



# Vacuum & Temperature Monitoring System (VTMS)

**Team Members:** Michaela Bellisle (ELE), Chris Alexander (ELE), Mariah Messinger (CPE)

**Technical Directors:** Richard Arthur, Peter Jones | **Consulting Technical Director:** Mike Smith (URI '01)



## PROJECT MOTIVATION

Seascan Inc. is a small company that manufactures oceanographic instrumentation and equipment to be used primarily in scientific research and development. They create a number of pressure tolerant enclosures to protect electronic equipment and battery packs when submerged in water. The housings are tested by using a vacuum pump, ball valve, and vacuum gauge and then monitored for 12 to 24 hours to ensure proper functionality.

This project is driven by the need to guarantee the enclosure is sealed every time it goes into the water once it is in the hands of the customer. The usage of vacuum, temperature, and humidity sensors will assure the user that the state inside of the enclosure is within the desired parameters. Additionally, a set of LEDs installed inside and/or outside of the housing would allow the user to easily check the status of the enclosure.

## KEY ACCOMPLISHMENTS

**Firmware Development:** Firmware has been developed using the Arduino IDE for the ESP32 board and peripherals. There are BLE functions responsible for transmitting and receiving packets of data to and from a BLE python client. The user is able to define the parameter settings with the GUI and the client will then transmit the data to the ESP32. The data packet contains the user defined parameter settings: device name, data collection frequency, desired sensor ranges and tolerances and the current date and time. If no data is received, the firmware will use the default parameter settings. Regardless of a BLE connection, the system will read the sensor data from the BME280, get the timestamp and store it on the microSD card. If there is a BLE device connected, the time stamp and sensor data is transmitted to the client. The system will also stay powered on as long as there is a BLE device connected. When there is no BLE client connected, the system will run through an LED sequence to indicate the status of the environment within the enclosure. If the sensor value is within the desired range, the green LED will be set high. If it is not, the firmware will either set the blue or red LED high for values that are lower or higher, respectively. After 5 cycles of sensor readings, the RTC alarm will be set and the system will be powered off.

**Schematic and PCB Fabrication:** Using a program called Circuitmaker, the selected components were placed and wired together on a schematic. Then a custom PCB was designed by placing and routing the components using the PCB design tool on Circuitmaker. An additional external circuit board with LEDs and a Reed Switch were also created for use with opaque pressure housings.

**Power Scheme:** The boards operate on a 5V battery and when not in use is in a "power efficient" mode. To turn on the board it activates through a variety of methods. The output of a load switch is triggered by at least one of three possibilities. The first is through an external RTC which provides a signal to the load switch at a user specified interval. The next is a reed switch which is triggered by a user swiping a magnet past the LED board or the main board. The final signal is from the ESP32 so that the board remains in the operating mode.

**Testing Prototype:** Both PCBs were populated, and initial tests were performed to ensure there were no shorts between signals. Afterwards, testing was conducted using the developed software and sample data to give varying results. There were some issues found during the testing but were resolved with some rework and firmware adjustments.

**Software:** Using PyCharm as an integrated development environment, the software for this system was written in Python, primarily using Tkinter, Bleak, and Asyncio. Tkinter was used to create a graphical user interface, Bleak to establish BLE communication with a peripheral device, and Asyncio to manage processes that needed to occur concurrently. The software can scan for nearby BLE devices, present them to the user so that they may choose one to connect to, then connect to the selected device. After connecting to the device the software will automatically read and store any data advertised. The graphical user interface also provides a way to view gathered data and enter desired parameters for the peripheral device.

## ANTICIPATED BEST OUTCOME

The ABO is a functional prototype that can monitor vacuum, temperature, and humidity levels and report the status to the user. Software will be created to display sensor data, remaining battery life, and accept user inputs. The system needs to be able to send and receive data between the main circuit and software via Bluetooth or WiFi connection. A schematic and PCB will also be fabricated for the hardware prototype. A system user manual describing the hardware and software components will accompany the prototype, along with test results and a demonstration of the working system.

## PROJECT OUTCOME

The Anticipated Best Outcome was achieved.

## FIGURES

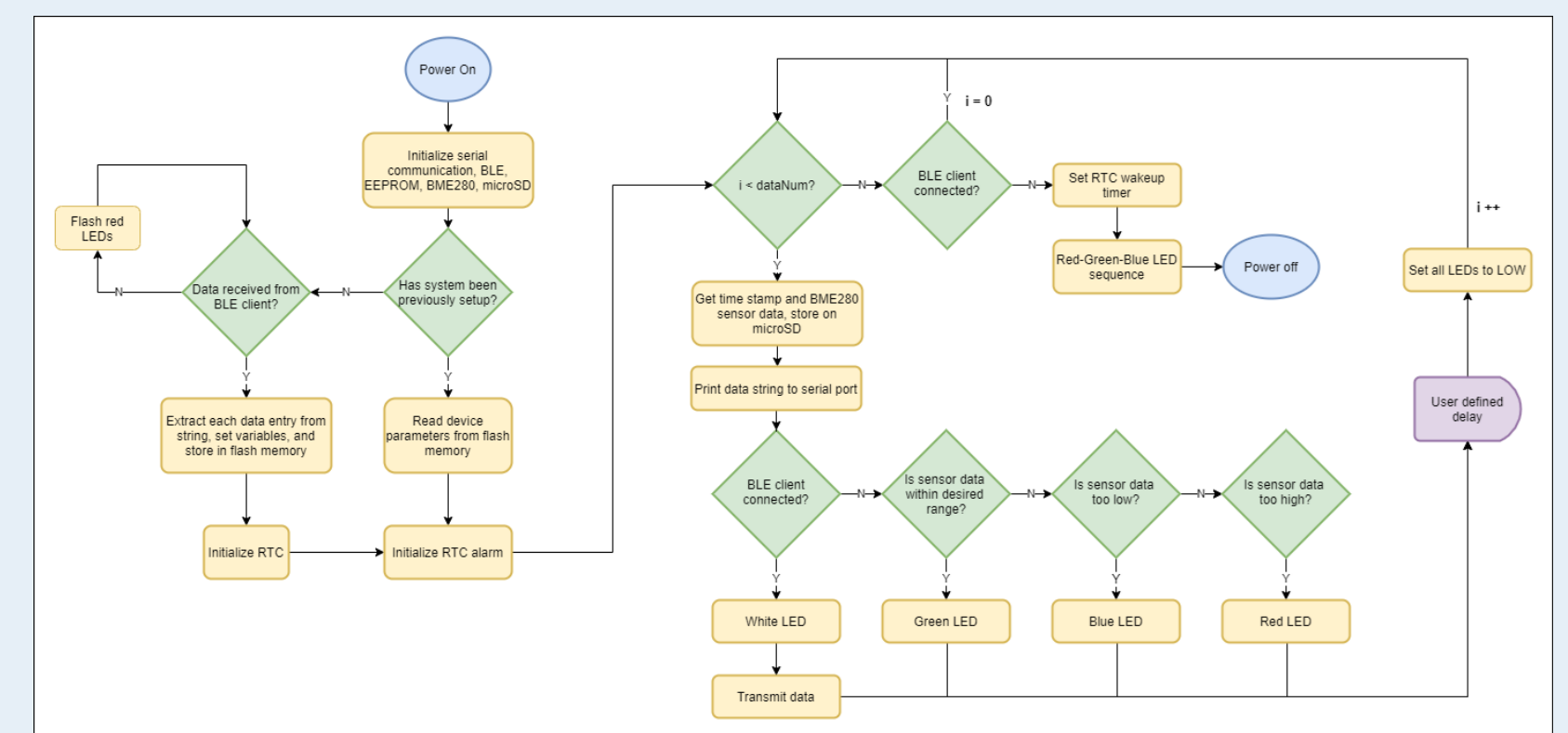


Fig. 1: Flowchart of embedded firmware

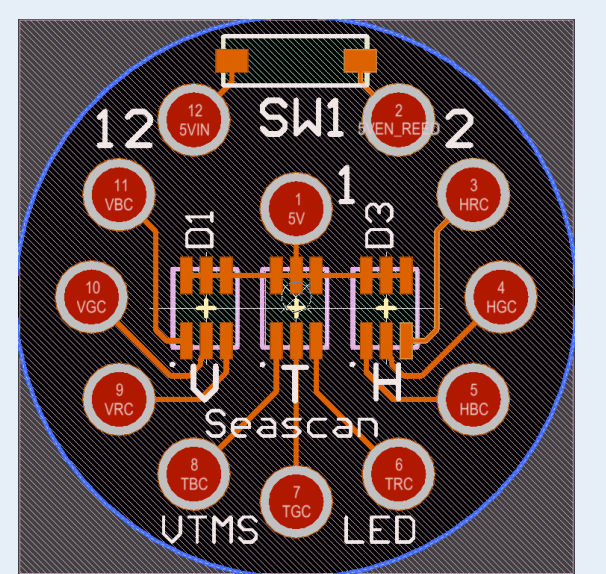
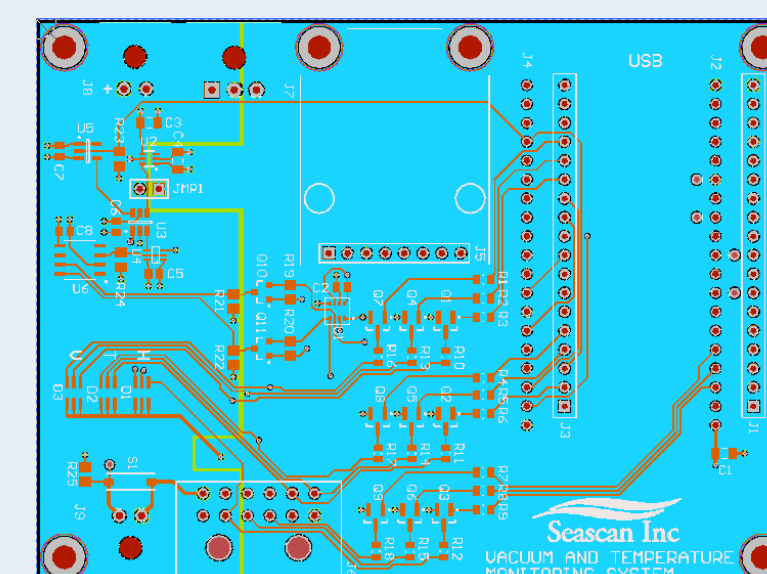


Fig. 2: The completed design of the Main PCB (left) and the external LED PCB (right) as seen on CircuitMaker

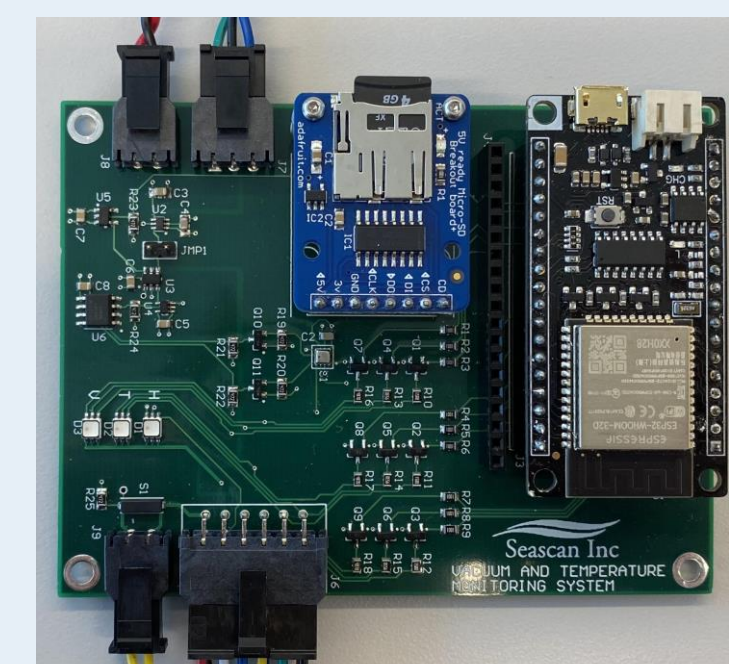


Fig. 3: The populated PCB of the main circuit (left) and the external LED circuit (right)

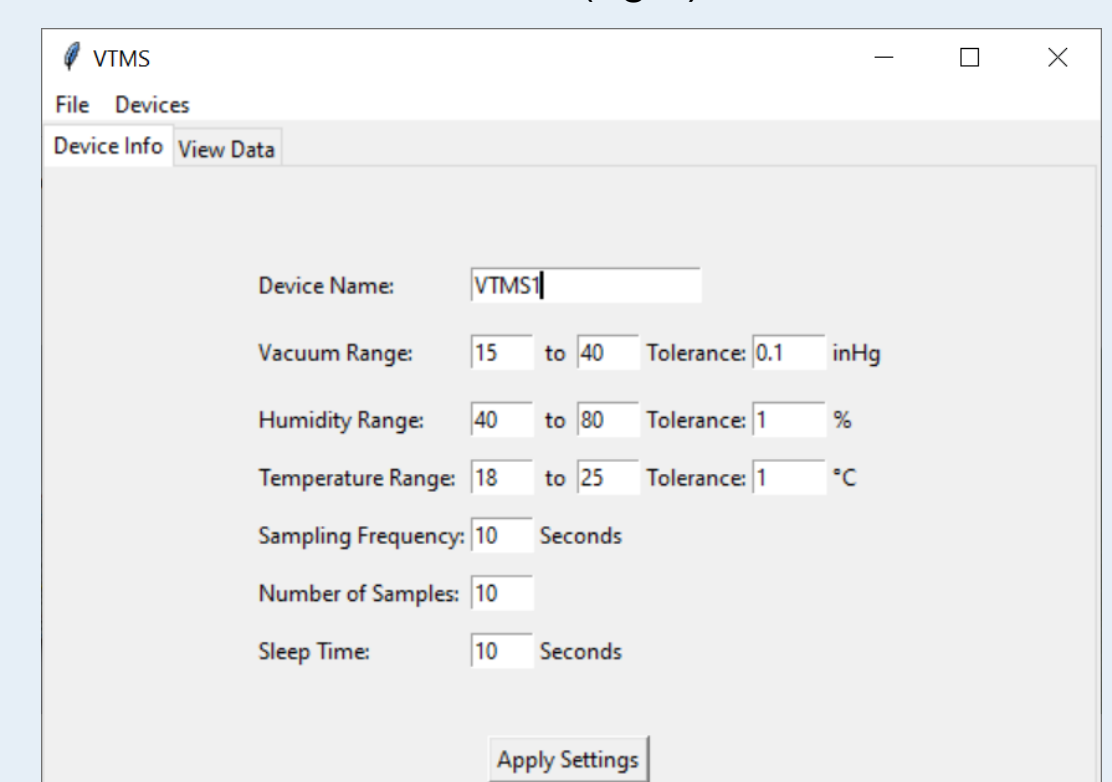


Fig. 4: The graphical user interface for the software. This allows the user to interact with a peripheral device in order to gather data.