



Anemoment TriSonica Data Logging and Web GUI

Executive Summary

The Anemoment TriSonica Data Logging and Web GUI is architected using well supported open-source software tools for the Raspberry Pi. By utilizing these tools the software will be portable, upgradable, and reliable. To meet the current End-User requirements, Pseudo Design estimates to expend 24 to 26 hours on development for the final product.

End-User Requirements

Raspberry Pi boots to a web browser pointing to the Anemoment logging graph

Graph displays running values for 1-minute

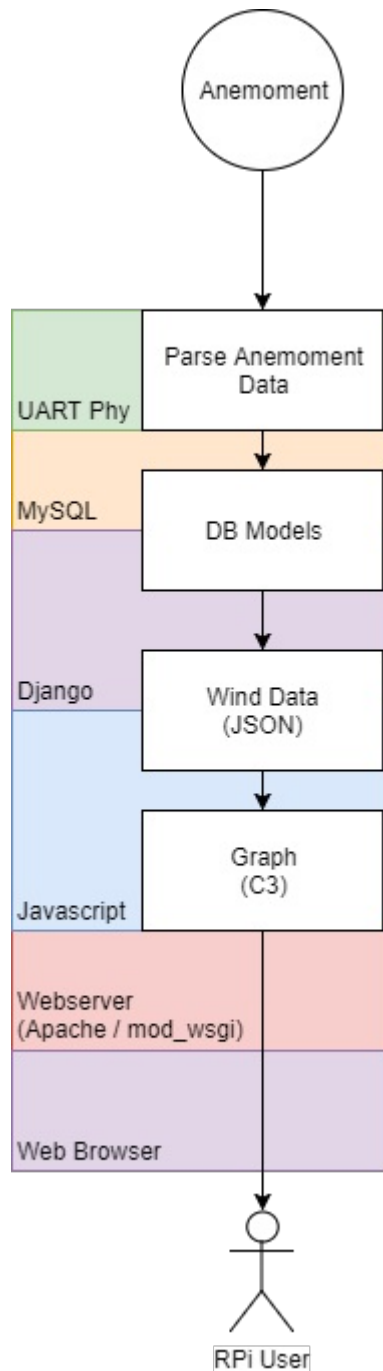
Graph targets a 10Hz refresh rate

Graph includes data for the following parameters

- Wind speed
- Wind Direction
- North-South Vector
- West-East Vector
- Up-Down Vector
- Temperature

Graph includes buttons for enabling-disabling each parameter

Software Architecture



The software architecture is built around open-source database and web frameworks. Data retrieved from the Anemoment TriSonica tool will be loaded into a database. The web framework will read data from that database, then serve web pages that generate interactive graphs from the data. This results in a highly portable and maintainable software project that can easily add new features in future versions.

Development Environment

The software will be developed using the [borg](#) project. Borg is a set of libraries that allow provisioning of virtual machines via simple YAML-formatted configuration files. This results in a development environment that is agnostic to the host OS, and the ability to deploy an RPi system image that is similarly provisioned from the configuration file.

The software will be written mostly in Python, with limited HTML and Javascript being required to generate web pages. Although the software project doesn't require a specific IDE for development, [PyCharm](#) is recommended.

Required COTS

UART Driver

The UART drivers are already built in to the Raspberry Pi. No additional work is needed to retrieve raw data from the Anemoment TriSonica.

Estimated Time to Implement: 0 Hours

MySQL

Borg already includes support for MySQL via [MariaDB](#). The setup scripts will port to the Raspberry Pi without issue.

Estimated Time to Implement: 0 Hours

Django

[Django](#) is a web development framework built in Python. While borg already includes support for Django, additional configuration is needed for deploying to a production web server.

Estimated Time to Implement: 2 Hours

Javascript

JSON data will be delivered to a [C3](#)-based graph and rendered on a web browser. Since modern web browsers support Javascript, no additional work is needed to bring up Javascript.

Estimated Time to Implement: 0 Hours

Apache Webserver With mod_wsgi

Data will be served to web browsers by an [Apache webserver](#). The Apache will use [mod_wsgi](#) to communicate with the Django framework.

Estimated Time to Implement: 4 Hours

Web Browser

The Raspberry Pi includes Firefox, which will render the web GUI without issue.

Estimated Time to Implement: 0 Hours

Custom Software

Custom software will be written using a [Test-driven development](#) process. The tests will be implemented using [Python unittest](#) and [Django testing guidelines](#).

Anemoment TriSonica Data Parser

Opens the UART port and reads available data output from the tool as defined in the [user manual](#).

Estimated Time to Implement: 3 Hours

Database Models

Create database models for the required data parameters along with a timestamp. Includes an API for functionality required by the web pages.

Estimated Time to Implement: 2 Hours

Wind Data Views

- May require [session tracking](#) to meet targeted refresh rate
- JSON data interface for N previous minutes of data
- Graph rendered on root page

Estimated Time to Implement: 4-6 Hours

Graph Webpage

- Generate a C3-based graph
- Include buttons for enabling/disabling parameters

Estimated Time to Implement: 3 Hours

Top-level app

- Runs on system startup
- Forwards TriSonica data to Database Models
- Verifies web browser is running

Estimated Time to Implement: 2 Hours

System Integration and Deployment

- Borg-side scripts for provisioning Raspberry Pi image
- Raspberry Pi-side script for configuring system

Estimated Time to Implement: 2 Hours

Final Documentation

Update this document to include high-level design decisions (password locations, install instructions)

Estimated Time to Implement: 2 Hours