

1.1. Problem Statement

Biosensing is a rapidly developing industry where new technologies for the detection of pathogens, contaminants, and other analytes are constantly being developed and improved upon. Biosensing technologies optimally provide quick, accurate, automated, and noninvasive alternatives for detection compared to manual techniques. Biosensing technologies provide continuous and autonomous monitoring of environments, providing a path to improved response and action. Optical detection mechanisms, which make use of changes in the intensity or wavelength of emitted light, provide superior selectivity and sensitivity compared to alternative biosensing mechanisms, improving the ability to monitor and respond to the presence of analytes in the environment. Optical biosensors also provide cost and size benefits when compared to alternative biosensing mechanisms such as electrochemical, impendence measurement, or acoustic technologies.

Excessive nitrate concentrations in bodies of water damage aquatic life and generate unsafe consumption conditions. Additionally, excessive nitrate concentration in agricultural runoff is indicative of inefficient fertilization processes. Available methodologies for identifying the concentration of nitrate in waterways require expensive manual sampling and external laboratory processing, making assessment of environmental and fertilization outcomes expensive and inefficient. Our project aims to provide a novel miniaturized nitrate biosensing technology through a microbial pill sensor for the efficient, autonomous, and economically viable detection of nitrate in waterways. Microbes contained within the microbial pill sensor have been genetically engineered to express **green fluorescent protein** (GFP) under the transcription control of the target analyte nitrate. Through excitation by 488 nm wavelength light, GFP produces a fluorescent response at the 532 nm wavelength. Through measurement of the fluorescent response of the expressed GFP, the microbial pill sensor aims to enable quick, accurate, and autonomous determinations of environmental nitrate concentrations and provide an improved, cost-efficient, and environmentally friendly alternative to current nitrate detection technologies.

The microbial pill sensor requires an optical detection module to measure the fluorescent response and a microcontroller module to transmit recorded measurements. Transmission via **Bluetooth low energy** (BLE) reduces power consumption, enabling extended device lifetime for continuous environmental monitoring. An external GUI application provides processing and visualization of recorded voltage measurements for analysis by the user. While the microbial pill sensor is intended for application in nitrate sensing, the design aims to be configurable to different applications. The microbial pill sensor provides functionality for the detection of different target analytes, provided that bioengineered microbes exhibiting GFP in the presence of the target analyte exist. The development of genetic memory circuits for analyte detection is a rapidly developing field, providing a feasible route to the development of additional compatible microbes. The microbial pill sensor's low power consumption, miniaturized size, autonomous operation, and wireless data transmission make it a novel biosensing technology with clear application in environmental monitoring.

1.2. INTENDED USERS

To understand potential users of the microbial pill sensor, the team used empathy maps and personas. Through this exercise, the team generated the empathy map shown in Figure 1-1. From this empathy map the following user personas were created, each with distinct needs and potential benefits from the project's implementation:

- Bioengineering Researchers
- Bioengineering Students
- Environmentalists

The first group includes bioengineering researchers. These users typically work on developing bioengineered bacteria for various applications and require a low-energy system to monitor these bacteria. Traditional systems rely on Wi-Fi for data transmission, leading to high energy consumption, which is not suitable for small-scale, continuous monitoring tasks. Our product addresses this need by using a low-power Bluetooth transmission system, offering a more energy-efficient alternative. This approach directly aligns with the microbial pill sensor's problem statement, which focuses on creating a low-power solution for bioengineered bacteria monitoring.

Another important user group is bioengineering students, who often experiment with different types of bioengineered microbes. These students need a versatile sensor package to test their microbes and determine whether the microbes are functioning as intended. The microbial pill sensor aims to allow for flexible testing of various GFP-emitting microbes, eliminating the need for multiple sensor packages. This benefits the students by providing a cost-effective and adaptable solution for their experimentation. The design supports multiple biosensing microbes, connecting directly to our problem statement, which aims to create a flexible, low-power sensor package that can be used across a range of microbial sensing applications in addition to direct application in nitrate detection.

Additionally, environmentalists also stand to benefit from our product. They are responsible for detecting pollutants or contaminants in natural water bodies to ensure ecosystem health. Their need is for a sensor that can perform long-term environmental monitoring without requiring frequent maintenance or battery replacement. Our low-power design allows the sensor to remain operational for an extended period before the batteries need to be replaced. This feature enables continuous monitoring and aligns with the need for sustained, uninterrupted data collection in the environment, directly supporting the overarching goal of creating a sustainable, energy-efficient monitoring system. The microbial pill sensor's direct application in nitrate detection provides a viable alternative for application in improving the health of waterways, increasing environmentalists' access to valuable information.

Finally, the microbial pill sensor is advantageous for farmers using nitrogen fertilizer. Assessment of nitrate runoff from the fertilization process enables improved assessment of fertilization processes. Improvement in fertilization processes boosts environmental health both in and surrounding agricultural fields and reduces unnecessary costs to the farmer. Application by farmers requires the microbial pill sensor to be an economically competitive solution to nitrate detection, requiring a cost-aware design of the microbial pill sensor.

Figure 1-1-1: Empathy Map

2. Requirements, Constraints, And Standards

2.1. Requirements & Constraints

The following list of functional, physical, environmental, and user experience-related requirements has been created to ensure the development of the desired microbial pill sensor with complete specified functionality. As will be discussed in section *5. Testing* and section *6. Implementation* later in this report, the current implementation of the microbial pill sensor fails to satisfy this complete list of requirements. Further details will be presented regarding the fulfillment and progress towards the different requirements and future development steps. The idealized microbial pill sensor design requirements are still presented as further development of the microbial pill sensor will aim to meet the final design requirements presented here. Further development of the microbial pill sensor will provide the necessary device sensitivity and microbe sensitivity to generate quantitative accuracy requirements for the device.

Functional Requirements:

F.1 House specified bio-detection microbe in housing chamber and maintain the microbes for the desired length of operation.

F.2 Activate LED component at specified interval to disperse light uniformly in housing chamber

F.3 The dispersed wavelength of light emitted from the LED produces fluorescent emission via GFP protein expressed by bio-detection microbe upon the presence of the desired analyte in the chamber

F.4 Photodetection component produces photocurrent proportional to the concentration of analyte via fluorescent emission from microbe

F.5 Filter membrane allows the solution containing the desired analyte to flow into the detection chamber.

F.6 Recorded voltages are wirelessly transmitted to an external device for processing via Bluetooth Low-Energy connection or a wired Serial connection.

F.7 A graphical user interface will display recorded voltages which can be converted to analyte concentration on external device via processing of recorded voltage values

Physical Requirements:

P.1 The microbial pill sensor should be no larger than $10x10x5\ mm^3$ **(constraint)**

P.2 Properly contained housing chamber that can be removed to replace specific microbe.

P.3 Properly contained compartment for holding batteries that can be replaced or recharged when needed.

Environmental Requirements

E.1 The housing will be composed of environmentally safe materials.

E.2 The microbe housing container does not allow other bacteria to enter the chamber due to risk of conjugation.

E.3 The microbe housing container does not allow microbe to exit the housing chamber.

E.4 The battery housing component will prevent environmental contamination resulting from degradation of batteries during the product lifecycle

User Experience Requirements

U.1 The external GUI will display voltage recordings.

U.2 The external GUI will record voltages over a specified interval as a function of time to monitor the time evolution of the system.

U.3 The external GUI will transfer recorded data into a CSV file in order to manage data.

2.2. Engineering Standards

Standards play a crucial role in engineering. They ensure the consideration of safety, quality, and consistency throughout the development of new technologies. Engineers are given clear guidance on creating reliable products and systems that meet industry and regulatory expectations when following these established guidelines. Standards help improve efficiency, encourage global collaboration, and prevent accidents by ensuring engineering designs are understandable by all and implemented correctly. Most importantly, they protect public health, safety, and the environment, all while promoting innovative designs and processes. The following standards have been selected for our microbial pill sensor project due to their relevancy in creating an effective and secure product:

- IEEE 802 Nendica Report: Flexible Factory IoT: Use Cases and Communication Requirements for Wired and Wireless Bridged Network
    - The application of this standard to our project revolves around the aspects of real-time monitoring, automated data collection, and processing. The standard also indicates the need for security in data transmission to ensure there is no outside interference, which could jeopardize not only the data but also the monitoring system. To conform to the standard, a strong and secure connection between the microbial pill sensor and the external GUI application is necessary.
- IEEE Standard for User Interface Elements in Power Control of Electronic Devices Employed in Office/Consumer Environments: IEEE 1621-2004
    - The main parts of this standard that apply to the project are focused on the user interface and how to make that interface accessible and consistent for users. By creating an accessible system that clearly informs users what features are used and how the data is perceived, this being the GUI system to display concentration and temperature values, the user is guaranteed a consistent experience.
- IEEE Standard for a Real-Time Operating System (RTOS) for Small-Scale Embedded Systems: IEEE 2050-2018
    - This standard applies to the task management mechanisms employed by the project, which break into managing functions based on timers and time-related functions, and memory management. Following a standard that helps simplify and organize the many interconnected systems that are incorporated in the Microbial Pill sensor will be crucial to creating an effective product.

These standards are very applicable to the project as they relate to the front-end user interface, managing data functions to minimize disruptions and possible errors, and ensuring there is no interference with transferring data through Bluetooth. IEEE 1621-2004 reflects on the interaction that the project will have with users. Following a standard that focuses on creating a consistent experience will guide the foundation of the GUI. IEEE 2050-2018 reflects on creating a system that manages all the parts of a system while being efficient. Since the microbial pill sensor will have excitation and emission detection circuits and a data transmission microcontroller, creating an efficient and well-structured system is a must.

A standard that was previously selected to be reviewed but was rejected is the IEEE/IEC 62704-2-2017 standard, which is IEEE/IEC International Standard Determining the peak spatial-average specific absorption rate (SAR) in the human body from wireless communications devices, 30 MHz to 6GHz. The reason this was previously considered is due to the long-term goal of the product range of uses. This standard covers the research conducted into looking at how transmitting Wi-Fi signals through human tissue can cause complications with signal processing and concerns for human tissue. This standard wasn't selected due to two main factors: the first being that the project's requirements must use a low-energy data transmission system, which Wi-Fi is not, and the second being that the project's scope is limited to environmental monitoring, not internal health monitoring. There have been multiple similar biosensor systems that have been developed to be used inside the gastrointestinal tract of a human or livestock. However, this project does not aim to meet those requirements.

To ensure the incorporation of these IEEE standards considered in the microbial electronic sensor pill project design, we can modify the design to improve data transmission connectivity and device functionality. Adding specific user-friendly elements to the user interface design, such as easily visible alerts and notifications as well as power status indicators, would ensure that the project design abides by these selected focal engineering standards.

3 Project Plan

3.1 Project Management/Tracking Procedures

The project management style that the team is adopting is a blend between waterfall and agile. The beginning of the project follows an agile style of management, with each individual team member making progress on different subsystems. The modular design of our microbial pill sensor enables team members to work independently on different portions of the system, such as the external housing design, PCB design, or GUI implementation.

When implementing the final design, the team bounced back between an agile and waterfall style. A waterfall management style was used to test the individual modules in the integrated system. As errors and design improvements or updates were identified in testing, the team implemented an agile management style to reach the necessary solutions for each module. Throughout the project, each member was assigned the majority responsibility of a single module or functionality of the system, which made an agile project management style beneficial for design development and progression of the project.

The team's progress was documented by the Weekly Reports and individual notes recorded by each member as they progressed through their individual portions of the project. These weekly reports and notes have been documented on the Microsoft Teams channel, as well as the team website. From the beginning of the project, in alignment with the team contract, each member has communicated responsibilities, accomplishments, and setbacks. Weekly meetings with our faculty advisor/client have determined progress and helped guide future development and areas where more progress is required.

3.2 Task Decomposition

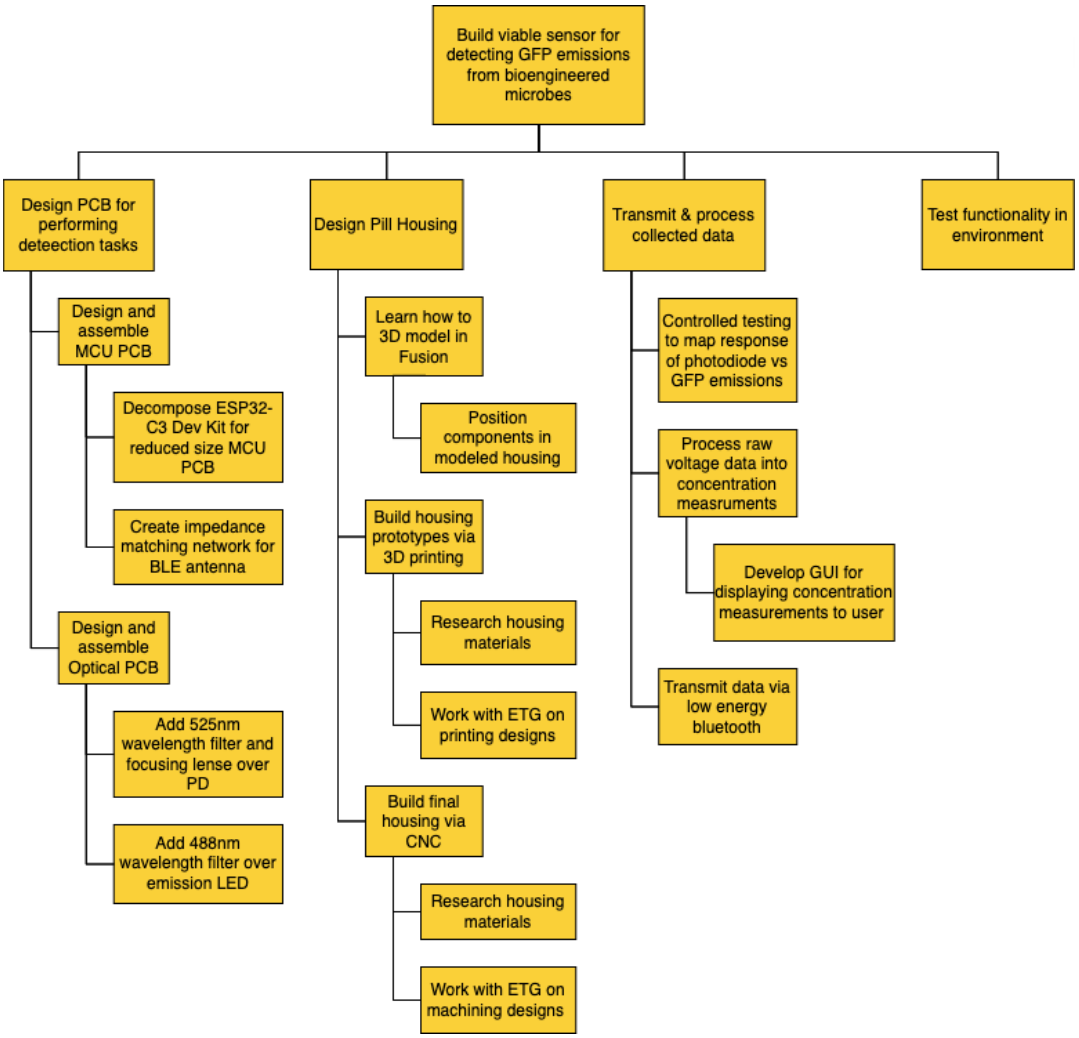

Figure 3-1: Task Decomposition Chart

The task decomposition chart depicted in Figure 3-1 was created to help boil down specific tasks into more manageable ones. The distinct functionalities of the microbial pill sensor, detection, data transmission, and housing of the microbes, can be seen in Figure 3-1. Subtasks to reach functional detection, data transmission, and the final housing were identified. The identification of subtasks provided each member with a clear and well-defined objective, rather than a general focus on the overarching goal of the microbial pill sensor. These subtasks were useful in understanding the entire system integration, as they provided an understanding of what functionalities needed to be added to the different modules to ensure their proper integration. Often, progress towards a subtask identified the need for an additional subtask, such as the addition of an emission filter was identified following assembly and testing of the optical detection PCB along with integration in the housing.

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Milestone 1: Develop a proof-of-concept prototype using BLE transmission of the light intensity of an LED via a photodetector component.

Milestone 2: Identification of all commercially available components used for the development of the final microbial pill sensor.

Milestone 3:

> **Milestone 3a:** Development of CNC acrylic chamber prototype for final microbial pill sensor.
>
> **Milestone 3b:** Development of optical PCB component for final microbial pill sensor.
>
> **Milestone 3c:** Development of microcontroller PCB component for final microbial pill sensor.
>
> **Milestone 3d:** Development of GUI for display of recorded measurements made by microbial pill sensor.

Milestone 4: Integrate individual modules together into a functional microbial pill sensor system.

Milestone 5: Implementation of microbial pill sensor successfully measures fluorescent response of GFP-expressing microbes in fluorescent beads.

Milestone 6: Testing and revision of the final microbial pill sensor in a functional environment produces accurate nitrate concentration measurements.

The milestones depicted represent the flow of project management implemented throughout the project as discussed in section *3.1: Project Management*. Milestone 3 depicts the portion of the project development process where an agile project management style was employed, where each individual member worked to complete the identified sub-milestones, corresponding to the various subtasks previously identified in *3.2: Task Decomposition*. The other identified milestones display the use of a waterfall project management style, where the entire team is actively working toward the completion of the same shared goal. Milestones 3 and 4 depict an iterative process, where progress towards Milestone 4 often requires adjustments and updates, returning the group to the individual development modules depicted in Milestone 3. Progress towards these milestones will be discussed later in *Section 6: Implementation* and *Section 8: Conclusions.*

Milestones 1, 2, 3, and 4 all correspond to the physical development of a component of the project or the physical development of a complete prototype. The evaluation of the completion of these milestones was qualitative, as will be discussed further when discussing the implementation of the microbial pill sensor. Visual examination of the GUI was used to characterize its proper functionality. Breadboard testing measurements were used to characterize the functionality of the PCB components, but quantitative accuracy measurements did not apply to these examinations.

Due to failure to reach the functional environment testing required to complete Milestone 6, quantitative benchmarks were not established for accurate nitrate concentration determination. Additional development of nitrate-responsive microbes by a third party was needed to determine microbe sensitivity and, thus, device sensitivity. Without the establishment of device sensitivity, quantitative benchmarks for the device's accuracy could not be established. Further development of the microbial pill sensor will lead to the development of quantitative benchmarks for the device performance.

3.4 Project Timeline/Schedule

An important part of project management and documentation is to develop a specific timeline for completing tasks and milestones. By decomposing each task into different variables, the team was allowed to create an accurate timeline and allocate resources to the areas of focus that require it. An updated Gantt chart, shown in Figure 3-2, was made prior to the start of the second semester of work to identify remaining tasks for the development of the microbial pill sensor. The developed Gantt chart follows a sprint schedule model, with each specific task having a dedicated timeline. Our task decomposition is identified on the left axis, where identification of subtasks and their dedicated timelines is used to further the entire module progression. The Gantt chart represents the proposed project plan and timeline. As in all projects, various setbacks, new issues, and changes in microbial pill sensor requirements prevented strict adherence to the proposed schedule. As can be seen in the depicted Gantt chart, tasks for the second semester centered around assembly, fabrication, testing, and integration of the completed modules.

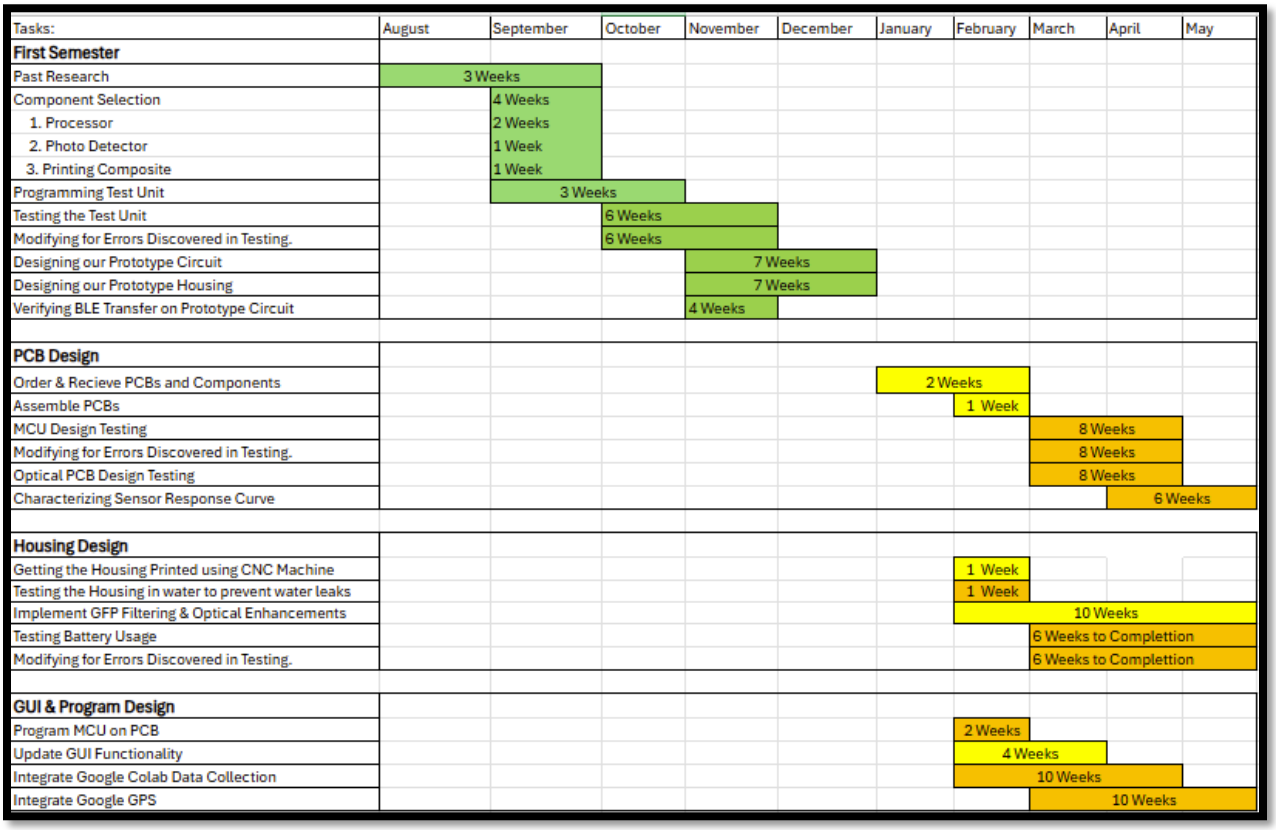

Figure 3-2: Gantt Chart

In the PCB design module, roughly 8 weeks were dedicated to testing and error modification following the arrival and assembly of the PCBs. Much of the proposed schedule matches the actual development of the PCBs. Modifying for errors, optical testing, and characterizing the sensor response curve extended until the end of April. Further development is required for characterizing the sensor response curve of the microbes. MCU design testing, as will be elaborated on in the testing portion of this document, was far briefer than expected and lasted only a week following delayed assembly.

The identified subtasks for GUI and Program Design reflect a change in the desired functionality of the microbial pill sensor. The integration of Google Colab for data transmission, processing, and display was

not possible. Via communication with our client, the design and subsequent subtasks were updated to a workable data transmission and display through Python.

When it comes to the housing design, the developed timelines were not followed at all. Delays in CNC fabrication of the housing extended well into April, limiting testing of the PCB design. The delay in the housing fabrication, along with delays in modifying for errors, prevented the identified subtasks of battery usage from ever being tested, and the implementation of the optical system from ever being conducted. Reflection on the proposed schedule depicts the interconnection of various subtasks, where delays in housing fabrication delayed the ability to properly test and characterize the detection modules of the sensor. The Gantt chart also depicts the flexible nature of our proposed microbial pill sensor, where various desired features of the device were changed throughout the project. While the Gantt chart initially served as a valuable resource for the dedication of personnel effort, deviation from timelines and changes in subtasks reduced its applicability towards the end of the project.

3.5 Risks and Risk Management/Mitigation

The task decomposition chart identifies six main subtasks that must be completed for this project to function:

- ST1: Design optical PCB for excitation and detection
- ST2: Design suitable housing chamber
- ST3: Transmission of data to external device via BLE
- ST4: Implementation of GUI to display transmitted data
- ST5: Integration of system modules
- ST6: Test system functionality in environment with live microbes

For ST1, the design of the optical detection PCBs could suffer from random dispersion of the emitted fluorescent response, reducing the amplitude of the incoming signal. A weak signal diminishes the detection limit and sensitivity of the microbial pill sensor and reduces the functionality of the device. This risk is mitigated by introducing a lensing system into the design that could be bought commercially and focus the produced fluorescent response onto the photodetector for increased signal strength. Additionally, this risk is mitigated by furthering the distance of the microbe cover slip from the lensing system to increase the amount of light that is able to reach the lensing system. However, this involves a trade-off with the intensity of the emitted light as the intensity decreases with an increasing distance from the detector. As will be discussed in *Section 5: Testing* and *Section 6: Implementation*, the discussed optical lensing system was not able to work as designed due to issues with the housing chamber. Further development of the housing design is required to provide a properly fixed lens position in the housing, as well as increase the distance to better allow light to travel to the photodetector.

A potential risk for ST2 is contamination or release of the microbes from the containment. The microbial pill sensor will house bioengineered microbes that could rapidly reproduce if released into the environment that the user wants to test. Therefore, ensuring that the housing design properly contains the microbes is necessary. Ensuring the durability and permeability of the housing properly contains the microbes will be required through the exploration of potential material options. Additionally, with environmental improvement as a motivation for the microbial pill sensor, the device must be environmentally friendly.

Therefore, ensuring the material used to build the housing chamber prevents damage to the interior electronics by the external environment while not harming the environment is a requirement of the microbial pill sensor. The development of the microbial pill sensor did not reach a point where containment of the microbes was prioritized, so further development of the device will assess this risk.

ST3 poses a potential risk of data not being sent to the GUI correctly, giving the user misleading information about the concentration of the analyte in the environment. As will be discussed in *Section 5: Testing*, early breadboard prototype testing was able to establish proper transmission of data via both serial and BLE connections using the dev kit MCU. Due to complications with transmission from our MCU PCB, which will be further discussed, the exploration of this risk was not able to extend beyond the dev kit MCU product to our decomposed MCU PCB. Further development of the MCU PCB will provide the necessary assessment of proper transmission.

ST5 represents a stage of development where risks were not properly accounted for. The integration of the separate modules proved more difficult than expected, as issues in input impedance arose between the optical detection PCB and the MCU dev kit. Little thought was put into potential errors regarding the ADC, and the mismatch between the recorded ADC voltage value and that measured via the ammeter was not solved. Further development of the microbial pill sensor data transmission will have to address this previously unaccounted-for risk.

Final system testing in the functional environment, a milestone that was not reached, brings complexities that may not be adequately represented by testing. These risks are represented in ST6, accounting for the complexities of submerging the sensor within the solution. Functional testing in a mock environment is required to assess risks to electronic components, transmission reliability, and intake membrane permeability. Additionally, the functional environment poses the same risk of allowing the microbes to escape the sensor capsule, potentially contaminating the environment. Thorough testing in a mock environment, truly representative of the functional environment, will be performed to mitigate these risks. Further development of the system is required to reach the functional environment testing for the microbial pill sensor.

3.6 Personnel Effort Requirements

As briefly discussed, the identification of subtasks through task decomposition, along with the development of a project timeline through a Gantt chart, enabled the team to identify which tasks required prioritization. The timelines previously depicted in the Gantt chart, Figure 3-2, roughly indicate the number of hours and effort dedicated to each task. Table 3-1 below shows a summary of the final person-hours required for the various subtasks that were worked on throughout the second semester of the project. A detailed estimate of personnel effort was not developed in the first semester, but can be assumed from the proposed Gantt chart, where each individual team member was assumed to be dedicating 8 hours of work towards the project a week. In Table 3-1, progression is tracked to the very specific tasks, and progress is documented by the hourly time invested. Additionally, person hours were documented towards specific tasks in the form of Weekly Reports, which enabled the assessment of personnel effort and the prioritization of different subtasks.

List of Tasks	**Team Hourly Contributions**	**Breakdown of Contributions**
Custom PCB Design & Assembly	52 Hours	• Designed custom Optical and MCU PCBs in KiCad ○ Completed design reviews with ETG to talk about design issues and potential fixes or workarounds • Selected components to be soldered onto custom PCBs • Assembled custom PCBs utilizing stencils and reflow oven as well as hand soldering. • Reviewed various photodetector circuits to optimize device sensitivity.
MCU & Optical PCB Testing	36 Hours	• Continuity and power delivery tests completed on custom PCBs • Initial operational tests with Dev Kit MCU to ensure Optical PCB workability • Identified updated resistor values following change in input resistance when moving from breadboard to PCB operation of photodetector. • Tested impact of emission and excitation filter on device detection.
Creating Sensor Response Curve	96 Hours	• Worked with the tested Optical PCB to characterize the response our PD has to the specified microbe emission light wavelength of 525nm ○ Tested various photodiode architectures to verify PD component selection will work for final product. • Mapped brightness values of LED in program to actual intensities of emitted light • Tested operation of optical detection outside of housing and characterized response as a function of LED brightness • Tested optical detection of fluorescent response of GFP-expressing microbes.

Housing Design	23 Hours	• Designed the cell housing based on the requirements.
		• Revised the design based on the drawbacks.
		  ○ Updated location of the emission and excitation holes for light to pass through
		  ○ Development of a blocking bar between the LED and the photodetector to prevent unwanted emission light from being detected by the PD.
		• Followed up for the 3d printing and the CNC design.

Fabrication & Testing Housing Design	10 Hours	• Tested operation of optical detection inside housing and characterized response as a function of LED brightness • Tested optical detection of fluorescent response of GFP-expressing microbes inside housing

Program MCU Operation	25 Hours	• Install Libraries and create setup functions for:
		  ○ BLE broadcasting and operating
		  ○ RGB LED operation
		  ○ 12-bit ADC Operation
		• Create looping operation protocol that controls the LED functions and period between collecting and sending a recorded voltage.
		• Ensure that data can be transferred via BLE or Serial connection based on user input.

Program GUI Operation with CSV Data Export	45 Hours	• Install Libraries and create setup functions for:
		  ○ BLE connection and COM Port recognition
		  ○ ADC voltage conversion
		  ○ CSV file creation and data-shoving
		  ○ Plot creation with correct axes, titles, and data display
		• Create main operating function which utilizes the GUI and User inputs in order to operate according to the user's needs.

Table 3-1: Personnel Effort Requirements

3.7 Other Resource Requirements

Financial resources were made available by the faculty advisor/client (Dr. Meng Lu) to provide the team with the ability to obtain all the necessary resources to design, purchase, and fabricate the necessary electronic components. PCBs, along with the necessary components, were purchased from the professional service company JLC PCB. On-campus soldering resources, made available through the Electronics Technology Group (ETG) and the ECpE department, were used to assemble necessary components on PCB boards. Additional components were purchased from Digi Key for initial breadboard prototype testing. Other components, such as the various optical filters used, the ESP32-C3 microcontroller dev kit, and lenses, were obtained directly from Dr. Lu's lab on campus. Equipment in Dr. Lu's on-campus laboratories was used for various testing procedures, such as emission spectrum analysis and emission intensity measurement. Software, GUI, and transmission development were conducted through publicly available software and resources such as Arduino IDE, Python, and available ESP32 libraries.

The housing chamber was fabricated numerous times, both using 3D printing and CNC machining throughout the duration of the project. Necessary 3D printing was made available through ETG, Dr. Lu's personal machine, and the 3D printing machine available in Dr. Lu's on-campus lab. CNC machining was made available and performed via ETG. Clear acrylic material for CNC machining was made available by Dr. Lu, and a dark acrylic material for CNC machining was purchased through ETG. Reliance on ETG for CNC machining also made the time of ETG a necessary resource for our project, and housing fabrication was often delayed by the service time of ETG. The GFP-expressing microbes were made available for testing via Dr. Lu's collaborators in the Biorenewable Research Laboratory. Proper handling equipment and disposal were handled through Dr. Lu's on-campus lab.

4 Design

4.1 Design Context

4.1.1 Broader Context

Farmers in the United States use around 150 pounds of nitrogen per acre. Average nitrate runoff is around 20 pounds per acre, contaminating waterways and public bodies of water. Excess nitrate and nitrogen levels in water bodies lead to excessive algal blooms, a process known as eutrophication. Excessive algal blooms reduce oxygen levels, damaging fish and other aquatic life. Additionally, high levels of nitrate also pose health risks to humans and other lifeforms consuming contaminated water. Current commercially available measurement methods for nitrate levels in runoff require sampling and off-site testing, limiting the frequency and availability of testing. The recent development of advanced technologies in nitrate detection has increased the autonomy of nitrate detection, but at a significant increase in cost, with a reduced lifetime. Our device aims to provide an easier route to determine nitrate concentration in agricultural runoff and other water bodies by providing continuous, cost-effective, and easy detection. Through continuous monitoring, farmers can more accurately assess fertilization strategies and amounts. Our device also aims to provide environmentalists with more frequent information about hazardous conditions in waterways, providing a route to increase the quality of our waterways and limit public health crises.

Area	Description	Examples
Public health, safety, and welfare	Provides environmentalists and farmers with real-time information regarding nitrate levels that allows for the improvement of waterway health.	Farmer John realizes that the nitrate levels in runoff to the south of his field are very high and addresses the problem through changes in fertilizer application.
Global, cultural, and social	Our project aims to improve environmental monitoring and make sure to use environmental protective practices.	Our capsule will be sealed so that no degradation of battery capsule results in environmental damage.
Environmental	Our project aims to improve environmental monitoring and make sure to use environmental protective practices. By providing improved environmental monitoring, we provide professionals with the proper tools to identify environmental issues as quickly as possible.	By identifying high levels of nitrate in a public waterway, environmentalists can work to improve the water quality of that water body.
Economic	Our product will be low cost and require no installation cost. This provides a cheaper alternative to current nitrate monitoring method.	Our capsule is cheaper than sending an environmental water sample to a professional lab for analysis, providing farmers and environmentalists cheaper, faster, and more effective sensing solutions.

Table 4-1: Design Implications

4.1.2 Prior Work/Solutions

Our biosensor capsule is a novel product, as no research or product has been documented using bioengineered bacteria for the detection of nitrate via an electronic capsule pill. Previous research has been conducted to demonstrate the ability of bioengineered bacteria to act as biosensors for nitrate, but none of these groups have integrated these bacteria into an electronic capsule for continuous environmental monitoring. Similar electronic biosensing capsules containing bioengineered bacteria have been documented, but have different analytes and different application environments.

A similar biosensor [**1**] using bioengineered microbes integrated into a miniature electronic capsule to detect gastrointestinal inflammation using bioluminescence in the presence of the target analyte provides the motivation for the design. The capsule uses a membrane to intake the solution into the microbe housing chamber and uses a proprietary threshold-based bioluminescence IC integrated with four photodiodes to identify the concentration of three different analytes in the solution simultaneously. Our microbial pill sensor aims to provide sensing capabilities for different target analytes and uses BLE for transmission rather than Wi-Fi. By using BLE, the microbial pill sensor will exhibit lower power consumption, which is advantageous for our application in the continued monitoring of agricultural runoff.

Bioengineered microbes provide significant advantages in specificity and robustness compared to current biosensing alternatives, such as electrochemical, microfluidic, or acoustic devices. *Escherichia coli* [**2**] has been bioengineered to detect nitrate through regulated expression of GFP. While the techniques, methodologies, and principles of bioengineered bacteria are beyond the scope and responsibilities of this project, the active development of microbial-based biosensing solutions compatible with the microbial pill sensor increases the device's functionality.

A commercially available nitrate sensor is produced by *Clear Water Sensors* [**3**]. This sensor uses advanced microfluidic lab-on-chip technology to perform nitrate and nitrite monitoring in rivers, wastewater, oceans, and autonomous vehicles with a maximum depth of 6000 m. The device has a total height of 56 cm, which is much larger than our proposed design. With a power consumption of 1.8 W and a limit of detection of .05 μM, this device provides a performance standard our microbial pill sensor should aim to reach in terms of sensitivity and power consumption. The device also has a lifetime of 1000 readings per canister, which results in 250 hours' worth of data collection per canister. Our microbial pill sensor should have an extended lifetime to provide a cheaper, continuous monitoring biosensing solution. This device, while relying on a different detection mechanism, provides market performance standards that our microbial pill sensor should meet to provide a viable alternative in the market.

4.1.3 Technical Complexity

The microbial pill sensor is a complex and novel engineering solution for a direct application in improving environmental monitoring strategies. The team aims to generate a first-of-its-kind solution that is not currently available in the market. The design consists of multiple components, identified as a housing chamber, an optical detection module, a microcontroller module, and an external GUI application. Each module is complex and presents its own subsets of challenges. Size restrictions require a well-designed housing chamber, along with an extensive evaluation of available components for the optical detection module. The optical detection module also poses challenges related to optical lensing and filtering, with our excitation and emission wavelengths being relatively comparable. Reduction of a commercially available MCU board to fit our available size requirements requires an understanding of circuit and PCB design. Additionally, the programming required for the successful function of the MCU, and the GUI requires significant hardware, software, and programming skills.

4.2 Design Exploration

4.2.1 Design Decisions

The microbial pill sensor's physical system layout was the most influential and first choice made in the design process. The team opted for a vertical design with the modular systems stacked, with each module component responsible for a specific function in the design. Size constraints limit the applicability of any complex optical systems for excitation and emission beam direction, forcing the excitation light to come from the same plane as the emission photo detection. The vertical layout of the system is compatible with the excitation LED and the emission photo detector being on the same PCB, with the MCU PCB connected underneath and the microbes directly overhead. The modular design allows for a further breakdown of individual members' responsibilities into a singular component.

Following the design choice of physical system layout, design decisions related to selecting the necessary components for the microcontroller and optical detection modules, as well as the circuit choice for the photodetection system. Component choices are crucial to the ability of the system to meet system requirements and functionality while maintaining the economic feasibility of the product. Considering system requirements in BLE transmission, reduced power consumption, and sizing, along with user needs and economic cost factors, the team decided that the ESP32 C3 microcontroller was optimal for the microbial pill sensor. The EOPD-525-1-0.9-1 photodiode was selected for photodetection due to its low cost per unit, correct peak wavelength sensitivity at 525 nm, and a high sensitivity of .3 A/W. The SK6812MINI programmable RGB LED was selected to generate the fluorescent response of microbes, largely due to its programmable nature and because it was the LED already used on our ESP32-MCU dev kit, used in breadboard prototype testing. As will be discussed throughout the remainder of this report, this was a poor design decision that limited the functionality of the system due to a misunderstanding of the functionalities of this device. Short lead times were also a factor in the decision to use the EOPD-525-1-0.9-1 photodiode and the SK6812MINI. The choice of filters and lenses used in the microbial pill sensor was not an active design decision, as we used the components made available by our faculty advisor/client.

As will be discussed in *Section 5: Testing*, a non-biased configuration was chosen for our photodiode following extensive testing of various photodiode configurations and analyzing the various response curves. Following the component and photodiode configuration selection, the PCB layouts were designed to properly position the necessary components for optical detection and make the necessary connections between the LED, photodiode, and MCU GPIOs. The MCU PCB consists of a reduction of the ESP32-C3 dev kit, where only the necessary components were included to meet design size requirements. The placement of the optical components within the housing was updated to properly align with our photodiode and LED placement on the PCB. The design decisions reflected here were the result of numerous prototyping and testing steps, which were used to refine the design to a working system.

4.2.2 Ideation

Creating a functional project inside a capsule is incredibly difficult. Factors the team has considered include the physical space, orientation of components, and modular connections inside the project. The original proposed idea was to surround the housing chamber with all the sensor monitoring systems. This would mean the bacteria housing chamber extended deep into the center of the project. After reviewing the design with Dr. Lu, simplifying the system into modules would improve the design. This decision resulted in discussions about the requirements and how to meet these needs while keeping a simplified design. Creating a vertical system that breaks down into separate compartments and PCBs simplifies the physical and spatial design. Initially, the options revolved around organizing the modular PCBs into a functional order. Understanding that the GFP sensor PCB should be placed directly below the housing container and

by implementing an optical lens that will disperse the excitation light and a focal lens to collect GFP emissions directly into the phototransistor, the design reached closer to the final design. Another design feature that was considered and is waiting to be implemented into the design is the use of double-sided PCBs to minimize space. Since the GFP sensor system will be controlled by the microcontroller PCB, combining these two systems onto one board simplifies the project and enables satisfaction of the sizing requirement.

4.2.3 Decision-Making and Trade-Off

Weighted Decision Matrix					
	Economic Consideration	Technical Needs	Simplification	Meeting Requirements	Totals
Weights	3	4	2	5	14
Central Housing Chamber	1	2	1	1	18
Vertical Pill Chamber	1	2	2	2	25
Modular System Breakdown	2	2	3	3	35
Focal Optical Lensing	3	4	4	4	53
Double-Sided PCB	3	5	4	5	62

Table 4-2: Weighted Decision Matrix for Ideated Design Considerations

Table 4-2 shows the weighted decision matrix for the project's design considerations. It follows the order of design features as they were created and implemented. A key feature of this project, which varies from others, is that our project doesn't have a very wide range of options to consider, so rather than generating and scrapping past ideas, the team has prioritized implementing useful considerations and revising them for improvements. The design considerations in the table all fall under the umbrella of system layout. Referencing the ideation of each consideration, it can be concluded that as each portion brought about positives, the next ideation works to fix the negatives of the previous creation.

This can be seen in the figure as the most recent feature implemented into the design is the double-sided PCB, and it has the highest weighted total. This total is a representation of how the team feels the ideation impacts the project's demands and design steps. Having a higher value results in a larger positive impact on the outcome of the project. The reason that the team has valued this final design idea for the spatial considerations is due to how it meets and improves all the physical and technical considerations, as well as improving the simplification of the design will reduce the amount needed for multiple PCBs. Integration of ideas through considerations and analyzing the impact an idea will have on a final product is crucial to creating a smooth, constantly developing design process.

4.3 Proposed Design

4.3.1 Overview

The microbial pill sensor must detect the desired analyte concentrations in the functional environment while meeting sizing, power consumption, and BLE transmission requirements. The following microbial pill sensor design considers system requirements, as well as user needs, environmental considerations, and economic feasibility. Figure 4-1 provides a physical 3D view of the microbial pill sensor. Figure 4-2 depicts a system sketch overview of the design, detailing the connections between the individual modules. The proposed design breaks the entire system into multiple subsystems or modules, each with a desired and specific functionality. These modules are a housing chamber designed to contain the microbes while flowing in the solution, an optical detection module to excite and measure the GFP fluorescent response, an MCU module to transmit the recorded voltage produced by the optical detection module, and a battery housing component to provide power. The physical 3D design vertically integrates these modules, meeting sizing requirements while providing easy access to the battery and microbe housing chambers. Providing access to battery and microbe housing chambers, while not incorporated in our system requirements, satisfies user needs relating to the continued operation of the microbial pill sensor, allowing users to easily replace the batteries or microbe nutrient gel.

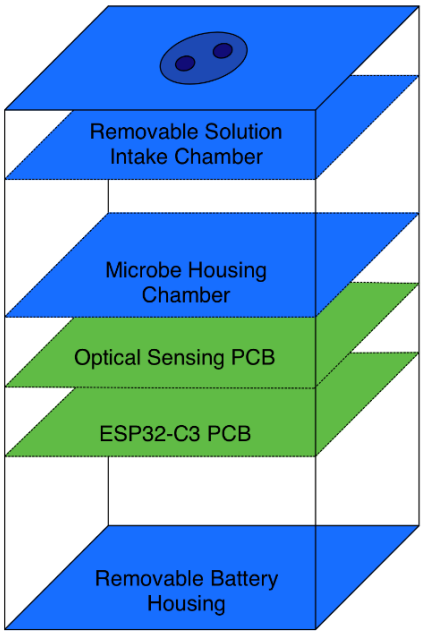

Figure 4-1: 3D Design Sketch

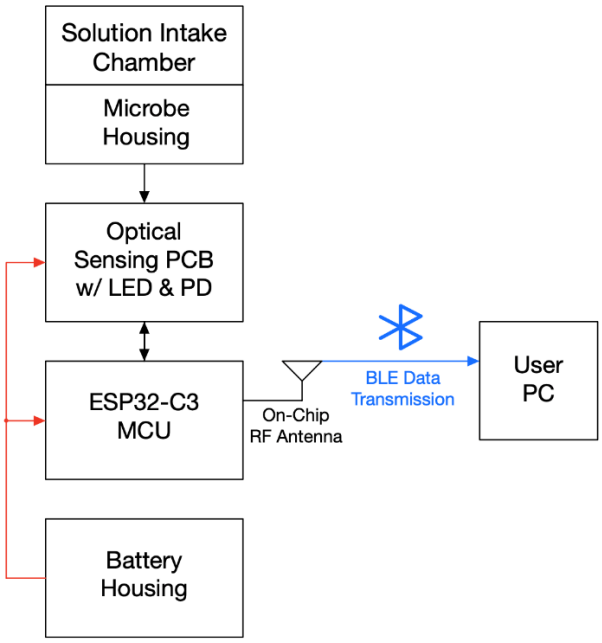

Figure 4-2: High Level System Breakdown

4.3.2 Detailed Design and Visual(s)

As previously described and visualized in *4.3.1: Overview*, the design of the microbial pill sensor is modular, resulting in the integration of multiple distinct components into the final device. The following sections will be broken down into describing the design of these individual modules. Integration of these modules involves the wiring connections between the optical detection PCB, the MCU PCB, and the battery, and placing the PCBs, battery, filters, and lens in the designated places within the housing.

Exterior Housing Design

A CAD model of the exterior housing for the microbial pill sensor has been designed and is shown in Figure 4-3, following the physical layout presented earlier in Figure 4-1. Initially, the design was envisioned with very compact dimensions, set at 5 x 5 x 10 mm^3. This original concept included distinct compartments for the battery cell, microbe housing, optical PCB housing, MCU cell, battery MCU cell, and battery MCU cell cap. Specific dimensions were allocated: the battery cell compartment was designed at 3(+0.2) x 5.3 x 5.3 mm^3, with an additional 0.2 mm height intended to connect the battery cell to the MCU housing compartment, measuring 3.2 x 5.3 x 5.3 mm^3. The microbe housing compartment dimensions were set at 3.8(+0.2) x 5.3 x 5.3 mm^3, capped by a detachable lid measuring 2.6 mm x 5.3 x 5.3 mm^3. Further considerations regarding the choice of material and coating required for fabrication were noted, and updates were planned following material assessments.

However, after initial prototyping with filament-based 3D printing, it became clear that the small dimensions posed significant fabrication challenges. Additionally, feedback from our advisor indicated that a larger size was necessary to adequately accommodate the MCU and sensor PCBs. Consequently, the prototype was redesigned to larger overall dimensions of 22 x 22 x 24 mm^3. The revised housing dimensions included a battery compartment of 22 x 22 x 7 mm^3, with an additional 2 mm allocated to connect it to the MCU housing compartment of dimensions 22 x 22 x 8 mm^3. The microbe housing was expanded to 22 x 22 x 9 mm^3, featuring a detachable lid measuring 22 x 22 x 3.5 mm^3 for ease of maintenance and microbe replacement.

Despite these adjustments, challenges persisted with filament-based 3D printing, particularly concerning detail precision and accuracy. To overcome these limitations, CNC machining was explored as an alternative fabrication method, offering enhanced accuracy and the potential for airtight sealing. Two primary materials were considered for CNC machining: acrylic and aluminum. Acrylic provided ease of machining but posed risks due to its vulnerability to melting under heat exposure. Aluminum offered greater structural integrity but raised concerns about corrosion and rust when exposed to water, though this could potentially be mitigated through appropriate coating.

For initial testing purposes, acetyl was selected due to its balance of machinability and durability. Additionally, design modifications were implemented to facilitate both battery power and USB input configurations within the same housing design by altering only one compartment. Updated dimensions were introduced to accommodate these features effectively. The updated design dimensions are 30 x 30 x 47 mm^3 for the version integrating a USB MCU, and the dimensions of the MCU with the battery is 30 x 30 x 60 mm^3 for broader compatibility. The final housing design now includes a battery compartment measuring 30 x 30 x 24 mm^3, capped by a detachable lid of dimensions 21 x 21 x 6 mm^3. An additional 3 mm height was incorporated to connect this compartment seamlessly to the MCU cell, sized at 30 x 30 x 12 mm^3. Lastly, the microbe housing compartment dimensions were finalized at 30 x 30 x 26 mm^3.

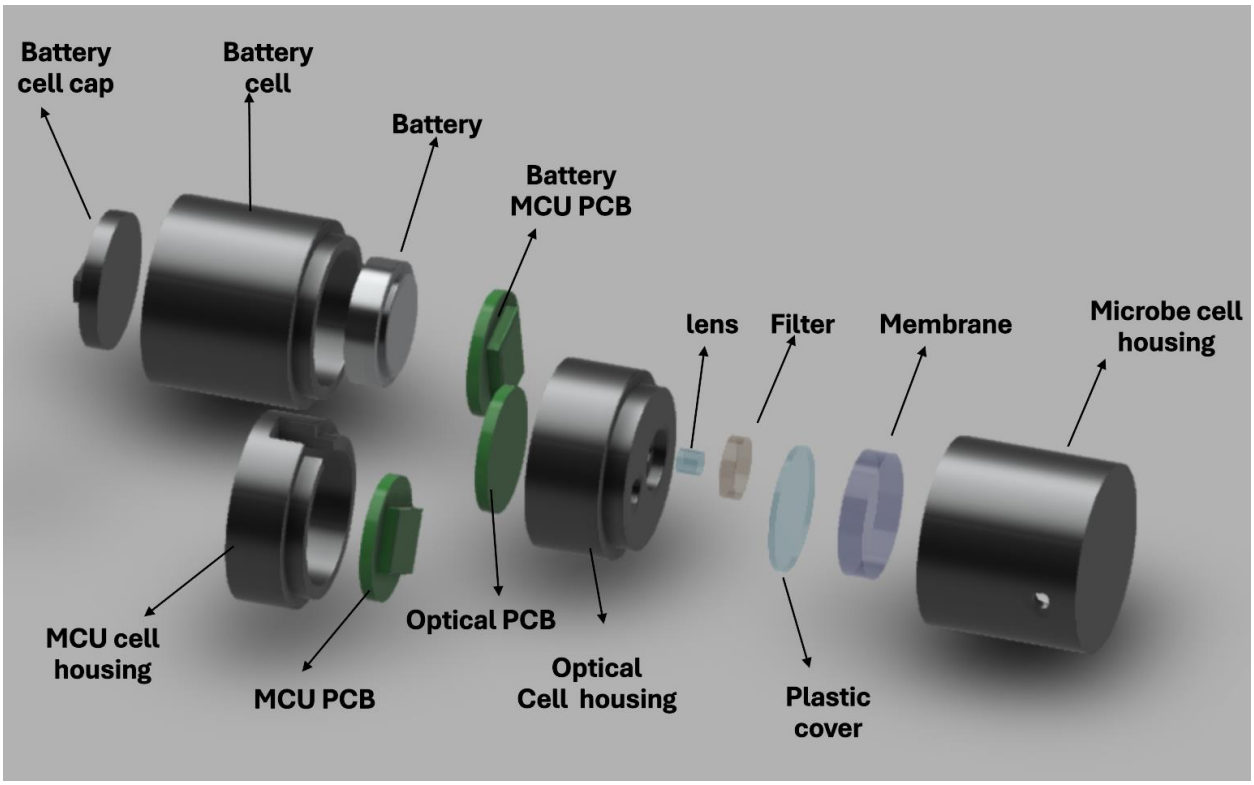

Figure 4-3: Exploded View of Housing Design

Microbe Housing Chamber

The microbe housing chamber has undergone a significant redesign to address both functional requirements and fabrication challenges. This chamber is specifically intended to contain bioengineered microbes embedded within agar gel, facilitating efficient optical biosensing through its transparent bottom interface. Initially, the chamber was designed with dimensions of 3.8(+0.2) x 5.3 x 5.3 mm^3, capped by a detachable lid measuring 2.6 x 5.3 x 5.3 mm^3. However, this original design proved impractical for filament-based 3D printing, primarily due to its limited dimensions and insufficient internal space.

Responding to these challenges, the chamber dimensions were increased to 22 x 22 x 9 mm^3, along with a detachable lid sized at 22 x 22 x 3.5 mm^3. This updated design provided adequate internal volume for microbial growth, facilitated easy replacement of nutrient gel, and allowed seamless integration of optical sensors necessary for accurate GFP fluorescence detection. The detachable lid enhanced accessibility for maintenance, significantly extending the operational life of the device.

Despite these enhancements, filament-based 3D printing still faced precision limitations, especially at this compact scale. Therefore, the fabrication method was shifted to CNC machining, allowing for more precise manufacturing and better structural integrity. The final optimized design now features dimensions of 30 x 30 x 26 mm^3, ensuring ample space to house microbes effectively. Additionally, an inlet was incorporated into the chamber to allow controlled flow of water both into and out of the cell, improving the device's practicality and functionality in real-world applications

.

Optical Cell Housing

The optical cell housing compartment was designed specifically to accommodate essential components, including the optical PCB, optical filter, and lens, within dimensions of 30 x 30 x 12 mm^3. Strategically integrated into the modular sensor assembly, this chamber significantly improves component accessibility and facilitates routine maintenance of optical and electronic elements.

The optical cell housing is carefully engineered with strategically placed openings to maximize the sensor's optical efficiency. One side features an aperture that allows excitation light to enter the microbial housing chamber, thereby activating the bioengineered microbes. On the opposite side, a secondary passage is constructed with two precisely designed layers—one to hold the optical filter and another to secure the lens. The optical filter, measuring 10 mm in diameter and 2 mm in height, is positioned to selectively allow light of 532 nm wavelength to pass through. Behind it, a lens measuring 4.5 mm in diameter and 3 mm in height is held in place by a reduced-diameter section at the end of the passage, effectively acting as a mechanical bump to prevent displacement. This layered configuration ensures that only the desired fluorescent light reaches the lens, which then focuses the light precisely onto the photodetector, thereby improving signal clarity and detection accuracy. The optical PCB is carefully positioned so that the photodetector aligns precisely with the focal point of the lens, maximizing detection efficiency.

Initial prototypes produced using filament-based 3D printing revealed considerable drawbacks—specifically, material-induced light absorption that degraded the optical signal and impaired sensor performance. To overcome these limitations, the fabrication process was transitioned to CNC machining, which provided higher dimensional precision, smoother surfaces, and better optical compatibility. This shift significantly reduced light interference and enabled more efficient transmission through the optical components, ultimately enhancing the overall accuracy and reliability of the sensor's data collection system.

The finalized optical housing maintains a critical distance of exactly 3 mm between the lens and the photodetector, carefully matched to the lens's focal length. This precise spacing ensures optimal detection sensitivity and accuracy in capturing microbial fluorescence signals, thus enhancing the microbial pill sensor's reliability.

For future enhancements, special attention is required to address challenges related to lens stabilization. Initial tests showed the lens shifting from its designated position, potentially affecting optical alignment and reducing the gap between the lens and filter. This issue was addressed by incorporating an additional structural element—a small bump or block positioned above the lens—to securely anchor it.

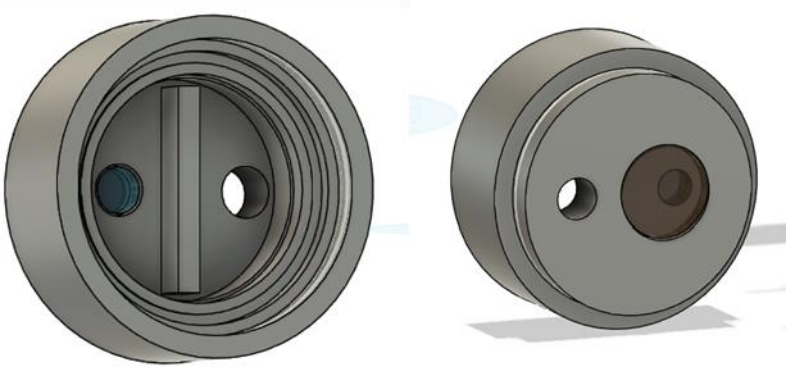

Figure 4-4: Optical Cell Housing

Optical Detection PCB

The optical detection module is the most crucial component to the functionality of the microbial pill sensor. Without the proper design of the optical detection module, the sensor will lack the ability to correctly detect the desired analyte in solution, rendering the device ineffective or even useless. The following circuit diagram, shown in Figure 4-5, depicts the system-level design of the optical detection module.

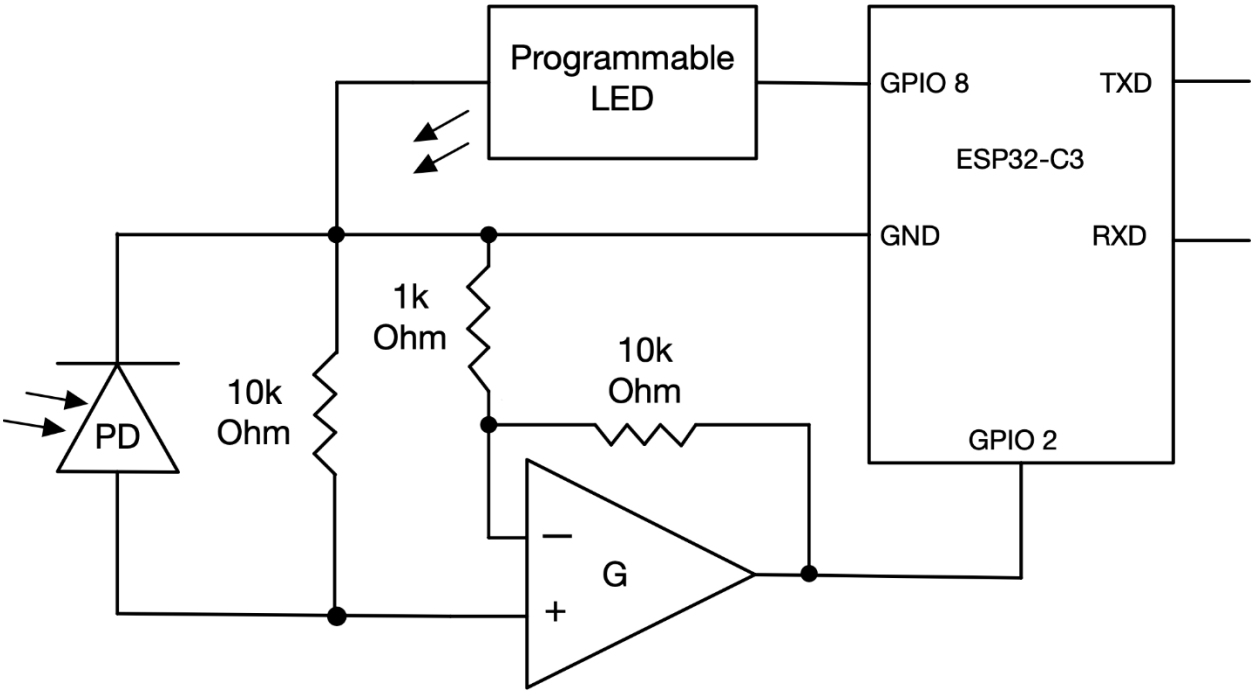

Figure 4-5: ESP32-C3 PCB Prototype Schematic

Figure 4-5 depicts the necessary connections between the ESP32 C3 microcontroller, the LED, and the photodetection component. The ESP32 C3 activates the 488 nm LED component at the desired timing interval, exciting the microbes with the necessary blue wavelength. The recorded fluorescent response is measured by the photodetection component, which produces a voltage proportional to the light intensity produced by the excited GFP expressed by the microbes. This voltage, in turn, is proportional to the concentration of nitrate. Amplification of the recorded voltage to a meaningful value is performed by an LM324 operational amplifier. This voltage value is then passed to the ESP32 C3 microcontroller through the ADC on GPIO port 2.

A bandpass filter, passing ranges 532 ± 20 nm, must be used to ensure that the produced voltage is not generated by any additional wavelengths of light other than those of GFP response due to nitrate present in the solution. Additional lensing will provide a more consistent response by focusing the emitted light on the photodiode. The addition of optical filtering and lensing components, obtained from our faculty advisor/client, is meant to increase sensitivity to nitrate concentration, making the microbial pill sensor a

more effective biosensing solution. The selected components were introduced in *Section 4.2.1: Design Decisions*, but more details are provided here.

The current choice for the LED component is the SK6812MINI. This device provides a controllable RGB LED in a 3.5 x 3.7 x .95 mm top SMD. This device is integrated into the ESP32-C3 dev kit used for prototyping. Digital control of this device is managed by the ESP32-C3 MCU via the Arduino IDE Adafruit NeoPixel package, which changes the relativity intensity of a red, blue, and green LED. Further details regarding the component, including depictions of pin layouts, optoelectronic characteristics, and spectral emission, can be found in Appendix A. As was previously introduced and will be further discussed in *Section 5: Testing*, the datasheet for this device was misunderstood, and it does not enable the microbial pill sensor to function properly, as the relative intensity of 488 nm light emitted by the device is too low to generate a measurable fluorescent response in the microbes. See Figure A-2 to see the low relative intensity of 488 nm light emitted by the device.

The design of the microbial pill sensor should be updated to instead use a blue LED with a peak emission wavelength around our desired excitation wavelength of 488 nm. Updating the choice of excitation LED will also require an update to the footprints used on the optical detection PCB. The design has not been updated to a better-suited LED in our implementation of the microbial pill sensor due to time constraints. Further development of the microbial pill sensor, which will be further discussed in *Section 8.3: Next Steps,* will update the design to a better-suited blue LED.

The current choice for the photodetector is the EPIGAP OSA Photonics EOPD-525-1-0.9 525 nm photodiode. With a peak sensitivity at 525 nm, sizing of 3.2 x 1.6 x 1.2 mm, and a responsivity of .3 A/W, this photodiode provides the desired functionality of the module. The photodiode is non-biased and in parallel with a 10k ohm resistor in our current design of the optical detection circuitry. Note that the 10k ohm resistor in parallel is not included on the PCB schematic, as it was soldered in parallel with the photodiode on the PCB following the investigation of different circuit configurations. Adding the photodiode configuration via soldering allowed for testing of different circuit configurations directly on the PCB with the photodiode, as breadboard prototype testing was not available for the selected photodiode. Further details and specifications regarding the photodiode can be found in Appendix B. An LM 324 operational amplifier is used to amplify the signal to a measurable level. The current design of the optical detection circuit gives a 10x gain to the signal. Further development of the microbial pill sensor should investigate the impact of different operational amplifiers on device speed and performance.

Multiple circuit configurations, including reverse bias, forward bias, photovoltaic transimpedance op amp, photoconductive transimpedance op amp, and non-biased configurations, were tested with different associated resistor values to determine which configuration provided the best performance for the microbial pill sensor. The different circuit configurations were tested using the SK16MINI RGB LED's green light output, where the brightness value was varied to change the intensity of light hitting the photodiode from the same distance as the microbes will be from the photodiode. The brightness values of the programmable LED were mapped to intensity values using light intensity measurement equipment in Dr. Lu's on-campus lab. The transimpedance op amp configurations were undesirable due to a slow response time, where the output voltage due to light was held for a significant time before dropping after the light shining on the photodiode was turned off. The slow response time of these configurations, leading to an increasing dark current, prevents the applicability of these configurations for detection in our system. Figure 4-6 depicts the responses of reverse bias and non-biased testing with different resistor values on the breadboard, where the output voltages have been normalized by the voltage value output at the maximum brightness. Measurements were taken using an ammeter available in the TLA on campus. Normalization of

the output voltages allows for direct comparison of the sensitivity of the different configurations. The maximum voltage values used to normalize are included in the legend of the plot.

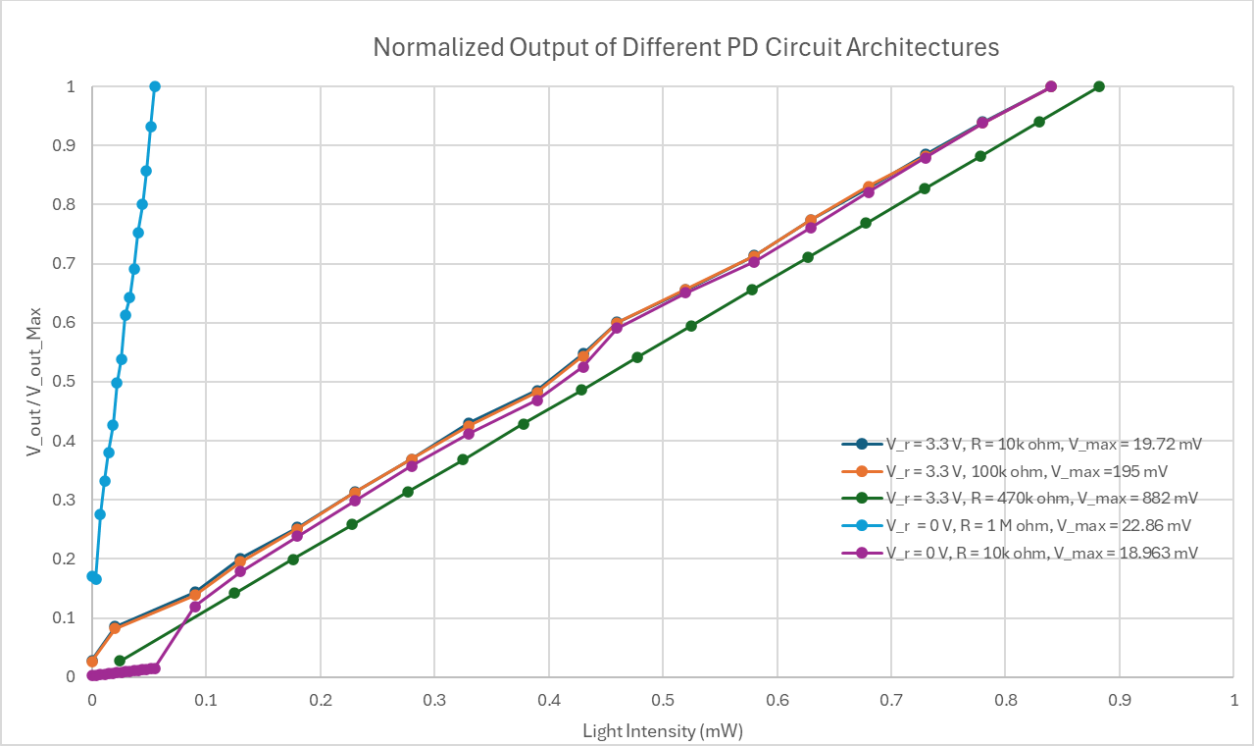

Figure 4-6: PD Circuit Response Curves

Here, it appears that the non-biased configurations are preferable, as the performance is similar without the need to introduce a biasing voltage to the terminal of the photodiode. Additionally, the reverse-bias configurations showed significantly more noise when reading values from the ammeter. While the non-biased, 1 M ohm demonstrates the most sensitivity, the 1 M ohm resistor was shown to lower response time. While it is not included in the figure, the ratio of the resistance value compared to the input resistance of the op amp had a major impact on the configuration performance. The 1 MΩ results pictured here were gathered using a breadboard op amp with an input resistance of ~1.8 MΩ. When the 1 MΩ resistor was used with the op amp on the PCB, with an input resistance of ~50 kΩ, the performance was damaged by the lower input resistance of the smaller op amp on the PCB. Thus, the non-biased configuration with a 10 kΩ resistor in parallel was chosen. Further development of the microbial pill sensor should explore lower resistance values, such as 1 kΩ, to investigate the impact on device sensitivity. One advantage of the smaller resistance value is a reduction in power consumption by the device, but the voltage values were significantly lower, in the range of ~ 10 mV, when using the smaller resistance values. The operational amplifier is then used to amplify this signal to a measurable level. Testing measurements characterizing the response of this device in the housing will be presented in *Section 5: Testing* and *Section 6: Implementation.*

Figure 4-7 depicts the designed KiCad schematic, utilizing the LM324 operational amplifier, the SK6812MINI RGB LED, the EOPD-525-1-0.9 525 nm photodiode, and other supporting passive components such as resistors, capacitors, and inductors. PCB layout footprints were assigned to all components based on their corresponding datasheet sizing requirements. As previously noted, the 10 kΩ resistor in parallel with the photodiode was soldered onto the PCB.

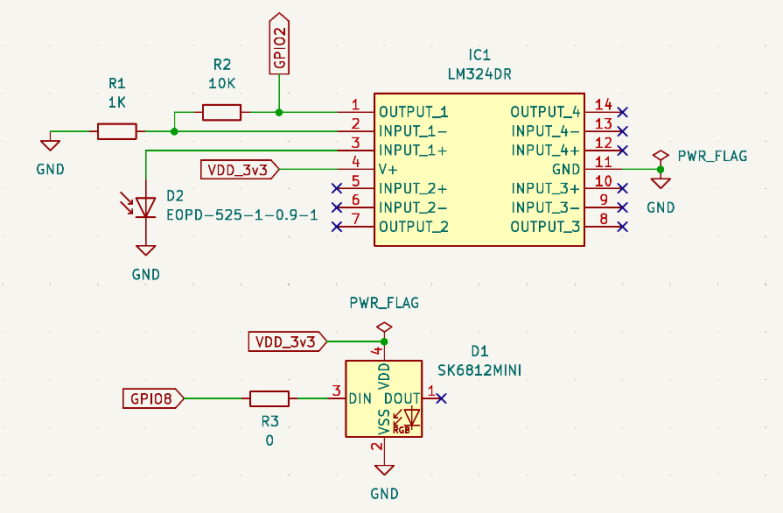

Figure 4-7: Custom Optical Sensor PCB Schematic

Figure 4-8 depicts the 2-layer PCB layout with all necessary connections made and footprints selected. The outer dimension of the PCB is currently set at 20 mm. Although this is larger than the minimum sizing requirement, the larger size enables testing of the device without being limited by soldering capabilities. Note that D2 (the photodiode) and D1 (the RGB LED) are spread out to allow room for the inclusion of lensing and filtering components. The housing design matches the locations of the photodiode and LED on the PCB to the necessary optical detection components and pathways for light to travel. Three open pads on the back copper of the PCB have been placed as solderable jumper points between the optical detection PCB and the MCU PCB.

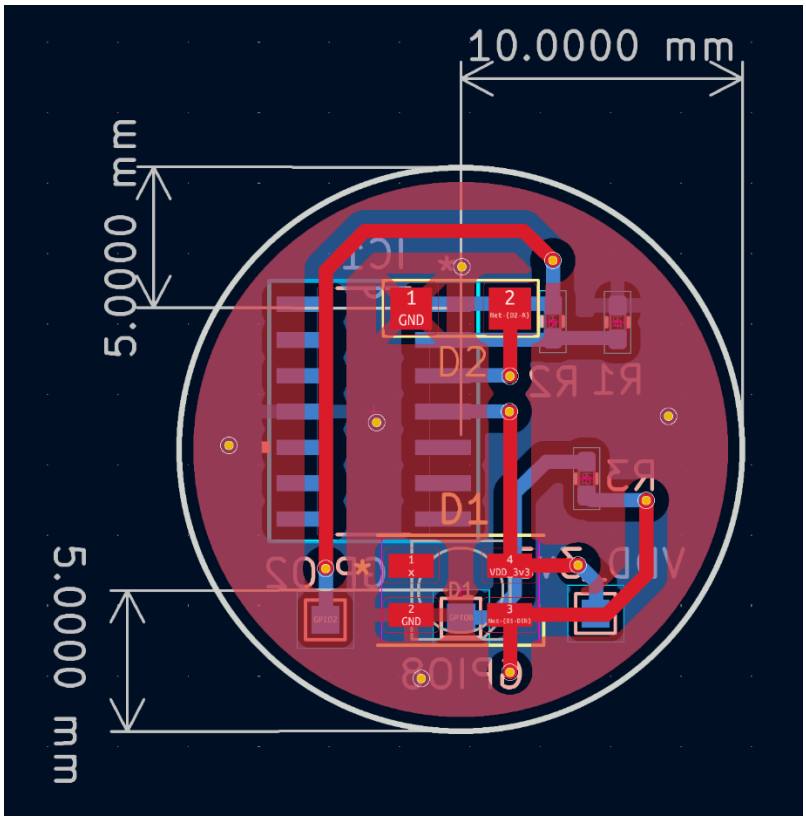

Figure 4-8: Custom Optical Sensor PCB

Microcontroller PCB

The microbial pill sensor requires a microcontroller unit to properly activate the LED at the desired timing interval and transmit recorded measurements via BLE. As previously discussed in *Section 4.2.1 Design Decisions*, the ESPRESSIF ESP32-C3 was identified as the MCU to be used due to its BLE transmission capability. Breadboard prototyping for the microbial pill sensor used the ESPRESSIF ESP32-C3-DevKitC-02, and this dev kit was used as the starting point of our MCU PCB implementation. A parts decomposition on the ESPRESSIF ESP32-C3-DevKitC-02 schematic was completed, see Figure 4-9, to identify the components from the dev kit that were necessary for the MCU PCB. The parts decomposition enables the functionality of the MCU dev kit while meeting sizing requirements. Initial designs of the MCU PCB include the USB port to enable coding of the MCU.

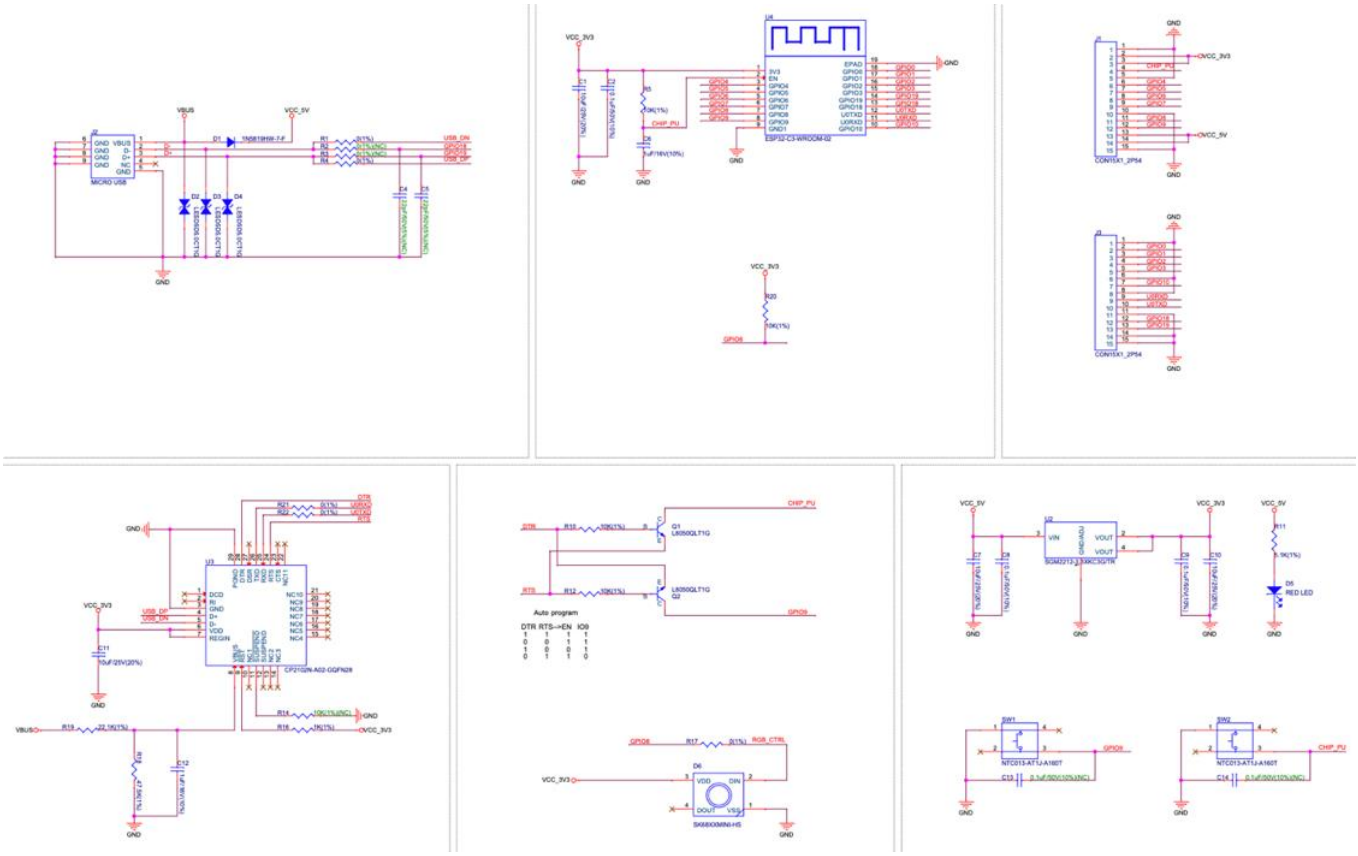

Figure 4-9: ESPRESSIF ESP32-C3-DevKitC-02 Schematic

Figure 4-10 depicts the custom MCU PCB design, showing pin connections between the various devices and components. Here, the push buttons and corresponding switching current amplifiers have been removed. The linear regulator was also changed from the SGM2212-3.3 in the dev kit to the LD1117S33 (component U4 in Figure 4-10) due to stock limitations of the original regulator. The LD117S33 has very similar operation, allowing continuous regulated 3.3 V output, sizing (7 x 6.5 x 1.8 mm^3), and packaging (SOT-223) characteristics as the SGM2212-3.3. PCB layout footprints were assigned to all components based on their corresponding datasheet sizing requirements.

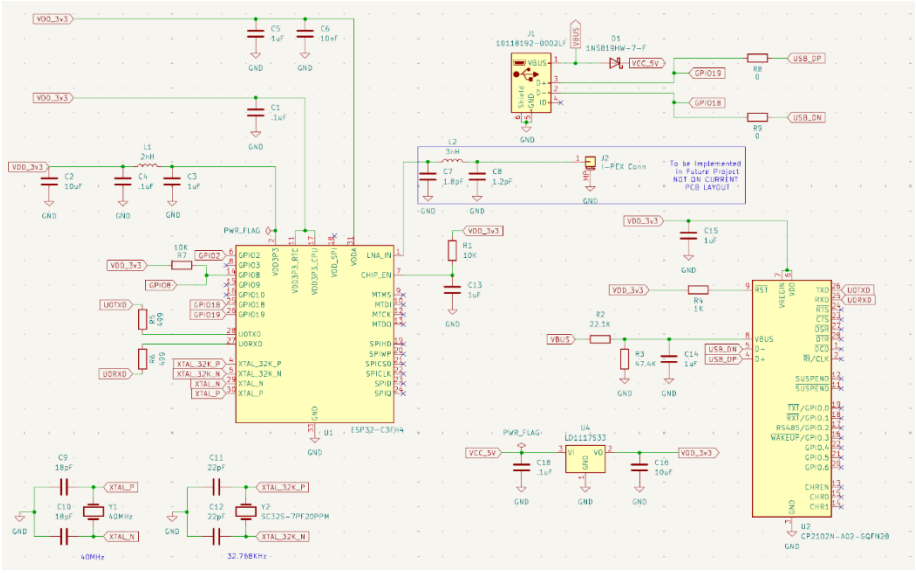

Figure 10: Custom MCU PCB Schematic

The 2-layer PCB layout, with connections made and footprints specified, is shown in Figure 4-11. The outer dimensions, like the optical detection PCB, are 20 mm. While this sizing is larger than the minimum sizing requirement, the increased size is designed to make soldering and testing feasible. PCB stencils and solder reflow are used to solder such a compact design. After communications with ETG, it was decided that such a compact, 2-layer PCB would not be solderable via a reflow oven with the technology available. Therefore, the 2-layer PCB was split into two separate, single-layer TEST MCU PCBs. Figures 4-12 & 4-13 show the two layers on their own respective PCBs. Figure 4-10 corresponds to the components placed on the top copper layer of the 2-layer MCU PCB design, while Figure 4-11 corresponds to the components placed on the bottom copper layer. These two single-layer PCBs allowed for the utilization of the available reflow oven and gave ample space for testing the custom MCU design.

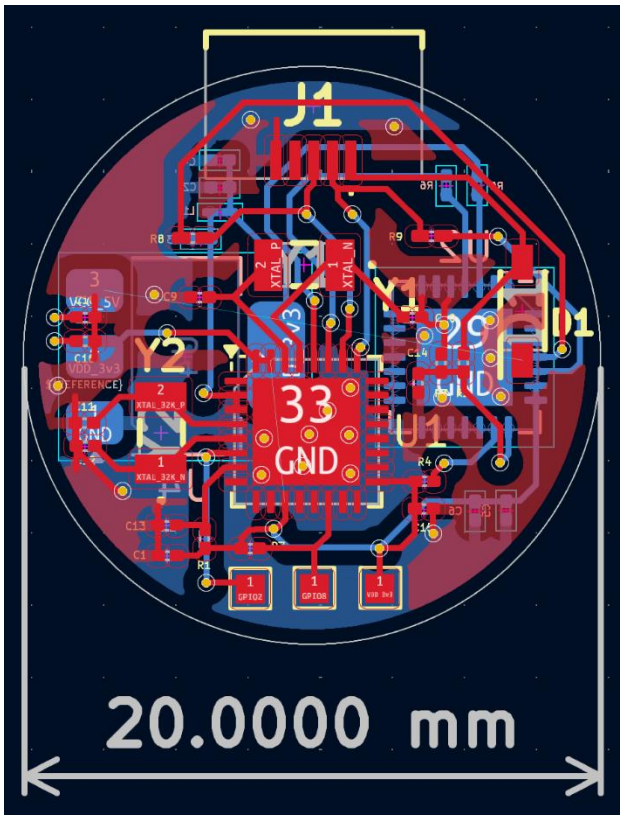

Figure 4-11: Custom 2 Layer MCU PCB Layout

As will be described in *Section 5: Testing*, the MCU PCB design did not function properly, as code was unable to be flashed onto the MCU. It is believed that this is due to the removal of the switching amplifiers for the boot and reset pins, as reading of available documentation suggests these are critical to the ability to flash code onto the MCU. Due to time constraints, an updated MCU PCB design was not developed. Further progress of the microbial pill sensor will require an updated MCU PCB design and validation through testing. Once validation of the MCU design has been completed, the PCB should be both manufactured and assembled by JLC PCB, who have the technology to reflow compact 2-layer PCBs.

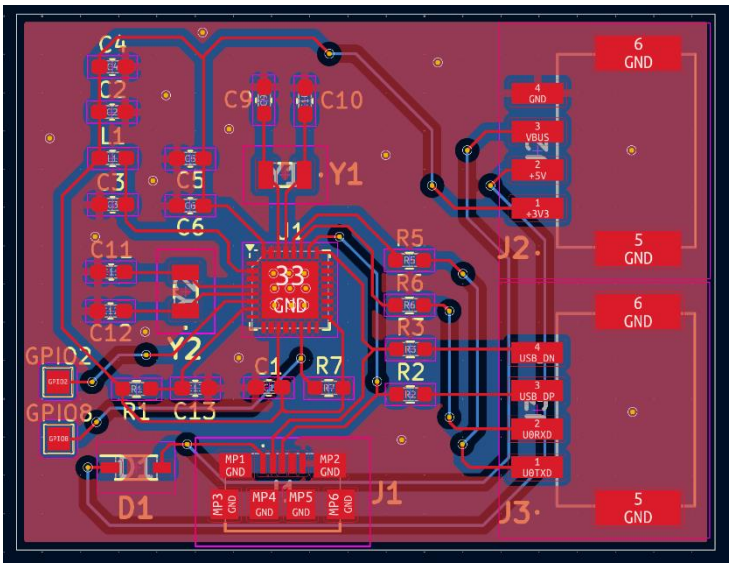

Figure 4-12: Single Layer Test MCU PCB (top copper)

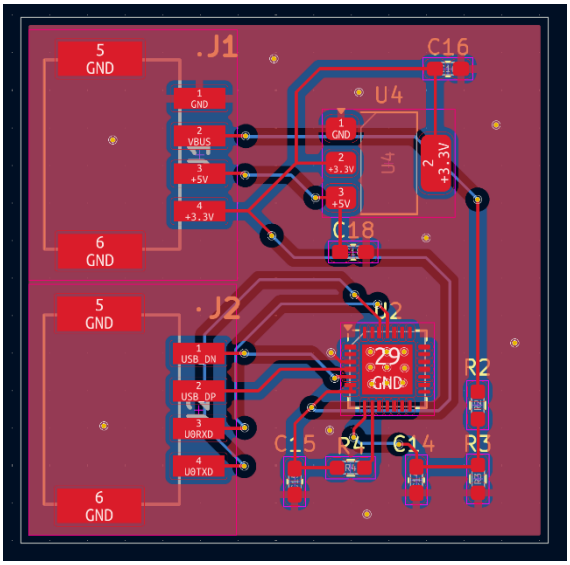

Figure 4-13: Single Layer Test MCU PCB (bottom copper)

Software and GUI Development

For the program development of the microbial pill sensor, both Arduino IDE and VSCode hosting Python are used. The Arduino IDE is used to program an MCU using a vast quantity of MCU chips, Devkit Boards, and libraries for specific MCUs and their functions. The vast array of resources for Arduino IDE has led to the programming being simplified down to a few key features. For the ESP32-C3 in the microbial pill sensor design, the main functions are to control the optical circuits, power on the excitation LED, collect an amplified response from the PD, and transfer the collected data via BLE to an external GUI application. The Adafruit NeoPixel library is used to operate the RGB LED. The BLEServer, BLEDevice, and BLEUtils libraries are used to create a BLE Server, register a device, and set custom UUIDs for the BLE connection. All other operations used in the program are methods developed in the Arduino Software. The current code functions on a ten-second loop that runs continuously to advertise the BLE connection. Once a connection has been made, the data collected at the end of the loop will be transmitted to the connected device and repeated. Future implementations of this program will require the BLE connection to be established before the optical components become operational in a way to minimize wasted power.

A Python GUI program is used to display and save recorded data in CSV format for end-user processing. The tkinter, matplotlib, and pytz libraries are used to create and operate the GUI. The bleak library is used to connect the program to the MCU's advertising signal. Both the Arduino IDE code for operation of the MCU and the Python GUI code can be seen in *Appendix 4: Code*.

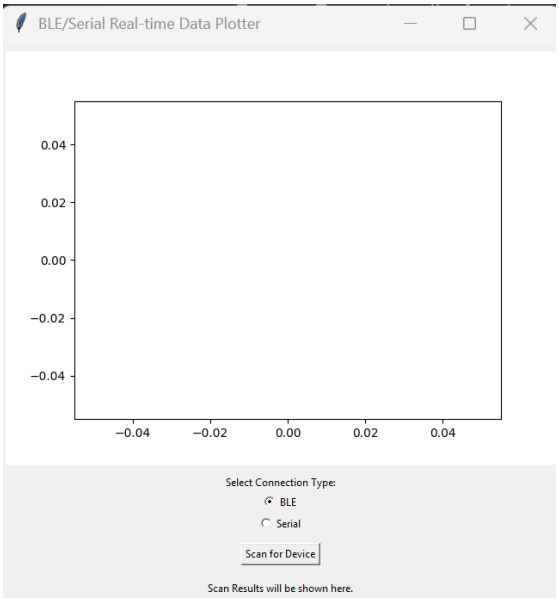

Figure 4-14: GUI Start Up Display

As a user begins to run the program, an interactive window opens, which allows a user to scan for BLE devices or select a Serial connection. Figure 4-14 shows a view of this window once it has been created. It is important to note that the next window will appear after the scan has been completed, so it is crucial to wait for this to be executed before clicking the scan for devices button again.

Once the scan is completed, a list of scanned devices opens in an additional window in which a user can select the specific device. Figure 4-15 (left) contains a screenshot of the detected devices window. The spaces with no titles are devices that are broadcasting a signal, but do not contain a specified name to connect to. The project's name is ESP32_ADC.

For the Serial connection selection, a similar process happens. A list of connected devices opens in an additional window in which a user can select the specific port that the device is connected to. Figure 4-15 (right) contains a screenshot of the detected ports window. If no devices are connected to the available ports, the window will remain blank with no COM ports showing. A user can detect which port they are using in their device manager.

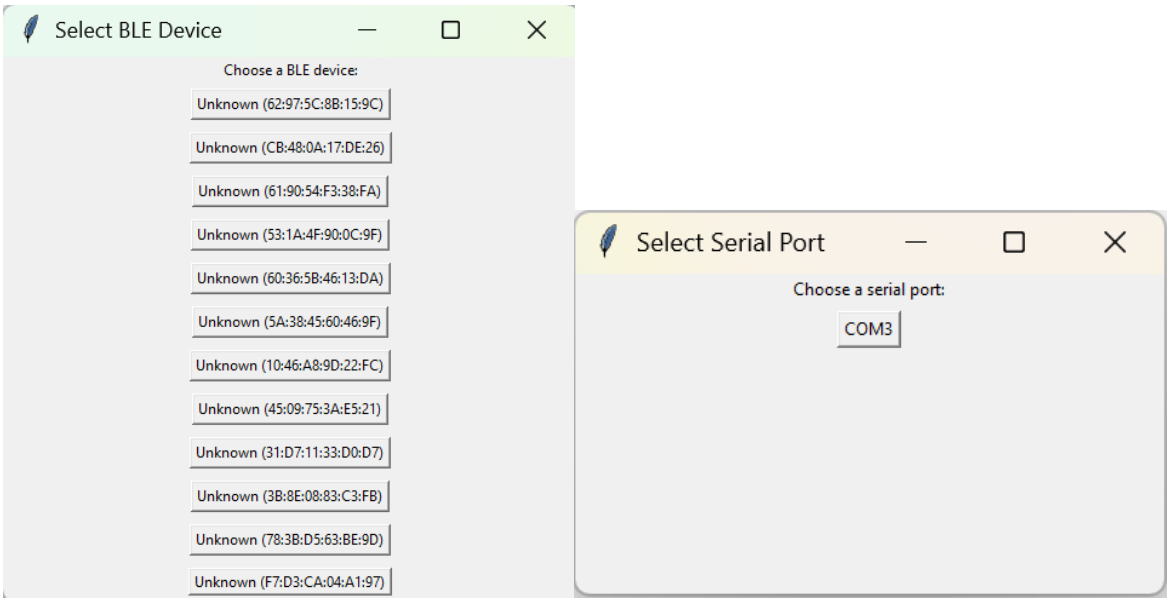

Figure 4-15: Select BLE Device Window and Select Serial Port Window

Upon establishing a connection, the original window changes into a time series of recorded amplified voltages. Figure 4-15 depicts a voltage response graph with four data points being recorded from the photodetector. This example was produced using the prototype discussed in *Section 6: Implementation*. The dips in recorded voltage correspond to the physical covering of the LED and a reduction in light intensity.

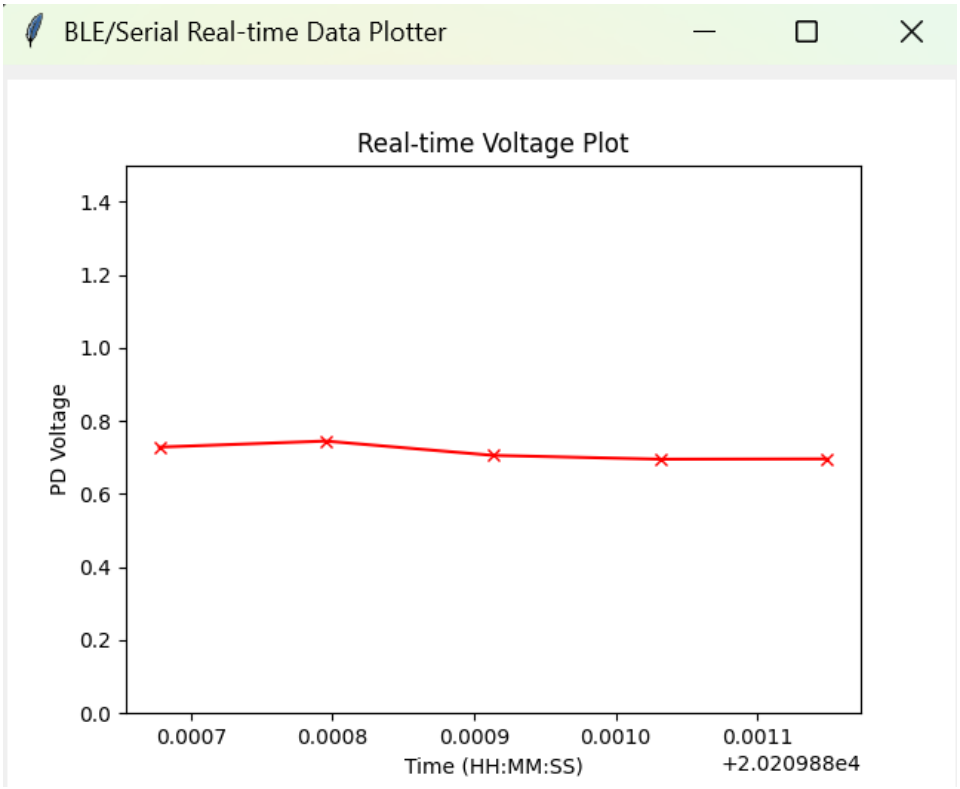

Figure 4-16: Real Time Voltage Graph 1

Battery Housing

Similarly to the temperature control PCB, the design and implementation of the battery housing is of low priority in the development of the microbial pill sensor. No work has been done regarding battery component selection or housing design. We have established that we want the battery component to be easily accessible and replaceable, continuing the life of the fabrication microbial pill sensor beyond the length of the battery. Ease of battery replacement will provide increased ease of use for our users, but little work has been done to implement this in the design. The physical design of this component will be incorporated into the overall exterior housing design, but consideration of the battery component and component protection will not occur until a functional prototype integrating the MCU and optical detection modules is demonstrated per discussion with the client.

Battery Housing and MCU Cell Housing

The battery housing compartment is specifically designed to securely hold the battery, which provides the necessary power for the MCU PCB and the optical PCB, thus ensuring consistent operation of the sensor. The dimensions for this compartment are set at 30 x 30 x 24 mm^3, offering ample space to accommodate a suitably sized battery while maintaining a compact overall structure.

To enhance user convenience and facilitate easy maintenance, the battery housing includes a detachable lid with dimensions of 21 x 21 x 6 mm^3 on one side. This removable cover simplifies battery replacement procedures, reducing downtime and making routine servicing straightforward and efficient.

Adjacent to the battery compartment is the MCU cell housing, specifically integrated into the same structural unit for optimal space utilization and improved device compactness. The MCU is strategically positioned opposite the battery compartment to minimize wiring complexity and ensure reliable electrical connectivity. This arrangement promotes an organized internal structure, improves durability, and simplifies the assembly process.

4.3.3 FUNCTIONALITY

The initial application for the microbial pill sensor is detecting levels of nitrate in agricultural field run-off to assess pesticide application and environmental impacts. The microbial pill sensor is intended to be submerged in agricultural run-off, in which the desired test solution will flow into the microbial pill sensor. The concentration of nitrate in the run-off will be detected by measuring the GFP fluorescent response of bioengineered microbes. A voltage value will be transmitted via BLE to an external platform where it will be converted to a concentration value and displayed via a GUI, allowing researchers, farmers, and other users to assess nitrate concentrations in agricultural run-off to better inform environmental and farming practices. Additional testing of the final device will establish reasonable lifetimes and transmission ranges of the final device, ultimately impacting the final functional interaction with the user.

Figure 4-17, shown below, depicts a storyboard depiction of the microbial pill sensor functionality with the user. Once placed, the microbial pill sensor functionality will be automated through the activation of the LED component at a specified timing interval and recording the produced voltage. The transmission and processing of the data will occur automatically and will populate the GUI with up-to-date, real-time concentration measurements.

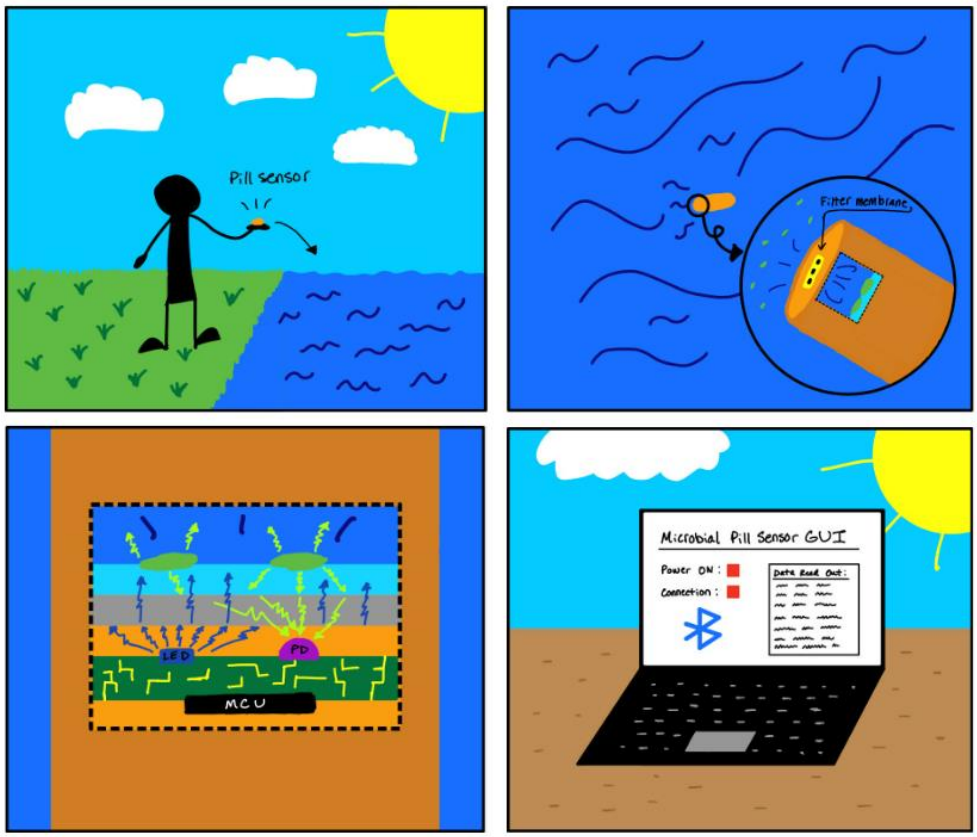

Figure 4-17: User Functionality Storyboard

4.3.4 Areas of Challenge

The microbial pill sensor is intended to provide competitive nitrate detection sensitivities to available marker nitrate detection technologies while using a novel microbe-based detection methodology in a small housing with an extended lifetime due to low power consumption. This is a novel technology, with many challenging areas of design, specifically in nitrate concentration detection at high sensitivities. The small design space of the microbial pill sensor, with the intended sizing requirement of 10 x 10 x 5 mm^3, greatly limits the applicability of advanced optical components or beam splitting technologies that may give rise to greater sensitivity. While decomposing the MCU itself was a challenging design task, laying out the MCU on a PCB board within the required size was incredibly challenging and may not even be feasible.

The sizing and economic constraints prevent the use of optical beam directing components, which force the detection and LED excitation to occur on the same PCB. This makes collecting the fluorescent response while avoiding the excitation LED incredibly challenging. The excitation wavelength, 488 nm, and the emitted wavelength, 525 nm, are relatively close, which makes detecting one while ignoring the other a challenging task for many economically sensible photodiode components. Additionally, the challenge of properly dispersing the blue excitation light across the entire housing chamber was not addressed and seems to have a negative impact on the functionality of the device.

Further development of the microbial sensor will require further improvement of the uniformity of the LED dispersion in the housing chamber and better capturing of the emitted response. The current challenge is only microbes situated near the excitation light get excited, meaning that little of the emitted response can reach the photodetector. Addressing this challenge, when paired with the functional transmission and photodiode detection that has already been achieved, will lead to a functional microbial pill sensor.

4.4 Technology Considerations

In designing the microbial pill sensor, several key technologies were selected to effectively meet the project's functional and design needs. Fusion 360 was chosen for its robust 3D modeling capabilities and cloud-based accessibility, allowing seamless collaboration across the team. This software enables easy import of PCB designs from KiCad, simplifying the integration of electronic and physical design stages. Additionally, Fusion 360 supports individual part design and assembly, making it ideal for creating and refining the modular components in the pill sensor. This cloud-based tool also facilitates the transition to 3D printing, supporting the development of the pill's physical structure. While AutoCAD is an alternative option, Fusion 360's integration with KiCad and user-friendly interface make it a more practical choice for this project.

KiCad was selected for designing the PCB layouts and schematics, mainly due to the team's prior experience and the open-source nature of the software. KiCad offers simplified PCB design workflows, extensive online resources, and handy tutorials for prototyping with the ESP32 C3 microcontroller. Additionally, KiCad's compatibility with Fusion 360 supports an efficient transfer of designs between PCB and physical model development. While KiCad lacks some advanced features in commercial PCB software, its open-source accessibility and robust community support make it a strong choice for the project's current needs. Fusion 360 could be a backup option for basic circuit sketches, but it does not offer the specialized electronic design tools available in KiCad.

The team selected Arduino IDE to program the ESP32 C3 microcontroller and manage Bluetooth data transmission. This open-source platform is known for its ease of use and strong community support, making it ideal for handling the sensor's BLE transmission and real-time data monitoring. Although Arduino IDE requires hard wiring of components, which can introduce connection issues that are

sometimes challenging to debug, its simplicity aligns with the project's focus on creating a reliable yet straightforward device. PlatformIO was considered an alternative due to its sophisticated debugging capabilities, but its added complexity was deemed unnecessary for this project phase.

For the graphical user interface (GUI), PySimpleGUI was selected for its ease of use and quick learning curve. This library enables the rapid development of a functional interface for displaying real-time data from the sensor. With minimal coding requirements, PySimpleGUI provides an accessible solution for creating an intuitive and responsive GUI. However, it does have limitations in customization and scalability, which might affect its viability in a more complex or polished application. PyQt was considered as an alternative due to its extensive widget options and customization capabilities. However, its steeper learning curve and potential licensing requirements for commercial applications made PySimpleGUI the preferred choice for this project phase.

For long-term data storage, CSV file creation was selected for the simplicity of transporting the data between different applications like Google Sheets and Microsoft Excel. With the ability to store data from each test, the ability to further record data and understand trends helps with the ability to test PD sensitivity, lensing, and filtering specifications, and microbial response emission trends.

.

5 Testing

5.1 Unit Testing

As previously noted in section *4: Design*, the microbial pill sensor has been broken down into simplified modules. These modules stand as the simplest systems needed to operate functionally and within requirements before the full microbial pill sensor can be integrated and operated. Each module, or unit, was tested individually prior to integration, except for the exterior housing.

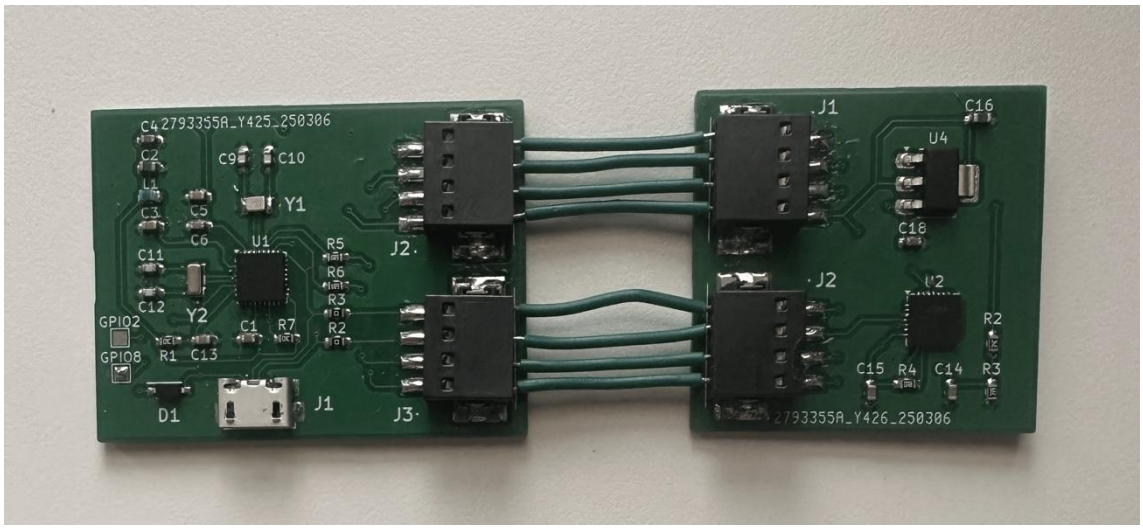

Figure 5-1: Assembled Test MCU PCB Module

Upon fabrication and assembly of the TEST MCU PCB module via soldering stencil and reflow oven, verification of its electronic functionality was conducted. To do so, continuity tests were done at extremely close solder points and across the board connectors to ensure no unexpected shorts or opens were present on the board. Next, a micro-USB cable was plugged into component J1, seen in Figure 5-1, supplying the boards with power from a laptop. A voltmeter was then used to ensure that power was being supplied to the correct places around the boards, as well as to ensure that there was a reliable 3.3V output by the voltage regulator component U4. Finally, to test the functionality of the TEST MCU module, the already verified GUI code was run. During MCU initialization, an error message occurred, signaling that something was wrong with the MCU circuit design.

After conducting further research into how loading code onto the Espressif ESP32-C3 Dev Kit functions, it was determined that the issue was the decision to leave out the switching amplifiers for the boot and reset pins in the design of the custom MCU PCBs, which are critical in the ability to flash code to the MCU. Recall *Section 4.3.2: Detailed Design*. As previously discussed, no further development, design updates, or further testing of the MCU PCB was conducted due to time limitations. The Espressif ESP32-C3 Dev Kit, used in breadboard prototyping, was used for all MCU device functionality throughout the rest of the testing and the implementation of the microbial pill sensor. See *section 6: Implementation* for further information on the utilization of the Espressif ESP32-C3 Dev Kit in the final design.

Similar continuity tests were conducted on the optical detection PCB. Breadboard measurements via an ammeter were conducted to verify that the voltage produced by the photodiode increased with light intensity shining on the photodiode. Breadboard measurements via an ammeter were also used to verify the proper amplification of the signal via the LM324 op amp included on the PCB. It was found that the op amp

saturated at an output voltage of 2.049 V when using a power supply voltage of 3.3 V. While this is a significantly lower saturated voltage than expected, this is not a concern for the microbial pill sensor as the output voltage is in the range of ~ 10 mV prior to amplification. Testing to verify the function of the photodetection PCB was quite straightforward. More complex testing tasks centered around investigating resistance and circuit configurations to improve device performance, as discussed in *Section 4.3.2: Detailed Design*.

As previously mentioned, with the MCU PCB board nonfunctional, the Espressif ESP32-C3 Dev Kit was used to implement MCU functionality. Arduino IDE debugging and verification features were used to verify the function of the MCU code. As the MCU's function within the device is interfacing between the different modules, further discussion of the MCU testing will occur in *Section 5.2: Interface Testing*. Visual Studio Code debugging and verification features were used to verify the function of the Python GUI code. Test data was generated within the program and was successfully plotted, verifying the ability of the Python GUI code to plot and save data to the desired CSV file. Testing of transmitted data will be discussed in *Section 5.2: Interface Testing*.

5.2 Interface Testing

The functional role of the ESP32-C3 MCU is to provide the user with an interface to perform and view the measurements made by the optical detection PCB. The MCU provides a 3.3 V supply to the optical detection PCB for both the LED and the operational amplifier on the board and controls the activation of the LED on the board. The MCU also collects the generated photovoltage and transmits it to the external device for display via BLE. The General-Purpose Input/Output pins were designated for these specific Power Supply, UART Transfer and Receive, and ADC Channel functions. These GPIO functionalities were implemented in the Arduino IDE via referencing of the associated component datasheets. With the MCU PCB being nonfunctional, interface testing was conducted using the Espressif ESP32-C3 Dev Kit. Further development of the MCU PCB, and thus the microbial pill sensor, will require the reperformance of interface testing on the functional MCU PCB once developed.

The optical detection PCB was connected to the Espressif ESP32 C3 Dev Kit on a breadboard via wires soldered to the jumper pads on the optical detection PCB. Connectivity tests were used to verify that the power supply and GPIO port for controlling the programmable LED were properly reaching the optical detection PCB. Code was developed and executed, which demonstrated the ability to control the timing, color, and intensity of the programmable LED via the Arduino IDE. The 12-bit ADC channel was tested using known, fixed voltages coming from a Lab Power Source. The Arduino IDE was used to print the voltage value coming from the ADC channel to the serial monitor. Direct comparison of the ADC value to the known supplied voltage was used to calibrate the ADC channel properly across a range of voltages. It was found that the ADC channel worked for higher voltages, but the reliability broke down at lower voltages in the range of a few or tens of millivolts, where the impact of noise on the ADC's accuracy was substantially increased.

In addition to the GPIO Ports of the MCU, another interface that must operate successfully for the project to work is the MCU BLE connection to the Python Processing and GUI. This system relies on both the programming of the MCU through the Arduino IDE Programming Software to be correct in having the ESP32-C3 MCU advertise for a connection and having the Python Script run a detection for BLE devices through the bleak and asyncio libraries in Python. Testing for this interface relied heavily on reviewing errors within each program, ensuring the libraries needed are properly installed and called. Test data was generated within the Arduino IDE, which was successfully transmitted, displayed in the GUI, and saved to a CSV file. Direct comparison across a variety of different test datasets was used to ensure proper

transmission, display, and saving of the test data. The entire system interface, from the ADC channel to the Python GUI, was tested using known voltage values fed to the ADC from a power supply. Demonstration of successful BLE transmission from the ADC input verified the successful interfacing of the various modules. The same process of using test datasets within the program and a known voltage supply was used to demonstrate proper serial connection between the MCU and an external device. Recall that both serial connection and BLE are available for data transmission. *Section 5.4: System Testing* will discuss complications arising from using the output voltages from the photodetection circuit rather than known voltages from a power supply.

Understanding the limitations of BLE is vital for the real-world application of the project. To test interferences and the range of the device's limits, range tests were conducted. These tests slowly extended the range between the MCU and a connected device until the connection was severed and data could no longer be transmitted successfully. The range was established to be 44 meters in air. It is likely that concrete buildings played a role in limiting transmission range in these tests. Further testing will be needed to determine the range of the device when submerged.

5.3 Integration Testing

The integration of the different software, the Arduino IDE code, and the Python GUI code was previously described in *Section 5.2: Interface Testing*. The integration of the various PCBs and optical components, such as the lens and filters, will be described here. Integration of these components required ensuring that these components fit and stayed in the housing, were not damaged by the housing, and were functional in the housing. As has been previously discussed, the MCU PCB could not be integrated into the housing as it was not fabricated. Instead, integration testing consisted of using the Espressif ESP32-C3 Dev Kit with wired connections to the optical PCB and a direct serial connection to an external device for power supply. The battery connection was not tested. The housing was left open where the MCU PCB was designed to sit to allow for the jumper wires connecting the photodetection PCB to the breadboard with the MCU. Further development and fabrication of the MCU PCB will enable integration of the entire system and testing of the closed housing capsule.

As previously discussed, the microbial pill sensor capsule was fabricated both via on-campus 3D printing resources and CNC machining. The CAD model of the housing was routinely updated until all components, including the photodetection PCB and the excitation and emission filters, were able to fit in their desired positions within the housing. The lens was unable to be properly positioned within the housing unless the housing was situated upside down, where gravity was used to hold the lens in its desired position. The housing design will need further development to ensure the lens is properly positioned in the housing. As has been discussed, the current sizing of the housing does not meet the minimum sizing requirement to enable easier testing of the system. As future development shrinks PCB components, the housing will also shrink. Integration testing of the fit of the various components will once again need to be conducted. *Section 5.4: System Testing* will discuss the functionality of the device following integration of the distinct components.

5.4 System Testing

The system was tested through the integration of all available modules into a final implemented system. As discussed previously, the Espressif ESP32-C3 Dev Kit was used to implement the non-functional MCU PCB. The Espressif ESP32-C3 Dev Kit was connected to the photodetection PCB via jumper wires soldered to the PCB, and a 3D printed stand was built to provide room for the wires under the housing. All optical components were placed in their designated positions in the housing. The system was tested in two stages, one where a programmable LED was used to mimic the microbes and another where tests were performed with live microbes.

When using the programmable LED to mimic the sample, the programmable LED, emitting green 500 nm light, was positioned at the microbe housing level. The brightness setting for the LED was variably increased to characterize the response of the system. The lens and filters were removed to test their impact on device performance. The results of this device characterization are presented in *Section 5.9 Results*. Figure 5-2 depicts a photo of this testing setup, with the programmable LED mimicking the sample positioned at the microbe level. The programmable LED mimicking the sample is operated by the MCU through GPIO pin 6.

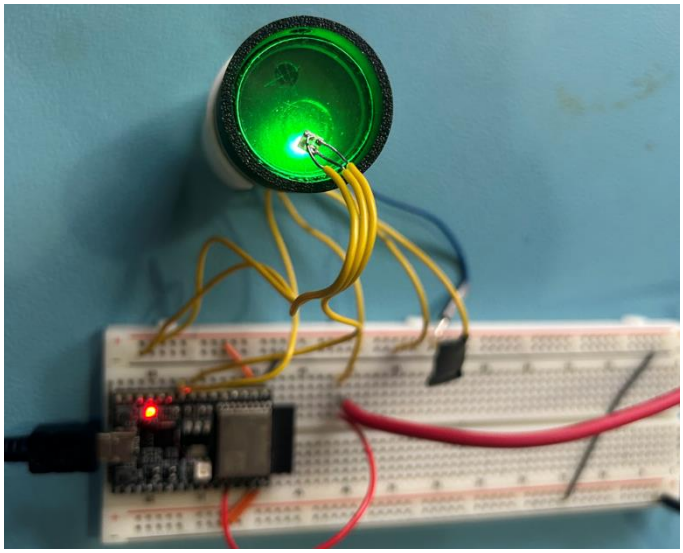

Figure 5-2: System Testing Setup

Following characterization of the system performance, live microbes contained in beads in a solution were placed on the microbe cover slip in the housing. The programmable LED on the photodetection PCB was activated for 10 seconds and then turned off for 10 seconds to excite and measure the fluorescent response of the microbes. Testing the device's performance with live microbes was used to characterize the functional detection ability of the device.

In both tests, the activation of the programmable LED on the photodetection board was controlled by the Espressif ESP32-C3 Dev Kit. The photogenerated voltages from the photodiode were amplified and collected by the ADC channel of the MCU Dev Kit and transmitted to the external device running the Python GUI. Both serial and BLE transmission were tested. The Python GUI was used to display the data and save it to a CSV file on the external device. An ammeter was plugged into the breadboard and used to compare with the measurements collected and transmitted by the MCU.

Further discussion of the results of system testing will be discussed in *Section 5.9 Results* and *Section 6: Implementation*.

5.5 Regression Testing

As the project has progressed, the team has opted to work in both an agile and waterfall management style. By breaking down the project into modules, each member has been tasked with creating a different component of the project. Throughout this process, collaboration has allowed for mostly seamless integration of these different modules. The separation of the device into distinct modules has largely limited the need for regression testing across modules due to their independence. Changes to a module did not impact another, except for the housing, which made progressing on each module individually possible.

Both the Arduino IDE and Python GUI were critical features for the transmission, display, and saving of collected measurements. Both features were independent of other developments in the housing or photodetection PCB, which prevented any breaking of their functionality. The photodetection PCB was also critical to the function of the device, but its performance was not impacted by the developments in the Arduino IDE, MCU, or Python GUI. The housing was built to fit around the fabricated photodetection PCB, as we were not able to order another PCB within our time constraints. The housing had to be routinely updated to ensure functionality with the photodetection PCB and optical components, but this was possible due to the fast turnaround time of 3D printing.

5.6 Acceptance Testing

The demonstration of design requirements was conducted through system testing, described in *Section 5.4: System Testing*. System testing demonstrated transmission, display, and recorded of the generated photocurrents within the housing, which depicts many of the functional design requirements of the device. System testing also allowed for characterization of the device performance, which will be depicted in *Section 5.9: Results.* As will be discussed in *Section 5.9: Results,* failure to progress to functional environment testing prevented the ability to test many device requirements, especially related to environment and user interfacing. Many requirements were not met or were unable to be tested due to the progression of the project. Recorded measurements pertaining to the microbial pill sensor's response were shared with the client to demonstrate the device's performance with respect to functional requirements. As further development of the microbial pill sensor ensues, further demonstration of requirements will be required as they become possible to test.

5.7 User Testing

The development of the microbial pill sensor did not progress to a point where rigorous testing of the microbial pill sensor's ability to meet user needs was conducted. Involvement with users consisted of interaction with our client, demonstrating the functionality of the device, interacting with the housing, and reviewing the GUI display. The design of the GUI display and recording of data in CSV file format was directly coordinated with the client, which ensures the current design meets the user's software needs. The Python GUI enables the user to set the activation frequency of the LED, preventing the need for the user to interact with the code on the MCU.

Users will interact with the microbial pill sensor through the housing to insert microbes and remove and replace the device's battery. The current housing design has a removable cap for the battery, which we concluded meets user accessibility needs. However, the current housing design does not include a way to place microbes in the device, which does not address user needs. Further development of the housing design is needed to ensure it meets user needs.

5.8 Results

As has been discussed throughout this document, the microbial pill sensor benefited from a modular design, where progress and development of the MCU PCB, photodetection PCB, housing chamber, and software development occurred independently. Through unit testing, see *Section 5.1: Unit Testing,* it was demonstrated that the code cannot be flashed on the MCU PCB. As discussed, it is believed that the source of this error is due to the removal of the push buttons and switching amplifiers when designing the MCU PCB. The design of the MCU PCB was not updated due to time constraints. The Espressif ESP32-C3 Dev Kit, used for breadboard prototyping in the beginning development stages of the microbial pill sensor, was used as a substitution for the MCU to test other system and interface functionalities. Unit testing revealed that the photodetection PCB could produce a measurable photogenerated voltage in response to increases in light intensity.

Interface and unit testing showed the MCU Dev Kit was capable of transmitting data from the ADC channel to an external device via either BLE or serial transmission, and the external GUI application could receive the transmitted data, displaying the data as a function of time, and saving the data timeseries in a CSV file. However, interface testing between the photodetection PCB and the ADC channel of the Espressif ESP32-C3 Dev Kit showed that the ADC channel was not able to reliably measure the photogenerated voltages. Testing with an ammeter was used to verify the discrepancy between the real and ADC recorded values. Figure 5-3 shows the GUI display for the photogenerated voltages recorded from the photodetection PCB using the MCU ADC. The errors in the data collection due to the significant impact of noise at low voltage levels are clearly visualized. Further investigation of the root cause of this error is required, but the initial belief is that a mismatch in input impedance between the ADC and the photodiode circuit could be the cause, due to the proper function of the ADC channel when using known voltages from a power supply.

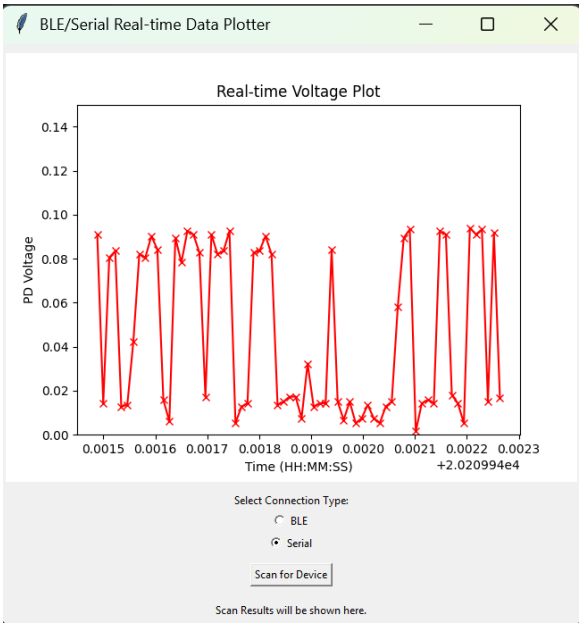

Figure 5-3: Real Time Voltage Plot from System Testing

The integration of the excitation filter, emission filter, and the photodetection PCB in the CNC-fabricated housing demonstrated that they were correctly positioned and held within the housing. However, the housing does not correctly hold the lens as the lens is not able to be properly fixed in its designated position. System testing demonstrated that the lens had zero impact on the performance of the photodetection. The setup for testing is depicted in Figure 5-2, see *Section 5.4: System Testing.* The photodetection capability of the system, with the photodetection PCB encompassed in the CNC-fabricated housing, is depicted in Figure 5-4. The system was tested with and without the optical components. Figure 5-4 shows that the device demonstrates zero sensitivity to 500 nm light when the filter and lens are included and demonstrates a sensitivity of 52.02 mV / mW for 500 nm light when the excitation filter, emission filter, and lens are not included. The negative impact of the optical filters demonstrates an error in the component choice of the programmable LED, which produces green light with a peak intensity at 500 nm and very little intensity at our desired 525 nm wavelength. The 500 nm wavelength produced by this LED is not outside the pass region of our emission filter, which explains the zero sensitivity to 500 nm light.

As previously described, the system testing demonstrated the ability of the MCU to transmit the data by either BLE or Serial transmission, and the ability of the external GUI program to plot it, as shown in Figure 5-3. However, as discussed, issues with the ADC prevent the transmission of accurate photogenerated voltage readings. Additional system-level testing was conducted with live GFP-expressing microbes, where the microbes were excited with blue light from the programmable LED. Testing of the microbes suggests that the emitted light intensity is ~ 2 μW when exposed to 460 nm light, which is likely below the limit of detection of our system. The system was unable to produce a noticeable response to the GFP-expressing microbes, which could be caused by several identified design flaws. As previously discussed, the programmable LED is a poor choice for the excitation of the expressed GFP due to its peak emitted wavelength intensity being centered around 460 nm, with a low intensity of 488 nm light emitted. This limits the fluorescent response of the microbes, which reduces the ability of the photodetection PCB to generate a noticeable signal. The addition of a functional lensing system would also lead to an increased signal strength that could potentially allow the system to detect the microbe's fluorescent response.

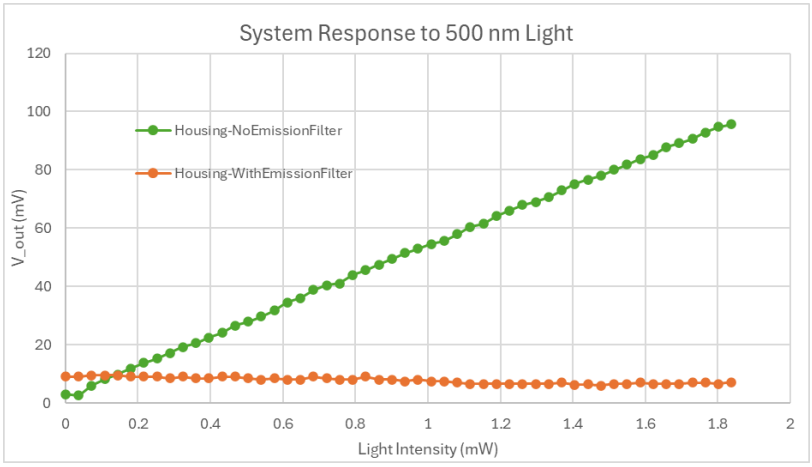

Figure 5-4: System Response to 500 nm light

Through testing, it became apparent that the external Python GUI program does a good job of meeting user needs. Through a user-friendly interface, the ability to control the activation frequency of the LED, plot real-time measurements, and save recorded measurements in a CSV file, all while using publicly available libraries and packages, the external GUI does a tremendous job of meeting user needs. The housing design needs considerable updating, introducing a removable top to insert the microbes and improving the lens positioning, to meet user needs.

The developed microbial pill sensor fails to address many crucial functional and non-functional requirements. Requirement F.3, requiring the system to produce a measurable fluorescent response from GFP-expressing microbes, along with requirement F.4, requiring the photocurrent to be proportional to the amount of analyte concentration, were not met as demonstrated by system testing. While these requirements are perhaps the most crucial for the microbial pill sensor to be functional, significant progress was made towards developing a system capable of generating a voltage proportional to the intensity of green light as seen by the system response in Figure 5-4. Additionally, functional requirements and user experience requirements related to housing the microbe, activation of the LED, transmitting data, and displaying data via an external GUI are all satisfied by the current implementation of the microbial pill sensor. The further development of a functional MCU PCB, along with an improved housing design and improved ADC sampling, is necessary to further the progress of the microbial pill sensor.

6 Implementation

6.1 Design Analysis

As discussed previously, the microbial pill sensor is not currently a finished or entirely functional product. The current implementation of the microbial pill sensor consists of a photodetection PCB contained within a housing chamber, along with functioning optical filters positioned in the housing. Due to issues with uploading code to the custom MCU PCB, the devkit module is being used to run the product to overcome this issue. The design that has been created works well as an initial stepping stone for understanding the complications of using optical components. The design served as an understanding of how these components can be used, and in new developments, updated designs can focus on how the components should be used to achieve the best efficiency.

The final implementation of this design incorporates the following components:

- The Optical PCB for microbe excitation and emission collection and 10x gain amplification.
- Excitation (488nm) and emission (525nm) wavelength optical filters.
- ESP32-C3 Dev Kit in place of custom MCU PCB.
- CNC housing with issues regarding microbe excitation and emission collection.
- GUI that allows for data to be transmitted and displayed to the user via BLE and serial connection.
- Inability to sample measured data via Dev Kit MCU ADC.

The final implementation of the microbial pill sensor can detect green light, transmitting a photogenerated voltage value, and displaying the recorded data via an external GUI. Understanding the limitations of specific aspects and using engineering methods to make improvements is key to creating a functional product and learning important skills. The use of low-energy data transfer works to accurately send data while minimizing power usage. Although the MCU PCB does not work, the optical PCB works. The LED is used to excite the microbes that produce an emission, and the PD accurately records the response. Although the final housing is much larger than the original design, it works to hold all the components together and allows for the system to work together.

7 Ethics and Professional Responsibility

7.1 Areas of Professional Responsibility/Codes of Ethics

Area of Responsibility	Definition	Relevant IEEE Code Clause	Team Interaction
Work Competence	Performing tasks with expertise and diligence to ensure high-quality results.	To undertake only those tasks for which they are qualified	Team members focus on their areas of expertise, such as PCB design or GUI development, while learning new tools (e.g., Fusion 360, Arduino IDE) to maintain technical proficiency and high project standards.
Health, Safety, and Welfare	Safeguarding the well-being of all stakeholders and the environment.	To hold paramount the safety, health, and welfare of the public.	The team has prioritized rigorous testing in controlled environments to ensure the microbial containment chamber functions effectively reducing ecological or health risks during use or testing.
Communication Honesty	Transparent and truthful reporting of progress and findings.	To be honest and realistic in stating claims or estimates.	Challenges in size constraints and material fabrication have been openly communicated with the advisor, enabling transparent progress tracking and iterative design improvements.
Sustainability	Ensuring the environmental friendliness and longevity of designs	To hold paramount the safety, health, and welfare of the public	Environmentally safe materials have been selected for housing, with continuous efforts to minimize ecological impact through design refinements, such as reusable components and sustainable packaging.
Social Responsibility	Contributing to societal and environmental improvements.	To improve the understanding of technology and its applications.	By addressing nitrate pollution in water, the project supports sustainable agriculture and ecosystem health, benefitting farmers, environmentalists, and broader communities reliant on clean water systems.

Table 7-1: Areas of Professional Responsibility

Throughout the process of developing the microbial pill sensor, the team has demonstrated strengths and areas for improvement in adhering to various professional responsibilities. The most notable strength of the team is Communication Honesty, exemplified by our commitment to transparency in addressing challenges encountered during the project. Numerous times, the team faced challenges or failures when testing a portion of the design. Poor component choices, such as the decision to use the SK16MINI RGB LED, design decisions, removing the push buttons from the MCU PCB, and failure to adhere to the established project timeline for the development of the housing were all communicated immediately and openly. By being transparent about challenges facing the team or the project, the team was able to openly discuss resolutions and next steps, which were able to keep the project progressing. The team was also honest with our client/faculty advisor about which desired device functionalities were feasible. Data display and GPS tracking through Google Colabs was a desired feature of the device at one point, but after investigation and attempts proved it would not be possible while maintaining BLE transmission, the inability to integrate these desired functionalities was openly communicated to our client.

While the team succeeded in communication honesty, the team has not performed well in the area of sustainability. The microbial pill sensor has a direct impact on environmental health, and while the team aims to make a sustainable and environmentally friendly device, no work has been actively done to ensure

the sustainability of the device. While the team has not actively made design or testing decisions to harm our responsibilities in sustainability, no work has been actively conducted to promote the sustainability of the microbial pill sensor. Further development of the microbial pill sensor will require a direct focus on sustainability and the environmental impact of the device.

7.2 Four Principles

Table 7-2 assesses the four principles regarding the microbial pill sensor and its public health, global, environmental, and economic impacts. The four principles assessed are beneficence, nonmaleficence, respect for autonomy, and justice. The broader context-principal pair of Environmental and Beneficence is incredibly important to our project, as it provides the entire motivation for the microbial pill sensor. Our biosensor is intended to improve environmental outcomes related to excessive nitrate runoff in waterways, providing a continuous monitoring system for early detection of hazardous conditions or inefficient fertilization processes. If the microbial pill sensor does not provide beneficence in environmental outcomes, our product has failed in its intended functionality.

The microbial pill sensor lacks consideration of factors in application within the broader context of Environmental and Nonmaleficence. While the microbial pill biosensor is not intended to obstruct natural environmental processes, interfere with natural ecosystems, or harm the environment or its inhabitants, no thought has been dedicated to the application of the biosensor in the environment. A potential issue facing the microbial pill sensor is animal consumption of the biosensor, as little thought has been placed into how the biosensor will be fixed in a dedicated position. Displacement of the microbial pill sensor could generate misleading readings, which could prompt environmentally disrupting action. As the project enters a more functional and final phase, engineering considerations will be applied to the application of the biosensor in the intended environment, alleviating any potential unintended environmental maleficence from the displacement of the biosensor.

	Beneficence	Nonmaleficence	Respect for Autonomy	Justice
Public health, safety, and welfare	Enhances public safety by creating a biosensing system that monitors health indicators and contaminants effectively.	Avoids harm by ensuring safe application of bioengineered bacteria and preventing misuse.	Provides accurate, accessible health data for informed decisions by users.	Ensures equal access to the technology for all communities regardless of socio-economic status.
Global, cultural, and social	Promotes better global health outcomes by detecting pollutants and diseases in diverse contexts.	Avoids harm by considering cultural sensitivities in system design and deployment.	Respects cultural autonomy by tailoring features to community-specific needs.	Distributes benefits equitably, ensuring marginalized groups also gain from the innovation.
Environmental	Reduces environmental risks by detecting harmful pollutants quickly and efficiently.	Prevents ecological harm through sustainable material sourcing and manufacturing.	Respects environmental autonomy by avoiding interference with natural ecosystems.	Balances environmental benefits across regions and protects resources for future generations.
Economic	Reduces health monitoring costs through efficient and affordable technology.	Avoids financial harm by ensuring affordability and transparency in costs for users.	Empowers users to make cost-effective decisions with reliable, accessible information.	Balances economic benefits for all stakeholders, including underprivileged communities.

Table 7-2: Four Principles

7.3 VIRTUES

With the application of the microbial pill sensor in improving environmental outcomes, our team has prioritized the virtue of commitment to the public good. Our device is intended to provide paths to improving the quality of public waterways, so design decisions or practices that result in damage to the public good fundamentally contradict the motivation for the project. The team has prioritized our commitment to the public good by ensuring the product is environmentally friendly. Concerns regarding contamination of the environment through poor membrane and material selection will be prioritized by the team as the design of the individual modules is finalized. In tandem with a prioritization of commitment to the public good is a commitment to quality. The team strongly believes that our microbial pill sensor has a strong application in improving continuous monitoring and biosensing technologies. Producing a low product quality damages the applicability of our device. Through our product research, we have established performance metrics that the final microbial pill sensor should reach to serve as a competitive biosensing solution in the market. As discussed, the metrics have not yet been met, but we believe the foundation has been laid so that further development of the microbial pill sensor will meet these metrics. Ensuring our product exceeds these benchmarks through extensive testing and strategic design decisions ensures the quality of the microbial pill sensor. Our team also prioritized clear and thorough documentation through the weekly reports, project scheduling, and tracking of accomplishments through the team channel. Producing a high-quality version of this design document provides thorough documentation of the design, ensuring our team is reaching our dedication to documentation through the development of the microbial pill sensor.

Cade Kuennen:

> Throughout my senior design work, I have consistently demonstrated a strong commitment to quality. This virtue is important to me because I believe that quality reflects the integrity of both the individual and the work itself. Delivering high-quality results ensures the solution we provide is reliable, effective, and impactful for stakeholders. I have exemplified this commitment by carefully reviewing project deliverables, ensuring designs meet both functional and aesthetic standards, and incorporating feedback from team members and mentors. For example, when completing the parts decomposition of the MCU dev kit, one of the potential dev kits had poor documentation. Instead of guessing what components were missing, I talked with my team and advisor and was able to find a dev kit with better documentation that could also work for the project. By prioritizing quality, I aim to not only meet but exceed project expectations. This dedication strengthens our team's outcomes and prepares us for professional challenges.
>
> Cooperation is a virtue I deeply value, although I feel I could have made a more active effort applying this virtue in the project. Effective collaboration is essential in any team-based effort because it fosters open communication, ensures diverse perspectives are considered, and strengthens the overall quality of the final product. While I have contributed individually and engaged in discussions in this project, I see overall room for improvement in proactively seeking input from teammates and understanding the inner workings of their individual sprint portions of a project. In future projects, I plan to demonstrate greater cooperation and become more familiar with the development process of my team members' portions of the project. By enhancing my cooperative approach, I can help create a more unified team dynamic and contribute to a stronger, more cohesive project result.

Rakesh Varma Penmetsa:

From the beginning of my senior design project—developing a microbial pill sensor—I have consistently demonstrated strong dedication to detail, adaptability, and a persistent commitment to high-quality outcomes. These qualities have guided my work in creating accurate and reliable designs that meet stringent project requirements. In my role focusing on the CAD modeling and physical housing design, I have extensively used Fusion 360 for creating precise 3D models, effectively translating conceptual layouts into practical, manufacturable designs. Initially, the compact dimensions chosen for the microbial housing chamber and other modules led to significant prototyping challenges, especially when using filament-based 3D printing. These early prototypes suffered from insufficient precision, structural weaknesses, and functional inadequacies. Recognizing these limitations, I proactively collaborated with our advisor and team members to iteratively refine the housing design.

While my work thus far has demonstrated consistent adaptability and meticulous attention to detail, I recognize the importance of enhancing my cooperative skills within the team. Cooperation is crucial, as it promotes effective communication, leverages diverse perspectives, and leads to superior project outcomes. Although I engaged collaboratively during critical redesign phases, I realize I have not sufficiently explored other team members' technical responsibilities—such as PCB layout, microcontroller programming, or sensor calibration. Moving forward, I plan to actively foster stronger collaboration by engaging regularly with teammates on their respective modules. Specifically, I will proactively seek feedback, offer my assistance during testing and debugging, and take greater initiative to understand and support their technical challenges. By further strengthening my cooperation skills, I aim to contribute to a more cohesive and successful overall project. Through this project, I have developed and applied several critical engineering skills, including detailed 3D CAD modeling (Fusion 360), iterative design methodologies, expertise in rapid prototyping (3D printing and CNC machining), component integration, problem-solving, and effective communication. By continuing to refine these skills, particularly emphasizing cooperation and interdisciplinary teamwork, I seek to enhance my personal growth and contribute meaningfully to the project's successful completion.

Wes Ryley:

A virtue that I believe to be crucial to a successful group project is the need for cooperation and clear documentation. Since our project has been broken down into separate modules, with each team member taking a specific area of focus, being able to communicate effectively and allow collaboration to help design the functions of the project has helped simplify the process and improve our team's efficiency. As the Data Transmission Design Lead, I've been tasked with programming both the MCU and creating the Python code that does the system's processing and GUI. Although my work doesn't interact with the PCB design or housing modules, through collaboration, our team has been able to verify that the software that has been developed will work in combination with the custom PCB boards.

One virtue that I feel I haven't been as accepting of during this process is the virtue of liberality. Since the project is linear in nature, having the PCBs collect data, have that data be sent via BLE, and having a program process and display the data, the need to be open to interesting and new ideas has been limited. Although I don't directly think this is a negative virtue, I believe that being focused on specific aspects of the project rather than trying to constantly innovate sporadically has been beneficial for me and the rest of the team. If I were to think back and apply this virtue to the project, I believe that handling tasks and issues that arose during debugging and testing could've been prevented or remediated quickly.

Alex Upah:

Honesty is an incredibly important virtue in ensuring the quality completion of a major project. Failing to be honest with teammates and myself regarding accomplishments, setbacks, or the quality of the work and documentation provided is damaging to the successful completion of the project. I have demonstrated honesty through the revision of my own work in the system's design, the design document, and my documentation procedure. I also believe I have provided honest and fair criticism, when necessary, regarding the work of my teammates.

Attentiveness to both your own work and the work of your teammates is a valuable virtue in the completion of a major project. The agile workflow implemented by the team this semester has resulted in individuals spending most of their time working on their own individual components. With separation from the design of the other modules, I did not focus on setbacks that occurred in the housing design that should have been prioritized. Ultimately, failure to stay on track schedule-wise with the housing was fatal to the development of a functional microbial pill sensor.

8 Closing Material

8.1 Conclusion

Our microbial pill sensor aims to develop a novel biosensing technology for the detection of nitrate in agricultural field runoff using bioengineered microbes. The microbial pill sensor uses the fluorescent response of GFP expressed by the bioengineered microbes in the presence of nitrate to produce a photocurrent proportional to the concentration of nitrate. The photocurrent value is transmitted to an external device via Bluetooth low energy, processed, and displayed by our external GUI application. Our design is modular, broken into an external housing module, an optical detection PCB module, a microcontroller PCB module, and an external GUI application.

Initial designs for the photodetection PCB, MCU PCB, and housing capsule have been completed, fabricated, and tested. All three modules exceed the current minimum sizing requirements in order to enable extensive and accessible testing of functionality. The housing capsule properly contains the fabricated PCBs, enabling the proper connections between the PCBs, as well as the optical emission and excitation filter. The housing capsule is currently unable to properly position the lensing component. The photodetection PCB, using the EOPD-525-1-0.9-1 photodiode, produces a proportional voltage response to 500 nm light, with a sensitivity of 52.02 mV / mW and a detection limit of 3 mV. The fabricated MCU PCBs were nonfunctional due to an issue in the decomposition of the ESP32-C3 MCU, but the ESP32-C3-Dev-Kit provided a functional replacement for the MCU functionality to enable system testing until the MCU PCB is functional. System-level testing, while unable to demonstrate proper detection of fluorescent response from GFP-expressing microbes, was able to characterize the response of the system to green 500 nm light as a function of light intensity.

Using the ESP32-C3 Dev Kit, activation of the excitation LED at the desired timing, along with Bluetooth low-energy and Serial transmission of photogenerated voltages, has been achieved through the Arduino IDE platform. An external Python GUI has been developed that plots the received data in real-time and saves the recorded data in a CSV file for further processing and analysis by the user. System testing demonstrated the ability of the housing to facilitate light detection and optical filtering.

Development of the microbial pill sensor, while not complete, was able to reach major milestones in BLE transmission, GUI development, and a high-sensitivity photodetection module. While the major, but lofty, goal of being able to detect nitrate concentration through GFP fluorescent response was not achieved, significant progress has been made towards the development of a novel microbial pill sensor for nitrate detection.

8.2 Value Provided

The current microbial pill sensor is nonfunctional and does not facilitate the detection of nitrate in agricultural runoff. However, significant progress has been made towards the development of the final device. The current implementation serves as a capable detector of green light and has functional data transmission and display. The external GUI is designed to handle all user needs through an easy-to-use interface for collecting and recording data.

The microbial pill sensor still needs extensive development to function as an economically competitive nitrate detection technology. The development of the project so far maintains that the devices fit in the broader context of biosensing technologies for environmental protection.

8.3 Next Steps

Future development of the microbial pill sensor will be passed to either a future senior design team or a graduate student in Dr. Lu's lab. Future development of the microbial pill sensor will focus on developing a functional MCU PCB, improving the sensitivity of the device through updating the housing design, and performing functional environment testing of the device.

The MCU PCB is crucial for the ability of the device to transmit measured voltage values to the external GUI application and is thus crucial to the development of the functional microbial pill sensor. While the current implementation of the microbial pill sensor uses the ESP32-C3-Dev-Kit to handle transmission responsibilities, future implementations must use a functional MCU PCB capable of data transmission while fitting with the housing chamber. Future development of the MCU PCB must fix errors regarding flashing of the MCU code through the inclusion of the boot and reset pins and necessary switching amplifiers. Additionally, a matching network for the antenna must be properly developed to enable BLE transmission to the external GUI.

Future development of the housing should aim to improve the ability of the device to detect the fluorescent response of GFP-expressing microbes by increasing the intensity of emitted light reaching the photodetection PCB. The photodetection PCB has been demonstrated to have a sufficient detection sensitivity with a limit of detection of 100 μW. The addition of a lensing system, along with better dispersion of excitation light throughout the microbe housing chamber, should be able to increase the intensity of the fluorescent response beyond the detection limit. The programmable LED used on the current photodetector PCB should be updated to a more suitable LED with a peak wavelength centered around the excitation wavelength of 488 nm. Additionally, future work should aim to minimize the size of the housing and PCBs to meet the established minimum sizing requirement. Future development should also consider the feasibility of this sizing requirement.

Lastly, future development of the microbial pill sensor will have to answer challenges regarding functional environment testing, examining device lifetime, transmission range, and environmental impact. While many further developments are required to reach a functional microbial pill sensor, a strong foundation has been laid by the work described in this design document that enables further development of a functional implementation.

9 References

[1] Inda-Webb, M.E., Jimenez, M., Liu, Q. et al. Sub-1.4 cm3 capsule for detecting labile inflammatory biomarkers in situ. Nature 620, 386–392 (2023). https://doi.org/10.1038/s41586-023-06369-x.

[2] Clear Water Sensors, "Chemical Sensor Series: Precision submersible chemical sensors. Nitrate Sensors," *Clear Water Sensors,* 2024. [Accessed Dec. 2, 2024]

[3] DeAngelis KM, Ji P, Firestone MK, Lindow SE. Two novel bacterial biosensors for detection of nitrate availability in the rhizosphere. Appl Environ Microbiol. 2005 Dec;71(12):8537-47. doi: 10.1128/AEM.71.12.8537-8547.2005. PMID: 16332845; PMCID: PMC1317476.

[4] Espressif Systems, "ESP32-C3 Series Datasheet v2.0," ESP32-C3 datasheet, 2024.

[5] EPIGAP OSA, "EOPD-525-1-0.9 Specifications", EOPD-525-1-0.9 datasheet, Aug. 2023.

[6] Dongguang OPSCO Optoelectronics CO., LTD, "SK6812MINI: Integrated Light Source Intelligent Control of Chip-On-Top SMD Type LED," SK6812MINI datasheet, Sept. 2015.

[7] Espressif Systems, "ESP Hardware Design Guidelines: ESP32-C3 Schematic Checklist" *Espressif Systems,* 2024. [Accessed Dec. 2, 2024]

10 Appendices

Appendix 1 – Operation Manual

Python Program Operation Manual:

Begin by installing Python; this can be done by navigating to the Python website and installing the latest version of Python 3.x. An important step during installation is to check the box for "Add Python to PATH." The next step is to install an operating system for Python; the project has been developed and tested on VSCode. To install VSCode, navigate to their website and install it for the specific operating system that the device is. After VSCode has been installed, open the software and install the Python extension, which can be found in the extensions tab. The next step involves creating a Python Environment; this can be done in VSCode by creating a folder and saving the Python program inside this folder. Verify this step by attempting to open the program through VSCode. The last step before being able to run the program is to ensure the necessary libraries are installed on the device. This can be done by opening the terminal in VSCode and running the following command: "pip install matplotlib pytz bleak pyserial." This will install all the necessary libraries. When the program runs, a window will open allowing the user to input their operating conditions. Select either BLE for wireless data transfer or Serial for USB-connected data transfer. Click on the "Scan for Device" button. Depending on the connection type selected, a new window will open. For the BLE connection, a list of all nearby BLE devices will open; select the ESP32_ADC device to connect to the designated device. For the Serial connection, a list of COM ports that are available will open, select the one the device is connected to. If the COM port is unknown, a user can open their device's device manager to check which COM port the device is connected to. After device selection, the program will run. The program will receive the ADC data, plot the voltage in real time, and save the data to the created CSV file, which can be found in the folder that was created earlier.

Arduino IDE Program Operation Manual:

Begin by installing the Arduino IDE software, which can be done on their website. Select the correct software for the operating system. Once installed, open the software. Navigate to the File tab, and in preferences, a user can install specific board models. In the additional boards manager URL, add "https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json" and then click OK. Go to Tools, Board, and Board Manager, search for the ESP32, and install ESP32 by Espressif Systems. Next is to install the required libraries. Go to Sketch, Include Library, and Manage Library, once here install the following libraries: Adafruit NeoPixel by Adafruit and ESP32 BLE Arduino by Neil Kolban. Once these libraries are installed, connect the device to the operating device, select the ESP32 board, and correct the COM Port the device is connected to and upload the sketch to the device. The device is now operating.

Appendix 2 – Alternate/Initial Version of Design

Initial versions of the microbial pill sensor design considered the addition of a temperature control module, designed to keep the housing at the desired temperature to maintain the microbe cultures in a functional environment. Due to the complexity of the temperature control module, along with the prioritization of the different described functionalities, the inclusion of temperature control module was dropped following discussion with the client.

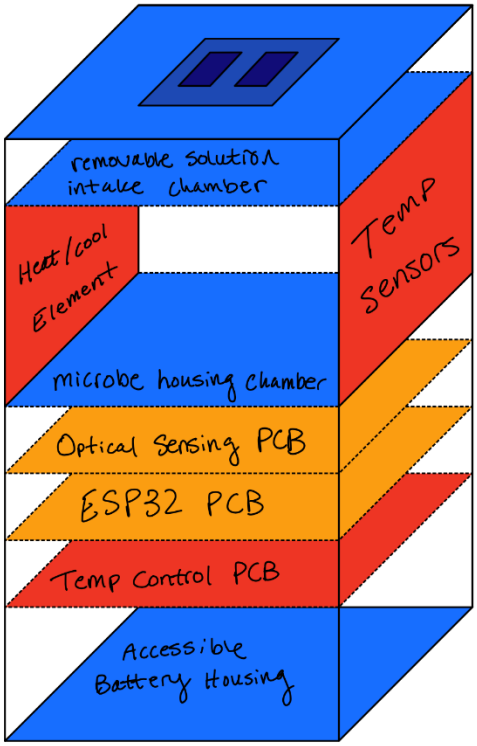

Figure A2-1: Initial 3D Design

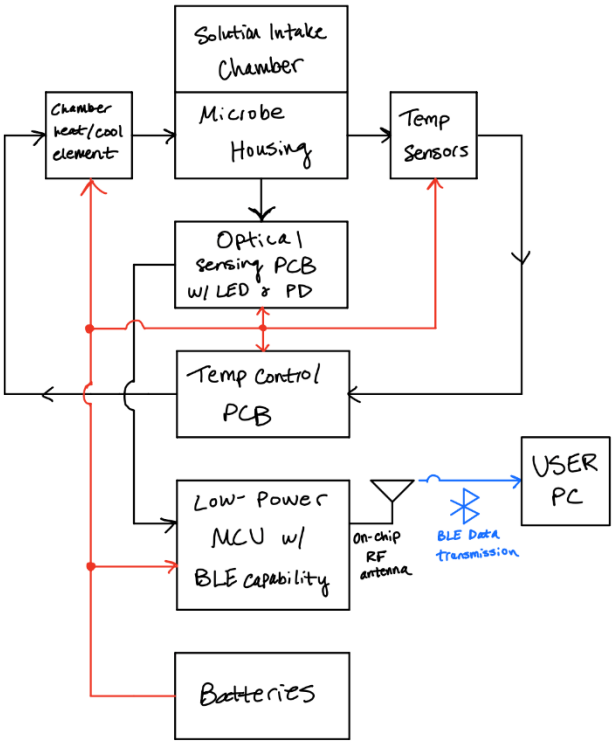

Figure A2-2: Initial System Sketch

Appendix 3 – Other Considerations

During the development of the Python Program, there was time spent looking into the possibility of using Google Colab as the operating software. Google Colab is a cloud-based environment which is tied into the Google functionality which creates easy access to Google Drive files and Google Sheets. In addition, the software has many useful widgets and libraries which can be used to create a simple GUI and easy data storage. During the designing of the new program many issues arose. Since it is a cloud-based environment, connecting input through BLE or Serial connection was incredibly demanding, required additional complications like hosting a server to connect the devices, and was scratch off as an idea due to the immense complications that were being found. However, the design process was not a waste of time as it gave the team insight as to how the CSV file should be created and operate with the program.

Appendix 4 – Code

Python Program:

```
import asyncio

import threading

import tkinter as tk

from tkinter import messagebox

import matplotlib.pyplot as plt

from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

from datetime import datetime

import pytz

from bleak import BleakScanner, BleakClient

from bleak.exc import BleakError

import serial

import serial.tools.list_ports

import csv

import os


# UUIDs for the ESP32-C3 device

SERVICE_UUID = "6E400001-B5A3-F393-E0A9-E50E24DCCA9E"

RX_UUID = "6E400002-B5A3-F393-E0A9-E50E24DCCA9E"

TX_UUID = "6E400003-B5A3-F393-E0A9-E50E24DCCA9E"


# Global data lists

ADC_ArrayValue = []

AmpVoltage_ArrayValue = []

timestamps = []


# Create main window
```

```
root = tk.Tk()
root.title("BLE/Serial Real-time Data Plotter")
root.geometry("650x650")

# Central Standard Timezone
CST = pytz.timezone("US/Central")

# Voltage conversion (NEED TO CONFIRM TO MATCH THE ARDUINO PROGRAM)
def convert_to_voltage(adc_value):
    return (adc_value * 3.3) / 4095 * (3.08/3.3)

#---------------- CSV FILE AREA ------------------------------------------------#
CSV_FILE_NAME = "data_log.csv"

# Initialize the file with headers if not already present
def initialize_csv():
    if not os.path.exists(CSV_FILE_NAME):
        with open(CSV_FILE_NAME, mode='w', newline='') as file:
            writer = csv.writer(file)
            writer.writerow(["Timestamp", "ADC Value", "Voltage"])

def log_to_csv(timestamp, adc_value, voltage):
    timestamp_str = timestamp.strftime("%Y-%m-%d %H:%M:%S")  # Format as YYYY-MM-DD HH:MM:SS

    # Check if file exists and is not empty
    file_exists = os.path.isfile(CSV_FILE_NAME)

    if file_exists:
```

```
        # Read the current CSV to check where the last line is
        with open(CSV_FILE_NAME, mode='r', newline='') as file:
            reader = csv.reader(file)
            rows = list(reader)

            # Find the first empty line
            empty_row_index = len(rows)
    else:
        # If file does not exist, create a new one and add headers
        with open(CSV_FILE_NAME, mode='w', newline='') as file:
            writer = csv.writer(file)
            writer.writerow(["Timestamp", "ADC Value", "Voltage"])  # Write headers
        empty_row_index = 0  # Start from the first line

    # Write the new data to the first empty line
    with open(CSV_FILE_NAME, mode='a', newline='') as file:
        writer = csv.writer(file)

        # If the file isn't empty, make sure to skip the header row
        if empty_row_index > 0:
            writer.writerow([timestamp_str, adc_value, voltage])



#-------------------------GRAPH AREA -----------------------------------------#
# Persistent figure and axes
fig, ax = plt.subplots()
line, = ax.plot([], [], marker='x', color='r', label="Voltage Values")
canvas = FigureCanvasTkAgg(fig, master=root)
```

```
canvas.get_tk_widget().pack(pady=10)


def update_graph():
    if not AmpVoltage_ArrayValue:
        return

    # Cap to 100 points
    max_points = 100
    if len(AmpVoltage_ArrayValue) > max_points:
        ADC_ArrayValue.pop(0)
        AmpVoltage_ArrayValue.pop(0)
        timestamps.pop(0)

    line.set_data(timestamps, AmpVoltage_ArrayValue)
    ax.relim()
    ax.autoscale_view()
    ax.set_ylim(0, .15)
    ax.set_xlabel("Time (HH:MM:SS)")
    ax.set_ylabel("PD Voltage")
    ax.set_title("Real-time Voltage Plot")
    canvas.draw()



# BLE data handler
async def receive_ble_data(client):
    try:
        while True:
            data = await client.read_gatt_char(TX_UUID)
```

```
            data_str = data.decode('utf-8')
            try:
                adc_value = int(data_str)
                ADC_ArrayValue.append(adc_value)
                voltage = convert_to_voltage(adc_value)
                AmpVoltage_ArrayValue.append(voltage)
                timestamps.append(get_cst_time())
                log_to_csv(get_cst_time(), adc_value, voltage)
                root.after(0, update_graph)
            except ValueError:
                print(f"Invalid BLE data: {data_str}")
            await asyncio.sleep(10)
    except BleakError as e:
        print(f"BLE error: {e}")

def start_asyncio_loop(device_address):
    loop = asyncio.new_event_loop()
    asyncio.set_event_loop(loop)
    async def main():
        async with BleakClient(device_address) as client:
            if client.is_connected:
                print(f"Connected to {device_address}")
                await receive_ble_data(client)
            else:
                print("Failed to connect")
    try:
        loop.run_until_complete(main())
    except Exception as e:
```

```
        print(f"BLE loop error: {e}")


def get_cst_time():
    now_utc = datetime.now(pytz.utc)
    now_cst = now_utc.astimezone(CST)
    return now_cst  # Return datetime object, not string



# Serial data handler
def start_serial_reading(port_name):
    def read_from_serial():
        try:
            with serial.Serial(port_name, 115200, timeout=1) as ser:
                print(f"Connected to serial port: {port_name}")
                while True:
                    if ser.in_waiting > 0:
                        line = ser.readline().decode('utf-8').strip()
                        print(f"Serial Read: {line}")
                        try:
                            adc_value = int(line)
                            ADC_ArrayValue.append(adc_value)
                            voltage = convert_to_voltage(adc_value)
                            AmpVoltage_ArrayValue.append(voltage)
                            timestamps.append(get_cst_time())
                            log_to_csv(get_cst_time(), adc_value, voltage)
                            root.after(0, update_graph)
                        except ValueError:
                            print(f"Ignored invalid serial data: {line}")
```

```
        except serial.SerialException as e:
            print(f"Serial error: {e}")
    threading.Thread(target=read_from_serial, daemon=True).start()

# BLE device scanning
async def scan_devices():
    devices = await BleakScanner.discover()
    return devices

def on_device_select(device, window):
    window.destroy()
    threading.Thread(target=start_asyncio_loop, args=(device.address,), daemon=True).start()

def on_serial_select(port_name, window):
    window.destroy()
    start_serial_reading(port_name)

# BLE or Serial scan handler
def scan_button_clicked(label, button):
    mode = connection_type_var.get()
    if mode == "BLE":
        def background_scan():
            devices = asyncio.run(scan_devices())
            if devices:
                def show_ble_window(devices):
                    win = tk.Toplevel(root)
                    win.title("Select BLE Device")
                    tk.Label(win, text="Choose a BLE device:").pack()
```

```
                for dev in devices:
                    name = dev.name or "Unknown"
                    tk.Button(win, text=f"{name} ({dev.address})",
                              command=lambda d=dev: on_device_select(d, win)).pack(pady=5)
            root.after(0, show_ble_window, devices)
        threading.Thread(target=background_scan, daemon=True).start()
    elif mode == "Serial":
        def scan_serial_ports():
            ports = list(serial.tools.list_ports.comports())
            if ports:
                def show_serial_window(ports):
                    win = tk.Toplevel(root)
                    win.title("Select Serial Port")
                    tk.Label(win, text="Choose a serial port:").pack()
                    for port in ports:
                        tk.Button(win, text=str(port.device),
                                  command=lambda p=port.device: on_serial_select(p, win)).pack(pady=5)
                root.after(0, show_serial_window, ports)
        threading.Thread(target=scan_serial_ports, daemon=True).start()

# Mode selector
connection_type_var = tk.StringVar(value="BLE")
tk.Label(root, text="Select Connection Type:").pack()
tk.Radiobutton(root, text="BLE", variable=connection_type_var, value="BLE").pack()
tk.Radiobutton(root, text="Serial", variable=connection_type_var, value="Serial").pack()

# Scan button and result label
```

```
scan_button = tk.Button(root, text="Scan for Device", command=lambda:
scan_button_clicked(scan_results_label, scan_button))

scan_button.pack(pady=10)


scan_results_label = tk.Label(root, text="Scan Results will be shown here.")

scan_results_label.pack(pady=10)


# Initialize CSV file

initialize_csv()


# Tkinter main loop

root.mainloop()
```

Arduino IDE Program:

```
#include <Arduino.h>

#include <Adafruit_NeoPixel.h>

#include <BLEDevice.h>

#include <BLEServer.h>

#include <BLEUtils.h>

#include <BLE2902.h>


#define PIN 6       // ESP32-C3 built-in RGB LED pin

#define PIN_7 7     // External RGB LED pin

#define NUMPIXELS 1  // Number of LEDs

#define ADC_PIN 2    // ADC Pin for reading voltage

#define LED_POWER


Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);  // Built-in RGB LED
```

```
Adafruit_NeoPixel externalPixels(NUMPIXELS, PIN_7, NEO_GRB + NEO_KHZ800);  // External RGB LED
#define DELAYVAL 500

// BLE UUIDs for UART service
constexpr char SERVICE_UUID[] = "6E400001-B5A3-F393-E0A9-E50E24DCCA9E";
constexpr char CHARACTERISTIC_UUID_RX[] = "6E400002-B5A3-F393-E0A9-E50E24DCCA9E";
constexpr char CHARACTERISTIC_UUID_TX[] = "6E400003-B5A3-F393-E0A9-E50E24DCCA9E";

BLECharacteristic* pCharacteristicTX = nullptr;

bool deviceConnected = false;
unsigned long lastNotificationTime = 0;
constexpr unsigned long NOTIFICATION_INTERVAL = 500;
int bright = 63;//NEW FOR TESTING

// BLE Server Callbacks
class MyServerCallbacks : public BLEServerCallbacks {
  void onConnect(BLEServer* pServer) override {
    deviceConnected = true;
    Serial.println("Device connected");
  }

  void onDisconnect(BLEServer* pServer) override {
    deviceConnected = false;
    Serial.println("Device disconnected");
    pServer->getAdvertising()->start();
    Serial.println("Advertising restarted");
  }
```

```
};

// Setup BLE and start advertising
void setupBLE() {
  BLEDevice::init("ESP32_ADC");
  BLEServer* pServer = BLEDevice::createServer();
  pServer->setCallbacks(new MyServerCallbacks());

  BLEService* pService = pServer->createService(SERVICE_UUID);

  // Create RX Characteristic (optional, unused here)
  BLECharacteristic* pCharacteristicRX = pService->createCharacteristic(
    CHARACTERISTIC_UUID_RX,
    BLECharacteristic::PROPERTY_WRITE | BLECharacteristic::PROPERTY_WRITE_NR
  );
  pCharacteristicRX->addDescriptor(new BLE2902());

  // Create TX Characteristic
  pCharacteristicTX = pService->createCharacteristic(
    CHARACTERISTIC_UUID_TX,
    BLECharacteristic::PROPERTY_NOTIFY
  );
  pCharacteristicTX->addDescriptor(new BLE2902());

  pService->start();
  pServer->getAdvertising()->start();
  Serial.println("Advertising started");
}
```

```
void sendVoltData() {
  if (deviceConnected) {
    // Read the ADC value from the specified pin
    int adcValue = analogRead(ADC_PIN);
    // Format the ADC value into a string for transmission
    char dataToSend[20];
    snprintf(dataToSend, sizeof(dataToSend), "%d", adcValue);
    // Send the ADC value via BLE notification
    pCharacteristicTX->setValue(dataToSend);
    pCharacteristicTX->notify();
    // Debugging output to serial
    Serial.println(dataToSend);
  }
}

void setColor2(uint8_t red, uint8_t green, uint8_t blue) {
  // Set color for built-in LED
  pixels.setPixelColor(0, pixels.Color(red, green, blue));
  pixels.show();
}
void setColor(uint8_t red, uint8_t green, uint8_t blue) {
  // Set color for external LED
  externalPixels.setPixelColor(0, externalPixels.Color(red, green, blue));
  externalPixels.show();
}

void setup() {
```

```
    Serial.begin(115200);

    setupBLE();  // Initialize BLE communication
    analogReadResolution(12);  // Set ADC resolution (optional, default is 12-bit)

    pixels.begin();
    pixels.clear();
    pixels.setBrightness(0);    // Set brightness for internal LED

    setColor2(0,0,255);

     externalPixels.begin();
     externalPixels.clear();
     externalPixels.setBrightness(63);  // Set brightness for external LED

    setColor(0, 0, 255);
}

void loop() {
    static unsigned long lastSendTime = 0;
    static unsigned long lastBrightTime = 0; // NEW FOR TESTING
    unsigned long currentMillis = millis();

    if (currentMillis - lastSendTime >= 1000) {  // Every 1 seconds
        lastSendTime = currentMillis;



        lastBrightTime = currentMillis;
```

```
      bright = bright + 5;

      externalPixels.setBrightness(bright);
      externalPixels.show();



      // Read the ADC value from the specified pin
      int adcValue = analogRead(ADC_PIN);

      // Format the ADC value into a string
      char dataToSend[20];
      snprintf(dataToSend, sizeof(dataToSend), "%d", adcValue);

      // Send via Serial
      Serial.println(dataToSend);

      // Send via BLE if connected
      if (deviceConnected) {
         pCharacteristicTX->setValue(dataToSend);
         pCharacteristicTX->notify();
      }
   }
}
```

Appendix 5 – Team Contract

Team Members:

1) Cade Kuennen 2) Rakesh Penmetsa

3) Wes Ryley 4) Alexander Upah

Team Procedures

1. Day, time, and location

- The team will meet weekly on Wednesdays from 12 – 1 pm in TLA or Lab 1125.
- The team will meet weekly with Dr. Lu on Mondays from 12 – 1 pm in a reserved conference room.
- Additional meetings may be scheduled through additional weekly communication.

2. Preferred method of communication updates, reminders, issues, and scheduling

- The team will use Microsoft Teams and Discord to communicate.
- Meetings will take place face-to-face, but virtual meetings are acceptable provided a valid reason for failure to make meeting in person.
- A team schedule has been created containing each individual's schedule, which serves as a tool for scheduling additional meetings outside of arranged weekly meeting times.

3. Decision-making policy (e.g., consensus, majority vote):
   - The team will use a majority rules policy when making decisions.
   - Split decisions will be decided by the individual responsible for the module/component.

4. Procedures for record-keeping

- The team will use shared documents saved to a team One Drive to record and archive meeting minutes.
- Each individual has edit and read privileges for every document within the One Drive space.

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:

- Team members must attend all agreed-upon meeting times on time and actively participate in them unless a valid reason for absence is presented.
- Should a team member not be able to attend a team meeting, the team member should notify the team of their absence as soon as possible.

2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:

- All team members are expected to fulfill communicated and agreed-upon commitments and responsibilities on various assignments and deadlines.
- If external events present reasonable challenges to fulfilling assignments, timelines, or deadlines, the team member is expected to notify the team as soon as possible or at least three days before the deadline.

3. Expected level of communication with other team members:

- Team members are expected to check Teams or Discord once a day for updates.
- Team members will send updates on project work if any major milestones are met or if completed work assists another team member in the completion of their responsibilities.
- Team members will send updates on project work if any major setbacks occur.

4. Expected level of commitment to team decisions and tasks:
    - All decisions agreed upon by the team are final unless a change in the course of action is decided by the team.
    - All team members are expected to fulfill assigned responsibilities to the best of their abilities.

Leadership

1. Leadership roles for each team member

- Cade Kuennen: PCB Design Lead
- Rakesh Penmetsa: Capsule Housing Lead
- Wes Ryley: MCU and Data Transmission Lead
- Alex Upah: Optical Detection Lead

2. Strategies for supporting and guiding the work of all team members:

- If a team member needs assistance or guidance, additional time outside of set meetings may be scheduled to reach set milestones.
- Team members will provide collaborative support when necessary.
- Dr. Lu will provide available assistance and support when the team faces challenges.

3. Strategies for recognizing the contributions of all team members:

- Team members will show approval and acknowledgment and ask questions regarding the independent work of individuals.
- Communication of completion of major milestones provides a route to recognition of contributions of members.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the

team.

- Cade Kuennen:
    - Experience in PCB design for low and high-frequency systems
    - Experience thinking about design from an end-user-oriented perspective
- Rakesh Penmetsa:
    - Experience in circuit design
    - Experience using cadence software
- Wes Ryley
    - Understanding of power usage to help preserve battery life and potential heating issues
    - Understanding of data transfer methods, specifically through Wi-Fi
- Alex Upah
    - Academic coursework in biosensing, bio-detection mechanisms, and related biosensing data analysis
    - Academic coursework in optical components such as photodiodes
    - Knowledge of signals and communication of modulated signals
    - Extensive experience in a research environment working directly with faculty members.

2. Strategies for encouraging and supporting contributions and ideas from all team members:

- Each team meeting will involve a discussion of current progress and a discussion of future tasks and steps forward.
- Discussion of future steps forward provides opportunities for contributions and ideas of all team members to be heard and supported.

3. Procedures for identifying and resolving collaboration or inclusion issues

- The team has agreed upon an open communication policy. If a team member sees an issue with how the team is operating, whether it is obstructing or limiting their ability, team members are encouraged to openly share constructive feedback.
- The team open communication policy expects that each team member is responsive to reasonable and constructive criticism.
- If an issue has been raised and a resolution has been discussed, yet no progress for improvement has been made, the team will contact the course instructors for assistance.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:

- Create an initial design of the system by November
- Order and receive the necessary components by the end of Thanksgiving break.
- Large-scale breadboard prototype of the system and initial version of functional GUI by the end of the first semester.
- Initial PCB design completed by the end of the first semester

2. Strategies for planning and assigning individual and teamwork:

- The team will assign individual and team responsibilities during weekly meetings

3. Strategies for keeping on task:

- During the weekly meeting, the accomplishments of the previous week will be discussed.
- If team members decide work appears to be off task, discussion will take place in adherence to the team's open communication policy.

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?

- Minor infractions not deemed detrimental to the team's ability to complete the project will be discussed within the group and with Dr. Lu.
- If infractions of the team contract are deemed detrimental to the ability of the team to complete project assignments within the expected quality and deadline, the team will notify Dr. Lu and course instructors of the infractions and their impact.

2. What will your team do if the infractions continue?

- The team will approach course instructors to discuss potential next steps regarding future action, specifically regarding course penalties and potential separation from the team.

****

a) I participated in formulating the standards, roles, and procedures as stated in this contract.

b) I understand that I am obligated to abide by these terms and conditions.

c) I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.

1) Cade Kuennen DATE: 9/11/2024

2) Rakesh Varma Penmetsa DATE: 9/11/2024

3) Wes Ryley DATE: 9/11/2024

4) Alexander Upah DATE: 9/11/2024

Appendix 6 – Additional Appendixes

Appendix A: SK6812MINI

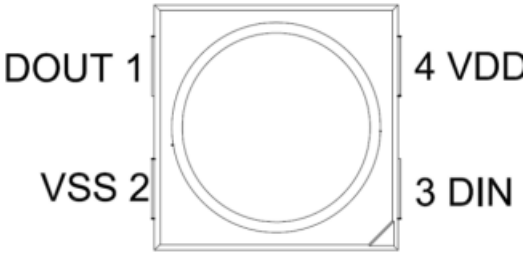

Figure A6-A-1: SK6821MINI Pin Outs 1

Parameter	Symbol	Range	Unit
Power supply voltage	VDD	+3.5～+5.5	V
Logic input voltage	V_{IN}	-0.5～VDD+0.5	V
Working temperature	Topt	-40~+85	℃
Storage temperature	Tstg	-50~+150	℃
ESD pressure	V_{ESD}	4K	V

Figure A6-A-2: SK6812MINI Electrical Characteristics

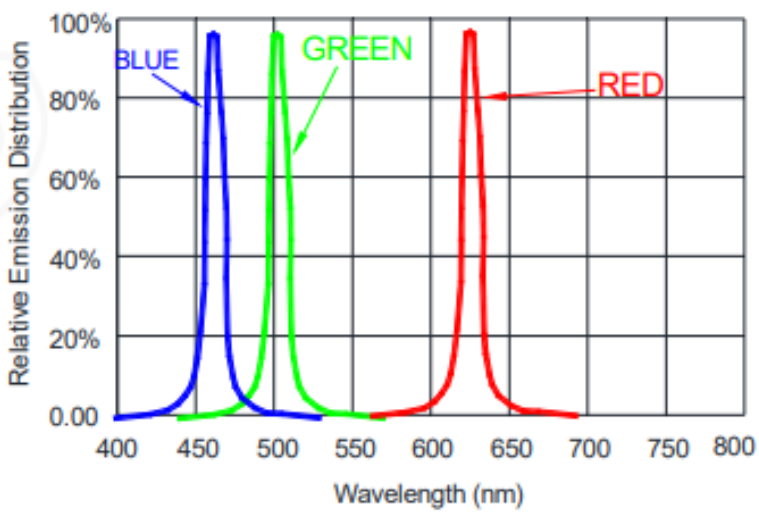

Figure A6-A-3: SK6812MINI Wavelength Emission Characteristics

The complete specifications can be found in the product datasheet: https://cdn-shop.adafruit.com/product-files/2686/SK6812MINI_REV.01-1-2.pdf

Appendix B: EOPD-525-1-0.9

Table B.1 depicts the optoelectronic characteristics of the device. A peak sensitivity wavelength of 525 nm provides excellent performance with our fluorescent excitation wavelength at 523 nm. A responsivity of 0.3 A/W is acceptable for this project, with the maximum demonstrated responsibility demonstrated in any photodiode being 0.5 A/W.

Measurement conditions Messbedingungen	$T_{ambient}$ = 23 °C; t_{test} ≤ 60 ms					
Parameter	**Symbol**	**Condition**	**Min**	**Typ**	**Max**	**Unit**
Emitting Color Farbe				Photo Diode Photodiode		
Peak Sensitivity Wavelength Peak Empfindlichkeitswellenlänge	λ_P	U_R = 0 V		525		nm
Sensitivity Range at 1% Empfindlichkeitsbereich bei 1%	λ	U_R = 0 V	410		580	nm
Spektral Bandwidth at 50% Spektrale Bandbreite bei 50%	$\Delta\lambda$	U_R = 0 V		70		nm
Responsivity at λ_P Empfindlichkeit bei λ_P	S_λ	U_R = 0 V		0.3		A/W
Dark Current Dunkelstrom	I_D	U_R = 5 V		5	30	pA

Table B-1: EOPD-525-1-0.9 Optoelectronic Characteristics

Figure B.1 depicts the graph of sensitivity as a function of wavelength. A moderately high sensitivity at 488 nm motivates our inclusion of a filtering component in the design to remove unwanted response to our LED.

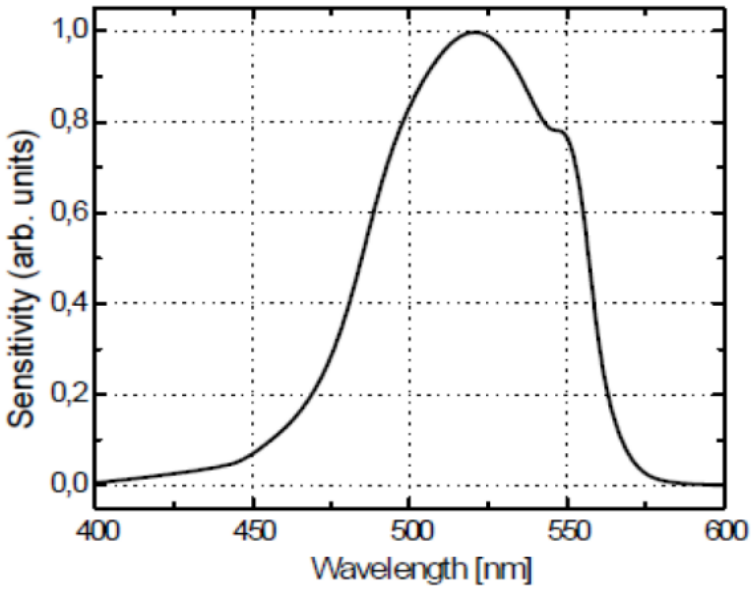

Figure A6-B-2: EOPD-525-1-0.9 Wavelength Sensitivity

The complete specifications regarding sizing, soldering, and temperature dependence can be found in the product datasheet: https://www.epigap-osa.de/wp-content/uploads/2023/09/EOPD-525-1-0.9.pdf

Appendix C: ESP32-C3

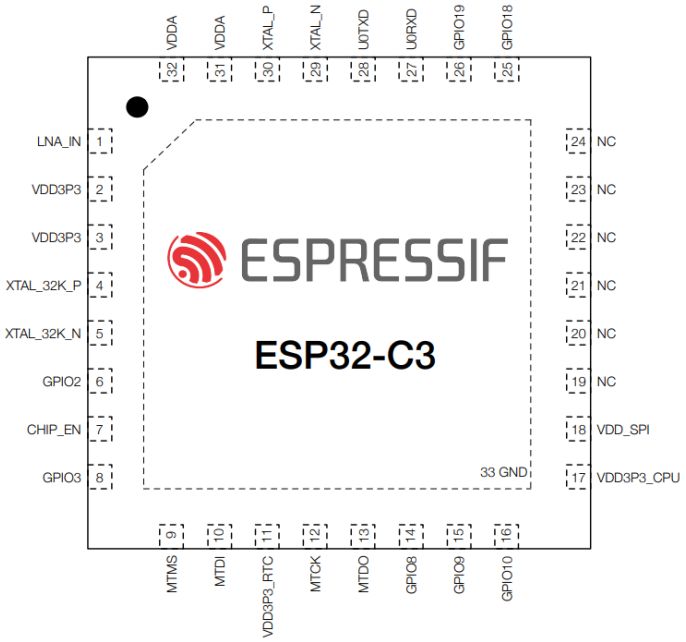

Figure C-1: ESP32-C3 Pin Layout

Pin No.	Pin Name	USB Serial/JTAG	JTAG	ADC1	ADC2	UART0 [1]	SPI0/1 [1]	SPI2 [1]	UART1	I2C	I2S	TWAI	LED PWM	RMT
1	LNA_IN													
2	VDD3P3													
3	VDD3P3													
4	XTAL_32K_P			ADC1_CH0		GPIO0	GPIO0	GPIO0	GPIO0	GPIO0	GPIO0	GPIO0	GPIO0	GPIO0
5	XTAL_32K_N			ADC1_CH1		GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	GPIO1
6	GPIO2			ADC1_CH2		GPIO2	GPIO2	FSPIQ	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2
7	CHIP_EN													
8	GPIO3			ADC1_CH3		GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3
9	MTMS		MTMS	ADC1_CH4		GPIO4	GPIO4	FSPIHD	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4
10	MTDI		MTDI		ADC2_CH0	GPIO5	GPIO5	FSPIWP	GPIO5	GPIO5	GPIO5	GPIO5	GPIO5	GPIO5
11	VDD3P3_RTC													
12	MTCK		MTCK			GPIO6	GPIO6	FSPICLK	GPIO6	GPIO6	GPIO6	GPIO6	GPIO6	GPIO6
13	MTDO		MTDO			GPIO7	GPIO7	FSPID	GPIO7	GPIO7	GPIO7	GPIO7	GPIO7	GPIO7
14	GPIO8					GPIO8	GPIO8	GPIO8	GPIO8	GPIO8	GPIO8	GPIO8	GPIO8	GPIO8
15	GPIO9					GPIO9	GPIO9	GPIO9	GPIO9	GPIO9	GPIO9	GPIO9	GPIO9	GPIO9
16	GPIO10					GPIO10	GPIO10	FSPICS0	GPIO10	GPIO10	GPIO10	GPIO10	GPIO10	GPIO10
17	VDD3P3_CPU													
18	VDD_SPI					GPIO11	GPIO11	GPIO11	GPIO11	GPIO11	GPIO11	GPIO11	GPIO11	GPIO11
19	SPIHD					GPIO12	SPIHD	GPIO12	GPIO12	GPIO12	GPIO12	GPIO12	GPIO12	GPIO12
20	SPIWP					GPIO13	SPIWP	GPIO13	GPIO13	GPIO13	GPIO13	GPIO13	GPIO13	GPIO13
21	SPICS0					GPIO14	SPICS0	GPIO14	GPIO14	GPIO14	GPIO14	GPIO14	GPIO14	GPIO14
22	SPICLK					GPIO15	SPICLK	GPIO15	GPIO15	GPIO15	GPIO15	GPIO15	GPIO15	GPIO15
23	SPID					GPIO16	SPID	GPIO16	GPIO16	GPIO16	GPIO16	GPIO16	GPIO16	GPIO16
24	SPIQ					GPIO17	SPIQ	GPIO17	GPIO17	GPIO17	GPIO17	GPIO17	GPIO17	GPIO17
25	GPIO18	USB_D-				GPIO18	GPIO18	GPIO18	GPIO18	GPIO18	GPIO18	GPIO18	GPIO18	GPIO18
26	GPIO19	USB_D+				GPIO19	GPIO19	GPIO19	GPIO19	GPIO19	GPIO19	GPIO19	GPIO19	GPIO19
27	U0RXD					U0RXD	GPIO20	GPIO20	GPIO20	GPIO20	GPIO20	GPIO20	GPIO20	GPIO20
28	U0TXD					U0TXD	GPIO21	GPIO21	GPIO21	GPIO21	GPIO21	GPIO21	GPIO21	GPIO21
29	XTAL_N													
30	XTAL_P													
31	VDDA													
32	VDDA													
33	GND													

[1] For UART0, SPI0/1, and SPI2 interface, the signals routed to fixed pins via IO MUX can also be routed to any GPIO pins via GPIO Matrix.

Figure C-2: ESP32-C3 Peripheral Pin Assignments

The complete specifications: https://www.espressif.com/sites/default/files/documentation/esp32-c3_datasheet_en.pdf

Appendix D: FGB25 & FGL550

FGB25: 25mm BG3 Colored Glass Bandpass Filter, 315 – 445 nm and 715 – 1095 nm

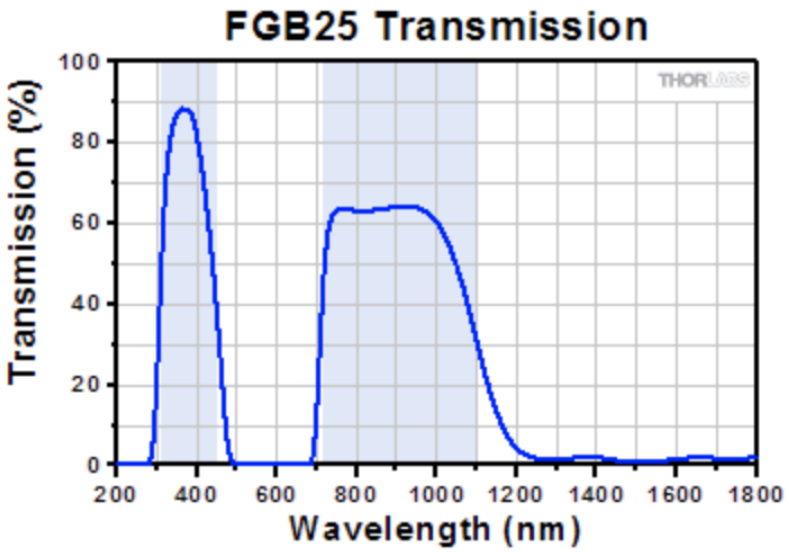

Figure D-1: FGB25 wavelength transmission spectrum

FGL550: 25 mm OG550 Colored Glass Filter, 550 nm Longpass

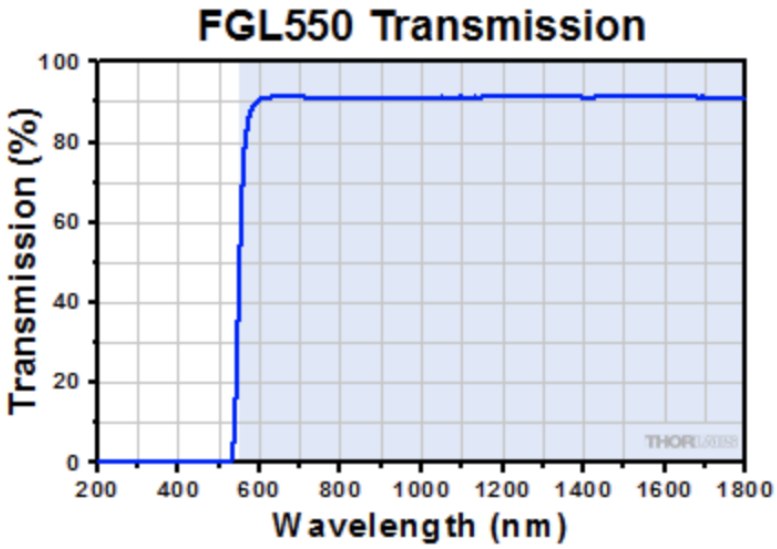

Figure D-2: FGL550 wavelength transmission spectrum

Visit product website for more information:
https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=3695