ICE Reader

Internal Combustion Engine Reader

Fredrick Collins

v1.0

Contents

- ***Page 1*** *- Project Description*
- ***Page 2****- Features, Users, Technology*
- ***Page 3****- Scope*
- ***Page 5*** *- Flowchart*
- ***Page 6*** *- Interactions*
- ***Page 7*** *- Art Specs, Wireframes*
- ***Page 9*** *- Feedback*

Project Description

I am proposing a plug-and-play solution to fault code reading, real-time feedback, data logging, and analysis for the commercial vehicle via the OBD-II port through ICE (*Internal Combustion Engine)* Reader. Utilizing a Raspberry Pi Zero for computing, ICE Reader will operate by drawing power from any 12V plug and interfacing with the ECU (engine control unit) over micro-USB to query and return a range of useful values. After powering on the device, ICE Reader will draw to an on-board OLED display complete with a joystick and two buttons to load a start menu. From the menu, users can start/stop data logging, check for diagnostic trouble codes, or select a real-time readout of values such as speed, intake pressure, rpm, and more.

This solution doesn't stop when the car does, because if the user wishes to view and manipulate a variety of data series they have logged, they can connect to the Raspberry Pi's own wireless network and *visit the ICE Reader website* to chart, study, view reports, and compare recorded data from their database as they please.

Note: This project will be expanding upon work I had done in my senior year of high school. Previously, I had established a connection between the Raspberry Pi and the ECU, queried intake pressure, and displayed it to the screen in real time - the buck stopped there. All user inputs via menu, queried values, data logging, data manipulating, and charting features will be new. For reference: 40 lines of python code vs near 1,000 of python, client-side js, server-side js, css, and html.

Features

- Read and record common ECU fault codes
- Start and stop recording a wide array of engine variables in real time via user interface
    - Display a wide array of engine variables in real time via OLED screen
- Charting capability supporting multiple series on the same x-axis
    - Adjustable display scale, automatic high/low/mean/r^2 labeling
    - Ability to add moving averages to smooth data

Users

- Car enthusiasts
    - Measure timing advance, turbocharger pressure, etc.
- Regular consumers
    - Check for trouble codes, measure temperature, etc.
- Insurers or parents
    - Monitor speed, throttle usage for safety concerns, etc.
- Educators
    - Provide students with an environment to view real engine data relationships and draw conclusions

Technology

- ❏ Raspberry Pi Zero W
    - ❏ Microcomputer running Raspbian, a linux-based operating system
    - ❏ Will run its own web server and put up a connectable network with hostapd
- ❏ Python 3

    - ❏ Programming language, will be used to write the on-board software for the reader
    - ❏ Several libraries for utility and connectivity will be imported
- ❏ Javascript/HTML/CSS/NodeJS
    - ❏ Suite of web languages that will be used to interact with an SQLite database and provide a user interface and charting application
    - ❏ NodeJS backend serves requested files from the webserver and interprets them

Special Consideration: This device only works on cars made AFTER 1995!

Scope

/ice/python/

-menu.py

contains code to render and make functional main menu, where user can utilize joystick and two buttons on the hardware unit to select from any of the programs below or power down

-dtc.py

simplest program in the menu, press button to select, and the reader runs an error code query and displays the response. press button B to go back

-log_run.py

after running, displays just the status of the program. will write the full array of recorded values to the SQLite database rapidly. press B button to go back

-display.py

after running, 12 pages holding various options like rpm, speed, intake pressure, temp, etc. is shown. any of these can be scrolled through and the screen will show a live readout of that value, converted to norms and labeled. press B button to go back

/ice/

-index.html

website file, holds all client side HTML, CSS. this is the user interface

-script.js

client side javascript file holding functions the user calls by interacting with the webpage. sends GET requests to the nodeJS server

-chartjs.js/chartjs-plugin-zoom.js/hammerjs.js/hammer.min.js.map

javascript library chart.js, draws to HTML canvas, provides aesthetic and interactive charts

-index.js

nodejs backend, serves requested data from the SQLite database to the client facing webpage, has custom api to select required data

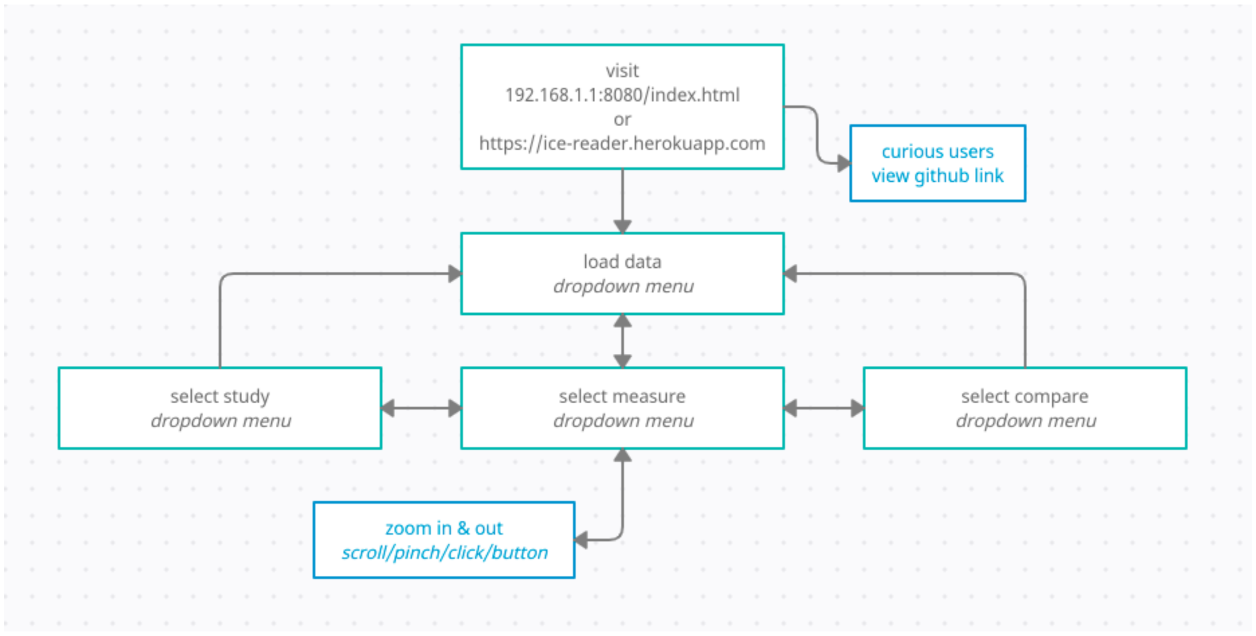
-package.json

json file contains instructions for nodeJS on how to meet dependencies and boot

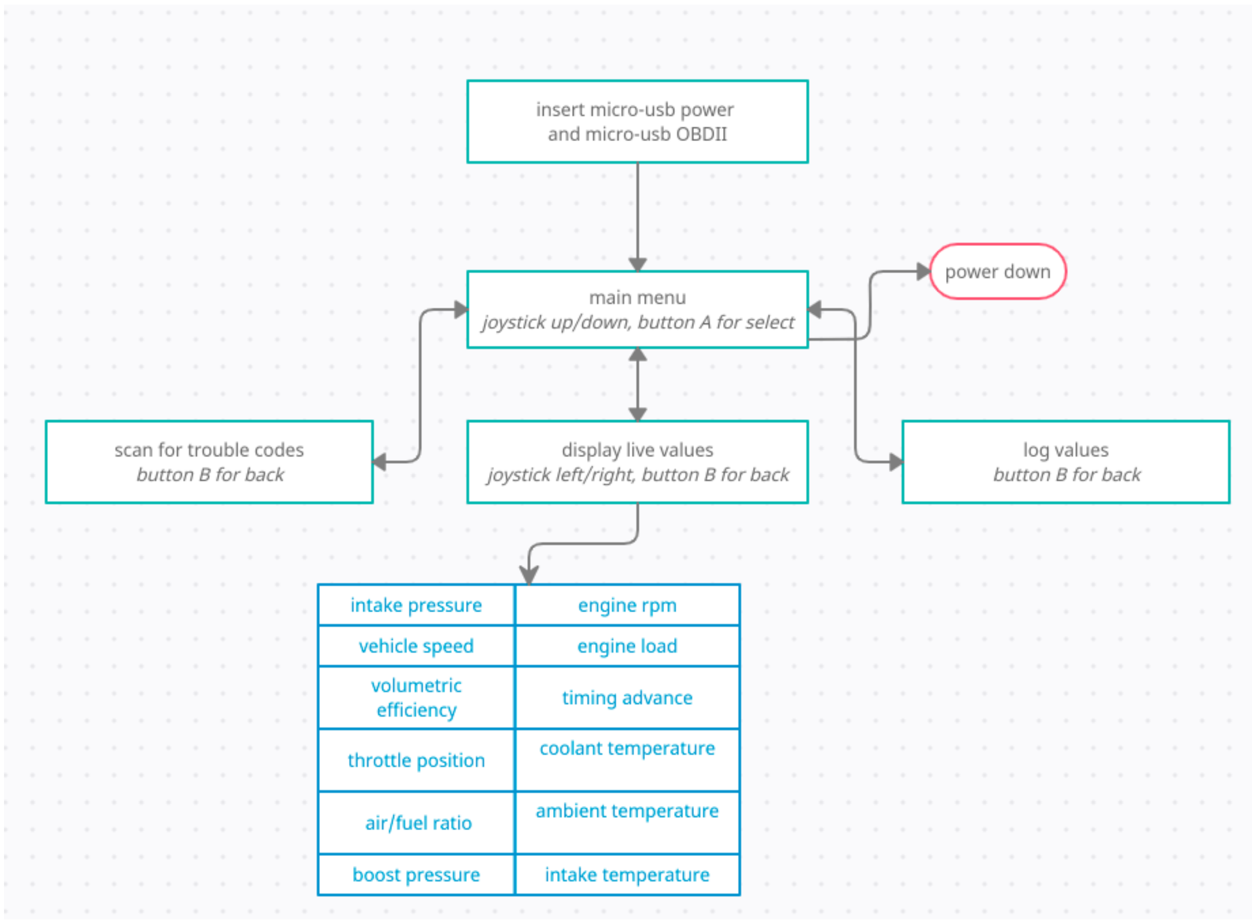
-ice.db

SQLite database stores data to be fetched and created via SELECT, INSERT statements

Site Flowchart



Raspberry Pi Flowchart



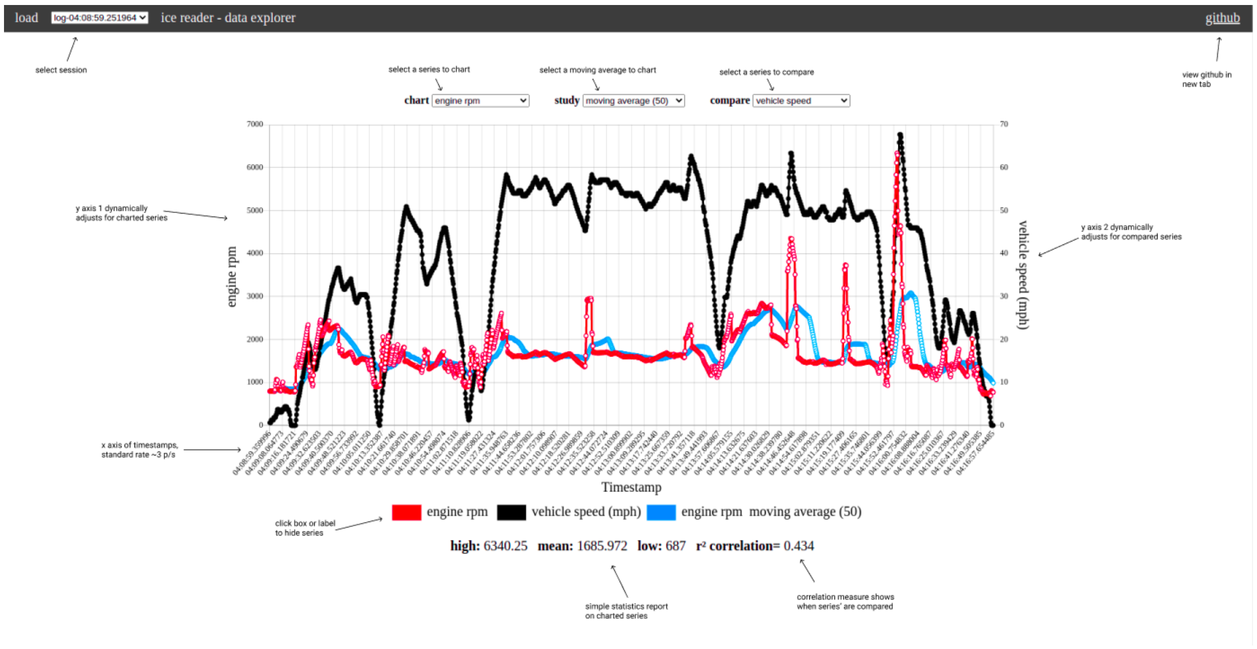
Interactions

- ❏ Load log via dropdown menu populated by server request
- ❏ Select measure, compare measure ping server to pull datasets and assign them to chart
- ❏ Study measure, calculate moving averages, correlation coefficient,
    - ❏ Always shown: high/low/mean report

Art Specs

- ❏ Font - Times New Roman, 12-24pt, bold as needed
- ❏ Colors
    - ❏ #FFFFFF background and navbar font
    - ❏ #3C3C3C top nav bar fill
    - ❏ #000000 dark font
    - ❏ Line 1: background color #FFFFFF, border color #FF0000
    - ❏ Line 2: background color #000000, border color #000000
    - ❏ Line 3: background color #ffffff, border color #009FFF

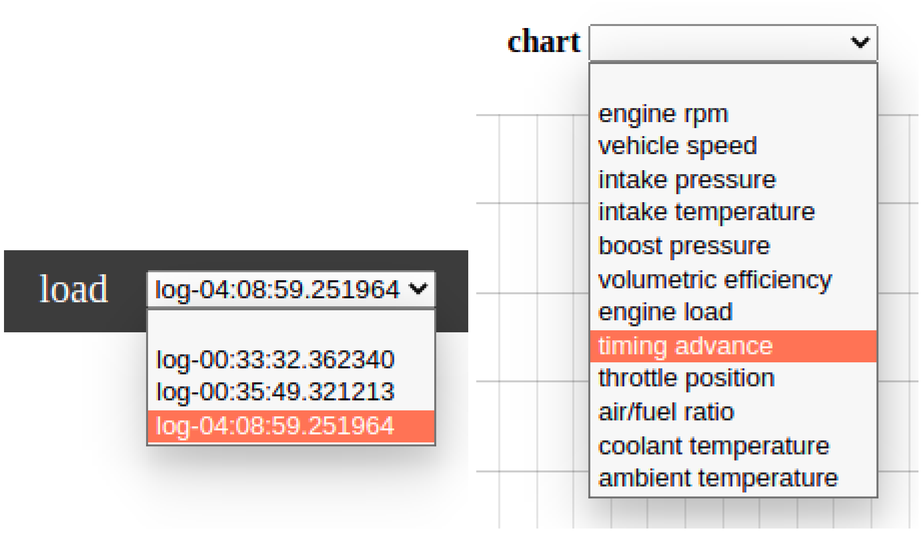
Wireframes

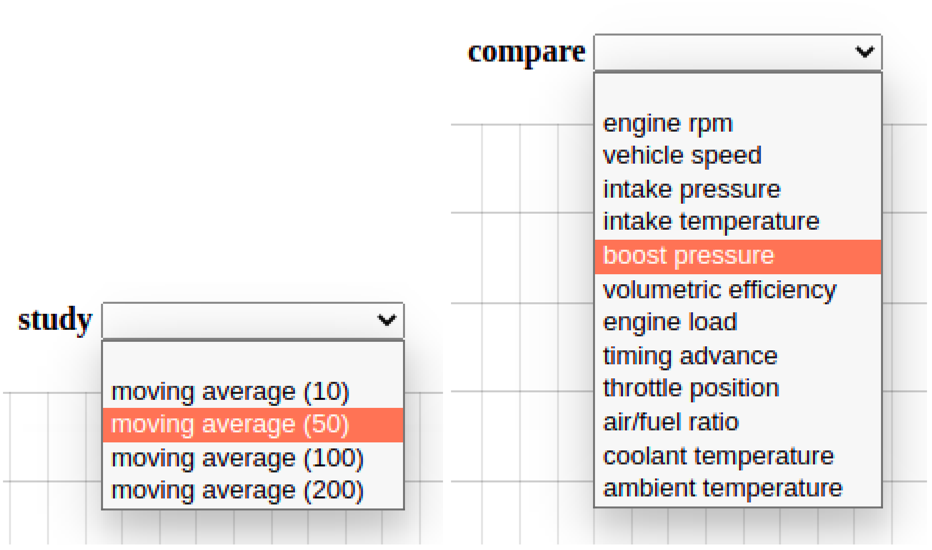

load
ice reader - data explorer
github
select session
select a series to chart
select a moving average to chart
select a series to compare
view github in new tab
y axis 1 dynamically adjusts for charted series
y axis 2 dynamically adjusts for compared series
engine rpm
vehicle speed (mph)
Timestamp
x axis of timestamps, standard rate ~3 p/s
click box or label to hide series
🟥 engine rpm
⬛ vehicle speed (mph)
🟦 engine rpm moving average (50)
simple statistics report on charted series
correlation measure shows when series' are compared

load log-04:08:59.251964

chart engine rpm **study** moving average (50) **compare** vehicle speed

high: 6340.25 **mean:** 1685.972 **low:** 687 **r² correlation=** 0.434





Timestamp

🟥 ~~air/fuel ratio~~ ⬛ vehicle speed (mph) 🟦 air/fuel ratio moving average (50)

high: 25.475 **mean:** 16.643 **low:** 11.554 **r² correlation=** 0.147

Timestamp

🟥 boost pressure (psi) ⬛ throttle position (%) 🟦 boost pressure (psi) moving average (200)

high: 17.793 **mean:** -5.652 **low:** -11.65 **r² correlation=** 0.826

Feedback (Suggested and Implemented)

- ❏ Use colored lines as opposed to grayscale
- ❏ Display report at all times
- ❏ Add zoom and pan feature
- ❏ Use SQLite to retain SQL-style capabilities in a more agile form
- ❏ Keep the color scheme simple and uniform
- ❏ Expand on detail in limitations and specifications documentation