TEAM 6(Personal Data Aquisition)

Adian Agee
Blake Babb
Jake Goodwin
Patrick Iacob
Project Partner: Chris Patton

2024

Contents

1 Abstract **3**

2 Change Log **3**

3 Product Requirements Document **4**

3.1 Problem Description 4

3.1.1 Scope 4

3.1.2 Use Cases 4

3.2 Purpose and Vision(Background) 4

3.3 Stakeholders 4

3.4 Preliminary Context 5

3.4.1 Assumptions 5

3.4.2 Constraints 5

3.4.3 Dependencies 5

3.5 Market Assessment and Competition Analysis 6

3.6 Target Demographics (User Persona) 7

3.7 Requirements 7

4 Software Design Architecture **7**

5 Software Design Process **7**

1 Abstract

The aim of this senior capstone project is to design and implement a versatile data logging system that integrates single board computers (SBCs), STM32 microcontrollers, and the Rust programming language. The system will be capable of collecting, storing, and processing data from various sensors and input sources, providing a flexible and reliable solution for a wide range of applications.

The project will involve the development of software for both the SBCs and STM32 microcontrollers, leveraging the unique capabilities of each platform. The Rust programming language will be used for its safety, performance, and ease of use, allowing for efficient and reliable code development.

1. Selection and integration of sensors and input devices for data collection.
2. Design and implementation of communication protocols between SBCs and STM32 microcontrollers.
3. Development of data processing algorithms and storage mechanisms.
4. Creation of a user interface for system configuration and data visualization.

The project will culminate in the deployment and testing of the data logging system in real-world scenarios, demonstrating its effectiveness and reliability. The system will be designed with scalability and modularity in mind, allowing for future expansion and customization.

Overall, this project aims to provide a comprehensive solution for data logging applications, showcasing the capabilities of modern embedded systems and the Rust programming language in real-world applications.

2 Change Log

... okay

3 Product Requirements Document

Authors: Jake Goodwin, Aidan Agee, Blake Babb, Patrick Iacob
DATE: 2023-11-23

3.1 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 much prerequisite knowledge for most potential users.

3.1.1 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 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.

3.1.2 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.

3.2 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.

3.3 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 teams 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.

3.4 Preliminary Context

3.4.1 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.

3.4.2 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 alloted 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.

3.4.3 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 refences 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.

3.5 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.

3.6 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.

3.7 Requirements

4 Software Design Architecture

5 Software Design Process