

# University of Colorado Colorado Springs

CS 3030-001 Topics Computer Science

Python OpenXC (et al.)

Armand Moussaoui

Derek Cotton

7 March 2019

---

For this class project the team will be working with the Python-based OpenXC (with alteration options for use of pyOBD and Caring Caribou) to read, document, analyze, and display the trace for a given vehicle (Land Rover LR4 or Chevrolet Malibu Classic 2005). To accomplish these tasks, the team have to learn the OpenXC system, expand the library's development and document the experience. OpenXC module is an open source project sponsored by the automaker Ford Motor Company to expand development in automobiles.

The OpenXC is an open source hardware and software that provides an interface to vehicle data from the OpenXC platform which lets the user extend a vehicle capabilities by developing custom applications and pluggable modules. OpenXC™ is a data-focussed API for cars, once the hardware module is installed, the vehicle data becomes accessible to applications using the OpenXC library. Data such as steering wheel angle (-600-600 degrees), torque at transmission (-500-1500 Nm), engine speed (0-16382 RPM), vehicle speed (0-655 km/h) [remains positive even when moving backward, we can use gear status to compute direction], and accelerator pedal position (0-100%)...etc.

The first and only required phase of our project will be to read and format vehicle data. From there we will design visualization, recording, organization and exportation tools. Machine Learning may be utilized to create and understanding driver experience via a Generative Adversarial Network to objectivise driving experience, or with a Convolutional Neural Network to isolate driving habits or something else perhaps.

The second (optional) phase would be to develop a machine cracking tool to be able to decipher proprietary CAN messages. This is a great potential market, as nearly all auto electronics seem to lack quality, refinement and a marketable, comprehensive aftermarket. This would allow for a secondary market for automobile computing, aside from self-driving initiatives, that can interface more safely, efficiently and effectively with digital life.

Our first goal is design a json data parser; because the device output its data in json format. Our second goal is to design a dashboard application (python GUI) to display our data in an intuitive manner (with optional ML interjection). Finally (optionally), we will try some command injection to see if we can control some features of the vehicle from our dashboard application.

Our application should be able to read data from the OpenXC module display it on the dashboard and send general OBDII commands to vehicle to perform tasks such as turning off

the check engine lights, etc. Ideally we will have deciphered more complex commands and have a test case for ML informatics.

References:

<http://openxcplatform.com/>

<http://openxcplatform.com/about/data-set.html>

[https://creativecommons.org/licenses/by/4.0/deed.en\\_US](https://creativecommons.org/licenses/by/4.0/deed.en_US)

<http://www.obdtester.com/pyobd>

<https://github.com/CaringCaribou/caringcaribou/>