The radar units are Delphi ESR 9.21.15. The pdf file provides instructions for making the cables and for the values sent by the radar.

Get drivers from:

www.kvaser.com/product/kvaser-leaf-light-hs-v2/#/!

The file canlib.py provides the means to read the Kvaser input

Radar\_data\_parser.py is the radar-extracting code used by the previous senior design team. The entirety of their code is at github.com/rsyvarth/dsrc-radar-collision-avoidance if you want to look at it.

Test\_radar\_camera.py is the main code I wrote last semester for data collection. It saves (mostly) unprocessed radar data and (mostly) processed radar data to separate text files, and simultaneously saves video from a camera.

* The RadarParser process is a shortened version of the radar\_data\_parser code, and only grabs a few relevant fields from the radar
* I tried to put all the I/O into classes that could make use of Python’s with statement. This is a handy way of ensuring that each device/file will be shut down after any kind of error, including an error in the shutdown of another device.

radar\_GFM.py is a helper function that does the processing for the radar. Points can be noisy in space (multiple points for an object, inaccuracy) or time (flickering). My approach so far uses only averaging and cutoffs (there is a list of constants that can be altered). There are definitely more advanced techniques for multi-target tracking, if you want to look into that.