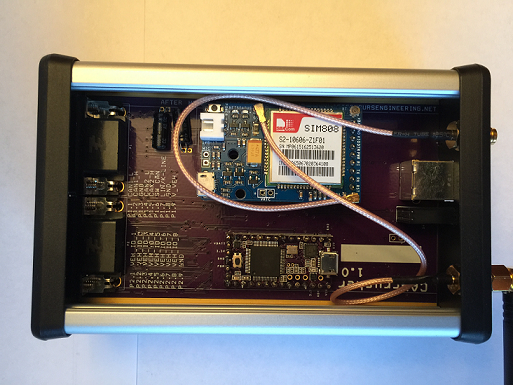
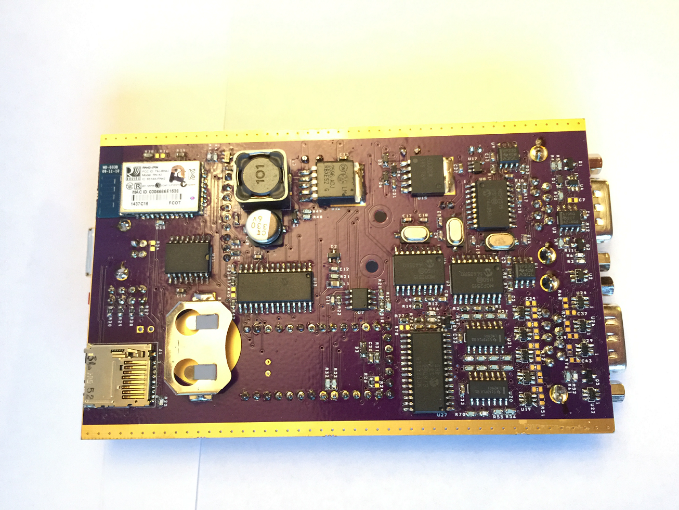
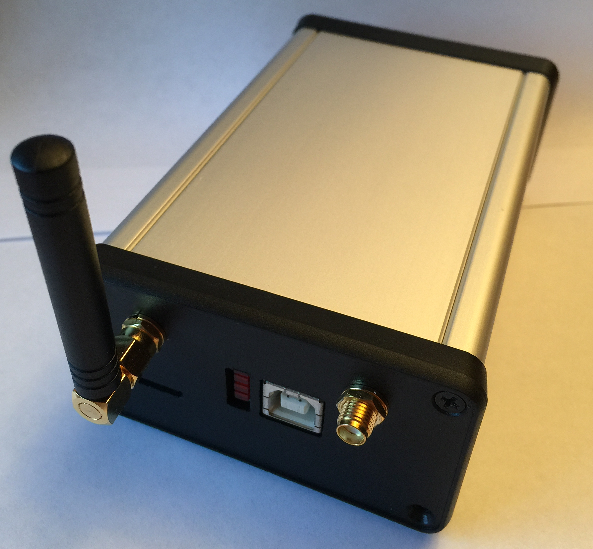
**CANcrusher Car Hack / Development Platform**





**Hardware/System Features:**

* 3 independent CAN channels supporting DW-CAN, SW-CAN, and LSFT CAN based on the MCP2515 CAN controller.
* 1 LIN/KLINE channel
* Bluetooth radio (RN42) with built-in SPP (acts like a serial port)
* SIM808 GSM / GPS Radio
* USB
* SD Datalogging using Bill Greiman's SDfat library for up to 64GB cards
* Real time clock using the DS3231 high precision RTC
* Teensy 3.1 (ARM Cortex M4) running @ 96MHz
* 8 Multi-purpose Inputs and Outputs for directly interfacing with the vehicle (outputs are open collector negative outputs with optional pull-up resistors.)
* LED Indicators (4 LEDs)

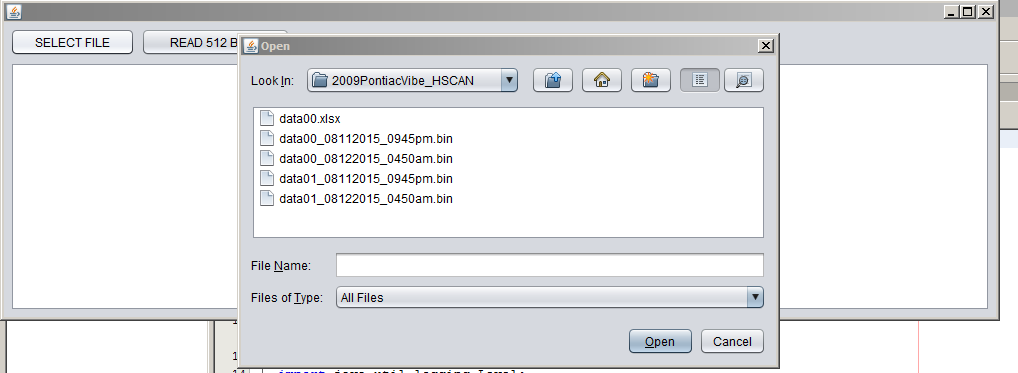
**SOFTWARE:**

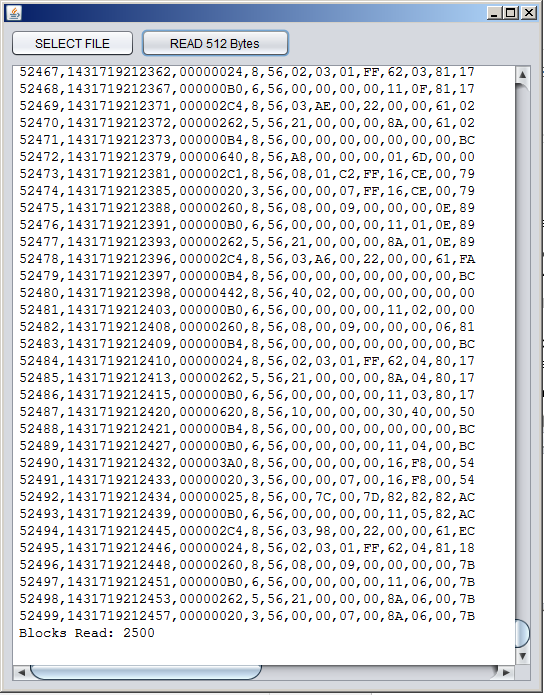
The firmware is Open Source (GNU General Public License), Arduino compatible and uses many of the libraries people know and love/hate.

On the computer-side, the initial software will be developed in Java with some of the following goals (some early versions of these features is done already):

* USB or Bluetooth communication with the CANcrusher
* CAN bus monitor allowing sending and receiving data, processing, reverse engineering, etc...
* Accepts CAN databases (\*.dbc) format
* Imports and exports log files in Vector and Intrepid Controls Systems format

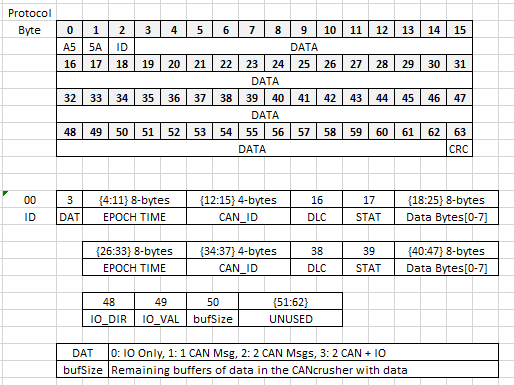
**Early version of the binary file exporting tool:**



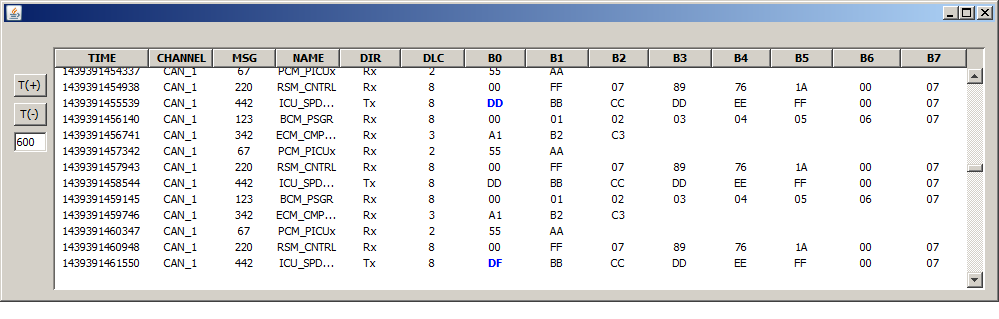


**COM Protocol for Bluetooth and USB interface:**

The protocol is still in development, but here is an example of the data packets that will be exchanged between the CANcrusher and the computer when not in logging mode.



**Early example of the CAN messages being sent to a Java GUI:**

****