Fredrick Collins

ICE Reader Technical Design

Software Structure Outline

The ICE Reader project's hardware component will run on a distribution of *Raspbian Lite*, which comes prepackaged with Python 2.7 and 3 - we will be using 3 for this project. Using the *Pip* package manager, the Pi will also be loaded with the necessary OBD and OLED I2C libraries to enable querying the ECU and displaying values to the OLED screen. The Raspberry Pi file setup will look something like this:

/home/pi/etc/rc.local

"sudo python3 /home/pi/ice/menu.py"

the rc.local file is run on each startup of the os

menu.py will boot the main menu of the program where programs can be run

/home/pi/ice/

-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 four of the programs below

-diagnostics.py

simplest program in the menu, press button to select, and a loading screen flashes quickly while the reader runs an error code query and displays the response. press any button to go back to menu

-display.py

after running, a submenu with choices like rpm, speed, intake pressure, temp, etc. is shown. any of these can be selected and from there the screen will clear to show a live readout of that value. press any button to go back

-log.py

after running, displays just the status of the program. will write the full array of recorded values to the sd card at intervals. press any button to go back

-web.py

runs log.py, but pushes data to locally hosted web server at intervals. user can view on their device by connecting to the pi's address

-export.py

searches for internet connection, pushes log data to mysql server. deletes after a successful response

Heroku will be used to host the web pages, and the setup should look something like this:

heroku/ice/

-explorer.php

the data exploration homepage. mocked up in prototype

-login.php

handles login authentication, doubles as landing page. has redirection to signup / reset password

-signup.php

handles new user signup, password verification.

-upload.php

handles .csv uploads, pushes to mysql server

-load.php

pulls from mysql server to load into data explorer

The Pi local web server option will run the same php files as above, but the handful of config and helper files I will likely need to use to get the web hosting functional are yet to be determined.

pi web server files

-mostly config files, to be determined

will run many of the same php files as the heroku server, just calling different functions to adapt to the circumstances that it will be run under (no internet connection, real time data loading)