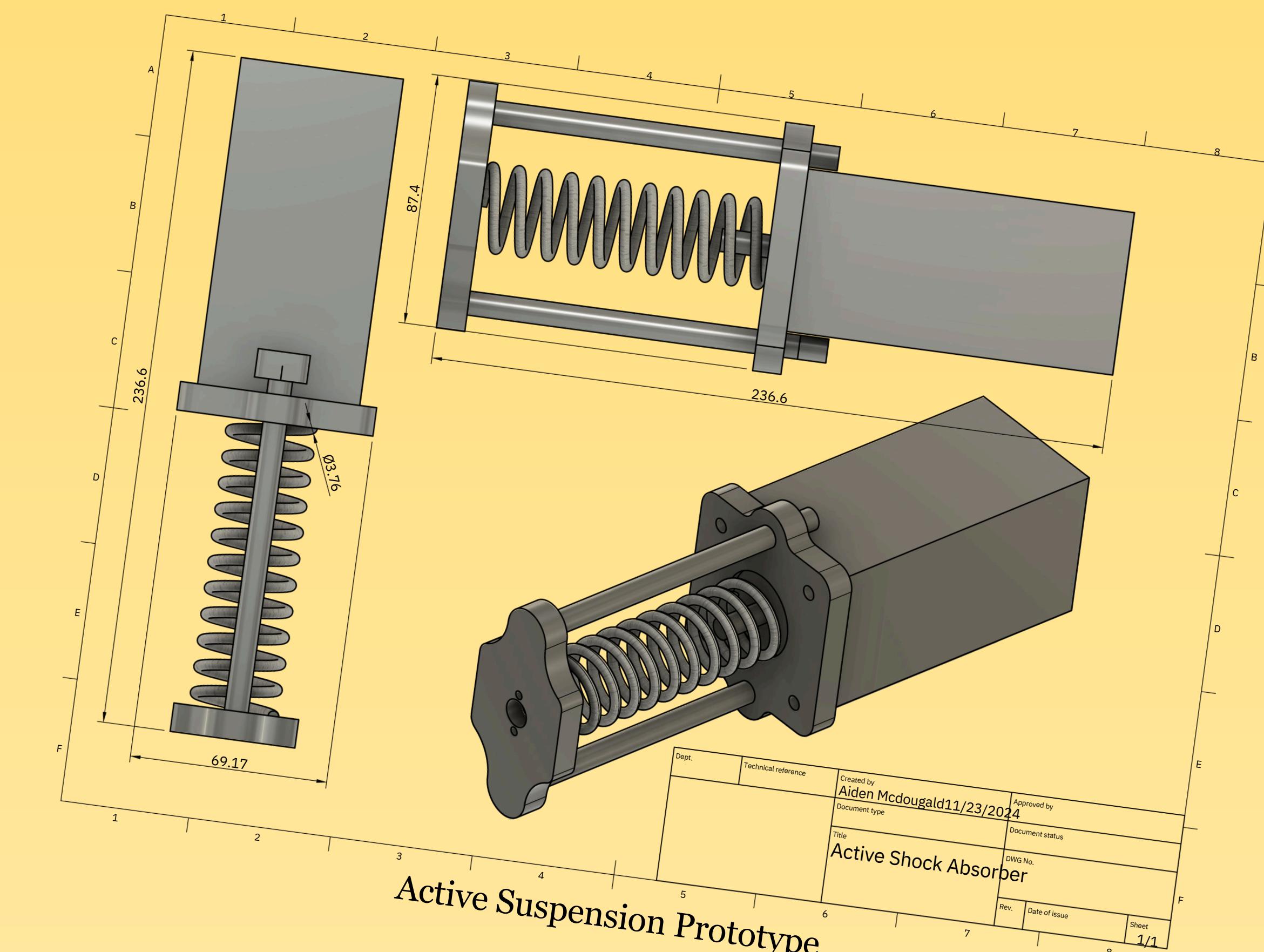
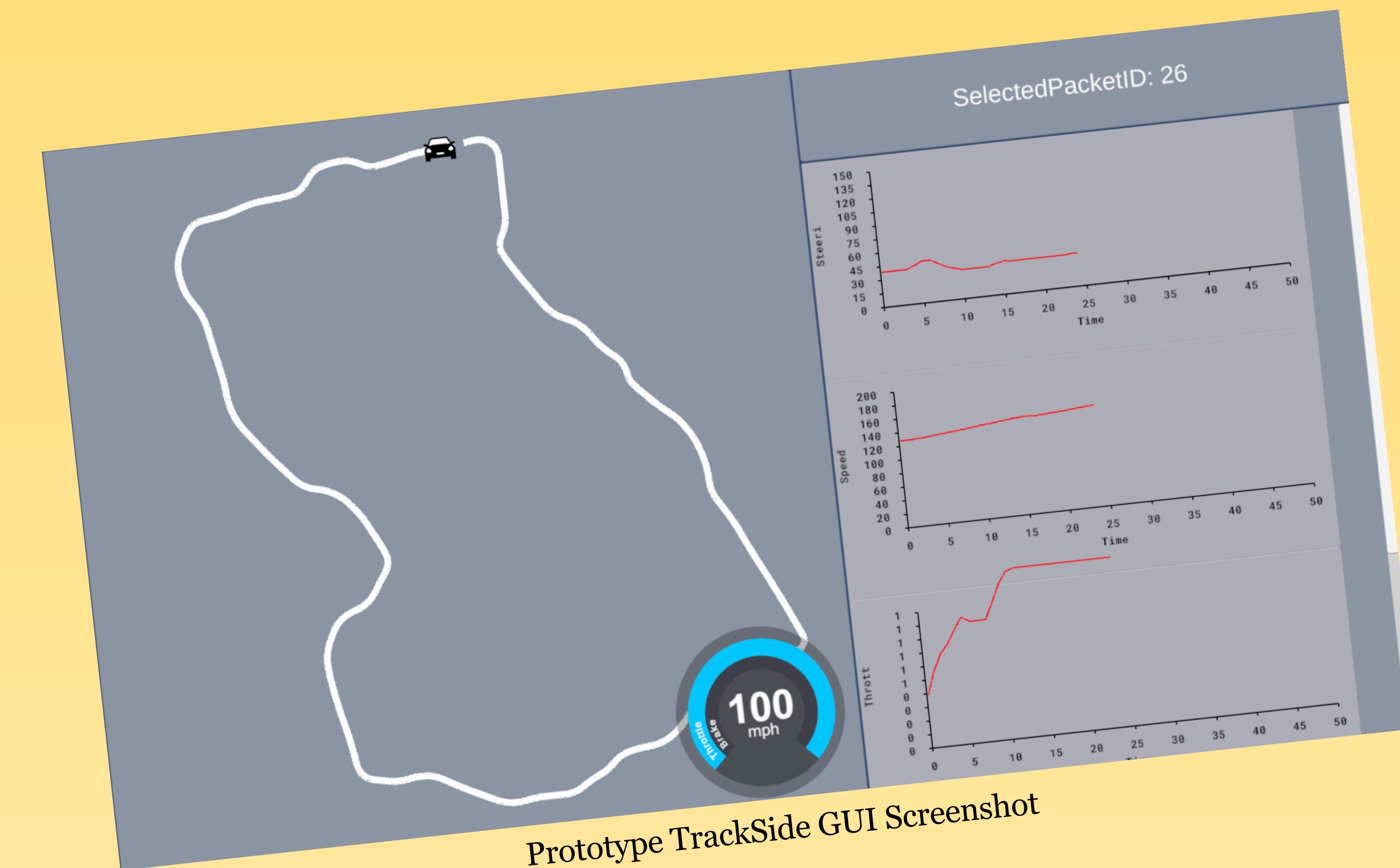


TrackSide and Race Car Software

Introduction

Formula SAE is a student competition focused on engineering design and innovation for a high-performance vehicle. The Telemetry App supports this project by providing real-time and historical vehicle data monitoring. The application is built with a Unity front end and a Golang backend to connect to the database in SQLite. This data is used to optimize performance and diagnose issues. For the car itself we have a Raspberry Pi-based Central Control Unit (CCU).



Objectives

- Provide real-time visualization of critical vehicle parameters during race and test sessions.
- Store telemetry data for post-session analysis and performance tuning.
- Ensure seamless integration with the Formula SAE vehicle systems, including sensors, the CCU, and other control units.
- Ecosystem of inter-compatible software running on different hardwares
- Each unit has its own unique software
- Research or develop a communication protocol that ensures safe and quick packet arrival
- Different platforms for different software i.e. Arduino and raspberry pi (Linux)
- Proper safety checks between different units

Design and Implementation

Tools & Technologies

- Programming Language: C# and Golang
- App: Unity Engine
- Database: SQLite
- Hardware: Raspberry Pi, CCU, various sensors

Data Flow

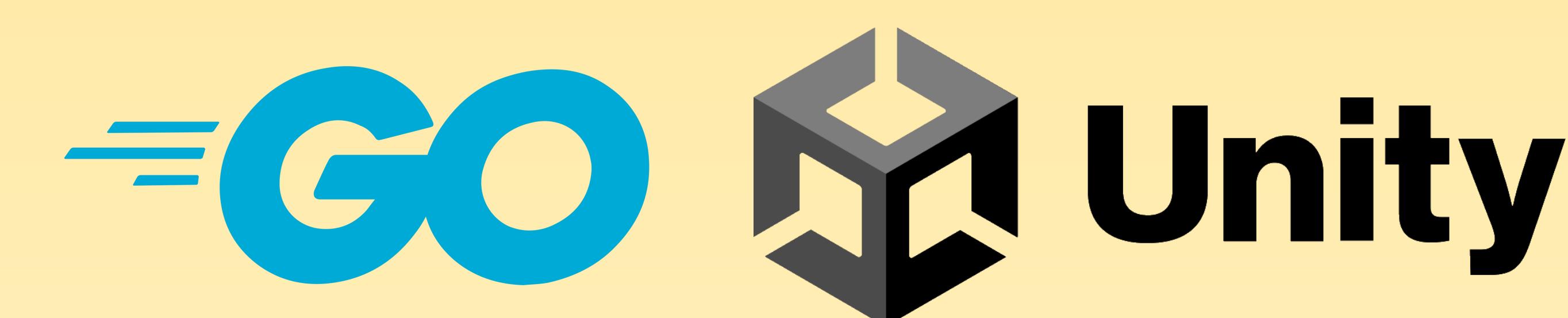
- Sensors collect analog data and send it to the CCU via I2C bus.
- The CCU processes the data and transmits it to the database via WiFi.
- The GUI retrieves data from the database for live or historical display.

System Architecture

- Hardware Interfaces
 - CCU (Central Control Unit):
 - Manages data collection from sensors and sends it to the database.
 - Sensors:
 - Collect analog signals for steering, throttle, brake positions, tire temperatures, and GPS location.
 - Raspberry Pi and Database Server:
 - Raspberry Pi processes sensor data; database server runs on a Windows laptop.
- Human Interfaces
 - Steering Wheel Interface:
 - Displays key metrics like speed, tire temperatures, and lap times on a steering wheel screen.
 - TrackSide:
 - GUI shows live diagnostics and allows for strategic decision-making during races.

Results (So Far)

- Functional Components:
 - Core database setup and communication.
 - Telemetry App for real-time data viewing.
- In Progress:
 - Complete car software
 - Advanced alert configurations and dashboard customization.



Future Work

- Integrate more advanced visualization tools for race strategy optimization.
- Expand functionality for driver coaching using historical performance trends.
- Ensure compatibility with additional vehicle subsystems (e.g., active suspension and traction control).

Unpaid Workers

