

**The Edward S. Rogers Sr. Department of
Electrical and Computer Engineering
University of Toronto**

**ECE496Y Design Project Course
Group Final Report**

Title: A Wearable Functional Near-Infrared Spectroscopy (fNIRS) based Brain Interface

Team #:	2024991	
Team members:	Name:	Email:
	Tony Kim	tonyhansol.kim@mail.utoronto.ca
	Kevin Liu	kevv.liu@mail.utoronto.ca
	Bella Huang	bella.huang@mail.utoronto.ca
	Ingrid Wu	ingrid.wu@mail.utoronto.ca
Supervisor(s):	Xilin Liu	
Section #:	4	
Administrator:	Steve Mann	
Submission Date:	March 21, 2025	

Group Final Report Attribution Table

Section	Student Names			
	Tony Kim	Kevin Liu	Bella Huang	Ingrid Wu
Executive Summary			RD, ET	
1.0 Introduction		RS, RD, ET		RS, RD, ET
2.0 Final Design	RD, ET			
2.1 Sensors	RD, ET			
2.2 Electrical Control Unit Hardware	RD, ET			
2.2.1 Sensing Design	RD, ET			
2.2.2 Emitter Control Interface	RD, ET			
2.2.3 MCU	RD, ET			
2.2.4 Power	RD, ET			
2.2.5 Connectors and Wiring	RD, ET			
2.3 Electrical Control Unit Firmware				RD, ET
2.3.1 Sensor Data Acquisition	MR, ET			RD
2.3.2 Emitter PWM Control	ET			RD
2.3.3 USB Data Transmission				RD, ET
2.4 Mechanical Design		RD, ET		
2.4.1 Head Harness Design		RD, ET		
2.4.2 Body Harness Design		RD, ET		
2.4.3 ECU Enclosure Design		RD, ET		
2.5 Software Design				RD, ET
2.5.1 Data Parsing				RD, ET
2.5.2 Data Processing and Interpretation		RS, RD, ET		
2.5.3 Web GUI			RD, ET, RS	
2.5.4 User Controls			RD, ET	
2.6 Final Design Assessment				RD, ET
3.0 Testing and Verification	RD, ET			
3.1 Sensor Hardware Testing	RD, ET			
3.2 ECU Hardware/Firmware Testing	RD, ET			
3.3 Software Verification	RD, ET		RD, ET	
3.4 Product Level Testing	RD, ET	RD, ET		
4.0 Summary and Conclusions			RD, ET	
5.0 Future Work				RD, ET
6.0 References			RD, ET	
7.0 Appendices	RD	MR		
All	FP	CM, FP	FP	FP

Abbreviation Codes:

Fill in abbreviations for roles for each of the required content elements. You do not have to fill in every cell. The “All” row refers to the complete report and should indicate who was responsible for the final compilation and final read through of the completed document.

RS – responsible for research of information

RD – wrote the first draft

MR – responsible for major revision

ET – edited for grammar, spelling, and expression

OR – other

“All” row abbreviations:

FP – final read through of complete document for flow and consistency

CM – responsible for compiling the elements into the complete document

OR - other

If you put OR (other) in a cell please put it in as OR1, OR2, etc. Explain briefly below the role referred to:

OR1: enter brief description here

OR2: enter brief description here

Signatures

By signing below, you verify that you have read the attribution table and agree that it accurately reflects your contribution to this document.

Name	Tony Kim	Signature	T.K	Date:	March 20, 2025
Name	Kevin Liu	Signature	K.L	Date:	March 20, 2025
Name	Bella Huang	Signature	B.H	Date:	March 20, 2025
Name	Ingrid Wu	Signature	I.W	Date:	March 20, 2025

Voluntary Document Release Consent Form

To all ECE496 students:

To better help future students, we would like to provide examples that are drawn from excerpts of past student reports. The examples will be used to illustrate general communication principles as well as how the document guidelines can be applied to a variety of categories of design projects (e.g. electronics, computer, software, networking, research).

Any material chosen for the examples will be altered so that all names are removed. In addition, where possible, much of the technical details will also be removed so that the structure or presentation style are highlighted rather than the original technical content. These examples will be made available to students on the course website, and in general may be accessible by the public. The original reports will not be released but will be accessible only to the course instructors and administrative staff.

Participation is completely voluntary and students may refuse to participate or may withdraw their permission at any time. Reports will only be used with the signed consent of all team members. Participating will have no influence on the grading of your work and there is no penalty for not taking part.

If your group agrees to take part, please have all members sign the bottom of this form.

Consent Statement

We verify that we have read the above letter and are giving permission for the ECE496 course coordinator to use our reports as outlined above.

Team #: 2024991

Project Title: A Wearable Functional Near-Infrared Spectroscopy (fNIRS) based Brain Interface

Supervisor(s): Xilin Liu Administrator: Steve Mann

Name	Tony Kim	Signature	T.K	Date:	March 20, 2025
Name	Kevin Liu	Signature	K.L	Date:	March 20, 2025
Name	Bella Huang	Signature	B.H	Date:	March 20, 2025
Name	Ingrid Wu	Signature	I.W	Date:	March 20, 2025

Group Highlights

Throughout this project, our team has successfully designed, developed, and validated a fully integrated, low-cost Functional Near-Infrared Spectroscopy (fNIRS) device. We effectively combined expertise across hardware design, firmware programming, mechanical engineering, and software development to deliver a comprehensive system capable of real-time monitoring of cerebral blood oxygenation levels. Our team undertook iterative development and effectively integrated multiple components, resulting in fully customized sensors and an electrical control unit implemented on designed printed circuit boards (PCBs), and mechanical structures tailored to the project's unique requirements.

Rigorous validation and testing procedures were conducted, confirming the accuracy, reliability, and overall performance of our final design. Each component, from the custom sensors to the software interface underwent extensive testing to ensure optimal functionality. These verification steps provided essential evidence supporting our design decisions, demonstrating that the developed fNIRS device meets its intended objectives. This comprehensive process has not only helped to validate our design choices but also laid a path forward for future next steps and broader adoption of fNIRS technology.



Figure 1. Final fNIRS Design - Photographed by Professor Steve Mann

Individual Contributions

Tony Kim

Tony contributed to the development, validation, and integration of the hardware and firmware aspects of the project. This ranged from schematic and printed circuit board (PCB) designs to full embedded software development, ensuring real time control and acquisition with great reliability and accuracy according to the desired requirements.

In the hardware design phase, Tony developed the complete schematics and PCB layouts for sensors, encompassing photodiode detection with analog filters and amplifiers and infrared light emission control with MOSFETs. He also developed the complete schematics and PCB layout of the electrical control unit (ECU) for the project, consisting of analog, digital, and power circuits that met the system requirements. He optimized cost in the part selection process while ensuring functional requirements were met. All the circuits were custom designed where Tony computed required calculations and ran simulations before ordering the design. Tony was also responsible for board assembly, where 30% of the components were soldered by hand (and the rest by the manufacturer).

For hardware bring up, Tony tuned the sensors (mainly the gains of the amplifiers) to ensure reliable readings could be acquired. He also developed the device drivers for the components on the ECU, verifying all hardware systems were working with firmware control.

Specifically, he worked on bringing up PWM, ADC, I2C, GPIOs, and UART peripherals, and set up an architecture to abstract hardware level code from the system level. Tony verified all hardware components with digital lab equipment and software test scripts.

After confirming hardware functionality, Tony developed higher level logic in the firmware, including a state machine for emitter controls and a sequencer for sensor channel reading selection. He contributed to adding real time capability to the firmware by utilizing interrupts with timers and direct memory access (DMA), and tuned for reliability and accuracy until completion. For integration, Tony worked with fellow team members to connect the hardware and the user facing software together, resulting in the final product.

Kevin Liu

Kevin contributed to both the mechanical design and software data processing aspects of the project. His mechanical design work addressed essential ergonomic and functional considerations of the physical design, improving the usability, portability, and wearer-comfort of the overall final system. The software data processing algorithm effectively translated theoretical principles into a practical, real-time signal conversion implementation suitable for low cost functional near-infrared spectroscopy devices.

In mechanical design, Kevin conceptualized and developed the head harness, integrating custom protective capsules, 3D printed using PLA materials. He engineered a twist-lock mechanism for accurate, intuitive, and secure installation of sensors and emitters, incorporating ergonomic principles and ensuring simple user operation. Kevin also designed the ECU enclosure, incorporating internal supports and strategic compartmentalization to enhance operational stability and convenience for the user. Furthermore, Kevin designed the main body harness with a custom 3D-printed mounting adapter for the ECU enclosure, facilitating seamless integration with the system's mechanical and electrical components. Each one of the mechanical component underwent multiple design iterations and refinements based on testing and team member, supervisor, and administrator feedback.

In software data processing, Kevin developed a computational pipeline based on foundational principles of functional near-infrared spectroscopy. He adapted existing researched methods and applied theory behind optical density transformations, the modified Beer-Lambert law, and correlation-based signal improvement techniques into a tailored real-time processing algorithm. Through the integration of established theoretical frameworks and research, Kevin provided a solution capable of accurately extracting and interpreting physiological signals from raw sensor data in real time.

Bella Huang

Throughout the project, Bella was responsible for the software interface component, bridging the fNIRS Electrical Control Unit (ECU) and the end users. Bella's primary role involved designing and implementing a web-based graphical user interface (GUI) to facilitate real-time visualization and interaction with fNIRS data.

Initially, as hardware components were still under development, Bella developed a multi-threaded WebSocket server capable of generating synthetic fNIRS data. This enabled Bella to perform early-stage software testing and ensure the robustness of data handling and visualization processes without relying on physical sensor inputs. Bella utilized Dash, a Python framework tailored for visualizations, in conjunction with Plotly.js, for rendering dynamic and responsive graphs and jQuery for asynchronous data handling. Additionally, Bella collaborated closely with Ingrid in researching various 3D brain visualization libraries and frameworks, selecting tools suitable for the project's requirements.

As the project progressed and the data processing pipeline was established, Bella transitioned the system architecture to utilize a backend Flask server powered by Flask-SocketIO to handle real-time, bidirectional communication between the firmware and connected clients. This setup allowed continuous streaming of processed sensor data packets from the data processing pipeline. The GUI featured a dynamic 3D brain mesh visualization, along with interactive activation plots displaying real-time, historical neural activity.

In addition, Bella implemented a sensor control module within the web-based interface, providing intuitive controls for users to manage hardware settings such as emitter states, PWM intensities, and multiplexer selections directly from the GUI. To ensure reliability and performance, Bella also conducted verification tests to validate real-time data streaming, efficient data queuing, multi-threading, and accurate visual rendering of data within the GUI.

Ingrid Wu

Ingrid contributed to the development, testing and integration of the system, particularly in firmware, sensor validation and software data processing. In the firmware development, she focused on the communication between the Electrical Control Unit (ECU) and the software interface. She implemented a USB communication protocol, ensuring efficient and structured data transmission from the hardware to the PC for further analysis. She also created PCB breakout boards for hardware validation and testing of the sensors with the ECU.

Additionally, she contributed in hardware testing/validation of the sensors, verifying emitter functionality, ADC accuracy and system synchronization. She worked on debugging and improving the sensor data acquisition process, ensuring reliable operation. She also assisted in troubleshooting sensor hardware issues, verifying that the ADC sampled correct values, and ensuring MUX selection aligned with expected sensor channels.

Beyond firmware and hardware, Ingrid contributed to software data processing, where she helped develop methods for decoding raw sensor packets, structuring data for processing, and ensuring compatibility with signal analysis pipelines. This includes ensuring that received data packets are accurately parsed and synchronised for fNIRS analysis.

Acknowledgements

We would like to express our sincere gratitude to those who have supported and guided us throughout the development of this project.

First and foremost, we extend our appreciation to Professor Xilin Liu for his invaluable mentorship, technical expertise, and insightful feedback, which played a crucial role in shaping our design and ensuring the success of our project. His guidance has been instrumental in helping us navigate challenges and refine our approach to developing an effective and efficient fNIRS system.

We would also like to acknowledge our administrator Professor Steve Mann for his continuous support and encouragement throughout the project. His contributions in facilitating resources, providing logistical assistance, and fostering an environment conducive to innovation have greatly contributed to the completion of our work.

Executive Summary

Monitoring of brain activity is fundamental for targeting neurological disorders and cognitive development. Functional Near-Infrared Spectroscopy (fNIRS) is a non-invasive technology that utilizes near-infrared light to measure real-time changes in blood oxygenation levels, providing insights into neural activity. This is based on the modified Beer-Lambert Law, which states that light absorption is proportional to the concentration of the absorbing substance and the distance by which light travels through. However, existing fNIRS devices are often expensive, bulky, and overly complex, restricting their use to specialized clinical and research environments.

This project addresses the limitations of existing fNIRS devices by developing a low-cost, ergonomic fNIRS device suitable for broader applications in education and personal health monitoring, while maintaining high accuracy and usability. The final design integrates hardware, firmware, mechanical structures, and an interactive software interface. It features 24 custom-designed sensor boards, consisting of 16 dedicated photodiode detectors and 8 emitter-detector modules employing dual-wavelength LED emitters (660 nm and 940 nm). These sensors are interfaced with a custom Electrical Control Unit (ECU), based on a low-power STM32 microcontroller. Custom mechanical components ensure comfortable and stable placement of sensor modules and the ECU on the user, while an interactive software interface enables intuitive real-time data visualization and sensor control.

Throughout development, rigorous validation and verification tests were performed, confirming the accuracy, safety, and comfort of the device. Key design achievements of the device include accurate data sampling with effective hardware and software filtering, reliable photodiode detector performance verified through controlled optical tests, low total fabrication cost (~\$750 CAD), comfortable and ergonomic wearability, and confirmed safe, non-invasive operation.

Future work includes further refinement of data transmission by implementing SD card logging for offline data storage. Additionally, integrating firmware support for onboard temperature sensors will improve system reliability. Optimizing the PCB designs for the ECU and sensor boards by reducing component size and refining placement will further enhance efficiency and minimize overall board dimensions. These improvements together will enhance the functionality, scalability, and overall system performance of the device.

Table of Contents

1. Introduction	12
2. Final Design	13
2.1 Sensors	15
2.2 Electrical Control Unit Hardware	18
2.2.1 Sensing Design	19
2.2.2 Emitter Control Interface	20
2.2.3 MCU	20
2.2.4 Power	20
2.2.5 Connectors and Wiring	20
2.3 Electrical Control Unit Firmware	21
2.3.1 Sensor Data Acquisition	22
2.3.2 Emitter PWM Control	23
2.4 Mechanical Design	24
2.4.1 Head Harness Design	25
2.4.2 Body Harness Design	26
2.4.3 ECU Enclosure Design	26
2.5 Software Design	28
2.5.1 Data Parsing	28
2.5.2 Data Processing and Interpretation	28
2.5.3 Web GUI (Graphical User Interface)	30
2.5.4 User Controls	32
2.6 Final Design Assessment	33
3. Testing and Verification	34
3.1 Sensor Hardware Testing	34
3.2 ECU Hardware/Firmware Testing	35
3.3 Software Verification	37
3.4 Product Level Testing	38
4. Summary and Conclusions	41
6. References	43
7. Appendices	44
Appendix A: Gantt Chart History	45
Appendix B: Financial Plan	47
Appendix C: Sensor Schematics and PCB Design	50
Appendix D: ECU Schematics and PCB Design	56
Appendix E: Cable Assembly and System Wiring Diagram	74
Appendix F: Sensor Noise Level Testing Results	78
Appendix G: ADC Calibration Tests	84

1. Introduction

This report presents the motivation, design, implementation, and testing procedures of a Wearable Functional Near-Infrared Spectroscopy (fNIRS) based Brain Interface, developed as part of the final-year design course, ECE496. It provides a detailed overview of the full stack integration, design, testing and verification procedures. Finally, the report concludes with recommendations for future improvements and research directions.

Monitoring and understanding brain activity is a critical challenge in medicine today as it aids in diagnosing complex neurological disorders and enhances our understanding of human cognitive functions. Functional Near-Infrared Spectroscopy (fNIRS) is an emerging brain-machine interfacing technology that uses near-infrared light emission and absorption to monitor brain activity by measuring changes in blood oxygenation and blood volume within the cerebral cortex. Increases in neural activity lower the concentration of oxyhemoglobin—an oxygen-carrying protein in the blood—and increase the concentration of deoxyhemoglobin. These changes in concentration directly affect the transmission of the near-infrared light projected into the scalp and cerebral cortex [1], compiling with the modified Beer-Lambert Law. The law states that the absorption of light is directly proportional to both the distance which light travels and the concentration of the absorbing substance [2]. Thus, fNIRS technology offers a non-invasive and valuable way to gain real-time brain insight without the need for surgery or implants.

Beyond simply measuring and processing light data, current fNIRS solutions incorporate hybrid or multimodal brain activity monitoring methods, such as the ability to perform Electroencephalogram (EEG) alongside fNIRS [3]. While fNIRS has shown promise in clinical and research settings, many of these existing devices are often expensive, bulky, and designed for specialized use [4]. This limits accessibility for average users and non-clinical applications, especially when the primary need is only utilizing near-infrared light for basic brain activity monitoring.

Our project oversees the design, validation, and integration of an fNIRS device along with a user-interactable computer software program. The fNIRS device consists of light source emitters, photodiode sensors, and an electrical control unit (ECU) to control, process, and transmit all inputs and outputs of the system. To achieve a low-cost, yet reliable design, the project encompasses off-the-shelf commercial light source emitters and photodiodes built onto custom printed circuit boards (PCBs) that will serve to interact and capture meaningful

brain activities. The main ECU is also custom-designed from schematic and PCB layouts to embedded software development, where a microcontroller is used as the primary computation unit. In addition to firmware, the computer software program is application-specific and developed from scratch.

To ensure the designed fNIRS device satisfies the project goals and requirements, a comprehensive validation process has been conducted where individual components underwent independent testing before gradually being integrated and tested as a whole. The goals and requirements table below outlines key project requirements that must all be satisfied to achieve a successful project.

Table 1: Goals and Requirements

ID	Project Requirement
1.0 Data Sample Rate	Ensure a 50Hz signal can be acquired with minimal noise
2.0 Acquiring data signal	Ensure detector optode voltage output matches photodiode light absorption behaviour
	ADC reading reflects voltage being read
3.0 Outputting data signals to PC	Read serial data from ECU on PC
	Plot and view output signals on a PC
4.0 Cost	Net material/fabrication cost of the fully assembled device below \$1000
5.0 Physical Properties	Assembled device is wearable

2. Final Design

The final design is a fully custom and affordable fNIRS device, integrating hardware, firmware, mechanical, and software components to meet the project's goals and requirements. There are three main subsystems in the design: the sensors, fNIRS ECU, and the PC running the user-interfacing software program. The detailed high-level system overview is as shown below in Figure 2.

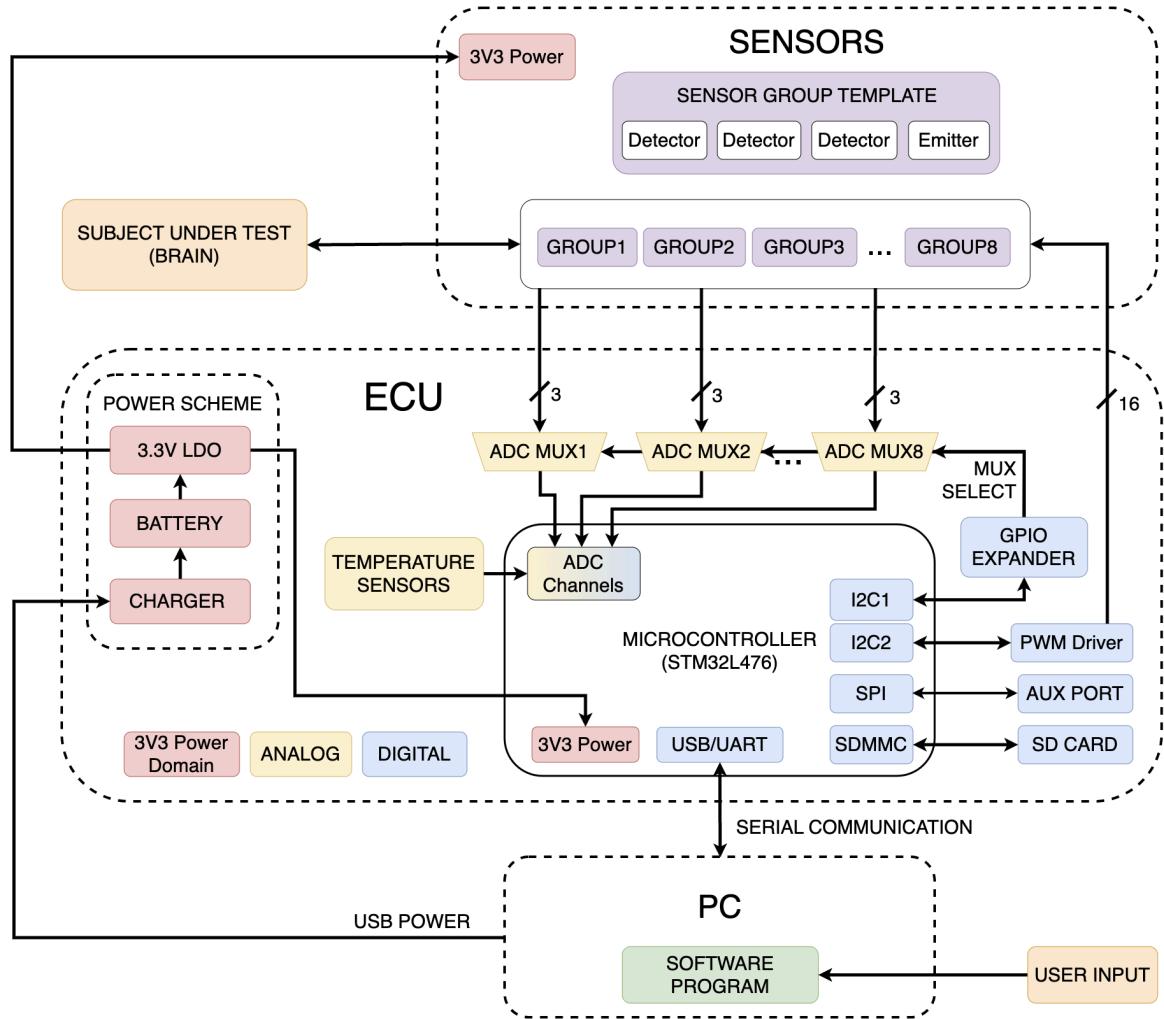


Figure 2. System Block Diagram

The sensor subsystem consists of eight sensor groups, each group consisting of three detectors and one emitter that measure brain activity based on the modified Beer-Lambert Law. The emitters and detectors are controlled and processed by the ECU and the sensor data is packaged and forwarded to the PC by the microcontroller onboard the ECU. The PC runs a software program for post-processing, analysis, display, and control of the entire system that users can interact with. Furthermore, the components are housed within a custom-design structure that ensures proper placement and stability while maintaining wearer comfort.

The development of these subsystems involved contributions from multiple team members, each specializing in different aspects of the design. The table below outlines the primary contributors for each section of this final design report, reflecting their roles in hardware, firmware, mechanical, and software development and integration.

Table 2: Author Contributions

Section	Primary Author(s)
2.1 Sensors	Tony Kim
2.2 Electrical Control Unit Hardware	Tony Kim
2.2.1 Sensing Design	Tony Kim
2.2.2 Emitter Control Interface	Tony Kim
2.2.3 MCU	Tony Kim
2.2.4 Power	Tony Kim
2.2.5 Connectors and Wiring	Tony Kim
2.3 Electrical Control Unit Firmware	Ingrid Wu, Tony Kim
2.3.1 Sensor Data Acquisition	Ingrid Wu, Tony Kim
2.3.2 Emitter PWM Control	Ingrid Wu, Tony Kim
2.3.3 USB Data Transmission	Ingrid Wu
2.4 Mechanical Design	Kevin Liu
2.4.1 Head Harness Design	Kevin Liu
2.4.2 Body Harness Design	Kevin Liu
2.4.3 ECU Enclosure Design	Kevin Liu
2.5 Software Design	Ingrid Wu, Kevin Liu, Bella Huang
2.5.1 Data Parsing	Ingrid Wu
2.5.2 Data Processing and Interpretation	Kevin Liu
2.5.3 Web GUI	Bella Huang
2.5.4 User Controls	Bella Huang
2.6 Final Design Assessment	All

2.1 Sensors

The sensors are custom-designed PCBs consisting of both near-infrared detection and emission circuitry, where a single board can be configured to be an emitter, a detector, or both. Different configurations allow the control of the distances between sensors on the head. The entire system consists of a total of twenty-four sensor boards, where there are sixteen

standalone detectors and eight fully populated boards with both emitters and detectors. Figure 3 below shows the 3D model of the sensor PCB design.

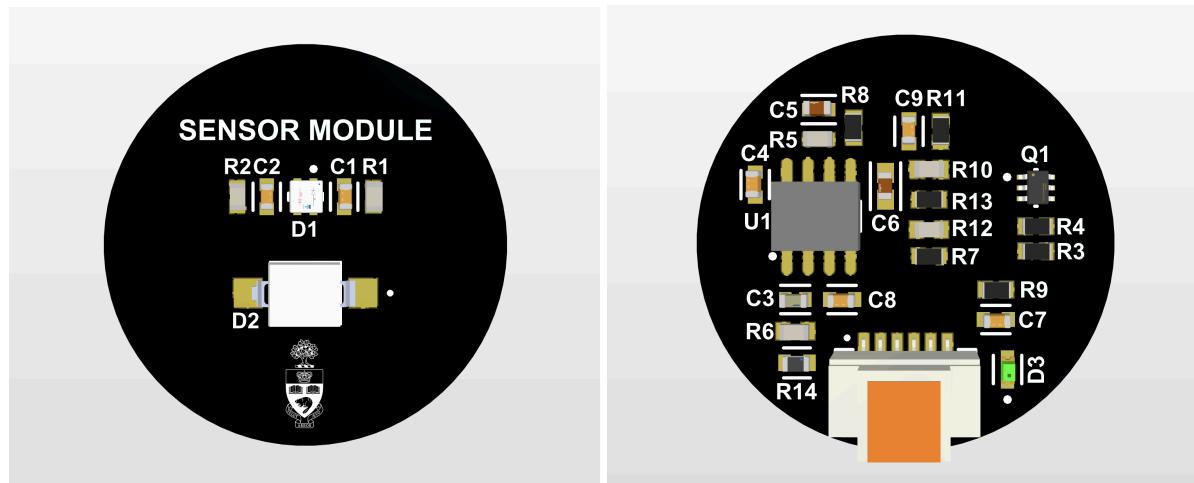


Figure 3. Fully assembled sensor printed circuit board in Altium Designer. The board is 2.5cm in diameter.

For emission, the design utilizes a dual-wavelength LED emitter (VSMD66694), capable of emitting 940 nm and 660 nm wavelengths. Each emitter is controlled independently via low-side MOSFET control, where the design uses a logic level gate 2 N-channel MOSFET component (BSD840N). The detection circuitry consists of photodiode sensing which encompasses the VBPW34S photodiode followed by two amplifier stages: transimpedance and non-inverting AC gain. Analog filtering is also incorporated for stability. Figure 3 shows the two amplifier stages of the detector schematic. Consult Appendix C for the full schematic design of the sensors.

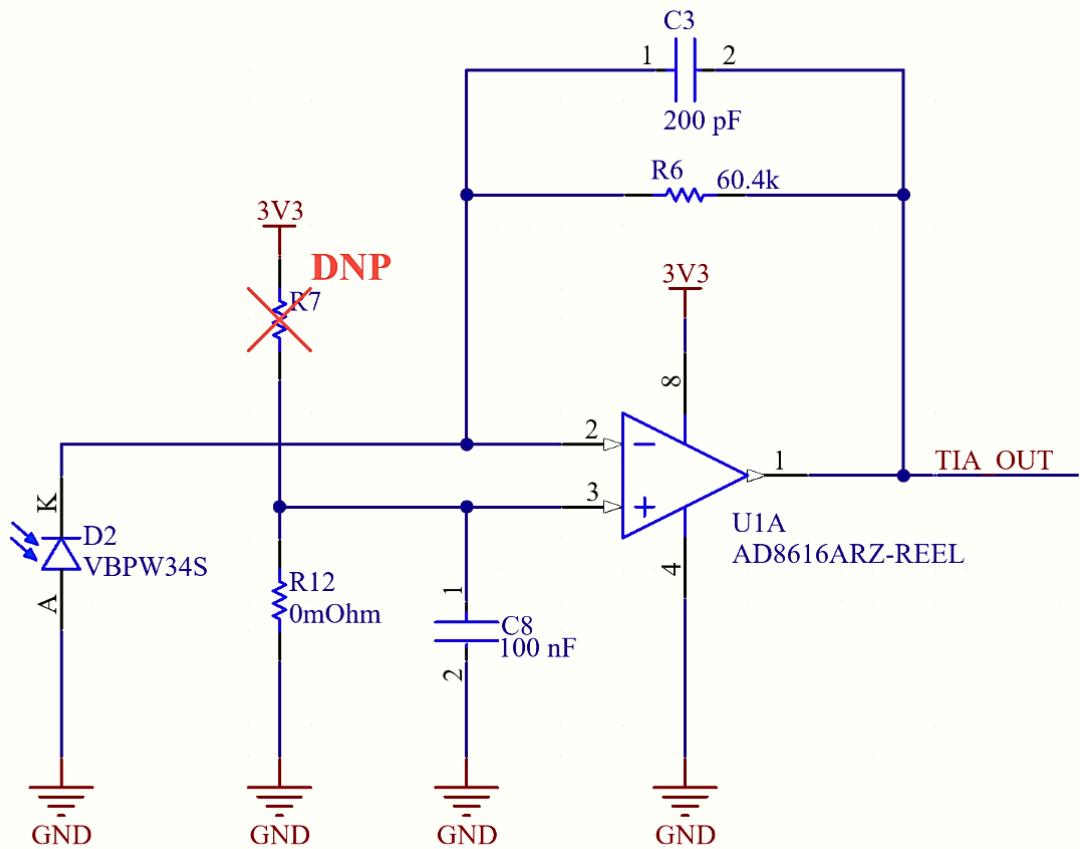


Figure 4. Photodiode and transimpedance amplifier (TIA) schematic

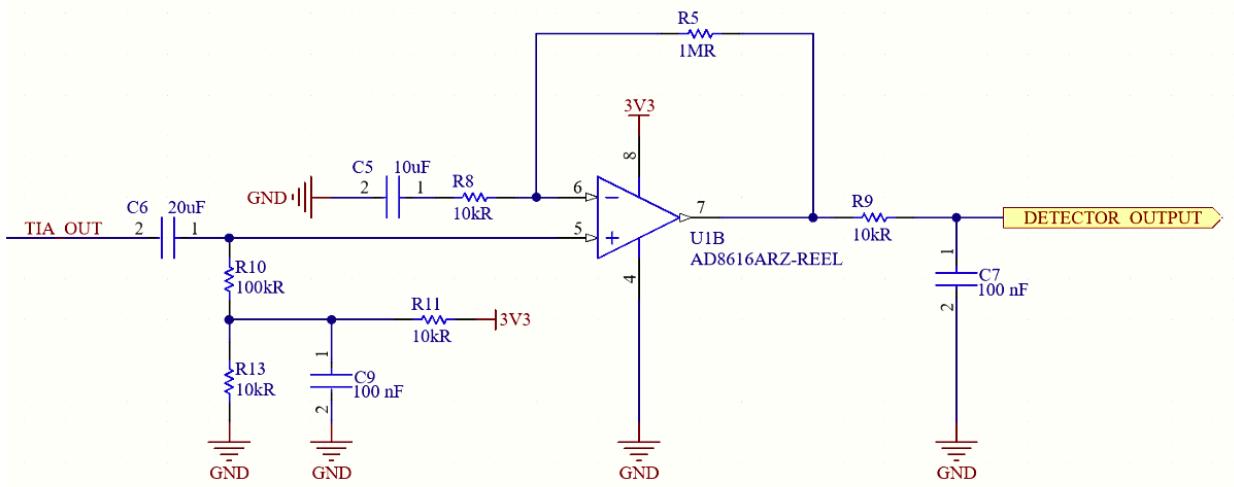


Figure 5. AC non-inverting gain amplifier with DC biasing and analog filtering

The VBPW34S photodiode is a silicon PIN diode with a spectral bandwidth of 430 nm to 1100 nm and a large sensitivity area of 7.5 mm^2 . It has a reported maximum current of 55 μA which was used to design the gain of the transimpedance amplifier set by the feedback resistor R6 in Figure 4. The bandwidth of the transimpedance amplifier is set by the feedback capacitor (C6) – approximately 13.175 kHz. Additionally, the operational amplifier, AD8616, from Analog Devices, was selected in the circuit design for its application in photodiode detection, with a low offset in the range of $23 \mu\text{V} - 60 \mu\text{V}$, a low noise level of $10 \text{ nV}/\sqrt{\text{Hz}}$, and a low input bias current within $0.2 \text{ pA} - 1 \text{ pA}$. A dual circuit package of this operational amplifier is used to also implement the AC non-inverting gain stage of the detector circuit, which takes the voltage output of the transimpedance amplifier as its input to further amplify the signal.

To read physiological and neurological signals, we extract the AC component out of the photodiode input by using a high pass filter with a cutoff frequency of 0.0796 Hz. Further, a DC offset is added to ensure that the entire waveform is visible at the output without the amplifier saturating due to its single-ended design with limits between 3.3 V and 0 V. The gain of the non-inverting amplifier was determined empirically and set to 101 V/V as seen in Figure 5 and an additional low pass filter is added at the output to increase the stability of the signal reading. Overall, the detected signal is not expected to exceed 10 Hz [5].

The sensors are powered by 3.3 V, running on the same power domain as the ECU. The ECU also provides two pulse-width modulated signals (PWMS) to drive and control the intensity of the emitters. The detected and processed analog output from the sensor is read by the ECU. For additional details on the sensor hardware design, see Appendix C.

2.2 Electrical Control Unit Hardware

The ECU is designed on a custom four-layer PCB with a mix of analog and digital circuits interfacing with the sensors and the PC running the software program. A microcontroller (MCU) serves as the main computation unit on the ECU. The hardware design is divided into five major blocks: Sensing, Emitter Control, MCU, Power, and Connectors as seen in the top-level page of the schematic (see Appendix D for the completed schematics and PCB layout). Figure 6 below shows the fully assembled 3D model of the ECU PCB.

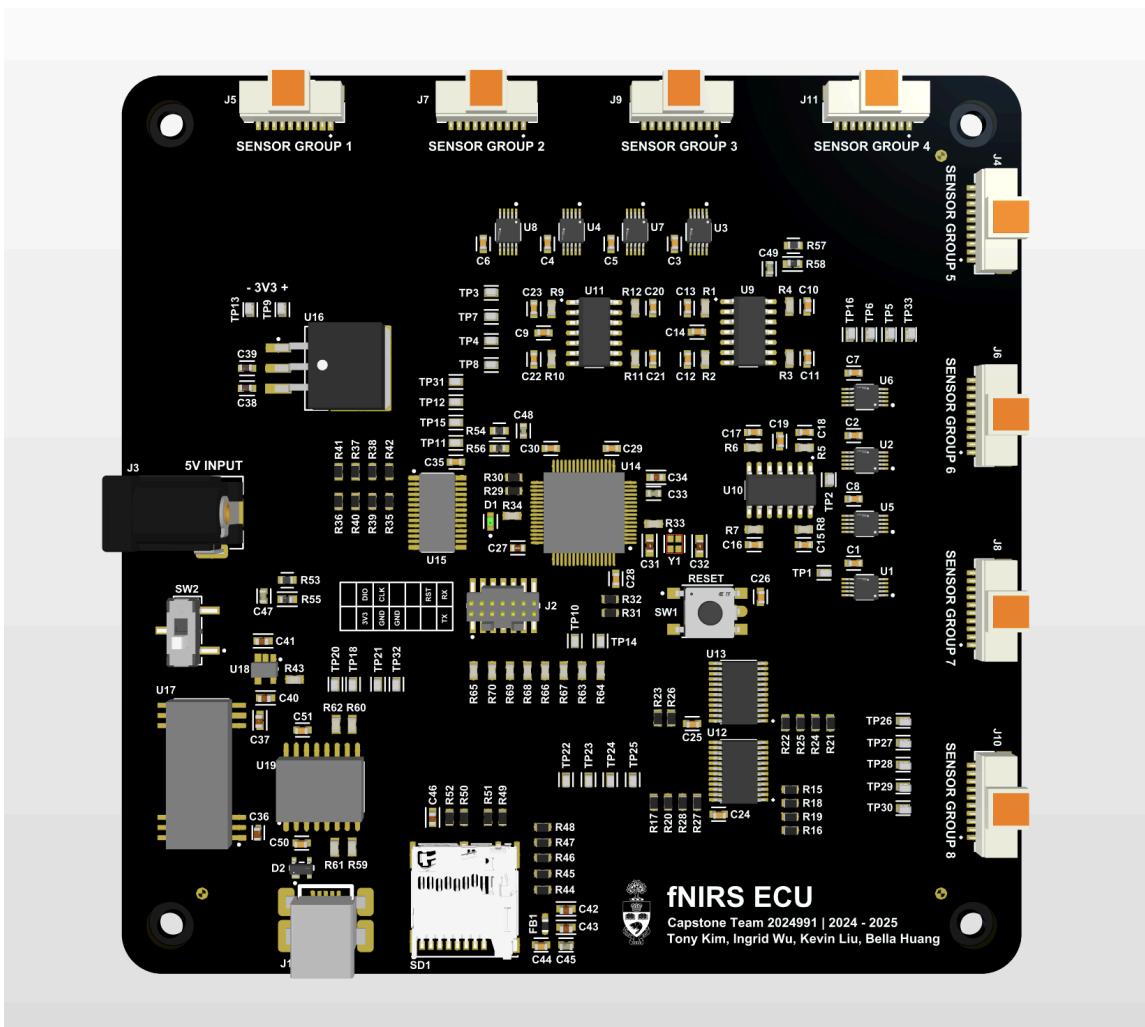


Figure 6. Fully assembled ECU printed circuit board in Altium Designer

2.2.1 Sensing Design

To account for a large number of sensors in the device, the design incorporates analog multiplexing to reduce the number of analog sensing circuits and the analog-to-digital converter (ADC) channels on the microcontroller, saving cost overall. The sensing design consists of eight three-to-one multiplexers (TMUX1104) followed by voltage follower (buffer) circuits and low-pass filtering upstream of the ADC channels on the microcontroller. The multiplexers are controlled by GPIOs, where I2C interface GPIO expander integrated circuits are used such that a lower pin count and cheaper microcontroller could be selected. Additionally, three thermistors are used to monitor ambient temperatures.

2.2.2 Emitter Control Interface

The emitter control hardware design consists of a single I2C (supporting fast mode plus) interface PWM controller integrated circuit (PCA9685) that drives a total of 16 logic-level PWM channels. The PWMs have a resolution of 12 bits and are configurable to frequencies between 24 Hz and 1526 Hz, and full duty cycle and phase shift control between 0 and 1. This emitter control design promotes cost savings as the price of the PCA9685 is ten times cheaper than other potential designs such as using a higher pin count microcontroller with 16 PWM channels.

2.2.3 MCU

The STM32L476RET6 microcontroller was selected for its low power and cost as well as its available peripherals to meet the hardware design demands. The core design uses two I2C buses to control and communicate with the GPIO expander and PWM controller integrated circuits respectively, a total of 12 ADC channels for sensing, SDMMC for writing data to an SD card, and USB interface for serial communication with the PC. For programming the microcontroller, serial wire debug and UART signals are made available, and an auxiliary SPI bus was exposed externally to allow for potential additional digital interfaces that the design may add in the future (i.e. sensors with digital communication instead of analog readings). For the USB interface, a digital USB isolator is included for electrical safety and improved signal integrity.

2.2.4 Power

The ECU is designed to be powered by a single lithium-ion cell. A low voltage drop-out regulator is used to regulate 3.3V for all the devices on the ECU as well as the sensors. Additionally, a battery charger circuit controlled by a mechanical slide switch is incorporated, where power input comes from the USB bus voltage. To maintain isolation of the USB interface, a 2 watt 5V-to-5V isolated DC DC converter is used.

2.2.5 Connectors and Wiring

For the sensing interface, there are eight connectors, where each connector maps to three sensors: two standalone detectors and one emitter-detector pair. A single connector provides power signals (3.3V and ground), three detector channels, and two PWMs for emitter control,

and a custom cable is made to split the signals between three sensors as seen in the figure below. See Appendix E for the cable assembly and system wiring diagram.

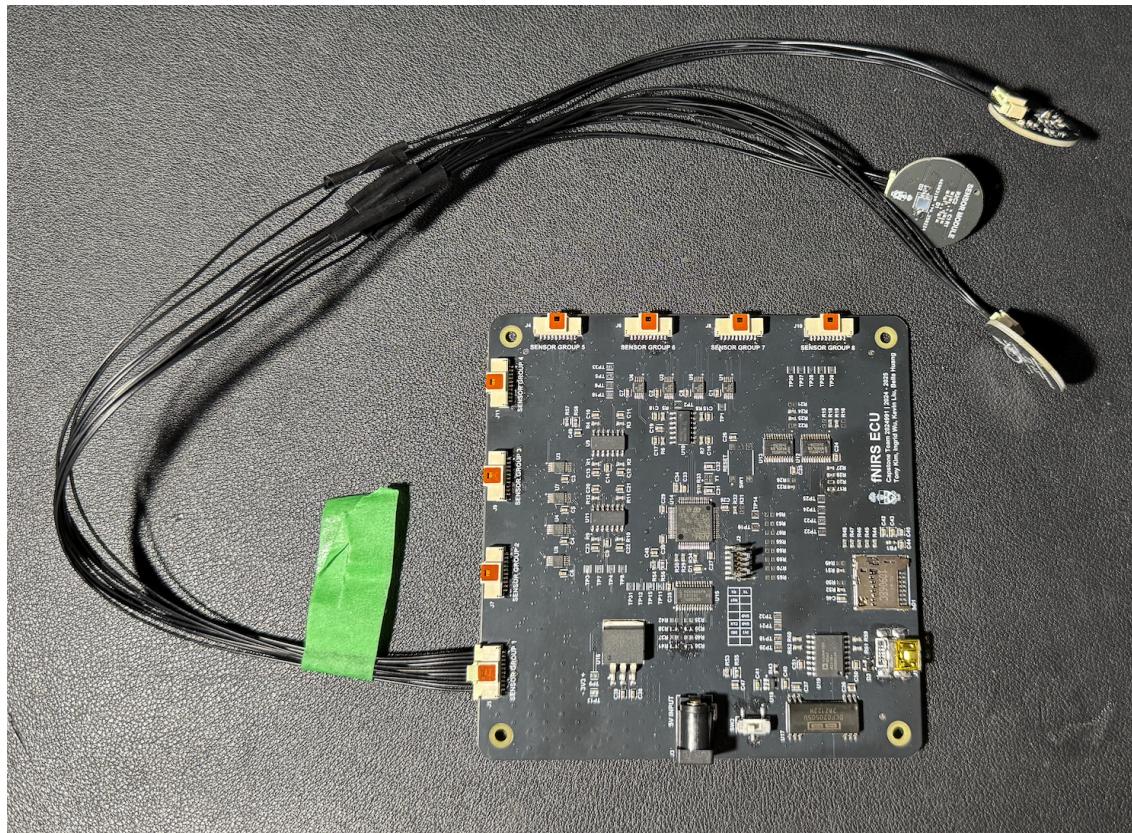


Figure 7. Sensing cable harness

2.3 Electrical Control Unit Firmware

Our project leverages an STM32L4 series microcontroller-based firmware running at 80MHz to acquire and transmit sensor data at high rates while maintaining accurate performance. The firmware is structured into hierarchical layers, each handling specific system functionalities, as shown in Figure 8. The main layer is the top-level module that ensures proper system operation and synchronization across tasks. The application layer handles MUX and emitter control, ADC sampling and sensor data acquisition. The driver layer handles control for hardware peripherals including the GPIO expander and PWM controller. Finally, the HAL layer is the lowest level, interfacing directly with the microcontroller's hardware peripherals, including I2C and USB communication, ADC conversion, and GPIO pin configurations.

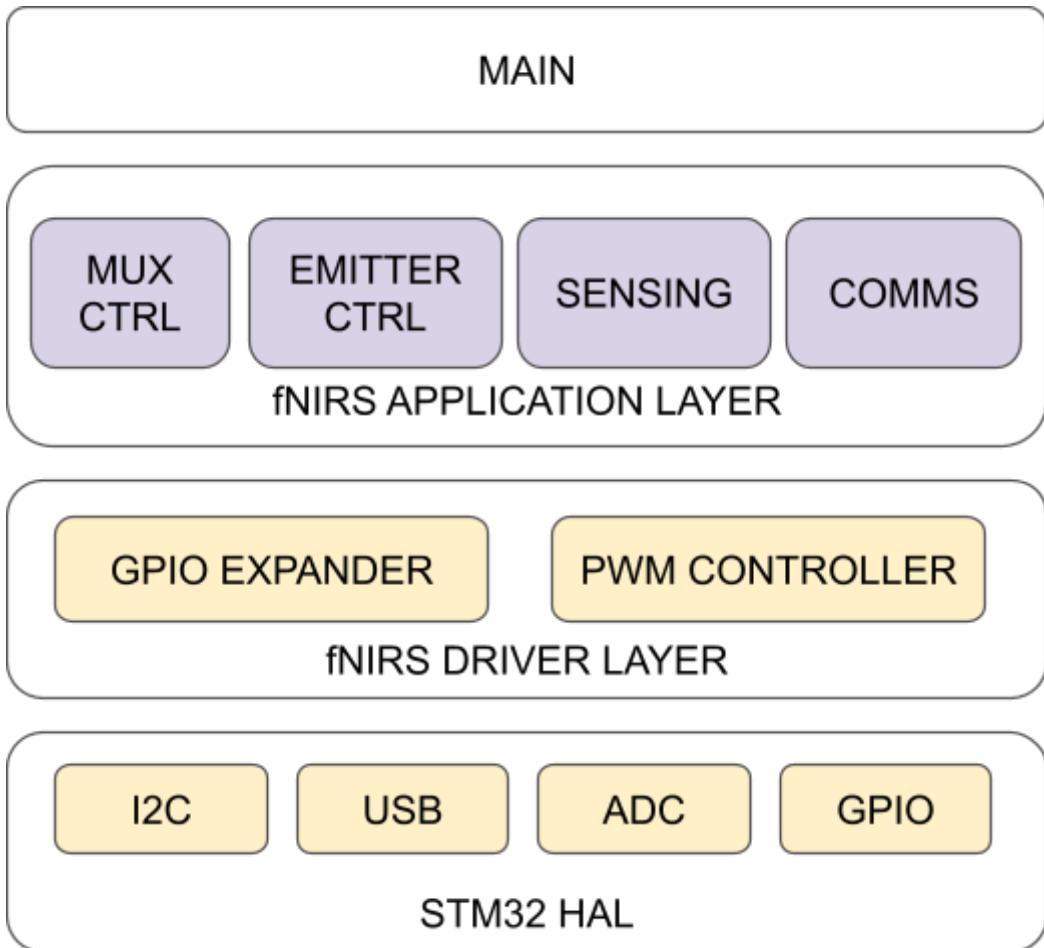


Figure 8. Firmware Architecture

2.3.1 Sensor Data Acquisition

The ADC is configured to capture sensor data across 8 channels per cycle using a 12-bit resolution. As the hardware design uses external multiplexers, the firmware controls GPIO expander circuits via I2C in fast mode (400 kHz) to set the desired sensor channel connected to the ADC input. The multiplexers are switched periodically every few milliseconds.

The start of conversion of the ADC is triggered externally by a high-priority hardware timer interrupt (timer 3), which ensures a 5 kHz internal sampling rate in which for three multiplexed channels, a sampling rate of approximately 1.6 kHz per channel can be achieved. The ADC clock, which determines the internal conversion speed, is set to 48 MHz, and the sample time per channel is set to 47.5 cycles to balance conversion speed and reading stability. As a result, each channel takes approximately 1 μ s to sample.

To optimize data handling, the firmware employs Direct Memory Access (DMA) in circular mode, allowing continuous ADC data transfers to memory without any CPU intervention. After sampling 8 channels, an interrupt service routine callback function processes the readings, which executes calibration, filtering (software low pass filter), and storing real-time data in memory. Logging frequency is then controlled by another hardware timer interrupt at a lower priority (timer 4) which populates the transmit buffer for USB data transmission.

2.3.2 Emitter PWM Control

In addition to managing sensor selection, the firmware also controls the light emitters through the I2C interface PWM controller. The PWM signal has a 12-bit resolution, which allows precise intensity adjustments through modifying the duty cycle. The firmware manages wavelength selection between 940 nm and 660 nm emitters using a state machine outlining the different operating modes as shown in Table 3 below.

Table 3: Emitter Operating Modes

Emitter Operating Modes	Description
Disabled	Both 940 nm and 660 nm emitters OFF
User Control	Emitter mode defined by user via user interface
Cycling	Periodically switch between 940 nm and 660 nm emitters every 5 seconds
940 nm Fully On	All emitters operate in 940 nm
660 nm Fully On	All emitters operate in 660 nm

Cycling mode is implemented by default, using a timer-based interrupt triggered every 5 seconds to switch the active emitter. The firmware communicates with the PWM driver via I2C at 400 kHz, ensuring rapid updates to duty cycle, frequency, and phase shift settings. By default, the emitters operate at a frequency of 1500 Hz, a duty cycle of 100%, and a phase shift of 0°.

2.3.3 USB Data Transmission

To ensure efficient data transmission, the USB interface runs in full-speed mode at 12 Mbit/s. Sensor data is formatted into structured packets containing three ADC sensor channel readings per module, along with the status of 940 nm and 660 nm emitters. Each data packet includes a packet identifier, sensor values (split into high/low bytes for USB transmission compatibility), and emitter statuses, allowing for easy parsing on the receiving end. The firmware also supports incoming USB control commands, enabling users to dynamically adjust settings such as emitter control, multiplexer selection, and sampling parameters.

2.4 Mechanical Design

The final mechanical assembly of the proposed design is illustrated in Figure 9 below. The system comprises three primary components: the head harness, the body harness, and the ECU enclosure.

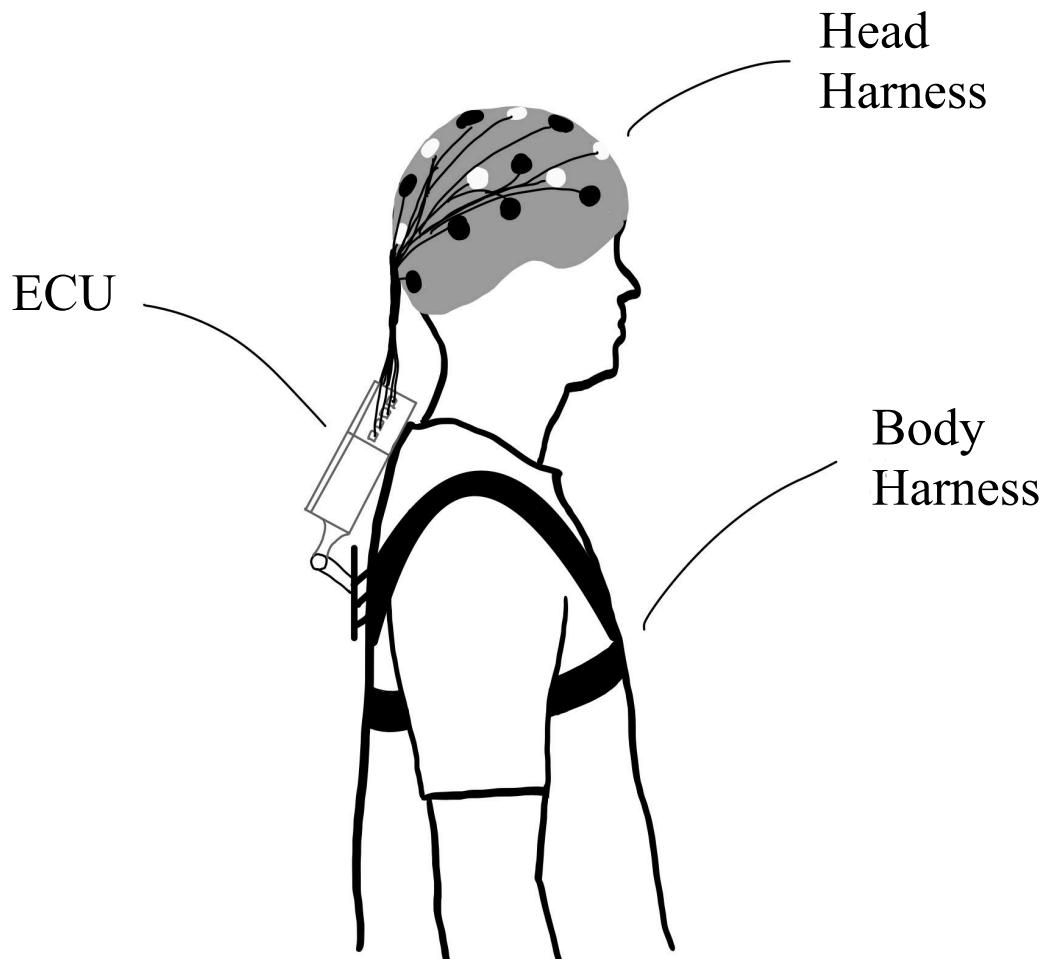


Figure 9. Hand-drawn Illustration of the Assembled Design

2.4.1 Head Harness Design

The head harness, shown in Figure 10, comprises a soft fabric base constructed from durable, comfortable materials that closely conform to the wearer's head shape. An optional attachment point is provided for a chin strap, facilitating additional stability and adjustable pressure should it be required.



Figure 10. Front-View of the Head Harness

Embedded within this fabric are custom PLA protective capsules designed with a twist-lock mechanism, securely stitched into the base material. These 3D-printed capsules, shown in Figure 11, house emitters (white-coloured) and sensors (black-coloured) arranged in triangular groups of three, spaced precisely 5 cm apart, from center to center. This layout enables data collection through both short and long-channel measurement capabilities. Mirroring ridges have been incorporated onto the capsule's curved surface to facilitate the twisting motion required for module installation and removal. Additionally, a ring featuring three walls firmly presses onto the PCB, ensuring stability and secure positioning throughout the assembly and operational phases.

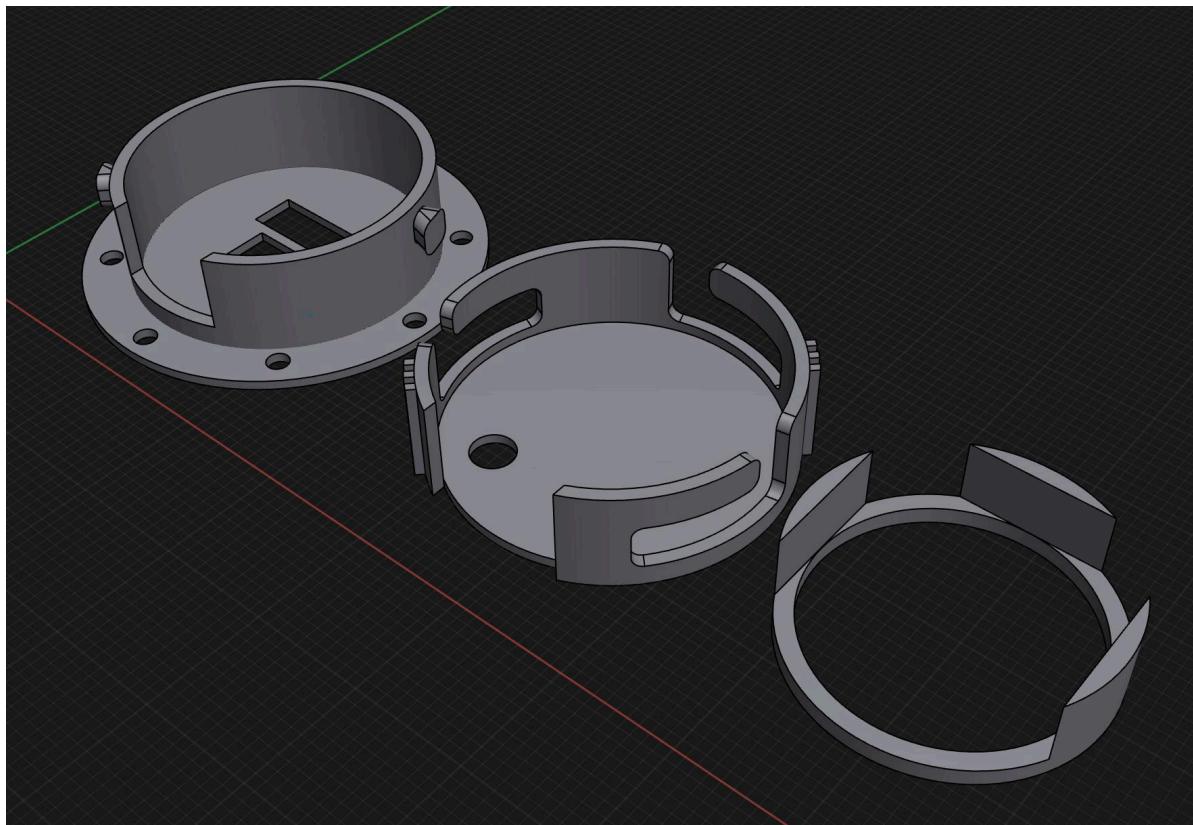


Figure 11. Emitter/Sensor Protective Capsule CAD Model

2.4.2 Body Harness Design

The body harness utilized in this system is deconstructed from a commercially available unit initially designed for mounting body cameras. It consists of a flexible and adjustable strap assembly that spans the upper torso, providing essential support and evenly distributing pressure across the shoulders of the wearer. For the purposes of this design, the original metal camera mounting plate has been repositioned to the user's back. Additionally, a custom mounting adapter was designed and 3D printed to replace the original camera mount, thus enabling precise integration with the rest of the mechanical components.

2.4.3 ECU Enclosure Design

Connectivity between the emitters, sensors, and the ECU is established via flexible cables, routed from the headpiece along the user's back to minimize interference with physical movement. The ECU is housed within a diamond-shaped 3D-printed enclosure, shown in Figure 12, designed to be removable and securely attachable to the body harness through a screw key-locking mechanism. Within the enclosure, the ECU PCB is elevated and centrally suspended by four structural support beams, effectively partitioning the enclosure into two

distinct compartments. The lower compartment is specifically designed to accommodate additional peripherals required by the system, including lithium-ion batteries and overhead cable assemblies. Rubber O-rings are positioned on the standoffs to firmly secure the ECU PCB in place. Furthermore, hollow beams integrated into the enclosure's top lid apply gentle downward pressure onto the standoff pillars, significantly reducing the potential for vertical displacement of internal components. Cable connections from the ECU to the emitters and sensors are purposefully routed through openings located on the upper two sides of the enclosure, optimizing the design symmetry for component stability during portable use.

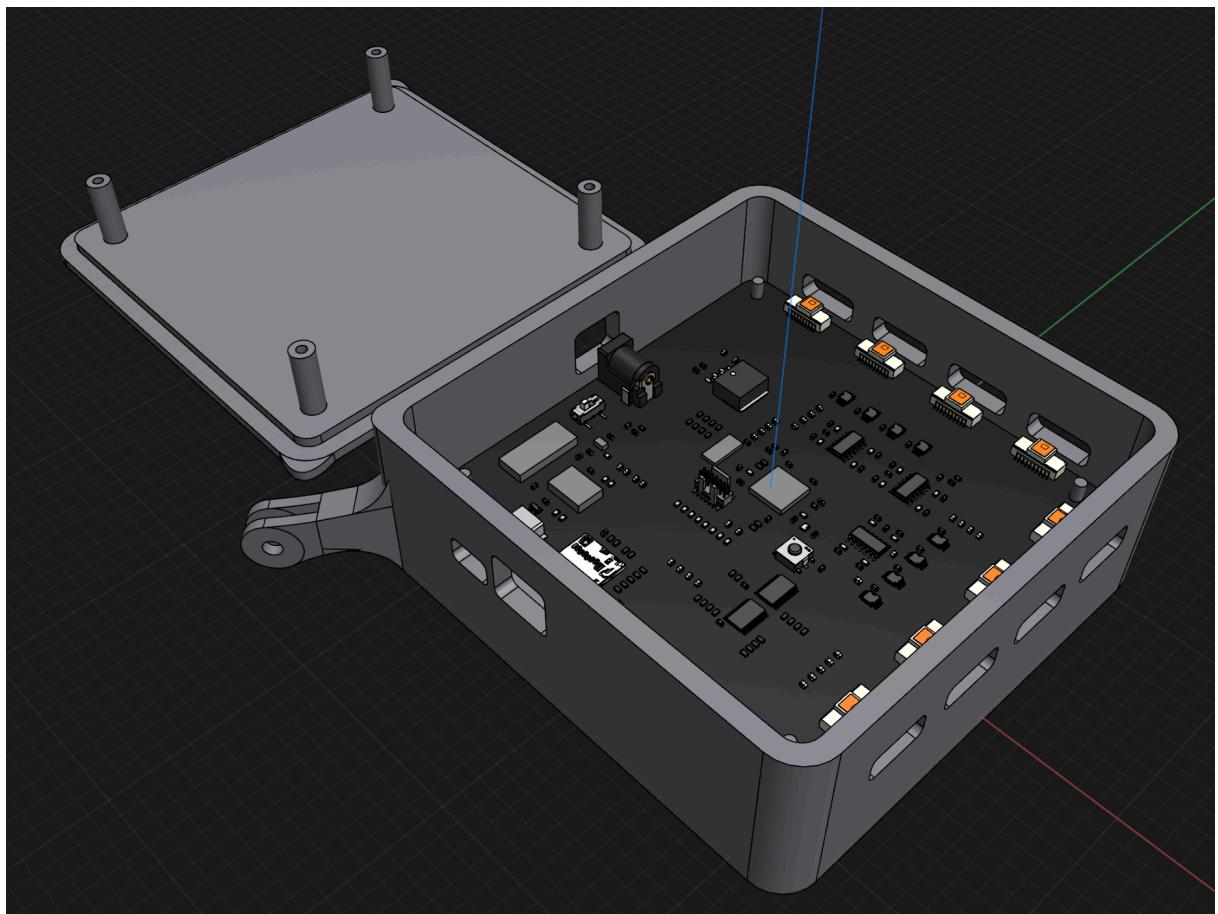


Figure 12. ECU Enclosure CAD Model

Overall, the final mechanical assembly emphasizes ease of wear, portability, and ergonomic design, strategically distributing the system's weight across the shoulders to mitigate neck strain. This distribution enhances usability, particularly during prolonged use, and effectively combines ergonomic considerations with technical functionality, thus ensuring precise data acquisition and optimal user comfort at the same time.

2.5 Software Design

The software system is designed to handle sensor data processing and visualization through data signal processing and calculations using the modified Beer-Lambert Law. It ensures seamless communication between the embedded firmware, data processing pipeline and graphical user interface (GUI) to provide analysis of the sensor readings.

2.5.1 Data Parsing

The data parsing module is responsible for decoding raw sensor packets received over USB from the ECU. Each packet contains 64 bytes of data, structured as eight rows of five values, corresponding to the eight sensor groups. The packet is parsed into an 8x5 array where each row contains five data fields: Group ID, Short Channel value, Long Channel value1, Long Channel value2, and Emitter Status. To ensure data accuracy and consistency, an accumulation buffer is used to maintain rolling averages of sensor readings. The buffering process follows these steps:

1. Initialize buffers to store incoming sensor values, and track the number of valid samples per sensor group
2. Parse incoming data packets and identify active emitter
3. If the emitter remains unchanged, individual channel values for each group are summed
4. If the emitter changes, the average values corresponding to the previous emitter are calculated before clearing the buffer for the new emitter
5. The averaged sensor data (8 x 5 array) is formatted and used for further processing of the modified Beer-Lambert Law (mBLL).

2.5.2 Data Processing and Interpretation

With the assistance of the NIRSimple library [6], the data processing pipeline operationalizes the theoretical foundations of fNIRS by converting parsed raw measurements into physiologically interpretable signals. It first interleaves the raw intensity readings acquired at the two wavelengths into a unified sample vector representing all sensor channels. This sample is then appended to a rolling buffer, which provides a temporal context for further processing. The raw intensities are converted to optical density (OD) changes following the relationship:

$$\Delta OD = -\log_{10}(\frac{I_t}{I_{avg}}),$$

Where I_t is the instantaneous intensity and I_{avg} is the mean intensity for the signal channel [2]. This logarithmic conversion is essential because it linearizes the light attenuation, making the signal directly proportional to the chromophore concentrations.

Following the ΔOD transformation, the modified Beer-Lambert law (mBLL) is applied to relate the changes in OD to the concentration changes of oxygenated (HbO_2) and deoxygenated (HbR) hemoglobin [2]. The mBLL accounts for tissue scattering by incorporating the differential path length factor (DPF) and the source-detector distance (L), and it uses known extinction coefficients (ϵ) to solve a set of linear equations that map to the two wavelengths (λ):

$$\Delta OD_{(\lambda 1)} = L * DPF * [\epsilon_{HbO_2(\lambda 1)} * \Delta HbO_2 + \epsilon_{HbR(\lambda 1)} * \Delta HbR]$$

$$\Delta OD_{(\lambda 2)} = L * DPF * [\epsilon_{HbO_2(\lambda 2)} * \Delta HbO_2 + \epsilon_{HbR(\lambda 2)} * \Delta HbR]$$

In this implementation, the DPF value is determined using an established age-dependent model as described by Scholkmann and Wolf [7] for a 22-year-old individual, while the extinction coefficients are adopted from the seminal study by Wray et al. [8]. With this information, solving the mBLL system of linear equations then effectively decouples the contributions of HbO_2 and HbR to the overall light absorption.

To further enhance the reliability of the results, correlation-based signal improvement (CBSI) is employed. CBSI leverages the typically strong inverse correlation between HbO_2 and HbR signals to suppress motion artifacts and other noise sources [9]. This relationship can be modelled by the following two equations:

$$x_0 = \frac{1}{2}(x - \alpha y)$$

$$y_0 = -\frac{x_0}{\alpha}$$

Where x is the measured ΔHbO_2 , y is the measured ΔHbR , x_0 is the corrected ΔHbO_2 , y_0 is the corrected ΔHbR , and α is the ratio of the standard deviation of the measured signals.

CBSI is proven to reflect strong real-time signal improvements, especially when exponential moving average filtering is applied [9], making it ideal for this software data processing pipeline with a rolling buffer implementation.

The final output of the processing algorithm is a 48-element vector, with each consecutive pair corresponding to the estimated concentrations of ΔHbO_2 and ΔHbR for a single sensor channel. This vector provides a comprehensive, real-time profile of hemoglobin dynamics that can be used for further analysis or visualization.

2.5.3 Web GUI (Graphical User Interface)

The Web GUI provides an interactive interface for real-time monitoring and visualization of fNIRS sensor data. This interface integrates several interconnected modules to deliver feedback on brain activation patterns. At the core of this setup is a backend Flask server supported by Flask-SocketIO, responsible for handling real-time, bidirectional communication between the firmware processing layer and end users. This setup allows continuous streaming of processed sensor data, ensuring that visualizations on the client side are updated with minimal delay as new data arrives, facilitating real-time analysis of neural activity patterns.

On the server side, the resulting outputs of the data processor stream — namely, optical density-based hemoglobin concentration changes — are continuously emitted to connected clients. This data is structured as packets containing concentration values across 24 sensor channels, reflecting real-time brain activity. Upon receipt, the client-side system maintains a rolling data queue, storing the most recent 20 packets. This preservation of short-term historical data enables users to identify patterns, trends, and fluctuations over time.

The front end utilizes HTML5 in combination with Bootstrap for responsive layout, Plotly.js for 3D rendering of dynamic graphs, and jQuery for asynchronous communication. The layout of the GUI is divided into panels for ease of user navigation. The left panel of the interface is dedicated to displaying a dynamic 3D brain mesh, constructed from anatomical datasets provided by the BrainNet Viewer library [10]. Specifically, the BrainMesh_Ch2_smoothed.nv file supplies detailed geometric data that defines the 3D brain surface, shown in Figure 13 below. The Automated Anatomical Labeling (AAL) atlas (aal.nii) file that contained the mapping of coordinates to defined brain regions was also utilized to map sensor positions precisely onto distinct brain regions.

3D Brain Mesh with Sensor Nodes

Sensor Type: □ Emitter, ■ Detector

Highlights: ■ 940nm wavelength, ■ 660nm wavelength, ■ Sensor Group Mapping

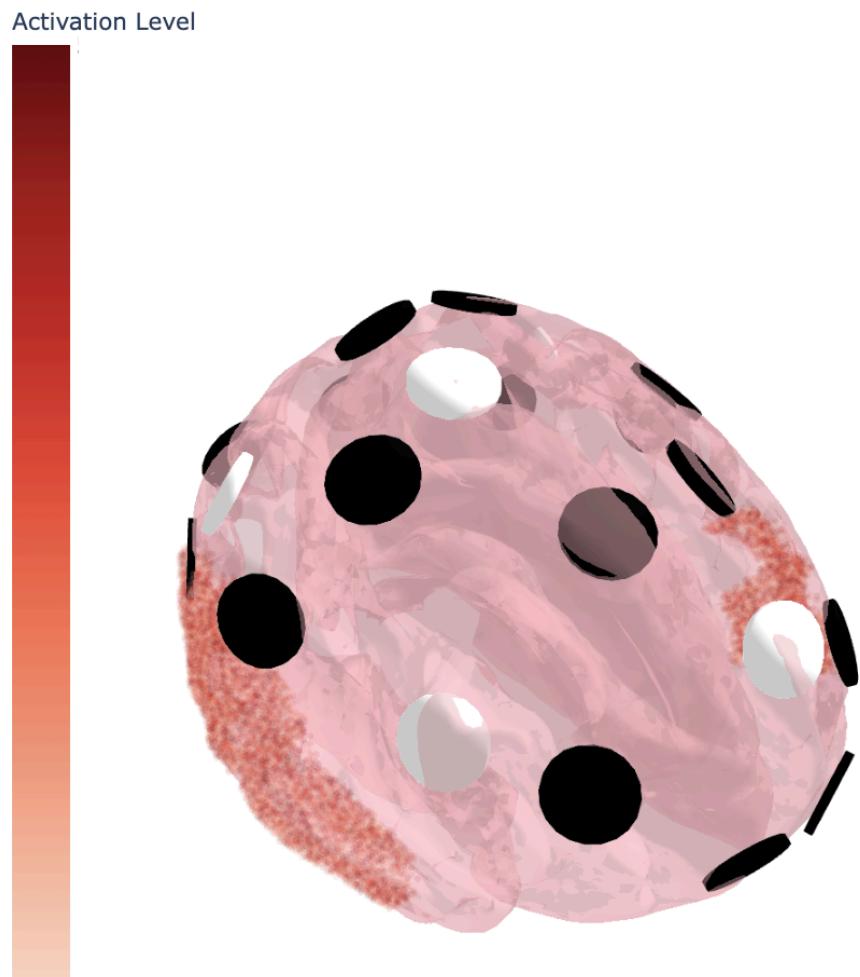


Figure 13. 3D Visualization of Activated Regions of the Brain

On the bottom panel are the activation plots presenting time-series data from each sensor group, illustrating changes in neural activity over successive data packets, as seen in Figure 14 below. This functionality, combined with jQuery for asynchronous data fetching and dynamic Plotly graphs, creates an engaging and responsive user experience.

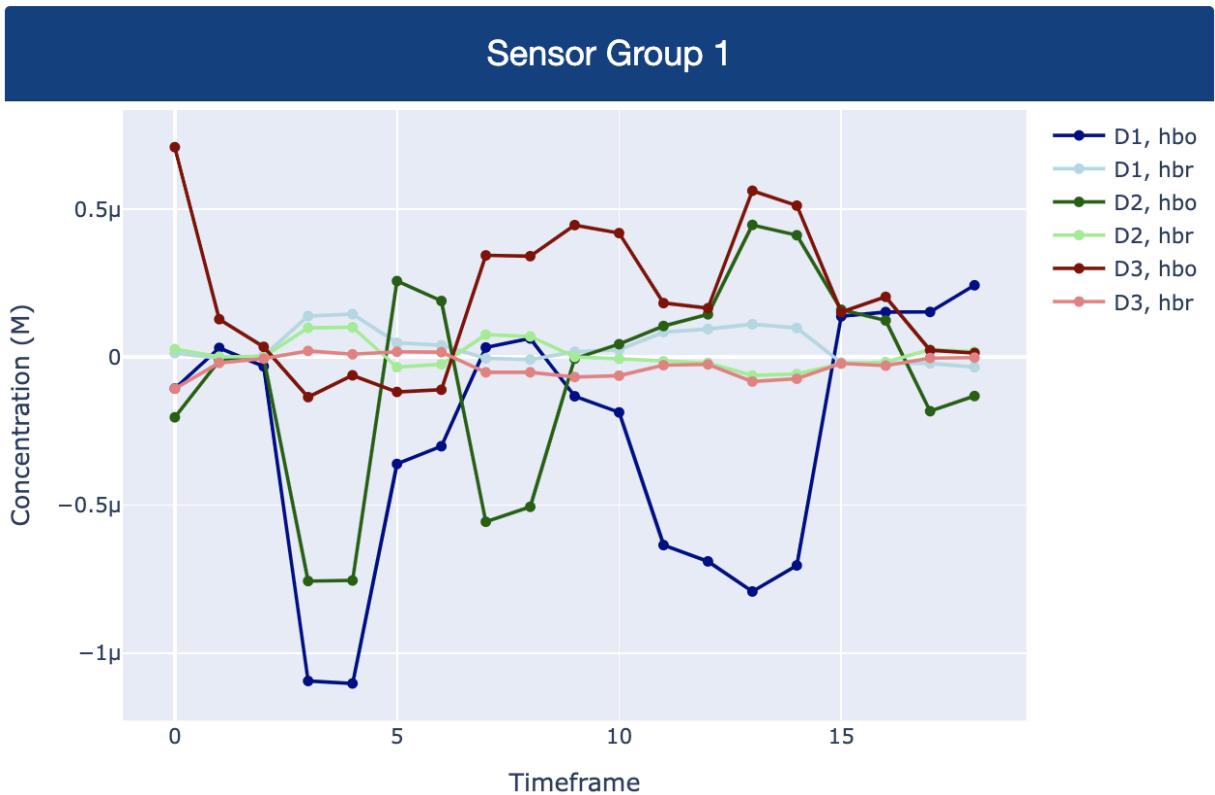


Figure 14. Sample Time-Series Data for Sensor Group 1

2.5.4 User Controls

The Sensor Controls module, integrated within the web-based GUI, provides real-time interaction with the underlying hardware. Through intuitive, web-accessible control panels shown in Figure 15 below, users can dynamically manipulate sensor configurations – such as managing emitter states and multiplexer (MUX) selections – directly from the control panel. These changes are immediately reflected in the UI and the components provide control over emitter behaviour, allowing users to set emitters into various operational modes as well as custom user-defined configurations. The PWM controls further allow precise activation of individual emitters at each wavelength (660 nm or 940 nm).

Control Panel

MUX Control

Override Enable:

Control State: **MUX Disabled**

Emitter Control

Override Enable:

Control State: **USER_CONTROL**

Emitter	940nm	660nm
Emitter 0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Emitter 1	<input type="checkbox"/>	<input type="checkbox"/>
Emitter 2	<input type="checkbox"/>	<input type="checkbox"/>
Emitter 3	<input type="checkbox"/>	<input type="checkbox"/>

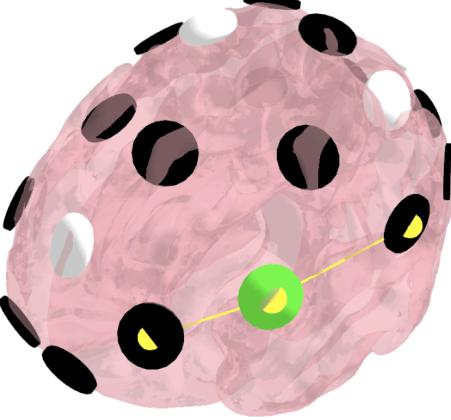


Figure 15. MUX and Emitter Control Panel

Behind these web controls lies a robust communication mechanism facilitated by Python's pySerial library. User selections and adjustments made in the GUI are immediately translated into serial commands sent directly to the physical sensors, ensuring rapid synchronization between the GUI's state and the hardware configuration.

In addition to providing direct hardware control, the interface includes real-time visual feedback that reflects current sensor settings. Active emitters are dynamically highlighted in distinct colours on the GUI's brain mesh model, clearly indicating their operational status as well as wavelength mode.

2.6 Final Design Assessment

The final design successfully delivers a fully custom and affordable fNIRS device, integrating hardware, firmware, mechanical and software components to meet the project's functional requirements. The effectiveness of the final design is demonstrated through its high-speed data acquisition and processing pipeline, allowing for minimum continuous sampling of 1.6 kHz per sensor channel, with effective hardware and software filtering to ensure high signal integrity. The ergonomic head-mounted structure ensures comfortable and

stable placement of the sensor and the ECU. Furthermore, an intuitive software interface provides data visualization, sensor control, and post-processing capabilities.

Despite these achievements, several areas for future improvement have been identified. Currently, data transmission only supports real-time logging via USB. The current hardware design also allows for SD card logging which the firmware for it has yet to be developed. Integrating an SD card interface would allow for onboard data logging. This would be beneficial for offline data storage when USB communication is unavailable, logging sensor data without the need for continuous host communication, post-processing and analysis, enabling performance optimization.

The PCB designs for the ECU and sensor boards can also be revised to optimize board space and reduce overall size and weight. By replacing larger components with smaller alternatives, and further optimizing component placement, we can achieve a smaller and more efficient board layout without sacrificing performance.

3. Testing and Verification

To ensure the designed fNIRS device satisfied the project goals and requirements, a comprehensive validation process was conducted where individual components underwent independent testing before gradually being integrated and tested as a whole.

3.1 Sensor Hardware Testing

The sensors were tested individually with lab equipment to ensure that near-infrared light detection and emission were achievable with applicable hardware specifications such as a minimal noise level. The table below summarizes the verification process.

Table 4: Hardware bring-up verification table for sensors

Test Item	Description	Verification Method	Verification Result
Emitter Functionality	Verify that 660 nm and 940 nm emitter turn on/off as controlled by the firmware	Switch emitters on and off and visually verify emitted light. For 940 nm, use an oscilloscope to verify conduction	[PASS], emitter enabling is observed
Emitter Intensity	Verify emitter	Adjust the duty cycle in	[PASS], emitter

Control with PWM	intensity modulation via PWM driver	the firmware, and verify its correspondence with brightness changes	intensity control is observed
Signal Validation	Verify that the sensor is capable of detecting heart rate	Place sensor on finger and compare readings to a commercial heart rate sensor. Adjust amplifier gain values by swapping passive components as needed. Do for both emitter wavelengths.	[PASS], the gain of the amplifiers were finalized empirically through this test as described in the Final Design section
Noise Level Tests	Measure noise levels of the sensor outputs	Place the sensors in a light-proof box and monitor signal fluctuations; measure the output signal and compute the noise level.	[PASS], see Appendix F for results

3.2 ECU Hardware/Firmware Testing

The bring-up of the ECU involved rigorous tests with firmware control to validate all individual circuits and devices before integrating them together with higher-level control logic. The table below outlines the hardware bring-up process to the full system-level firmware implementation.

Table 5: Hardware bring-up verification table for ECU

Test Item	Description	Verification Method	Verification Result
Blinky	Be able to program the microcontroller on the ECU, running a simple program that blinks an LED and writes to the serial port	Flash test firmware and observe LED blinking and serial output via UART.	[PASS], observed LED blinking and serial port messages
GPIO Expander	Be able to control	Send commands via I2C	[PASS], successful

control	the GPIO expander device through I2C interface (fast mode) from the microcontroller and drive individual GPIOs	in the firmware to the GPIO expander to switch MUX channels and ensure the corresponding sensor is selected. Measure the voltage at the expected MUX output pin to confirm it corresponds to the selected channel.	I2C communication established, verified correct GPIO states
PWM control	Be able to have full duty, phase, and frequency control of PWM channels on the device through I2C interface (fast mode)	Configure different PWM settings in firmware by modifying duty cycle and phase shift, observe and measure changes in output signals using an oscilloscope	[PASS], PWM output observed at expected duty cycle and phase shift values
ADC verification and calibration	Ensure the ADC accurately captures sensor signals and appliances calibration values	Input known voltage levels and verify ADC readings match expected values. Calibrate the values as needed with empirically determined scale and offset values	[PASS], see Appendix G for calibration tests
Multiplexing Sequencing logic	Ensure firmware can cycle through multiplexers at a fixed frequency	Use hardware profile timer to measure the time it takes to switch between channels and measure switching with an oscilloscope	[PASS], verified stable multiplexer cycling at expected timing intervals
USB communication and packet design	Verify that sensor data is correctly formatted and transmitted over USB	Setup USB transmission in the firmware and log output packets using a PC with hardcoded values	[PASS], confirmed correct packet structure and reliable data transmission
Sensing and Multiplexer synchronization	Ensure ADC samples the correct sensor channels in	Setup 3x 50Hz PWM signals with varying duty cycles and phase shifts at	[PASS], successfully validated ADC readings and

	sync with the multiplexer switching	the sensor inputs and confirm logged data packets through USB can read corresponding input signals	synchronization with MUX switching
Emitter Control Logic	Validate state machine operation for emitter cycling between 660 nm and 940 nm wavelengths	Log emitter states in firmware and observe expected switching behaviour between the two wavelengths.	[PASS], observed correct emitter toggling
ECU System testing	Integrating all components together to ensure no timing mismatches, bugs, or unexpected behaviours	Run the ECU with its running software for 5 hours and ensure all functions are working as expected	[PASS], system operates continuously with no unexpected errors or behavior

3.3 Software Verification

Software verification revolved around our custom program running on the PC and ensuring post-processing, analysis, display, and user control were reliable. The table below outlines the software verification.

Table 6: Software verification table

Test Item	Description	Verification Method	Verification Result
Data Parsing	Verify accurate parsing of sensor data packets received via USB	Input predefined sensor data packets via USB and check parsed output against known values	[PASS], parsed data matches predefined input
OD and mBLL Calculation	Verify accurate application of OD and mBLL software processing steps to derive Δ	Perform Δ OD and mBLL calculations by hand and compare with the software-processed results. Verify the inverse	[PASS], Δ HbO and Δ HbR concentration outputs match theoretical calculations and

	HbO and Δ HbR concentrations	relationship between Δ HbO and Δ HbR concentrations.	exhibit inverse relationships.
Real-Time Data Streaming	Validate real-time streaming of processed data from backend Flask-SocketIO server to clients	Confirm consistent streaming by monitoring client-side reception of real-time processed data packets from the server	[PASS], data streaming maintains integrity and is continuous at regular intervals
3D Brain Visualization	Confirm accurate rendering and real-time updates of 3D brain mesh and sensor activations on GUI	Visually inspect GUI during active data streaming, ensuring volume rendering correctly reflects current states	[PASS], GUI correctly displays real-time sensor activation levels on 3D brain mesh
Time-Series Visualization	Ensures time-series plots accurately reflect live sensor data	Stream known sequences of sensor data and visually confirm that displayed plots match expected patterns	[PASS], time-series plots accurately match streamed data
Sensor Control	Verify GUI controls effectively translate user inputs into hardware control commands via serial communication	Manipulate GUI controls (ex. Emitter states, PWM settings) and monitor resulting serial commands sent to the hardware	[PASS], GUI controls reliably generate correct serial commands, confirmed by hardware response

3.4 Product Level Testing

After individual testing was completed, systems were integrated together, from bench-level tests to building up the fully assembled product. The table below outlines the verification that aligns with our project requirements, reporting product-level verification.

Table 7: Verification table

ID	Project Requirement	Verification Method	Verification Outcome
1.0 Data Sample Rate	Ability to read 50Hz signal with minimal noise	Input a 50Hz, 3.3V sinusoidal waveform with a function generator into the ADC channel(s) and log ADC readings. Plot the output and expect to see the same waveform as the input. Repeat test for different frequencies.	[PASS], the output waveform matches with the input sinusoidal waveform at 50Hz
2.0 Acquiring data signal	Detector optode voltage output matches photodiode light absorption behaviour	Apply a controlled light intensity to the emitter and verify that the photodiode sensor outputs the expected voltage level. Use an oscilloscope/multimeter to measure the photodiode's voltage response, comparing it to expected values based on emitter intensity.	[PASS], the responses of ADC readings due to changing light intensity were observed on the waveform using an oscilloscope / plotted in software. A baseline test of capturing heartbeats was conducted.
	ADC reading reflects voltage being read	Sweep voltage values from 0 to 3.3V, in increments of 0.2V on the ADC input and log digital readings. 0-3.3V should linearly map to 0-4095, accurate to 1%. If accuracy falls outside of the 1% range, add scale and offset values to the digital data in firmware based on results, and repeat the test (i.e. calibrating the ADC).	[PASS], the fluctuation of ADC readings was observed on the waveform using an oscilloscope / plotted in software. A baseline test of capturing heartbeats was conducted.

3.0 Outputting data signals to PC	Read serial data from ECU on PC	Connect the PC to the ECU and confirm that data is transmitted successfully by verifying the integrity of data packets received over the serial interface. Run tests with known data sequences to check for consistency and completeness.	[PASS], received matching transmitted data packets on PC.
	Plotting	Connect the PC to the data output source and ensure that signals are getting transmitted correctly. Plot the output on a graph, checking for expected trends and proper representation of signals.	[PASS], data received plotted using Excel, matplotlib, and displayed onto the web GUI. Correct signal behaviours were observed.
4.0 Cost	Net material/fabrication cost of the fully assembled device	Accumulate the cost of the fully assembled device and compare with the target, <\$1000	[PASS], the total cost of the assembled device is ~\$750.
5.0 Physical Properties	Wearability of the assembled device	Physically wear the assembled device to check for freedom of movement and the integrity of the components	[PASS], the device was worn, transported during usage, and afterward removed with ease and comfort.
	Non-invasive	Check if the device penetrates the skin, skull, or induces any other physical harm and sickness to the user.	[PASS], the device has been operating safely without causing any injury, or physical strain, to the user.

4. Summary and Conclusions

The project successfully designed and validated a low-cost, ergonomic Functional Near-Infrared Spectroscopy (fNIRS) device aimed at extending its applications beyond clinical settings. By leveraging the modified Beer-Lambert Law, the final design effectively monitors brain activity in real-time through measuring changes in cerebral blood oxygenation levels. The final system utilized commercially available near-infrared emitters operated at 660 nm and 940 nm wavelengths, in combination with photodiode sensors. These components were integrated into a custom-designed Electrical Control Unit (ECU), complemented by firmware, mechanical structures, and a user-friendly software interface.

Comprehensive validation testing was conducted at multiple levels – sensor, ECU, software, and the fully assembled device – to ensure that the design met all project goals and technical requirements. At the sensor level, tests confirmed reliable emitter operation, PWM-based intensity control, and low-noise signal acquisition. A key validation step involved detecting heart rate signals by placing the sensors on a finger and comparing its output to a commercial heart rate monitor. The successful detection of heartbeats confirmed that the sensor array and amplification circuitry could capture real physiological signals, demonstrating the device's sensitivity and ability for monitoring cerebral hemodynamics.

ECU testing verified all critical subsystems including GPIO, PWM, and ADC functionality. The ADC reliably captured voltage sweeps from 0 to 3.3V with high level of accuracy across the entire input range. Firmware-controlled multiplexing and emitter cycling were validated through oscilloscope measurements, confirming the correct timing and synchronization of data acquisition across multiple channels. Additionally, as part of the sensor validation process, the readings obtained from our sensors were compared to a commercial heart rate sensor, which verified the accuracy of our photodiode readings.

Software validation confirmed the system's ability to reliably parse USB data packets and stream data in real time using Flask-SocketIO. The processing algorithms for calculating optical density and applying the modified Beer-Lambert Law were also verified. The resulting ΔHbO and ΔHbR concentration changes aligned with theoretical expectations and showed the expected inverse relationship, demonstrating the systems' ability to track cerebral hemodynamic activity. A 3D brain model and time-series plots accurately reflected real-time

sensor data. User control of sensor parameters, such as emitter states and PWM settings, was also successfully validated.

Final product-level validation further demonstrated that the complete system met all established project requirements. The ADC captured and reproduced 50 Hz sinusoidal input signals with minimal noise, validating the system's real-time sampling rate. Controlled light intensity tests verified that photodiode voltage outputs aligned with expected absorption behaviour. Serial communication between the ECU and PC was reliable, with data successfully transmitted and verified through Excel, matplotlib, and the custom web UI.

Notably, the device achieved its target goal of affordability with the final assembled device costing approximately \$750, well below the targeted price of \$1000. Practical usage tests demonstrated that the device is comfortable and easy to wear, fulfilling the requirement of user wearability. The device is also confirmed to be non-invasive, ensuring it can be operated without causing physical discomfort or harm to the user.

Overall, the validated fNIRS system reliably measured physiological signals, including heart rate, and demonstrated its ability to provide changes in ΔHbO and ΔHbR concentrations. These results demonstrate a strong potential for fNIRS technology to be used in wider applications beyond traditional clinical settings, thereby fulfilling the project's core objective.

5. Future Work

While our current project meets all of our initial requirements and objectives, future research can explore several areas to enhance functionality and expand applications.

Future work could focus on leveraging Artificial Intelligence and machine learning techniques to improve interpretation of the fNIRS data. Currently, oxygenation level analysis relies on classical signal processing methods based on the modified Beer Lambert Law, but incorporating deep learning models could enable more accurate noise reduction and advanced cognitive state classification. AI-driven analysis could also help in identifying trends in brain activity, leading to more meaningful insights from the fNIRS data.

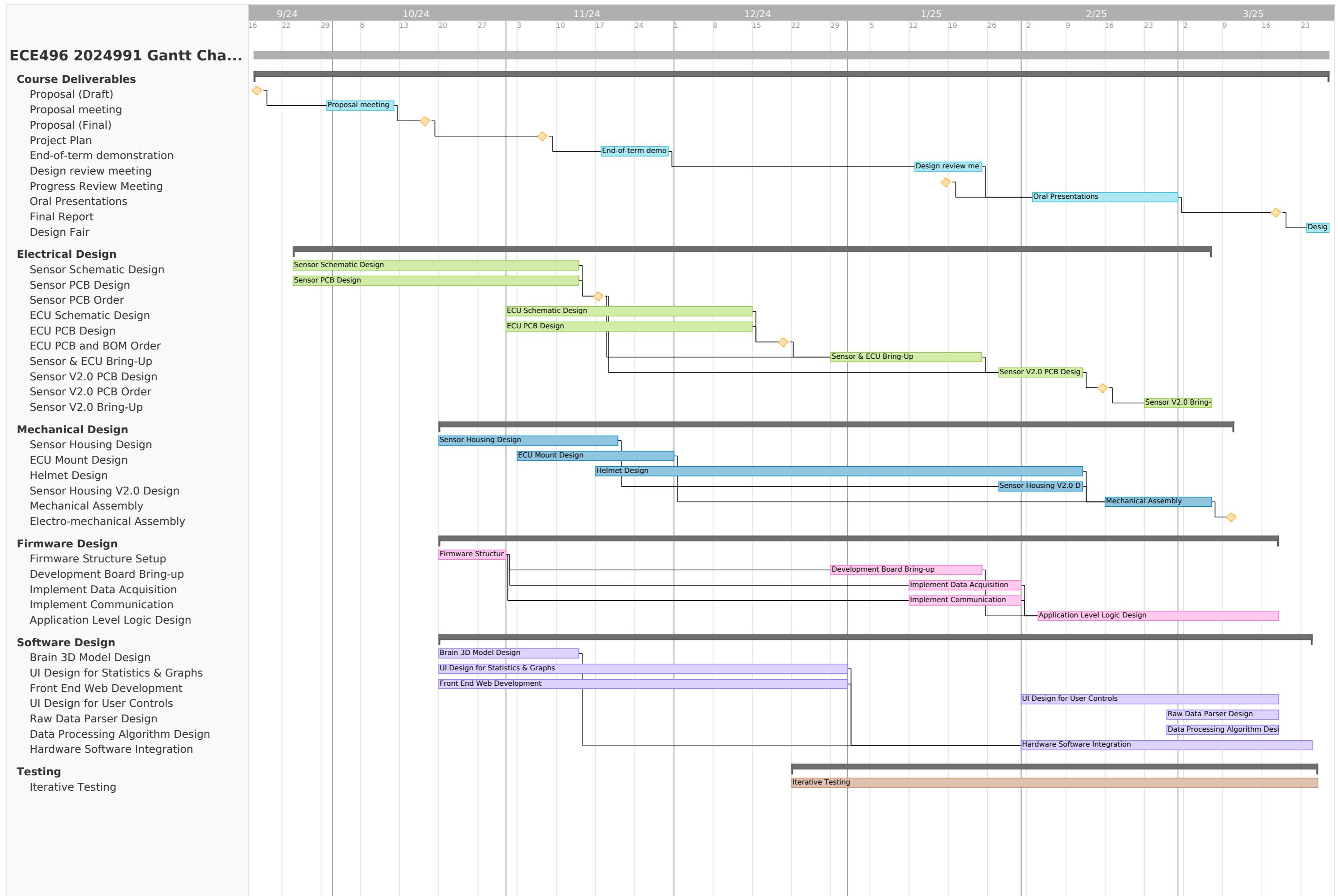
To maximise the impact of our design, future research could involve real-world validation in clinical and research settings. Collaborations with medical researchers or psychologists could refine usability and validate its effectiveness in different applications.

6. References

- [1] F. Herold, P. Wiegel, F. Scholkmann, and N. Müller, "Applications of functional near-infrared spectroscopy (fNIRS) neuroimaging in exercise–cognition science: A systematic, methodology-focused review," *Journal of Clinical Medicine*, vol. 7, no. 12, p. 466, Nov. 2018, doi: <https://doi.org/10.3390/jcm7120466>.
- [2] W. B. Baker et al., "Modified beer-lambert law for blood flow," *Biomedical Optics Express*, vol.5, no. 11, p. 4053, Oct. 2014. doi:10.1364/boe.5.004053
- [3] V. Quaresima and M. Ferrari, "Functional near-infrared spectroscopy (FNIRS) for assessing cerebral cortex function during human behavior in natural/social situations: A concise review," *Organizational Research Methods*, vol. 22, no. 1, pp. 46–68, Jul. 2016. doi:10.1177/1094428116658959
- [4] J. Uchitel, E. E. Vidal-Rosas, R. J. Cooper, and H. Zhao, "Wearable, Integrated EEG–fNIRS Technologies: A Review," *Sensors*, vol. 21, no. 18, p. 6106, Sep. 2021, doi: <https://doi.org/10.3390/s21186106>.
- [5] F. Klein, "Optimizing spatial specificity and signal quality in fNIRS: an overview of potential challenges and possible options for improving the reliability of real-time applications," *Frontiers in Neuroergonomics*, vol. 5, Jun. 2024. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fnrgo.2024.1286586/full>. [Accessed: Mar. 20, 2025].
- [6] J. Benerradi, "NIRSimple," NIRSimple - NIRSimple 0.1.6 documentation, <https://hanbnrd.github.io/NIRSimple/index.html> (accessed Mar. 1, 2025).
- [7] F. Scholkmann and M. Wolf, "General equation for the differential pathlength factor of the frontal human head depending on wavelength and age," *Journal of Biomedical Optics*, vol. 18, no. 10, p. 105004, Oct. 2013. doi:10.1117/1.jbo.18.10.105004
- [8] S. Wray, M. Cope, D. T. Delpy, J. S. Wyatt, and E. O. R. Reynolds, "Characterization of the near infrared absorption spectra of cytochrome aa3 and haemoglobin for the non-invasive monitoring of cerebral oxygenation," *Biochimica et Biophysica Acta (BBA) - Bioenergetics*, vol. 933, no. 1, pp. 184–192, Mar. 1988. doi:10.1016/0005-2728(88)90069-2
- [9] X. Cui, S. Bray, and A. L. Reiss, "Functional near infrared spectroscopy (NIRS) signal improvement based on negative correlation between oxygenated and deoxygenated hemoglobin dynamics," *NeuroImage*, vol. 49, no. 4, pp. 3039–3046, Feb. 2010. doi:10.1016/j.neuroimage.2009.11.050
- [10] "NITRC: BrainNet Viewer: Tool/Resource Info," N I T R C, <https://www.nitrc.org/projects/bnv> (accessed Mar. 20, 2025).

7. Appendices

Appendix A: Gantt Chart History



Appendix B: Financial Plan

Material	Priority	Cost per Unit	Quantity (# or hours)	Total Cost	Requires Funding	Kept/Paid for by Students
Sensor PCBs revision 1	1	1.37	25	34.25	No	Yes
fNIRS ECU + Assembly	1	115.965	2	231.93	Yes	No
Sensor PCBs revision 2 + Assembly	1	2.206	25	55.15	No	Yes
Sensor PCBs revision 2 bulk order + Assembly	1	2.7236	25	68.09	Yes	No
USB connector and switch for ECU	1	19.96	1	19.96	No	Yes
Wires	1	22.15	1	22.15	No	Yes
Wires	1	18.76	1	18.76	No	Yes
Wires	1	35.32	1	35.32	No	Yes
PLA Filament	1	30.5	1	30.5	No	Yes
Thick Helmet Liner	1	15.36	1	15.36	No	Yes
Thin Helmet Liner	3	1.88	6	11.28	No	Yes
TPU Filament	2	35.01	1	35.01	No	Yes
Chest Mount Strap Harness	1	25.98	1	25.98	No	Yes
NUCLEO-L476RG	2	24.4	1	24.4	No	Yes
Sensor breakout boards V1	1	1.1325	8	9.06	No	Yes
Sensor breakout boards V2	1	1.54375	8	12.35	No	Yes
Wires	1	25.75	1	25.75	No	Yes
Wires	1	17.43	1	17.43	No	Yes
Sensor PCBs surface mount components	1	3.3644	25	84.11	Yes	No
PCB surface mount components	1	121.25	1	121.25	Yes	No
Total				898.09		
Total Requiring Funding				505.38		

Student Labour		Cost per Unit	Quantity (# or hours)	Total Cost
Student 1		5	500	2500
Student 2		5	500	2500
Student 3		5	500	2500
Student 4		5	500	2500
Total Student Labour (Unfunded)				10000

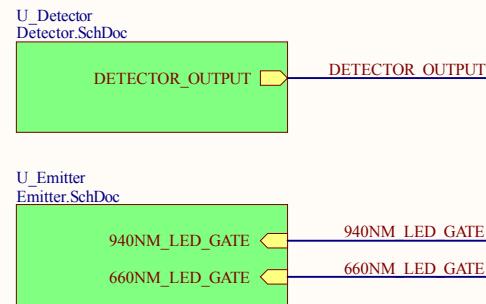
Total Cost of Project				10898.09
Total Cost Requiring Funding				505.38

Appendix C: Sensor Schematics and PCB Design

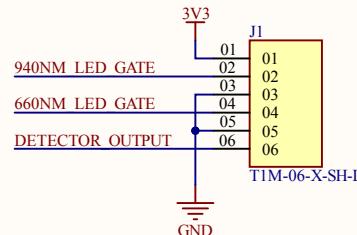
A

SENSOR MODULE - TOP

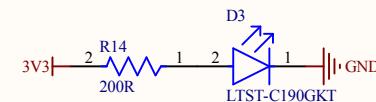
Sensing



Connector

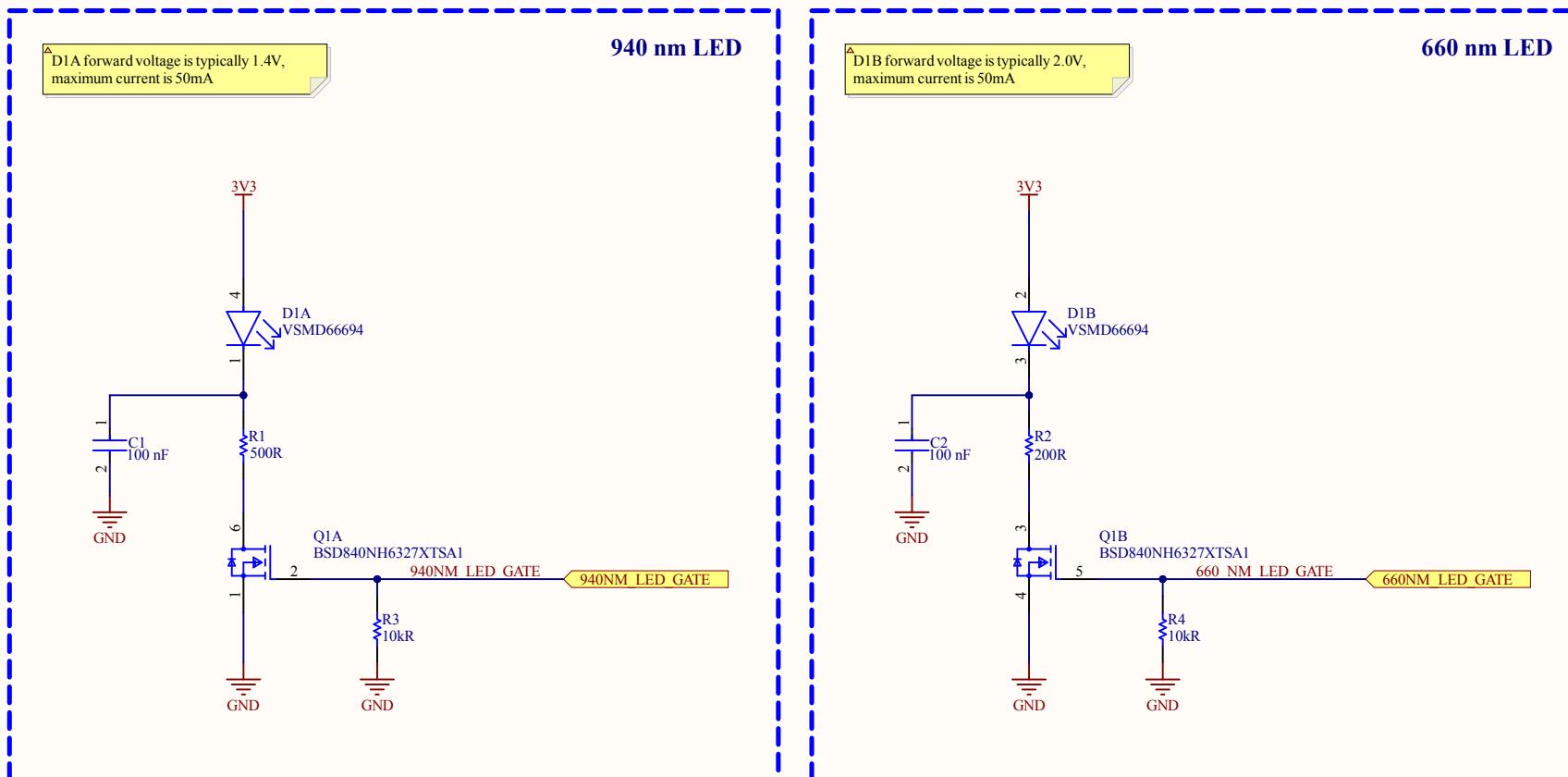


Power Good LED



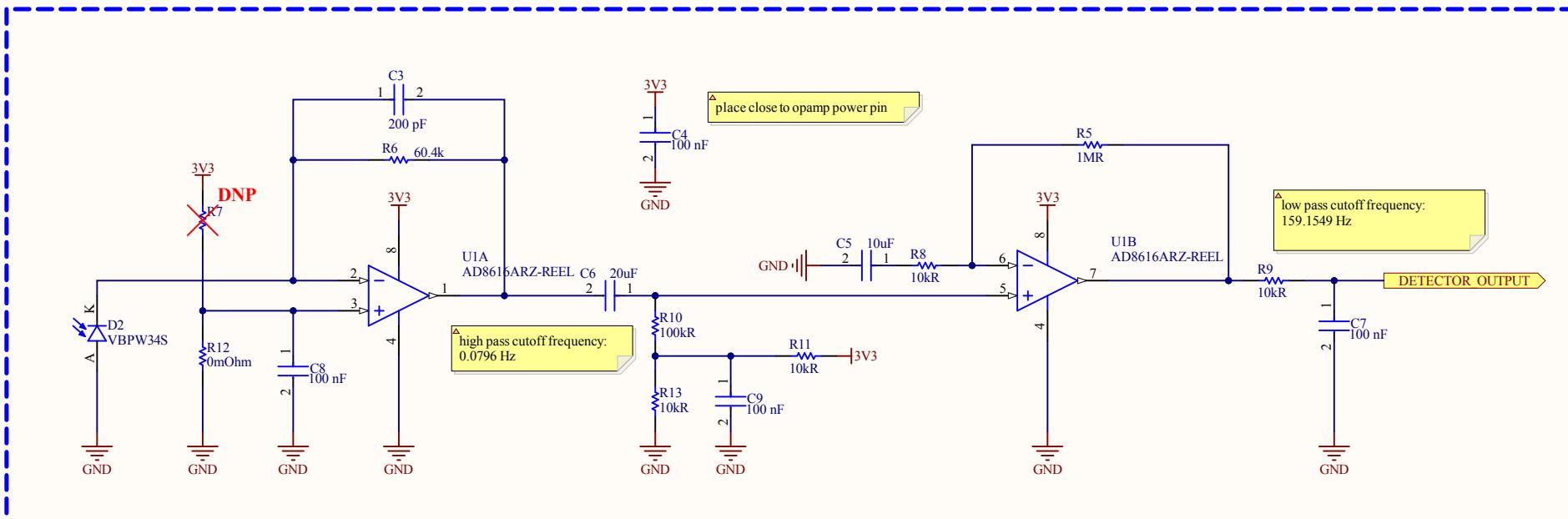
Sensor Module - Top		
Size	Number	Revision
A4	UofT Capstone Team 2024991	1
Date: 3/21/2025	Sheet 1 of 3	
File: C:\Mac\Top.SchDoc	Drawn By: Tony Kim	

OPTODE SOURCE EMITTER



Title Sensor Module - Emitter		
Size A4	Number <i>UofT Capstone Team 2024991</i>	Revision 1
Date: 3/21/2025	Sheet 2 of 3	
File: C:\Mac\...\Emitter.SchDoc	Drawn By: <i>Tony Kim</i>	

OPTODE DETECTOR



Transimpedance Amplifier Design Notes:

- Photodiode capacitance is typically 70pF
- Op-amp input capacitance is typically 2.5pF
- Photodiode max current is approximately 55uA ($3.3/55\mu\text{A} = 60\text{k}\Omega$)
- Bandwidth = $1/(2\pi * 60.4\text{k} * 200\text{p}) = 13.175\text{kHz}$

Title Sensor Module - Detector		
Size A4	Number <i>UofT Capstone Team 2024991</i>	Revision 1
Date: 3/21/2025	Sheet 3 of 3	
File: C:\Mac\Detector.SchDoc	Drawn By: <i>Tony Kim</i>	

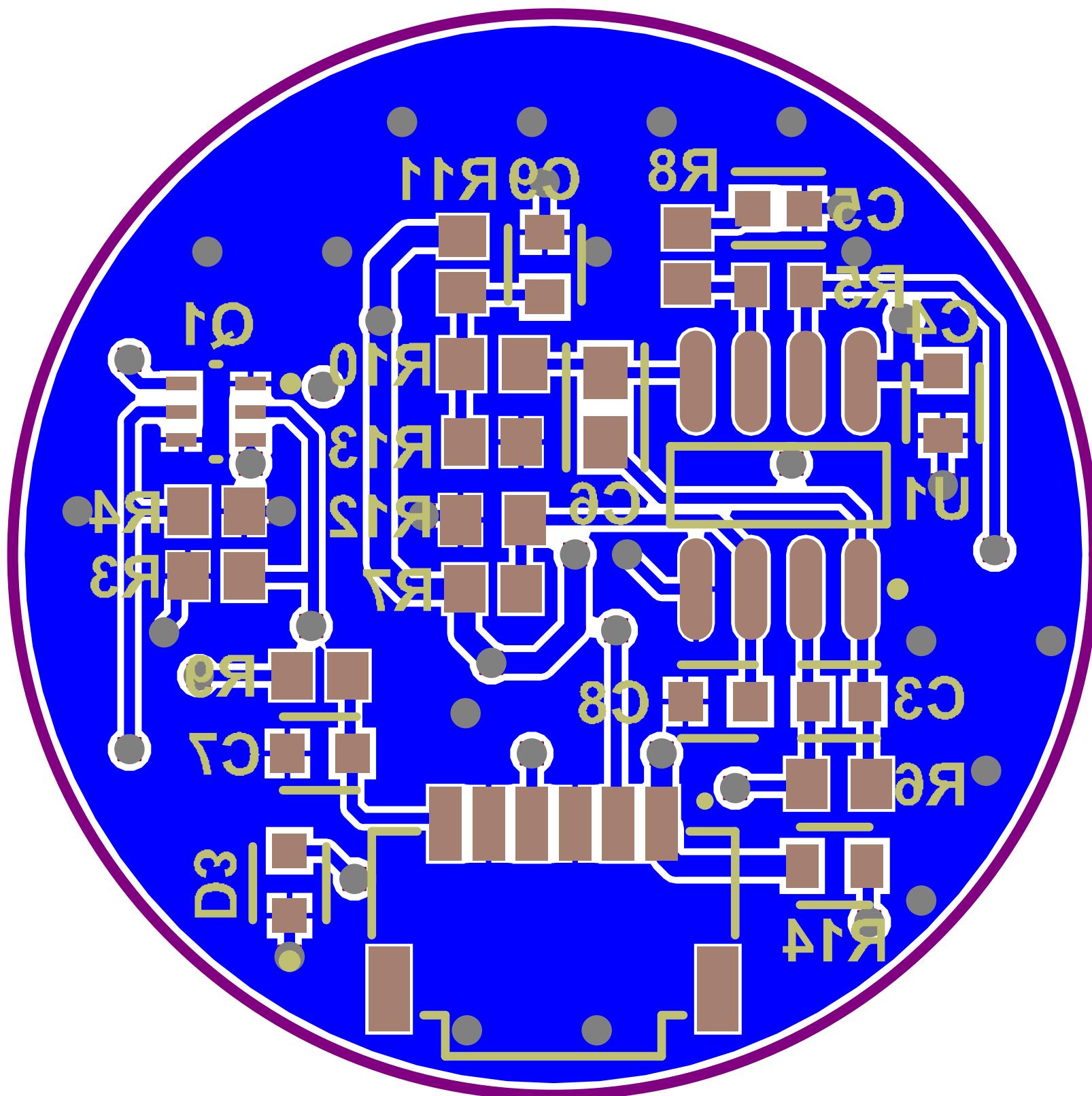
SENSOR MODULE

R2C2 C1R1

D1

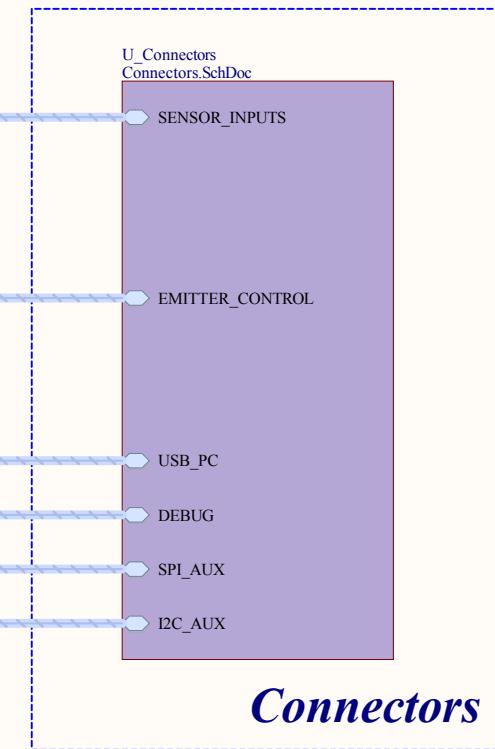
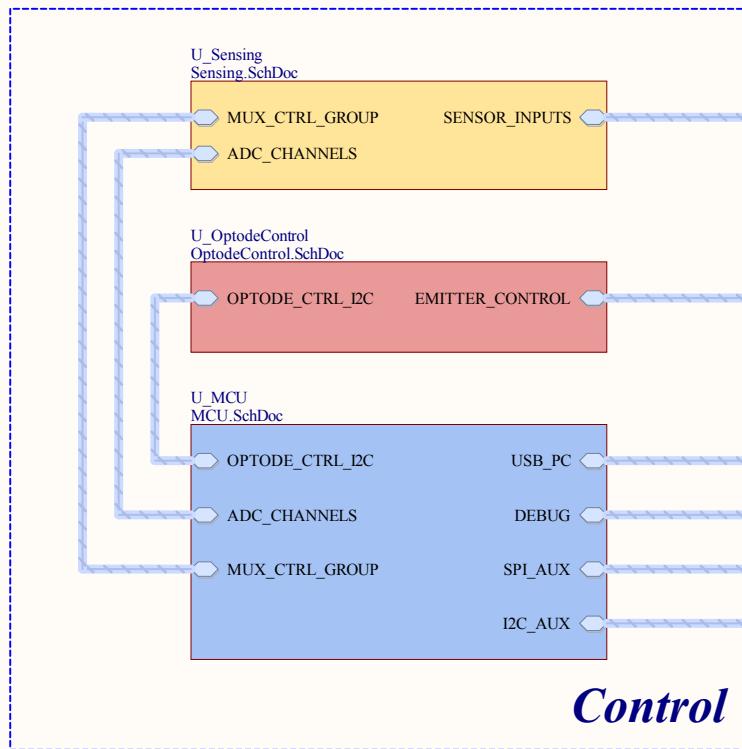
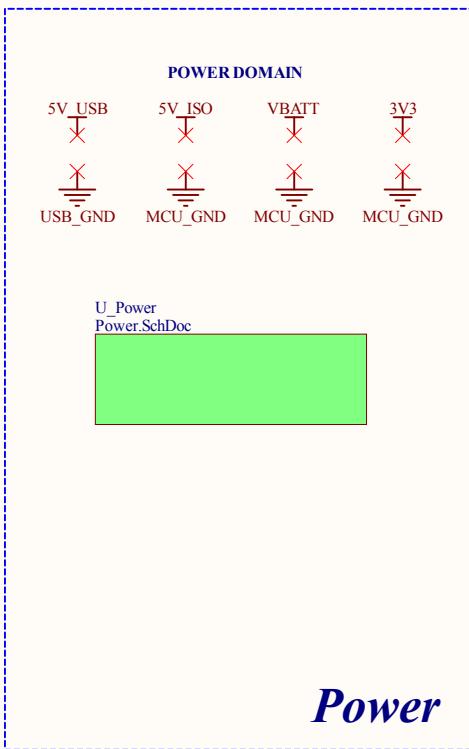
D2





Appendix D: ECU Schematics and PCB Design

fNIRS - TOP

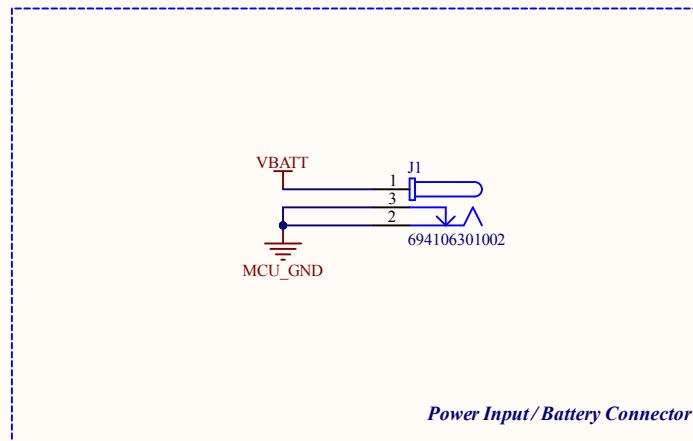
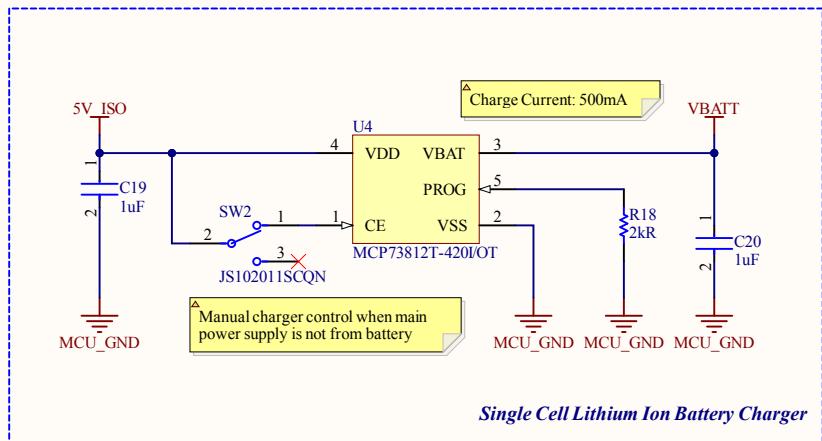
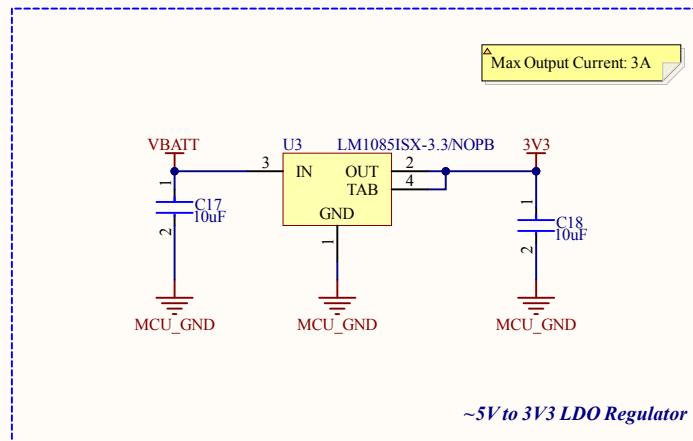
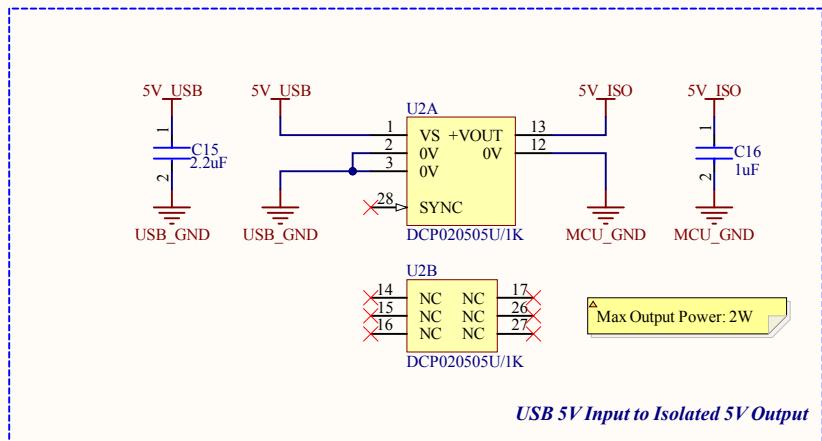


Legend

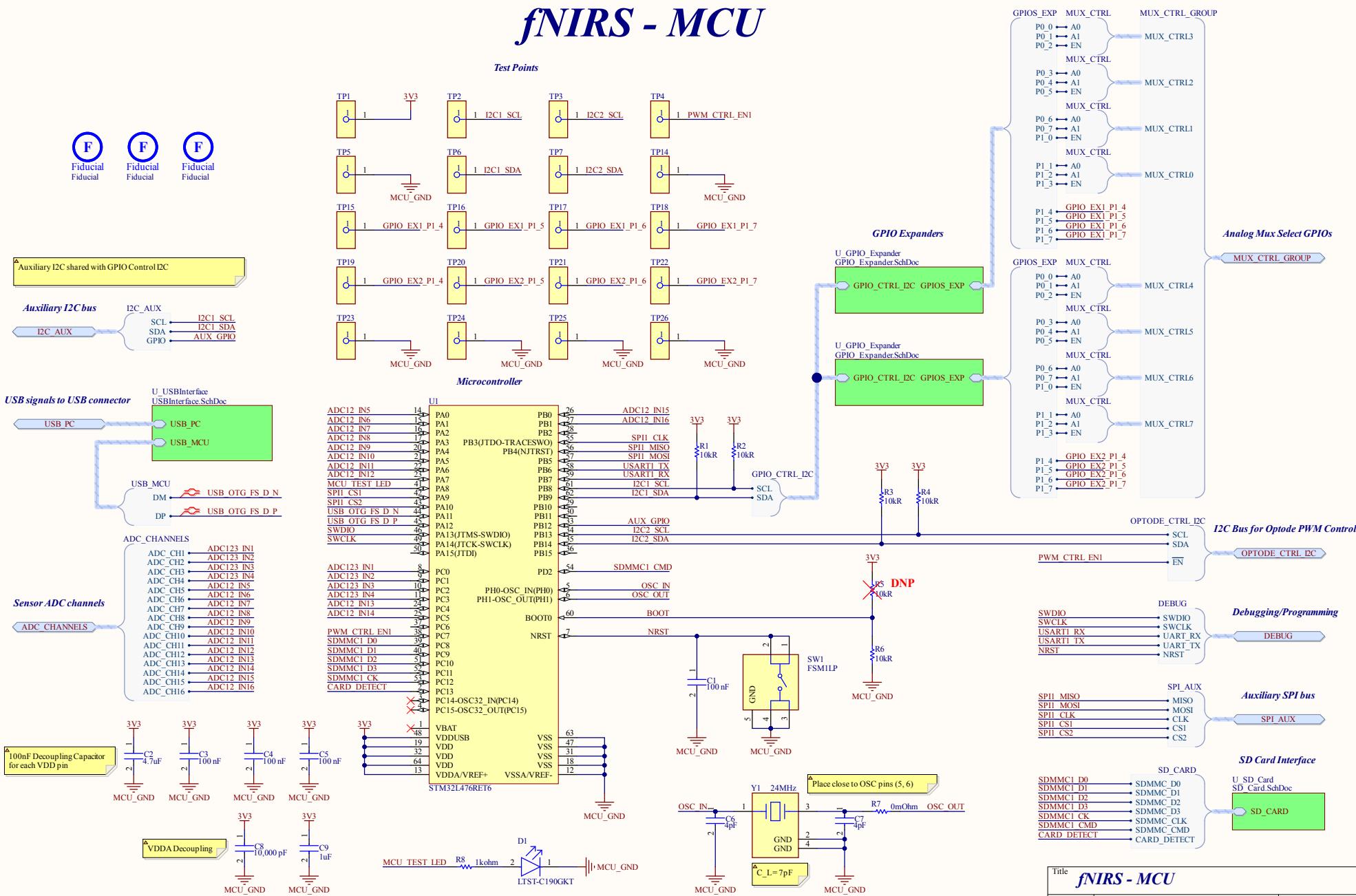


Title <i>fNIRS - Top</i>		
Size A4	Number <i>UofT Capstone Team 2024991</i>	Revision 1
Date: 3/21/2025	Sheet 1 of 13	
File: C:\Mac\Top.SchDoc	Drawn By: <i>Tony Kim</i>	

fNIRS - Power



Title		
Size	Number	Revision
A4	<i>UofT Capstone Team 2024991</i>	1
Date:	3/21/2025	Sheet 2 of 13
File:	C:\Mac\Power.SchDoc	Drawn By: <i>Tony Kim</i>



A

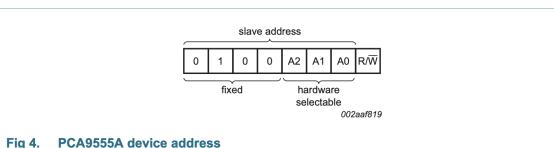
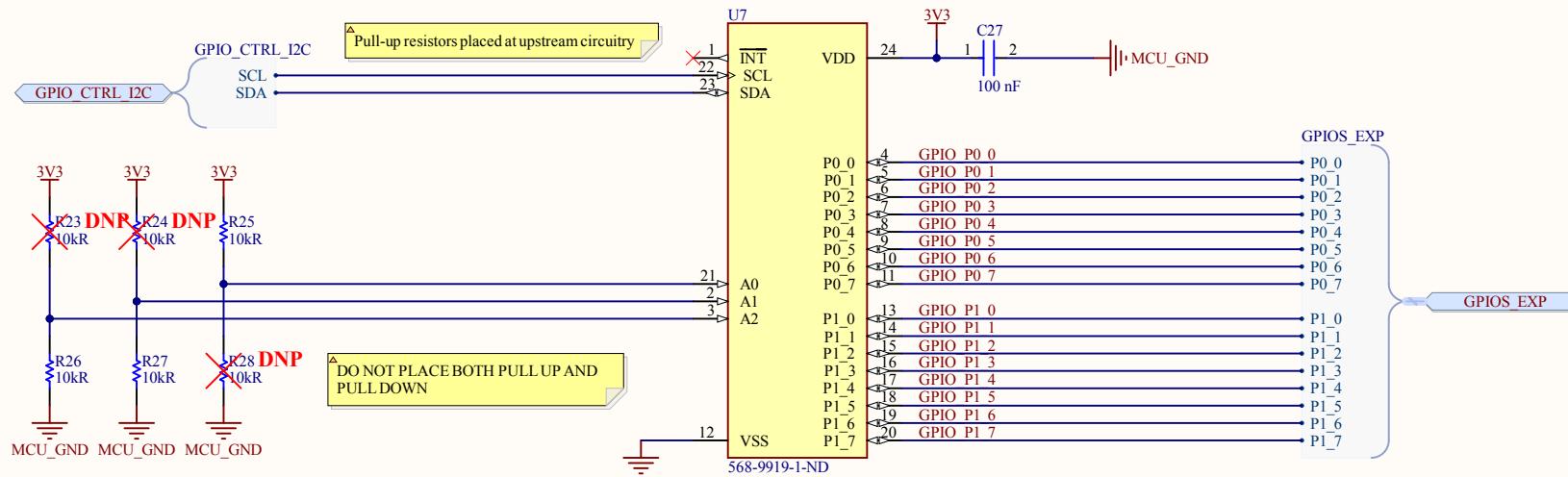


Fig 4. PCA9555A device address

A2, A1 and A0 are the hardware address package pins and are held to either HIGH (logic 1) or LOW (logic 0) to assign one of the eight possible slave addresses. The last bit of the slave address (R/W) defines the operation (read or write) to be performed. A HIGH (logic 1) selects a read operation, while a LOW (logic 0) selects a write operation.

Title **fnIRS - GPIO Expander**

Size A4	Number <i>UofT Capstone Team 2024991</i>	Revision 1
Date: 3/21/2025	Sheet 4 of 13	
File: C:\Mac\GPIO_Expander.SchDoc	Drawn By: <i>Tony Kim</i>	

A

B

C

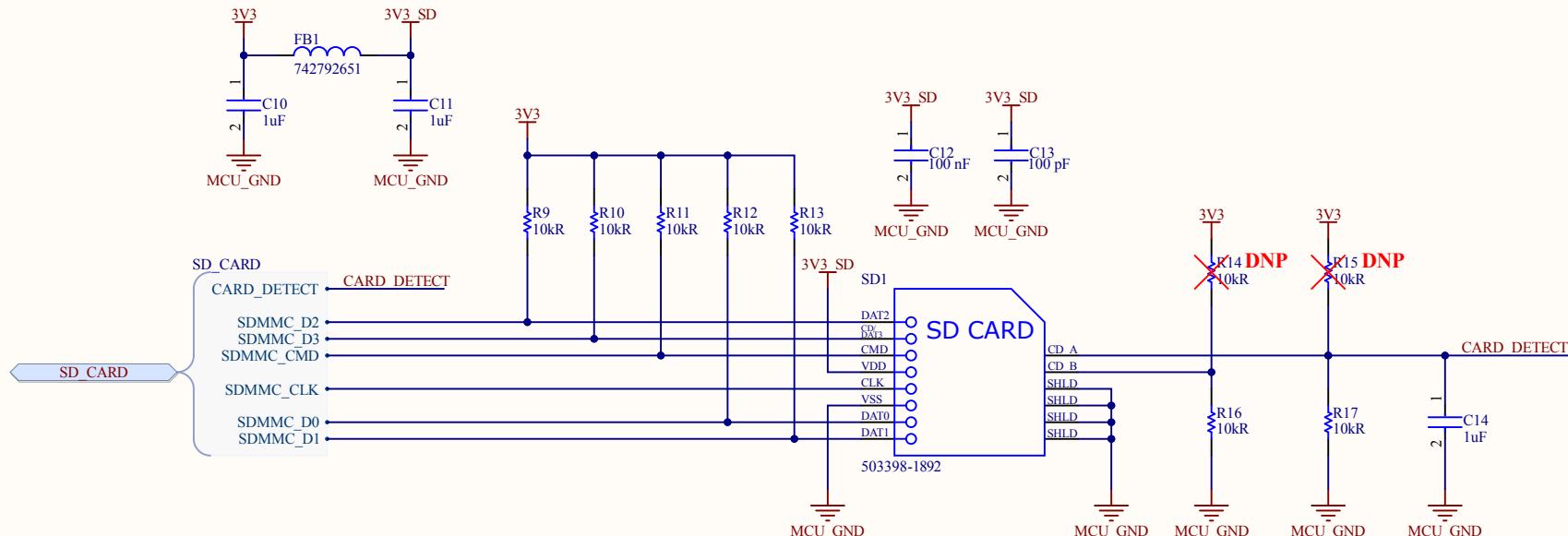
D

A

B

C

D



Title fnIRS - SD Card		
Size A4	Number <i>UofT Capstone Team 2024991</i>	Revision 1
Date: 3/21/2025	Sheet 5 of 13	Drawn By: <i>Tony Kim</i>

A

A

B

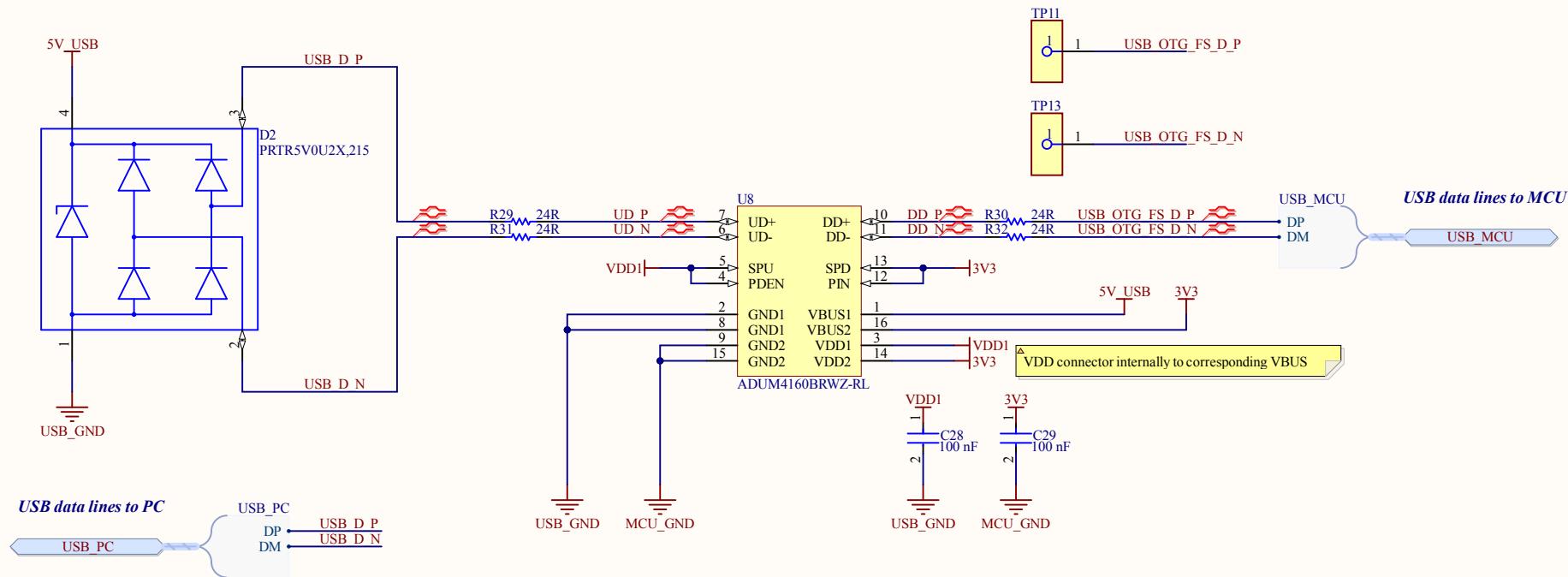
B

6

6

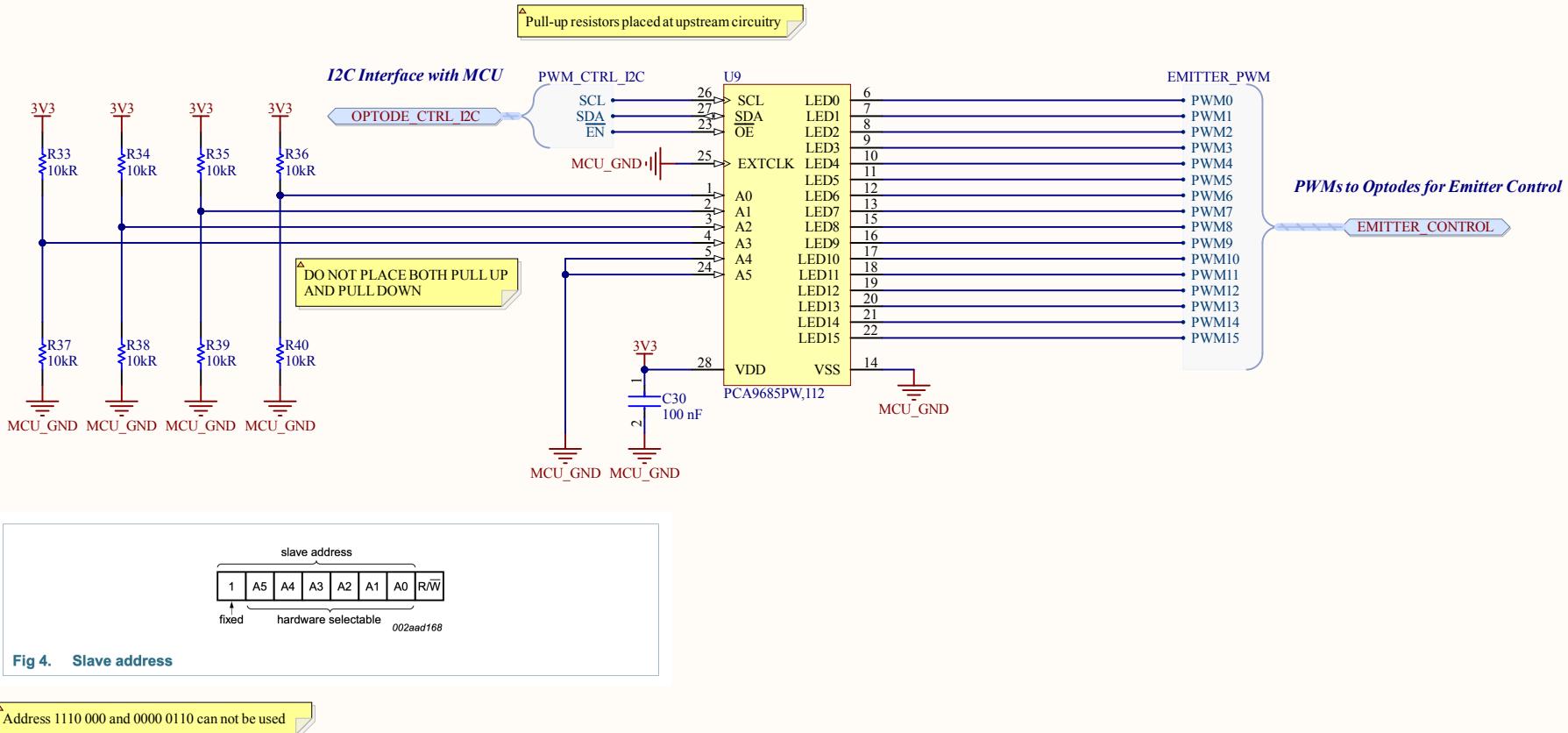
8

8



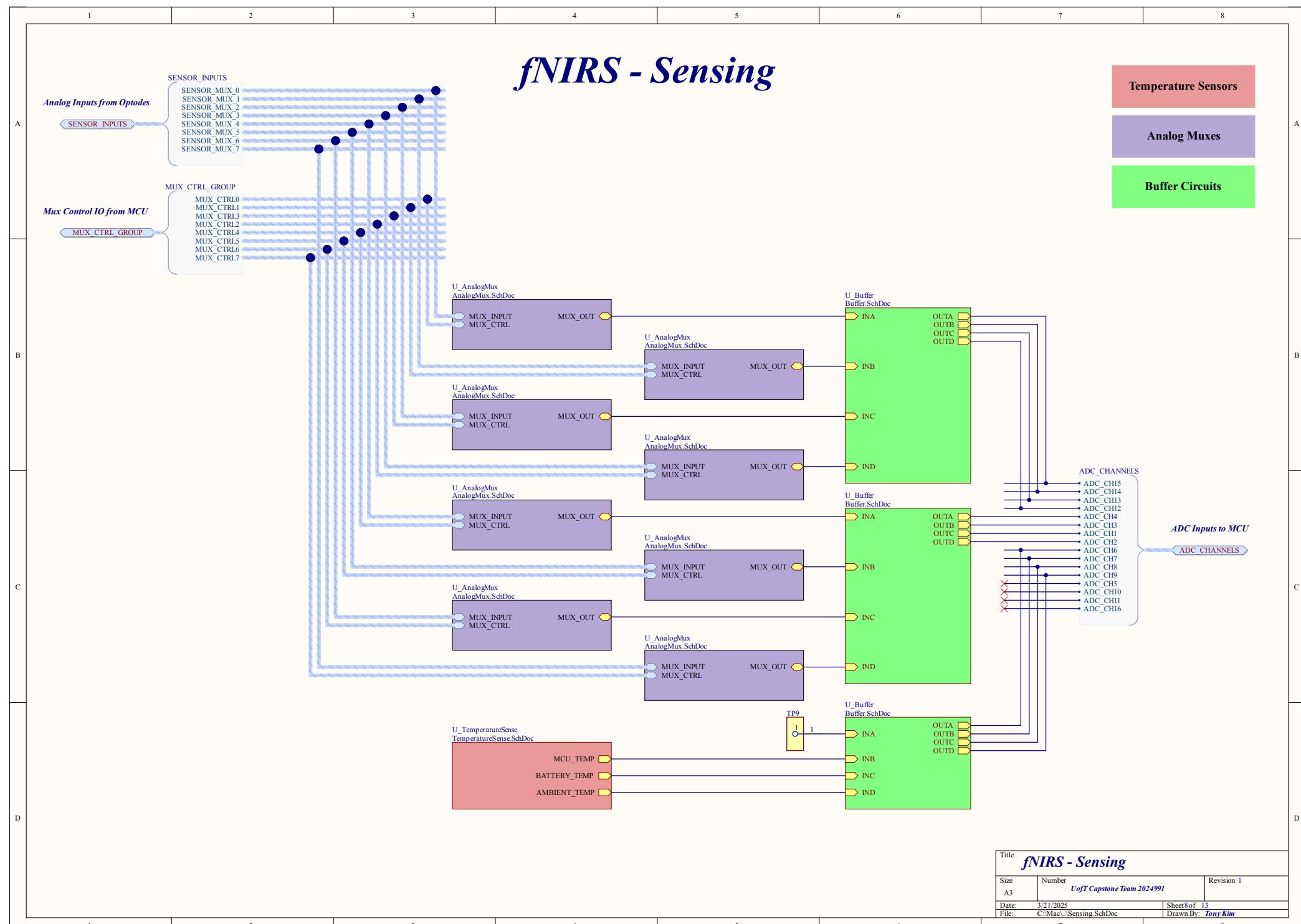
Title		<i>fnIRS - USB Interface</i>	
Size	Number	UofT Capstone Team 2024991	
Date:	3/21/2025	Sheet	of 13
File:	C:\Mac\USBInterface.SchDoc	Drawn By:	<i>Tony Kim</i>

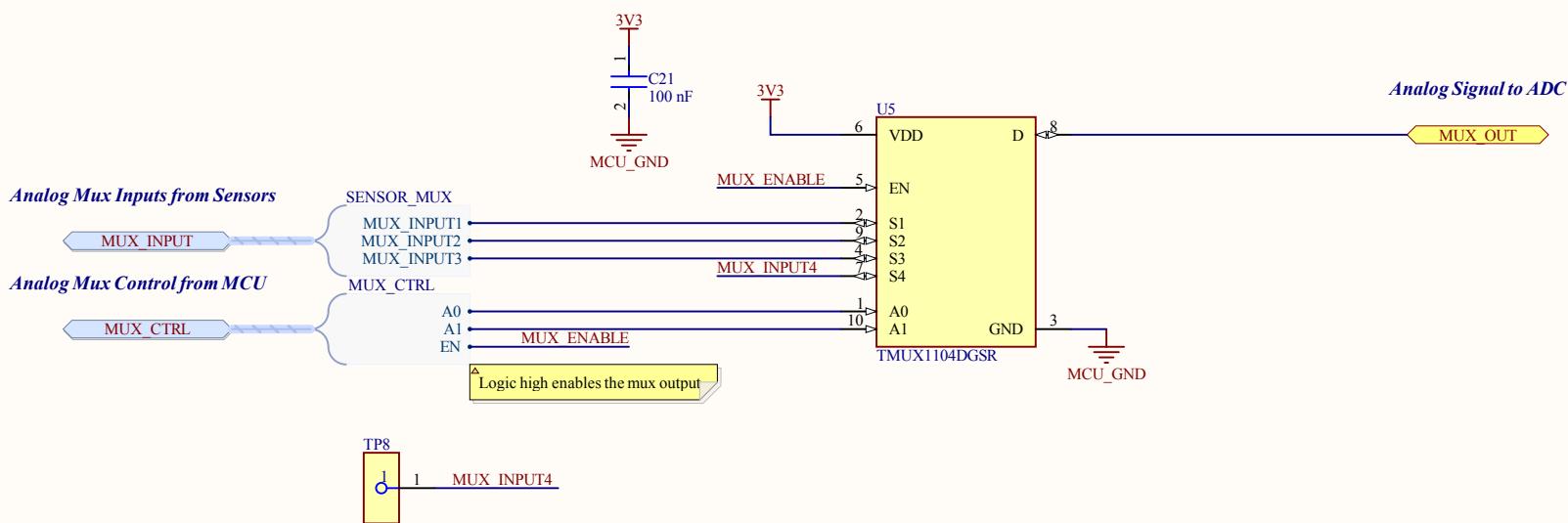
fNIRS - Optode Control



Title fNIRS - Optode Control		
Size A4	Number UofT Capstone Team 2024991	Revision 1
Date: 3/21/2025	Sheet 7 of 13	
File: C:\Mac\OptodeControl.SchDoc	Drawn By: Tony Kim	

fNIRS - Sensing





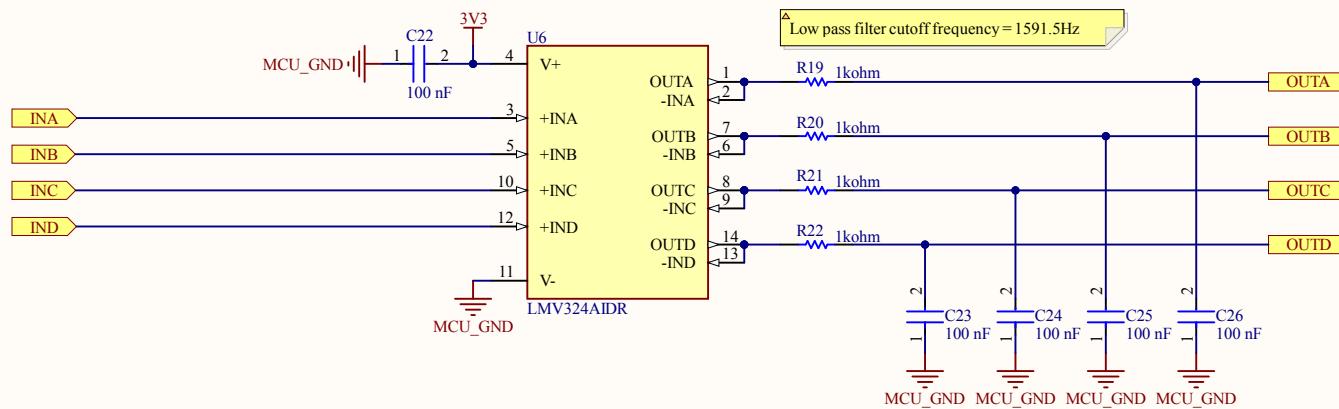
Title fnIRS - Analog Mux		
Size A4	Number UofT Capstone Team 2024991	Revision 1
Date: 3/21/2025	Sheet 9 of 13	
File: C:\Mac\AnalogMux.SchDoc	Drawn By: Tony Kim	

A

A

B

B



C

C

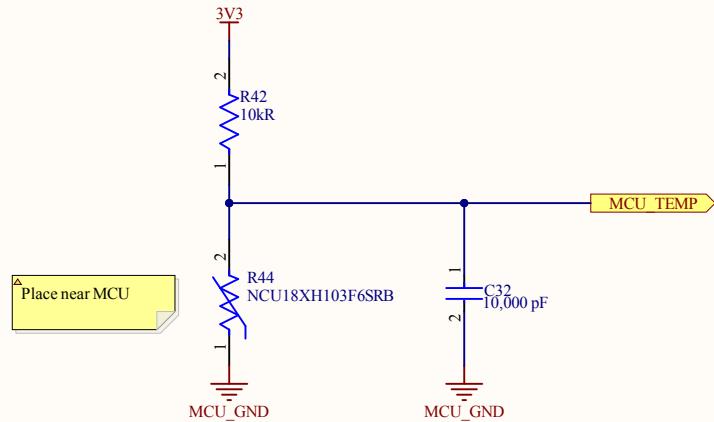
D

D

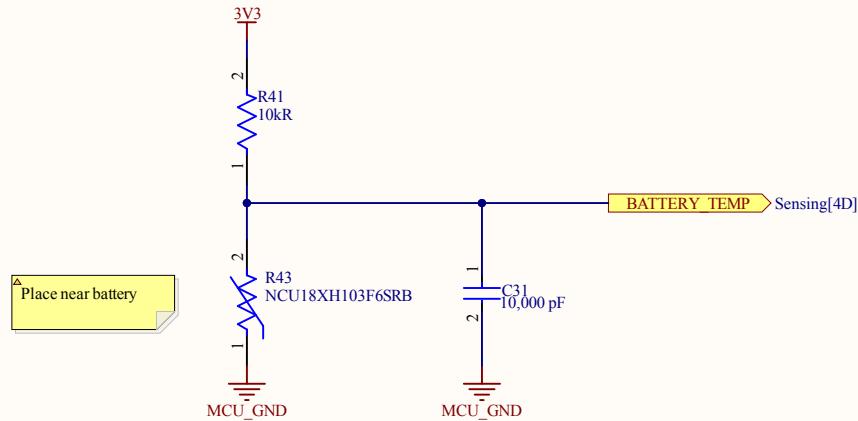
Title ***fnIRS - Buffer***

Size	Number	Revision
A4	<i>UofT Capstone Team 2024991</i>	1
Date: 3/21/2025	Sheet 1 of 13	
File: C:\Mac\Buffer.SchDoc	Drawn By: <i>Tony Kim</i>	

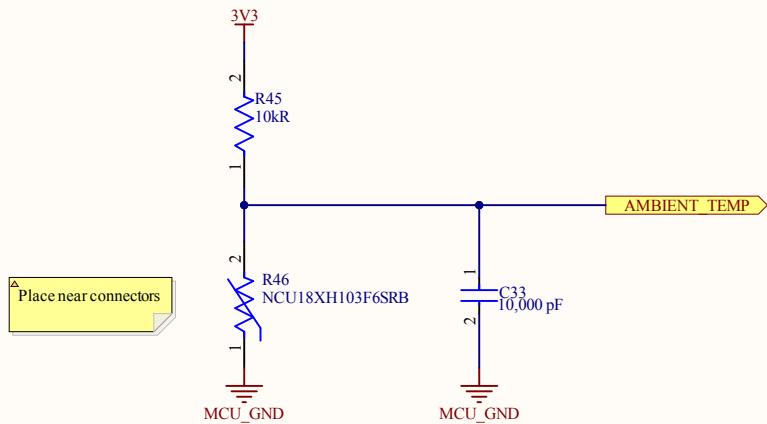
A



B



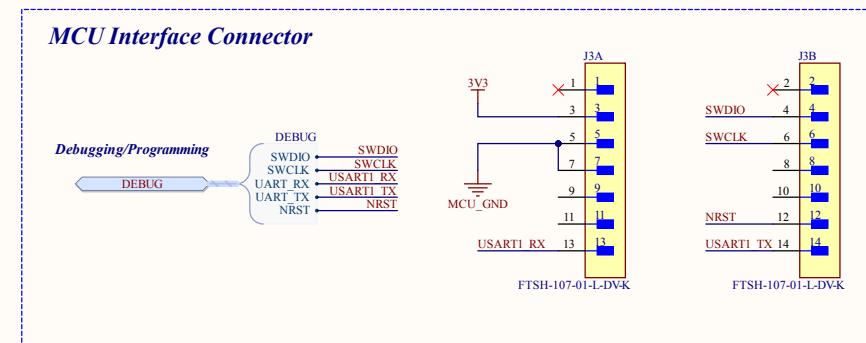
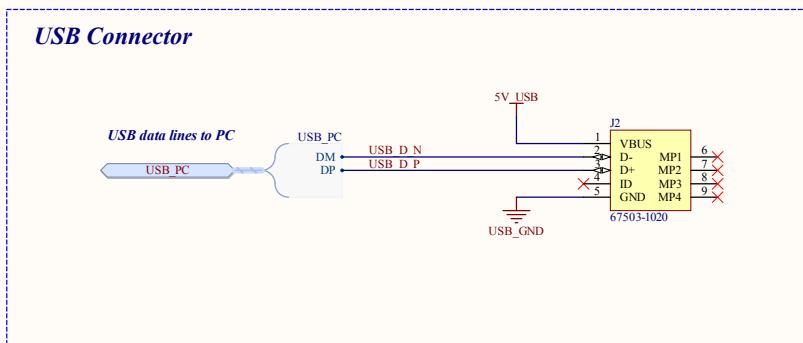
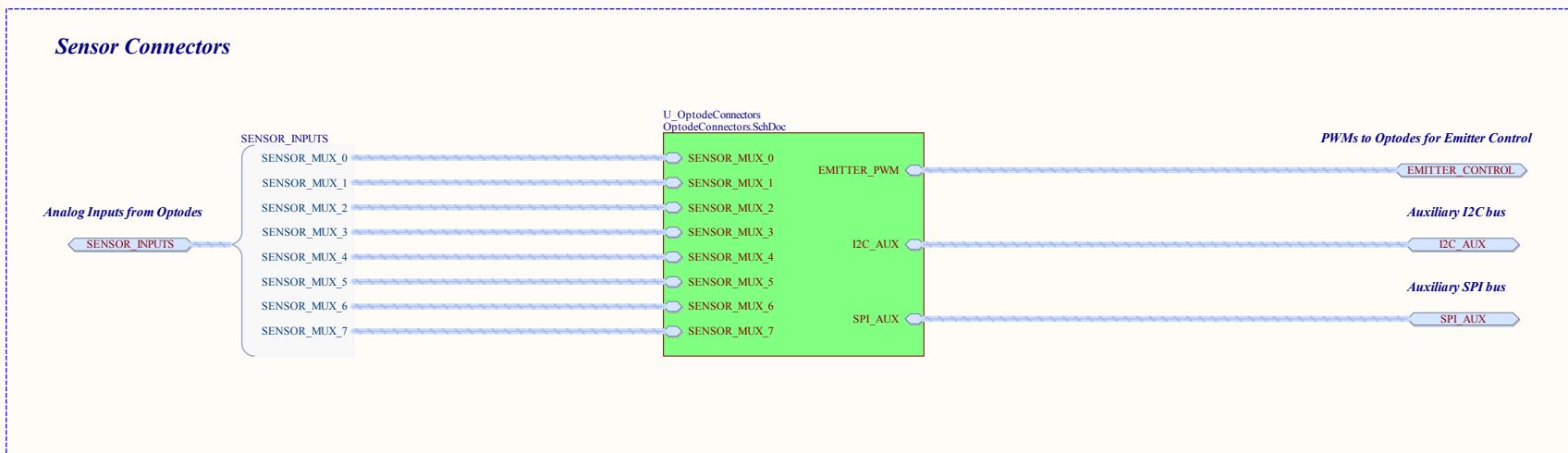
C



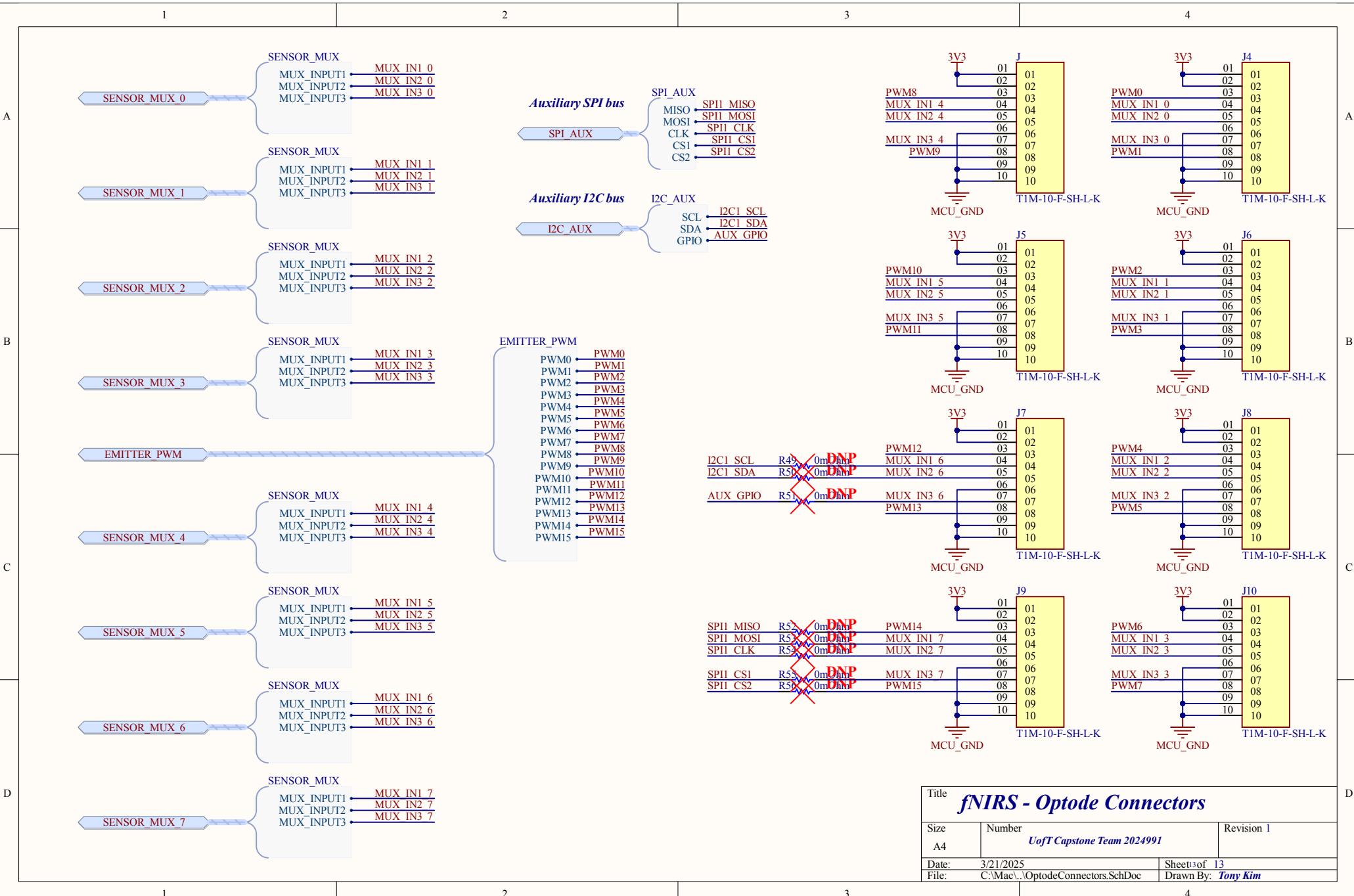
D

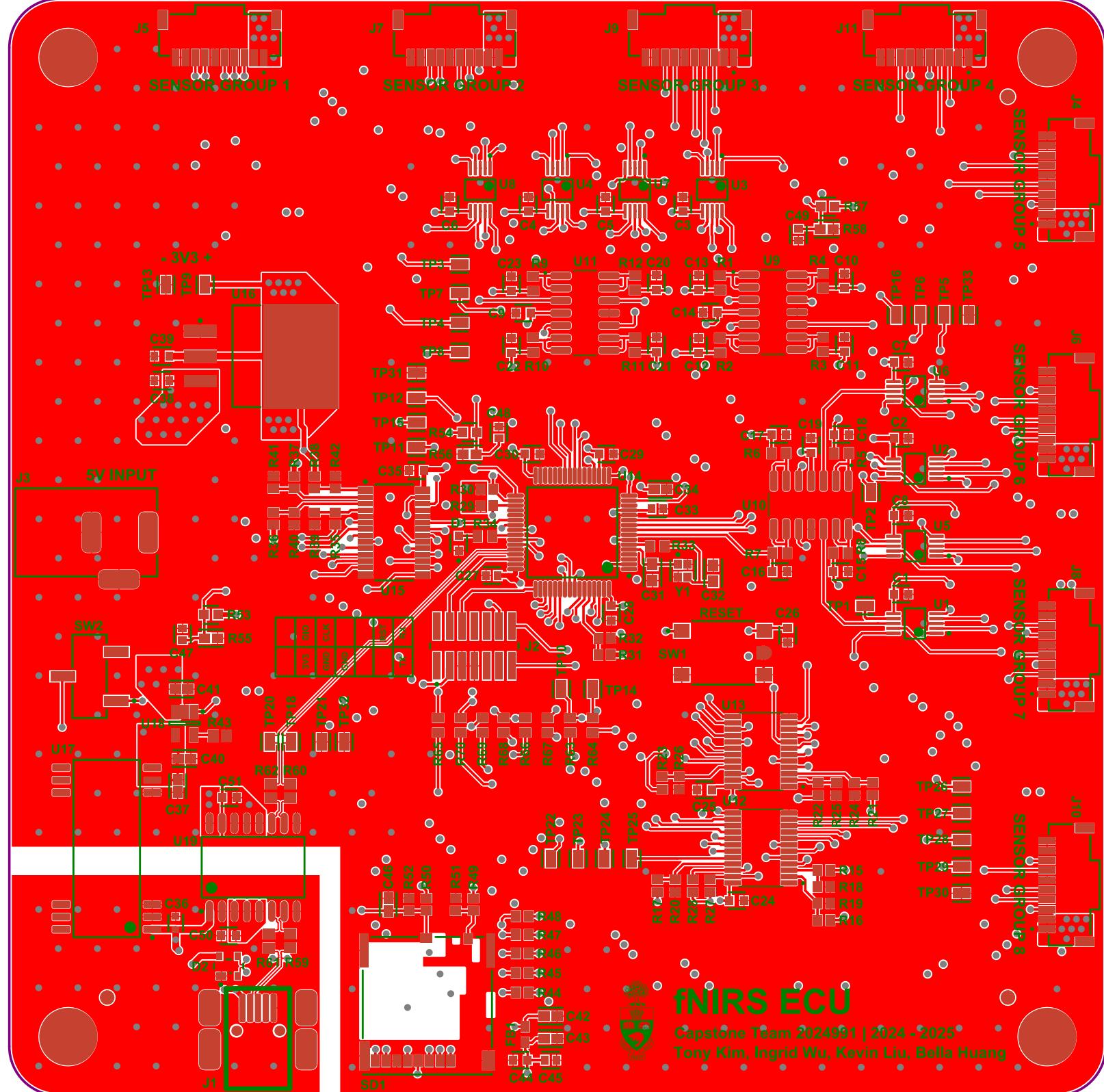
Title fnIRS - Temperature Sense		
Size A4	Number UofT Capstone Team 2024991	Revision 1
Date: 3/21/2025	Sheet 1 of 13	
File: C:\Mac\TemperatureSense.SchDoc	Drawn By: Tony Kim	

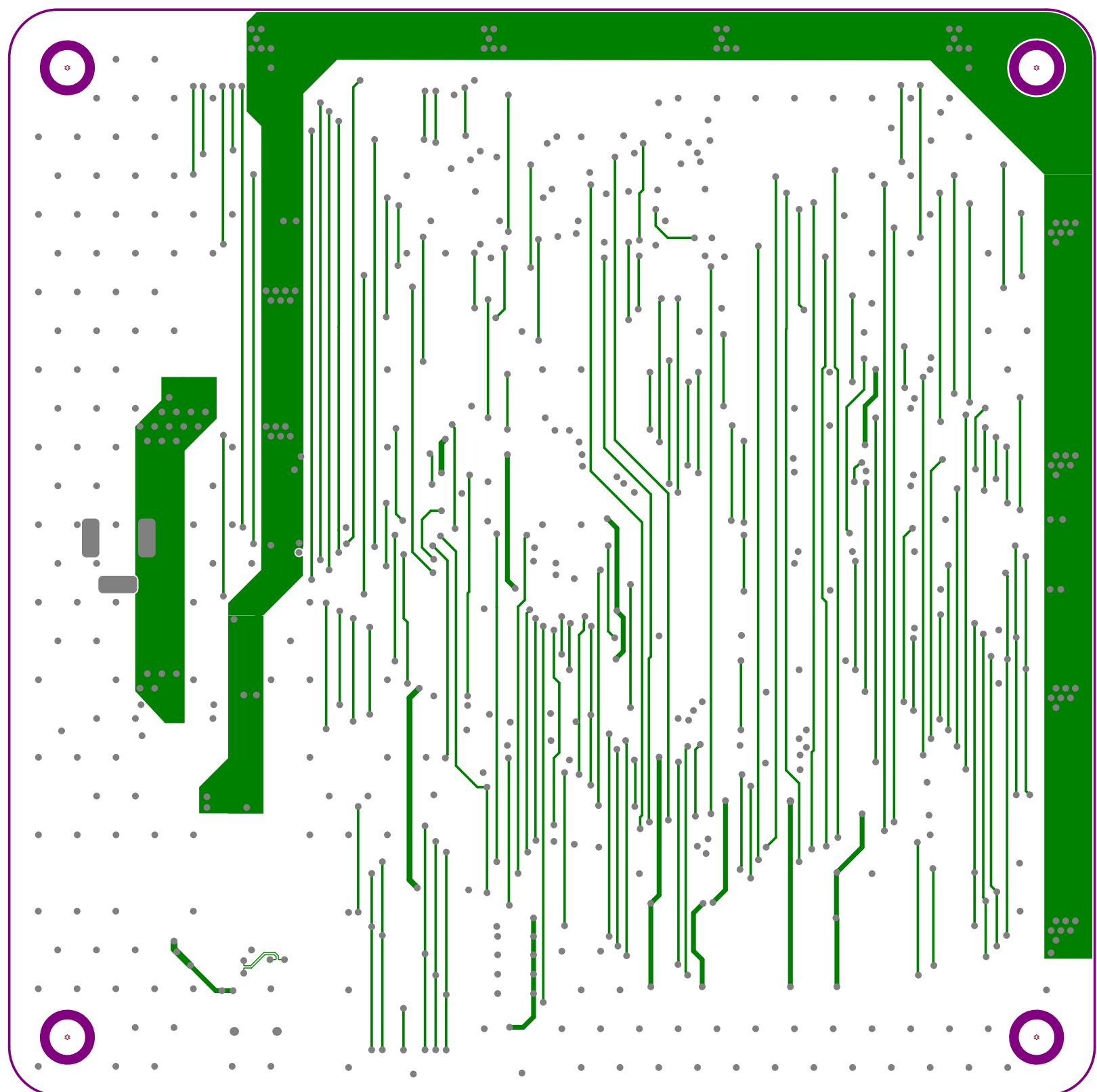
fNIRS - Connectors

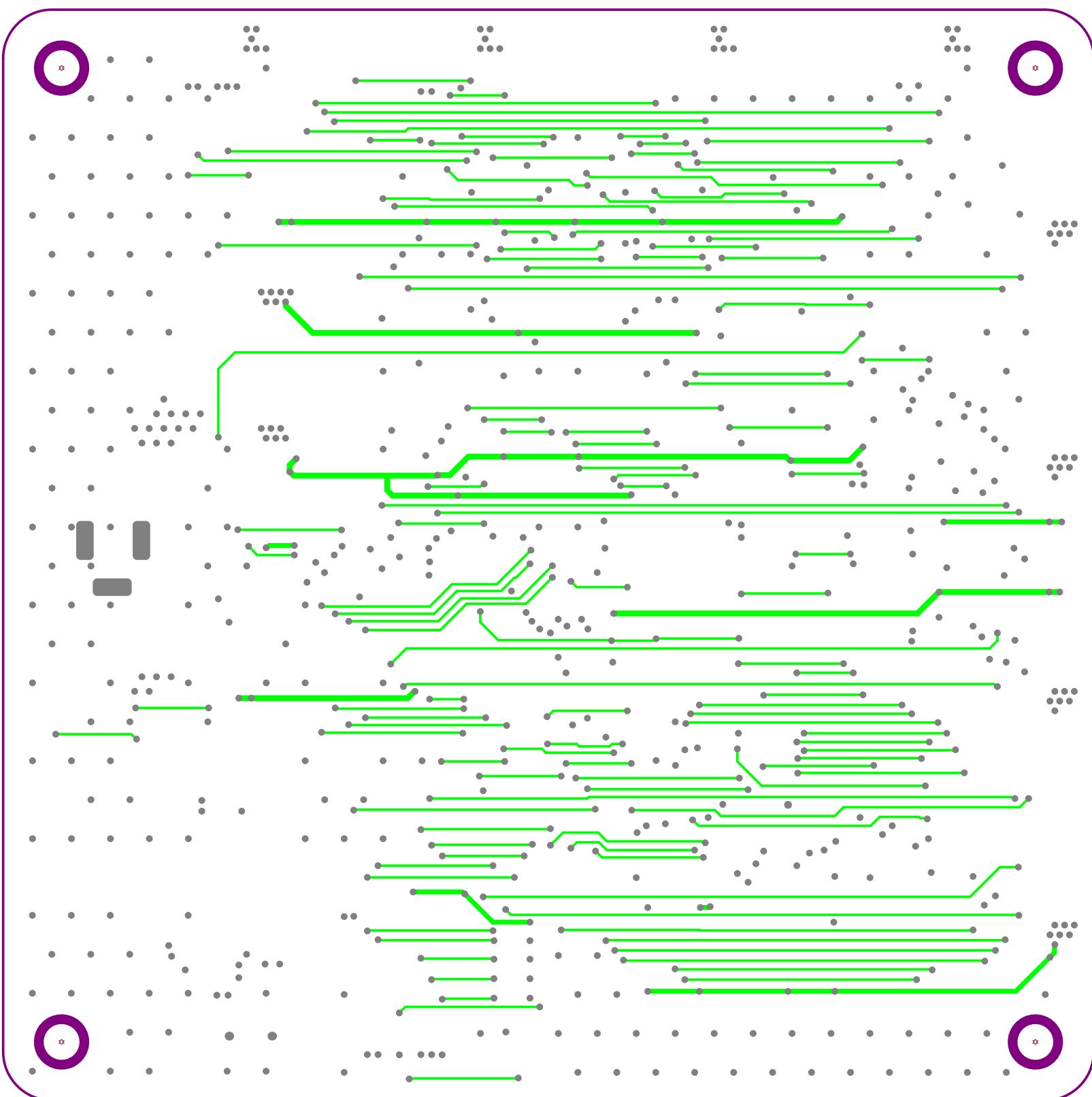


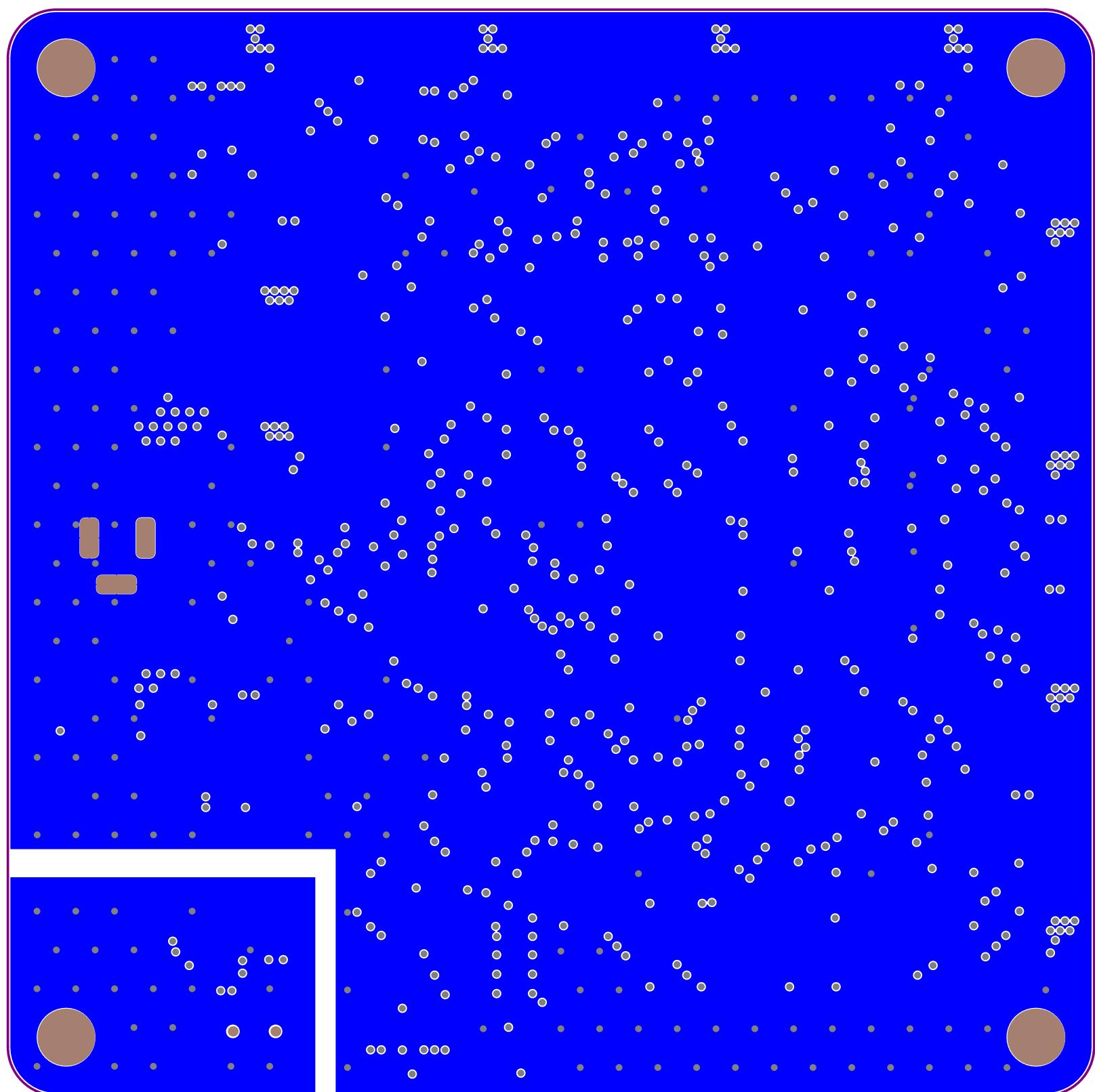
Title fNIRS - Connectors		
Size A3	Number UofT Capstone Team 2024991	Revision 1
Date 3/21/2025	Sheet 1 of 13	
File: C:/Mac/_Connectors.SchDoc	Drawn By: Tony Kim	







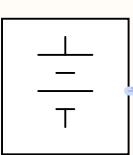




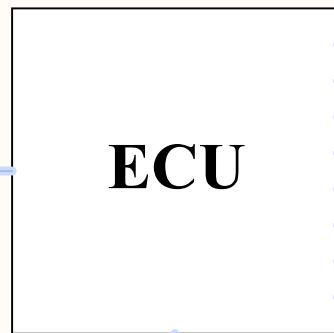
Appendix E: Cable Assembly and System Wiring Diagram

1 2 3 4

A

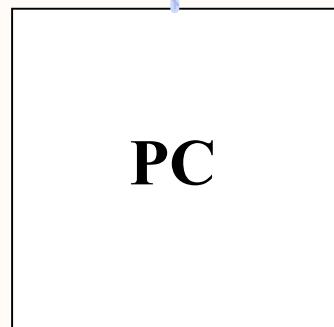


Power
Cable

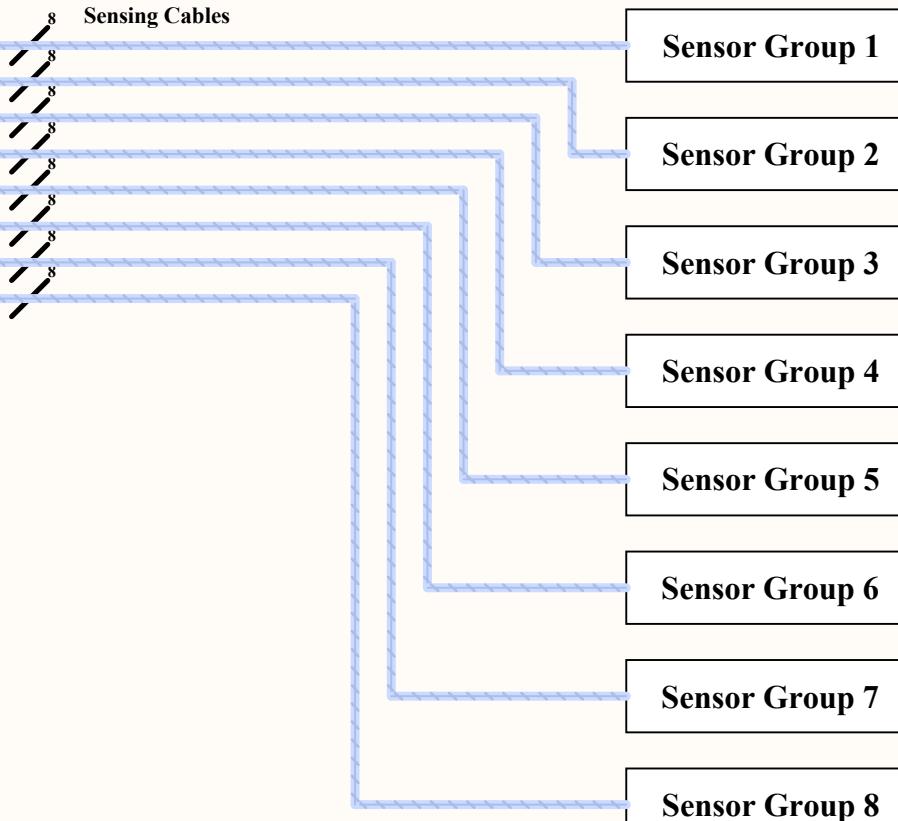


ECU

Standard
USB Mini Cable



Sensing Cables



A

B

C

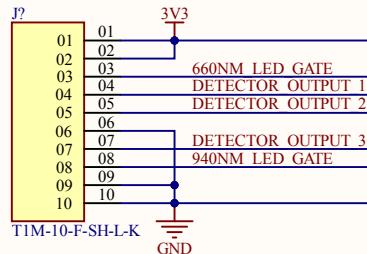
D

Title <i>fnIRS System Wiring Diagram</i>		
Size A4	Number <i>UofT Capstone Team 2024991</i>	Revision 1
Date: 3/21/2025	Sheet of 2	Drawn By: <i>Tony Kim</i>
File: C:\Mac\...\SystemWiring.SchDoc		

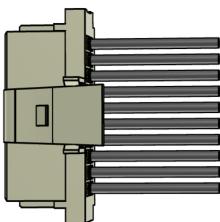
1 2 3 4

Sensing Cable

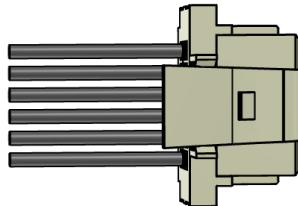
ECU Sensing Connector



ECU Cable Harness: S1SS-10-28-GF-10.00-L3

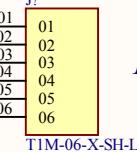


Sensor Cable Harness: S1SS-06-28-GF-10.00-L3

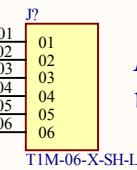


Sample Cable Harnesses at: <https://www.samtec.com/products/s1ss>

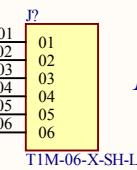
Detector 2



Detector 1 with Emitter



Detector 3



Sensing Cable Wiring Diagram

Size	Number	Revision
A4	UofT Capstone Team 2024991	1
Date:	3/21/2025	Sheet of 2
File:	C:\Mac\Documents\SensingCables.SchDoc	Drawn By: Tony Kim

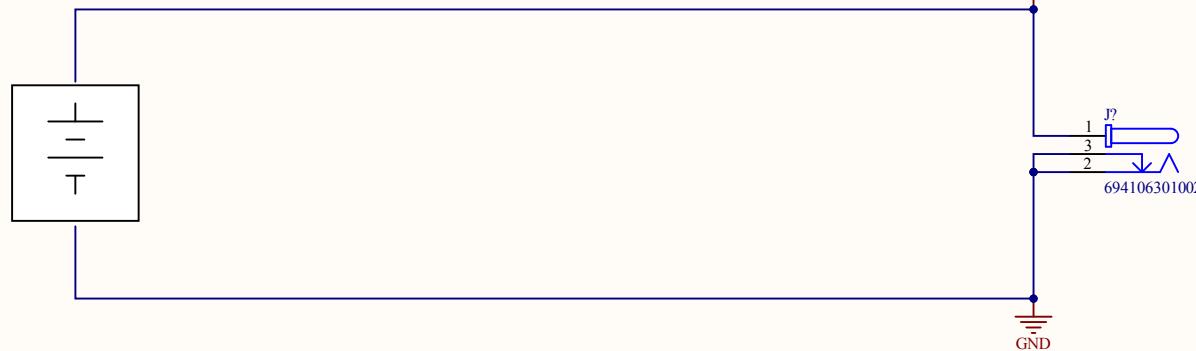
A

Power Cable

A

B

Battery voltage should not exceed 5V



C

Barrel Connector can be acquired: <https://www.digikey.ca/en/products/detail/globtek-inc/V02/12341457>

B

D

Title		
Size	Number	Revision
A4	<i>UofT Capstone Team 2024991</i>	1
Date: 3/21/2025	Sheet 3	of 3
File: C:\Mac\PowerCable.SchDoc	Drawn By:	<i>Tony Kim</i>

Appendix F: Sensor Noise Level Testing Results

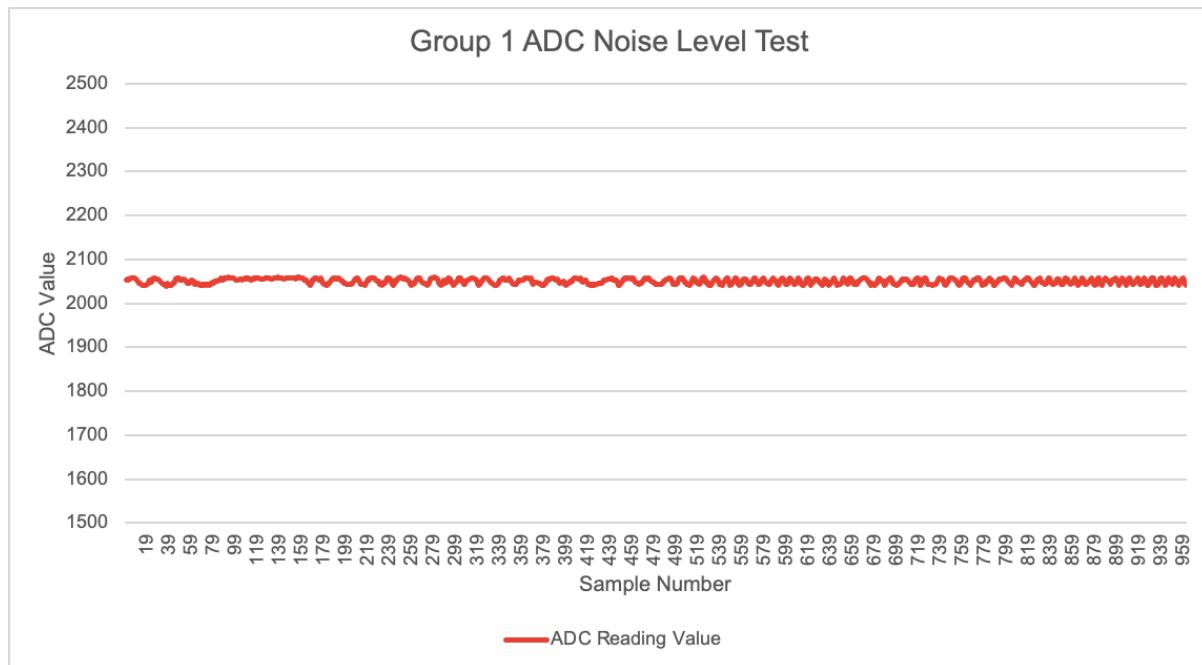
Test Setup: Place sensor inside a box that is covered and measure ADC readings - collected about 1000 samples for sampling frequency of 1kHz.

Analysis: For each data point (which is in ADC counts)

1. Compute the Demean value (current sample - average of all samples)
2. Compute the RMS value in ADC counts
3. Compute the RMS value in volts (multiply by ADC reference voltage / resolution (2^{12}))
4. Compute the Noise Level in dB (20 log (rms value in volts))

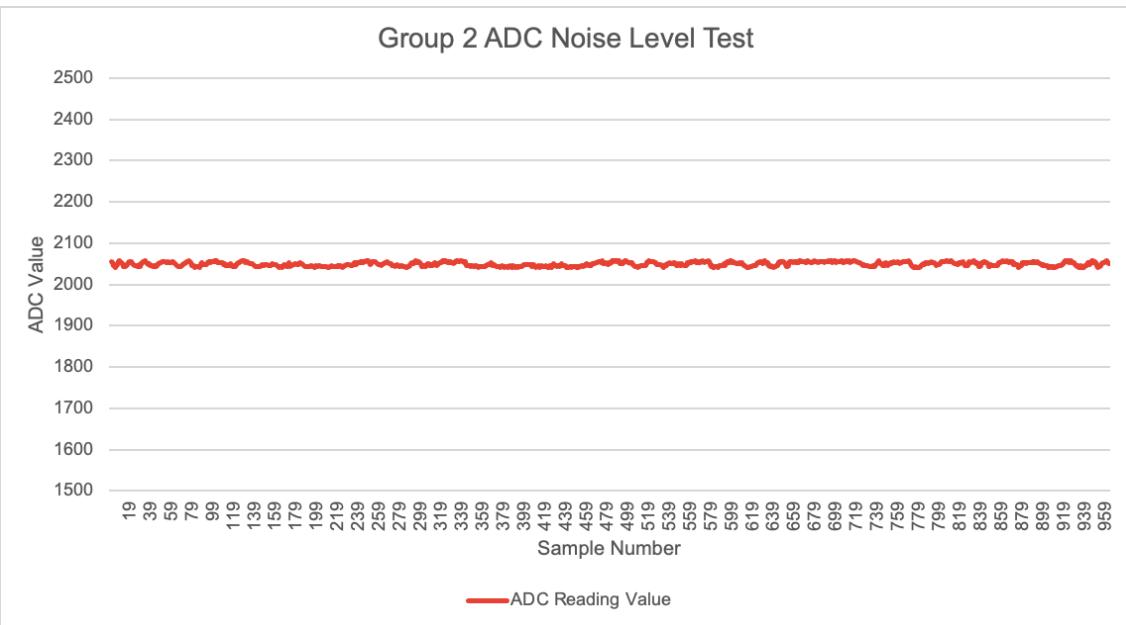
Sensor Group 1 Results:

RMS value in Volts	Noise Level (dB)
0.003968200825	-46.25598098



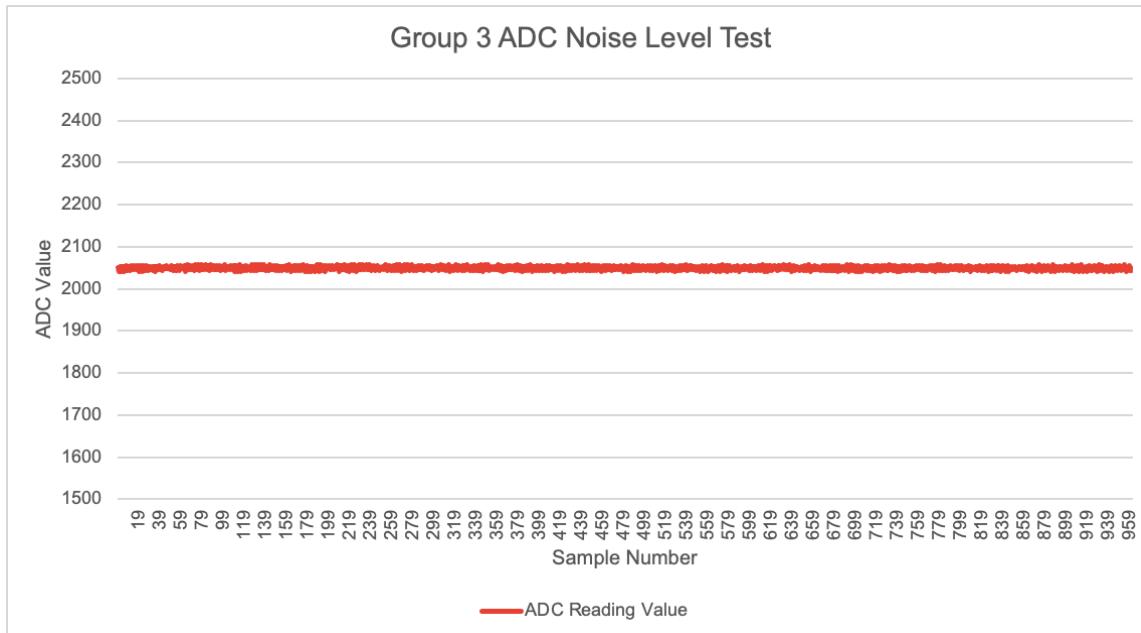
Sensor Group 2 Results:

RMS value in Volts	Noise Level (dB)
0.003968200825	-48.02812714



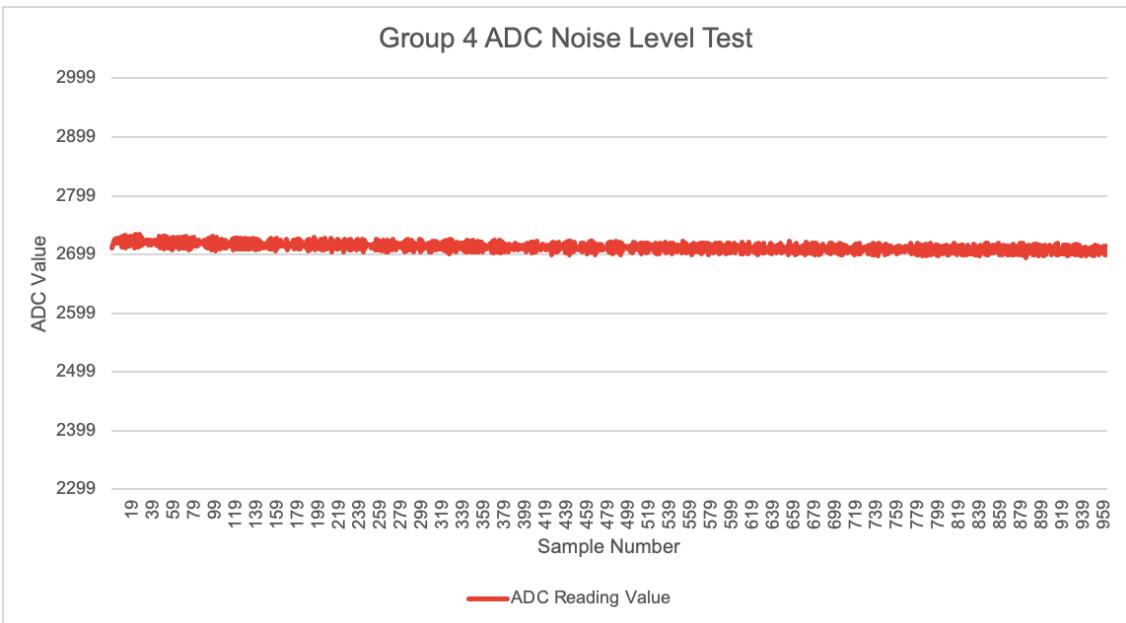
Sensor Group 3 Results:

RMS value in Volts	Noise Level (dB)
0.004102380598	-47.73928101



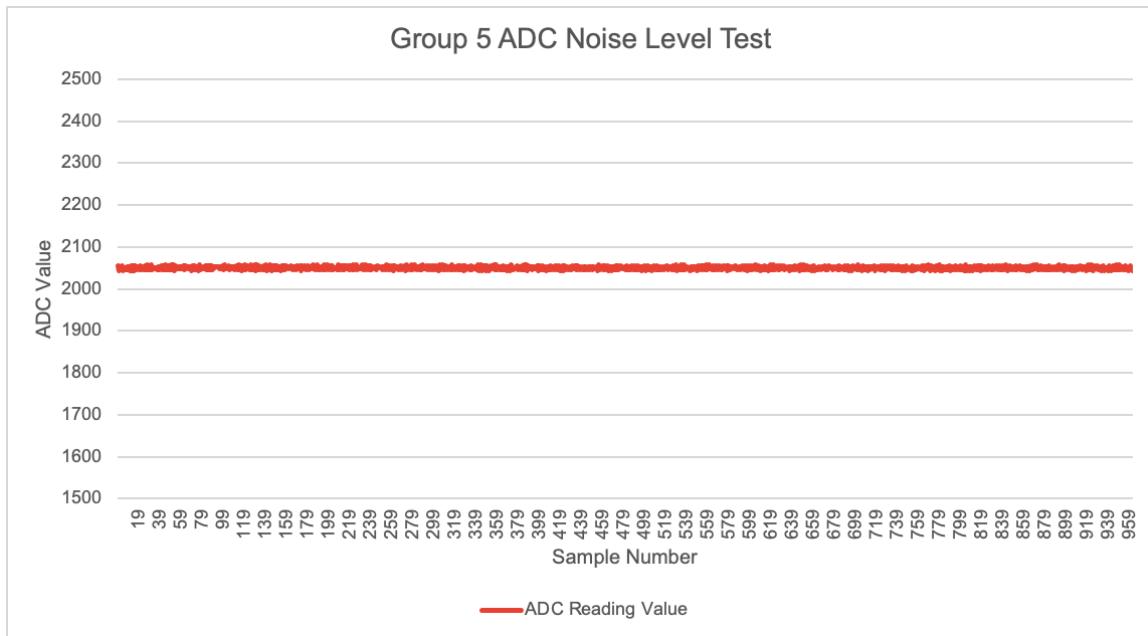
Sensor Group 4 Results:

RMS value in Volts	Noise Level (dB)
0.006702306008	-43.47551495



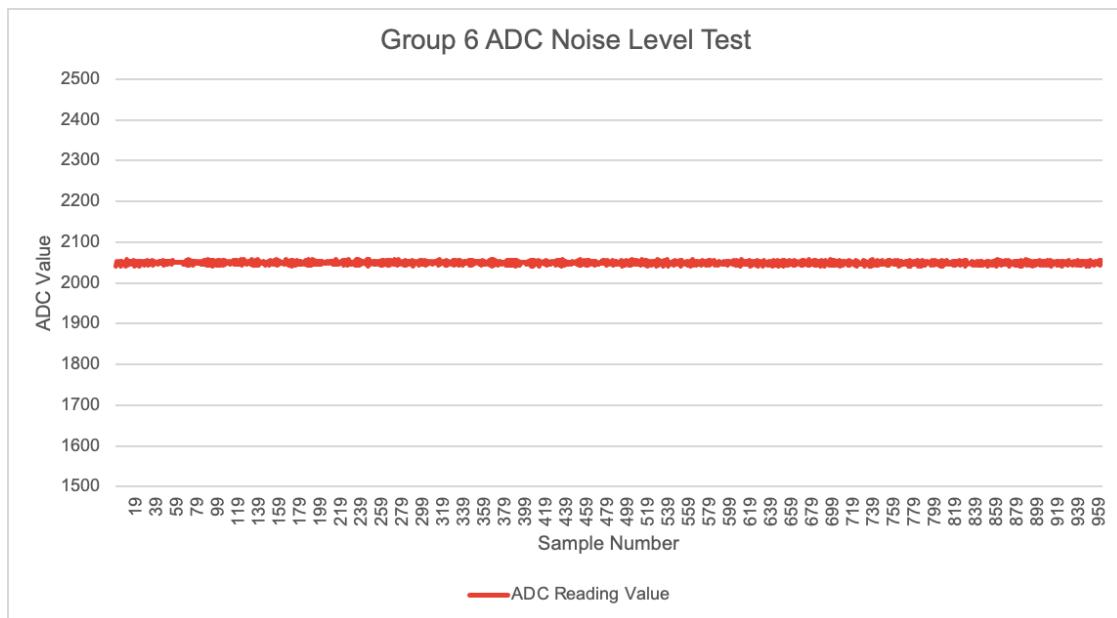
Sensor Group 5 Results:

RMS value in Volts	Noise Level (dB)
0.0001500089477	-76.47765671



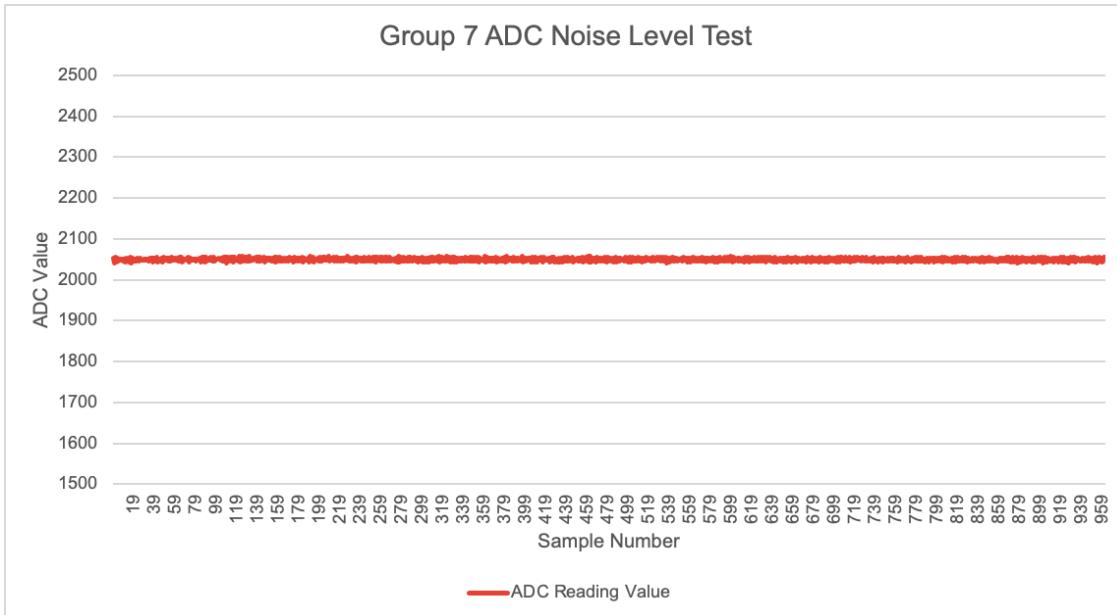
Sensor Group 6 Results:

RMS value in Volts	Noise Level (dB)
0.003837880792	-48.31817038



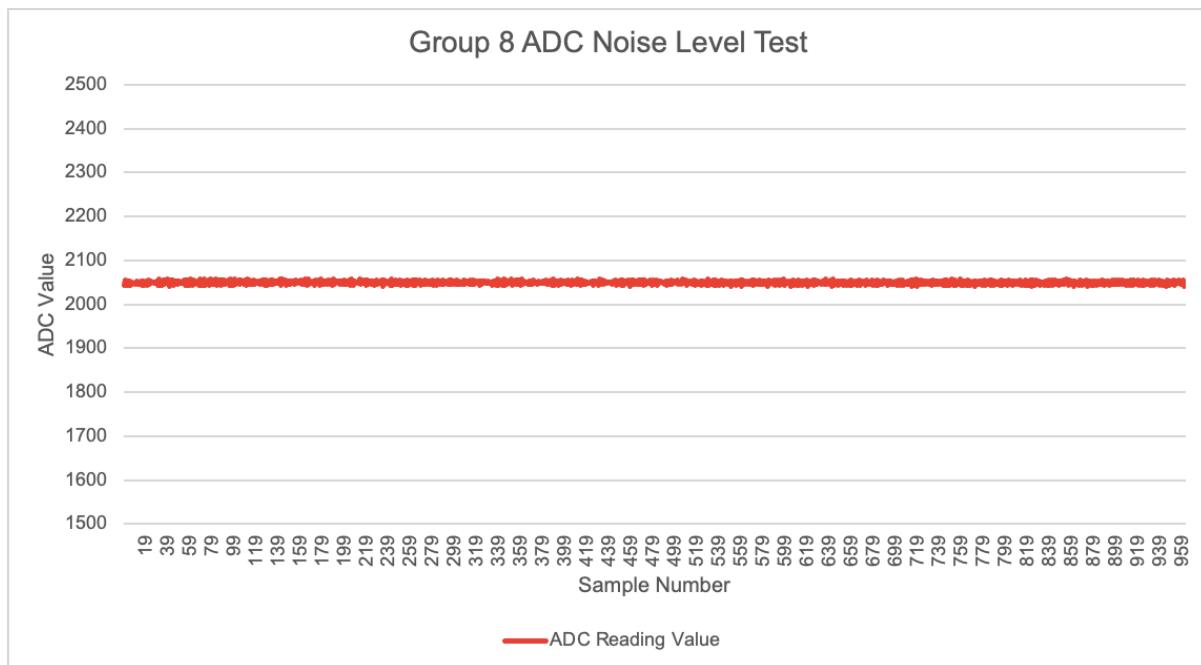
Sensor Group 7 Results:

RMS value in Volts	Noise Level (dB)
0.003381601199	-49.41755222



Sensor Group 8 Results:

RMS value in Volts	Noise Level (dB)
0.003727577159	-48.57146716



Appendix G: ADC Calibration Tests

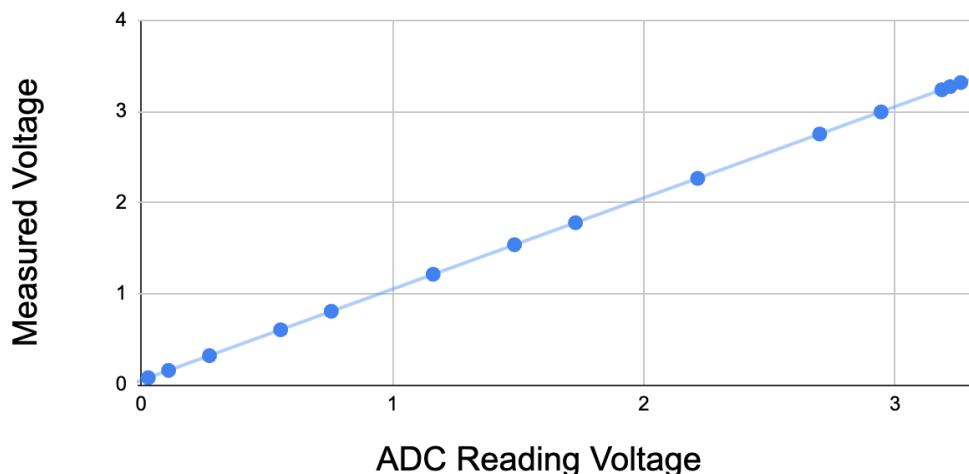
Test Setup: Sweep voltage on sensor channel and compare digital readings to actual value.
Values are calibrated with Fluke 179 True RMS multimeter (0.5% precision)

Experimental Data:

Measured Input Voltage	ADC (Digital) Reading	ADC Reading in Voltage
0.079	35	0.02831623932
0.159	135	0.1092197802
0.321	336	0.2718358974
0.605	686	0.5549982906
0.808	935	0.7564481074
1.214	1436	1.161774847
1.538	1836	1.485389011
1.78	2136	1.728099634
2.266	2737	2.214329915
2.753	3337	2.69975116
2.996	3639	2.944079853
3.239	3938	3.185981441
3.274	3979	3.219151893
3.318	4031	3.261221734

ADC1 Calibration

● Measured Input Voltage — $1*x + 0.0498$



Result: Scale value = 1, offset = 0.0498