

Product Requirements Document(PRD)

Authors: Jake Goodwin, Aidan Agee, Blake Babb, Patrick Iacob

DATE: 2023-11-23

Problem Description

Existing personal data acquisition devices are either too expensive or too DIY for most potential users. Many are designed for aerospace, automotive or research purposes, and are too expensive and unnecessarily complicated for casual users. The only other option is for users to build their own devices from prefabricated parts, which is too complicated and requires too much prerequisite knowledge for most potential users.

Scope

The scope of this project is to develop a prototype for a personal data acquisition device. This prototype will have the ability to collect real time data from a variety of sensors, including accelerometers, gyroscopes, GPS modules, and thermometers. These components will need to be combined in a printed circuit board for the final prototype. The scope of this project also includes development of a web-based UI that both presents the data gathered by the prototype and sends commands to the physical device to record data and configure sensors.

Use Cases

The user will take the product along with them on an outdoor activity and subject it to normal conditions for that activity.

The user will connect the product to a phone or laptop they brought with them, and view the data stream and take samples using the user interface, and save the data locally.

The user will choose and connect selected modules to the system using the CAN(controller area network) bus.

Purpose and Vision(Background)

Our purpose is to develop a personal data acquisition system that records all the data a user might want, and is cheap and easy to set up and use. It should be able to record data on acceleration, force, position, etc. require minimal setup, and can be hooked up to bike, go-kart, etc.

Stakeholders

Capstone Team

The capstone team are the main decision makers for the project, and will need extensive information for the product's requirements and implementation details. They will also need oversight from the project partner and TA.

Project Partner

The project partner will be working very closely with the capstone team, and will need to know the team's capabilities and status, and the status of the project.

Project TA

The TA needs to be informed on project progress and any issues the team may be having.

Capstone Instructors

The instructors require much of the same information as the TA, but because they are working less closely with the team there is less urgency.

Users

Users will need to know the product's capabilities, limitations and intended use.

Preliminary Context

Assumptions

- We have a suitable power supply of 12v to power the system.
- The end user has a device capable of connecting to an ad-hoc network.
- The data to be logged doesn't require more speed than the CAN 2.0 standard.
- The environment it's meant to be used in is electrically noisy.

Constraints

- As undergraduate students, our team has limited experience in the field, so we will have to learn a lot to deliver the product.
- Our budget is limited, so we will have to choose components carefully based on price.
- We are limited to three terms to deliver our product.

With these constraints factored, the biggest concerns for the feasibility of our project are the skills that need to be learned and limited time allotted to do so and complete the project. As an example, the team has primarily non-formal experience in hardware organization but has thus far worked efficiently in that aspect of the project. These risks are mitigated by the expertise and technical support offered by our project partner, and we consequently find the scope of our project realistic.

Dependencies

- The rust language, (reduces bugs and helps with memory safety.)
- C compiler(s), (C ABI is still used as a way to interface with libs.)
- Rust Embassy Library. (Embedded rust lib to reduce boilerplate)
- Rust Rocket(web server)
- STM SDK and HAL (Good references for the actual hardware.)
- The CAN standard.
- The Unix networking stack
- SQLite and or rust file I/O
- Rust Libraries available for individual sensor modules.

Some possible bottlenecks that could occur given our current dependencies would be centered around sensor modules not having an existing library written in rust. This would add more development time to the project.

However, we've researched workarounds and discovered tools to generate the needed interfaces for rust from a C header file.

Market Assessment and Competition Analysis

RexGen: Proprietary CAN bus based data logger, hard to find tutorial or documentation and is prohibitively expensive for hobbyists. Also unable to guarantee that their system is memory safe.

CANedge1: CANedge1: It has open source elements to it and documentation that is accessible, but still does not meet the requirements for its cost.

DEWESoft sells test and measurement equipment. Their products are not a good fit for our users because they are designed for industry, and therefore overkill and are prohibitively expensive for an individual.

Omega Engineering sells data loggers that can record the data our users would want, can connect to a remote device over Bluetooth and have easy to use interfaces. However most of their data loggers only record one or two types of data, so a user would need to buy many of them, which would be inconvenient and expensive.

An Apple Watch can track a user's activity data, and send it to an iPhone with an easy to read interface. However, the Apple Watch is limited in what kind of data it can record, and would not be appropriate for our users due to its many other unneeded functions.

There are guides on the internet that instruct a user on how to build their own data acquisition device using Arduino or Raspberry Pi microcontroller much more inexpensively than the other alternatives. However, this requires the user to have background knowledge in circuitry and programming, and requires a lot of time and effort to set up.

Target Demographics (User Persona)

Terry is an amateur Go-kart enthusiast who was brought into the hobby 8 months ago by friends and has become entrenched in the hobby since then. They are looking for a way to improve their performance but need more information about their current racing habits to do that.

Alice is a CTO of a large company that has decided to data log the forces and location their products experience during shipping through multiple contracted pilots and routes. She needs a system that isn't cost prohibitive to deploy in large numbers and can be customized for her company's other projects as needed.

John is an extreme snowboarder looking to collect data from his downhill tricks in order to help his friend create realistic and smooth animations for a snowboarding video game. He needs a data logging system that can endure cold environments and is modular so he can keep down the bulk/weight of the system while carving toeside and hitting some sweet jumps.

James is a competition mountain biker who wants to record and analyze data during rides for performance improvement. Uses a smartphone and needs an easy-to-use interface. He needs a system to compare data between runs.

Requirements

User Stories and Features (Functional Requirements)

User Story	Feature	Priority	GitHub Issue	Dependencies
As a mountain biker, I want to be able to view my instantaneous speed at any point in my journey.	Gps	Must Have	TBD	Common firmware.
As a motorsport hobbyist, I want to be able to record the g-forces I experience while going around tight corners.	Accelerometer.	Must Have	TBD	Common firmware.
As a winter sports enthusiast, I want to be able to track the turning speed of my snowboard.	Yaw rate sensor.	Must Have	TBD	Common firmware.

Non-Functional Requirements

- Delay on data transmission should be at an acceptable level
- Code should be well documented, following coding standards and best practices
- User interface should intuitive and fast to use

- The product should use security best practices whenever possible

Data requirements

- Analog data will be converted to digital.
- Sensor data must be reliable, resistant to EMI
- Sensor modules must adhere to the CAN protocol.

Integration requirements

- All interfaces will be rust doc documented.
- Tests will ensure API usage integrity.
- All modules that are to use the interface must pass integration tests.

User interaction and design

- Use a web server to interface with the user
- Can display live data, configure sensors, and download data
- Build using EGUI and Rust Rocket
- Should focus on ease of use for less experienced users
- Should be able to clearly provide all data and provide configurability



UI Mockup from project partner

User Documentation

The user documentation will be produced in markdown or LaTeX into a PDF or webpage. This documentation will cover the basic usage of the system and instructions on how to build it.

Testing and Quality Assurance

Testing will be done through TDD(test driven development) using the supported testing frameworks for rust and C. These tests will allow us as developers to ensure the assumptions we make about our code matches the actual behavior of it.

Quality assurance will mostly be handled by adherence to style standards enforced by the languages LSP(language server protocol) servers. The two that will see extensive use in this project being:

1. Rust-analyzer
2. clangd

Bug and issue tracking will all be handled by GitHub's Issue and project system. This also serves as a way to allow public contributions in the future to the code base.

Milestones and Timeline

Item	Description	Duration
Schematics	The wiring schematics	2 months
PCB	PCB gerber files	1 month
uC	Firmware for STM	
Sensor FW	Sensor module firmware	
UI	Web user interface	
Server	The back end web server	

Open questions

Currently, there are no open questions in the project.

Out of scope

- Support for more than the listed sensors.
- HAL development
- Radiation Hardening
- Full EMI sheilding
- Full support for 10 channels saturated with sensor data at 5kHz
- Water resistance at any depth or submersion
- Documentation beyond rustdocs/doxygen and markdown/latex.
- Wireless connectivity beyond ad-hoc wifi.