OpenSense v3 – Simplifying Sensors for Educational Experimentation

Brandon Collins, James Eyler, Daniel Gonzalez, Noah Wilfond

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — OpenSense V3 is a modular, open-source sensor platform designed for education and rapid prototyping applications. Built around the ESP32-WROOM microcontroller, the system supports up to five sensors through dedicated ports, each capable of interfacing with I²C devices, with two ports additionally supporting analog inputs and configurable voltage levels of 3.3V or 5V. Data acquisition is complemented by onboard microSD storage and a browser-based interface that enables system configuration and real-time data visualization. To further demonstrate system versatility, the platform integrates a monochromator-based optical spectrometer. Prototype hardware and optical subsystems were successfully designed, fabricated, and tested, validating the feasibility of a low-cost, customizable solution that empowers users to build, measure, and better understand real-world sensing systems.**

Index Terms **analog-digital conversion, data acquisition, wireless access points, optical sensors, sensors, web servers, multiplexing equipment.**

I. Introduction

Sensor platforms are essential tools for data acquisition systems, enabling the collection, processing, and communication of environmental and experimental data across a wide range of applications. Many commercial sensor systems are expensive and operate as closed "black boxes" that support only predefined sensor lists and conceal their internal logic making it less practical for someone with less technical experience to utilize and understand the system. Such limitations impede learning, limit customizations, and raise barriers for use, especially for student teams and resource constrained laboratories.

To address these challenges, OpenSense was developed by the University of Central Florida's DRACO Lab [1]. The OpenSense platform has had two previous iterations. These previous versions validated the concept of a low-cost educational sensor node, but were constrained by pre-coded sensor libraries, single-voltage operation, and limited user configurability. OpenSense v3 addresses each of these shortcomings through a comprehensive redesign of the hardware, firmware, and user interface.

The system is built around an embedded ESP32-WROOM microcontroller unit (MCU) with integrated wireless capabilities. It is designed to be able to simultaneously read up to five sensors at once, either through digital I²C formatting or an analog signal using an ADC. Two of the sensor ports are equipped to supply either 3.3V or 5V sensors which are switched via a power multiplexor system controlled by the MCU. A FreeRTOS firmware stack is used to sample these sensors at a rate of 10 Hz, log record data to an SD card, and stream real-time telemetry to a browser dashboard over Wi-Fi at 2 Hz. The user-interface also incorporates a parameter-driven sensor driver abstraction that allows new sensors to be added without the need to adjust the system's firmware. To validate the platform's extensibility, several sensors were developed in parallel, the most advanced of which is an optical spectrometer subsystem that illustrates its capability to support advanced sensing applications in both project-based and educational settings.

II. System Components

A. Microcontroller Unit (MCU)

The ESP32-WROOM-32E serves as the central processing and communication hub of the OpenSense v3 platform. It manages all I²C transactions, controls the microSD interface, hosts the WiFi SoftAP and HTTP/WebSocket server, and executes the FreeRTOS task scheduler. GPIO pins directly control the power multiplexors, level shifters, and switches that control the logic level that the adjustable voltage sensor ports operate at.

B. I²C Multiplexor

The ESP32's integrated I²C bus can only handle a limited number of devices before running into issues with address conflicts and data timing issues. To avoid any addressing conflicts, OpenSenseV3 uses a TCA9548A 8-channel I²C multiplexor to selectively activate individual sensor channels through software control, effectively isolating devices that may share identical I^2C addresses.

C. Analog to Digital Converter (ADC)

Analog sensor support is provided through an MCP3221 12-bit ADC, which is directly connected to the SDA and SCL lines of the I^2C bus in order to accommodate analog-output sensors alongside digital I^2C devices. The MCP3221 offers a sampling rate of 22.3 kbps and draws minimal active current.

D. Dual-Voltage Sensor Path

Two of the five sensor ports support selectable 3.3 V or 5 V operation through a chain of three configurable components. A TPS2117 power multiplexer routes the appropriate supply rail to the sensor VCC, an NTB0102DP bidirectional level shifter translates I²C signals between voltage domains by stepping incoming sensor data down to 3.3 V logic and shifting outbound signals up to 5 V when required, and a TS3A24157 Single Pole Double Throw (SPDT) switch directs the data lines through the appropriate voltage path. Together, these components form dual-voltage sensor interfaces that are controlled via GPIO pins and configured by the user through the web-based interface.

E. Micro-SD Connector

Acquired sensor data is logged locally to a microSD card via the DM3AT-SF-PEJM5 connector from Hirose Electric Co Ltd, interfaced to the ESP32. This local storage enables fully autonomous, off-grid data collection independent of any wireless client connection, with logged CSV files retrievable either by removing the card or downloading them directly through the web-based interface.

F. Demonstration Sensors

For demonstration purposes, multiple sensors of different types were developed. A LMT87DCK was used to develop a basic temperature sensor. For a humidity sensor the HIH-6130 was selected. The VEML7700 was used to create a brightness sensor. And to create a basic pressure sensor a 34-00015 FSR was used. The temperature and pressure sensors use analog connections to test the main board's capacity to work with analog sensors while the brightness and humidity sensors use the main board's I^2C ports. The power consumption of the sensors is minimal so they help in reaching the goal of running up to five sensors simultaneously on the platform.

To demonstrate the platform's ability to integrate an advanced custom sensor, a Czerny-Turner reflection spectrometer targeting 300–800 nm was designed as a proof-of-concept sensor module. The optical configuration employs two Thorlabs CM127-025-G01 concave mirrors (25 mm focal length) for collimation and focusing, and a Thorlabs GH13-12V holographic diffraction grating (1200 lines/mm) to disperse incident light across the target wavelength range. A SER00046 microservo rotates the grating to scan wavelengths sequentially, with each angular position mapped to a specific wavelength via the grating equation. Light intensity at each wavelength is measured by a TEMD5080X01 PIN photodiode paired with a transimpedance amplifier, converting photocurrent to a voltage readable by the ADC. An ATTiny84 microcontroller handles servo PWM generation and photodiode readout, forwarding the combined spectral and position data to the OpenSense v3 platform over I²C where the platform's data system can analyze the data and display the spectrum.

III. Hardware Detail

A. Hardware Block Diagram

The top-level hardware architecture of OpenSense v3 is shown in Fig. 1. At the center of the design is the ESP32-WROOM-32E MCU, which serves as the sole master for all I^2C transactions, SD card data storing, and wireless communication. The diagram below illustrates at a basic level the board's functions and communication methods for both internal and external components.

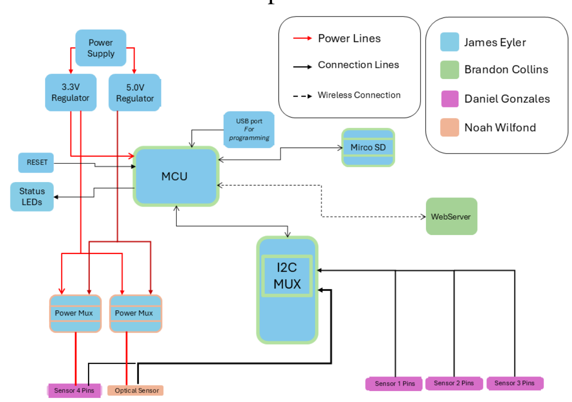


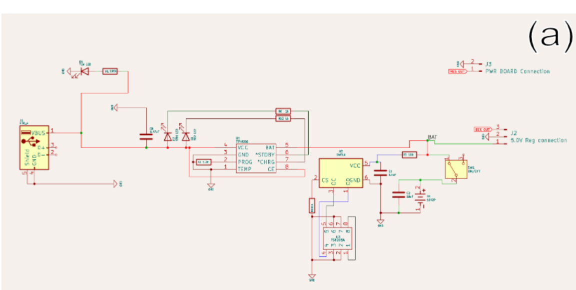
Fig. 1. Block Diagram of the OpenSense v3 Hardware showing power and data flow between systems.

All five sensor ports interface with the MCU through a TCA9548A 8-channel I²C multiplexer, allowing each port to be addressed individually without conflicts. Three ports operate at a fixed 3.3V, while the remaining two support user-selectable 3.3V or 5.0V operation through a GPIO-controlled switching chain. An MCP3221 ADC on each configurable port enables analog sensor support by converting outputs to digital values over the shared I²C bus. The MCU logs data locally to an onboard microSD card via SPI and can be powered and programmed through an onboard USB-A port. Status LEDs provide system feedback, including USB detection, a heartbeat indicator for MCU operation, and TX/RX activity for UART communication with the onboard USB-to-UART bridge. All control signals and peripheral connections are documented in the schematic to support future modification and community-driven development.

B. Power System

The system is powered by a 5.0V rail on the main board, supplied either through the onboard USB-A port or an attachable battery power module. The battery module uses a 18650 lithium-ion cell with a variable output voltage, which is regulated to 5.0V using an attachable boost regulator that accepts inputs from 2.8V to 4.2V. Power source selection is handled by a TPS2117 power multiplexer configured in priority mode, automatically selecting USB as the primary source and switching to battery power when the USB connection is removed, without requiring firmware intervention. An LM2596S buck converter then steps the 5.0V rail down to a stable 3.3V supply for the MCU, I²C multiplexer, and other low-voltage peripherals.

Both the 5.0V and 3.3V rails are routed to the dual-voltage sensor ports, where a TPS2117 configured in manual mode allows the user to select the appropriate supply voltage for each connected sensor through the web-based interface. Fig. 2 shows the schematic diagrams of the relevant power systems.



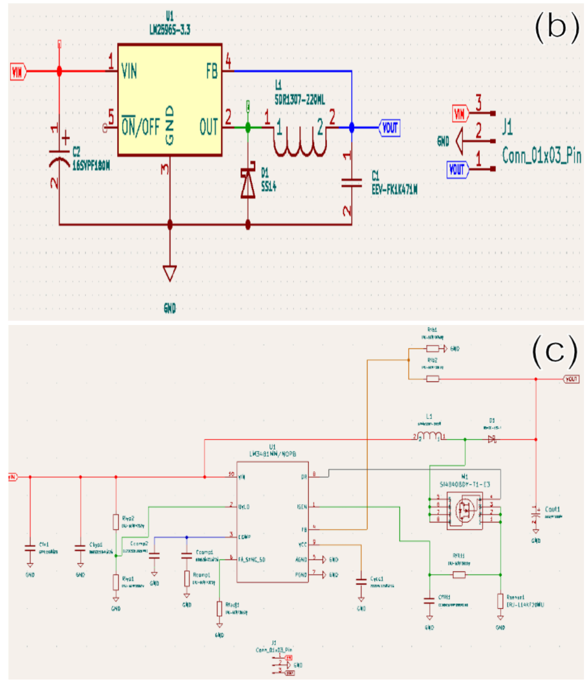


Fig. 2. Power supply schematics. (a) Battery Power + Charging PCB Design. (b) LM2596S 3.3 V Regulator. (c) LM3481 5V regulator

C. Path Switching

The complete voltage path switching system, illustrated in Fig. 3, uses the TS3A24157 analog switch, NTB0102DP level shifter, TPS2117 power multiplexor, and MCP3221 ADC to dynamically route between the 3.3V and 5.0V sensor domains while ensuring safe and reliable I²C communications. At the input side, the TS3A24157 routes the SDA and SCL lines to either voltage domain based on a GPIO signal, defaulting to 3.3V to protect the MCU on power-up. When the 5.0V path is selected, the SDA and SCL lines are switched to an alternate path that then passes through the NTB0102DP to handle bidirectional level shifting between the MCU and sensor logic levels. The TPS2117 independently selects the appropriate supply rail for the sensor VCC. Both the sensor and the MCP3221 ADC receive power from the selected rail, allowing analog sensors to be integrated into the system over the same shared I²C bus.

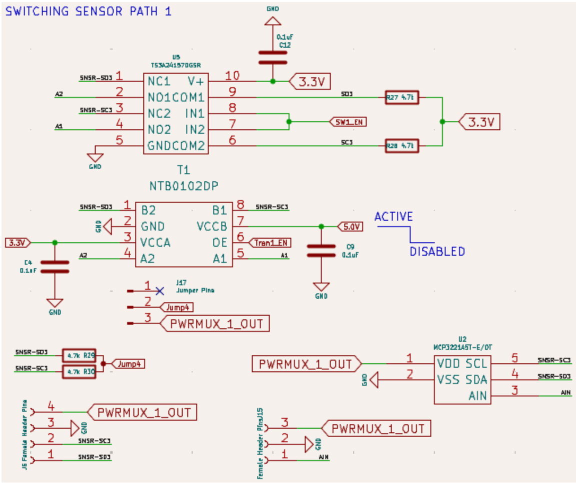


Fig. 3. Path switching system schematic.

D. Data Storage

Local data logging is handled by a DM3AT-SF-PEJM5 microSD connector interfaced to the ESP32. Sensor data is written to the card in CSV format, with a header row listing channel names and units, and sector-aligned batch writes used to avoid the multi-hundred-millisecond flash erase stalls that occur when small writes span sector boundaries. Logged files are accessible either by removing the card directly or by downloading them through the web interface, supporting both field and bench use cases.

E. Sensor Design

The four sensors below were redesigned using a combination of their major component's data sheets along with the designs of various breakout boards to find the best designs to use that would be functional as default sensors and allow easy testing for the main board.

The temperature sensor is on the simpler side and simply consists of the LMT87DCK with 2 decoupling capacitors to help with the output.

The pressure sensor consists of the 4-00015 FSR as well as a decoupling capacitor and pull up resistor which helps make the reading more precise with different levels of pressure.

The humidity sensor uses the HIH-6130 and then has 2 decoupling capacitors as well as 2 pull up resistors which can be disabled by an I^2C jumper just as a safeguard depending on if the main board's own pull up resistors are functioning.

The brightness sensor makes use of the VEML 7700 with a design similar to the humidity sensor involving 2 decoupling capacitors and 2 I^2C pullup resistors that can be disabled with the I^2C jumper.

F. Optical Monochromator Calculations

The monochromator is designed as a Czerny-Turner reflective system targeting a spectral range of 300-800 nm, with a center wavelength (λ_c) of 550 nm and a target spectral resolution of ≤ 5 nm. The component angles were determined from the grating pitch (G) and total diffraction angle (Φ) which is around 30° for a Czerny-Turner geometry [2]:

$$\alpha = sin^{-1}\left(\frac{\lambda_c G}{2cos(\frac{\Phi}{2})}\right) - \frac{\Phi}{2} \quad (1)$$

$$\beta = \Phi - \alpha \quad (2)$$

Solving for the condition where $\alpha \simeq \beta$ yields an incident angle of α = 9.58° and a reflected angle of β = 10.42°. The rotation required to scan the full wavelength range was derived by rearranging the reflection grating equation:

$$a[sin(\theta_m) + sin(\theta_i)] = m\lambda \quad (3)$$

The wavelength at the detector can be expressed as a function of the incident angle, allowing us to calculate how small changes in the grating rotation correspond to measurable shifts in wavelength. The resulting angle-to-wavelength relationship is plotted in Fig. 4, showing that a total grating rotation of approximately 14° is required to span the full 300–800 nm range.

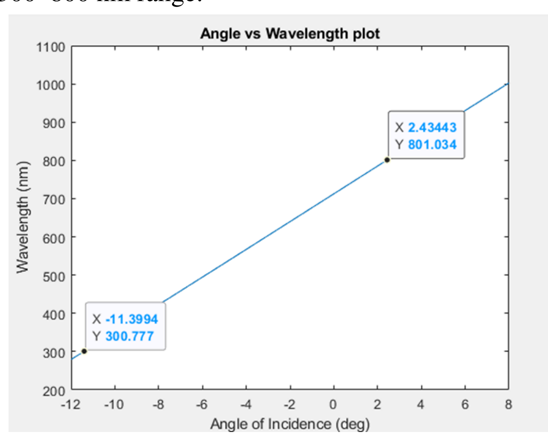


Fig. 4. Angle vs. Wavelength plot. Selected are the points which represent the edges of our target range, allowing us to determine the angle that the diffraction grating needs to rotate.

The photodiode signal chain begins with a transimpedance amplifier, which converts the photocurrent from the TEMD5080X01 photodiode into a measurable voltage. The expected optical power incident on the photodiode is calculated as the product of the incident intensity (P_{in}), system efficiency (η), and photodiode active area (A), as shown in equations (4) and (5):

$$P_{in} = \eta * I_{in} * A \tag{4}$$

$$I_p = P_{in} * S \tag{5}$$

The gain resistor (R_1) and feedback capacitor (C_1) of the transimpedance amplifier, shown in Fig. 5, are then sized using equations (6) and (7) to set the output voltage range and bandwidth respectively [3]:

$$R_1 = \frac{V_{o,max} - V_{o,min}}{I_p} \tag{6}$$

$$C_1 = \frac{1}{2\pi * R_1 * f_p} \tag{7}$$

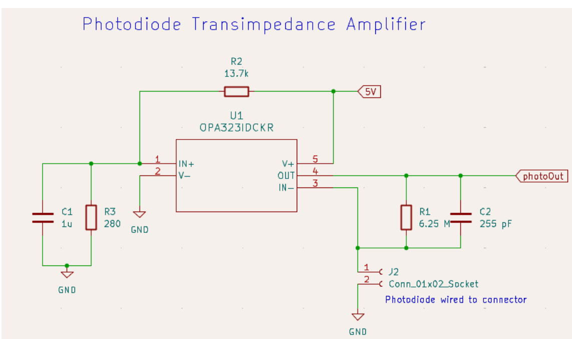


Fig. 5. Schematic of the transimpedance amplifier circuit.

An ATTiny84 microcontroller manages both the servo PWM signal used to rotate the diffraction grating and the readout of the transimpedance amplifier output, forwarding the combined wavelength and intensity data to the OpenSense platform over I²C, as shown in Fig. 6.

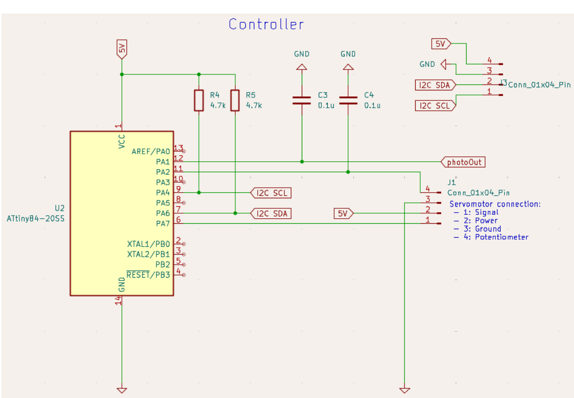


Fig. 6. Schematic of the servo-motor controller and I²C communication, being controlled by the ATTiny84.

IV. Software Architecture and User Interface

A. Software Architecture Overview

The OpenSense v3 firmware is built on the ESP32-WROOM using ESP-IDF and FreeRTOS. The software is organized as a set of cooperating services that separate sensor sampling, data logging, telemetry, driver configuration, and web interaction so that slower operations such as SD-card writes or browser updates do not interfere with time-sensitive sampling. This architecture supports the project goal of a modular educational platform while also making the firmware easier to maintain and extend.

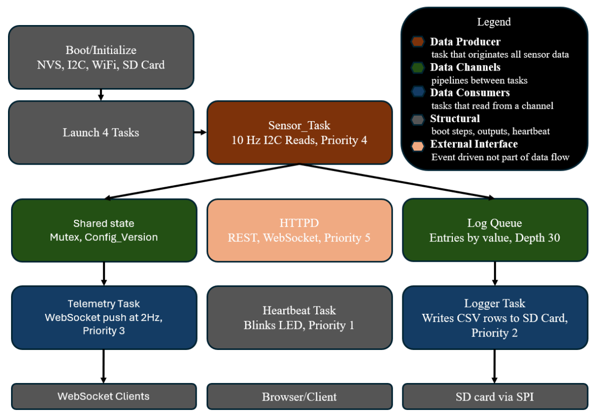


Fig. 7. Software Architecture Block Diagram

B. System Organization & Task Structure

At startup, the firmware initializes the hardware interfaces, loads saved sensor configurations from NVS, mounts the SD card if available, and then launches the runtime tasks. During normal operation, sensor data flows in one direction: drivers acquire measurements, the latest values are written into shared system state, log records are pushed into a queue, and telemetry frames are generated for connected web clients. This separation prevents network or storage latency from directly disrupting sensor reads.

The firmware uses multiple FreeRTOS tasks with priorities assigned according to timing importance. The sensor_task runs every 100 ms at priority 4 which is the highest priority user application task because it maintains the system's 10 Hz sampling rate. It reads each active sensor using the driver assigned by the saved configuration, converts raw values into engineering units, updates shared state, and queues log entries. The telemetry_task runs every 500 ms at priority 3 and packages the latest shared-state data into JSON for connected

WebSocket clients. The push rate is intentionally decoupled from the sample rate to avoid stalling the network stack when TCP send buffers are full. The logger_task runs at priority 2, blocks on the log queue, and writes CSV records to the SD card. The heartbeat_task runs at priority 1 and toggles a status LED as a simple runtime indicator. The HTTP server is handled by ESP-IDF's native default web stack at priority 5 and serves the dashboard, REST endpoints, and WebSocket connections. All user-created tasks are assigned priorities below the internal Wi-Fi and TCP/IP tasks so that network stability is preserved.

C. Timing and Reliability

The software prioritizes deterministic sampling by isolating sensor reads from slower subsystems. I²C transactions occur in the sensor task, while logging and telemetry operate on queued or cached data rather than directly on the bus. Configuration data is also cached per task and refreshed only when a configuration-version value changes, reducing repeated memory copies. In the logging path, fixed-size records are queued by value rather than allocated dynamically, avoiding heap fragmentation and limiting runtime jitter. Together, these decisions allow the platform to maintain predictable sampling behavior even when SD-card writes or browser activity vary in latency.

D. Hybrid Driver Model

A key feature of OpenSense v3 is its hybrid driver architecture. The firmware includes a generic driver intended to support most sensors through parameterized configuration, as well as several preset drivers for sensors that require custom initialization or conversion logic. The generic driver is designed to interpret metadata such as device address, register location, byte count, endianness, masks, scaling, offset, and units, allowing many I²C or analog sensors to be integrated without modifying the core firmware.

Preset drivers are included for devices that need more specialized handling, such as multi-step reads, conversion delays, or custom post-processing. This hybrid approach gives the platform two advantages: it remains usable out of the box with validated built-in drivers, while still supporting the larger project goal of user-defined sensor integration. In practice, this means new sensors can often be added either by creating a new parameter set for the generic driver or, when necessary, by writing a dedicated preset driver.

E. Logging and Web Interface

The SD-card logging subsystem is intentionally decoupled from sampling. Rather than writing directly to storage during acquisition, the sensor task only queues log entries, and the logger task performs the actual CSV writes. This protects the timing of sensor reads from unpredictable SD-card latency. On the user side, the ESP32 hosts a browser-accessible dashboard that provides live charts, sensor configuration controls, and CSV download capability. Telemetry is sent using WebSockets so that the interface can update in near real time without repeated polling.

F. Fault Tolerance

The architecture also supports graceful degradation. If the SD card cannot be mounted, the system may continue sampling and broadcasting telemetry without logging. If a particular sensor fails to initialize, the remaining channels can still operate. This modular behavior is important for an educational platform because it allows the system to remain demonstrable and useful even when one subsystem is unavailable.

V. Prototype Validation and Testing

A. Fabrication Status

Prototype fabrication focused on the main MCU board and the separate regulator boards. The main board’s fabrication was successful and was programmable allowing us to upload and test the firmware and verify heartbeat LED behavior. The 3.3 V and 5.0 V regulator boards were also fabricated and were reported to output their intended voltage and current levels. This result is important because it shows that the project progressed beyond paper design status

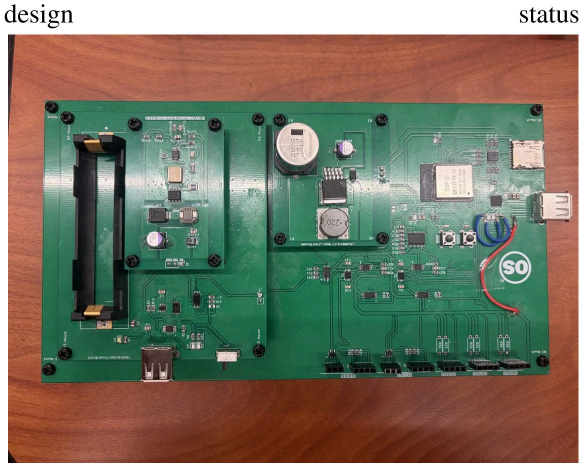


Fig. 8. Fabricated OpenSense v3 hardware, including the main board and regulator subsystems.

B. Power-System Validation

Power-path behavior was tested experimentally on a breadboarded power subsystem. When the system was powered by the power supply emulating the battery input, the output measured approximately 3.69 V, matching the expected battery-driven output. When powered by charger only or by charger plus power supply, the measured output was approximately 4.89 V, showing that the external input took priority. These results support the intended behavior of automatic source selection and indicate that the power architecture operated as designed in bench testing.

C. Regulator Validation

The regulator subsystem produced an important lesson in practical embedded power design. Initial attempts to validate switching regulators directly on breadboards were unsuccessful because switching converters are highly sensitive to parasitic inductance, board layout, and high-frequency loop behavior. Rather than treating this as a failed result, the team adapted by validating regulator behavior through simulation and revising design choices after additional review. This is a meaningful engineering result because it reflects an understanding of layout-dependent power electronics rather than only schematic-level behavior.

D. Optical Prototype Testing

The monochromator subsystem underwent at least two optical prototypes shown in Fig. 9. The first prototype used a simplified approach and demonstrated the sensitivity of system performance to geometric alignment, especially the positioning of the diffraction grating with respect to the axis of rotation. The second prototype corrected these alignment issues and improved the optical path. These tests confirmed both the usefulness of the underlying angle-to-wavelength calculations and the practical importance of alignment in turning a theoretical optical path into a functional instrument.

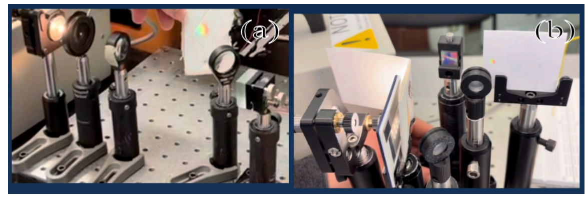


Fig. 9. Optical prototype evolution for the monochromator subsystem. (a) First prototype. (b) Second prototype

VI. Discussion

OpenSense v3 demonstrates a meaningful shift from fixed educational sensor kits toward a configurable platform architecture. Its most important contribution is not any single sensor, but the combination of modular hardware, mixed-interface support, and a software path intended to reduce the burden of adding new sensing devices. This matters in education because students learn more from platforms that reveal system structure and tradeoffs than from devices that simply report finished measurements. The platform's architecture also positions it well for future lab activities in embedded systems, instrumentation, and applied optics.

At the same time, the prototyping process exposed several practical limitations. One of the sensors was delayed by a PCB-footprint issue, and the regulator tests showed that high-frequency switching designs cannot be treated as breadboard-friendly circuits. The optical subsystem likewise showed that even a conceptually sound monochromator requires precise mechanical alignment to function as intended. These are not minor details; they are central lessons in turning design intent into working hardware. For a senior design conference paper, these lessons strengthen the work because they demonstrate system-level engineering rather than only idealized subsystem descriptions.

Cost remains an important consideration. The project targeted a future volume cost below $60 per unit for the main device, but the prototype-stage budget was much higher due to iterative PCB fabrication, testing hardware, sensors, and the monochromator subsystem. This distinction should be preserved: OpenSense v3 is intended to be low-cost in scalable deployment, but the current prototype cost reflects development-stage realities rather than finished manufacturing economics.

VII. Conclusion

This paper presented OpenSense v3, a modular open-source sensor platform developed for educational experimentation and adaptable instrumentation. The system was designed to address the limitations of expensive and closed commercial sensor platforms by combining five-channel sensor support, mixed analog and I²C compatibility, selectable voltage paths, local CSV logging, and a browser-based user workflow. The architecture is centered on an ESP32-WROOM MCU and extends beyond typical classroom sensing through the inclusion of a custom monochromator-based optical subsystem.

The project achieved important design and prototyping milestones. The main board and regulator subsystems were fabricated, core board bring-up was demonstrated, power-path behavior was tested, and optical prototypes were assembled and refined. Although some elements such as full end-to-end integration remained incomplete at the time of reporting, the work establishes a strong foundation for a reusable educational sensing platform. The value of OpenSense v3 lies in its transparency, extensibility, and potential to help students move from simply using sensors to understanding and building sensing systems themselves.

Future work should focus on finalizing the user-configurable sensor framework, validating sustained multi-sensor operation at the target rate, and further refining spectrometer alignment and spectral calibration. With those improvements, OpenSense v3 can become not only a successful senior design prototype but also a practical platform for future classroom and laboratory deployment.

Acknowledgement

The authors acknowledge the support of the Design of Resilient Architectures for Computing (DRACO) Lab and thank Dr. Mike Borowczak for project sponsorship and guidance. The authors also thank the faculty advisors and reviewers who supported the development, design review, and presentation of this project.

Reference

[1] J. Bartlett, R. Colon, and J. Ecle, “Open Sense Sensor Platform,” Senior Design - Group 21. [Online]. Available: https://www.ece.ucf.edu/seniordesign/fa2023sp2024/g21/index.html

[2] Ibsen Photonics, “Spectrometer Design Guide,” Ibsen Photonics A/S, Farum, Denmark, White Paper, 2023. [Online]. Available: https://ibsen.com/wp-content/uploads/ebook/spectrometer-resources/Spectrometer-Design-Guide.pdf

[3] J. Caldwell, “1 MHz, Single-Supply, Photodiode Amplifier Reference Design,” Texas Instruments, TIDU535, Oct. 2014. [Online]. Available: https://www.ti.com/lit/ug/tidu535/tidu535.pdf?ts=1774279277270