RECON 2013

# Hot-Wiring of the Future: Exploring Automotive CANs

Chris Hoder
Ted Sumers
(Grayson Zulauf)

SIEGE technologies

THAYER SCHOOL OF ENGINEERING AT DARTMOUTH

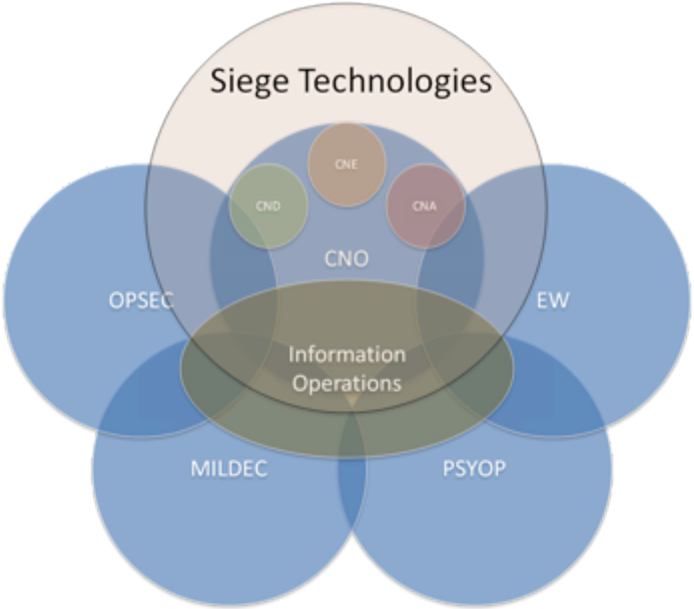
RECON 2013

# Hot-Wiring of the Future: An Introduction

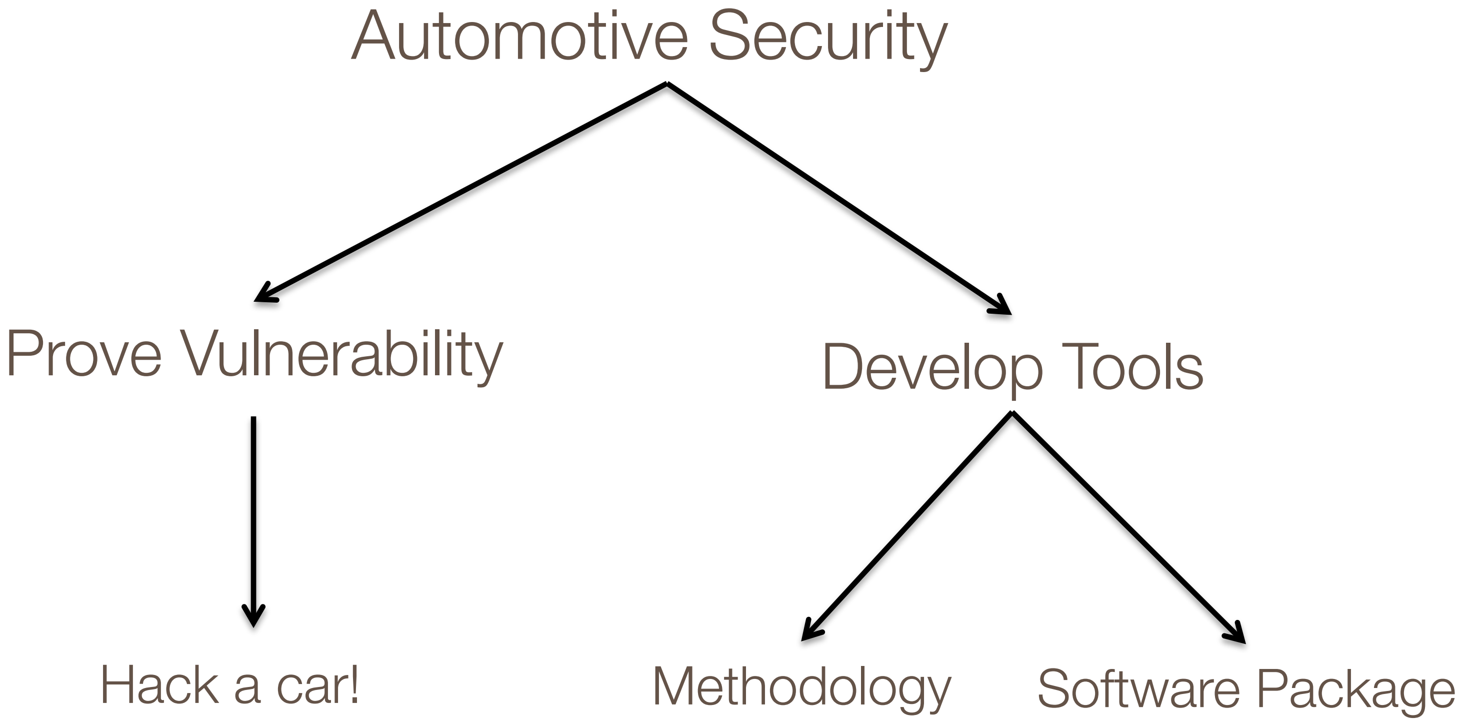
[www.youtube.com/watch?v=p3-fjJZhACg](www.youtube.com/watch?v=p3-fjJZhACg)

# The Project

- Thayer School of Engineering at Dartmouth College
  - Capstone project with industry sponsor
- Siege Technologies
  - Private firm, founded in 2009
  - Cyber Security / Cyber Warfare R & D
  - Provides world-class technical solutions to US DoD and IC
  - Sponsor of Car CAN bus project



SIEGE technologies



Automotive Security

Prove Vulnerability

Develop Tools

Hack a car!

Methodology

Software Package

# The Dream Team

## Ted

- Embedded Systems
- Computer Architecture

## Chris

- Software Engineer
- Python

## Grayson

- “I basically wanna build Stuxnet”

- Ted’s working at A AUTOMATIC then going to Penn Engineering

- Chris is headed to Microsoft

- Grayson’s working at MOTIV

# The Goals

- Reverse engineer a vehicle
- Build tools for muggles
- Explain it all to a bunch of mechanical engineers

# WHAT IS A CAR?

- Four wheels and an engine

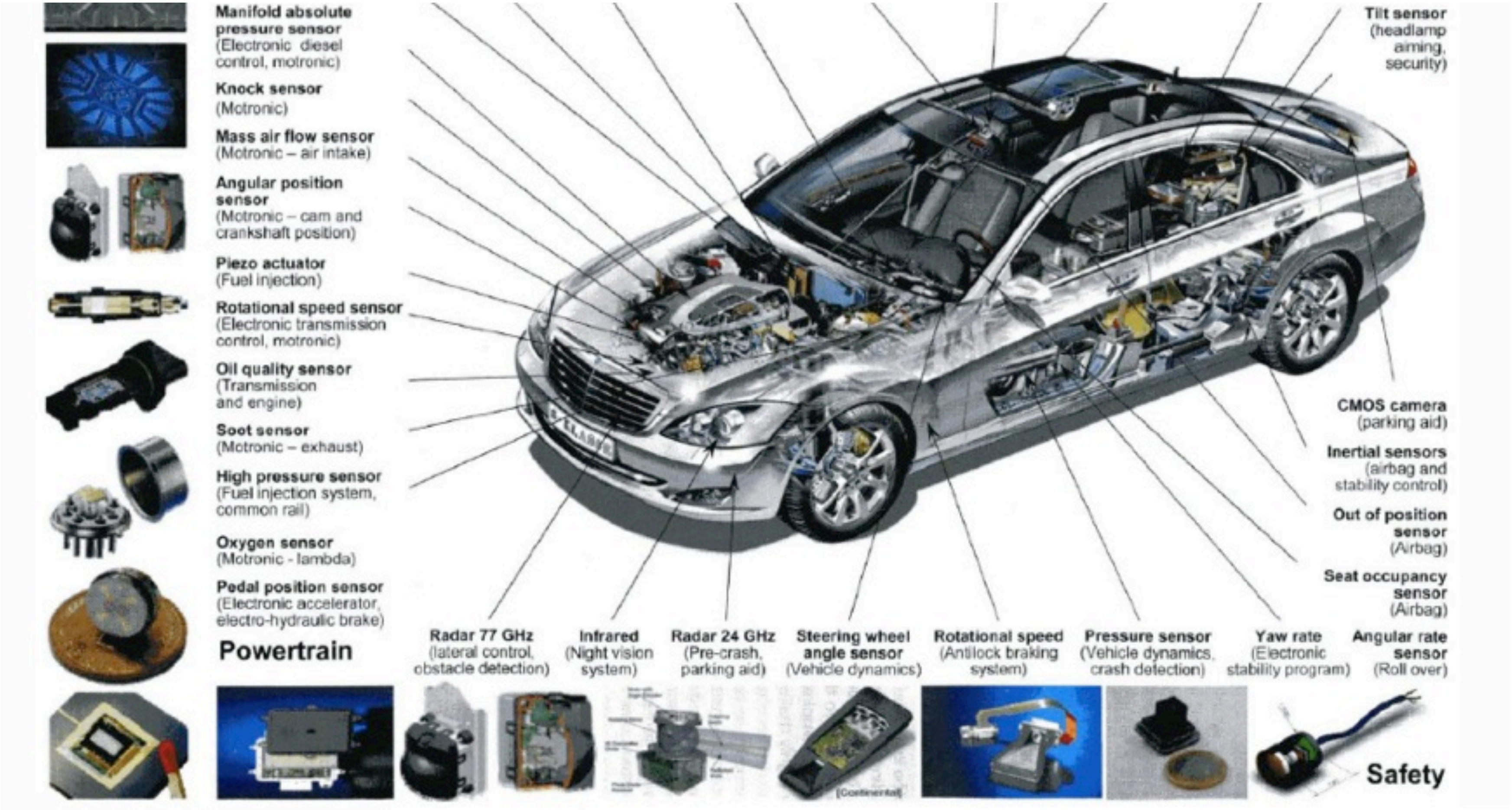
- All sorts of magic
  - Dashboard display
  - Keyless entry
  - GPS
  - Bluetooth

# WHAT IS A CAR?

- Four wheels and an engine

- All sorts of magic
  - Dashboard display
  - Keyless entry
  - GPS
  - Bluetooth

# Cars are **complex, highly-connected networks**


Manifold absolute pressure sensor (Electronic diesel control, motronic)
Knock sensor (Motronic)
Mass air flow sensor (Motronic – air intake)
Angular position sensor (Motronic – cam and crankshaft position)
Piezo actuator (Fuel injection)
Rotational speed sensor (Electronic transmission control, motronic)
Oil quality sensor (Transmission and engine)
Soot sensor (Motronic – exhaust)
High pressure sensor (Fuel injection system, common rail)
Oxygen sensor (Motronic - lambda)
Pedal position sensor (Electronic accelerator, electro-hydraulic brake)
Powertrain
Radar 77 GHz (lateral control, obstacle detection)
Infrared (Night vision system)
Radar 24 GHz (Pre-crash, parking aid)
Steering wheel angle sensor (Vehicle dynamics)
Rotational speed (Antilock braking system)
Pressure sensor (Vehicle dynamics, crash detection)
Yaw rate (Electronic stability program)
Angular rate sensor (Roll over)
Tilt sensor (headlamp aiming, security)
CMOS camera (parking aid)
Inertial sensors (airbag and stability control)
Out of position sensor (Airbag)
Seat occupancy sensor (Airbag)
Safety

# CAN: Controller Area Network

## Background

- Low-level network protocol
  - Introduced by Bosch in 1986
  - Part of OBD-II standard
  - Started appearing in 2003, mandated in US in 2008
  - Numerous varieties (TTCAN, CANopen, etc.)

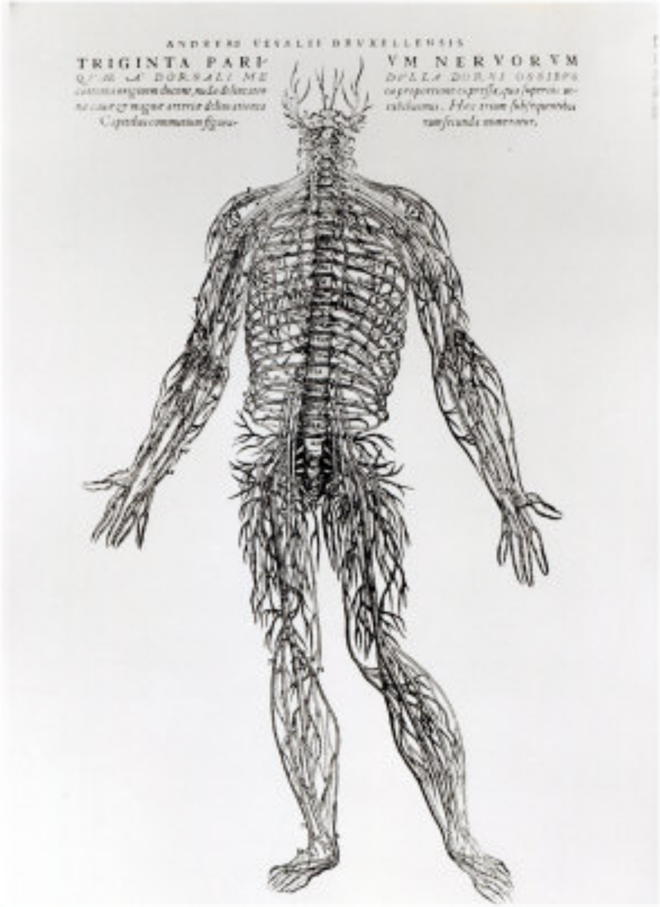
# CAN: Controller Area Network

Background

- Multi-master broadcast network
  - CSMA/CR with bitwise arbitration
  - ALL nodes receive ALL messages
- Optimized for speed and reliability, **not security**
  - **Most transmissions are not authenticated**
- Short messages
  - Max 8 data bytes

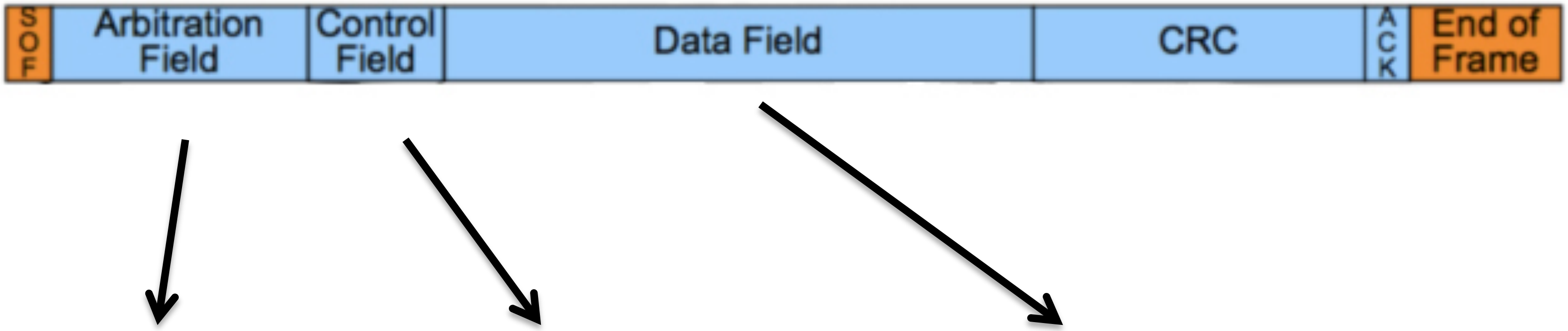
# CAN: Controller Area Network

An analogy!



# CAN: Controller Area Network

A CAN packet



38 88 B0 F9 | 08 | 05 00 10 20 EA AE 9A A3

ArbID | DLC | Data Payload

# State of the Art

## CAN Protocol Analyzers

Inexpensive (<$200)

- Handheld scan tools for OBD-II port
  - Standard ISO-mandated PIDs and diagnostics

Midrange ($200-2000)

- Link to computer, built in Oscope, etc.

Proprietary ($10-50K, licensed by manufacturer)

- In-depth analysis
- Writing capabilities
  - Re-flash ECU firmware

# State of the Art

## Existing Research

### Hobbyists

- Lots of scattered, (generally) poorly documented work
- CANduino

### Academics

- Center for Automotive Embedded Systems Security (autosec.org)

# Center for Automotive Embedded Systems Security

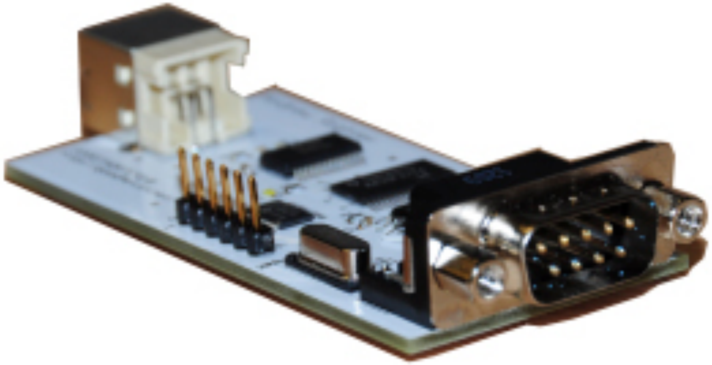
- autosec.org
- Examine vulnerabilities and attack surfaces
  - "…were able to forcibly and completely disengage the brakes while driving…."
  - "…Bluetooth and cellular vectors allow total car compromise by exploiting flaws in the telematics unit…"

# Center for Automotive Embedded Systems Security

- autosec.org
- Examine vulnerabilities and attack surfaces
  - "…were able to forcibly and completely disengage the brakes while driving…."
  - "…Bluetooth and cellular vectors allow total car compromise by exploiting flaws in the telematics unit…"
- BUT:
  - "…we have **purposely omitted crucial details** from our papers that would be required to replicate our work."
  - **THAT'S NO FUN!**

# The GoodTHOPTER10

*Travis Goodspeed and Andrew Righter*

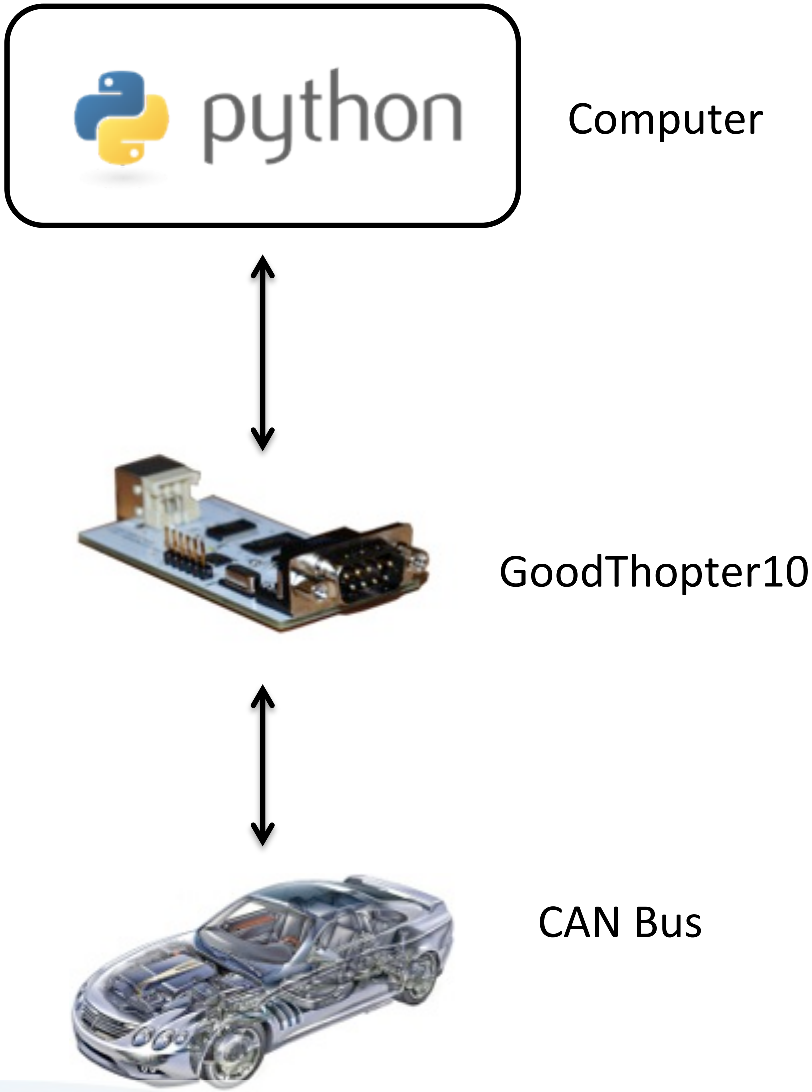


"…a sexy GoodFET CAN device…"

– A Neighbor

"THAT'S your final project??"

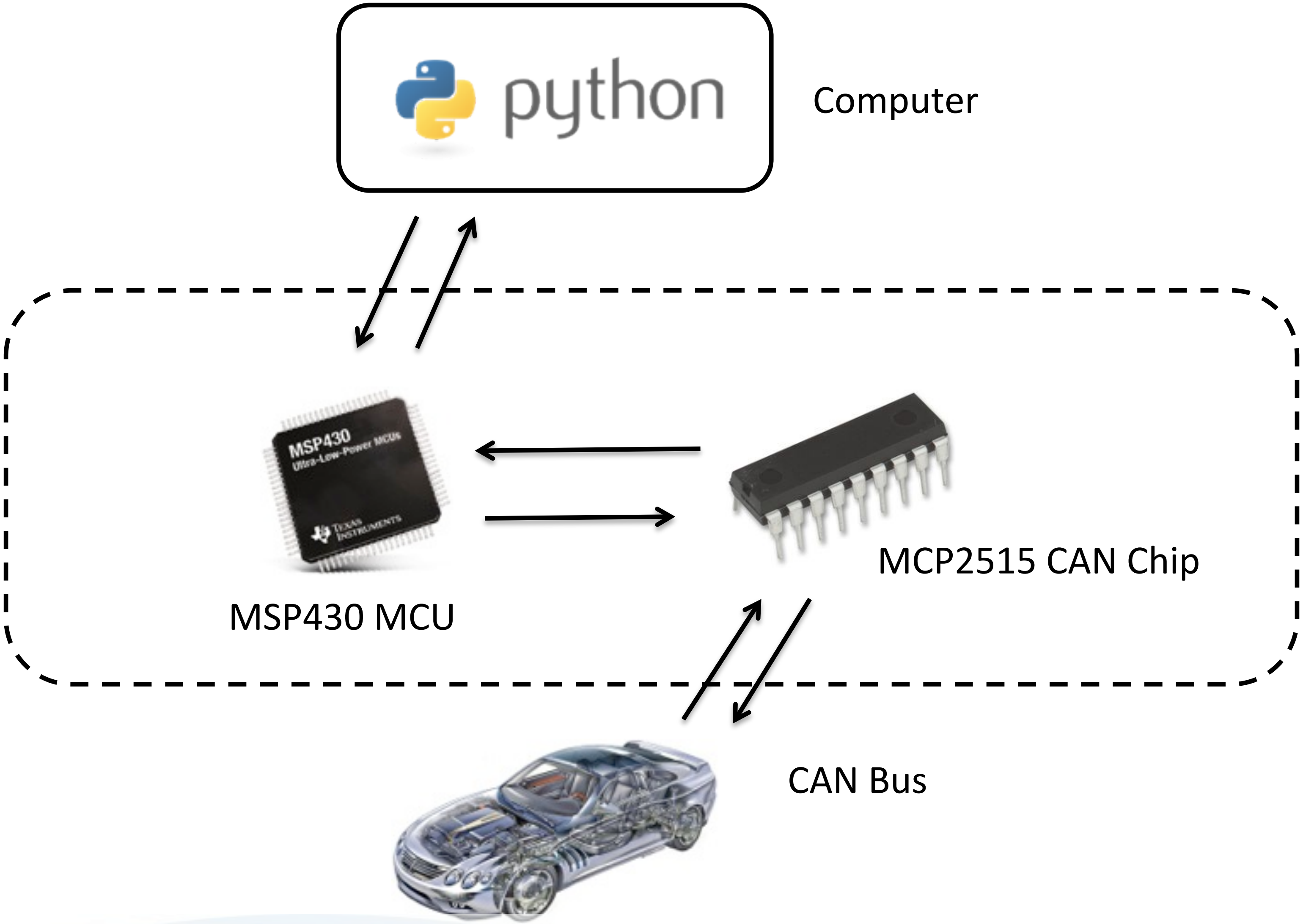
– Mechanical Engineers



Computer

GoodThopter10

CAN Bus



Computer

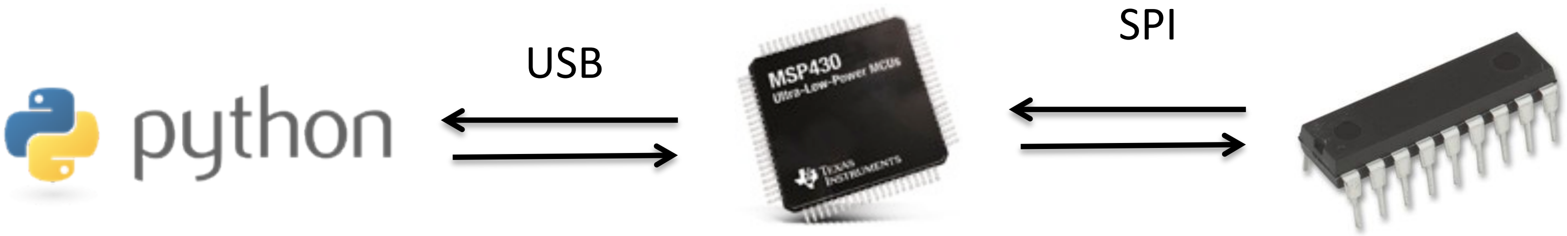
MSP430 MCU

MCP2515 CAN Chip

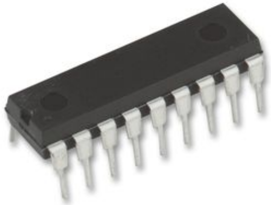
CAN Bus

- MSP430 relays to MCP2515
  - Configure CAN timing, filters
  - Read and write CAN packets

- Slow relative to CAN
  - MSP430 firmware needs a little tweaking

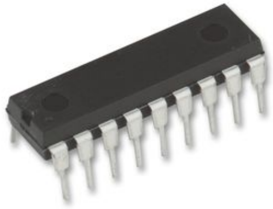


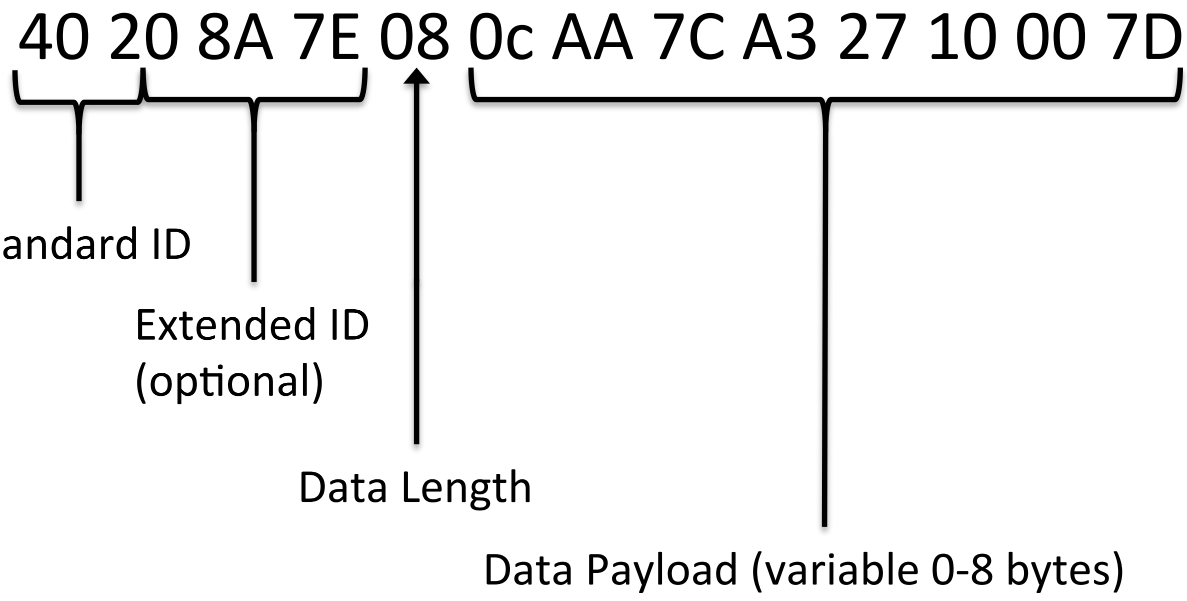
MCP2515 CAN Chip



- Hardware filtering for specific IDs and data
- (Interrupts for packet RX and errors)

# MCP2515 CAN Chip





40 20 8A 7E 08 0c AA 7C A3 27 10 00 7D

Standard ID

Extended ID (optional)

Data Length

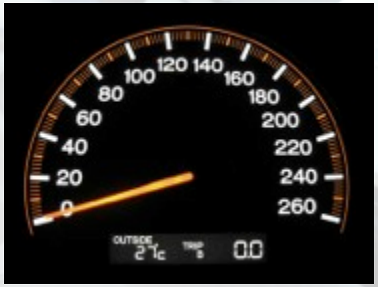
Data Payload (variable 0-8 bytes)

WE BOUGHT A CAR!

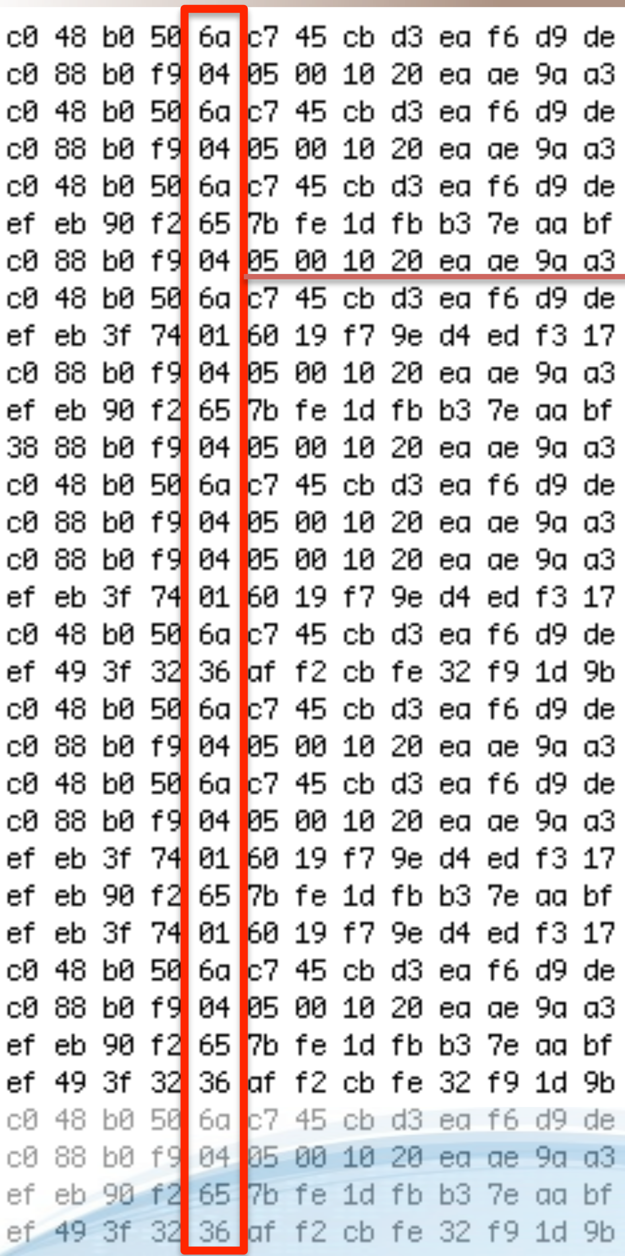
```
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
ef eb 90 f2 65 7b fe 1d fb b3 7e aa bf
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
ef eb 3f 74 01 60 19 f7 9e d4 ed f3 17
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
ef eb 90 f2 65 7b fe 1d fb b3 7e aa bf
38 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
ef eb 3f 74 01 60 19 f7 9e d4 ed f3 17
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
ef 49 3f 32 36 af f2 cb fe 32 f9 1d 9b
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
ef eb 3f 74 01 60 19 f7 9e d4 ed f3 17
ef eb 90 f2 65 7b fe 1d fb b3 7e aa bf
ef eb 3f 74 01 60 19 f7 9e d4 ed f3 17
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
ef eb 90 f2 65 7b fe 1d fb b3 7e aa bf
ef 49 3f 32 36 af f2 cb fe 32 f9 1d 9b
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
ef eb 90 f2 65 7b fe 1d fb b3 7e aa bf
ef 49 3f 32 36 af f2 cb fe 32 f9 1d 9b
```

# Understanding Packets

```
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
ef eb 90 f2 65 7b fe 1d fb b3 7e aa bf
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
ef eb 3f 74 01 60 19 f7 9e d4 ed f3 17
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
ef eb 90 f2 65 7b fe 1d fb b3 7e aa bf
38 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
ef eb 3f 74 01 60 19 f7 9e d4 ed f3 17
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
ef 49 3f 32 36 af f2 cb fe 32 f9 1d 9b
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
ef eb 3f 74 01 60 19 f7 9e d4 ed f3 17
ef eb 90 f2 65 7b fe 1d fb b3 7e aa bf
ef eb 3f 74 01 60 19 f7 9e d4 ed f3 17
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
ef eb 90 f2 65 7b fe 1d fb b3 7e aa bf
ef 49 3f 32 36 af f2 cb fe 32 f9 1d 9b
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
ef eb 90 f2 65 7b fe 1d fb b3 7e aa bf
ef 49 3f 32 36 af f2 cb fe 32 f9 1d 9b
```



# (NOT) Understanding Packets

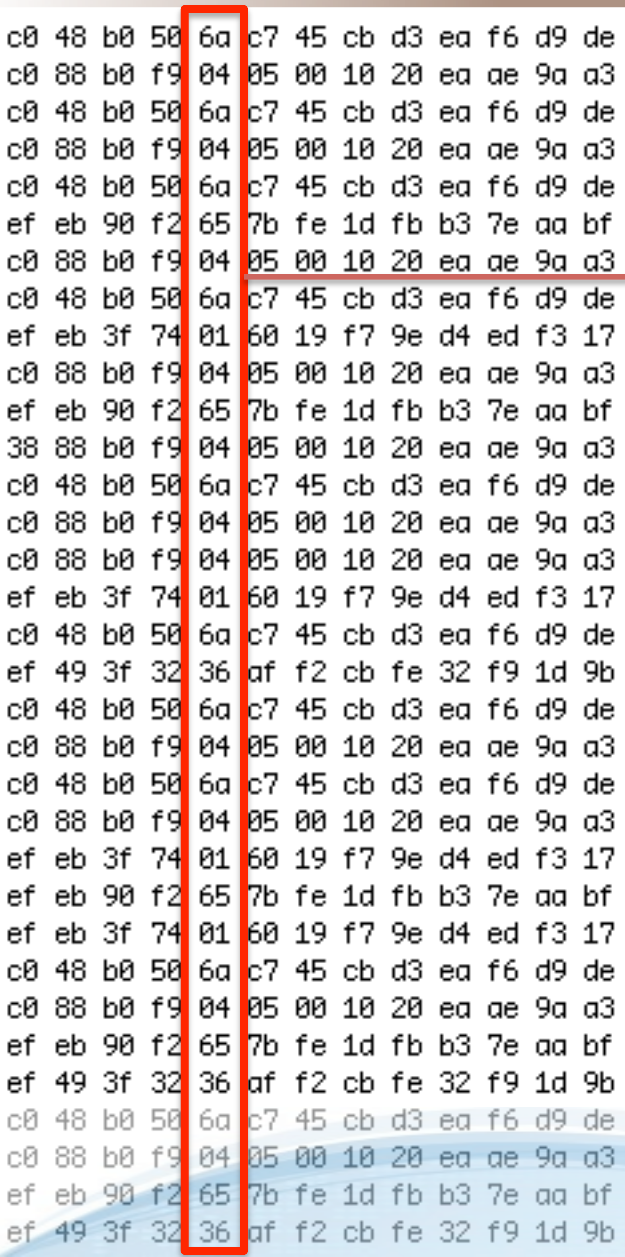


```
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
ef eb 90 f2 65 7b fe 1d fb b3 7e aa bf
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
ef eb 3f 74 01 60 19 f7 9e d4 ed f3 17
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
ef eb 90 f2 65 7b fe 1d fb b3 7e aa bf
38 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
ef eb 3f 74 01 60 19 f7 9e d4 ed f3 17
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
ef 49 3f 32 36 af f2 cb fe 32 f9 1d 9b
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
ef eb 3f 74 01 60 19 f7 9e d4 ed f3 17
ef eb 90 f2 65 7b fe 1d fb b3 7e aa bf
ef eb 3f 74 01 60 19 f7 9e d4 ed f3 17
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
ef eb 90 f2 65 7b fe 1d fb b3 7e aa bf
ef 49 3f 32 36 af f2 cb fe 32 f9 1d 9b
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
ef eb 90 f2 65 7b fe 1d fb b3 7e aa bf
ef 49 3f 32 36 af f2 cb fe 32 f9 1d 9b
```

**Data Length Code doesn’t make any sense.**

**Data is complete noise.**

# (NOT) Understanding Packets



```
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
ef eb 90 f2 65 7b fe 1d fb b3 7e aa bf
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
ef eb 3f 74 01 60 19 f7 9e d4 ed f3 17
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
ef eb 90 f2 65 7b fe 1d fb b3 7e aa bf
38 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
ef eb 3f 74 01 60 19 f7 9e d4 ed f3 17
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
ef 49 3f 32 36 af f2 cb fe 32 f9 1d 9b
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
ef eb 3f 74 01 60 19 f7 9e d4 ed f3 17
ef eb 90 f2 65 7b fe 1d fb b3 7e aa bf
ef eb 3f 74 01 60 19 f7 9e d4 ed f3 17
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
ef eb 90 f2 65 7b fe 1d fb b3 7e aa bf
ef 49 3f 32 36 af f2 cb fe 32 f9 1d 9b
c0 48 b0 50 6a c7 45 cb d3 ea f6 d9 de
c0 88 b0 f9 04 05 00 10 20 ea ae 9a a3
ef eb 90 f2 65 7b fe 1d fb b3 7e aa bf
ef 49 3f 32 36 af f2 cb fe 32 f9 1d 9b
```

**Data Length Code doesn't make any sense.**

**Data is complete noise.**

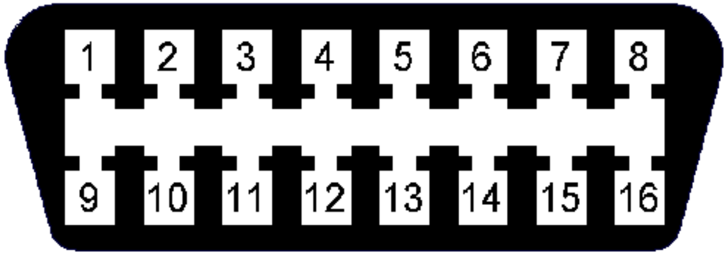
*Wow, CAN is tricky!*

WE BOUGHT A CAR…

THAT DIDN’T USE CAN.

# Onboard Diagnostic Port (OBD-II)

1996: Made mandatory for all cars sold in the U.S.
2008: SAE requires all new US vehicles use the CAN bus



1 – Vehicle specific
2 – J1850
3 – blank
4 – Chassis Ground
5 – Signal Ground
6 – CAN High
7 – ISO 9141 2K
8 – Vehicle specific

9 – Vehicle specific
10 – J1850 bus
11 – Vehicle specific
12 – Vehicle specific
13 – Signal Ground
14 – CAN Low
15 – ISO 19141 2L
16 – Battery Power

# LESSONS LEARNED

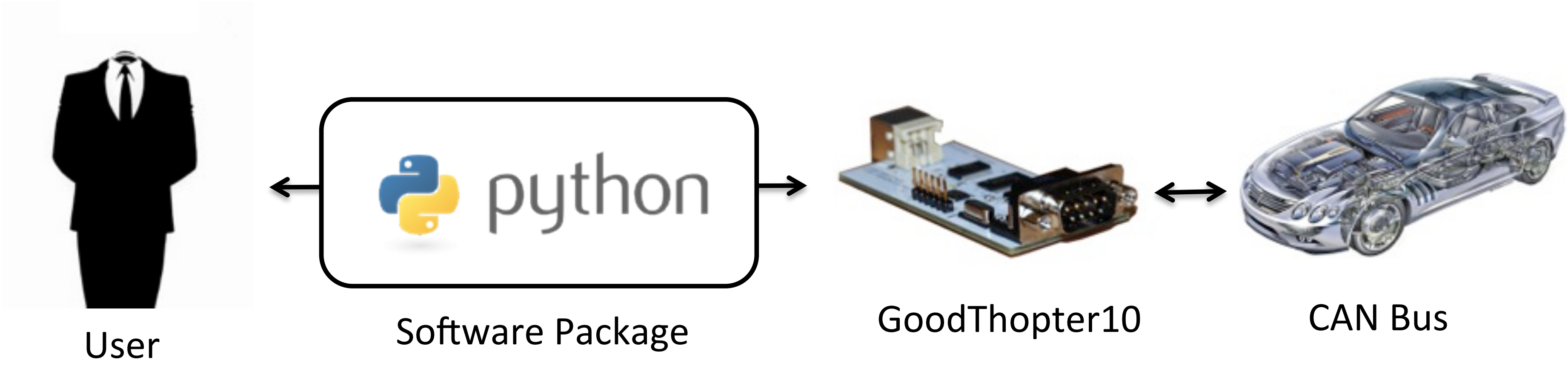
- Hard to find reliable information on manufacturers' protocols
- Double-check with physical pinout

There are ***TWO TYPES*** of OBD-II to RS-232 cables!

- One of them routes J1850 to CAN…

…WE BOUGHT ANOTHER CAR.

# ..and now, the project!

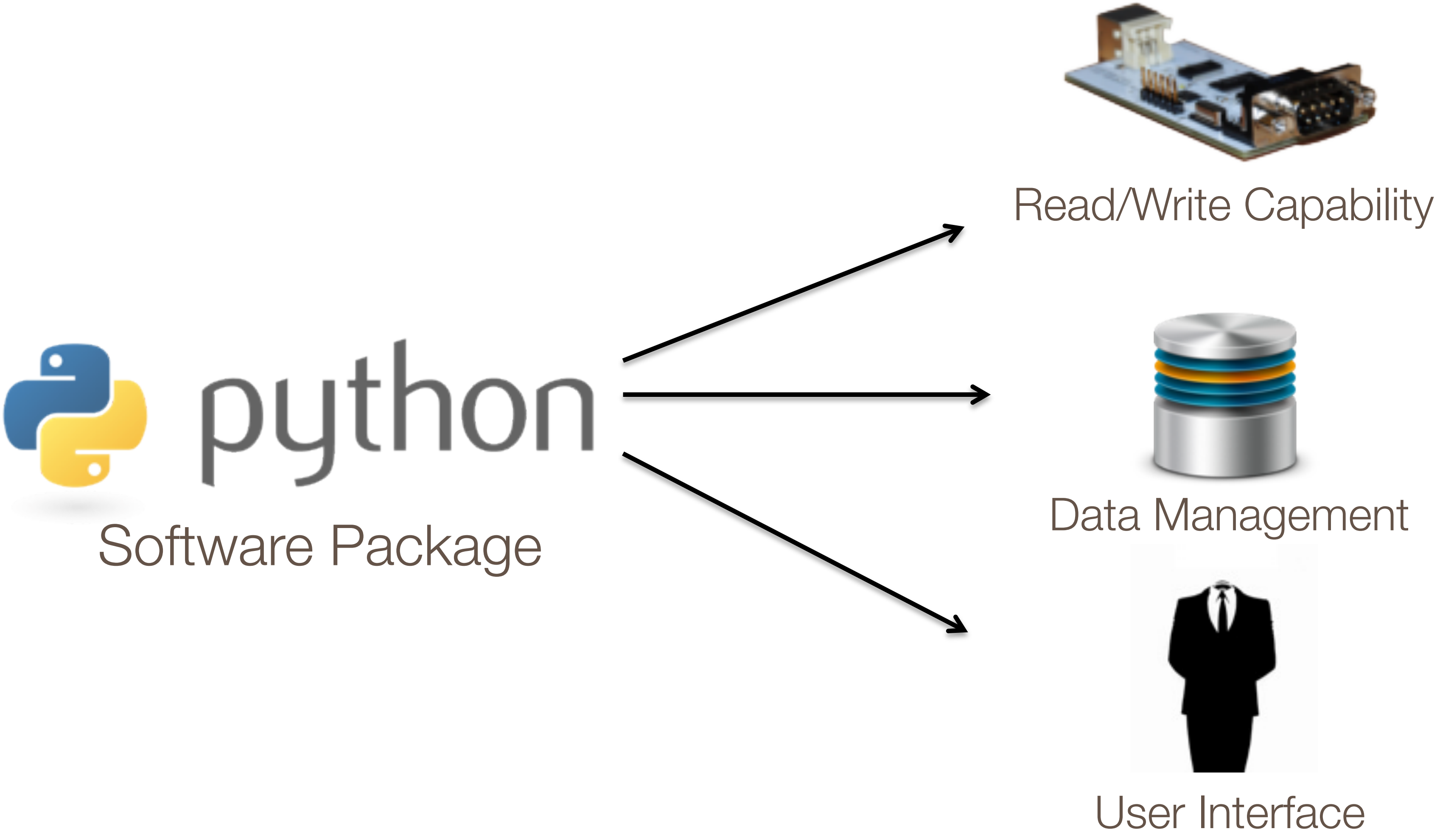


User

Software Package

GoodThopter10

CAN Bus

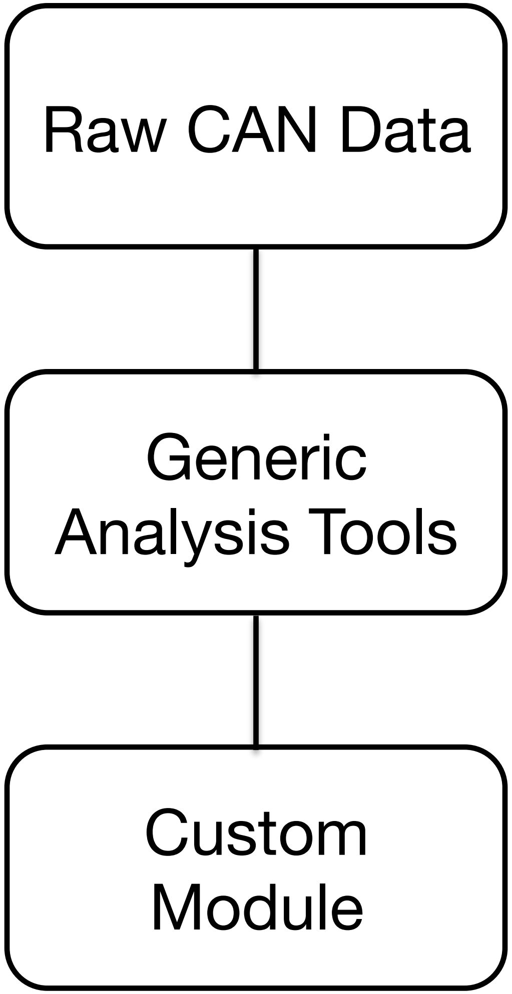


Read/Write Capability

Software Package

Data Management

User Interface



**Raw CAN Data**

- Simple communications with Goodthopter10
- Digital Logic communications

**Generic Analysis Tools**

- Experimental Methods
- Advanced communications

**Custom Module**

- Car/User Specific
- Not necessary

# User Interface

https://www.youtube.com/watch?v=hpwBmTw7Gm0

# User Interface

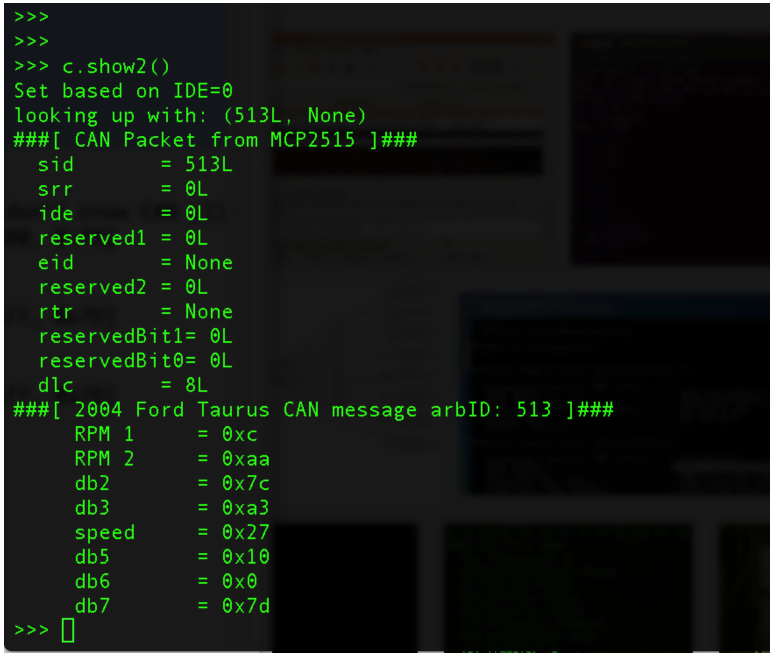
Key Features

- Data logging integrated with MYSQL
- Sniffing options
  - Up to 6 positive filters
  - Set length, comment tagging
- Custom packet injection
  - Set delay time between packets
  - Set total number of writes
- Experimental methods
  - Basic fuzzing and packet response
- Documentation Window
- Custom module option

# Packet Manipulation with Scapy

## Current project

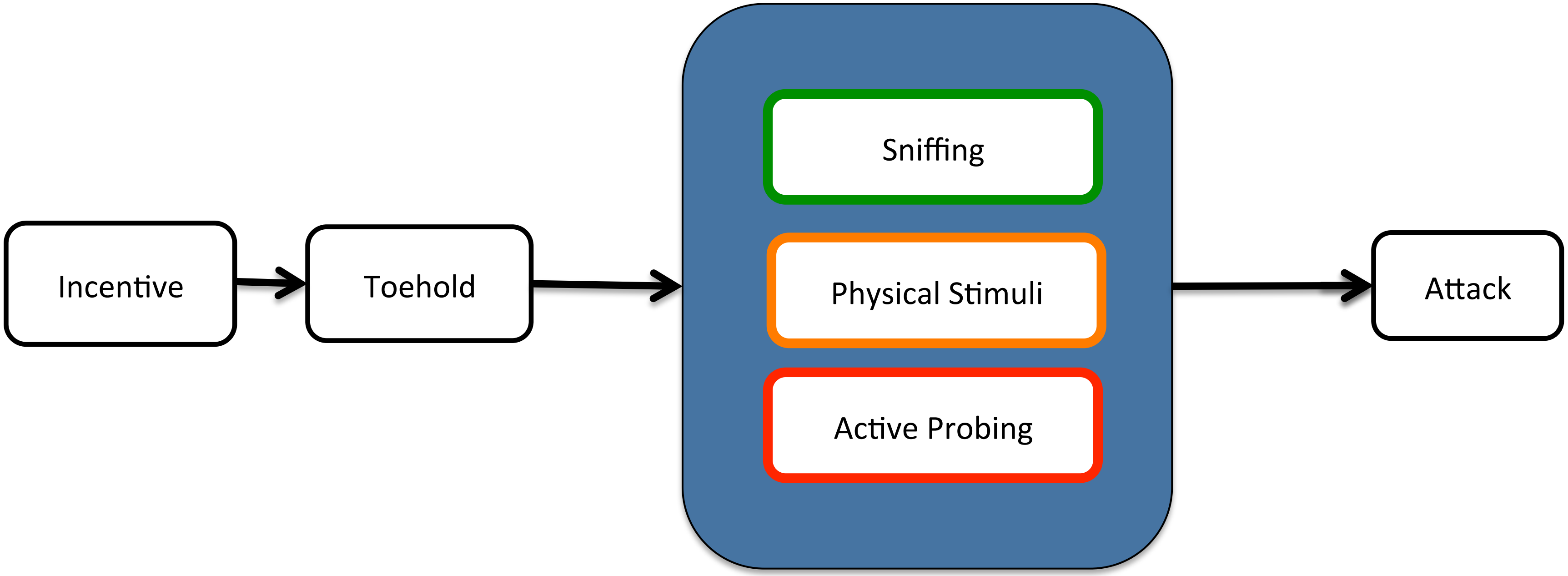
- Wrote custom scapy protocol: CAN_MCP2515
- Rewriting codebase to integrate new protocol



```
>>>
>>>
>>> c.show2()
Set based on IDE=0
looking up with: (513L, None)
###[ CAN Packet from MCP2515 ]###
  sid        = 513L
  srr        = 0L
  ide        = 0L
  reserved1 = 0L
  eid        = None
  reserved2 = 0L
  rtr        = None
  reservedBit1= 0L
  reservedBit0= 0L
  dlc        = 8L
###[ 2004 Ford Taurus CAN message arbID: 513 ]###
     RPM 1      = 0xc
     RPM 2      = 0xaa
     db2        = 0x7c
     db3        = 0xa3
     speed      = 0x27
     db5        = 0x10
     db6        = 0x0
     db7        = 0x7d
>>>
```

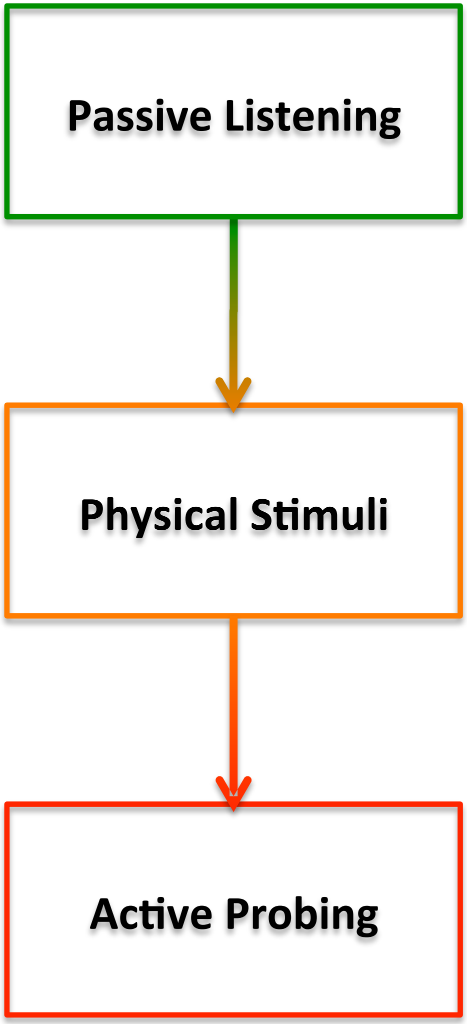
# Reverse-Engineering the CAN Bus

AKA “Explaining Hacking to Mechanical Engineers”



Identify Exploits

# Hacking a Car

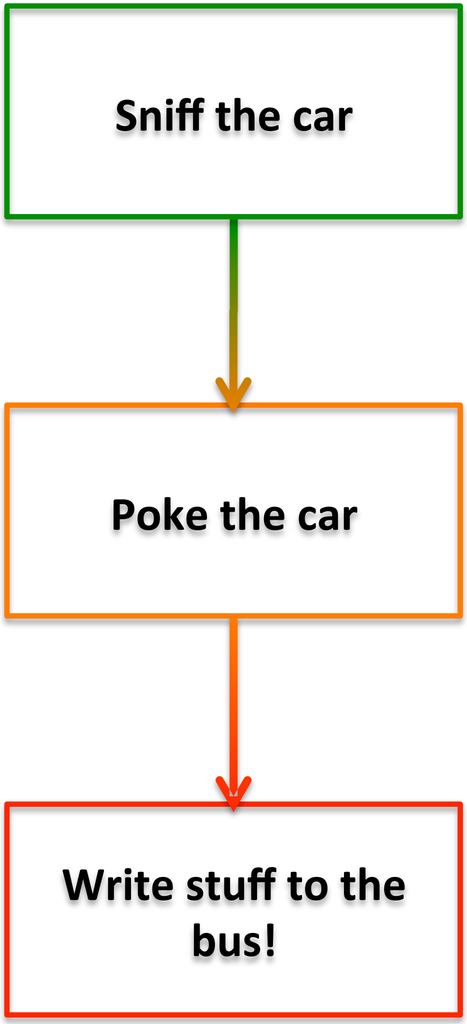


Passive Listening

Physical Stimuli

Active Probing

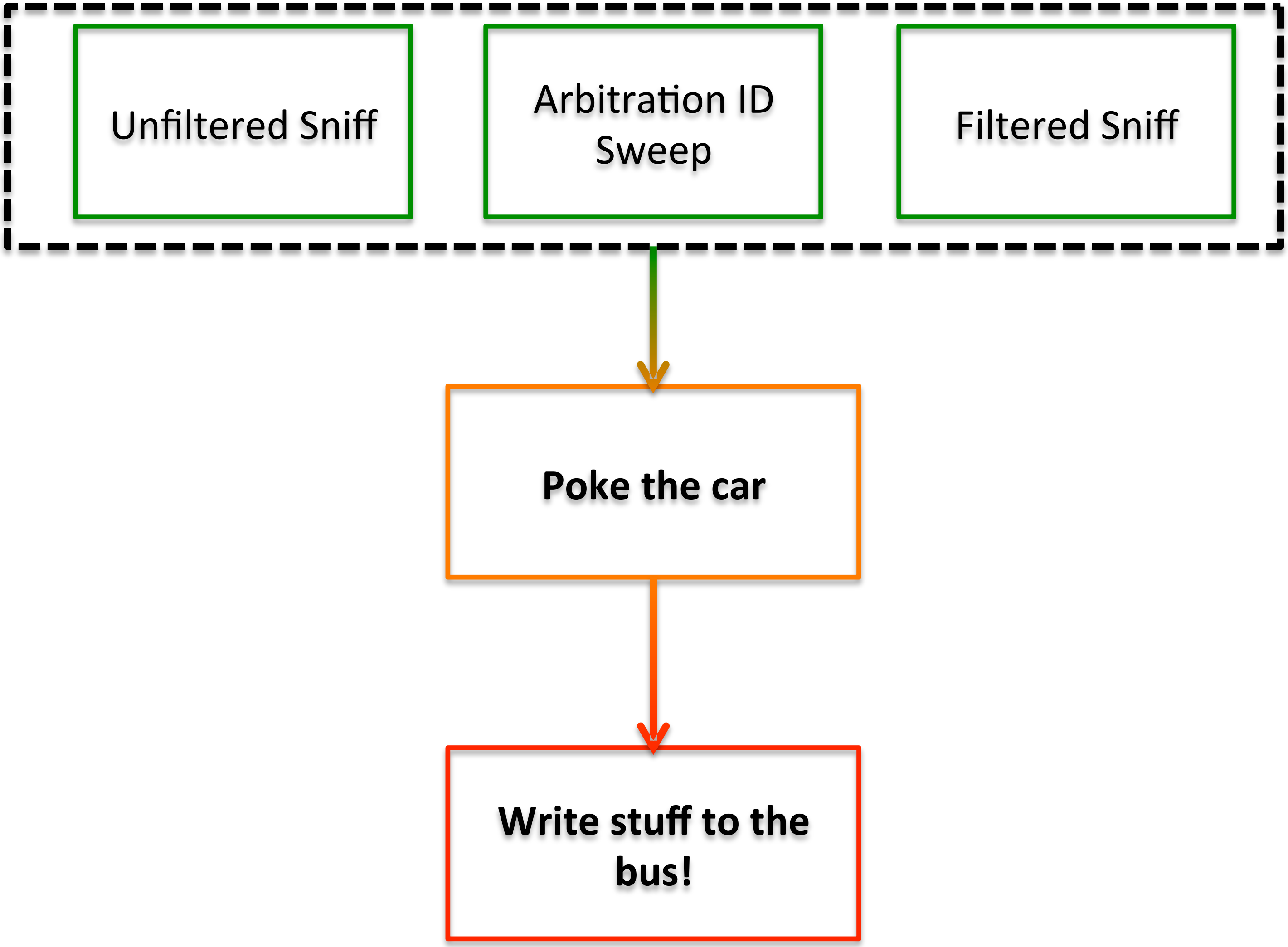
# Hacking a Car



Sniff the car

Poke the car

Write stuff to the bus!


Unfiltered Sniff
Arbitration ID Sweep
Filtered Sniff
Poke the car
Write stuff to the bus!

Unfiltered Sniff

Arbitration ID Sweep

| ArbID | 201 | 202 | 211 | 212 | 230 | 420 | 421 | 430 | 4E0 | 4FF |
|---|---|---|---|---|---|---|---|---|---|---|
| Freq (packets/ second) | 115 | 62 | 50 | 50 | 112 | 10 | 0.2 | 50 | 1 | 1 |

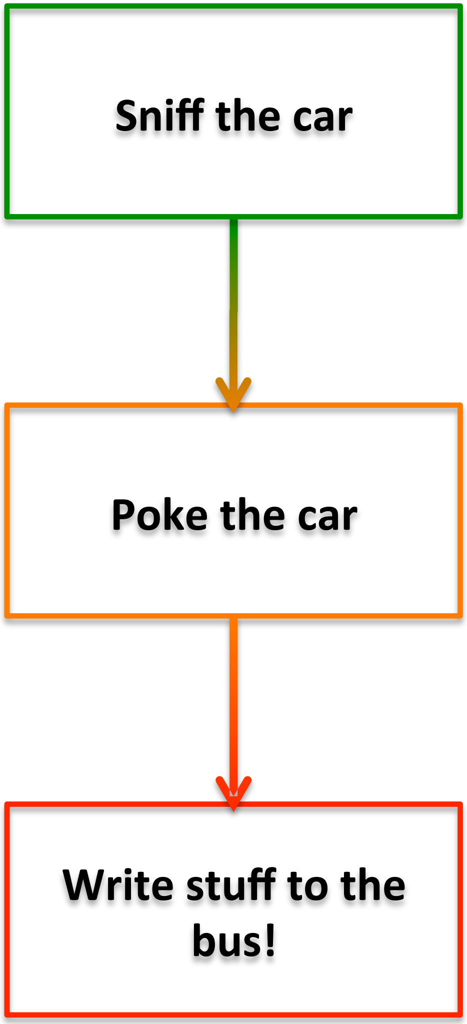
Filtered Sniff

## A Case Study – ID 0x201

| ArbID | Frequency (packets/second) | Background Packets |
|---|---|---|
| 0x201 | 62 | OC xx 7D xx 27 10 0 7D |

*xx = varying

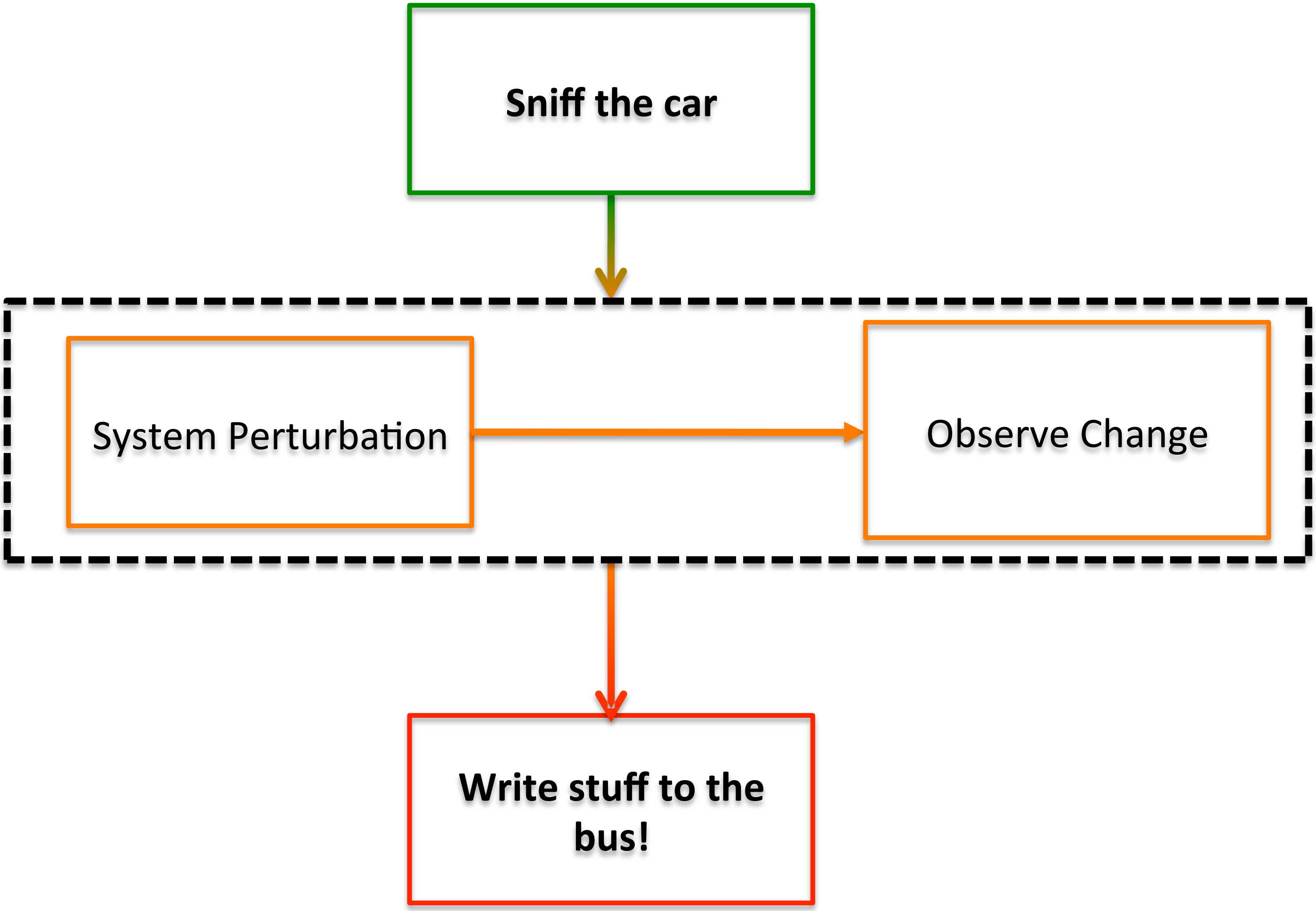
# Hacking a Car



Sniff the car

Poke the car

Write stuff to the bus!

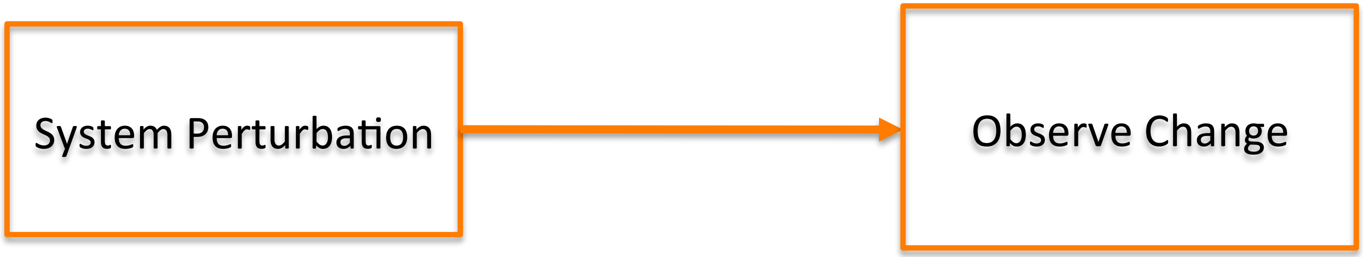


Sniff the car

System Perturbation

Observe Change

Write stuff to the bus!



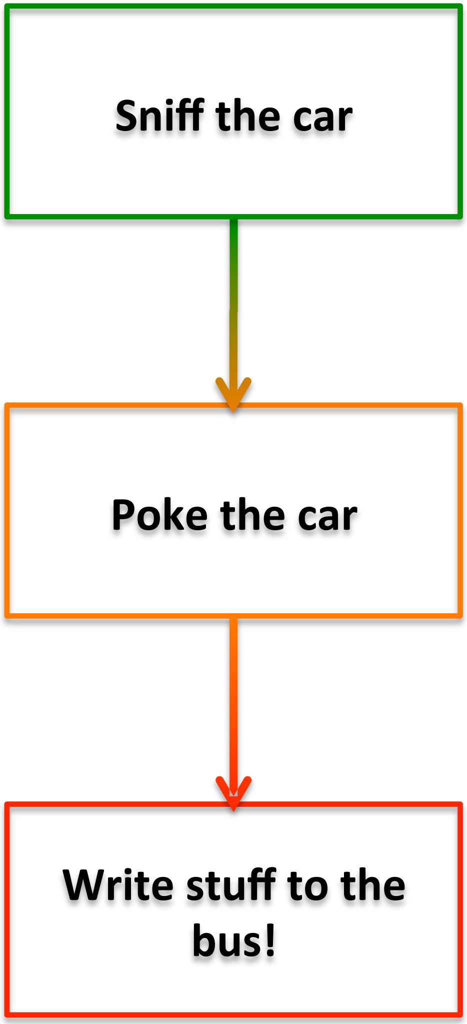
System Perturbation

Observe Change

## A Case Study – ID 0x201

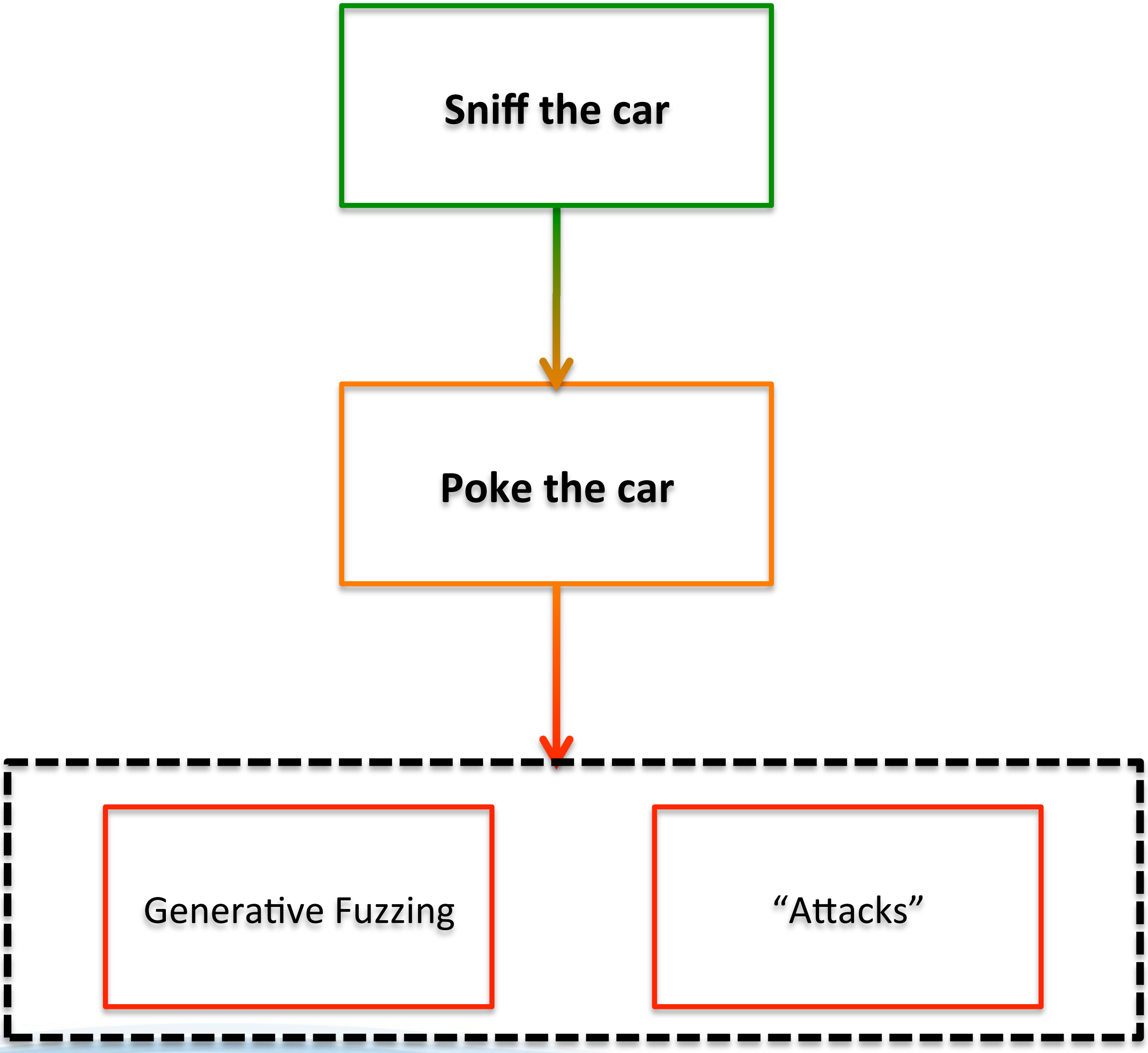
| ArbID | Perturbation Response | Data Byte Correlation |
|---|---|---|
| 0x201 | Revving Engine | Bytes 0, 1 |
|  | Increase while driving | Byte 4 |

# Hacking a Car


Sniff the car

Poke the car

Write stuff to the bus!



Sniff the car

Poke the car

Generative Fuzzing

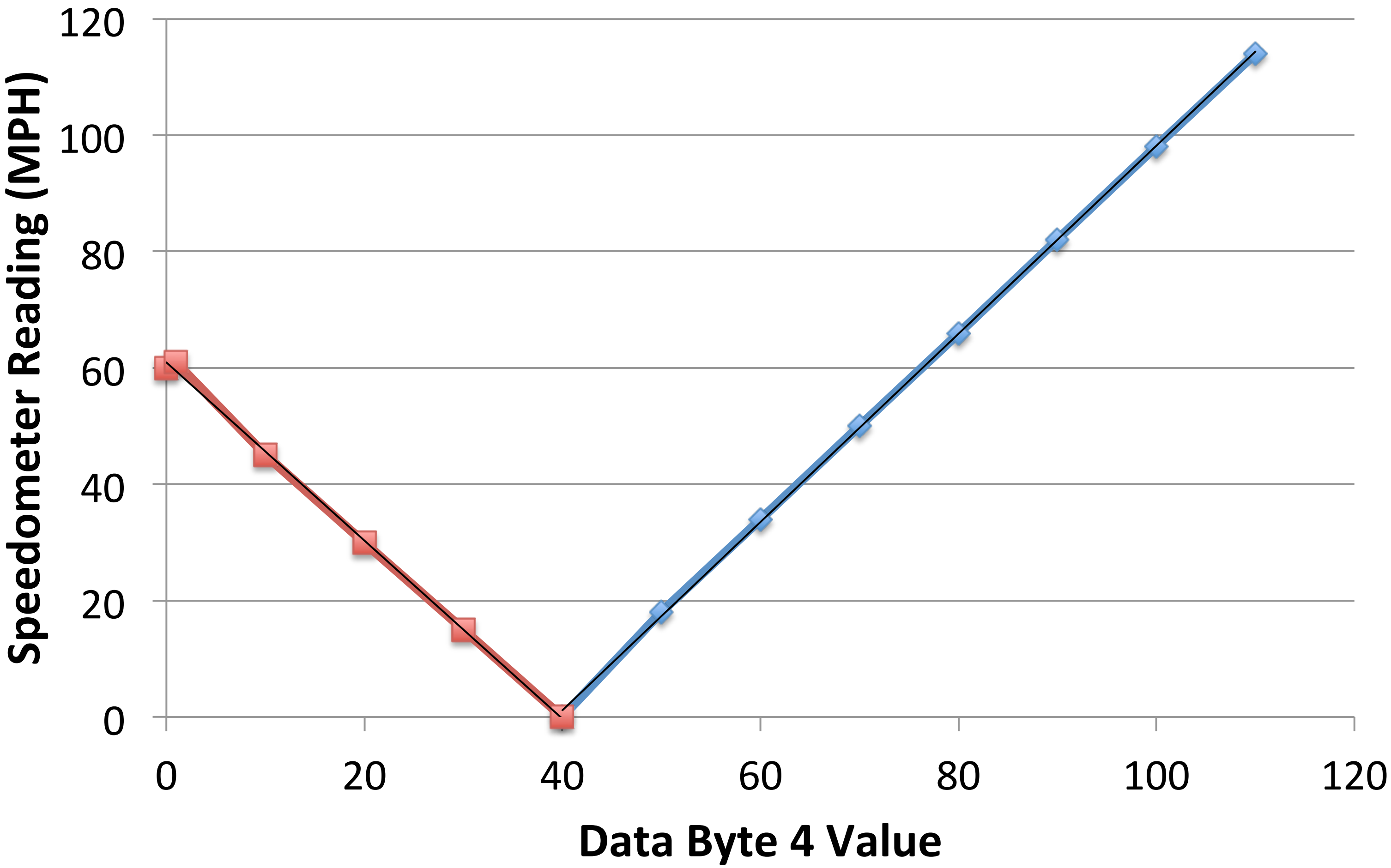
“Attacks”

Generative Fuzzing

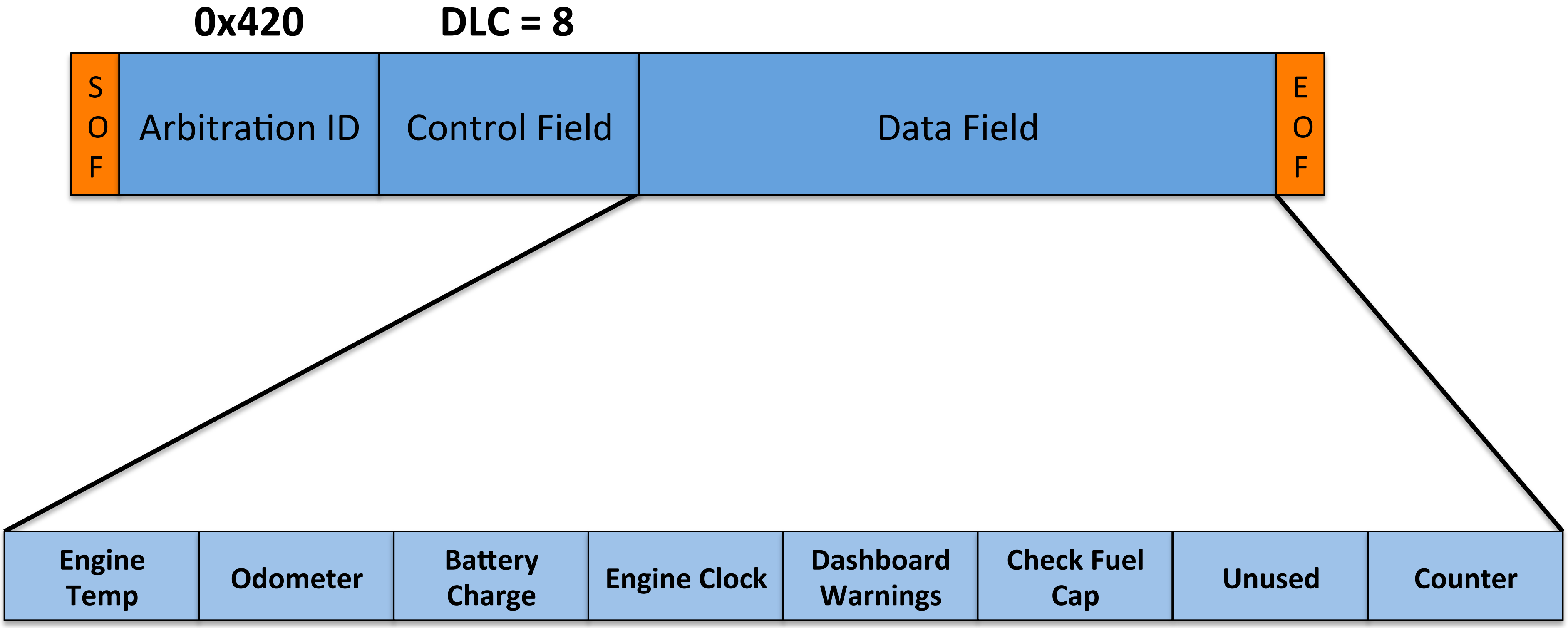
## A Case Study – ID 201

| ArbID | Fuzzing Response | Refined Inferences |
|---|---|---|
| 201 | Dashboard components change | Bytes 0, 1 = RPM |
| | | Byte 4 = Speedometer |

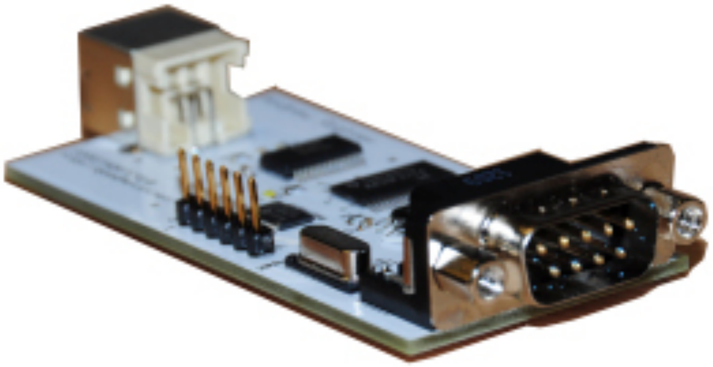
# Higher Level Protocols: ID 0x201



# Higher Level Protocols: ID 0x420



# CAN Bus Analyzer – State of the Art



VS

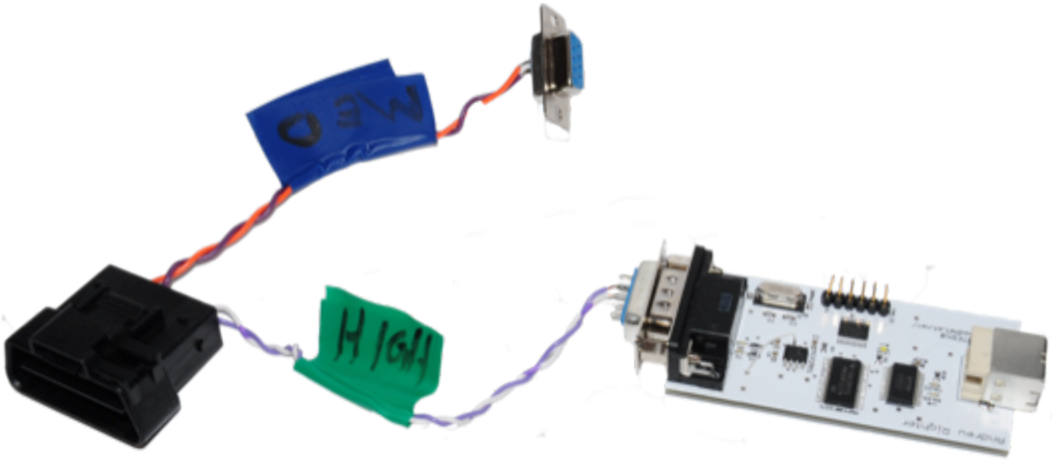
# CAN Bus Analyzer – State of the Art

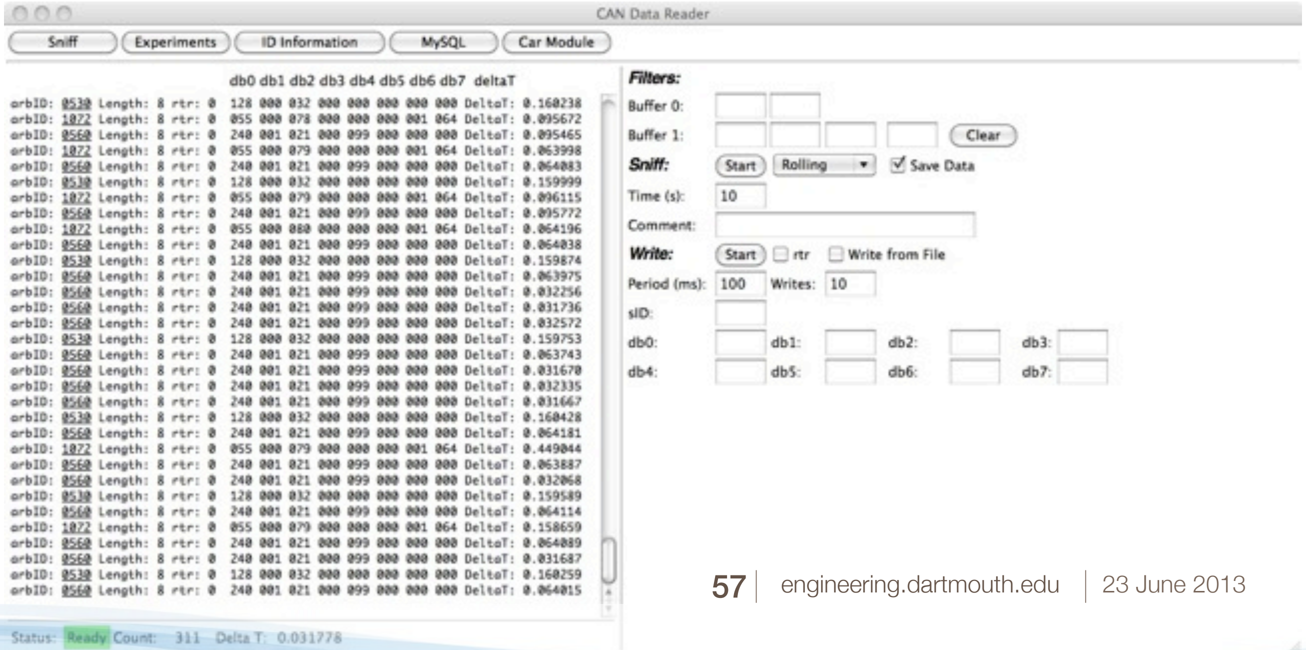
| | Microchip Analyzer | Our Package |
|---|---|---|
| **View Types** | Hex or Decimal | Decimal |
| **Read Speed** | 1000 packets/second | 112 packets/second |
| **Read Packet Types** | Standard, Extended | Standard, Extended |
| **Read Packet Lengths** | Varying | Varying |
| **Sniff Views** | Fixed, Rolling | Fixed, Rolling |
| **Write Speed** | Up to 10 packets/second | Up to 1000 packets/second |
| **Write from file** | ✗ | ✓ |
| **Write packet types** | Standard, Extended | Standard |
| **Write packet Lengths** | Varying | 8 bytes |
| **Error checking** | ✓ | ✗ |
| **Data logging** | ✓ | ✓ |
| **Customizable** | ✗ | ✓ |
| **MySQL Integration** | ✗ | ✓ |
| **Save to Wireshark** | ✗ | ✓ |
| **Intuitive User Interface** | ✓ | ✓ |
| **Notation Section** | ✗ | ✓ |
| **Cost** | $ 100.00 | $ 27.50 |

# CAN Bus Analyzer – State of the Art

| | **Microchip Analyzer** | **Our Package** |
|---|---|---|
| **View Types** | Hex or Decimal | Decimal |
| **Read Speed** | 1000 packets/second | 112 packets/second |
| **Read Packet Types** | Standard, Extended | Standard, Extended |
| **Read Packet Lengths** | Varying | Varying |
| **Sniff Views** | Fixed, Rolling | Fixed, Rolling |
| **Write Speed** | Up to 10 packets/second | Up to 1000 packets/second |
| **Write from file** | ✗ | ✓ |
| **Write packet types** | Standard, Extended | Standard |
| **Write packet Lengths** | Varying | 8 bytes |
| **Error checking** | ✓ | ✗ |
| **Data logging** | ✓ | ✓ |
| **Customizable** | ✗ | ✓ |
| **MySQL Integration** | ✗ | ✓ |
| **Save to Wireshark** | ✗ | ✓ |
| **Intuitive User Interface** | ✓ | ✓ |
| **Notation Section** | ✗ | ✓ |
| **Cost** | $ 100.00 | $ 27.50 |

# Conclusion: The only tools you need





```
                                       db0 db1 db2 db3 db4 db5 db6 db7  deltaT
arbID: 0530 Length: 8 rtr: 0  128 000 032 000 000 000 000 000 DeltaT: 0.160238
arbID: 1072 Length: 8 rtr: 0  055 000 078 000 000 000 001 064 DeltaT: 0.095672
arbID: 0560 Length: 8 rtr: 0  240 001 021 000 099 000 000 000 DeltaT: 0.095465
arbID: 1072 Length: 8 rtr: 0  055 000 079 000 000 000 001 064 DeltaT: 0.063998
arbID: 0560 Length: 8 rtr: 0  240 001 021 000 099 000 000 000 DeltaT: 0.064083
arbID: 0530 Length: 8 rtr: 0  128 000 032 000 000 000 000 000 DeltaT: 0.159999
arbID: 1072 Length: 8 rtr: 0  055 000 079 000 000 000 001 064 DeltaT: 0.096115
arbID: 0560 Length: 8 rtr: 0  240 001 021 000 099 000 000 000 DeltaT: 0.095772
arbID: 1072 Length: 8 rtr: 0  055 000 080 000 000 000 001 064 DeltaT: 0.064196
arbID: 0560 Length: 8 rtr: 0  240 001 021 000 099 000 000 000 DeltaT: 0.064038
arbID: 0530 Length: 8 rtr: 0  128 000 032 000 000 000 000 000 DeltaT: 0.159874
arbID: 0560 Length: 8 rtr: 0  240 001 021 000 099 000 000 000 DeltaT: 0.063975
arbID: 0560 Length: 8 rtr: 0  240 001 021 000 099 000 000 000 DeltaT: 0.032256
arbID: 0560 Length: 8 rtr: 0  240 001 021 000 099 000 000 000 DeltaT: 0.031736
arbID: 0560 Length: 8 rtr: 0  240 001 021 000 099 000 000 000 DeltaT: 0.032572
arbID: 0530 Length: 8 rtr: 0  128 000 032 000 000 000 000 000 DeltaT: 0.159753
arbID: 0560 Length: 8 rtr: 0  240 001 021 000 099 000 000 000 DeltaT: 0.063743
arbID: 0560 Length: 8 rtr: 0  240 001 021 000 099 000 000 000 DeltaT: 0.031670
arbID: 0560 Length: 8 rtr: 0  240 001 021 000 099 000 000 000 DeltaT: 0.032335
arbID: 0560 Length: 8 rtr: 0  240 001 021 000 099 000 000 000 DeltaT: 0.031667
arbID: 0530 Length: 8 rtr: 0  128 000 032 000 000 000 000 000 DeltaT: 0.160428
arbID: 0560 Length: 8 rtr: 0  240 001 021 000 099 000 000 000 DeltaT: 0.064181
arbID: 1072 Length: 8 rtr: 0  055 000 079 000 000 000 001 064 DeltaT: 0.449044
arbID: 0560 Length: 8 rtr: 0  240 001 021 000 099 000 000 000 DeltaT: 0.063887
arbID: 0560 Length: 8 rtr: 0  240 001 021 000 099 000 000 000 DeltaT: 0.032068
arbID: 0530 Length: 8 rtr: 0  128 000 032 000 000 000 000 000 DeltaT: 0.159589
arbID: 0560 Length: 8 rtr: 0  240 001 021 000 099 000 000 000 DeltaT: 0.064114
arbID: 1072 Length: 8 rtr: 0  055 000 079 000 000 000 001 064 DeltaT: 0.158659
arbID: 0560 Length: 8 rtr: 0  240 001 021 000 099 000 000 000 DeltaT: 0.064089
arbID: 0560 Length: 8 rtr: 0  240 001 021 000 099 000 000 000 DeltaT: 0.031687
arbID: 0530 Length: 8 rtr: 0  128 000 032 000 000 000 000 000 DeltaT: 0.160259
arbID: 0560 Length: 8 rtr: 0  240 001 021 000 099 000 000 000 DeltaT: 0.064015

Status: Ready Count: 311 Delta T: 0.031778
```

**Filters:**

Buffer 0:

Buffer 1:

**Sniff:** Rolling ▾ ☑ Save Data

Time (s): 10

Comment:

**Write:** □ rtr □ Write from File

Period (ms): 100 Writes: 10

sID:

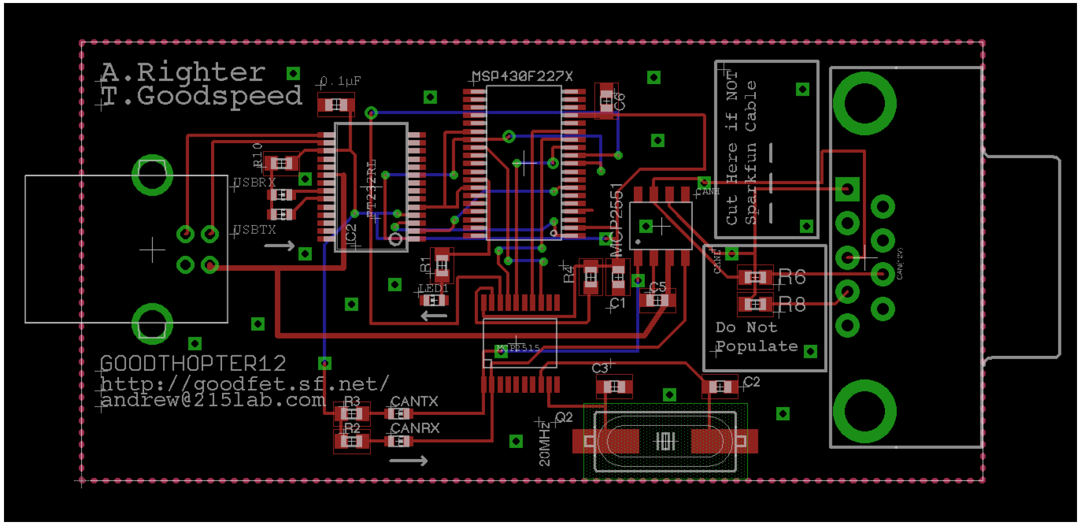
db0: db1: db2: db3:

db4: db5: db6: db7:

# GET ON THE BUS!

# HERE'S YOUR TICKET!

http://goodfet.sourceforge.net/hardware/goodthopter12/



Email Travis:

*1. Your complete mailing address*

2. The number of PCBs you want

# OPEN-SOURCE GOODNESS

Goodfet: **http://goodfet.sourceforge.net/**
CAN GUI: **goodfet/contrib/reCAN/mainDisplay.py**

Python Packages:

- python-sqlite3
- py-serial
- python-tk
- MySQLdb (optional)
- Scapy (optional)

# Ongoing Work

- Improve GoodTHOPTER12 firmware
- Reverse engineer more manufacturers' HLPs
  - Sniff traffic and test methodology on more cars
- Support for ISO-mandated PIDs
  - http://en.wikipedia.org/wiki/OBD-II_PIDs
- Integration with Scapy
  - River Loop Security

# CAN Caveats

- Some manufacturers implement segmented buses
- Other protocols
  - FlexRay (more expensive, German vehicles)
  - LIN (cheaper, limited capabilities)

# Thanks to:

**Project Sponsor:**

Siege Technologies and Daniel Bilar

**Advisors:**

Professor Sergey Bratus<br>
Professor Vincent Burke

**Friendly Neighbors:**

Travis Goodspeed<br>
Andrew Righter<br>
Ryan Speers (River Loop Security)

**Thayer School of Engineering:**

*(sorry about the car, guys!)*

# Citations

## Image Citations

Slide 10: Brooks, Quinn. "Why Is My Gas Mileage so Bad??" SC200. Penn State, 18 Nov. 2012. Web. 17 June 2013. <http://www.personal.psu.edu/afr3/blogs/siowfa12/2012/11/my-car-is-about-ten.html>.

Slide 11: Strang, Thomas, Dr, and Matthias Rockl. "Vehicle Networks." Vehicle Networks. Deutsches Zentrum Für Luft- Und Raumfahrt E. V. 17 June 2013. Lecture.

Slide 18: "Banque D'images Et De Portraits." BIU Santé, Paris. N.p., n.d. Web. 17 June 2013. <http://www.biusante.parisdescartes.fr/histmed/image?01080>.

Slide 35: The Dark Knight - Legendary Pictures, Syncopy Films, DC Comics, Warner Bros. Pictures

Slide 68: N.d. Photograph. n.p. Web. 17 Jun 2013. <http://dnok91peocsw3.cloudfront.net/inspiration/122137-612x612-1.png>.

## Additional Resources

“CAN Datasheet.” Bosch Semiconductors. N.p.. Web. 3 Mar 2013. http://www.bosch- semiconductors.de.

Checkoway, Stephen, et al. “Comprehensive experimental analyses of automotive attack surfaces.” Proceedings of USENIX Security. 2011.

Koscher, Karl, et al. “Experimental security analysis of a modern automobile.” Security and Privacy (SP), 2010 IEEE Symposium on. IEEE, 2010.

Wright, Alex. “Hacking cars.” Communications of the ACM 54.11 (2011): 18-19.

“CAN Bus Analyzer.” Microchip. MicroChip. Web. 3 Mar 2013. http: //www.microchip.com/stellent/idcplg? IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en546534.

Wolf, Marko, Andr Weimerskirch, and Thomas Wollinger. “State of the art: Embedding security in vehicles.” EURASIP Journal on Embedded Systems 2007 (2007).

Richards, Pat. “A CAN Physical Layer Discussion.” Microchip. Microchip, 16 Sep 2005. Web. 3 Mar 2013. http://www.microchip.com/stellent/idcplg?IdcService= SS_GET_PAGE&nodeId=1824&appnote=en012057