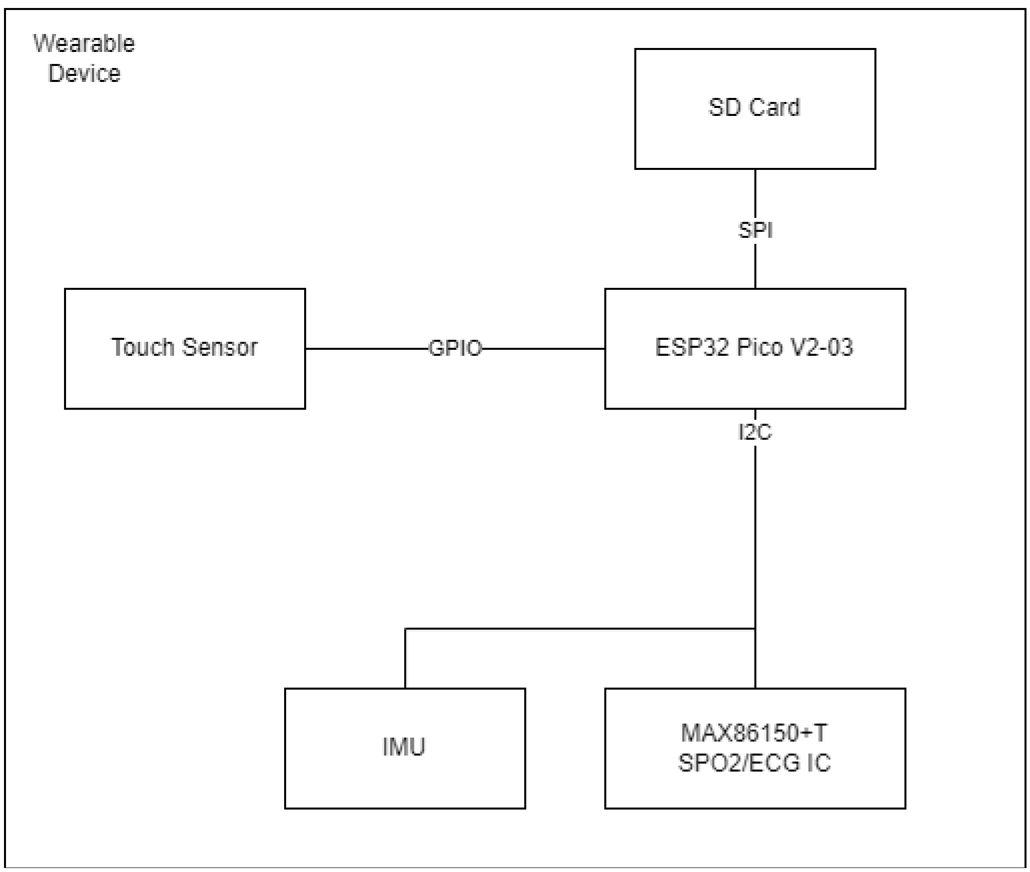
5 - Testing (Iteration 1)

In our first iteration of the project, we chose to focus on collecting more data than the standard on-the-market smartwatch and provide an easy interface for VetDogs to be able to collect the data and send it to our team for analysis and usage. This design consists of a wearable device with an IC that records red, IR, and green light reflections from the skin, as well as the state of a toggle button, then stores the data to an SD card. The VetDogs workers can then mount the SD card to their computer and send the data to our team, allowing for a quick review and collection of data for the iterative development of an algorithm for detecting PTSD episodes. A quick feedback loop allows for significantly faster iterations of the wearable device design and provides us quick benchmarks for device accuracy, usefulness, and, most importantly, a quick user-feedback loop.



5.1 Unit Testing

Wearable Device Components:

MAX SPO2/cardiac output sensor (ECG):

- Does the sensor output voltage match the requirements for the ESP Pico? Use a Voltmeter or oscilloscope to test.
- Does the sensor output (after converting/processing) match an actual SPO2 sensor?
- Does the sensor output match an actual heart rate detector?
- We can also use the available firmware and dev kit for the MAX device to test our configuration in real time.

Notable values and acceptable ranges to look for in testing:

- 40-60 bpm is consistent with bradycardia (lower end of about 0.666 Hz).
- 200 bpm is approximately the upper-end normal "maximum" for a 14-year-old. (upper end of 3.333 Hz).
- Peaks in the blood-flow volume graph should be between 0.666 Hz and 3.333 Hz, depending on the situation.
- SPO2 (if below 95% medical intervention is recommended, the normal amount is between 95%-100%) Acceptable range is between 95 and 100.
- Sensors output voltage within appropriate minx/maxes for the sensor. No current or voltage spikes, shorts, or otherwise harmful phenomena. (sensor component), test by writing output data to logs.

Should be tested in both low-stress (control) environments as well as higher-stress environments, with supervision.
Use a pulse oximeter (FDA-approved) to provide "proper" values at an interval and qualitatively compute calculations based on the sensor output.
Sensors should be tested on students before being sent to VetDogs for further testing.

Accelerometer:

- Check that output is within the acceptable range.
- Check values match a usable, compressible format for the Microcontroller.

ESP32 Pico Microcontroller:

- Assuming the MAX sensor provides valid values over I2C, the data is parsed and prepared to be written to a file.
- Assuming the button works properly, while the toggle is active, the data output marks the data as PTSD episode active. Otherwise, the data is marked as false (not active).
- Given proper sensor data input, the microcontroller encodes the data in Protobuf format with consistent schema (microcontroller data processing component)

SD card:

- Data persists after the microcontroller turns off or the card is removed

- Data/files are visible when attached to a computer.

Touch sensor toggle (button):

- On startup, the sensor output is low,
- When the sensor output is high, the output remains high.
- On being pressed again, the sensor output is low until the sensor is pressed again.

Wearable codebase:

- The codebase should have its functions thoroughly checked using a unit testing framework (like XUnit).

Algorithm:

- Our algorithm should detect PTSD episode symptoms with approximately 80% accuracy, and classify time intervals.
- The algorithm should consistently detect PTSD episodes correctly (AUC of the ROC curve of 0.65).
- The algorithm should have a sufficiently small false positive rate so that episodes can be predicted further in advance.

5.2 Interface Testing

User - MAX:

- Use provided firmware and similar devices to verify sensor data.
- Use an oscilloscope to record sensor output and compare with example data from other studies.

MAX over I2C to Pico:

- Outputs to file and verify the file data.
- Assert sensor readings are within the reasonable range.

Touch sensor - GPIO - Pico:

- Output to file and verify manually.

Pico - SPI SD card protocol - MicroSD card:

- Write file to SD card, plug SD card into computer, read manually.
- Write file to SD card, read file from SD card.

5.3 Integration Testing

MAX sensor output is written to SD card:

- Activate wearable device, attempt to read from SD card files, if file does not exist then there is an issue on this path.
- Check the data read from the file, if it contains continuous cardio/respiratory data, then pass, otherwise if missing data, or wrong/corrupt data then fail.

- Hardware can be simulated with CEEFIT or FIT frameworks in C, then we can test the integrations with software, without the hardware working (yet)

Touch Sensor → SD card:

- Activate wearable device, attempt to read from SD card files, if file does not exist then there is an issue on this path
- Check data is read from the file, if it contains continuous data for toggle then good, otherwise if corrupt/missing values, then the test fails.
- After running tests using CEEFIT or the prototype device, we can plot the data and verify validity by checking for outliers. The firmware for the MAX can be used to provide graphs of what the data should look like.

5.4 System Testing

- Writes/processing of data from the sensor's process in live-time (within 20 ms per sample). This would provide an upper-level benchmark for success in sampling.
- Test what battery life/voltage the device has a performance/quality drop-off. Use voltage transformers in Coover to modify input voltage until the device performance drops.
- Microprocessor resource usage does not have spikes. Performance is sufficient to maintain consistent, reliable data. We can investigate OSQuery, system calls (by OS), or drivers to log system performance and add warnings for issues.

5.5 Regression Testing

All future system enhancements will be external additions. As an illustration, our team has contemplated a mobile application to establish a connection with the wearable device; however, it has been temporarily set aside to prioritize the development of the wearable and collar devices. If the internals of our devices end up being upgraded or switched out for any reason, we refer to section **5.1** in regard to testing the new components.

Any modifications made to our wearable and collar devices will solely be made to make the system more functional and reliable to use. In order to test that these changes are working correctly, we will verify the output of the system through the system logs.

Automating testing using FIT frameworks can abstract the hardware design.

By using software/simulation-based testing frameworks, we can require tests to pass before approving any changes, insertions, or modifications.

5.6 Acceptance Testing

First Prototype:
The first prototype is to verify our hardware and give us a basis for iteration to meet final requirements. Our goal is to capture and save sensor data reliably in a simple version of the device.

Requirement: Capture and save sensor data to SD card

- All unit, integration, and system tests pass
- After unit and integration tests, send prototypes to America's VetDogs to test with veterans. We have already discussed the possibility of this with them.

While wearing the device, accurate vitals and sensor readings are read and stored onto an SD card. Veterans at VetDogs can be provided with the device, partnered with a monitoring person, and the data of their vitals and any episodes is stored on the SD card. Monitoring staff writes down start and end times of PTSD episodes for our team's later verification. The VetDogs staff then email the data from the SD card to our team. This provides direct user feedback for the wearable device size, design, and interface challenges in the form of comments.

Obtaining the data early provides us a set of data to divide into training(development) and testing data. One partition of the data is used for developing an algorithm that predicts and detects PTSD episodes, the other is used to verify success by applying the algorithm across it's values.

Final design requirements and tests, not applicable to first prototype:

Requirement: Reliably detect PTSD episodes.

- Input simulated PTSD episode data and ensure that device detects them.
- Have the device worn by someone prone to episodes and verify that it detects their episodes as they occur.

Requirement: Able to disable

- Simulate/trigger device and disable with button and/or app.

Requirement: Collar device is not disruptive

- Trigger the collar device and ensure that it is not noticeable by humans.

By obtaining quick feedback from veterans, we will be informed early on in the process of discomfort with the device, challenges with the wearable device buttons/interfaces, and obtain samples of their "normal" and "elevated" vitals.

5.7 Security Testing (if applicable)

Security testing is not needed for our wearable device or our collar device.

While the Phone Application is a background idea for the time being, it is important to plan ahead in case it becomes a priority in the future. Due to the nature of phone applications, security testing is important as to not allow any breaches in user data. As such, the tests we plan to put into place are as follows:

- Testing the encryption of data transmission via our communication protocols (Bluetooth).
- Ensuring that the firmware for the microcontroller and sensors are up to date.
- End-to-end encryption of vitals can be ensured via digital signatures. Man-in-the-middle attacks can test this.

5.8 Results

At this time, we do not have any results. Initial prototypes are in the works and are thus not able to produce results for the time being. In the future, we are hoping to send our wearable device prototype to America's VetDogs veterans so that they can collect data on PTSD episodes. This data can be used to create an algorithm for which to detect oncoming PTSD episodes.