



POLYCAN

HACKING TRACTORS FOR REPAIR

Modern vehicles contain many electronic devices which communicate via the CAN bus. John Deere tractors produce fault codes on the CAN network when problems occur, but diagnostic software costs thousands of dollars. PolyCAN is the open source CAN bus reader and analyzer for heavy duty vehicles.

PROJECT GOALS

- Create an easy to use tool to read from the CAN bus
- Log data from a tractor in real time
- Visualize CAN messages in a clear and intuitive format
- Create a suite of log analysis tools
- Develop methods to find unknown commands
- Populate database with known J1939 standard codes
- Collect high volume of log data
- Replay recorded packages/logs to tractor
- Build a tractor simulation test bench



iFixit

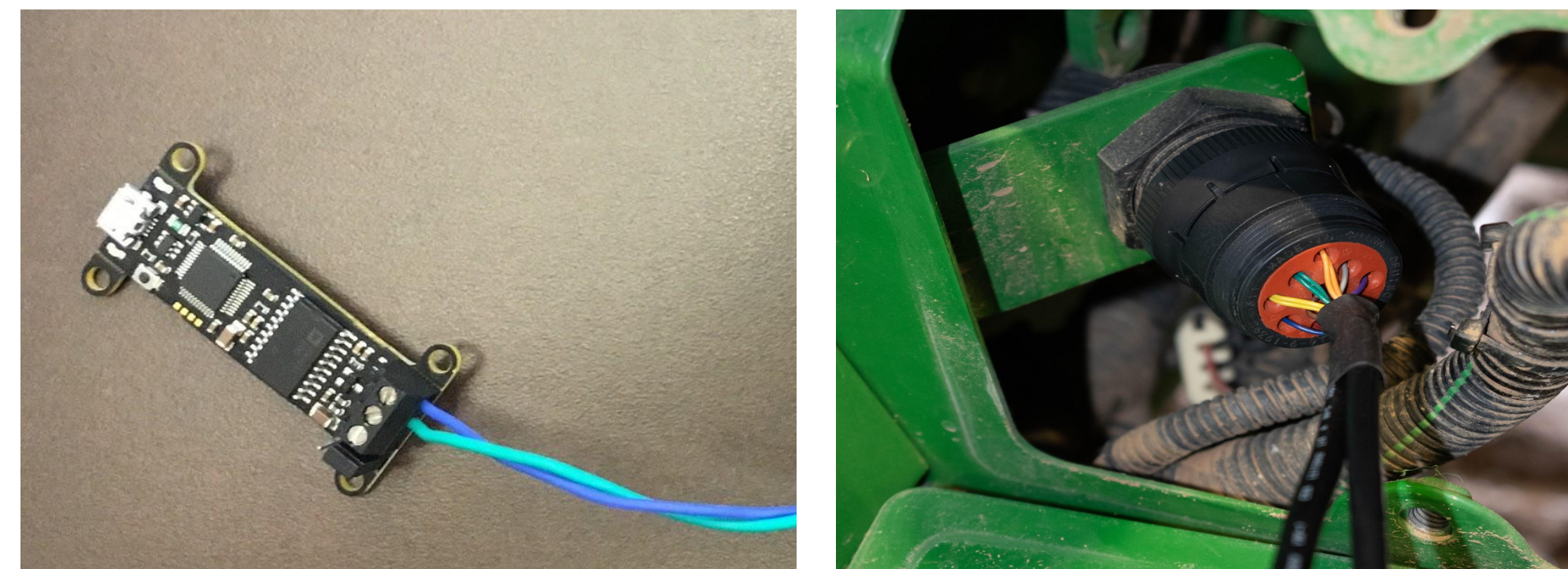
“The free repair guide for everything, written by everyone”

- Free repair guides for electronic devices
- 50,000 manuals, 144,000 solutions for 13,000 devices
- Sells replacement parts and tools
- Located in San Luis Obispo

HARDWARE

Readout configuration:

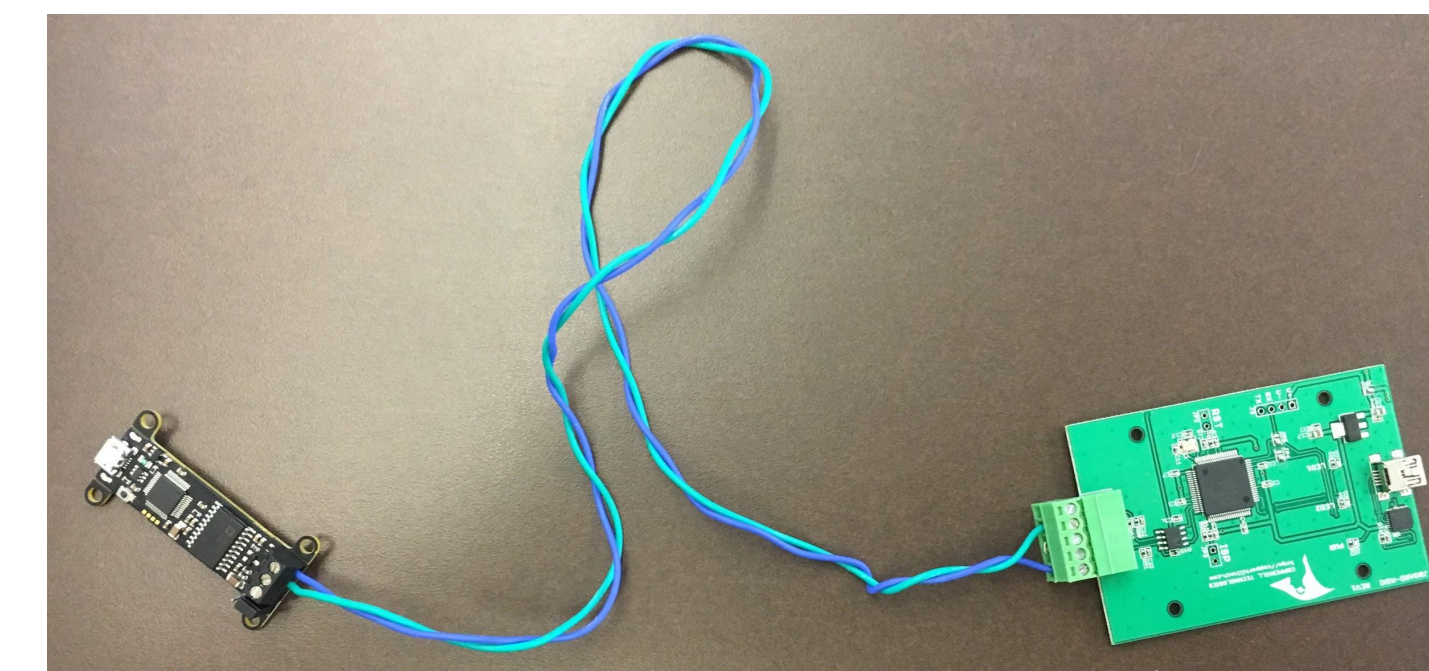
- ~Board: CANable ~750 commands/second
- 12 pin harness plugs into the tractor CAN bus
- Harness connects to CANable using 2 CAN wires
- Connects via USB to a linux machine running PolyCAN



Simulation configuration:



- Board 1: CANable ~750 commands/second
- Board 2: ECU Simulator Board ~300 commands/second
- Board 1 connected to linux machine running PolyCAN
- Board 2 connected to windows machine running tractor simulator
- Boards connected to each other via CAN wires
- Enable bidirectional data communication between the boards

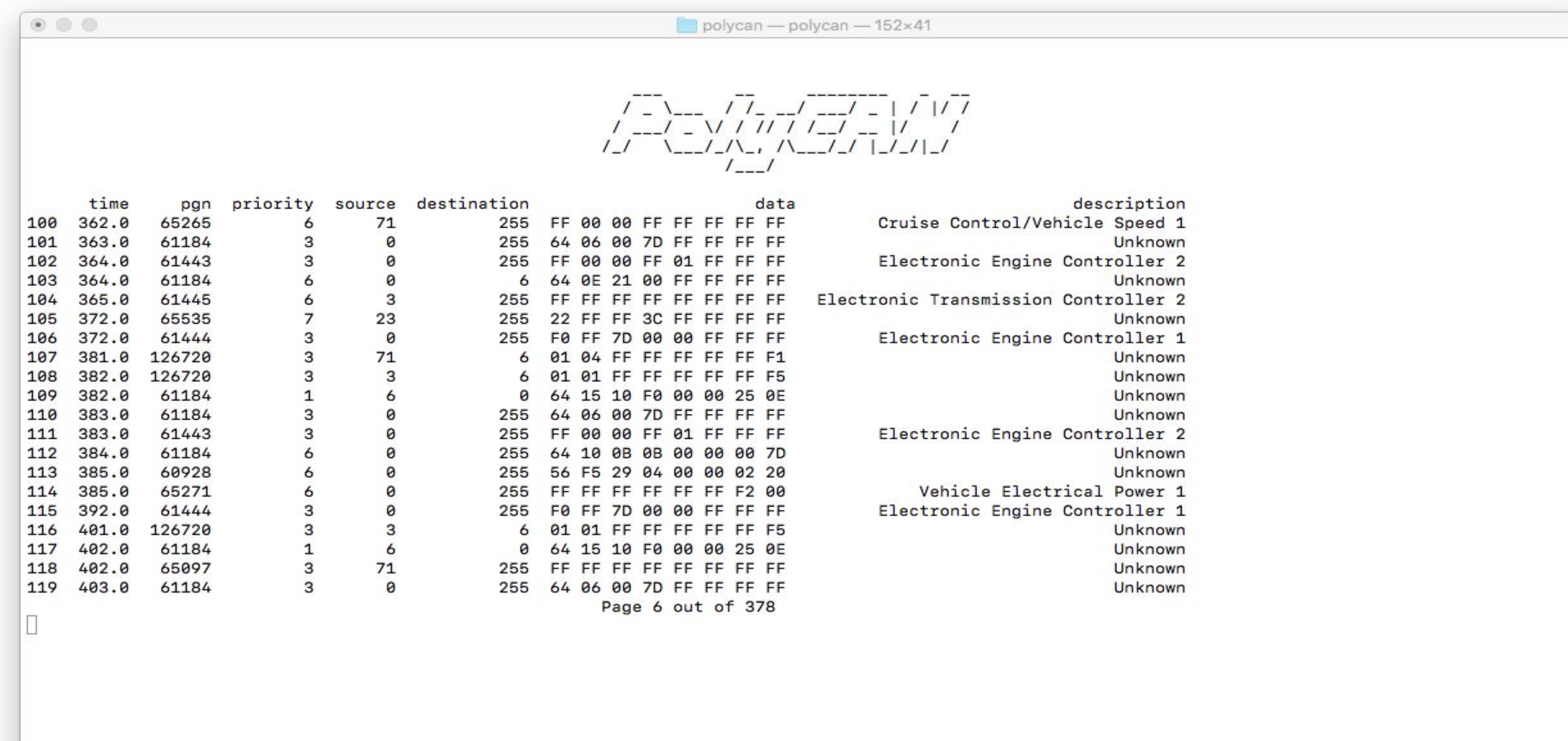


SOFTWARE

I/O - SocketCAN Linux utility allows reading CAN data via USB COM port

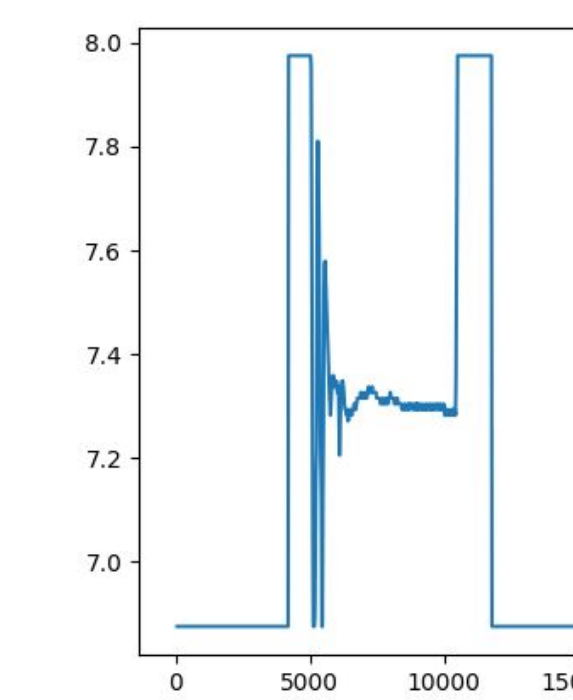
STORAGE - logs are stored in .csv format on local machine or in MySQL database

ANALYSIS - tools for filtering, sorting, pattern matching and detailed view built in Python using Pandas, NumPy, SciKit Learn



FEATURES

- Receive and store CAN bus messages from tractor
- Send commands to tractor while reading CAN bus
- Store received logs to file system
- Backup logs with database
- Online/Offline mode
- User login with access to DB and guest mode
- Logs:
 - Display with known commands stored in a DB
 - Filter by different options
 - Sort by different options
 - Statistical analysis
 - Find packet transmission patterns
 - Plot PGNs
 - Compare
 - Manipulate (for sending back to CAN bus and get different results)



(Figure shows plotted PGN 61443 (electronic engine controller 2) from recorded log while tractor startup)

RESULTS

- Stable receipt/transmission of CAN data
- Ability to effect dashboard output as expected by sending manipulated logs to CAN bus
- Didn't find as many unknown commands as planned (also due to tractor limitations, not able to drive)
- User interface, designed to be used by anyone
- Methods to find out new commands
- Testbench to simulate CAN bus for further groups, speeding up development time

Morgan Swanson, Sam Malicoat, Maksim Bondaruk, Colin Weber

Instructor/Advisor: Foaad Khosmood
Client: Chris Cline, iFixit

Capstone Class CPE450
Fall and Winter 2018/19