AN AUTOMATED APPROACH TO EXTERNAL VENTRICULAR DRAINS

By

David Kapelyan

Isiah Lashley

Ralph Nathan

Final Report for ECE 445, Senior Design, Spring 2025

TA: Jason Jung

7 May 2025

Project No. 66

Abstract

This project presents a prototype system for automating cerebrospinal fluid (CSF) drainage in patients requiring External Ventricular Drains (EVDs). The device integrates a dual-sensor configuration, pressure and flow, connected to a microcontroller that governs solenoid valve actuation based on user-defined thresholds. A graphical user interface (GUI) enables real-time data monitoring and system manual override. The system converts a traditionally external setup into a compact, palm-sized, digitalized device mountable at the level of the external auditory meatus (EAM), eliminating the need for manual zeroing of the pressure sensor and enabling continuous data logging of intracranial pressure (ICP) and CSF flow. A fail-safe normally-closed valve ensures safety during power loss or system failure. While not yet intended for clinical use, the prototype demonstrates reliable pressure-flow monitoring and responsive valve control, offering a functional foundation for future development of safer, more automated EVD systems.

Keywords—External Ventricular Drain (EVD), cerebrospinal fluid (CSF), intracranial pressure (ICP), external auditory meatus (EAM), solenoid valve control, pressure sensor, flow sensor, microcontroller, biomedical device, automation, graphical user interface (GUI).

Contents

1 Introduction

1.1 Problem

External Ventricular Drains (EVDs) are commonly used to manage intracranial pressure (ICP) by diverting excess cerebrospinal fluid (CSF) or blood in acute conditions such as hydrocephalus, traumatic brain injury, and meningitis. The brain produces 450–500 cc of CSF daily, with a typical volume of approximately 150 cc, requiring precise regulation. Traditional EVDs are external, gravity-based systems that must be aligned/zeroed at the external auditory meatus (EAM) to ensure accurate pressure monitoring. However, patient movements such as turning, transport, coughing, or repositioning can lead to misalignment, leading to overdrainage of up to 60–80 cc within minutes and posing risks of hemorrhage, brainstem herniation, or death. Manual clamping and re-zeroing during repositioning disrupt continuous drainage and increase the likelihood of human error. Furthermore, outside specialized neurocritical care units, staff unfamiliar with EVD management face a heightened risk of errors, including improper re-zeroing or inadvertently lowering the drain height, which can critically compromise safe CSF regulation.

1.2 Solution

Our proposed system automates CSF drainage, enhancing safety, stability, and reducing manual intervention. Pressure and flow sensors communicate with a microcontroller, controlling a solenoid valve to regulate drainage based on intracranial pressure (ICP). The valve opens only when ICP exceeds a user-defined threshold and closes when ICP returns to safe levels, preventing overdrainage despite patient movement or transient pressure spikes caused by actions such as sneezing. Additionally, the flow sensor ensures drainage rates do not exceed user-preset limits, automatically closing the valve if exceeded and requiring manual reopening after the collection bag is emptied. The External Ventricular Drain (EVD) is fixed level with the external auditory meatus (EAM) and secured to the mastoid bone, eliminating manual zeroing and further minimizing error risks.

1.3 Visual Aid

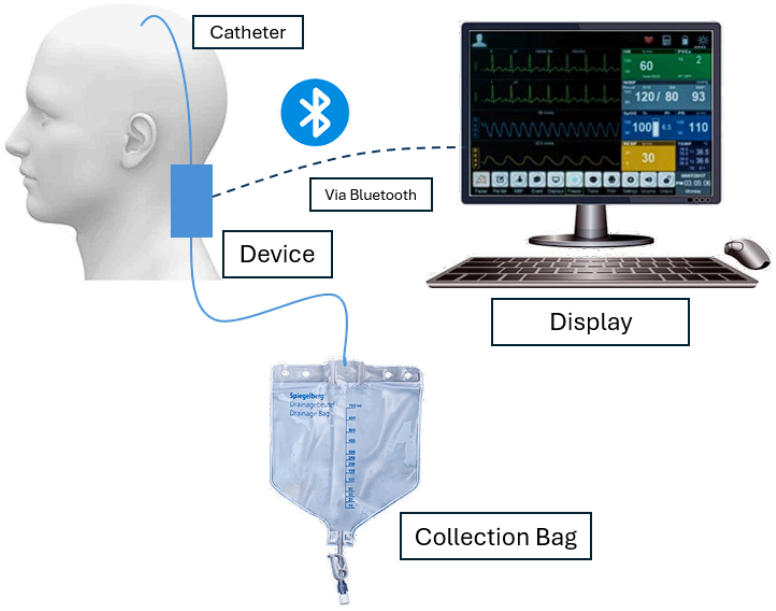

Figure 1: Visual aid of device physical design and placement

To eliminate the need for manual zeroing relative to the external auditory meatus (EAM), the device is designed to be securely mounted to the patient's mastoid bone, establishing a fixed and consistent reference point (Figure 1). The system facilitates cerebrospinal fluid (CSF) drainage through tubing connected to a collection bag and interfaces with a graphical user interface (GUI) via Bluetooth. The GUI enables clinicians to configure pressure and flow thresholds, manually reset the solenoid valve, and monitor real-time pressure and flow measurements. Additional features include graphical data visualization, alarm notifications for abnormal readings, and data logging capabilities for clinical review and documentation.

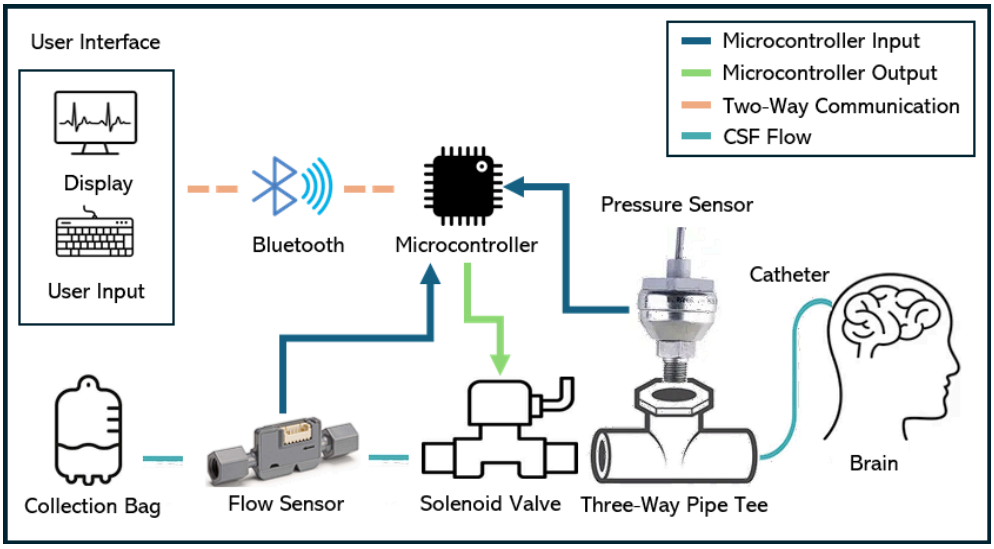

Figure 2: Visual aid of the overall system design

Our design features a three-way pipe tee to which the catheter is connected, with a pressure sensor mounted on top, perpendicular to the CSF flow, and a solenoid valve at the opposite end, which remains closed by default in the absence of a driving voltage. As CSF fills the tee, the pressure sensor, in direct contact with the fluid, measures the CSF pressure, corresponding to the patient's intracranial pressure (ICP), and transmits the data to a microcontroller. If the detected pressure exceeds the maximum user-set value, the microcontroller activates the solenoid valve to allow CSF drainage. Once the pressure drops to the minimum user-set threshold, the microcontroller signals the solenoid valve to close, halting drainage.

CSF flows through the solenoid valve to a flow sensor via a tube before reaching a collection bag. The flow sensor monitors the drainage rate and transmits the data to the microcontroller. If the flow rate exceeds a user-set limit, typically 10 cc/hour, the microcontroller signals the solenoid valve to close, stopping drainage until manually reopened, typically when the nurse empties the collection bag, to ensure controlled CSF removal.

1.4 High-Level Requirements

For our project to be successful, the system must meet three key requirements:

1. Automated Flow Regulation: The system must autonomously control CSF drainage via a solenoid valve. The valve should open when the intracranial pressure (ICP), measured by a pressure sensor, exceeds a user-set threshold (e.g., 16 mmHg) and close when it reaches a maximum user-set limit (e.g., 100 mmHg).
2. Flow Rate Sensing: The system must limit the flow rate to 10 ccs per hour, using a flow rate sensor to monitor and adjust drainage via the solenoid valve accordingly.
3. Real-Time Monitoring: The microcontroller must support Bluetooth communication with a display monitor, providing near real-time data with a maximum delay of 10-30 seconds (as approved by Dr. Pappu).

2 Design

2.1 Block Diagram

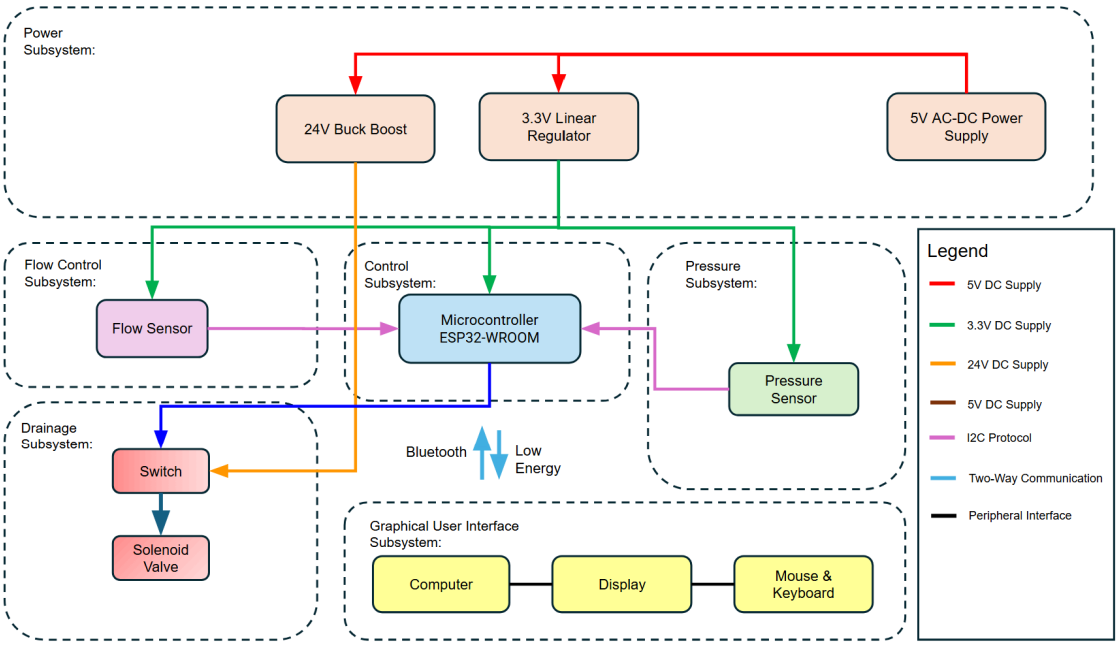

Figure 3: Block Diagram

2.2 Physical Design

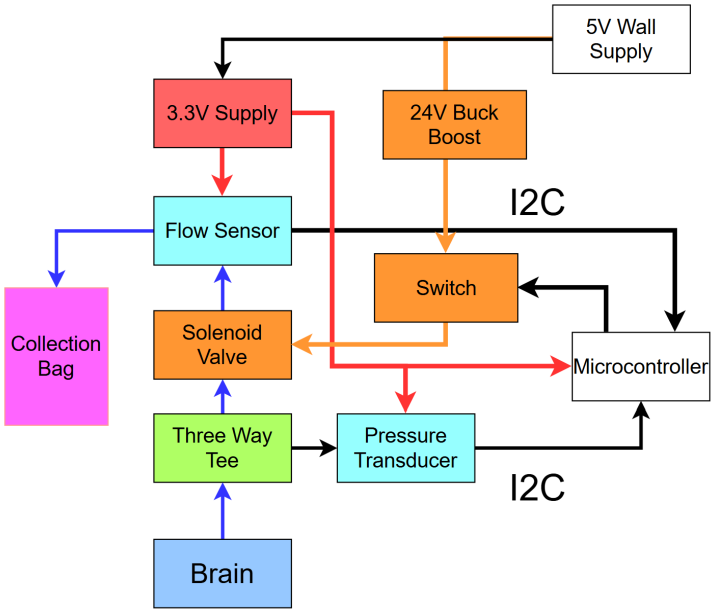

Figure 4: Physical Design

The mechanical design of the system consists of several stages. The first stage includes a three-way tee connected to the input, a pressure transducer, and a solenoid valve. The solenoid valve is then connected to a flow sensor in line with the collection bag. By placing the flow sensor at the valve's output, we ensure that we measure only CSF outflow, preventing any interference from the initial fluid buildup in the first stage.

2.3 Subsystem Description & Requirements

2.3.1 Microcontroller Subsystem

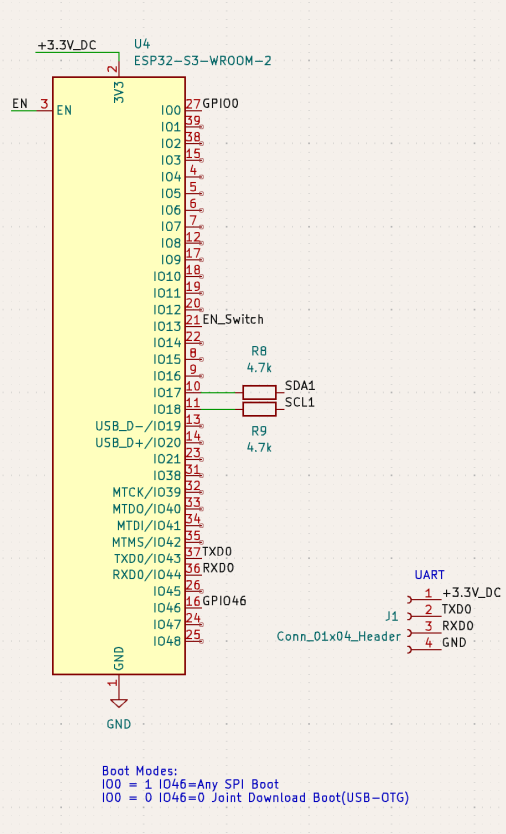

Figure 5: ESP32 circuitry

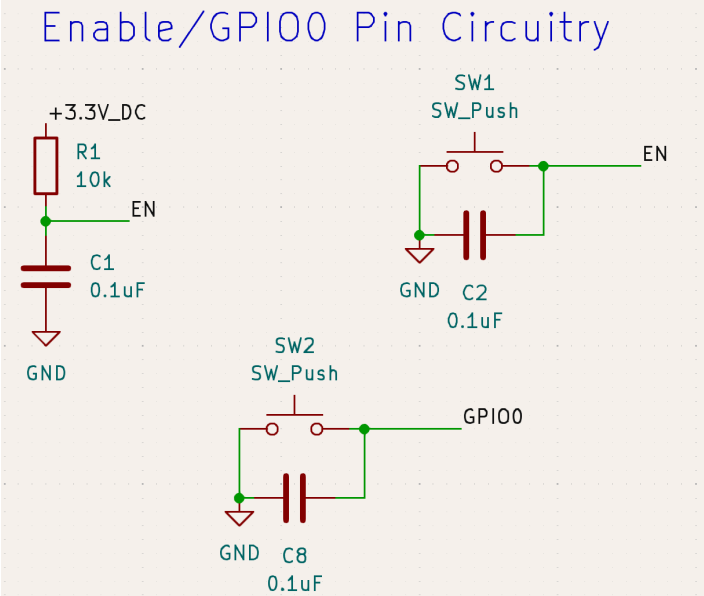

Figure 6: Enable control circuitry for ESP32

The Microcontroller Subsystem serves as the central processing unit, managing sensor inputs and control outputs to enable automated, real-time regulation of CSF drainage. The pressure sensor outputs an analog voltage, which is converted to a digital signal via an external Analog-to-Digital Converter (ADC). The microcontroller reads the ADC output via the I²C protocol, translating it into pressure readings. It also receives flow rate data from the flow sensor and processes both inputs to determine appropriate valve actuation. If the pressure or flow rate exceeds user-set thresholds, the microcontroller outputs a voltage signal to drive the solenoid valve, controlling CSF drainage. Additionally, it interfaces with the Graphical User Interface (GUI) via Bluetooth, transmitting real-time pressure and flow data. We are utilizing the Espressif ESP32-WROOM-32 microcontroller, which features built-in Wi-Fi and Bluetooth, making it well-suited for wireless communication. The ESP32's integrated RF transceiver ensures reliable data transmission for seamless system monitoring and control.

2.3.2 Power Subsystem

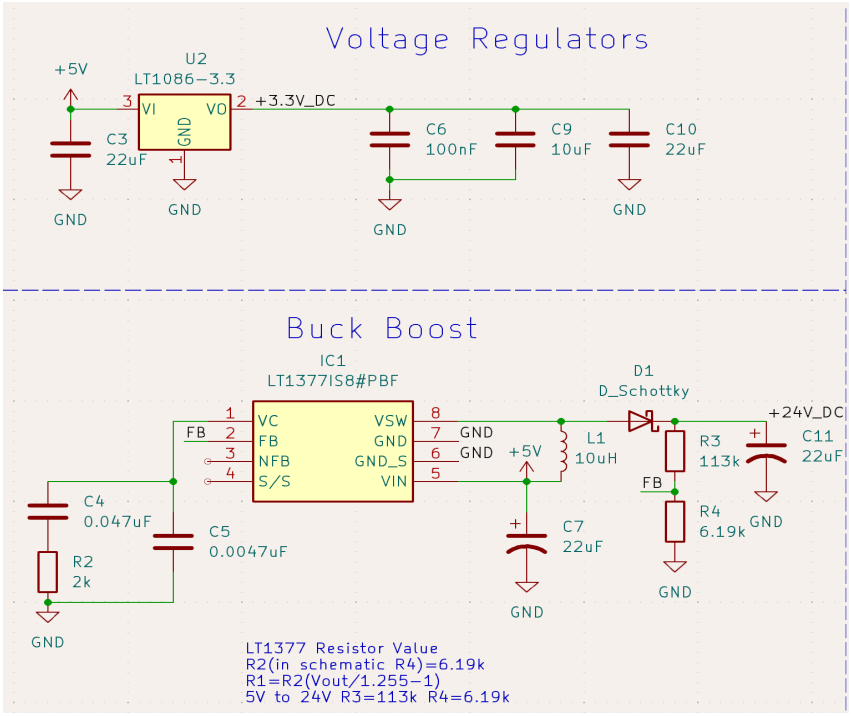

Figure 7: Power subsystem schematic

The Power System Circuit ensures a stable and reliable power supply to all subsystems, enabling continuous operation of the microcontroller, pressure and flow sensor, and solenoid valve. Given the medical nature of the system, power reliability is crucial.

We used a 3.3V linear regulator to step down a 5V input into our circuit board. The 3.3V linear regulator is utilized to power the microcontroller, pressure sensor, and flow sensor. The 3.3V linear regulator utilized was the Analog Devices LT1086, which has a current rating of 1.5A. For this project, a linear regulator with a maximum current output of 259mA was necessary as the board had to be designed with the assumption that the ESP32 would potentially operate at its max current rating, which is 250 mA and the sensors draw 9mA of current.

The solenoid used consumes 2W of power at 24V, drawing 83.3mA of current. To power the solenoid, a 24V rail is created by using a buck-boost to step up the 5V input to 24V. The buck boost chosen was the Analog Devices LT1377, which has a current rating of 1.5A. A buck boost with a current rating significantly greater than the minimum specification was chosen to ensure that there were no issues with non-idealities coming into play as a result of high current draw, which can decrease the maximum gain of the buck-boost.

We chose a linear regulator over switching regulators, such as a buck converter, because minimizing noise inputted into a microcontroller is generally recommended for proper functioning. Switching regulators have ripple noise on their output, which can cause microcontrollers to not function properly.

2.3.3 Drainage Subsystem

The solenoid valve regulates cerebrospinal fluid (CSF) flow based on pressure readings, enabling automated drainage only when necessary. It opens when intracranial pressure (ICP) exceeds a user-defined upper threshold and closes once pressure falls below a user-defined lower threshold, thereby preventing overdrainage.

The selected solenoid valve is a normally closed (NC) type, meaning it remains closed by default and automatically returns to the closed position in the event of a power loss, ensuring fail-safe operation. It operates on a 24V DC power supply, has a power rating of 2W, and requires an input current of approximately 83 mA.

To control the valve, a switch is placed between the 24V rail and the solenoid input, driven by a digital output signal from the microcontroller. This configuration enables precise, real-time control of CSF flow based on sensor feedback.

2.3.4 Pressure Subsystem

The pressure sensor measures intracranial pressure (ICP) by detecting the hydrostatic pressure transmitted through the cerebrospinal fluid (CSF) in the drainage catheter. The sensor used is the **Amphenol NPI-19J-015A2**, which has a measurement range of 0 to 15 psi (0 to 775.7 mmHg) and outputs a 14-bit digital signal via I^2C with an accuracy of ±1%. It is powered using a 3.3V linear voltage regulator.

This transducer is piezoresistive, meaning it senses pressure by measuring resistance changes in a piezoresistive material. As the diaphragm deforms under pressure from the CSF, the material's resistance changes proportionally, allowing the sensor to determine fluid pressure. The transducer is connected to a three-way tee, which is in direct hydraulic continuity with the brain’s ventricles via the catheter. When the solenoid valve is closed, this configuration ensures stable and accurate ICP readings. Since fluctuations as small as 1 mmHg (0.019 psi) must be detected, a high-precision sensor is essential for safe and effective system operation.

2.3.5 Flow Control Subsystem

The flow rate sensor monitors cerebrospinal fluid (CSF) drainage to prevent excessive flow, typically defined as greater than 10 cc/hour. If the flow rate exceeds the user-defined threshold, the microcontroller activates a voltage-controlled switch positioned between the 12V power rail and the solenoid valve. By default, the switch remains open (non-conductive), which means a closed solenoid valve; when a control signal is applied, the switch closes, allowing current to flow to the solenoid and open the valve accordingly.

The system uses the Sensirion SLF3S-0600F, a low-power liquid flow sensor that operates using thermal mass flow measurement. It outputs 16-bit digital data via an I^2C interface, with the flow rate (in µL/min) proportional to the output value. The sensor supports a differential pressure range of −600 Pa

to +600 Pa and offers high accuracy, with an offset error of ±0.5 Pa and full-scale error of ±3 Pa, ensuring reliable monitoring of CSF flow.

The SLF3S-0600F operates at 3.3V, making it fully compatible with the ESP32 microcontroller. Its digital I^2C output eliminates the need for external analog-to-digital conversion, simplifying system integration and reducing signal noise.

2.3.6 Graphical User Interface Subsystem

The Graphical User Interface (GUI) provides an interactive platform for medical professionals to monitor intracranial pressure (ICP), cerebrospinal fluid (CSF) flow rate, and solenoid valve status in real time. It also supports data logging and manual override of the solenoid valve, enhancing patient safety. The system utilizes the Espressif ESP32-WROOM-32 microcontroller, which features built-in Wi-Fi and Bluetooth; communication with the GUI is established via Bluetooth Classic. This connection enables the GUI to receive sensor data and transmit control commands to the system. The GUI is cross-platform, compatible with Windows, Linux, and macOS, and is developed in Python using the Tkinter library.

3. Design Verification

The verification process was conducted to ensure that the prototype functioned as intended. This involved individually testing and validating the performance of the pressure sensor, flow sensor, GUI, and microcontroller. Once each component was verified, they were integrated and tested as a complete system to confirm that the prototype operated correctly as a whole.

3.1 Microcontroller Subsystem

The system must support both I^2C and Bluetooth communication to enable real-time acquisition of sensor data and wireless transmission to external devices, such as a PC or mobile application. This ensures seamless integration with various sensors and enhances system flexibility. The ESP32 firmware should be developed, verified, and uploaded using the Arduino IDE, with all outputs monitored through the Serial Monitor to confirm proper functionality.

During testing, the Serial Monitor displays real-time data for pressure, flow, and Bluetooth connection status, providing immediate feedback on sensor readings and system connectivity. The graphical user interface (GUI) must accurately reflect the data received from the ESP32 via Bluetooth, ensuring real-time visualization and validation of sensor functionality.

3.2 Power Subsystem

The system must continuously supply at least 25 mA at 5V ± 0.1V to support the connected output components. This voltage regulation is essential to ensure the stable operation of downstream devices that depend on a consistent power source.

To verify this requirement, GPIO 32 on the ESP32 is connected to the enable pin of the buck-boost converter. A digital multimeter is used to measure the converter's output voltage during

system operation, confirming that it remains within the specified 5V ± 0.1V range. This test validates both the proper enablement of the converter and its ability to maintain the required output voltage under load.

3.3 Drainage Subsystem

The solenoid valve must exhibit a response time of less than one second upon receiving a HIGH or LOW signal from the ESP32. This quick response is critical for promptly controlling fluid flow, ensuring system responsiveness, and operational safety. To verify proper operation, the voltage across the solenoid terminals is measured using a digital multimeter during both signal states, confirming that the valve receives the appropriate actuation voltage.

For testing, the user issues an “OPEN” or “CLOSE” command via the graphical user interface (GUI), which triggers the ESP32 to set GPIO32 to HIGH or LOW, respectively. Water is poured into the system’s input to observe flow behavior. The GUI should accurately indicate the valve’s state, and water should only flow into the collection container when the valve is open. Coupled with voltage measurements across the solenoid, this test confirms that both the electrical actuation and mechanical response of the valve are functioning within the required specifications.

3.4 Pressure Subsystem

The system must maintain an accuracy of at least ±1 mmHg to ensure precise pressure regulation within the cerebrospinal fluid (CSF) drainage system. Upon powering on, the ESP32 begins reading pressure data from the Amphenol NPI-19J-015A2 sensor, which should initially detect and display atmospheric pressure. Users can manually set a pressure threshold, allowing the system to determine when drainage is required based on real-time measurements.

To verify the sensor’s accuracy, two validation methods were employed. First, the sensor’s measured pressure was compared to known atmospheric pressure while exposed to open air. Second, hydrostatic pressure variation was tested by vertically displacing the sensor (in the z-direction) and observing the corresponding pressure change.

To eliminate air from the system and ensure accurate hydrostatic behavior, a pump-driven setup was devised to fill the tubing with water. The input tube was elevated above the three-way tee and submerged in a bowl of water. The tee’s output was connected to a needle valve, which was linked to a pump that discharged into another water-filled bowl. When the pump was activated, it forced water upward through the system, displacing trapped air. Once the tubing was fully primed, the needle valve was closed and the input tube sealed, creating a closed fluid-filled system. This configuration allowed for controlled hydrostatic pressure changes. A lab jack was used to precisely adjust the height of the tee, enabling accurate measurement of pressure variations and validation of the sensor’s performance.

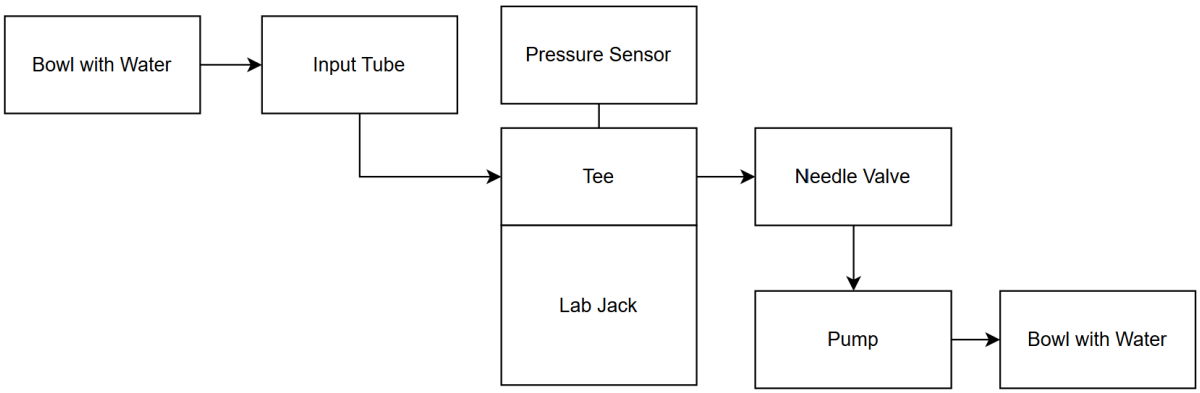

Figure 8: Fluid pressure testing setup

The hydrostatic pressure of a liquid varies as a function of height and is governed by the equation $P = \rho gh$, where ρ is the fluid density, g is the acceleration due to gravity, and h is the height of the fluid column. By measuring pressure at different heights, the sensor's ability to accurately detect fluid pressure was verified. This was confirmed by validating the linear relationship $\Delta P = \rho g \Delta h$, demonstrating that pressure changes were proportional to height changes, as expected. Once the sensor's performance was validated, its resolution was calculated based on its 14-bit digital output. Given a measurement range of 0 to 775.7 mmHg, the resolution is $775.7 \text{ mmHg}/2^{14} = 0.0473$ mmHg, confirming its suitability for detecting small pressure fluctuations in the CSF drainage system.

3.5 Flow Control Subsystem

The system must achieve a flow measurement accuracy of ±5% for water at rates up to a full-scale value of 2 mL/min; for flow rates exceeding this range, a reduced accuracy of ±10% is acceptable. To validate this requirement, 167 µL of water is measured using a micropipette and introduced into the system's input tubing via a funnel. As the fluid moves through the system, the flow sensor captures real-time data, which is transmitted by the ESP32 to the graphical user interface (GUI) via Bluetooth.

The GUI displays the measured flow rate in cubic centimeters per minute (cc/min) in both text and graphical formats, enabling real-time visualization. Additionally, it calculates and displays the total volume delivered per hour (cc) as a separate text field. This dual representation helps verify that the system meets the specified flow accuracy requirements and provides users with clear, actionable data.

3.6 Graphical User Interface Subsystem

The system must display real-time pressure and flow data with a delay no greater than 10–30 seconds to ensure timely monitoring and clinical decision-making. To initiate the test, Bluetooth is enabled on the ESP32 via the Arduino IDE, and a Python executable establishes communication between the ESP32 and the graphical user interface (GUI). Upon successful connection, a pop-up window appears

displaying real-time graphs of pressure and flow sensor data. Additional on-screen indicators include live readings of flow rate, pressure, total volume, and the solenoid valve status, along with a text box that logs sensor updates every second.

System performance is verified by monitoring the text box to ensure it reflects near-instantaneous sensor values and accurately indicates the current state of the solenoid valve (open or closed). Users may also export a complete log of the session in CSV format, providing a detailed, time-stamped record of pressure, flow, and valve behavior for further analysis or documentation.

4. Costs

4.1 Parts

Table 1 Parts Costs

Part	Manufacturer	Retail Cost ($)
NPI-19J-015A2 Pressure Sensor	Amphenol Advanced Sensors	$78.85
SLF3S-0600F Flow Sensor	Sensirion AG	144.27
AMV-ENM-24-01 Solenoid Valve	BMT Fluid Control Solutions	93.61
1/8 NPT to M3	Mcmaster Carr	7.98
1/4 -28 UNF to 5/32 push connect	Mcmaster Carr	7.90
1/8 NPT to 5/32 push connect	Mcmaster Carr	2.92
1/4-28 UNF to 1/8 NPT	Mcmaster Carr	15.48
1/8 NPT Tee	Taisher	12.99
ESP32 Dev Board	ELEGOO	19.99
Buck boost module	WWZMDiB	9.99
5/32 plastic tubing	uxcell	6.19
20-200 uL micropipette	ONiLAB	26.99
5V AC-DC Adapter	GuanTing	6.99
3D-Printed Enclosure		5.00
4.7K Ohm Pull-up Resistors	Yageo	0.50
Lab Jack		14.99
Water Pump	HSH-Flo	19.99
Zip Ties	Harbor Freight	5.00
Screw Terminal Block	Tnisesm	9.99
Labor		40 per hour * 500 = 20000
Total		**20345.35**

4.2 Labor

Table 2 Labor Breakdown

Category	**Hours per person**		
	David	Ralph	Isiah
Circuit Design and Construction	120	20	20
Testing and Debug	60	70	40
Logistics/Documentation	60	70	50

We dedicated approximately 10 hours per week per team member to this project, representing a significant portion of our time and effort. Driven by a strong motivation to see the project succeed, we invested considerable energy into the design and implementation process. Over the course of 11 weeks, we estimate that our team collectively spent approximately 500 hours working on the development, testing, and refinement of the system.

5. Conclusion

This project successfully developed a prototype system for automating cerebrospinal fluid (CSF) drainage in external ventricular drains (EVDs) to enhance patient safety, improve clinical workflow efficiency, and minimize manual intervention. By integrating pressure and flow sensors with a microcontroller-controlled solenoid valve, the system is capable of regulating CSF outflow in real time based on user-defined intracranial pressure (ICP) and flow thresholds. The addition of Bluetooth communication and a graphical user interface (GUI) enables continuous monitoring, user configuration, and data logging capabilities. This automation reduces the risks associated with overdrainage, eliminates the need for manual leveling at the external auditory meatus (EAM), and streamlines clinical oversight.

5.1 Accomplishments

The key accomplishments of the project are summarized as follows:

- **Improved Patient Safety**: The system autonomously controls the solenoid valve through real-time communication with both pressure and flow sensors to prevent overdrainage or underdrainage, enhancing stability and reducing risk during patient movement or sudden ICP fluctuations.

- **Reduced Clinical Workload**: Manual recording of drainage volumes and ICP values was mitigated through the implementation of an integrated digital data logging system, enabling automated storage and export of measurements in .csv format, thereby reducing clinical workload and minimizing the likelihood of human error.

- **Accurate Setup and Zeroing**: As a prototype, our current design represents an initial step toward miniaturization. While already significantly more compact than traditional fully external systems, the palm-sized device can be anchored at the EAM/mastoid level, eliminating the need for manual zeroing or re-leveling and improving both positional accuracy and operational consistency.

- **Real-Time Monitoring and Logging**: The system continuously tracks real-time pressure, flow, and cumulative volume while transmitting data over Bluetooth Classic to a Python-based GUI with a delay of less than 30 seconds. Data logging features support exportable CSV files and hourly flow rate resets.

These outcomes demonstrate the feasibility and clinical value of an automated, sensor-based EVD system and lay the groundwork for future integration into patient care environments.

5.2 Failures

We were unable to get our PCB to function correctly. Upon reviewing the design after the board had been discarded, we concluded that the issue most likely stemmed from an error in how the ESP32 was programmed. Specifically, we failed to recognize that when using a UART adapter, the Rx0 and Tx0 pins must be cross-connected, Rx0 to Tx0 and Tx0 to Rx0. A comparison of our schematic with an official reference design from Espressif confirmed that the circuit itself was correct. Alternatively, the failure may have resulted from a soldering error, such as incorrect placement of a component within the ESP32's programming circuitry or thermal damage to the ESP32 due to excessive heat during manual soldering.

5.3 Ethical considerations

During the development of our **External Ventricular Drain (EVD) Automation Project**, we prioritized ethical engineering and regulatory alignment with IEEE and medical device standards. Although the system is a non-clinical prototype, the risks of excessive or insufficient cerebrospinal fluid (CSF) drainage demanded a strong focus on patient safety and system reliability. In line with the **IEEE Code of Ethics (2025), Clause I.1**, we integrated fail-safes such as a normally-closed solenoid valve that defaults to a closed state during power or microcontroller failure, and a GUI-enabled manual override for emergency control. The GUI also displays real-time valve status to support user awareness.

For future clinical deployment, the system would require compliance with **FDA 21 CFR Part 820** and likely a **510(k) premarket notification** to establish equivalence to existing devices. By proactively addressing these ethical and regulatory considerations, we aimed to uphold the highest standards of safety, accountability, and professional integrity.

5.4 Future work

The most important improvement that could be made to the system is to make it lighter and more compact. Our current design utilizes an enclosure that is space-inefficient and excessively bulky for use on a real patient. Additionally, the use of metal pipe adapters instead of plastic ones was an oversight that significantly increased the system's overall weight.

A practical solution to make the system viable for real-world use would be to divide it into two parts. This separation would allow the portion mounted to the patient's head to be smaller and lighter. The first part would be housed in a compact enclosure secured to the patient's head, while the second part would be located downstream of the drainage tube and connected to the flow line. The two enclosures would be linked via wiring. The head-mounted enclosure would contain the pressure sensor, input tee, and plastic tubing inputs. The second enclosure would house the flow sensor, PCB, and solenoid valve. By separating these components, only the necessary elements are mounted to the patient's head, thereby improving patient comfort and enabling more secure attachment.

The device should also be modified to be MRI-safe. Since EVDs are often used in settings where MRI scans are necessary to assess intracranial conditions, the device must not contain ferrous metals such as iron or nickel. This can be achieved by replacing the pipe tee adapters with plastic versions, selecting a solenoid valve made from plastic and bronze (e.g., BMT AMV-MVN-24-01), and using a pressure sensor with a titanium housing (e.g., TE Connectivity XP5).

Lastly, the graphical user interface (GUI) could be improved for better usability. While functional in its current state, the GUI is unintuitive for new users. The interface should be redesigned to include individual text boxes for each input parameter, replacing the existing command-line style input terminal. Additionally, override commands should be implemented as clickable buttons rather than text inputs. These updates would significantly enhance the system's ease of use, making the interface more intuitive and reducing the need for extensive user training.

5.5 Special Thanks

We would like to extend our sincere thanks to our TA, **Jason Jung**, our course supervisor, **Dr. Yang Zhao**, and our project advisor, **Dr. Suguna Pappu**, for their invaluable support throughout the development of this project. Their mentorship and guidance were instrumental to our progress, and we are deeply grateful for their contributions. We hope to continue refining and advancing this project toward its full potential.

References

[1] IEEE, “IEEE Code of Ethics,” ieee.org, https://www.ieee.org/about/corporate/governance/p7-8.html. Accessed 13 Feb. 2025.

[2] Bertuccio, Alessandro et al. “External Ventricular Drainage: A Practical Guide for Neuro-Anesthesiologists.” *Clinics and practice* vol. 13,1 219-229. 31 Jan. 2023, doi:10.3390/clinpract13010020

[3] FDA Class II Medical Devices, “FDA Class II medical devices. (n.d.).”, https://www.rimsys.io/blog/fda-class-ii-medical-devices. Accessed 6 March 2025.

[4] The Python Standard Library, “Graphical User Interfaces with Tk”, https://docs.python.org/3/library/tk.html. Accessed 20 March 2025.

[5] SLF3S-0600F Liquid Flow Sensor. SLF3S-0600F Liquid Flow Sensor.

[6] NovaSensor NPI-19 Digital I2C Pressure Sensor.

Appendix A Requirement and Verification Table

Table 3 System Requirements and Verifications

Subsystem	Requirement	Verification	Verification status (Y or N)
1. Microcontroller Subsystem	1. Must support I²C and must Support bluetooth communication for sensor data acquisition and real-time transmission	1. Serial Monitor will display results of pressure , flow, and also bluetooth connection availability. 2. GUI reflects data from ESP32 (communicated via Bluetooth)	Y
2. Pressure Control Subsystem	1. Must have an accuracy of at least ±1 mmHg for precise pressure regulation.	1. Results displayed in Serial Monitor on Arduino IDE 2. GUI reflects atmospheric pressure and also shows change in pressure on graph from liquid input 3. Average pressure is also displayed on the GUI.	Y
3. Flow Control Subsystem	1. Must have an accuracy of 5% for water for 2 ml/min full scale flow rate otherwise 10% accuracy	1. Flow Data (cc/min) will be displayed in text and graph in the GUI 2. Total Volume / Hour (cc) will be displayed in text in the GUI	Y
4. Drainage Subsystem	1. The solenoid must have a response time less than 1 second when receiving a HIGH/Low signal	1. GUI reflects when the solenoid valve is open/closed. 2. Water will drain to the collection container when solenoid valve is open and vice-versa when closed	Y
5. Graphical User Interface Subsystem	1. Must display real-time pressure and flow data with at most a 10–30-second delay.	1. Text box shows almost instantaneous values measured by pressure and flow sensor along with solenoid status (open/close) 2. Log of data can be saved in csv file	Y

6. Power Subsystem	1. Must supply at least 25mA continuously at 5V ± 0.1V to support output components.	1. Board powers on 2. Sensors power on and accurately display results in Arduino IDE 3. Solenoid Valve can be open or closed by manual input	Y