Heavy Vehicle Networks and Chip Level Forensics

Jeremy Daily, Ph.D., P.E.

Jeremy.Daily@colostate.edu

SYSTEMS ENGINEERING
COLORADO STATE UNIVERSITY

Heavy Vehicle Networks

Understanding SAE J1939 and CAN data

SYSTEMS ENGINEERING
COLORADO STATE UNIVERSITY

Chip Level Forensics

Extracting and decoding data from memory in electronic control units

SYSTEMS ENGINEERING
COLORADO STATE UNIVERSITY

Scope

- Chip level forensics is the process of extracting and decoding data from memory bearing chips inside electronic control units
- Utilized when network based forensics (i.e. downloads and diagnostics) are not tenable
- Board level and chip level forensics are treated similarly

Network Level

Data collected using a vehicle network or bench setup

- Cummins PowerSpec
- Bosch CDR Tool

Board Level

Data collected through internal debug and programming ports

- KTAG
- PE Micro

Chip Level

Data collected from a chip extracted from the board and read using a chip reader

- Xeltek Super Pro

Hybrid: Chip Transplants

Problem Statement

- We want to connect to a truck…

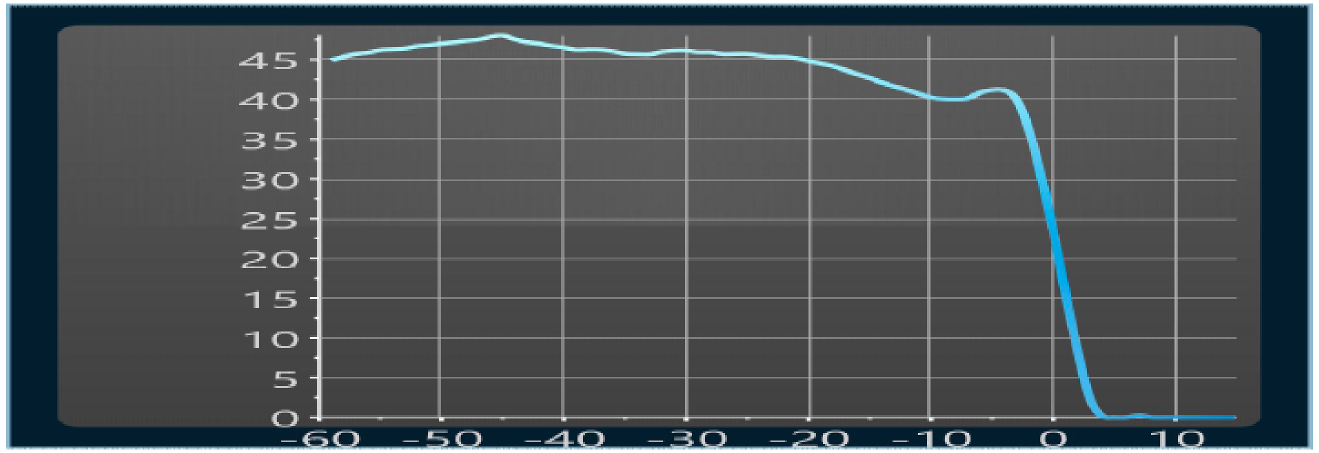
6

Vehicle Sudden Deceleration Report Record 1

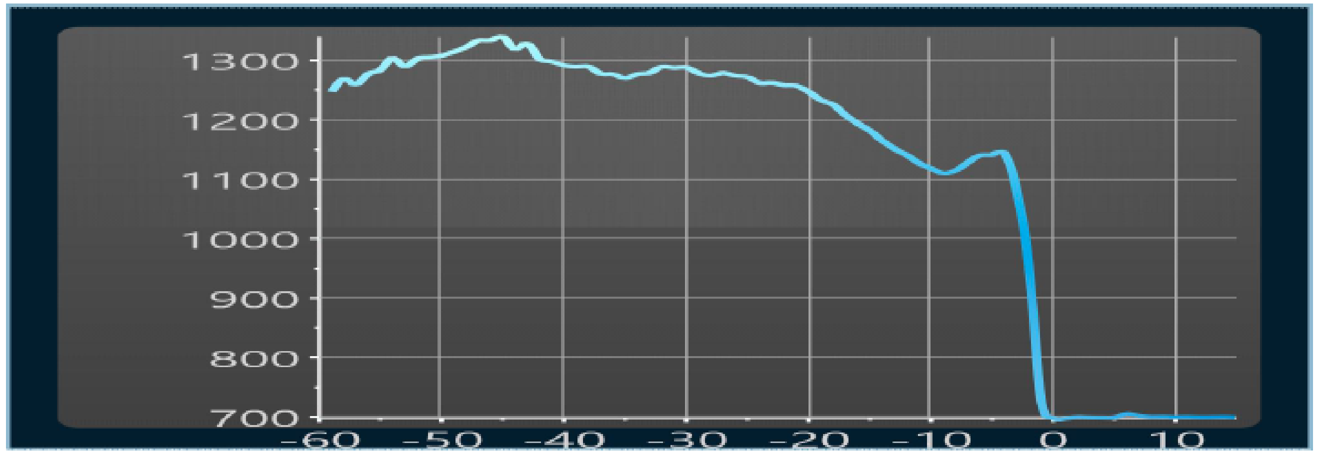
Engine Type	ISX 2013	Ecm Code	EF10067.34
Engine Serial Number	79749226	Software Phase	9.40.0.53
Unit Number	0000000000	Extraction Date	09-18-2017 10:00:16
Sudden Decel Threshold Rate:	7.00 mph	ECM Run time	8120:27:45

Occurrence Date: N\A	ECM Run Time at Occurrence: 7472:53:17
Air Temperature (°F) at Occurrence: 69	Occurrence Distance (mi): 371978.5

Vehicle Speed



Engine Speed

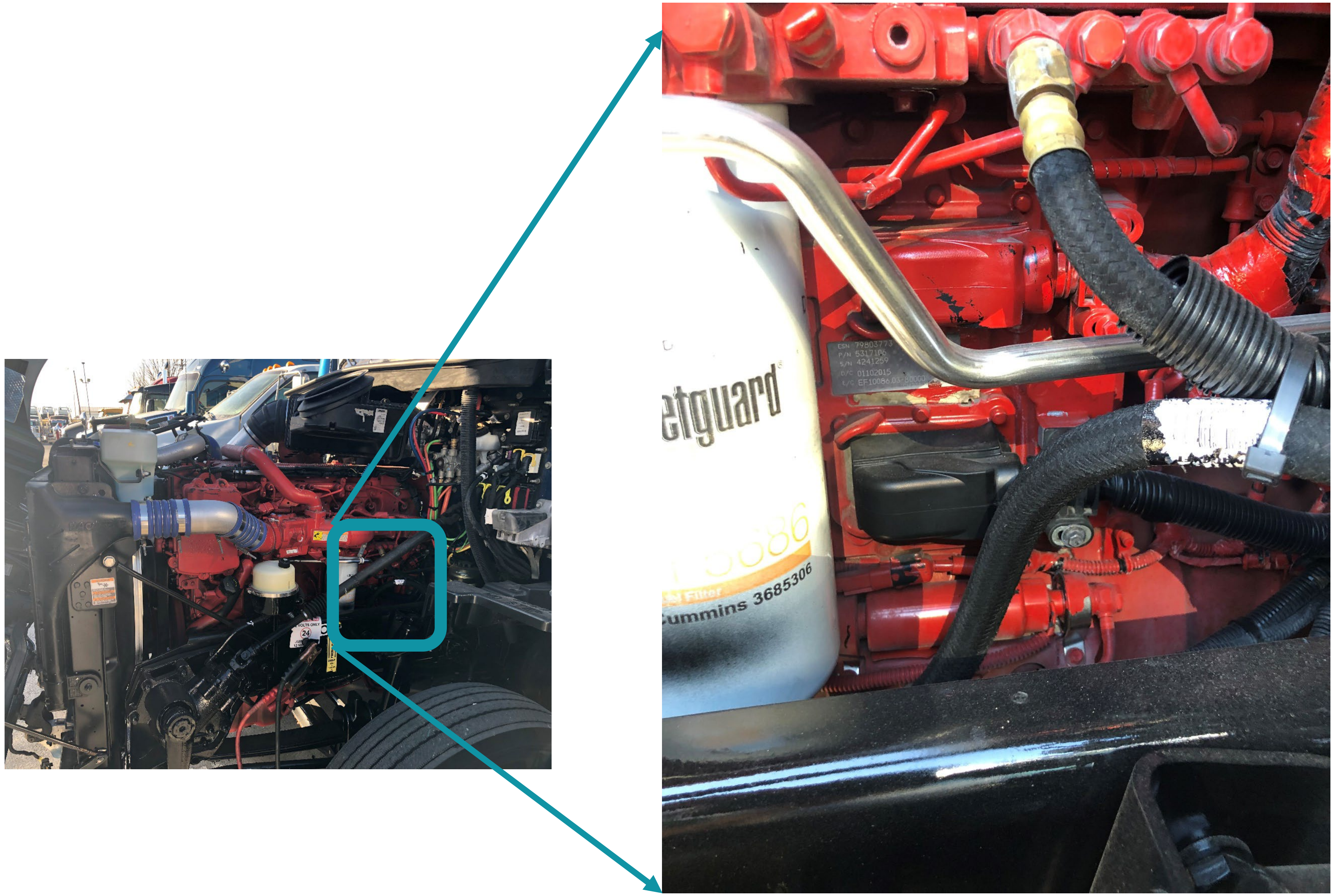


…and get data.

Example from
Cummins PowerSpec

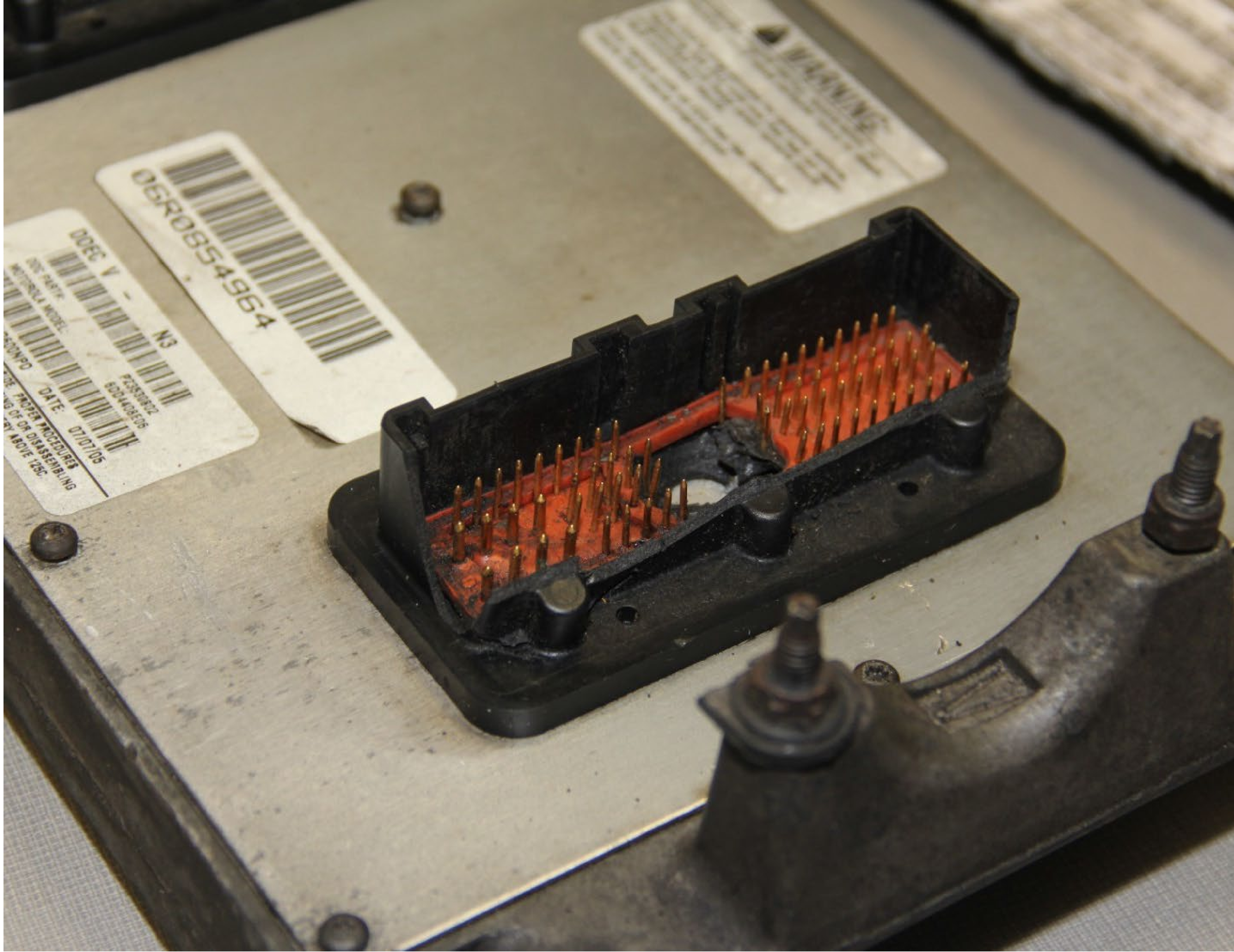
Engine Control Module Location

Engine Control Module Location



Sometimes the ECU is Broken

The Recovered Module: No Communications



DDEC Decoding (See SAE 2015-01-1450)

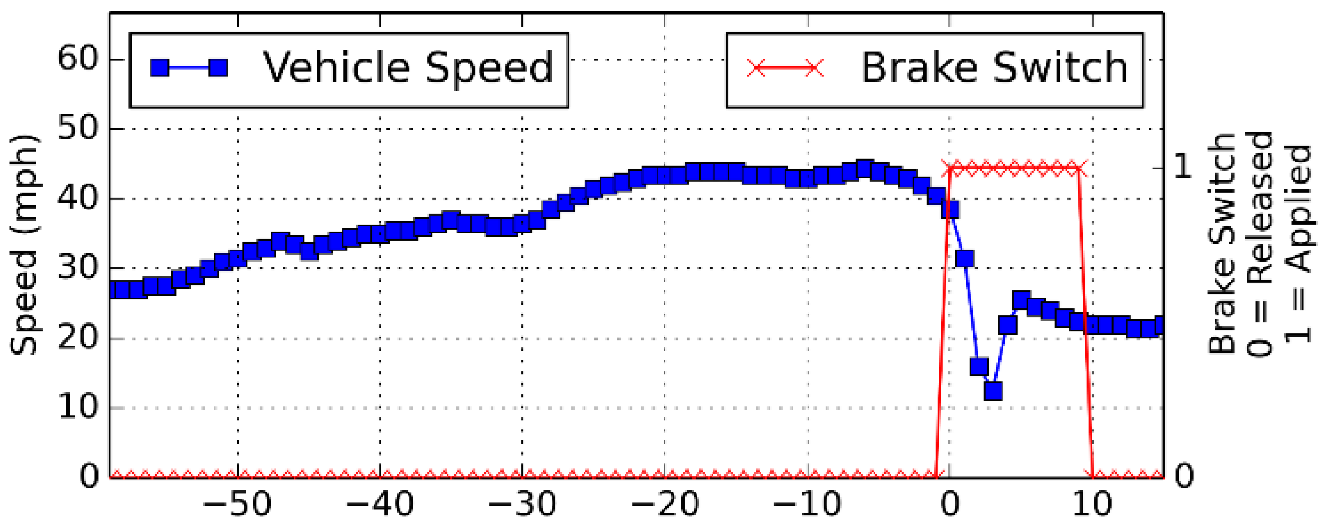
Extracting Event Data from Memory Chips within a Detroit Diesel DDEC V

2015-01-1450

Published 04/14/2015

Jeremy Daily, Andrew Kongs, James Johnson, and Jose Corcega
University of Tulsa

CITATION: Daily, J., Kongs, A., Johnson, J., and Corcega, J., "Extracting Event Data from Memory Chips within a Detroit Diesel DDEC V," SAE Technical Paper 2015-01-1450, 2015, doi:10.4271/2015-01-1450.



DDEC Data Decoding Overview

1. Problem definition
2. Figuring out what to look for (Produce Known Data)
3. Locating known data in memory from an exemplar ECM
4. Finding data in the subject ECM (Unknown)
5. Decoding and presenting the data

Normal: DDEC Reports

DDEC® Reports - Hard Brake #1

Print Date: 10/2/2013 2:30 PM
University of Tulsa

,

Trip: 09/17/12 12:26:15 To 10/02/13 (CST)
Vehicle ID: DDEC 6 TIB
Driver ID:
Odometer: 619.0 mi
Engine S/N: 06R1003832

Trip Distance	619.0 mi	Trip Time	0:00:00
Trip Fuel	0.00 gal	Fuel Consumption	0.00 gal/h
Fuel Economy	0.00 mpg	Idle Time	0:00:00
Avg Drive Load	0 %	Idle Percent	0.00 %
Avg Vehicle Speed	0.0 mph	Idle Fuel	0.00 gal
		Parked Regen Time	0:00:00

Incident Time: 10/2/2013 1:07:54 PM (CST) Incident Odometer: 619.0 mi

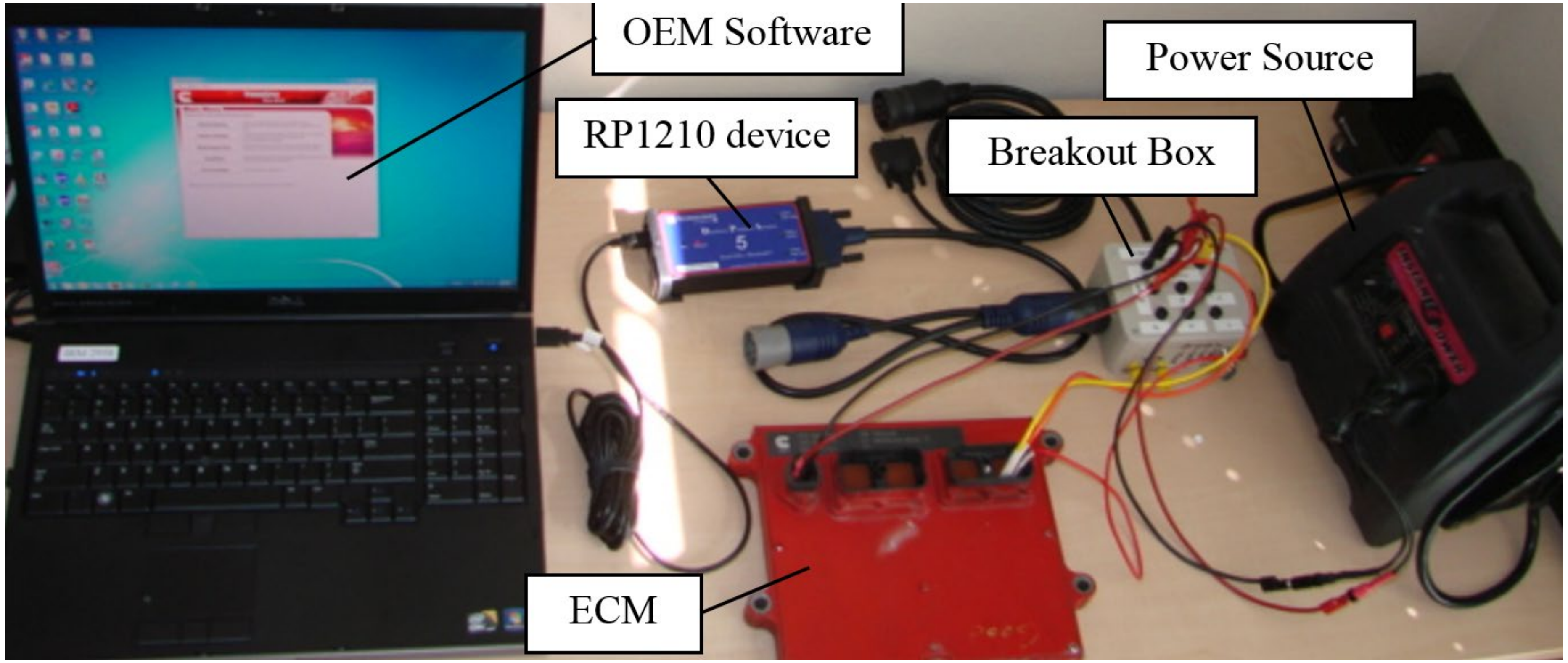
Time	Vehicle Speed (mph)	Engine Speed (rpm)	Brake	Clutch	Engine Load (%)	Throttle (%)	Cruise	Diag. Code
-0:59	23.5	0	No	No	0.00	0.00	No	Yes
-0:58	22.0	0	No	No	0.00	0.00	No	Yes
-0:57	20.0	0	No	No	0.00	0.00	No	Yes
-0:56	18.0	0	No	No	0.00	0.00	No	Yes
-0:55	16.0	0	No	No	0.00	0.00	No	Yes
-0:54	14.0	0	No	No	0.00	0.00	No	Yes
-0:53	12.0	0	No	No	0.00	0.00	No	Yes
-0:52	10.0	0	No	No	0.00	0.00	No	Yes
-0:51	8.0	0	No	No	0.00	0.00	No	Yes
-0:50	6.5	0	No	No	0.00	0.00	No	Yes
-0:49	4.0	0	No	No	0.00	0.00	No	Yes
-0:48	2.5	0	No	No	0.00	0.00	No	Yes
-0:47	1.0	0	No	No	0.00	0.00	No	Yes

A direct approach may be needed

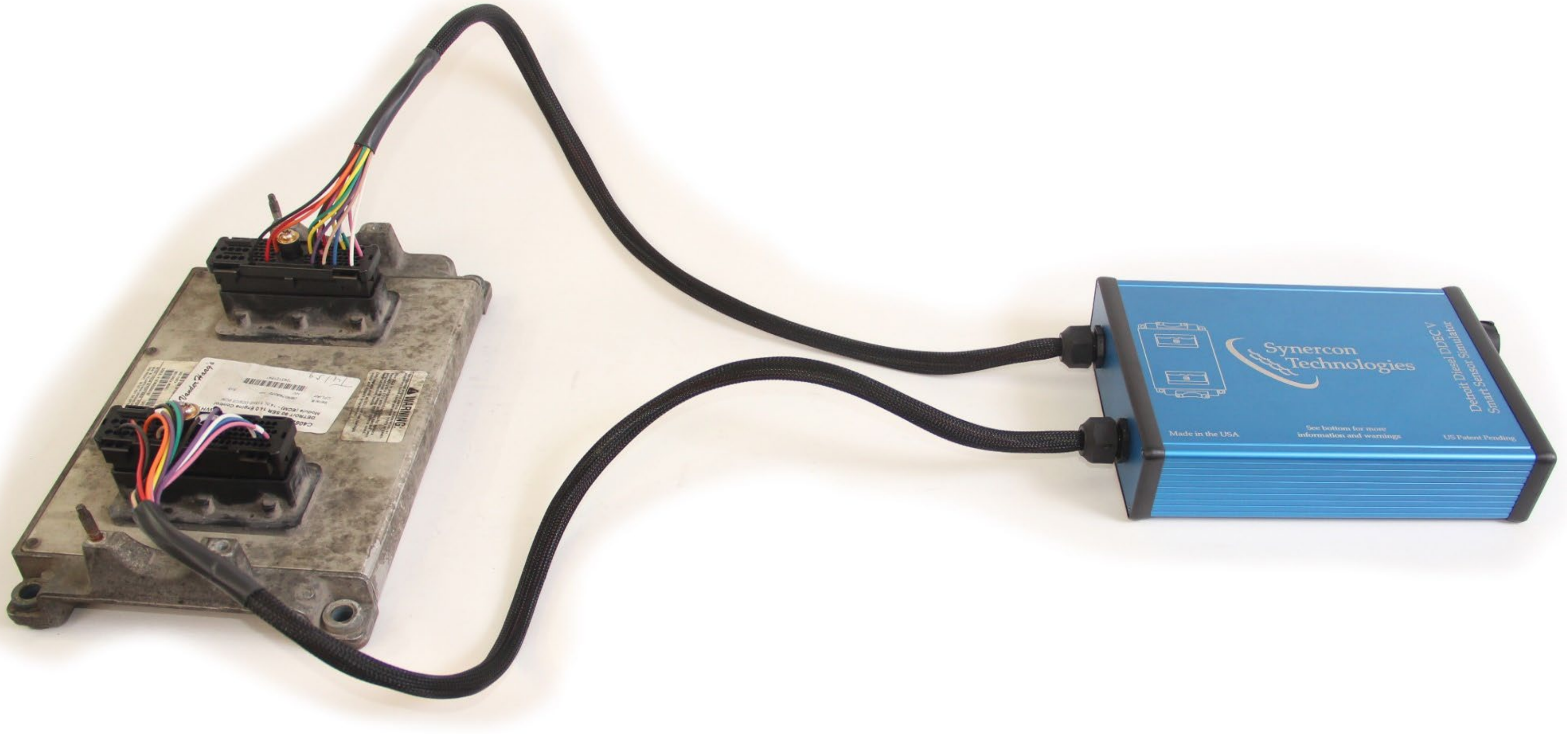
- The electrical system is compromised.

Bench Top Download (or Image?)

- But this sets new faults.

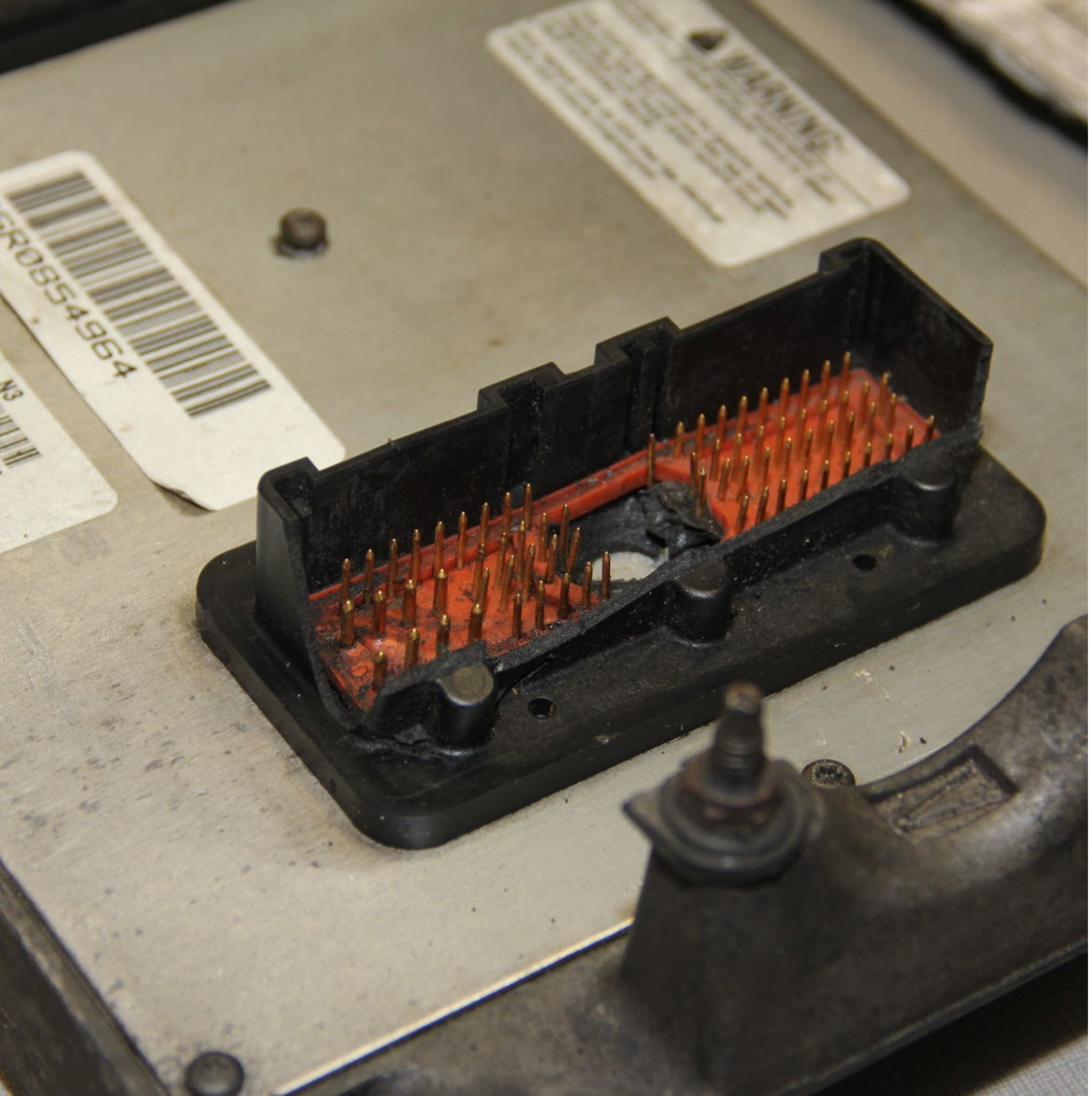


Bench Top Download (Fault Free)



But, sometimes it’s not that easy.

The electrical system is compromised.

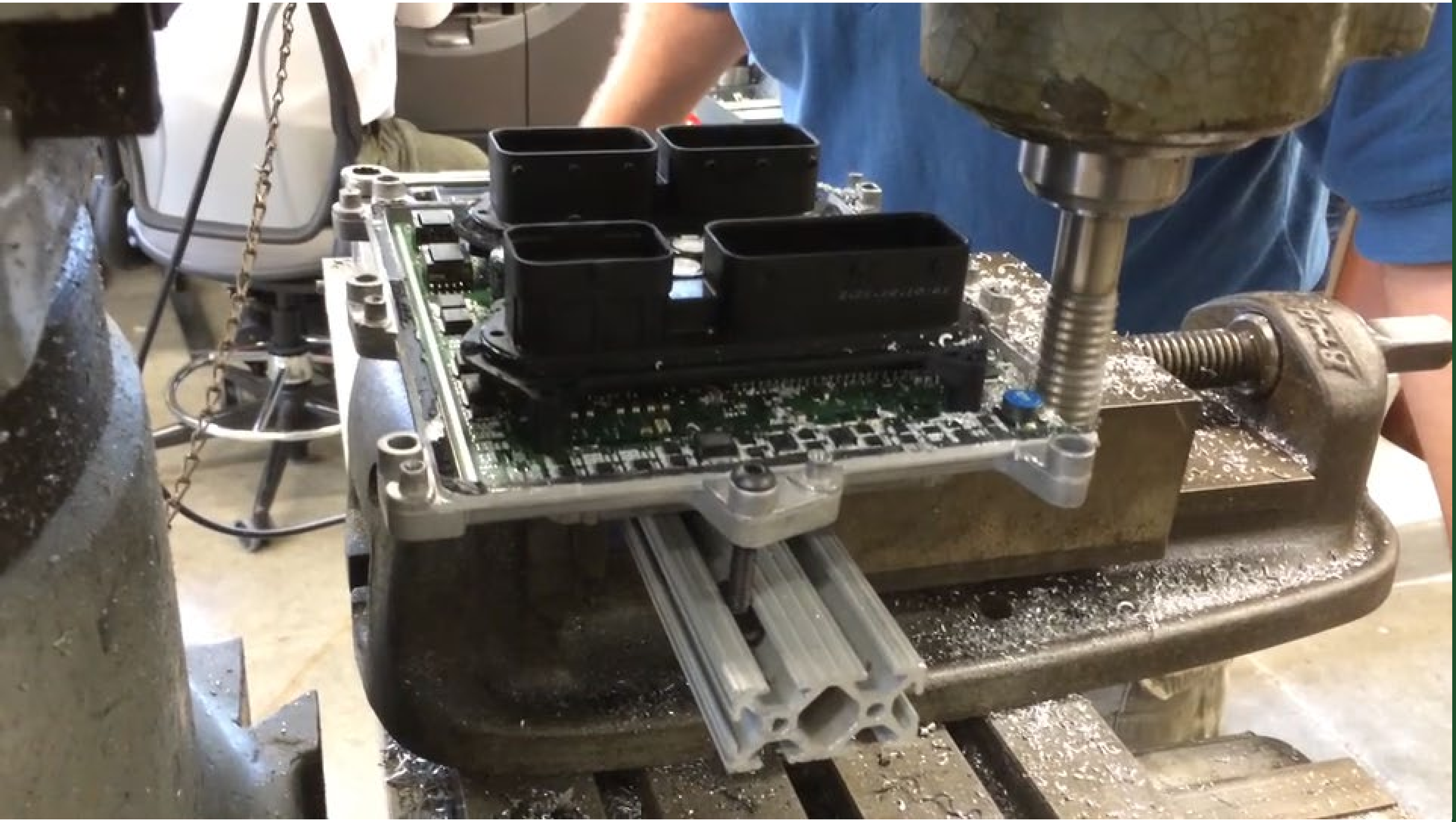


Attempted Download

- Able to connect, but throws a J1708 Network Error
- This isn’t covered in the manual…
- Let’s take a peek inside the module.

Gaining Chip Access

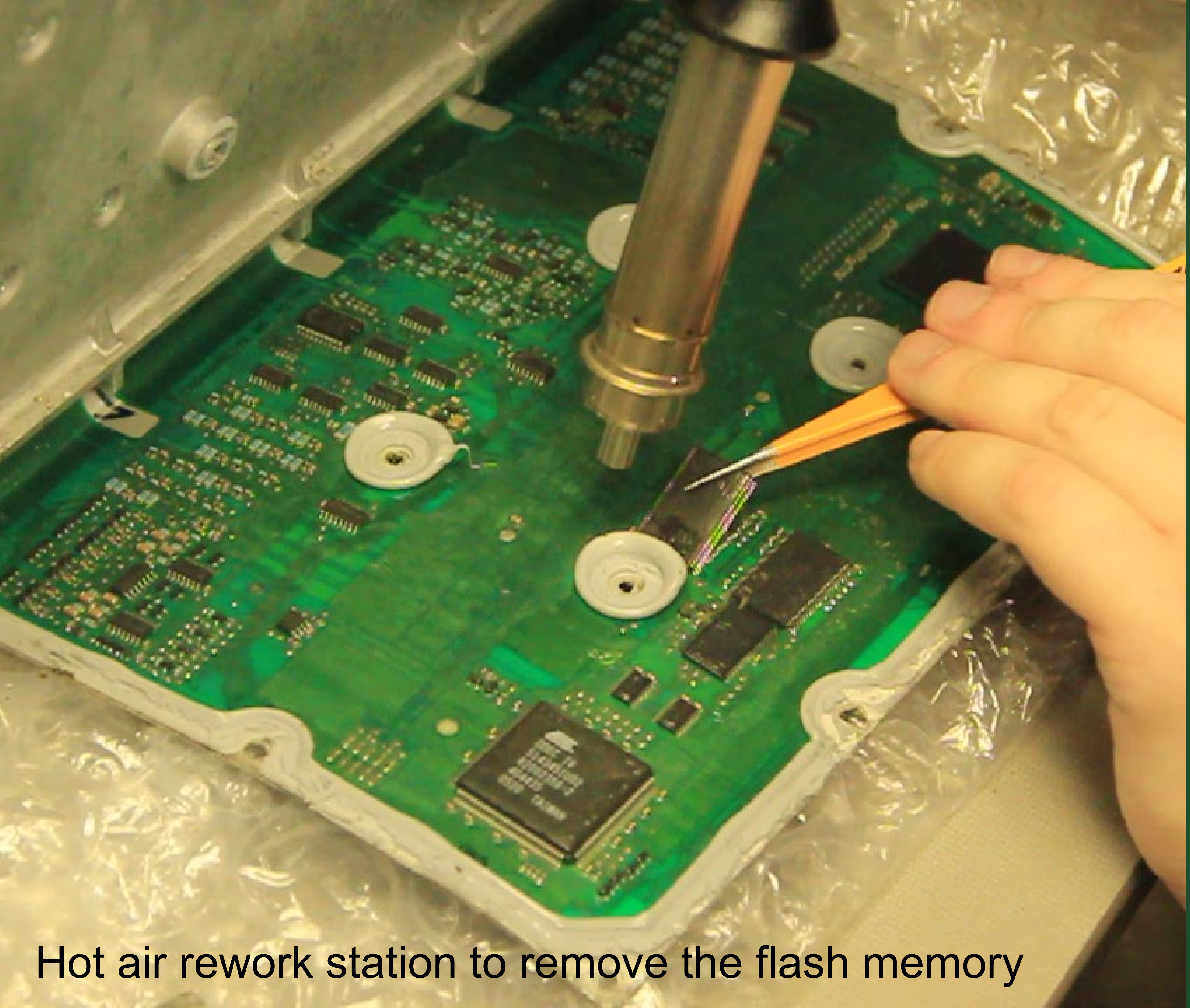
- Accessing the chips with a vise and brute force…



Gaining Chip Access

with a milling machine…

Chip Removal


Hot air rework station to remove the flash memory

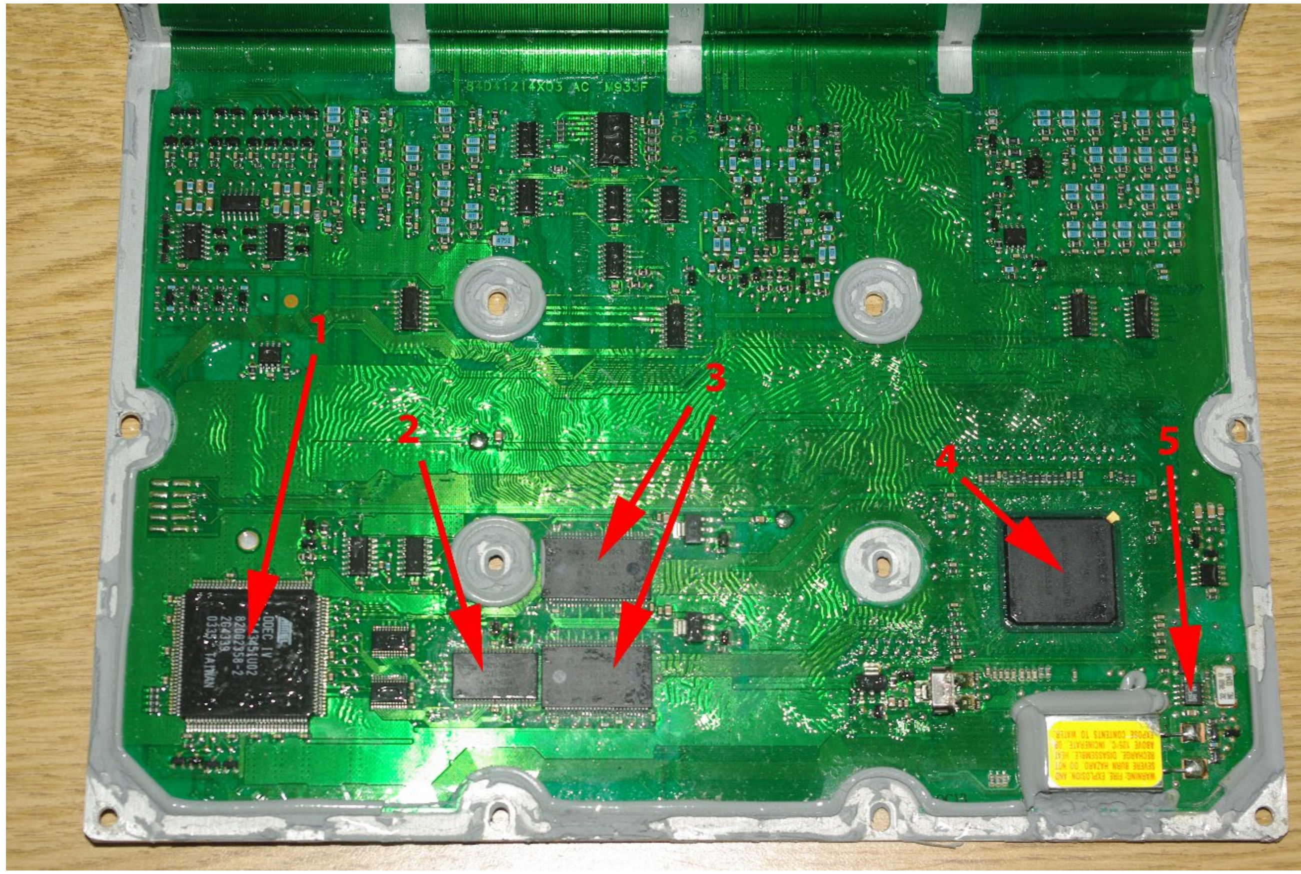
Crash Induced Access

The enclosure is in pieces

Chip Identification

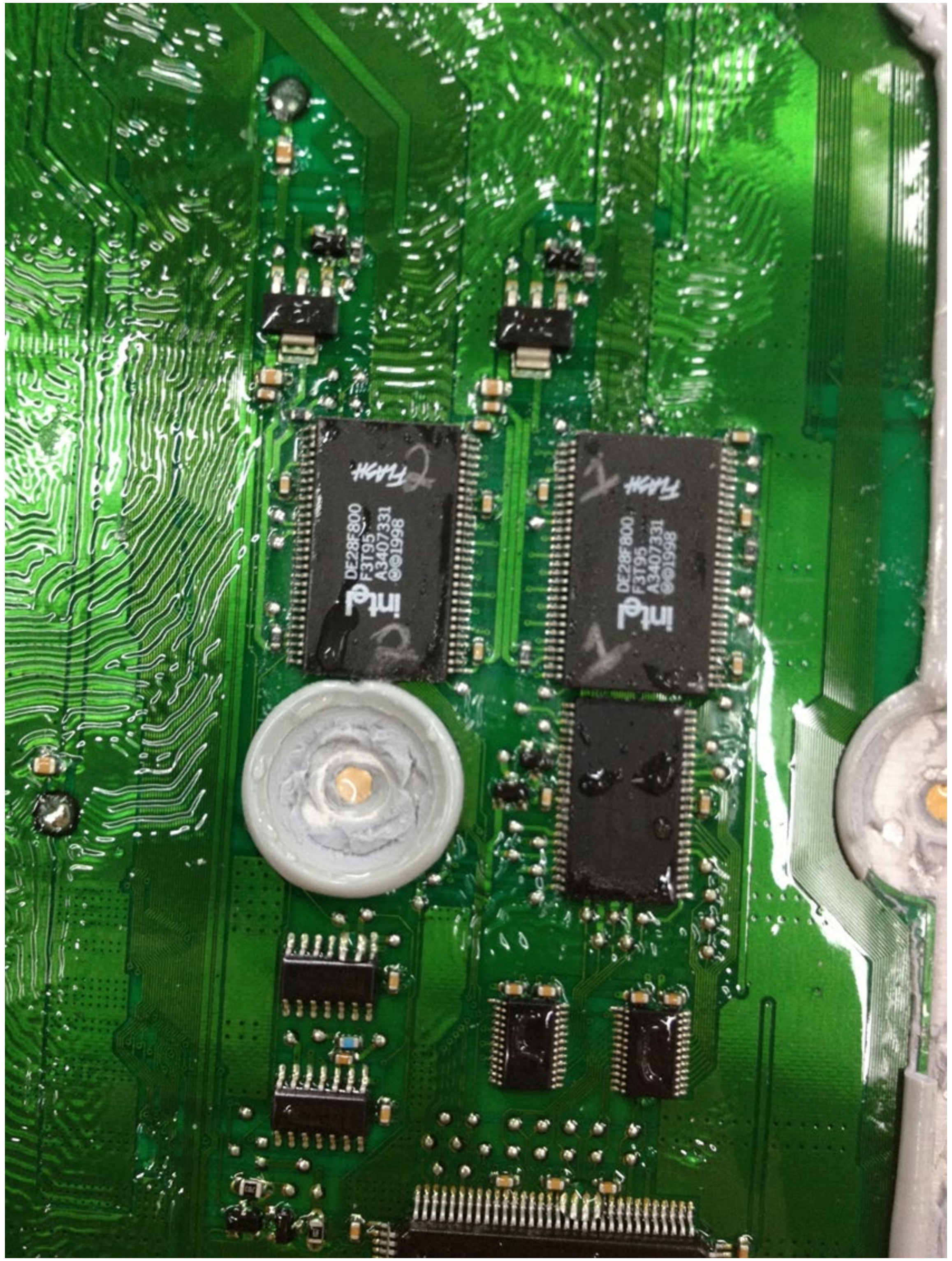
- DDEC 5

1. Custom ASIC – similar to later DDEC4
2. Cypress CY62137VLL SRAM
3. AMD AM29BL802CB Flash Storage ICs
4. MPC555LF8MZP40 32-bit CPU
5. Real-time clock IC EM V3020



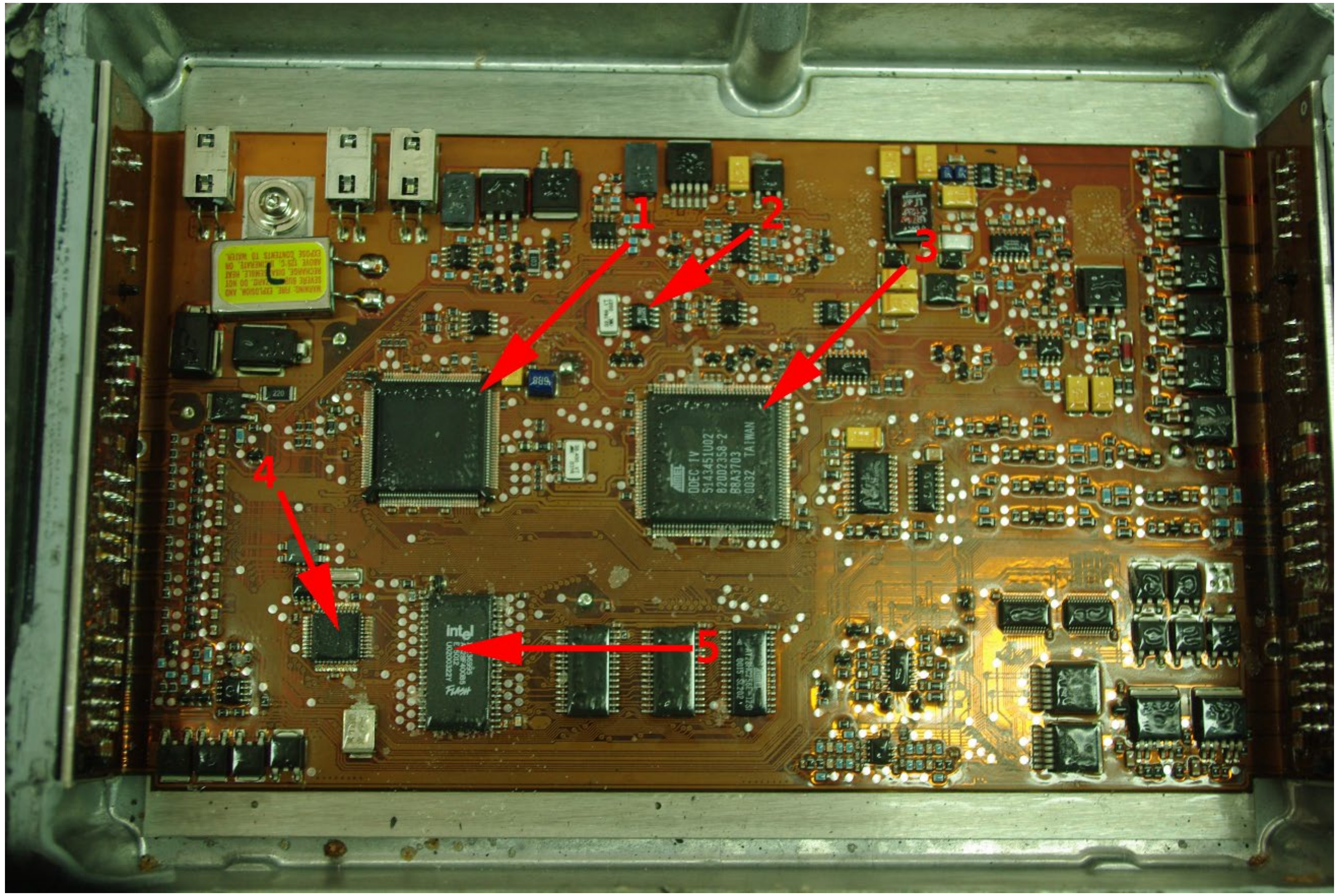
Another DDEC V

- Data is stored on flash memory.
- This DDEC5 used an Intel chip.
- Each chip stores 1 megabyte



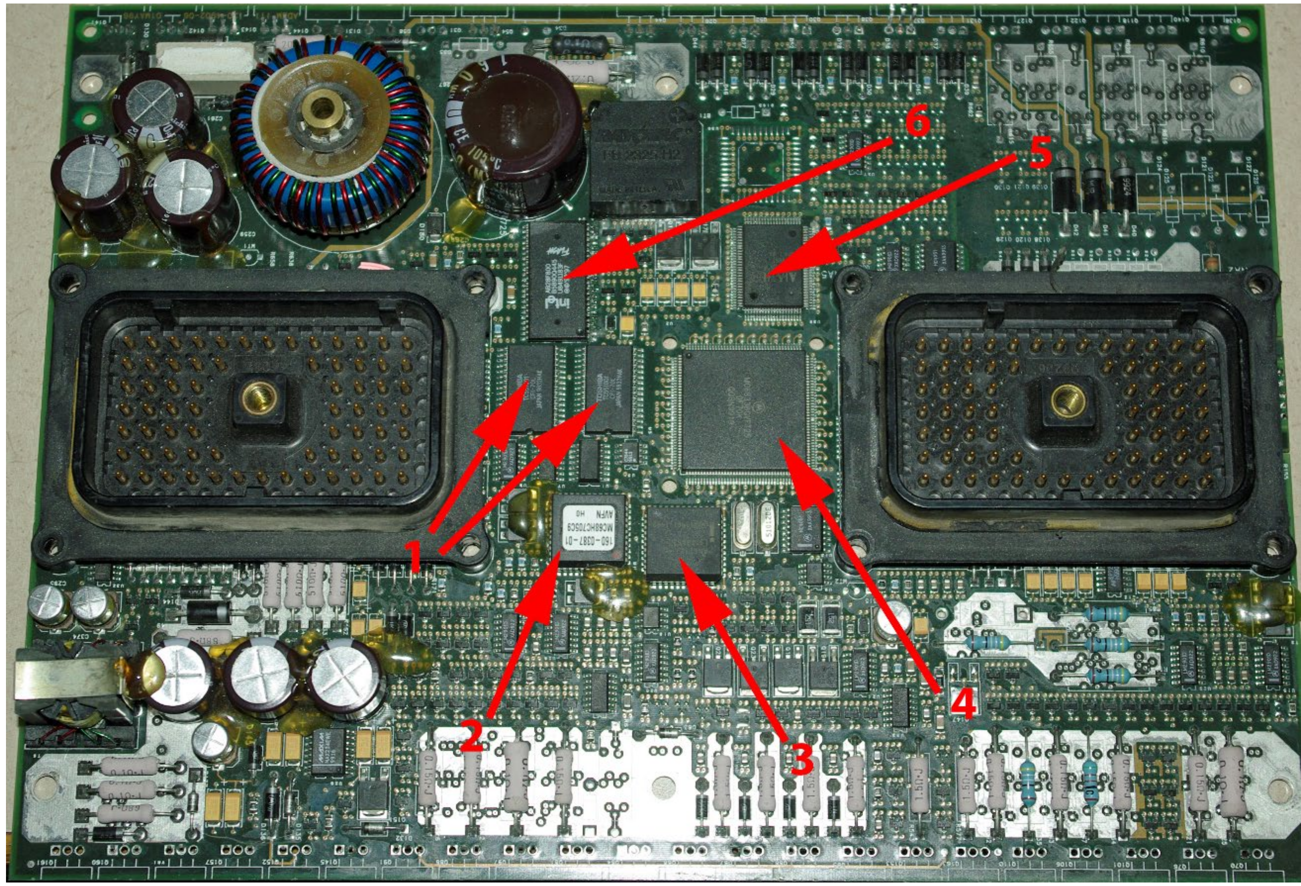
DDEC IV Chip Identification

1. MC68332 – 32-bit CPU
2. Real-time Clock controller
3. Presumed Custom ASIC controller
4. CAN Controller
5. Intel Flash Storage IC AB28F400

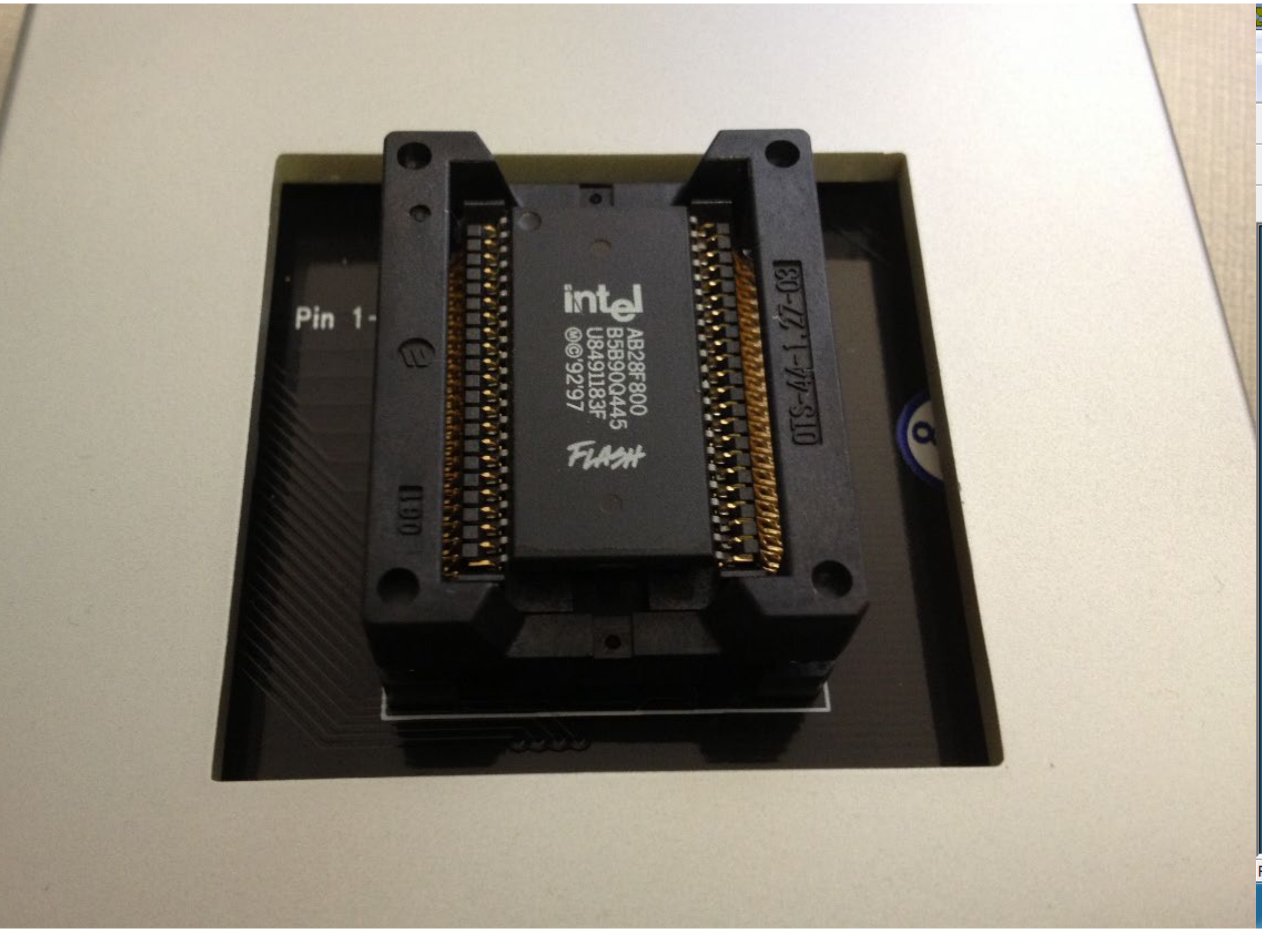


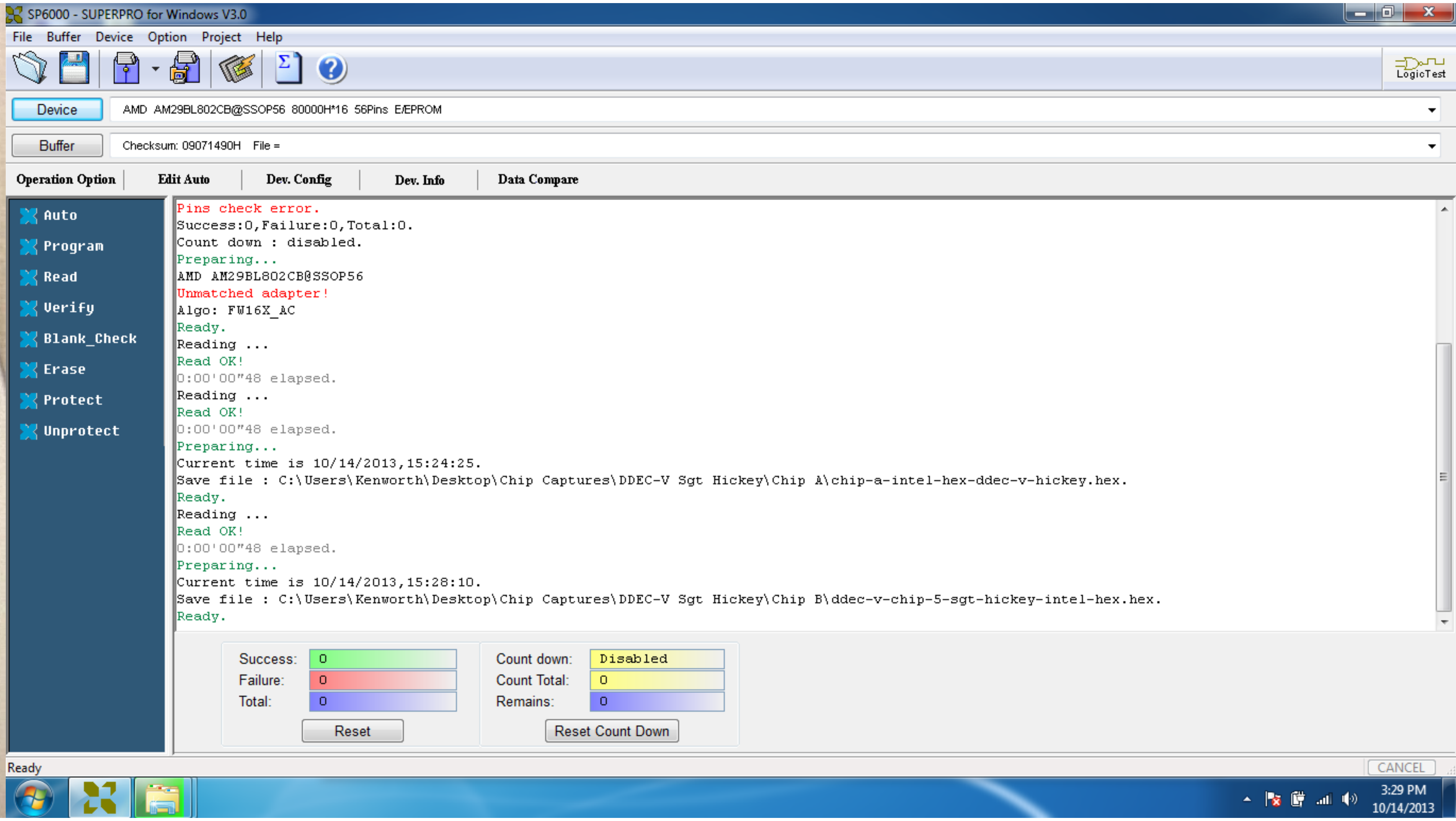
CAT ADEM III Chip Identification

1. Toshiba SRAM
2. MC68HC705C9A 8-bit Microcontroller (EEPROM)
3. Intel CAN 2.0 Controller
4. MC68336 32-bit Microprocessor (note: Mask-ROM + SRAM)
5. AMI IC Branded Caterpillar, Presumed ASIC
6. Intel AB28F800 5V Flash Storage

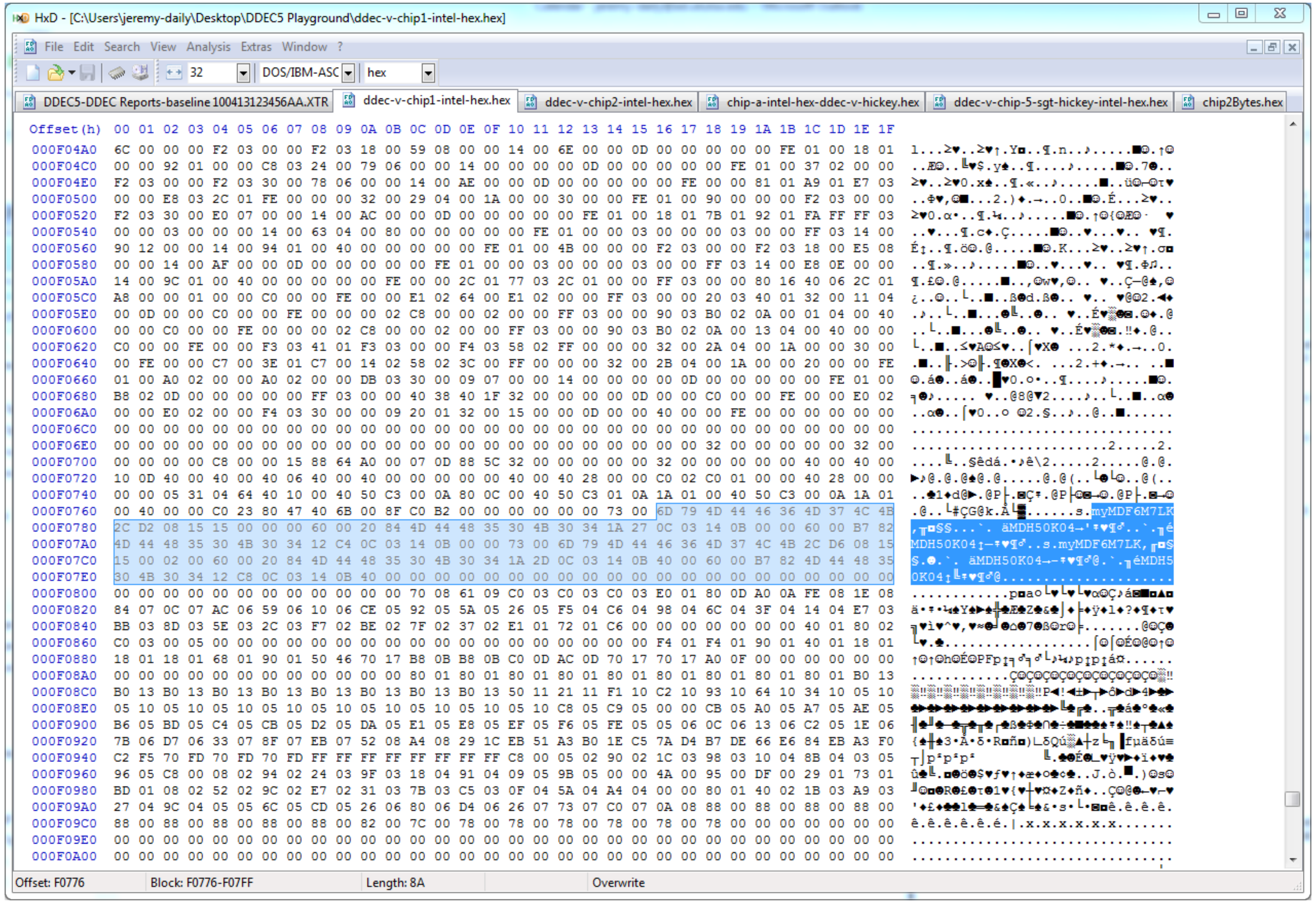


Reading Memory Chip Contents





Results in a Hex Editor



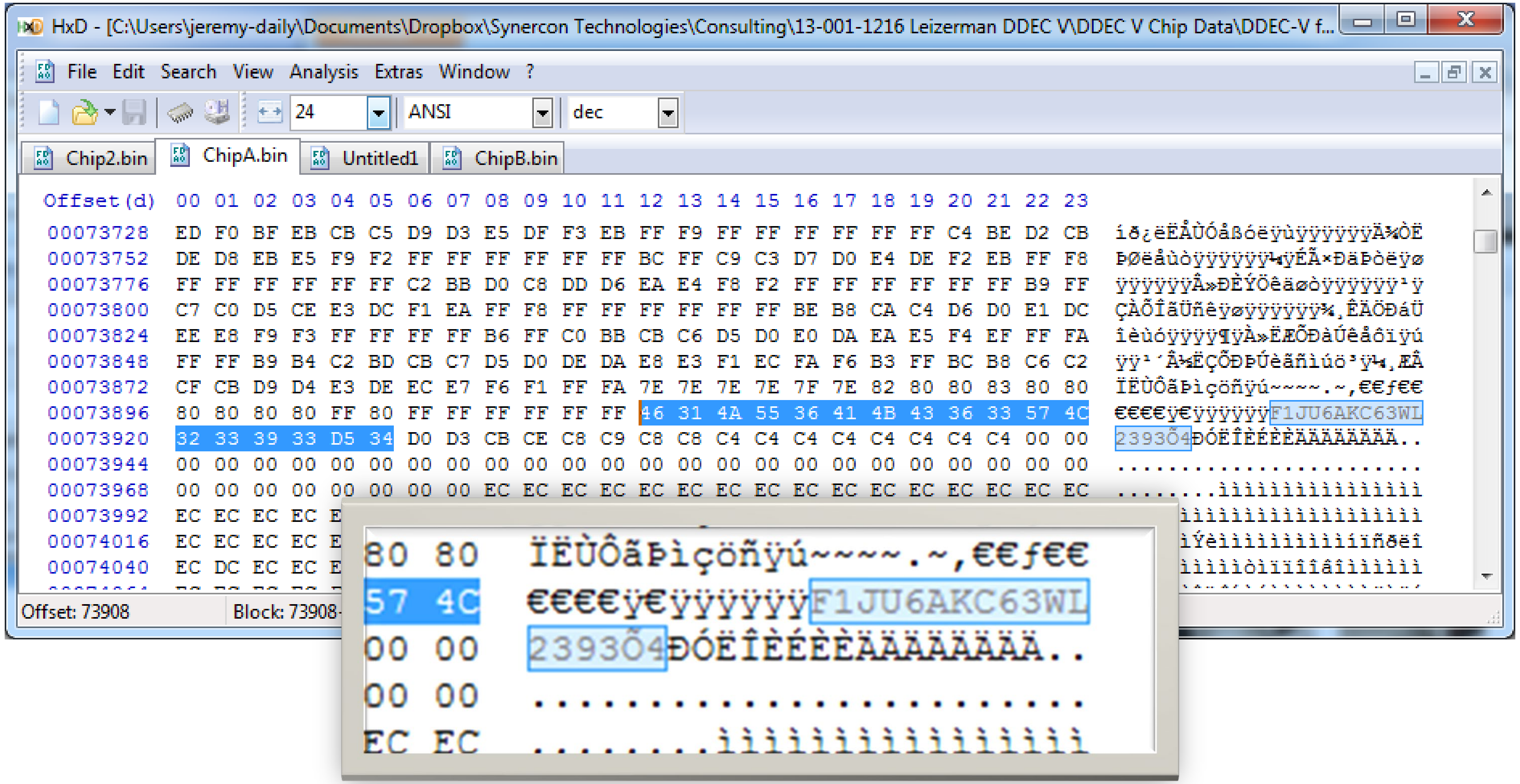
Understanding Hex Data as Powers of 2

Bit position	8	7	6	5	4	3	2	1	Notes
Exponent	7	6	5	4	3	2	1	0	1 less than position
2^Exponent	128	64	32	16	8	4	2	1	Value of the bits
Bits	1	0	0	1	1	1	0	0	Example
Bit values	128	0	0	16	8	4	0	0	156
Nibbles	9				C				Use letters for numbers > 9
Byte (hex)	9C								Concatenate Nibbles
Decimal	156								

- Hex is base 16 condensed representation of binary (base 2)
- Uses 0-9, A-F to get 16 characters
- Each character is a nibble (4-bits), 2 nibbles is a byte (8-bits)
- All data in computers and networks are represented as binary (1 or 0)

Human Readable Data

Letters and numbers are encoded using ASCII. Look for known ASCII, like VIN and Serial Number.



Hex Data…

BOSCH

CRASH DATA RETRIEVAL

Hexadecimal Data

```
B600:  20 50 08 00 00 00 00 00
B608:  00 00 00 00 AA 00 00 00
B610:  00 00 00 00 AA 45 F9 F9
B618:  F9 F9 F9 9D F9 F9 FF AA
B620:  AA AA 00 00 00 00 00 00
B628:  00 00 00 00 00 00 00 00
B630:  00 00 00 00 00 00 00 00
B638:  00 00 00 00 00 00 00 00
B640:  00 00 00 00 00 00 00 00
B648:  00 00 00 00 00 00 00 00
B650:  00 00 00 00 00 00 00 AA
B658:  00 25 82 20 50 08 00 00
B660:  00 00 00 00 00 00 00 00
B668:  00 00 00 00 00 00 00 00
B670:  00 00 00 00 00 00 00 00
B678:  00 00 00 00 00 00 43 69
B680:  00 00 00 55 AA AA AA AA
B688:  00 00 00 BE 86 00 01 16
B690:  87 00 01 6C 88 00 01 7D
```

Manufacturers specify what the hex data means

Sometimes, manufacturers use standards for the meaning of data

Reverse engineering processes can help decode non-standardized data

Meaning Applied to HVEDRs

- Standards Based Meaning
- SAE J1587

A.84 ROAD SPEED

Indicated vehicle velocity.

Parameter Data Length: 1 Character
Data Type: Unsigned Short Integer
Bit Resolution: 0.805 km/h (0.5 mph)
Maximum Range: 0.0 to 205.2 km/h (0.0 to 127.5 mph)
Transmission Update Period: 0.1 s
Message Priority: 1
Format:

PID	Data
84	a
a—	Road speed

SAE International™ SURFACE VEHICLE RECOMMENDED PRACTICE

SAE J1587 JUL2008

Issued 1988-01
Revised 2008-07

Superseding J1587 FEB2002

Electronic Data Interchange Between Microcomputer Systems in Heavy-Duty Vehicle Applications

- SAE J1939-71
- SAE J1939-73

SAE International™ SURFACE VEHICLE RECOMMENDED PRACTICE

SAE J1939-71 FEB2010

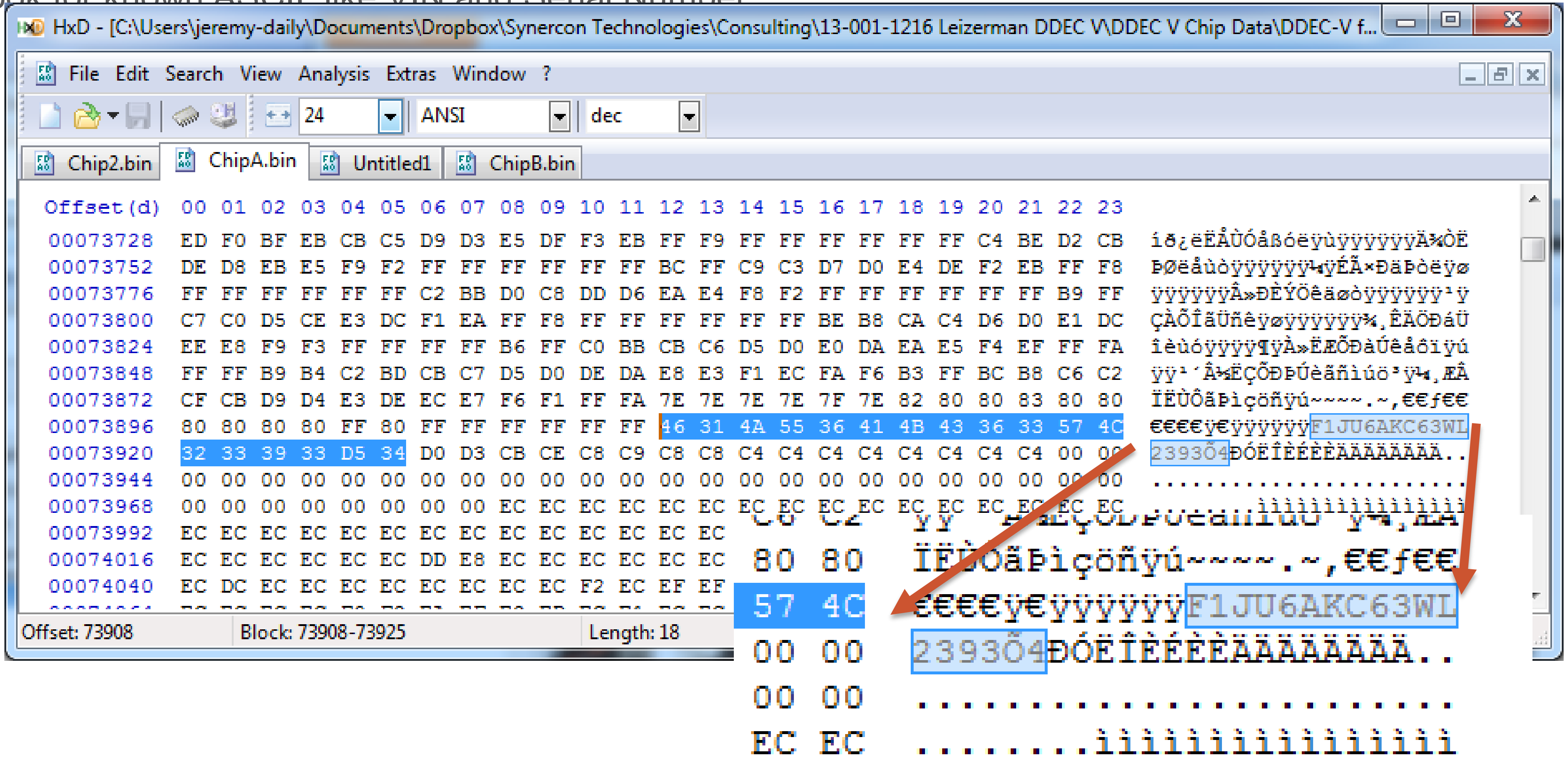
Issued 1994-08
Revised 2010-02

Superseding J1939-71 JAN2009

Vehicle Application Layer (Through February 2009)

Human Readable Hex

- Letters and numbers are encoded using ASCII.
- Strategy: Look for known ASCII like VIN and Serial Number



2 Byte Reversals

- The flash memory is used such that the bytes are stored with bytes that are reversed.
  - The VIN from the raw memory says:
  - F1 JU 6A KC 63 WL 23 93 ◊4
- After swapping every 2 bytes, it becomes:

  1FUJA6CK36LW32394

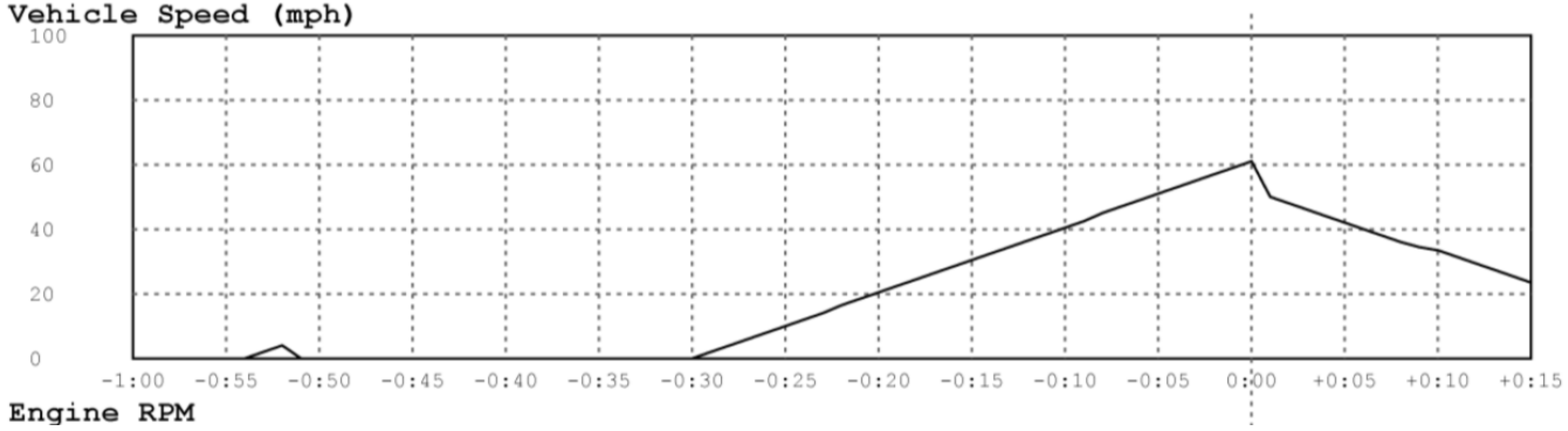
- This is 18 bytes, but VINs are 17 characters
- We can also find serial numbers (search for "R6")

Simulated Data

- Issue: Still need to decode the data…
- Strategy: Get an exemplar ECM and put a known speed record on it to find the Hard Brake and Last Stop Events.

DDEC® Reports - Hard Brake #1

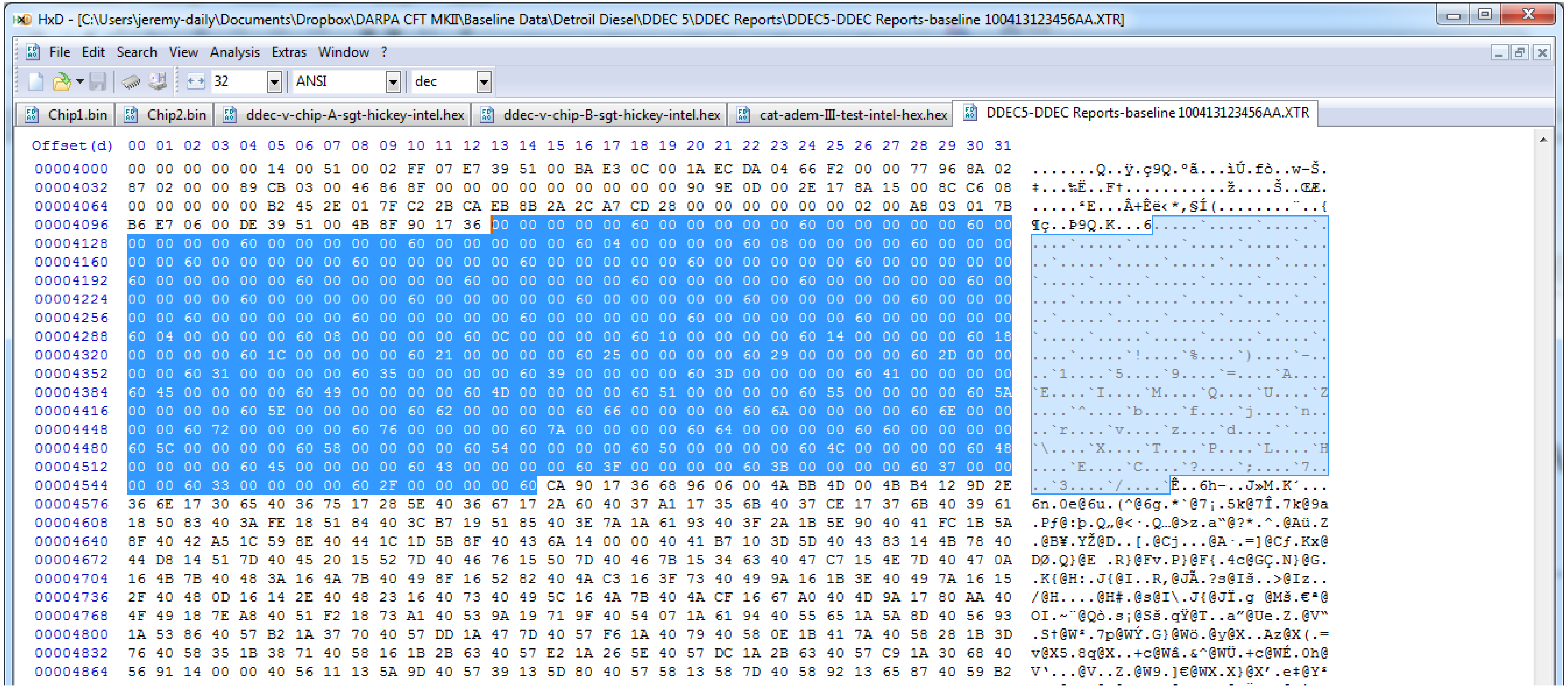
Print Date: 10/4/2013 1:23 PM		Trip:	12/12/05 20:56:39 To 10/04/13 (PST)
DDC		Vehicle ID:	DDEC5-TEST
		Driver ID:	
, -		Odometer:	532323.9 mi
( ) -		Engine S/N:	06R0760090
Trip Distance	473875.7 mi	Trip Time	20869:22:45
Trip Fuel	94635.50 gal	Fuel Consumption	4.53 gal/h
Fuel Economy	5.01 mpg	Idle Time	11330:35:08
Avg Drive Load	46 %	Idle Percent	54.29 %
Avg Vehicle Speed	49.7 mph	Idle Fuel	7417.38 gal
Incident Time: 10/04/13 7:14:18 (PST)		Incident Odometer:	532323.0 mi



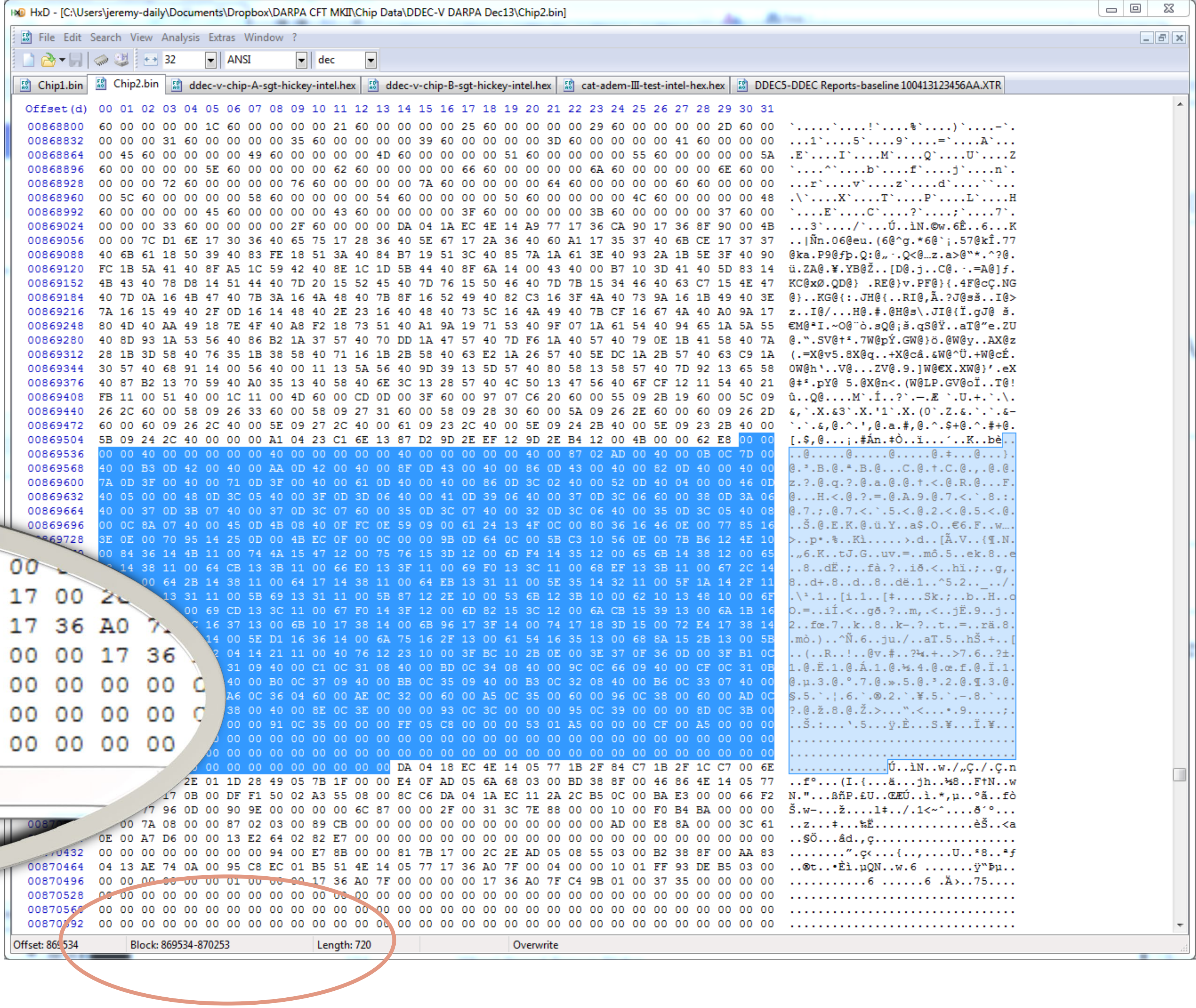
Get help from the network logs

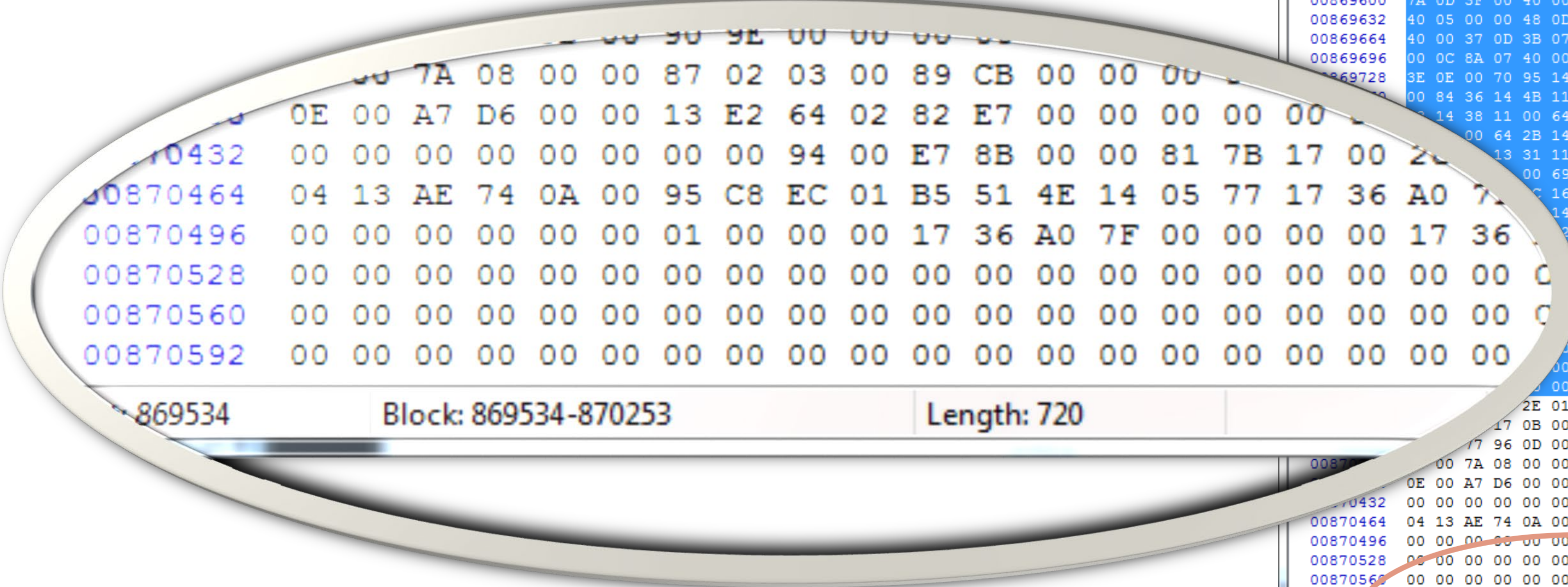
- DDEC Reports downloads data in 9 groups called data pages.
- Use J1587 Transport layer to reconstruct the network traffic.
- *.XTR file is close to a network log.
- We can map the XTR file contents to DDEC Reports elements (See SAE 2014-01-0495)
- Enables pattern matching for data elements like Mileage and Times.

Find the patterns (Hard Brake)

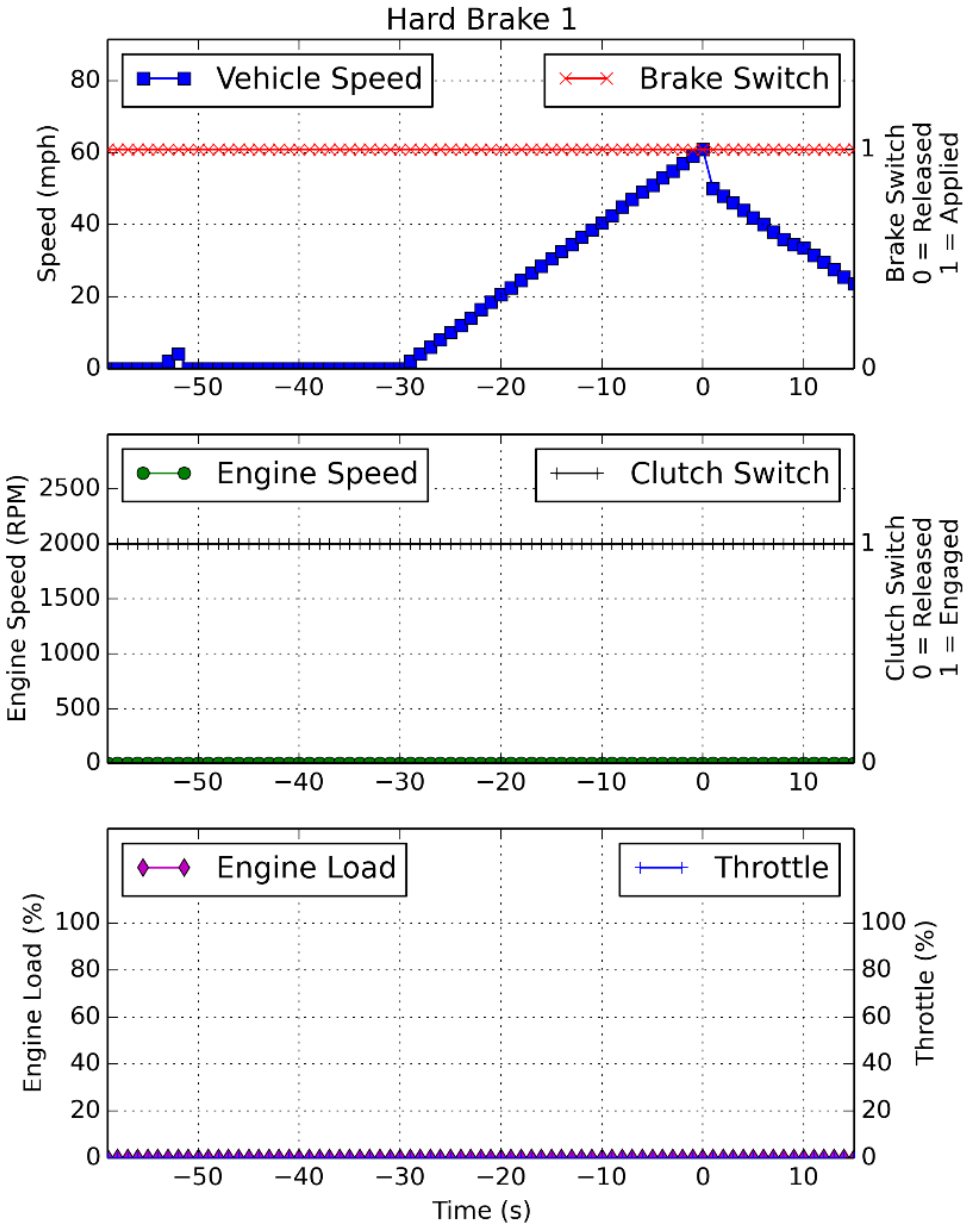


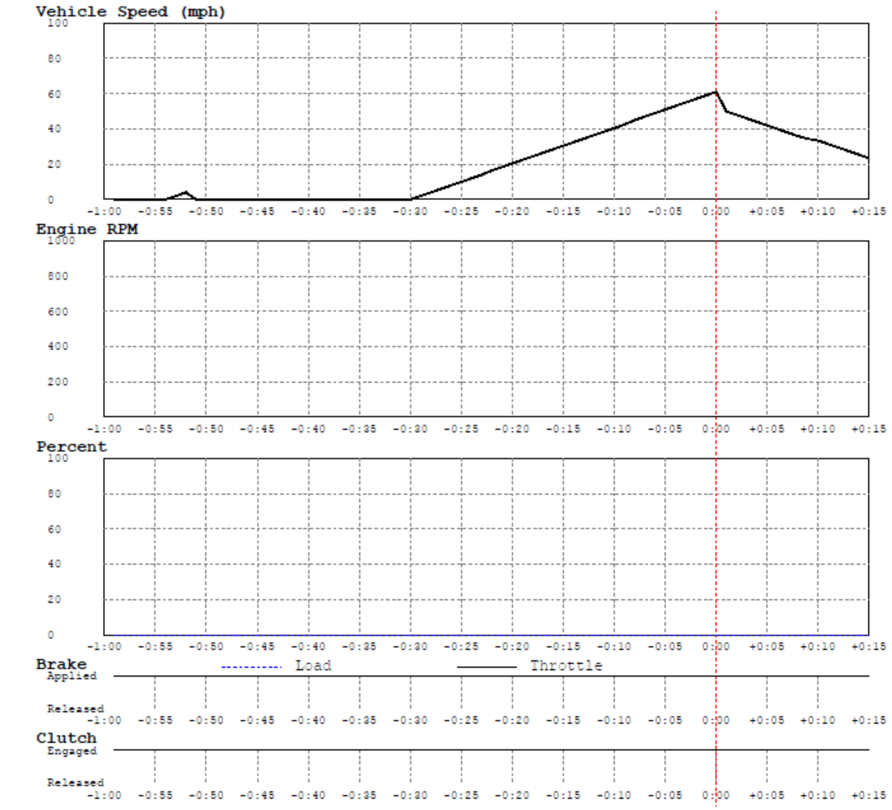
Last Stop Data





Hard Brake 1 Comparison





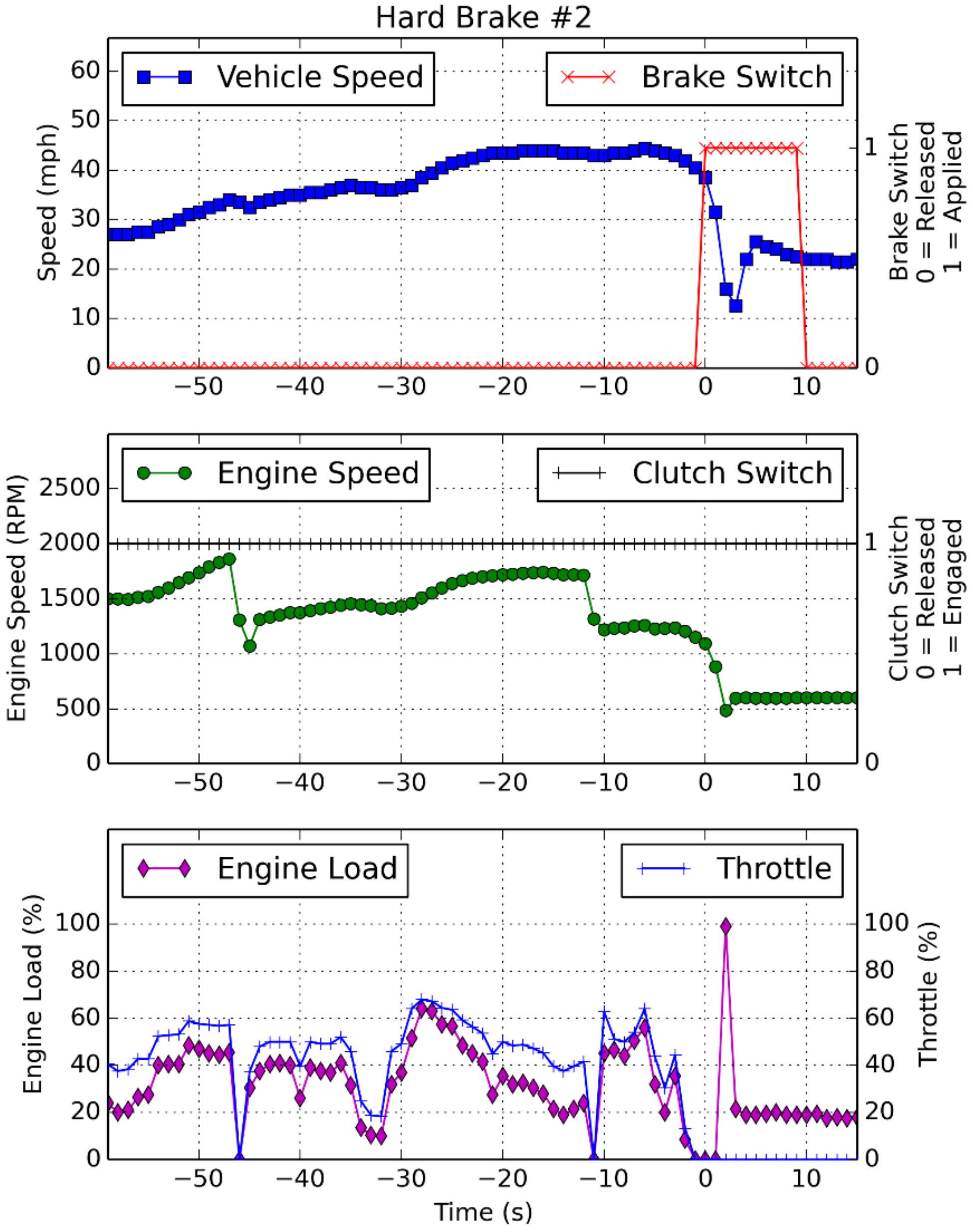
DDEC® Reports - Hard Brake #1

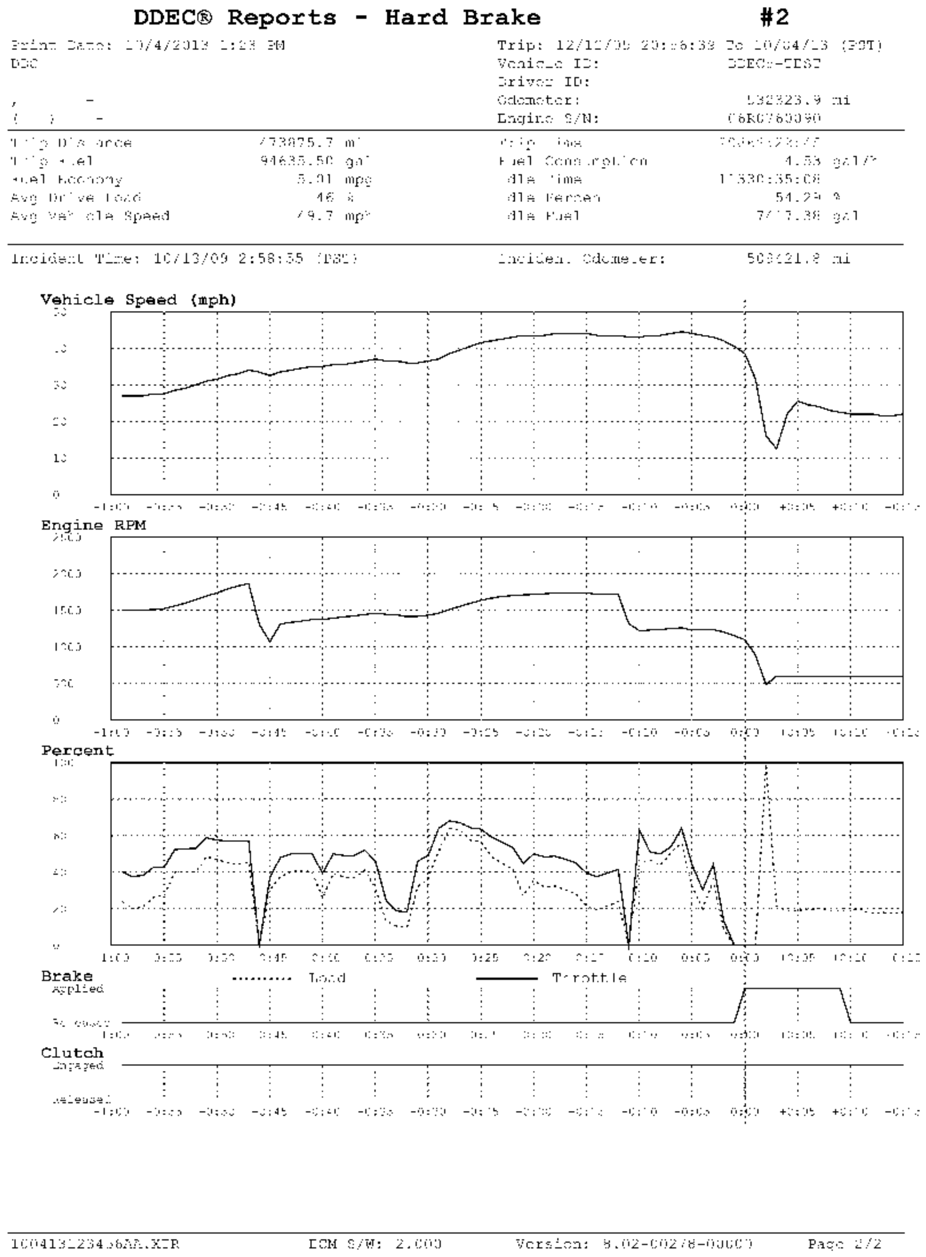
Print Date: 10/4/2013 1:23 PM		Trip: 12/12/05 20:56:39 To 10/04/13 (PST)	
DDC		Vehicle ID:	DDEC5-TEST
		Driver ID:	
, -		Odometer:	532323.9 mi
( ) -		Engine S/N:	06R0760090
Trip Distance	473875.7 mi	Trip Time	20869:22:45
Trip Fuel	94635.50 gal	Fuel Consumption	4.53 gal/h
Fuel Economy	5.01 mpg	Idle Time	11330:35:08
Avg Drive Load	46 %	Idle Percent	54.29 %
Avg Vehicle Speed	49.7 mph	Idle Fuel	7417.38 gal

Incident Time: 10/04/13 7:14:18 (PST) Incident Odometer: 532323.0 mi

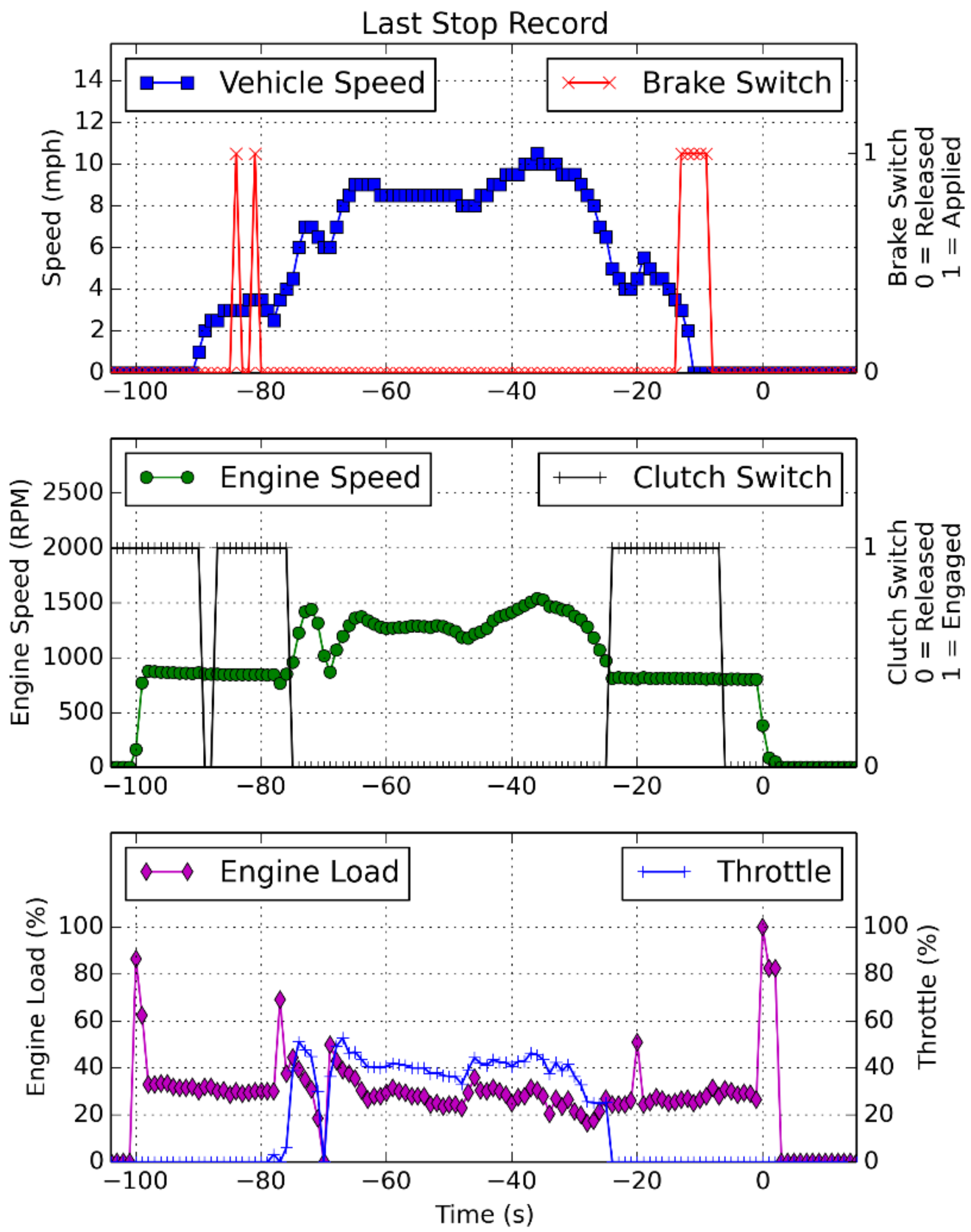
100413123456AA.XTR ECM S/W: 2.000 Version: 8.02-00278-00000 Page 1/2

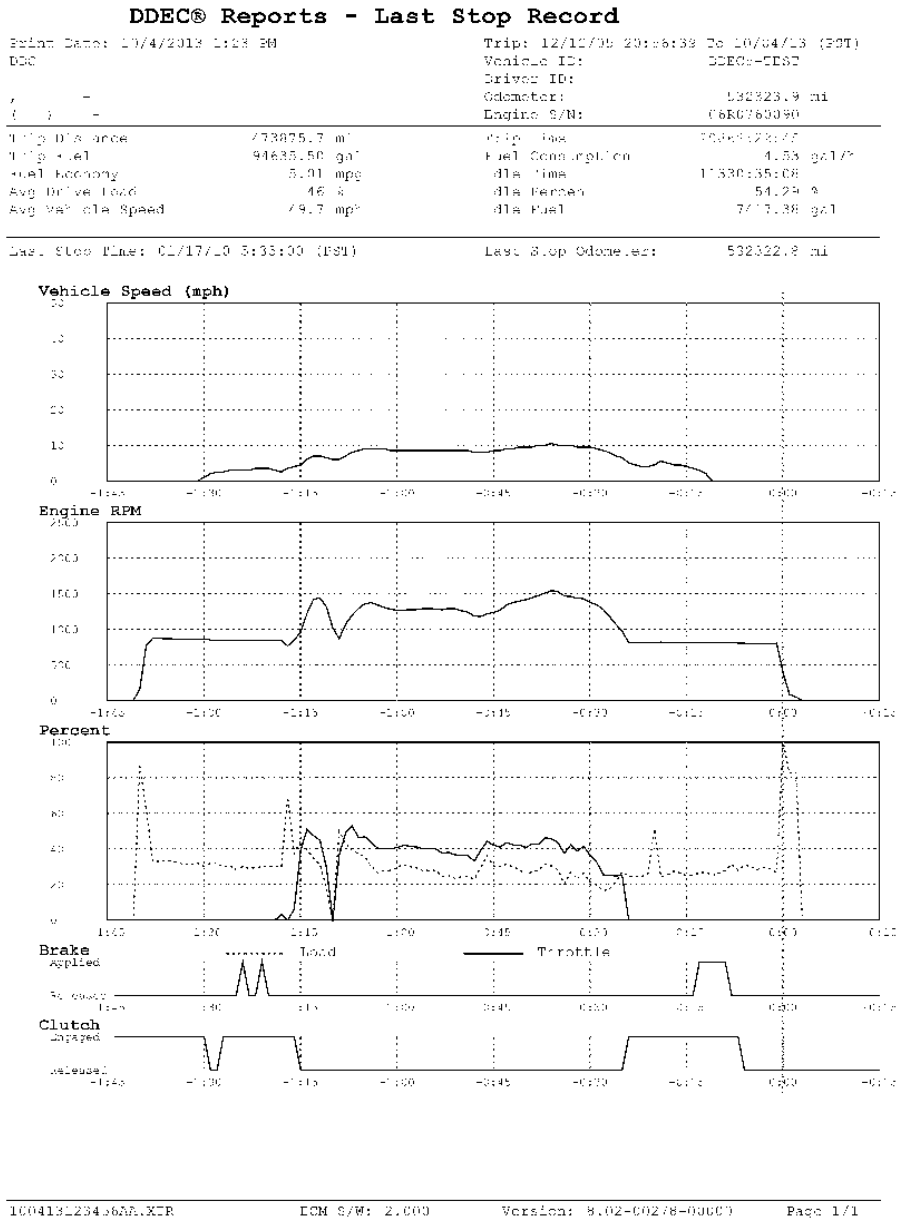
Hard Brake 2 Comparison





Last Stop Comparison





Daily Engine Usage

DDEC® Reports - Daily Engine Usage

Print Date: 8/21/2013 11:08 AM
University of Tulsa
800 S. Tucker Dr
Tulsa, OK 74104
(918)631-3056

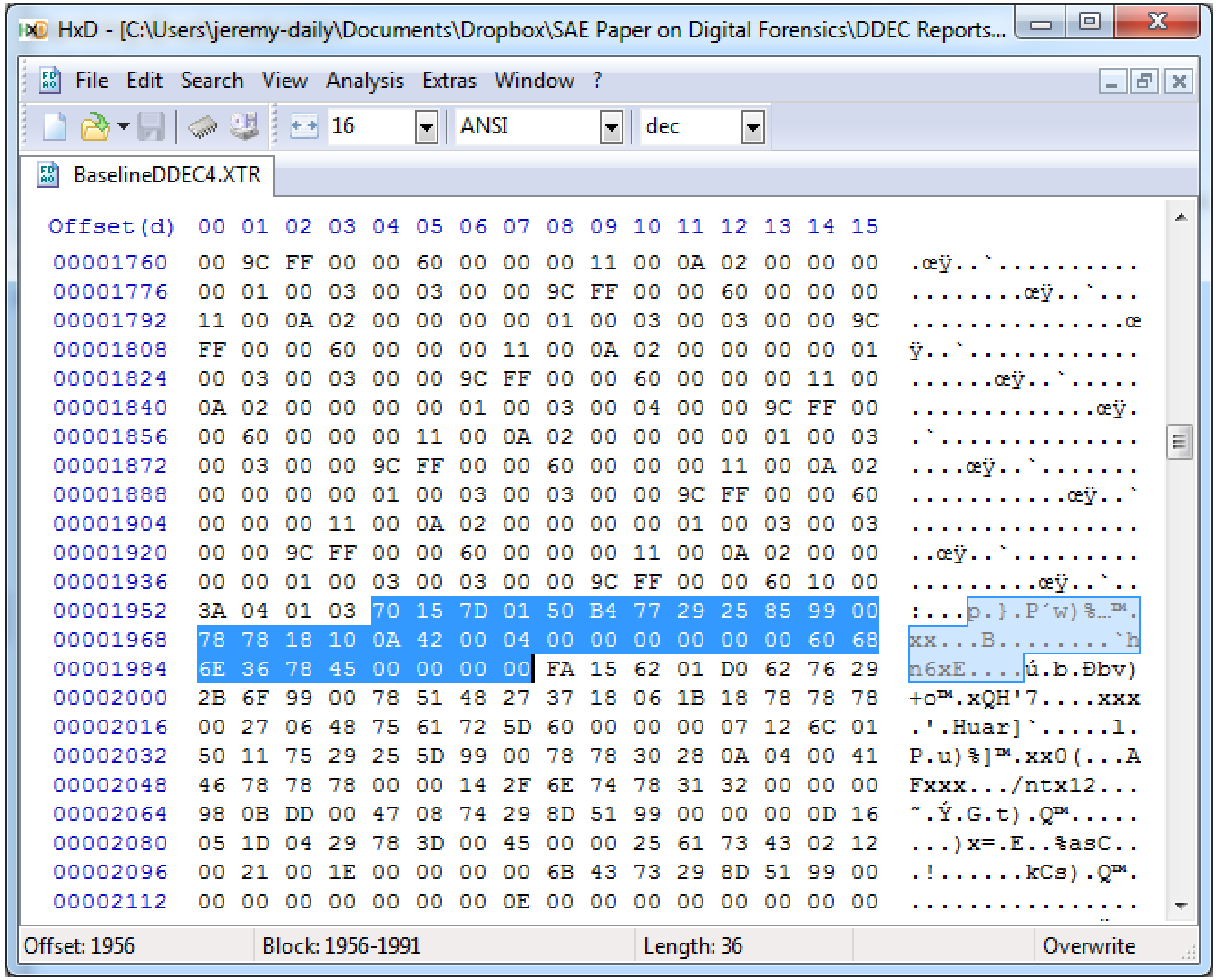
Date Range: 01/18/07 To 01/07/00 (EST)

Vehicle ID: TIB DDEC4
Driver ID:
Engine S/N: 06R0499534

Date:	1/18/2007	
Start Time:	00:00:00	EST
Odometer:	1006109.00	mi
Distance:	548.80	mi
Fuel:	95.25	gal
Fuel Economy:	5.76	mpg
Average Speed:	59.54	mph

Total (hh:mm)	09:13	06:00	08:47
Hour (EST)	Drive (min)	Idle (min)	Off (min)
00:00-02:00	0	120	0
02:00-04:00	0	120	0
04:00-06:00	96	24	0
06:00-08:00	104	16	0
08:00-10:00	110	10	0
10:00-12:00	54	66	0
12:00-14:00	120	0	0
14:00-16:00	69	4	47
16:00-18:00	0	0	120
18:00-20:00	0	0	120
20:00-22:00	0	0	120
22:00-24:00	0	0	120

Daily Engine Usage Log Data - .XTR file



Determining Data Meaning in the Daily Engine Usage Log

Interpreted Data

Bytes Sequence	Hex Value (s)	Decimal	LSB Value	Meaning	Value
0-1	70 15	5488	0.1 mile	Distance	548.8 miles
2-3	7D 01	381	0.25 gal	Fuel	95.25 gallons
4-7	50 B4 77 29	695710800	1 sec from epoch	Start Time	17 Jan 2007 at 23:00:00 CST
8-11	25 85 99 00	10061093	0.1 mile	Odometer	1006109.3 miles
12-23	78 78 18 10 0A 42 00 04 00 00 00 00	120 120 24 16 10 66 0 4 0 0 0 0	1 Minute	Idle Time	Same as Decimal
24-35	00 00 60 68 6E 36 78 45 00 00 00 00	0 0 96 104 54 120 69 0 0 0 0	1 Minute	Drive Time	Same as Decimal

All other data are calculated

Interestingly, the .XTR file contains minutes, but the chip memory contains seconds.

Chip Memory Contents

XTR file has 36 Bytes for 1 day in the Daily Engine Usage Log.

However… The memory record containing the Daily Engine Usage data is contained in a circular 30-day buffer with each day holding 66 bytes.

This was determined by locating the odometer readings since the MSB's were the same. There were 66 bytes from one 4-byte odometer reading to another.

Data Description	Unit	Location and sequence	Word Size (LSB last)	LSB Value	Example
Start Time Stamp	Seconds	1, 0, 3, 2	U32	1	Figure 16
Odometer	Miles	5, 4, 7, 6	U32	1/640	Figure 17
Distance Traveled	Miles	9, 8, 11, 10	U32	1/640	Figure 18
Fuel Used	Gallons	12, 13	U16	0.125	Figure 19

Daily Engine Usage Time

- XTR file = 24 bytes
- Memory Chips = 48 bytes, so there twice the bytes that are in memory but not transmitted on the network.

- XTR file has minutes coded as single bytes (0-255)

- Memory stores times in seconds as 2 bytes (16 bit) (0-65536)

- Only Drive time and Idle time in each 2-hour block are recorded in memory.

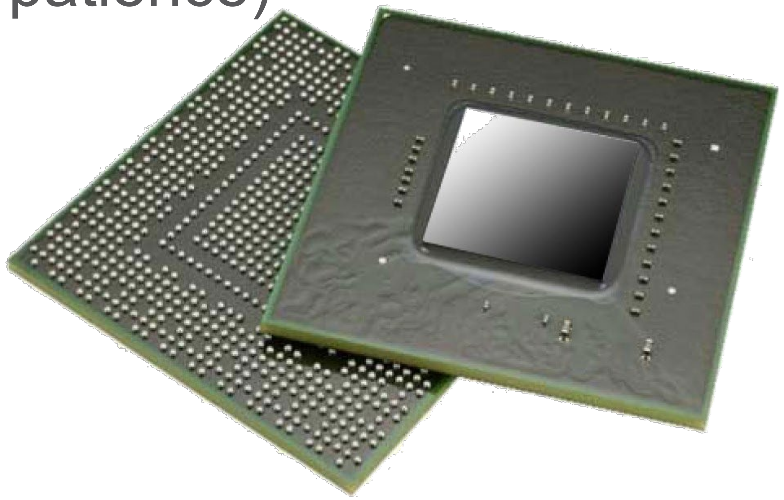
- Drive + Idle seconds in memory contents did not always sum to 7200 seconds (2 hours)

Decoded Daily Engine Usage Log

Start Date	Start Time	Odometer	Distance	Fuel	Total Daily Time		00:00-02:00		02:00-04:00		04:00-06:00		06:00-08:00		…
Central Standard Time		Miles	Miles	Gallons	Idle (HH:MM)	Drive (HH:MM)	Idle	Drive	Idle	Drive	Idle	Drive	Idle	Drive	
Thu, 07 Jan 2010	02:00:00AM	530196.8	346.5	76.750	15:23	08:04	82:33	26:49	65:43	54:17	20:38	99:22	55:49	41:00	
Fri, 08 Jan 2010	02:00:00AM	530543.3	470.0	111.625	13:60	09:58	120:00	00:00	108:47	11:12	00:00	120:00	05:12	114:48	
Sat, 09 Jan 2010	02:00:00AM	531013.3	506.1	111.750	13:57	09:43	120:00	00:00	120:00	00:00	49:13	49:57	03:28	116:33	

Issues

- Broken ECUs usually lose power before the events have a chance to write.
- Hard Brake and Last Stop data are from the previous events (i.e. not interesting).
- The process of removing the chip is destructive.
- Reinstalling the chip requires special equipment (and patience)
- Ball Grid Array (BGA) chips are particularly challenging.



https://www.forentec.ch/weitere-services/mobile-forensics-training/teeltech-bga-chip-off-forensic/

Sometimes there are serious issues…

Missing Data??

Can we read memory in place?

The chip removal method is challenging.

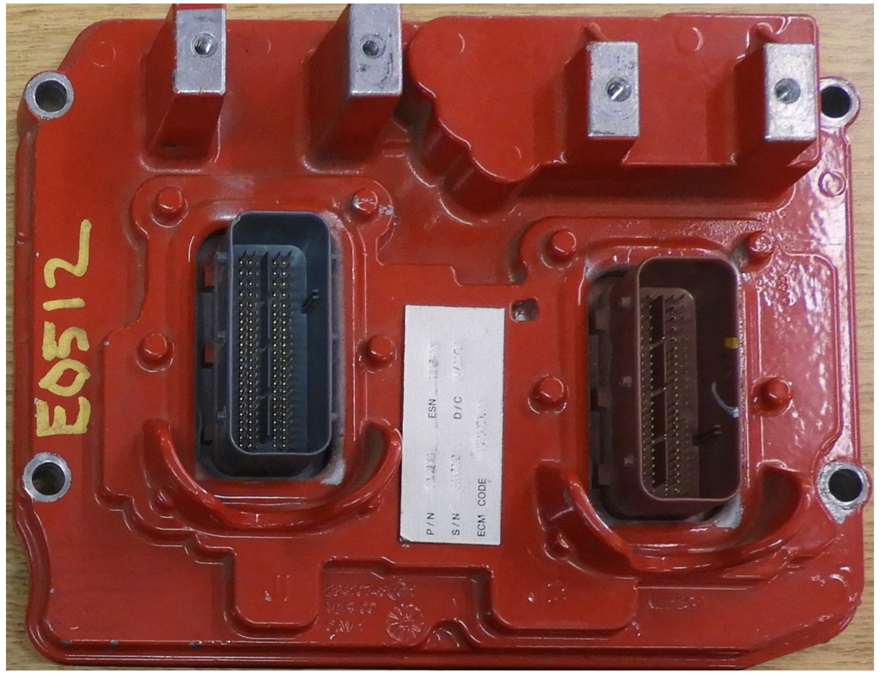
Case Study: Cummins

Cummins CM870

(MPC5xx)

Cummins CM2350

(MPC55xx-57xx)



Case Study: Cummins

Cummins CM870

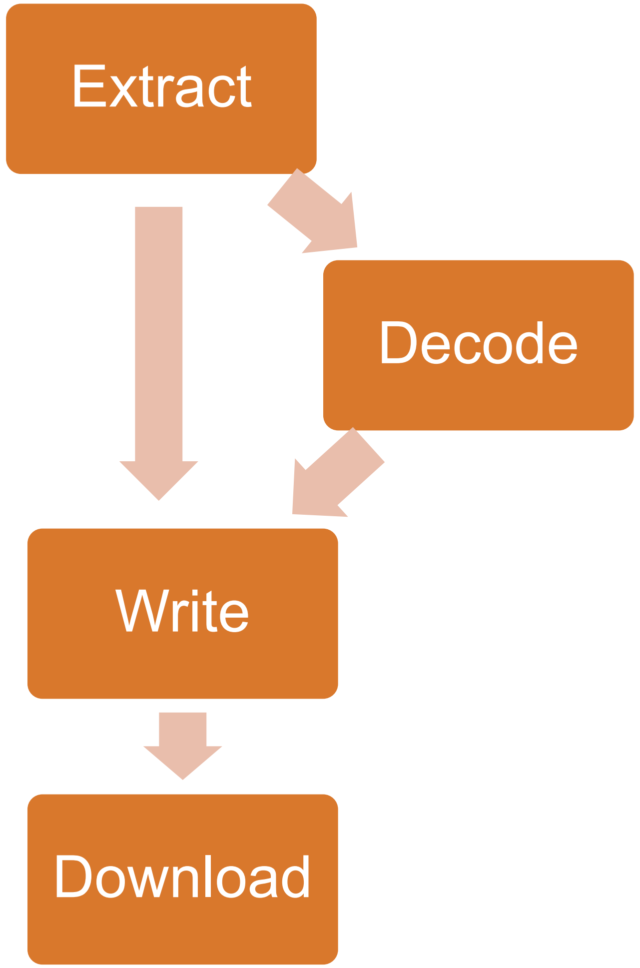
(MPC5xx)

Cummins CM2350

(MPC5674F)

Goals and Objectives

- Safely extract binary data from the broken ECU
- Decode the extracted data to recover
  - Sudden Deceleration Records
  - Data Plate Information
  - Fault Codes
  - Parameters and Settings
- Write the extracted binary to a working ECU
- Download the data using diagnostic or forensic computer programs



Data Extraction via the JTAG Port

- Use the Joint Test Action Group (JTAG) specified programming port for the microprocessor on the ECUs printed circuit board.
- Case study used the following tools:

AlienTech K-TAG Master kit

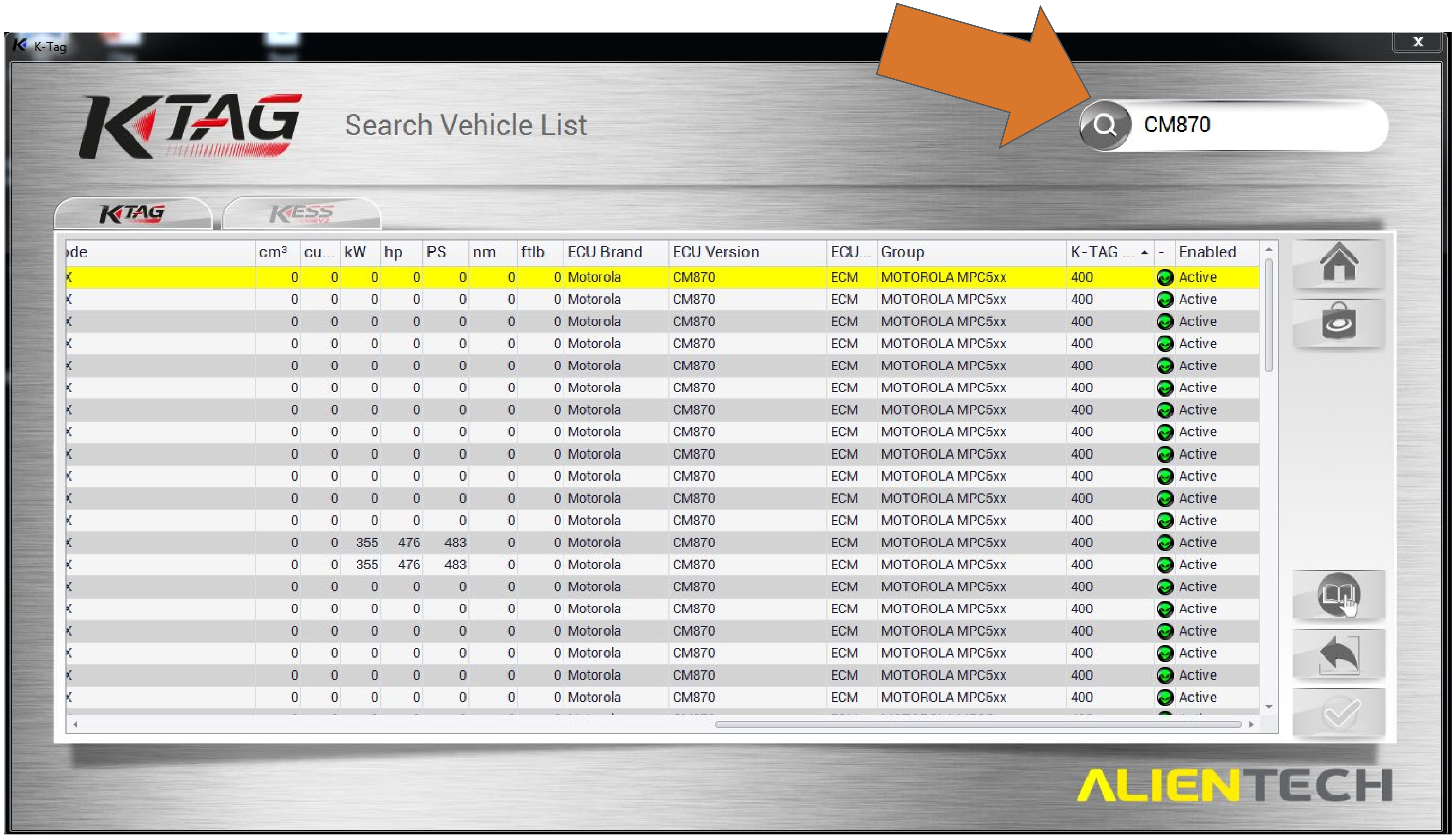
PEmicro CYCLONE FX Programmer

Using the K-TAG Tool

Example for the CM870

K-TAG Software Startup

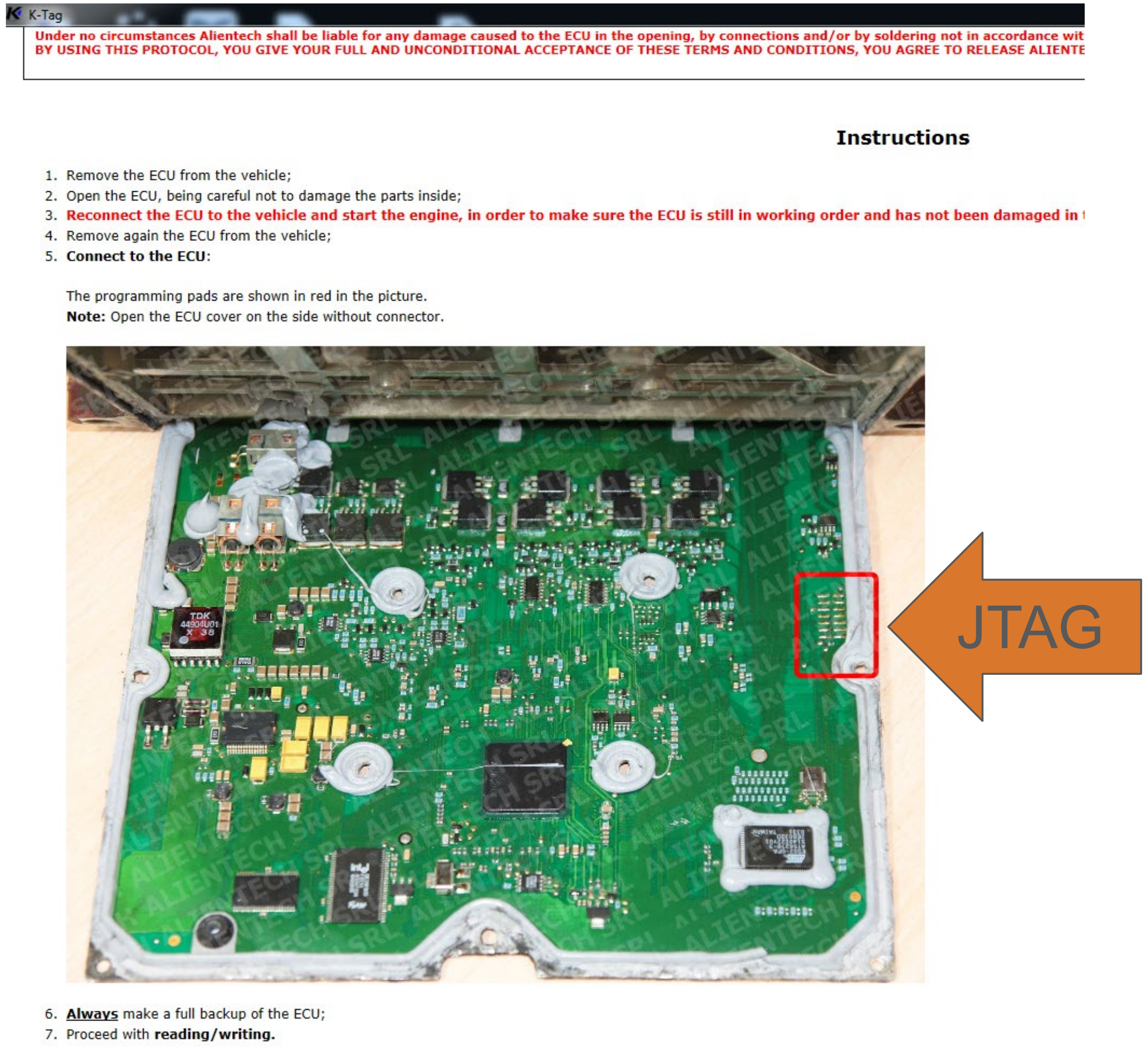
Search for ECM



de	cm³	cu...	kW	hp	PS	nm	ftlb	ECU Brand	ECU Version	ECU...	Group	K-TAG ...	-	Enabled
X	0	0	0	0	0	0	0	Motorola	CM870	ECM	MOTOROLA MPC5xx	400		Active
X	0	0	0	0	0	0	0	Motorola	CM870	ECM	MOTOROLA MPC5xx	400		Active
X	0	0	0	0	0	0	0	Motorola	CM870	ECM	MOTOROLA MPC5xx	400		Active
X	0	0	0	0	0	0	0	Motorola	CM870	ECM	MOTOROLA MPC5xx	400		Active
X	0	0	0	0	0	0	0	Motorola	CM870	ECM	MOTOROLA MPC5xx	400		Active
X	0	0	0	0	0	0	0	Motorola	CM870	ECM	MOTOROLA MPC5xx	400		Active
X	0	0	0	0	0	0	0	Motorola	CM870	ECM	MOTOROLA MPC5xx	400		Active
X	0	0	0	0	0	0	0	Motorola	CM870	ECM	MOTOROLA MPC5xx	400		Active
X	0	0	0	0	0	0	0	Motorola	CM870	ECM	MOTOROLA MPC5xx	400		Active
X	0	0	0	0	0	0	0	Motorola	CM870	ECM	MOTOROLA MPC5xx	400		Active
X	0	0	0	0	0	0	0	Motorola	CM870	ECM	MOTOROLA MPC5xx	400		Active
X	0	0	0	0	0	0	0	Motorola	CM870	ECM	MOTOROLA MPC5xx	400		Active
X	0	0	355	476	483	0	0	Motorola	CM870	ECM	MOTOROLA MPC5xx	400		Active
X	0	0	355	476	483	0	0	Motorola	CM870	ECM	MOTOROLA MPC5xx	400		Active
X	0	0	0	0	0	0	0	Motorola	CM870	ECM	MOTOROLA MPC5xx	400		Active
X	0	0	0	0	0	0	0	Motorola	CM870	ECM	MOTOROLA MPC5xx	400		Active
X	0	0	0	0	0	0	0	Motorola	CM870	ECM	MOTOROLA MPC5xx	400		Active
X	0	0	0	0	0	0	0	Motorola	CM870	ECM	MOTOROLA MPC5xx	400		Active
X	0	0	0	0	0	0	0	Motorola	CM870	ECM	MOTOROLA MPC5xx	400		Active
X	0	0	0	0	0	0	0	Motorola	CM870	ECM	MOTOROLA MPC5xx	400		Active

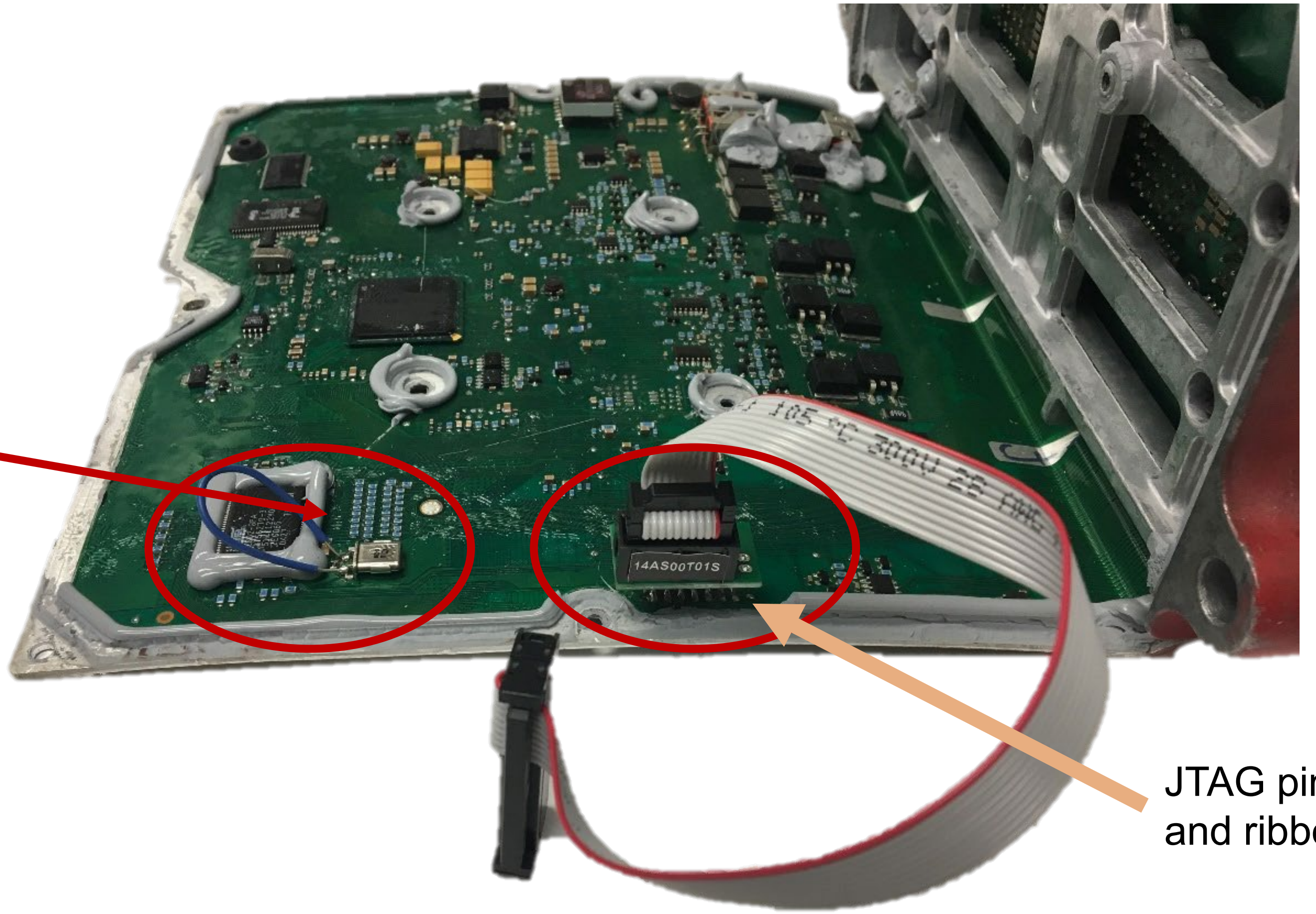
Locate the JTAG Port

- After choosing the correct ECU, click on the bookmark icon for wiring instruction



JTAG Port Connection

- Follow the instruction on the software, solder the port and attach the ribbon.



Bridge the clock crystal

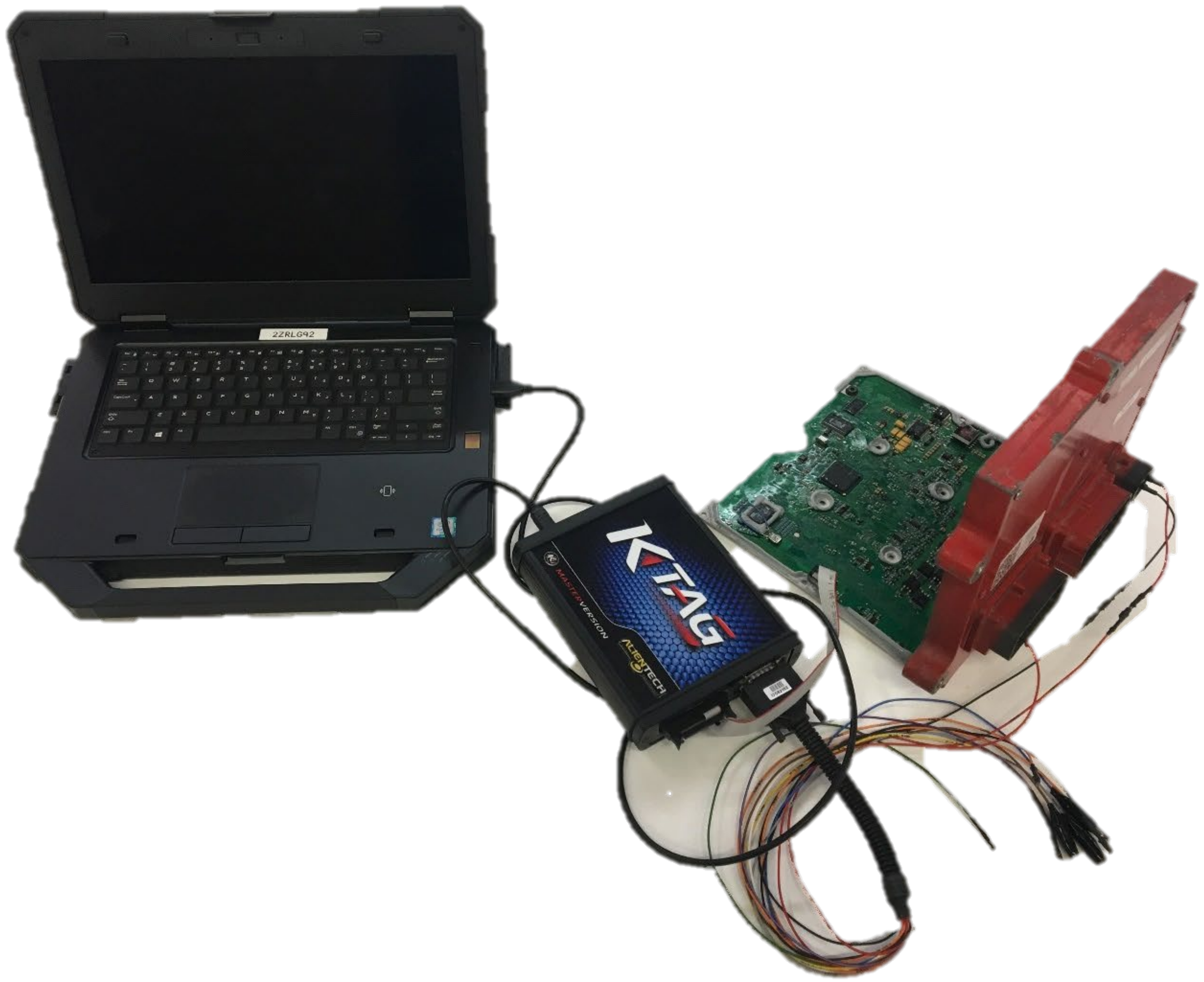
JTAG pin headers and ribbon cable

Connect Power

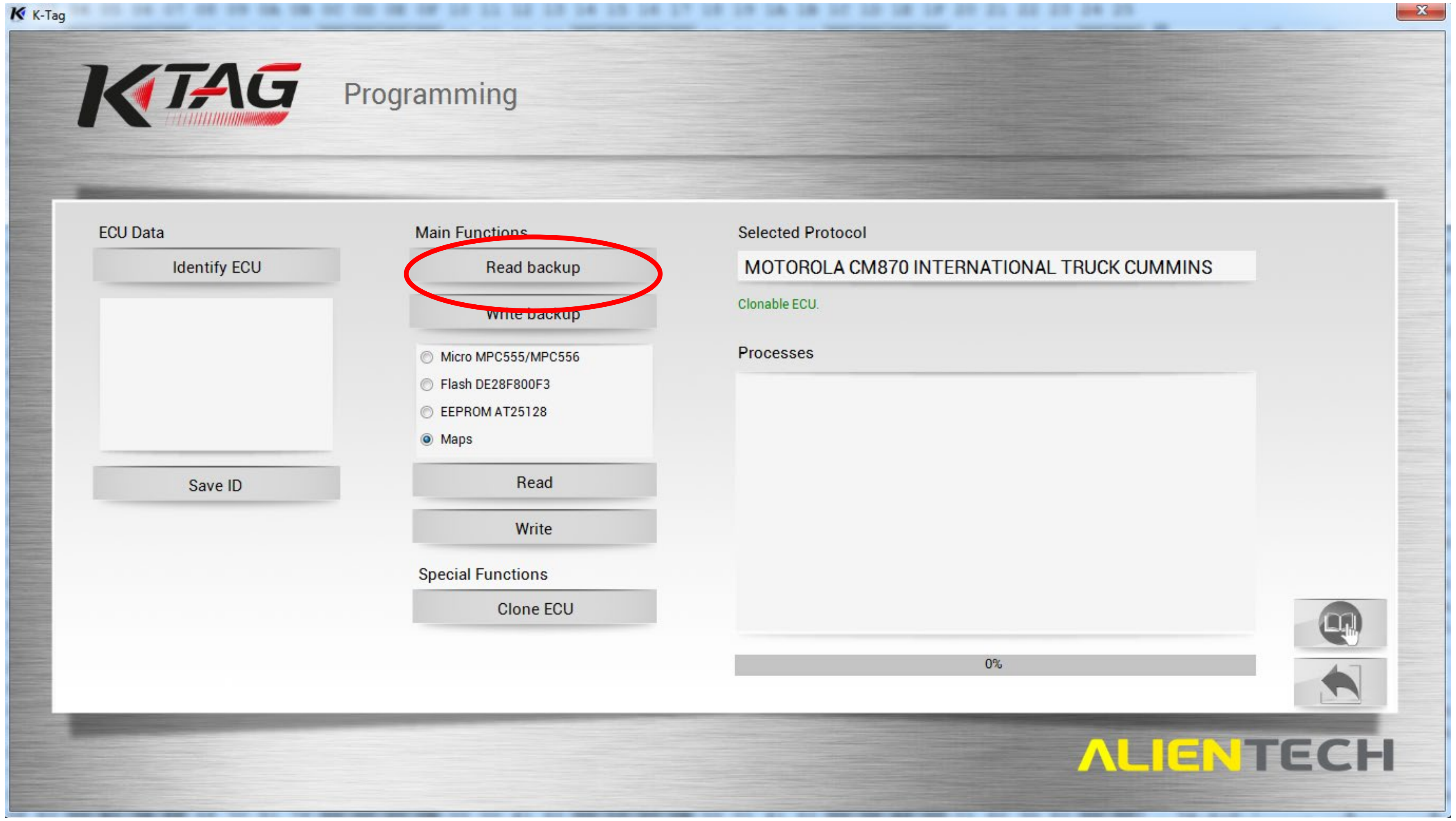
- Attach the D-sub cable to the ECU connector
- Power
- Ignition

PC Connection Setup

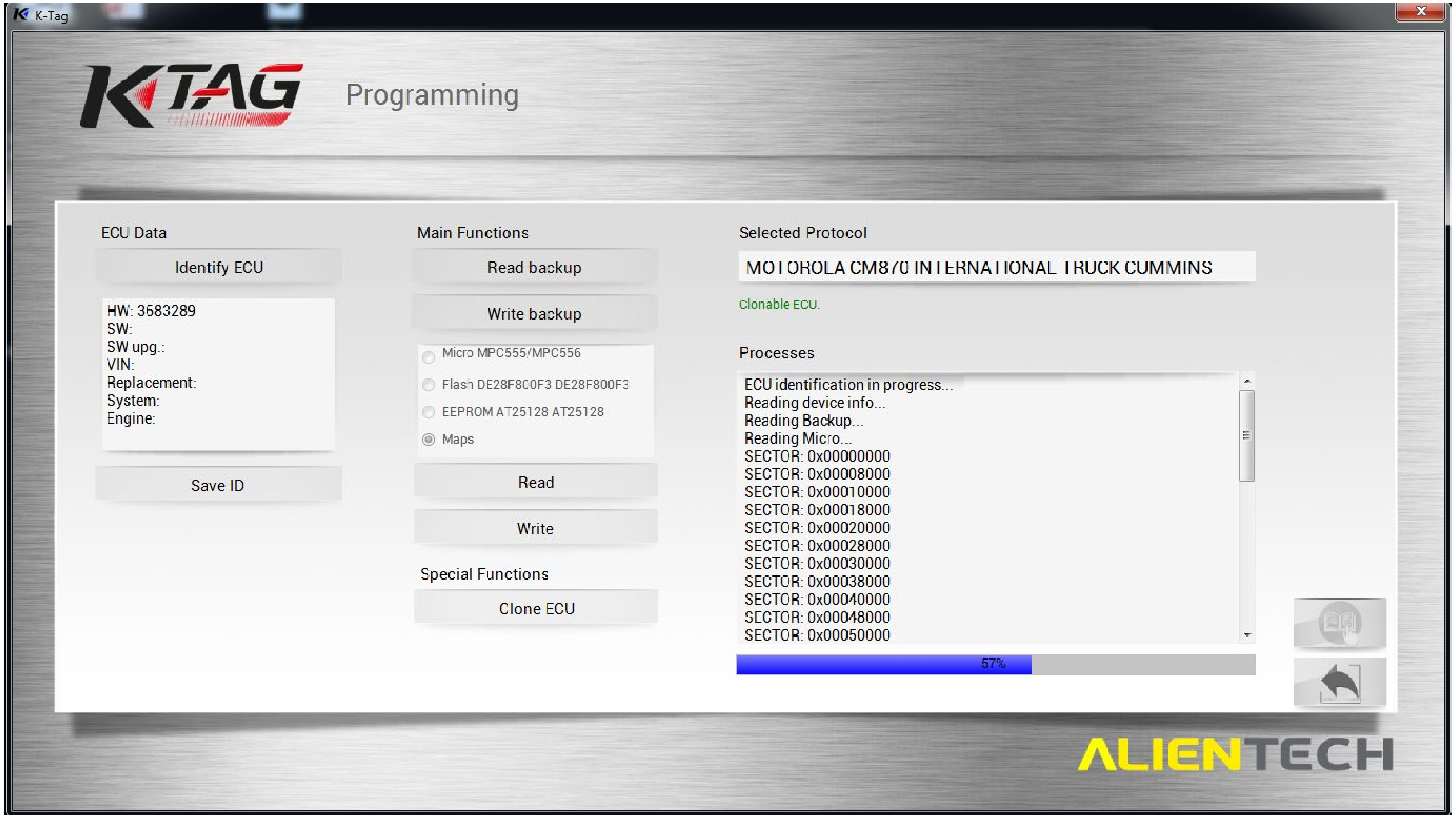
- Connect the KTAG to the ECU and the computer



Retrieving Data

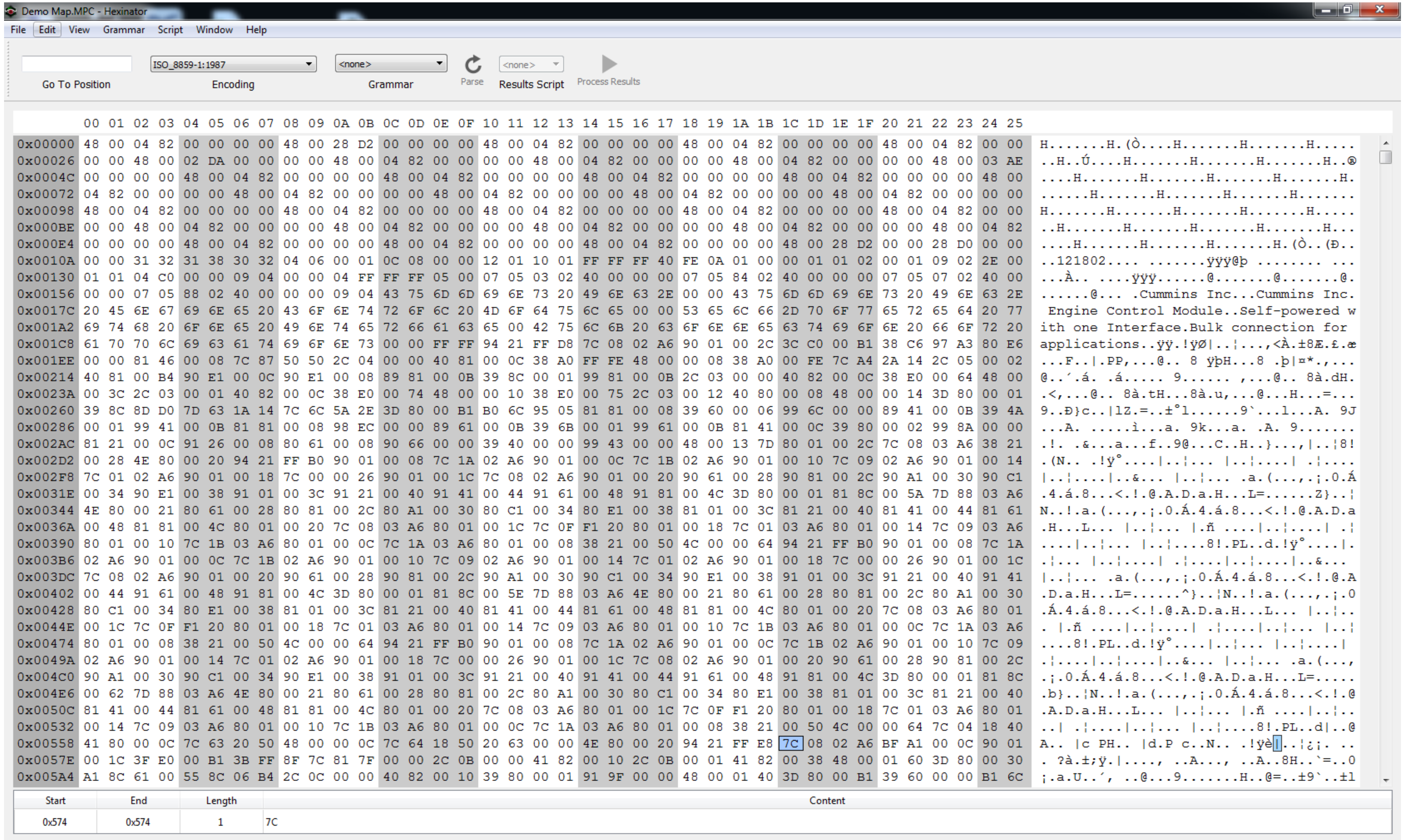


Retrieving Data



Save the Binary Data

Binary Result in Hex Editor

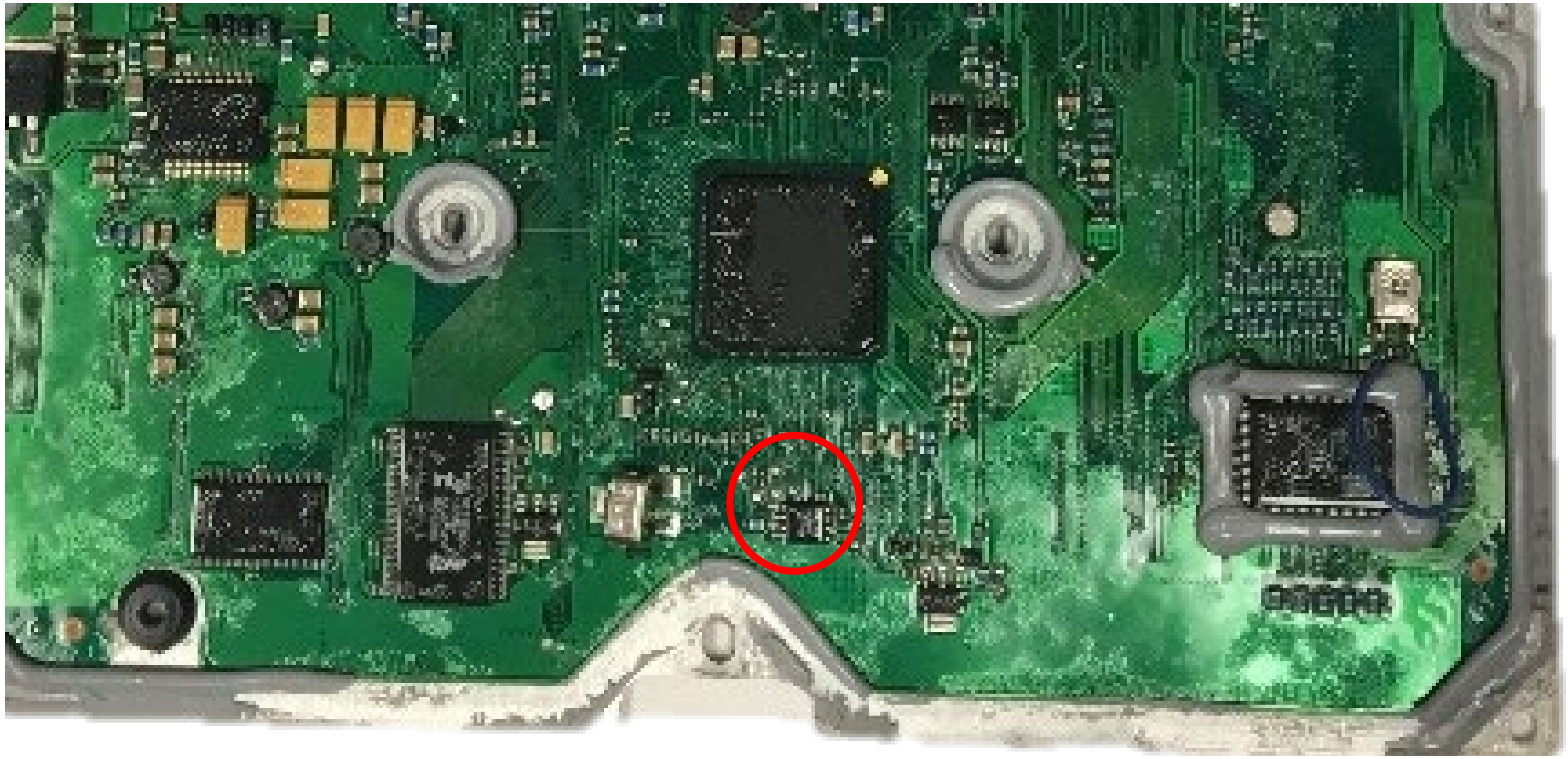


Chip Level Access for Flash

- Flash chip, can be retrieved using individual chip reader

Chip Level Access: EEPROM

- For CM870, EEPROM carries dataplate information via ASCII



Dataplate Attribution Data

- Closer look at PowerSpec and EEPROM (for CM870)

Engine Dataplate Report

Engine Type	ISX 02	Ecm Code	AB10402.22
Engine Serial Number	79076145	Software Phase	6.5.4.2
Unit Number	25175	Extraction Date	04-02-2018 05:48:00

ECM Information

Module Name	CM870
Ecm Code	AB10402.22
Software Phase	6.5.4.2
ECM Serial Number	23052876
ECM Part Number	3683289

Engine Information

Engine Model	ISX 02
Engine Build Date	N/A
Engine Serial Number	79076145
Do Option	1325
SC Option	11145

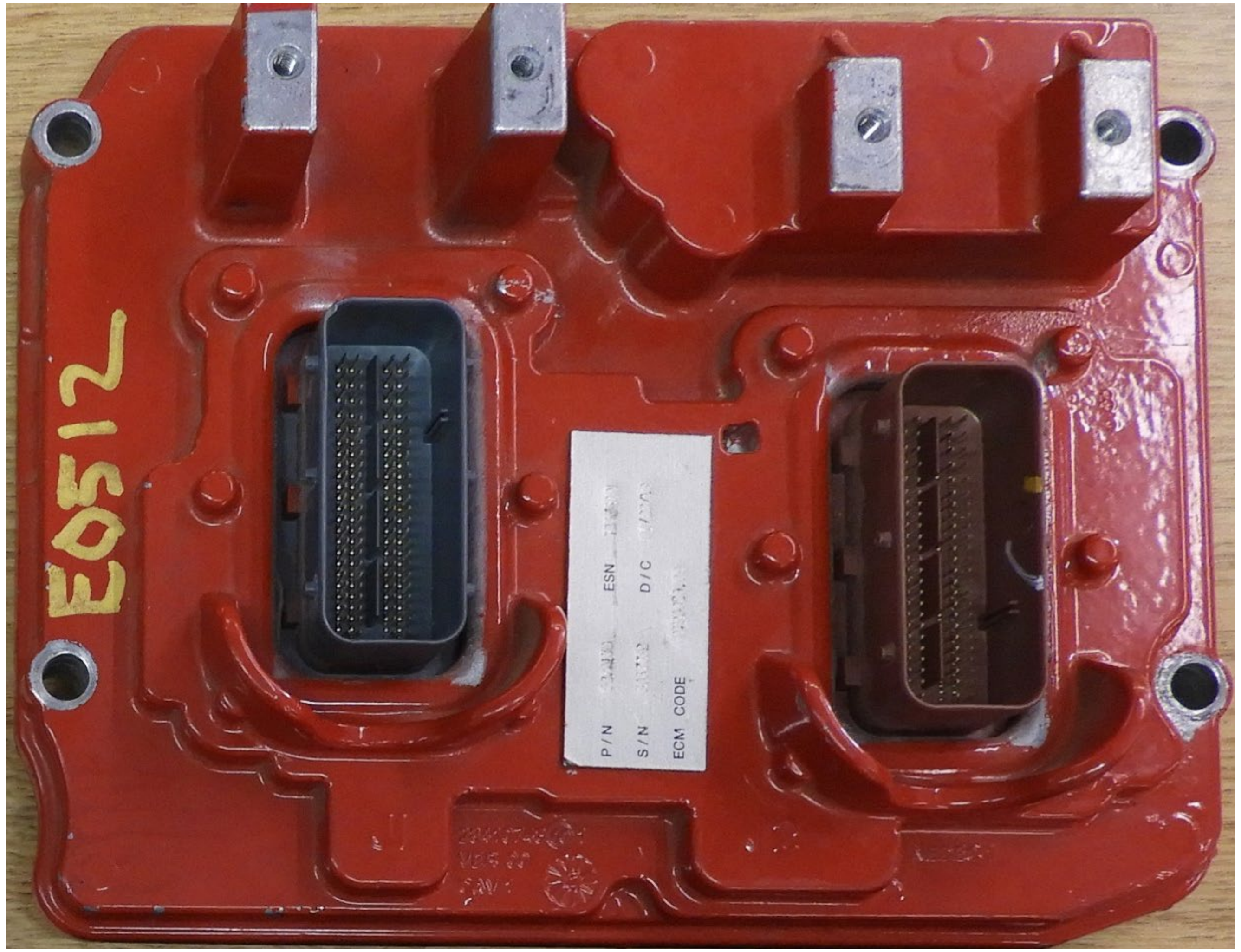
Vehicle Information

Vehicle Identification Number (VIN)	4V4NC9TG25N391063
Vehicle or Equipment Year	
OEM Vehicle Equipment Model	STA15
Customer Name	central
Customer Location	utah
Vehicle Unit Number	25175

```
.......................u5..p...)..N,.....
............................^CMMNS.......
...........¶.1.¯¨!H...................FC0
P388..r.°......TS.83Ù._ÂL..AB10402.22
       ..+....-
         ..                 STA15         4
V4NC9TG25N391063              central
 utah              25175      497294010049729
452003409167300497294340049728955004972882
60049729267004972951800
 .XÃnjdb@@000000000000000000000000000000..
... . .....d.!..... .........ì...........
```

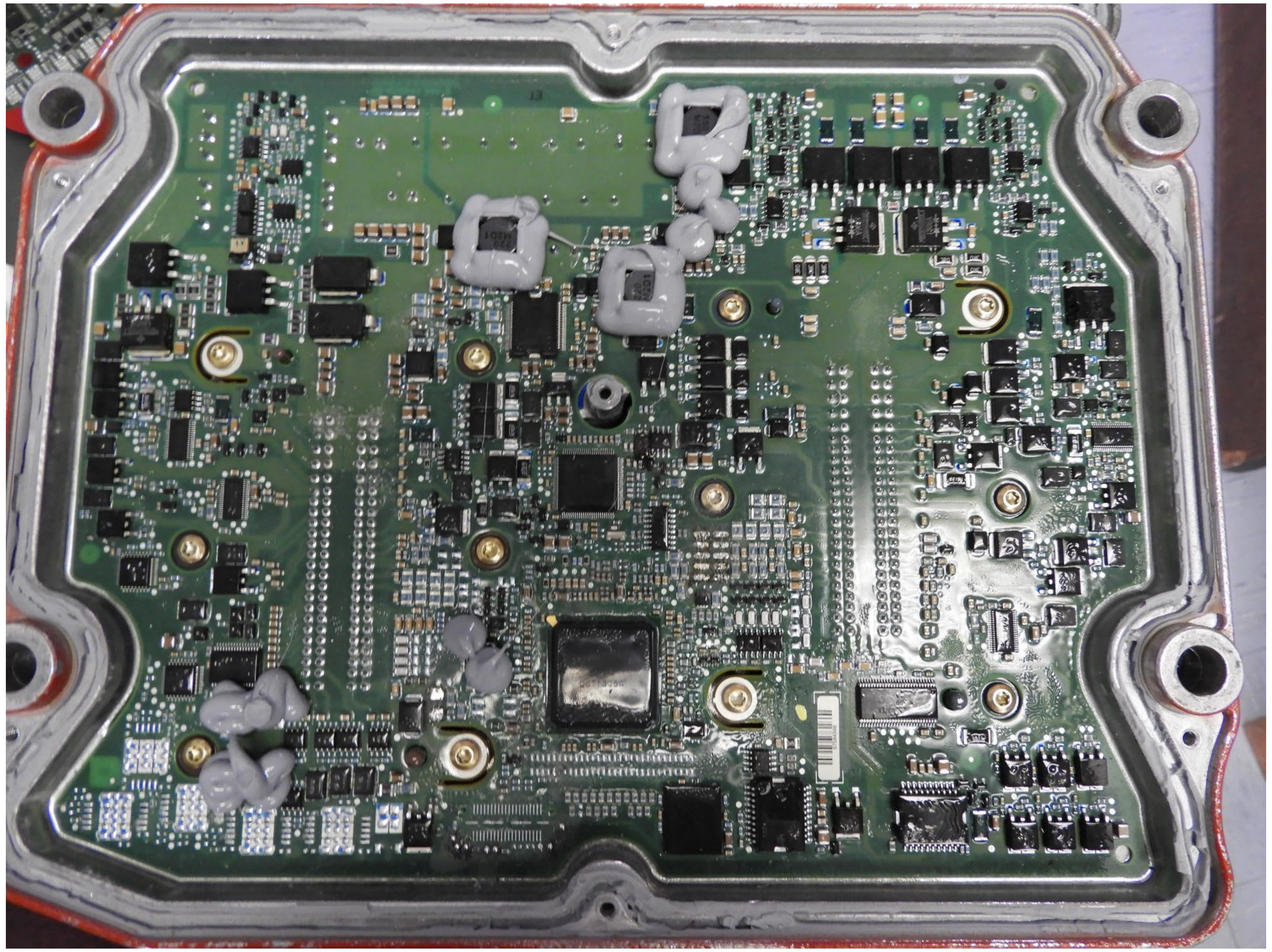
Part of the Hex editor Window

Example for the Cummins CM2350



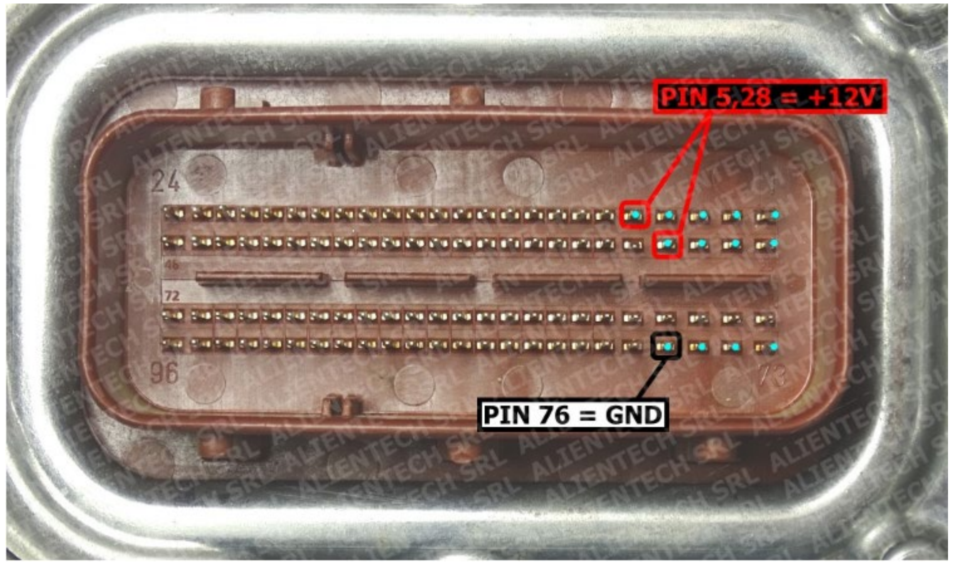
CM2350 Component Identification

Open the ECM to identify JTAG port

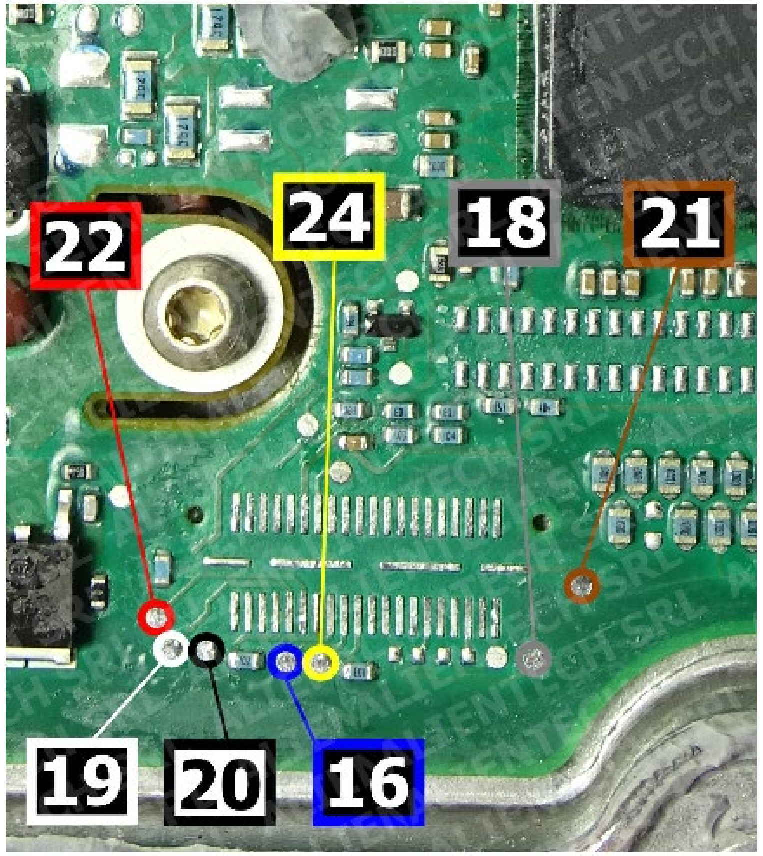


Where is the JTAG port?

CM2350 Connections from KTAG Software

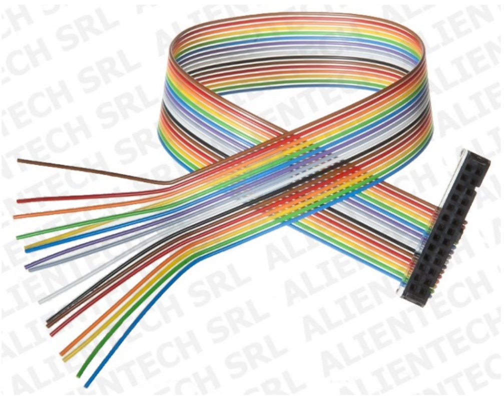


Power, ignition and ground

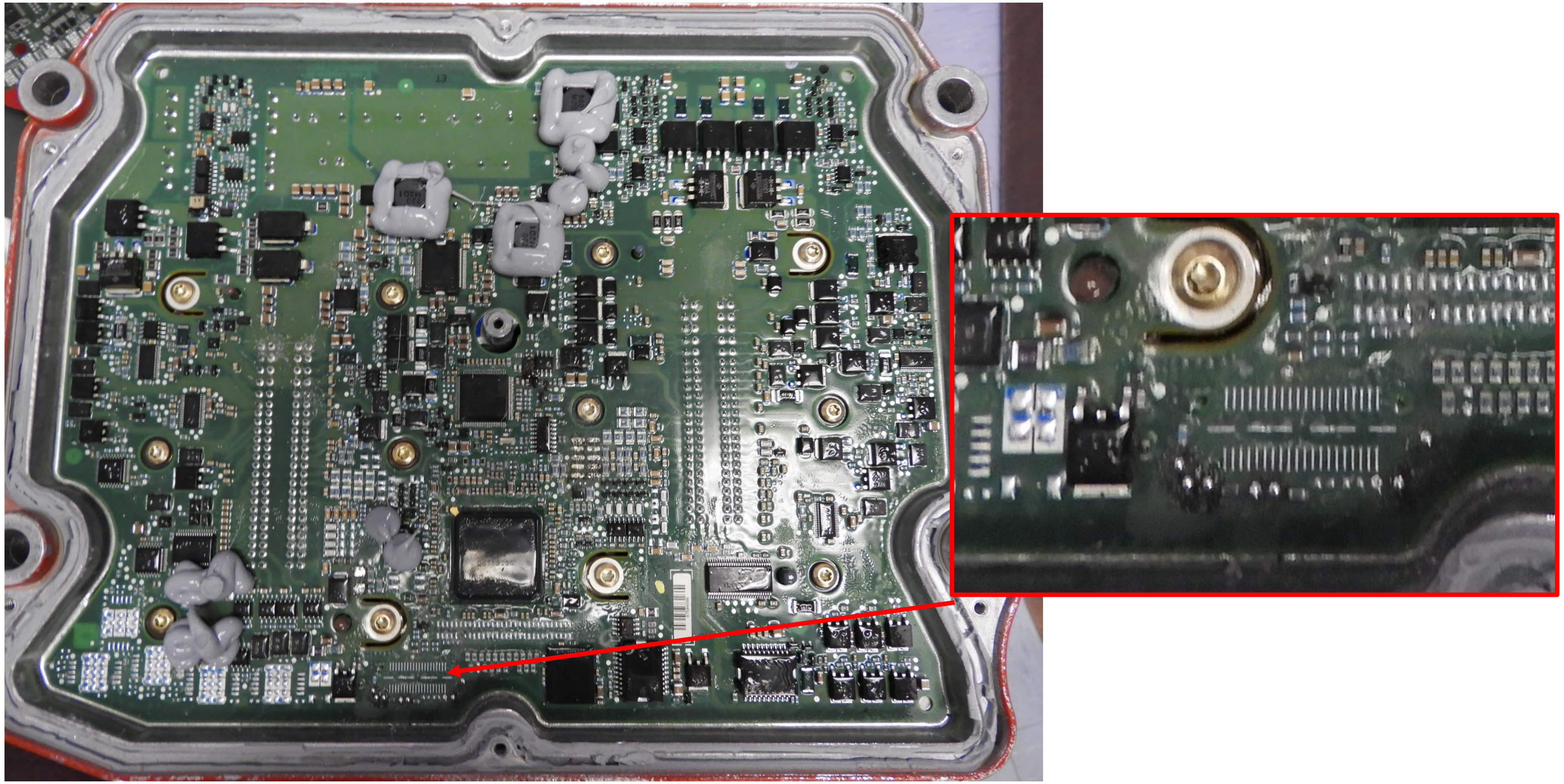


Test Pad Connections

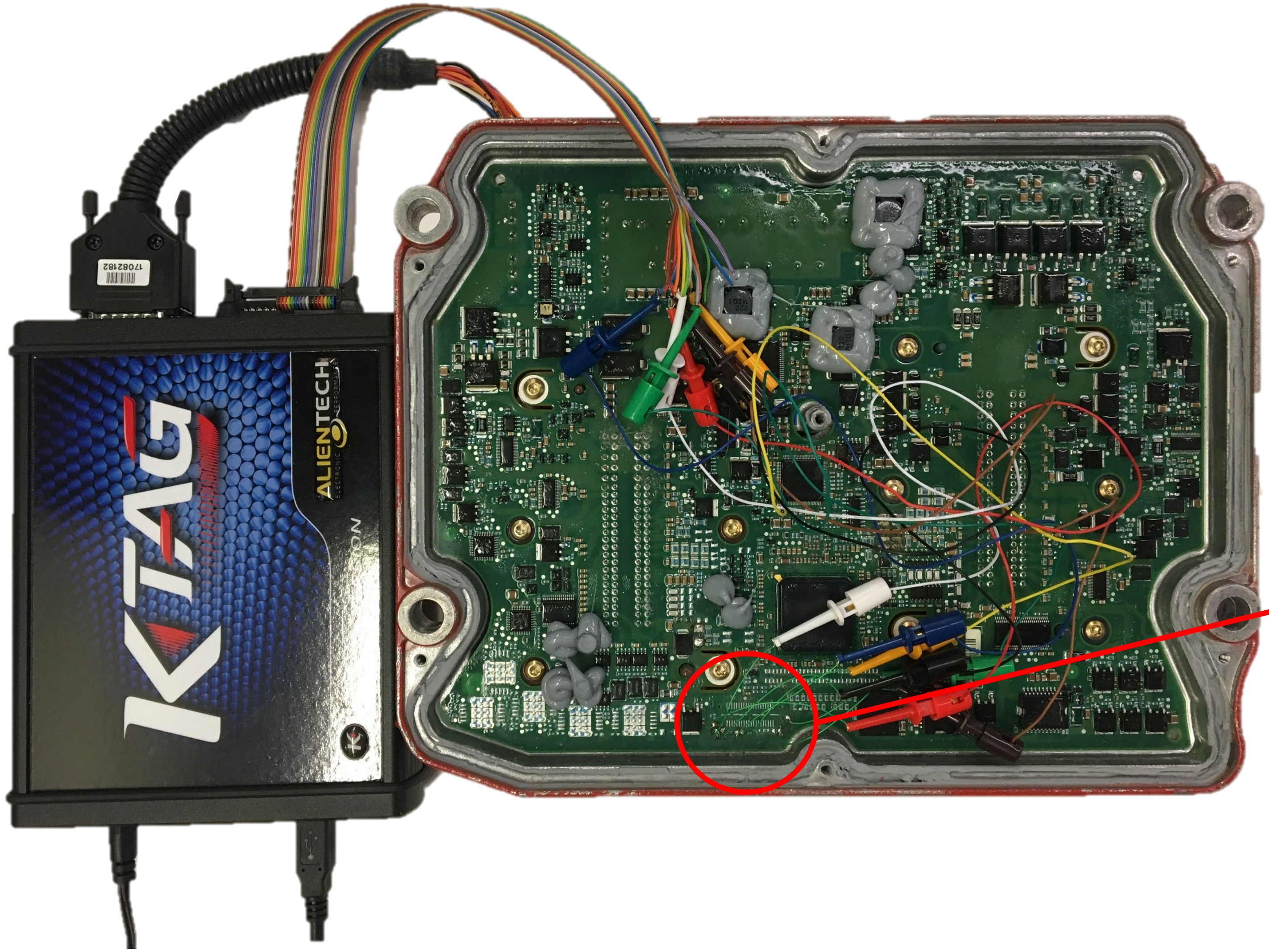
Ribbon Cable

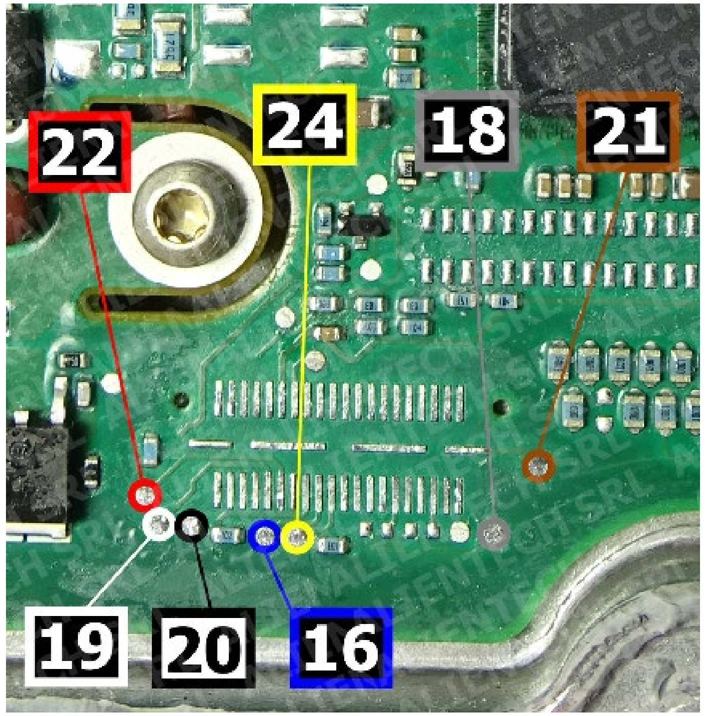


JTAG is built into a Nexus debug port



KTAG Wiring Connections





Summary of Binary Data Collection

CM870

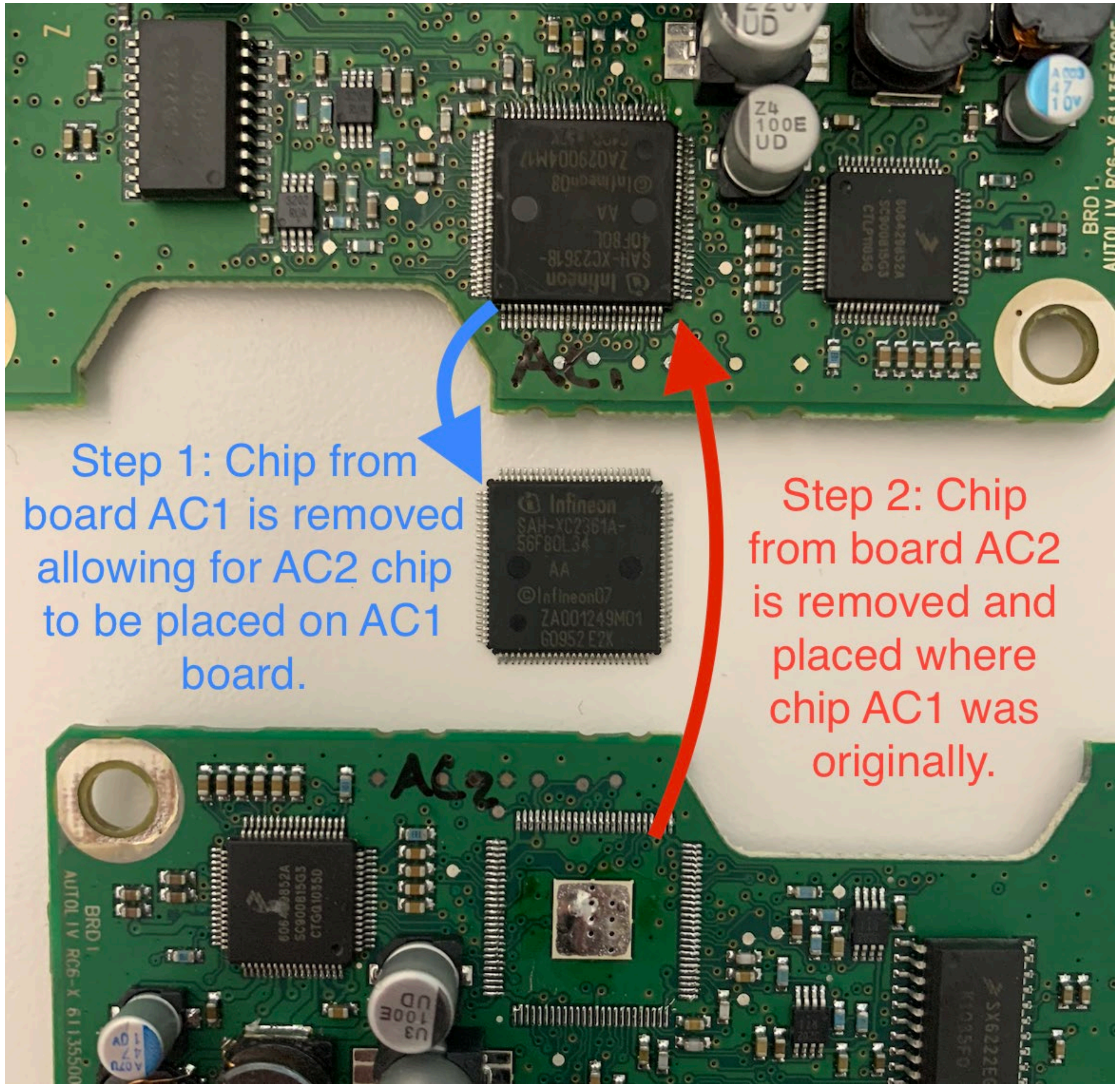
- Three chips
  - Microprocessor
  - External Flash
  - EEPROM (16 kilobytes)
- 1 MB Flash (Intel)
- Flash can be separately extracted with the Xeltek chip reader
- Flex PCB Assembly
- EEPROM contains Data Plate

CM2350

- Single Chip
  - Microprocessor contains flash memory
- 3 MB Flash
- Ball Grid Array
- Non-Standard JTAG Pinout
- Traditional FR4 Printed Circuit Board

Chip Swap

[Removing a Chip by Carson Green: https://youtu.be/q_pJ8OYdXkQ](https://youtu.be/q_pJ8OYdXkQ)



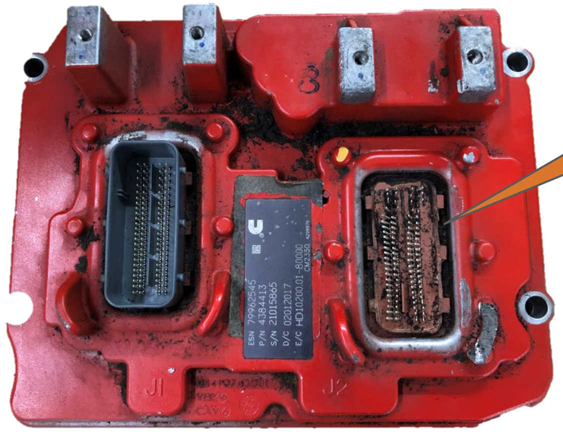
https://www.youtube.com/channel/UCWaZ7puxImvRRNtnOxc4bAA

Introducing the Virtual Chip Swap

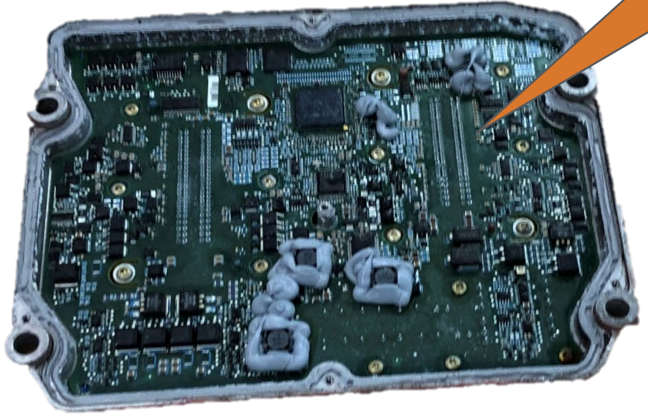
We can extract data from an ECU, but can we write data back to it?

Cloning an ECU

Damaged ECU

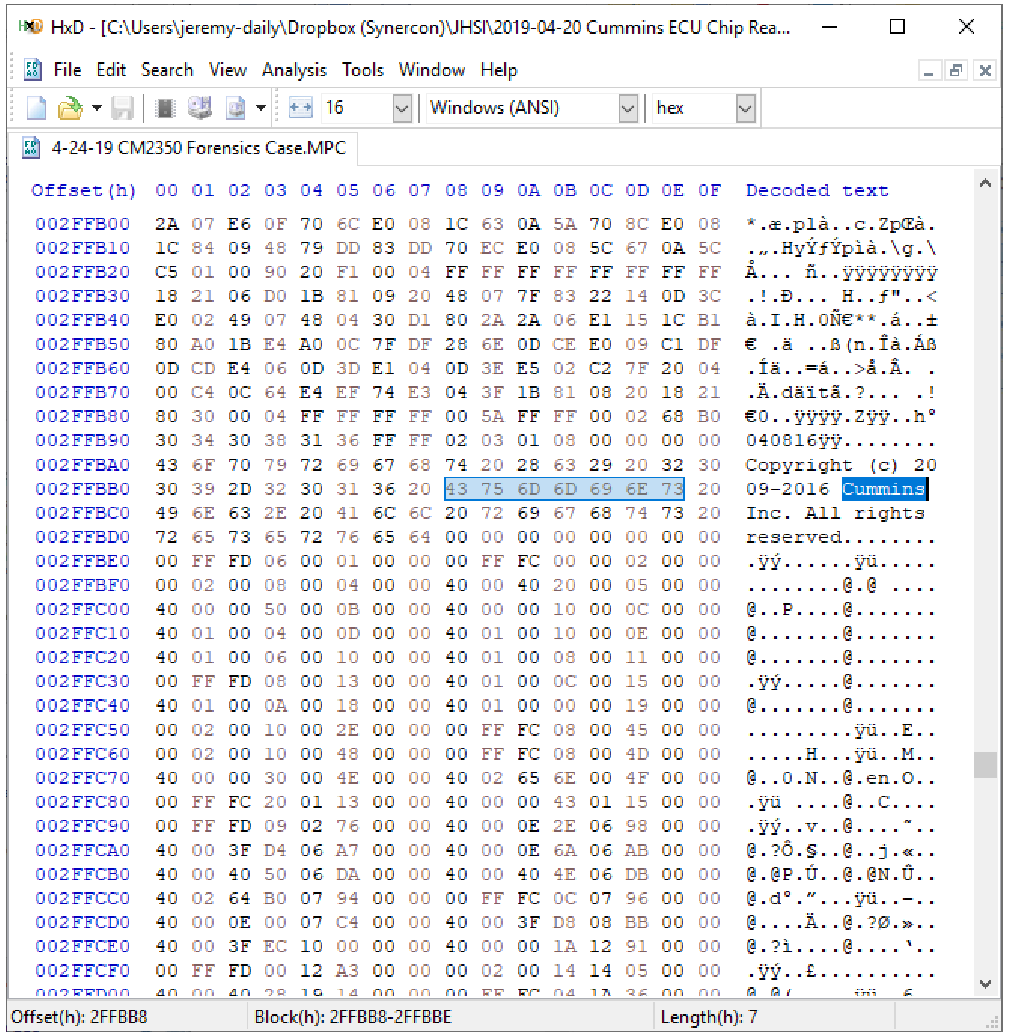


Broken Connector



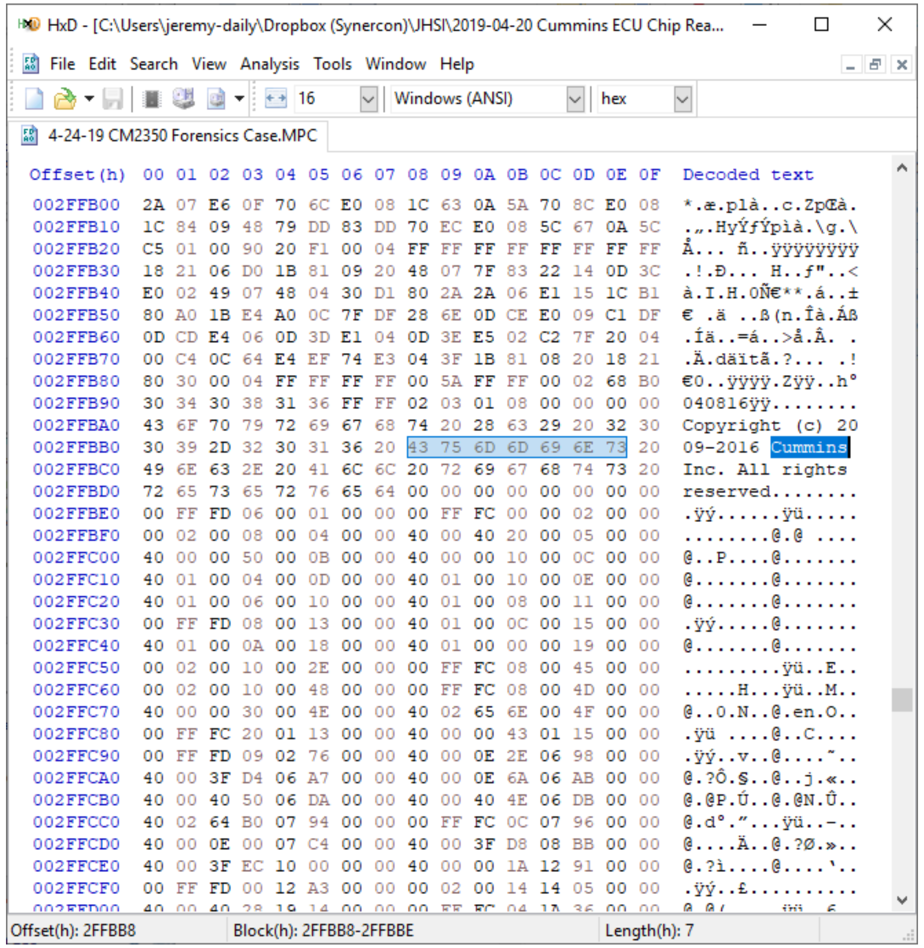
Broken Connector Backside

Binary File



Cloning Process

Binary File



Load Binary onto working ECU;

Perform a bench download

Interpret Data from ECU after Cloning

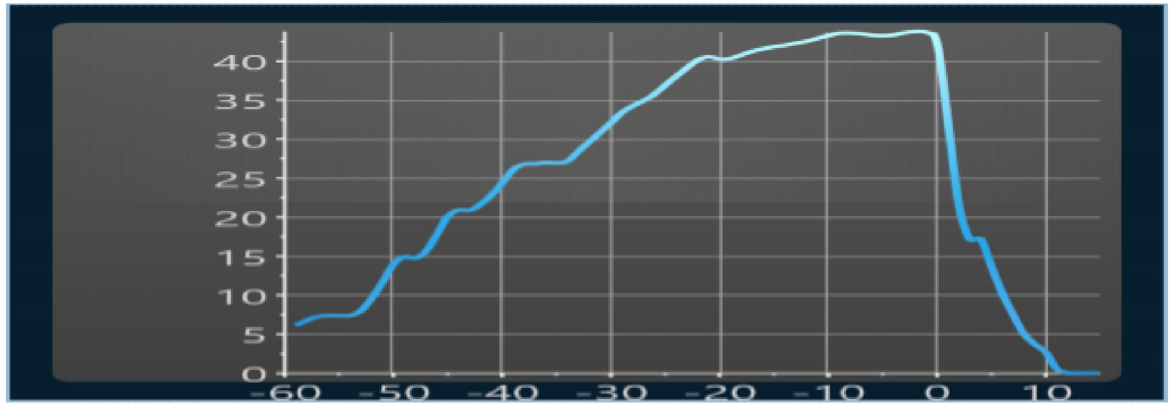
- After cloning the ECU, download with traditional benchtop methods
- Example: Cummins PowerSpec with a DPA5 connected to a Smart Sensor Simulator 2

Sudden Vehicle Speed Deceleration Report Record 1

Engine Type	ISX 2010	Ecm Code	CL10132.39
Engine Serial Number	60811136	Software Phase	7.70.0.71
Unit Number	0000000000	Extraction Date	04-03-2019 04:37:51
Sudden Decel Threshold Rate:	N/A	ECM Run time	8227:56:32

Occurrence Date: N\A	ECM Run Time at Occurrence: 2899:18:36
Air Temperature (°F) at Occurrence: 72	Occurrence Distance (mi): 109323.4

Vehicle Speed

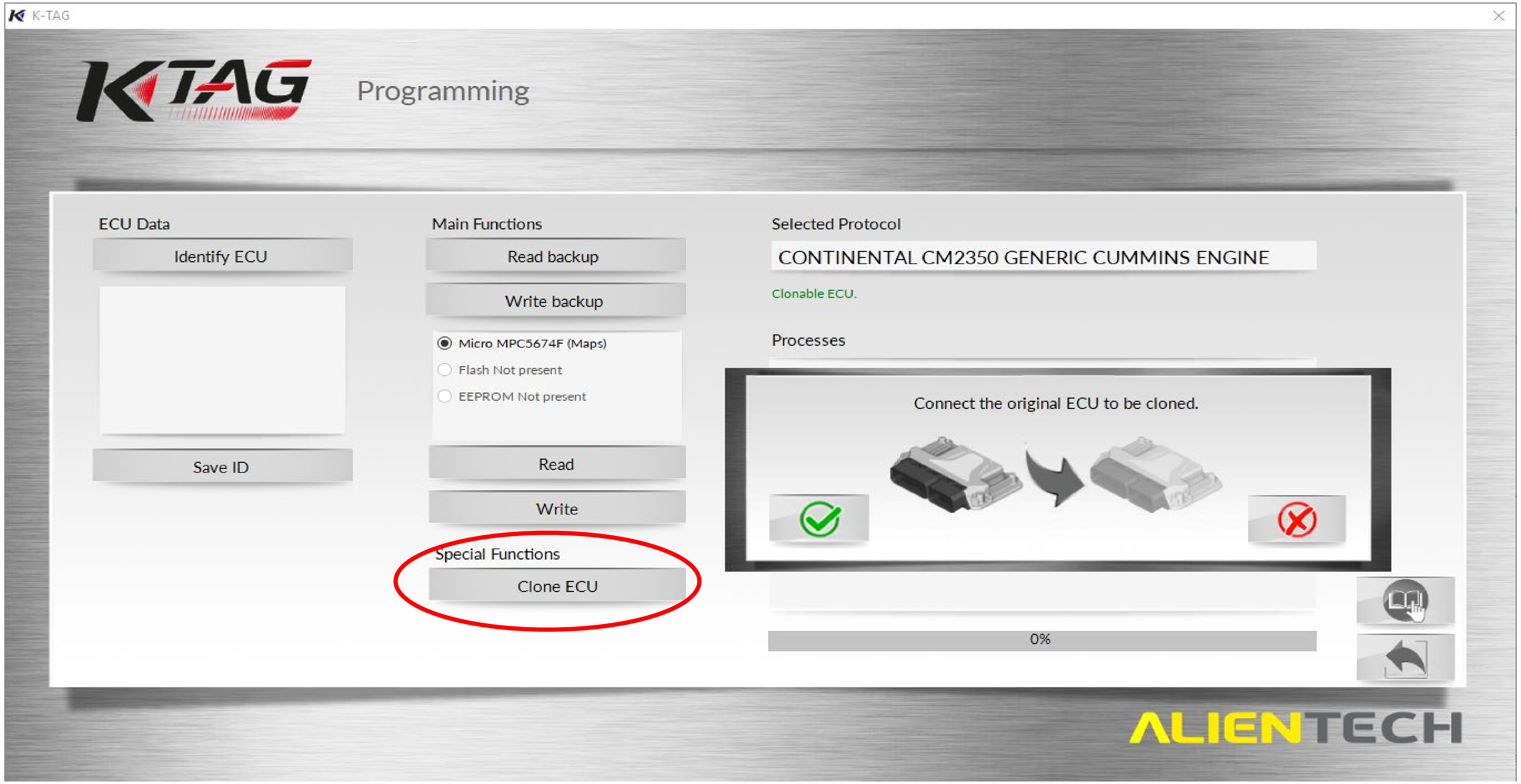


Record 1

Time (Seconds)	Vehicle Speed (mph)	Engine Speed (rpm)	Engine Load (%)	Throttle (%)	Brake Status	Clutch Status	Cruise Status	Lamp Status
-59	6	683	34.8	30.1	-	-	-	-
-58	7	788	33.6	32.2	-	-	-	-
-57	7	813	21.9	27.5	-	-	-	-

Cloning Process with the K-TAG

Select Clone ECU function and follow instructions



Cloning Example: Two Different ECUs

Engine Dataplate Report (A)

Engine Type	ISX 2013	Ecm Code	EF10067.36
Engine Serial Number	0	Software Phase	9.40.13.61
Unit Number	0000000000	Extraction Date	07-03-2018 12:37:22

ECM Information

Module Name	CM2350
Ecm Code	EF10067.36
Software Phase	9.40.13.61
ECM Serial Number	3032682
ECM Part Number	5290170

Engine Information

A

Engine Model	ISX 2013
Engine Build Date	010185
Engine Serial Number	0
Do Option	1874
SC Option	12186

Vehicle Information

Vehicle Identification Number (VIN)	1HSDJAPR7EH784086
Vehicle or Equipment Year	2014
OEM Vehicle Equipment Model	***************
Customer Name	CustomerName**
Customer Location	***************
Vehicle Unit Number	0000000000

Engine Dataplate Report (B)

Engine Type	ISB 2013	Ecm Code	DT90216.08
Engine Serial Number	73993334	Software Phase	21.60.71.2
Unit Number	**********	Extraction Date	07-03-2018 12:14:56

ECM Information

...e	CM2350
	DT90216.08
...se	21.60.71.2
...umber	6069868
ECM Part Number	4358814

Engine Information

B

Engine Model	ISB 2013
Engine Build Date	110516
Engine Serial Number	73993334
Do Option	94370
SC Option	92050

Vehicle Information

Vehicle Identification Number (VIN)	2NKHHM7X3HM156651
Vehicle or Equipment Year	**********
OEM Vehicle Equipment Model	T300
Customer Name	CustomerName**
Customer Location	***************
Vehicle Unit Number	**********

Extraction Date not in ECU

ECM dataplate of the two ECMs **before** cloning

Cloning Results: A is now B

Engine Dataplate Report (A)

Engine Type	ISB 2013	Ecm Code	DT90216.08
Engine Serial Num		hase	21.60.71.2
Unit Number		ate	07-03-2018 03:51:29

ECM Infor

Module Name	CM2350
Ecm Code	DT90216.08
Software Phase	21.60.71.2
ECM Serial Number	3032682
ECM Part Number	5290170

Engine Information

Engine Model	ISB 2013
Engine Build Date	110516
Engine Serial Number	73993334
Do Option	94370
SC Option	92050

Vehicle Information

Vehicle Identification Number (VIN)	2NKHHM7X3HM156651
Vehicle or Equipment Year	**********
OEM Vehicle Equipment Model	T300
Customer Name	CustomerName**
Customer Location	***************
Vehicle Unit Number	**********

A

S/N and P/N are unchanged

Extraction Date 3.5 hours later

Engine Dataplate Report (B)

Engine Type	ISB 2013	Ecm Code	DT90216.08
Engine Serial Number	73993334	Software Phase	21.60.71.2
Unit Number	**********	Extraction Date	07-03-2018 12:14:56

ECM Information

e	CM2350
	DT90216.08
se	21.60.71.2
umber	6069868
mber	4358814

Engine Information

Engine Model	ISB 2013
Engine Build Date	110516
Engine Serial Number	73993334
Do Option	94370
SC Option	92050

Vehicle Information

Vehicle Identification Number (VIN)	2NKHHM7X3HM156651
Vehicle or Equipment Year	**********
OEM Vehicle Equipment Model	T300
Customer Name	CustomerName**
Customer Location	***************
Vehicle Unit Number	**********

B

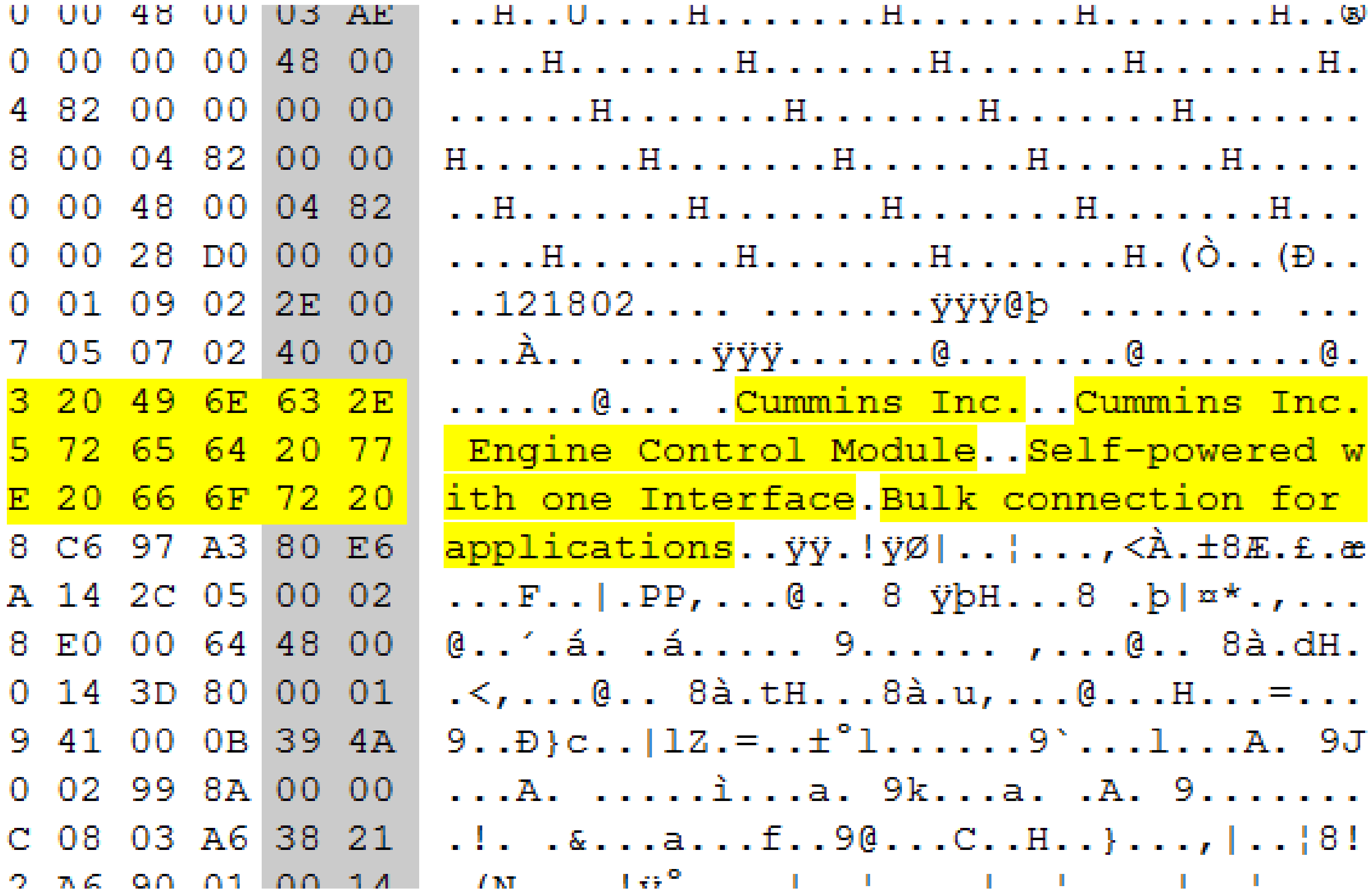
ECM dataplate of the two ECMs **after** cloning

Data Analysis

What if cloning doesn’t work (or you don’t want to spend the money on another ECU)?

Data Analysis: Human Readable Text

Some meaningful strings are present indicating that the data is not encrypted



Data Analysis: Binary File

- Flash chip data should carry Sudden Deceleration information (vehicle speed, engine RPM, etc.)
- The file is 3 MB so how can we locate the Sudden Decel?


```
000001FF70  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF
................
000001FF80  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF
................
000001FF90  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF
................
000001FFA0  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF
................
000001FFB0  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF
................
000001FFC0  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF
................
000001FFD0  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF
................
000001FFE0  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF
................
000001FFF0  FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF
................
0000020000  94 21 FF F0 7C 08 02 A6-93 E1 00 0C 90 01 00 14  .!..
|◘┐....
.  .¶
0000020010  3F E0 00 B2 A3 FF 5B 96-3C E0 00 B2 38 E7 30 BA
?.....[.<...8.0.
0000020020  7C E6 3B 78 3D 00 00 B2-39 08 30 B8 7D 05 43 78
|.;x=...9◘0.}|Cx
0000020030  3D 20 00 B1 A1 29 E3 74-7C 09 F8 00 41 81 00 1C  =
...).t|    ..A..
0000020040  39 80 20 00 B1 88 00 00-3C 80 00 B1 A0 84 E3 76  9.
.....<......v
0000020050  B0 87 00 00 48 00 00 24-3C 80 00 B1 A0 84 E3 70
....H..$<......p
0000020060  B0 88 00 00 3D 00 00 B1-A1 08 E3 72 B1 07 00 00  ....
=....◘.r. ..
0000020070  7D 44 40 50 38 8A 20 00-3D 80 00 B2 B0 8C 30 BC  }
D@P8. .=.....0.
0000020080  3C E0 00 B2 A0 E7 62 68-A1 45 00 00 7D 07 51 D6
<.....bh.E..} Q.
0000020090  3C A0 00 B0 38 A5 5A 58-A1 65 00 00 A1 26 00 00
<...8.ZX.e...&..
00000200A0  7C 6B 49 D6 3C C0 00 B0-38 C6 5A 5A A1 46 00 00  |kI.
<...8.ZZ.F..
00000200B0  54 89 04 3E 7C 8A 49 D6-7D 83 40 50 7D 04 62 14  T.┘>
|.I.}.@P}┘b¶
00000200C0  2C 08 00 00 40 81 00 2C-7D 08 6E 70 3D 60 00 00  ,◘..
@..,}◘np=`..
00000200D0  61 6B 8C A0 7C 08 58 00-40 80 00 0C 7D 04 43 78  ak..
|◘X.@..
```

Data Analysis

One approach:

- Observe that vehicle speed for Record 1 in this ECM has 150 mph for the first 60s.
- There should be repeated data in the binary.

Record 1

Time (Seconds)	Vehicle Speed (mph)	Engine Speed (rpm)	Engine Load (%)	Throttle (%)	Brake Status	Clutch Status	Cruise Status	Lamp Status
-59	150	0	0.0	0.0	-	-	-	On
-58	150	0	0.0	0.0	-	-	-	On
-57	150	0	0.0	0.0	-	-	-	On
-56	150	0	0.0	0.0	-	-	-	On
-55	150	0	0.0	0.0	-	-	-	On
-54	150	0	0.0	0.0	-	-	-	On
-53	150	0	0.0	0.0	-	-	-	On
-52	150	0	0.0	0.0	-	-	-	On
-51	150	0	0.0	0.0	-	-	-	On
-50	150	0	0.0	0.0	-	-	-	On
-49	150	0	0.0	0.0	-	-	-	On
-48	150	0	0.0	0.0	-	-	-	On
-47	150	0	0.0	0.0	-	-	-	On
-46	150	0	0.0	0.0	-	-	-	On
-45	150	0	0.0	0.0	-	-	-	On
-44	150	0	0.0	0.0	-	-	-	On
-43	150	0	0.0	0.0	-	-	-	On
-42	150	0	0.0	0.0	-	-	-	On
-41	150	0	0.0	0.0	-	-	-	On
-40	150	0	0.0	0.0	-	-	-	On
-39	150	0	0.0	0.0	-	-	-	On
-38	150	0	0.0	0.0	-	-	-	On
-37	150	0	0.0	0.0	-	-	-	On

Data Analysis

Converting the actual vehicle speed from the report to raw data format using SAE J1939-71

SPN 84 ***Wheel-Based Vehicle Speed***

Speed of the vehicle as calculated from wheel or tailshaft speed.

Data Length:	2 bytes	
Resolution:	1/256 km/h per bit, 0 offset	
Data Range:	0 to 250.996 km/h	Operational Range: same as data range
Type:	Measured	
Supporting Information:		
PGN reference:	65265	

Data Analysis Approach

- Convert 150 mph to km/h: 241.4 km/h
- Convert 241.4 km/h to bit:

  241.4 km/h x 256 bit/km/h = 64798
- Convert 64798 to Hex: F1 66
- Look for repeating F1 66 pattern in the report

NO PATTERN! WHAT NOW?!

Data Analysis

- Convert 150 mph to data format without converting to km/h
- Convert 150.000 to hex:

  150 * 256 = 38,400

- Convert 38,400 to Hex: 96 00

Data Analysis

We got something!

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
0x0FB1D0	00	00	00	00	00	00	00	00	0C	31	00	00	00	00	00	00	.... 1......
0x0FB1E0	00	00	00	73	00	02	1E	FA	15	9B	85	8A	12	F6	01	D2	...s.. ú.....ö.Ò
0x0FB1F0	79	DC	00	00	96	00	00	00	00	00	00	00	00	00	00	01	yÜ..............
0x0FB200	00	00	96	00	00	00	00	00	00	00	00	00	00	01	00	00	....
0x0FB210	96	00	00	00	00	00	00	00	00	00	00	01	00	00	96	00	....
0x0FB220	00	00	00	00	00	00	00	00	00	01	00	00	96	00	00	00	....
0x0FB230	00	00	00	00	00	00	00	01	00	00	96	00	00	00	00	00	....
0x0FB240	00	00	00	00	00	01	00	00	96	00	00	00	00	00	00	00	....
0x0FB250	00	00	00	01	00	00	96	00	00	00	00	00	00	00	00	00	....
0x0FB260	00	01	00	00	96	00	00	00	00	00	00	00	00	00	00	01	....
0x0FB270	00	00	96	00	00	00	00	00	00	00	00	00	00	01	00	00	....
0x0FB280	96	00	00	00	00	00	00	00	00	00	00	01	00	00	96	00	....
0x0FB290	00	00	00	00	00	00	00	00	00	01	00	00	96	00	00	00	....
0x0FB2A0	00	00	00	00	00	00	00	01	00	00	96	00	00	00	00	00	....
0x0FB2B0	00	00	00	00	00	01	00	00	96	00	00	00	00	00	00	00	....
0x0FB2C0	00	00	00	01	00	00	96	00	00	00	00	00	00	00	00	00	....
0x0FB2D0	00	01	00	00	96	00	00	00	00	00	00	00	00	00	00	01	....
0x0FB2E0	00	00	96	00	00	00	00	00	00	00	00	00	00	01	00	00	....
0x0FB2F0	96	00	00	00	00	00	00	00	00	00	00	01	00	00	96	00	....
0x0FB300	00	00	00	00	00	00	00	00	00	01	00	00	96	00	00	00	....
0x0FB310	00	00	00	00	00	00	00	01	00	00	96	00	00	00	00	00	....

Data Analysis

- But is it the vehicle speed log?
- Tracing the speed along to compare with the report. The 2-byte is 12 bytes apart. Find the first change in the speed,   58 8A

0x0FB4C0	00	00	00	00	00	00	00	00	00	01	00	00	96	00	00	00	....
0x0FB4D0	00	00	00	00	00	00	00	01	00	00	96	00	00	00	00	00	....
0x0FB4E0	00	00	00	00	00	01	00	00	96	00	00	00	00	00	00	00	....
0x0FB4F0	00	00	00	01	00	00	96	00	00	00	00	00	00	00	00	00	....
0x0FB500	00	01	00	00	96	00	00	00	00	00	00	00	00	00	00	01	....
0x0FB510	00	00	96	00	00	00	00	00	00	00	00	00	00	01	00	00	....
0x0FB520	96	00	00	00	00	00	00	00	00	00	00	01	00	00	96	00	....
0x0FB530	00	00	00	00	00	00	00	00	00	01	00	00	58	8A	00	00	............X...
0x0FB540	00	00	00	00	00	00	00	01	00	00	06	58	00	00	00	00	...........X....
0x0FB550	00	00	00	00	00	01	00	00	00	74	00	00	00	00	00	00	.........t......
0x0FB560	00	00	00	01	00	00	00	08	00	00	00	00	00	00	00	00	....
0x0FB570	00	01	00	00	00	01	00	00	00	00	00	00	00	00	00	01	....

Data Analysis: Confirmation

- Convert 58 8A to actual vehicle speed number:
- 58 8A to Decimal: 22,666
- Convert 22,666 to actual number:

  22,666 / 256 = 88.54 mph ~ 89 mph

Record 1

Time (Seconds)	Vehicle Speed (mph)	Engine Speed (rpm)	Engine Load (%)	Throttle (%)	Brake Status	Clutch Status	Cruise Status	Lamp Status
-16	150	0	0.0	0.0	-	-	-	On
-15	150	0	0.0	0.0	-	-	-	On
-14	150	0	0.0	0.0	-	-	-	On
-13	150	0	0.0	0.0	-	-	-	On
-12	150	0	0.0	0.0	-	-	-	On
-11	150	0	0.0	0.0	-	-	-	On
-10	150	0	0.0	0.0	-	-	-	On
-9	150	0	0.0	0.0	-	-	-	On
-8	150	0	0.0	0.0	-	-	-	On
-7	150	0	0.0	0.0	-	-	-	On
-6	150	0	0.0	0.0	-	-	-	On
-5	150	0	0.0	0.0	-	-	-	On
-4	150	0	0.0	0.0	-	-	-	On
-3	150	0	0.0	0.0	-	-	-	On
-2	150	0	0.0	0.0	-	-	-	On
-1	150	0	0.0	0.0	-	-	-	On
0	150	0	0.0	0.0	-	-	-	On
1	89	0	0.0	0.0	-	-	-	On
2	6	0	0.0	0.0	-	-	-	On

Data Analysis

- What about engine RPM, load, throttle, etc.?
- The 14-byte block in Record 2 data, this has more than just 150 mph

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
0x0FB950	00	00	00	02	14	F8	12	BD	00	00	00	00	00	00	00	00	.....ø.½........
0x0FB960	00	03	01	81	12	7C	00	00	00	00	00	00	00	00	00	02	.....|..........
0x0FB970	00	1B	12	BF	00	09	00	00	00	00	00	00	00	02	00	02	...¿. ....
0x0FB980	11	3E	00	21	08	C4	00	00	00	00	00	02	00	00	11	42	.>.!.Ä.........B
0x0FB990	00	2C	29	03	00	00	00	00	00	02	00	5C	11	E7	00	3E	.,)........\.ç.>
0x0FB9A0	42	8E	00	00	00	00	00	02	02	7D	19	38	00	76	10	8F	B........}.8.v..
0x0FB9B0	00	00	00	00	00	00	03	AB	15	37	00	3E	00	00	00	00	.......«.7.>....
0x0FB9C0	00	00	00	00	04	0A	17	82	00	94	20	B5	00	00	00	00	.... µ....
0x0FB9D0	00	00	04	C1	21	E3	00	A4	1F	7B	00	00	00	00	00	02	...Á!ã.¤ {......
0x0FB9E0	05	96	1A	5C	00	00	00	00	00	00	00	00	00	02	05	E8	...\...........è
0x0FB9F0	1A	40	00	81	0E	9D	00	00	00	00	00	00	06	72	21	77	.@..........r!w
0x0FBA00	00	B6	26	8D	00	00	00	00	00	00	07	BE	26	98	00	99	.¶&.......¾&....
0x0FBA10	17	84	00	00	00	00	00	00	08	75	28	36	00	20	00	00	.........u(6. ..
0x0FBA20	00	00	00	00	00	02	08	6F	1D	BF	00	5B	00	00	00	00	.......o ¿.[....
0x0FBA30	00	00	1E	FA	15	9B	78	6F	12	F7	01	D2	79	DC	00	00	.. ú..xo.÷.ÒyÜ..
0x0FBA40	96	00	00	00	00	00	00	00	00	00	00	01	00	00	96	00	....
0x0FBA50	00	00	00	00	00	00	00	00	00	01	00	00	96	00	00	00	....

Data Analysis: Result

Byte	3 & 4	5 & 6	7 & 8	9 & 10
Hex	07 BE	26 98	00 99	17 84
Convert to Decimal	1,982	9,880	153	6020
Resolution	1/256 mph/bit	1/8 RPM/bit	1/4 %/bit	1/256 %/bit
Actual Number	7.74 mph	1,235 RPM	38.25%	23.52%
	Vehicle Speed	Engine Speed	Throttle	Engine Load

Record 2

Time (Seconds)	Vehicle Speed (mph)	Engine Speed (rpm)	Engine Load (%)	Throttle (%)	Brake Status	Clutch Status	Cruise Status	Lamp Status
11	6	840	14.6	32.3	-	On	-	-
12	6	1071	38.6	45.5	-	-	-	-
13	8	1235	23.5	38.3	-	-	-	-
14	8	1287	0.0	8.0	-	-	-	-
15	8	952	0.0	22.8	-	On	-	-

99

Data Analysis: Alternative Approach

Another approach to locate sudden deceleration data in binary file:

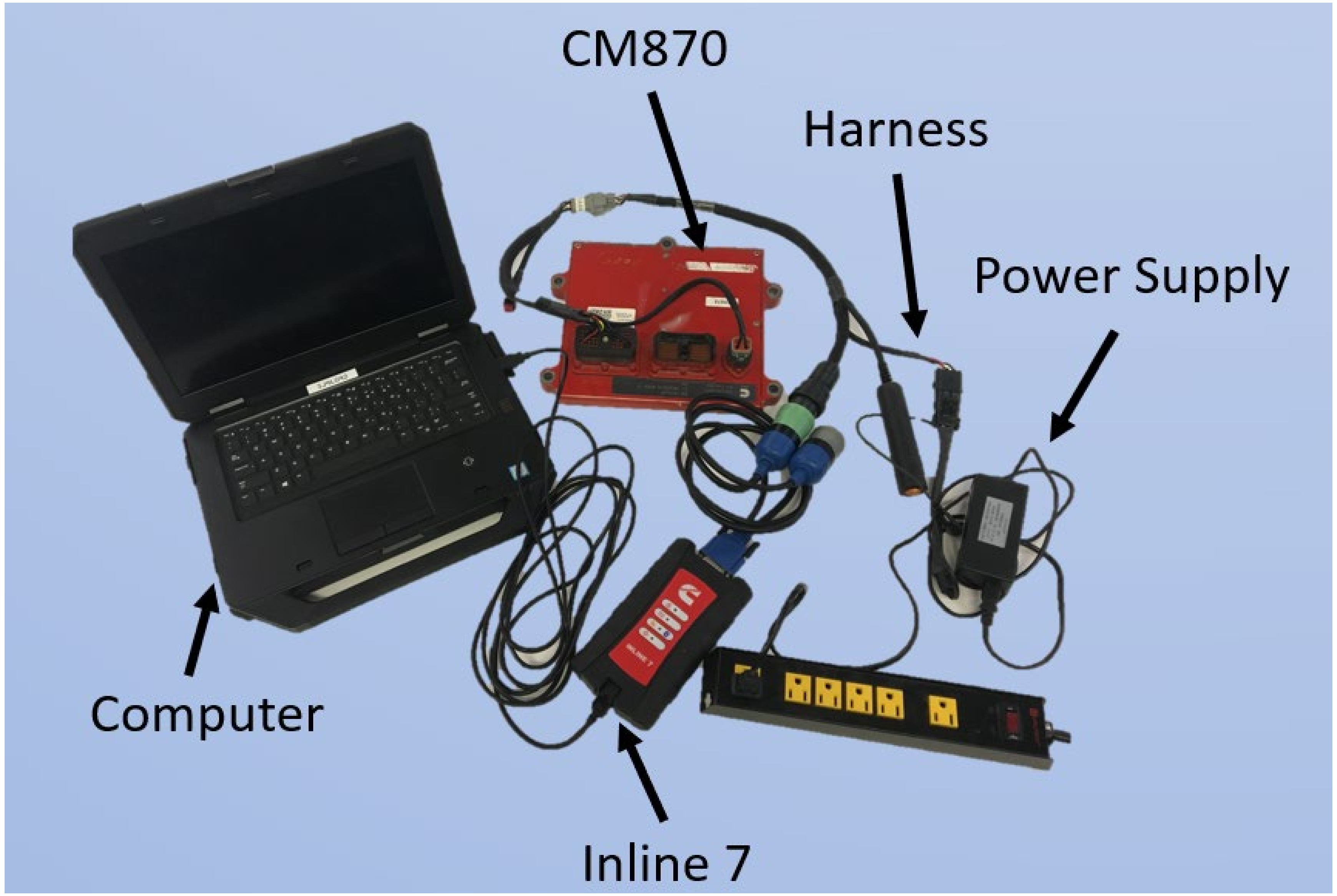
- Log CAN traffic when downloading sudden deceleration report with Cummins PowerSpec
- Compare the log file with binary to find the pattern.

Tools

Photo from https://www.diesellaptops.com/collections/cummins/products/cummins-inline-7-data-link-adapter

Connecting Harness

Setup



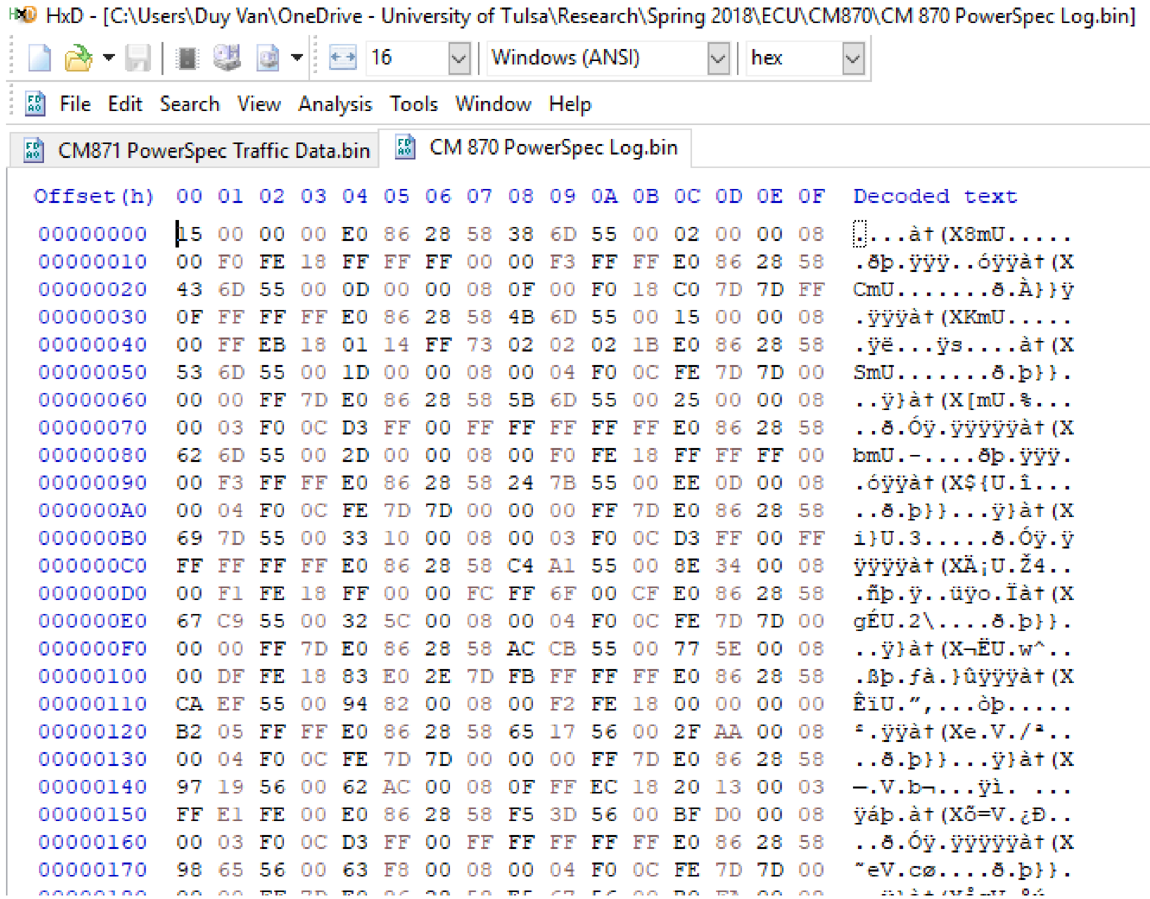
NMFTA CAN Logger

National Motor Freight Traffic Association and National Science Foundation CAN Data Collection Project

Open Source at:
https://www.github.com/SystemsCyber/CAN-Logger-3/

Downloading Data

CAN traffic was also logged while downloading



HxD - [C:\Users\Duy Van\OneDrive - University of Tulsa\Research\Spring 2018\ECU\CM870\CM 870 PowerSpec Log.bin]

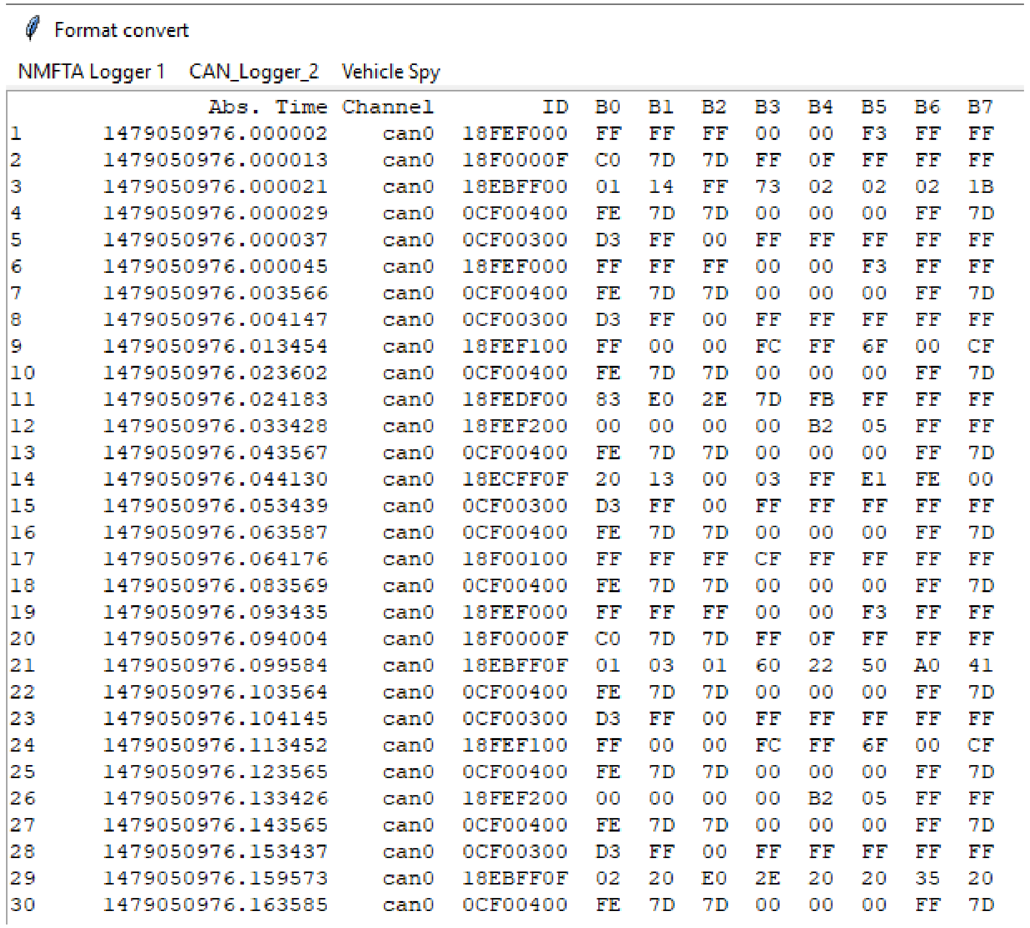
16 | Windows (ANSI) | hex

File Edit Search View Analysis Tools Window Help

CM871 PowerSpec Traffic Data.bin | CM 870 PowerSpec Log.bin

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	15	00	00	00	E0	86	28	58	38	6D	55	00	02	00	00	08	....àt(X8mU.....
00000010	00	F0	FE	18	FF	FF	FF	00	00	F3	FF	FF	E0	86	28	58	.ðþ.ÿÿÿ..óÿÿàt(X
00000020	43	6D	55	00	0D	00	00	08	0F	00	F0	18	C0	7D	7D	FF	CmU.......ð.À}}ÿ
00000030	0F	FF	FF	FF	E0	86	28	58	4B	6D	55	00	15	00	00	08	.ÿÿÿàt(XKmU.....
00000040	00	FF	EB	18	01	14	FF	73	02	02	02	1B	E0	86	28	58	.ÿë...ÿs....àt(X
00000050	53	6D	55	00	1D	00	00	08	00	04	F0	0C	FE	7D	7D	00	SmU.......ð.þ}}.
00000060	00	00	FF	7D	E0	86	28	58	5B	6D	55	00	25	00	00	08	..ÿ}àt(X[mU.%...
00000070	00	03	F0	0C	D3	FF	00	FF	FF	FF	FF	FF	E0	86	28	58	..ð.Óÿ.ÿÿÿÿÿàt(X
00000080	62	6D	55	00	2D	00	00	08	00	F0	FE	18	FF	FF	FF	00	bmU.-....ðþ.ÿÿÿ.
00000090	00	F3	FF	FF	E0	86	28	58	24	7B	55	00	EE	0D	00	08	.óÿÿàt(X${U.î...
000000A0	00	04	F0	0C	FE	7D	7D	00	00	00	FF	7D	E0	86	28	58	..ð.þ}}...ÿ}àt(X
000000B0	69	7D	55	00	33	10	00	08	00	03	F0	0C	D3	FF	00	FF	i}U.3.....ð.Óÿ.ÿ
000000C0	FF	FF	FF	FF	E0	86	28	58	C4	A1	55	00	8E	34	00	08	ÿÿÿÿàt(XÄ¡U.Ž4..
000000D0	00	F1	FE	18	FF	00	00	FC	FF	6F	00	CF	E0	86	28	58	.ñþ.ÿ..üÿo.Ïàt(X
000000E0	67	C9	55	00	32	5C	00	08	00	04	F0	0C	FE	7D	7D	00	gÉU.2\....ð.þ}}.
000000F0	00	00	FF	7D	E0	86	28	58	AC	CB	55	00	77	5E	00	08	..ÿ}àt(X¬ËU.w^..
00000100	00	DF	FE	18	83	E0	2E	7D	FB	FF	FF	FF	E0	86	28	58	.ßþ.ƒà.}ûÿÿÿàt(X
00000110	CA	EF	55	00	94	82	00	08	00	F2	FE	18	00	00	00	00	ÊïU.”,...òþ.....
00000120	B2	05	FF	FF	E0	86	28	58	65	17	56	00	2F	AA	00	08	².ÿÿàt(Xe.V./ª..
00000130	00	04	F0	0C	FE	7D	7D	00	00	00	FF	7D	E0	86	28	58	..ð.þ}}...ÿ}àt(X
00000140	97	19	56	00	62	AC	00	08	0F	FF	EC	18	20	13	00	03	—.V.b¬...ÿì. ...
00000150	FF	E1	FE	00	E0	86	28	58	F5	3D	56	00	BF	D0	00	08	ÿáþ.àt(Xõ=V.¿Ð..
00000160	00	03	F0	0C	D3	FF	00	FF	FF	FF	FF	FF	E0	86	28	58	..ð.Óÿ.ÿÿÿÿÿàt(X
00000170	98	65	56	00	63	F8	00	08	00	04	F0	0C	FE	7D	7D	00	˜eV.cø....ð.þ}}.

CAN traffic logged in raw format



Format convert

NMFTA Logger 1 | CAN_Logger_2 | Vehicle Spy

	Abs. Time	Channel	ID	B0	B1	B2	B3	B4	B5	B6	B7
1	1479050976.000002	can0	18FEF000	FF	FF	FF	00	00	F3	FF	FF
2	1479050976.000013	can0	18F0000F	C0	7D	7D	FF	0F	FF	FF	FF
3	1479050976.000021	can0	18EBFF00	01	14	FF	73	02	02	02	1B
4	1479050976.000029	can0	0CF00400	FE	7D	7D	00	00	00	FF	7D
5	1479050976.000037	can0	0CF00300	D3	FF	00	FF	FF	FF	FF	FF
6	1479050976.000045	can0	18FEF000	FF	FF	FF	00	00	F3	FF	FF
7	1479050976.003566	can0	0CF00400	FE	7D	7D	00	00	00	FF	7D
8	1479050976.004147	can0	0CF00300	D3	FF	00	FF	FF	FF	FF	FF
9	1479050976.013454	can0	18FEF100	FF	00	00	FC	FF	6F	00	CF
10	1479050976.023602	can0	0CF00400	FE	7D	7D	00	00	00	FF	7D
11	1479050976.024183	can0	18FEDF00	83	E0	2E	7D	FB	FF	FF	FF
12	1479050976.033428	can0	18FEF200	00	00	00	00	B2	05	FF	FF
13	1479050976.043567	can0	0CF00400	FE	7D	7D	00	00	00	FF	7D
14	1479050976.044130	can0	18ECFF0F	20	13	00	03	FF	E1	FE	00
15	1479050976.053439	can0	0CF00300	D3	FF	00	FF	FF	FF	FF	FF
16	1479050976.063587	can0	0CF00400	FE	7D	7D	00	00	00	FF	7D
17	1479050976.064176	can0	18F00100	FF	FF	FF	CF	FF	FF	FF	FF
18	1479050976.083569	can0	0CF00400	FE	7D	7D	00	00	00	FF	7D
19	1479050976.093435	can0	18FEF000	FF	FF	FF	00	00	F3	FF	FF
20	1479050976.094004	can0	18F0000F	C0	7D	7D	FF	0F	FF	FF	FF
21	1479050976.099584	can0	18EBFF0F	01	03	01	60	22	50	A0	41
22	1479050976.103564	can0	0CF00400	FE	7D	7D	00	00	00	FF	7D
23	1479050976.104145	can0	0CF00300	D3	FF	00	FF	FF	FF	FF	FF
24	1479050976.113452	can0	18FEF100	FF	00	00	FC	FF	6F	00	CF
25	1479050976.123565	can0	0CF00400	FE	7D	7D	00	00	00	FF	7D
26	1479050976.133426	can0	18FEF200	00	00	00	00	B2	05	FF	FF
27	1479050976.143565	can0	0CF00400	FE	7D	7D	00	00	00	FF	7D
28	1479050976.153437	can0	0CF00300	D3	FF	00	FF	FF	FF	FF	FF
29	1479050976.159573	can0	18EBFF0F	02	20	E0	2E	20	20	35	20
30	1479050976.163585	can0	0CF00400	FE	7D	7D	00	00	00	FF	7D

CAN traffic logged in readable format

CAN Data Logging Results

There is pattern for the Sudden Deceleration records

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B
000F31BC	00	00	00	00	00	00	00	00	00	00	00	00
000F31C8	00	00	00	00	00	00	00	00	00	00	00	00
000F31D4	00	00	00	00	0C	31	00	00	00	00	00	00
000F31E0	00	00	00	73	00	02	1E	FA	15	9B	85	8A
000F31EC	12	F6	01	D2	79	DC	00	00	96	00	00	00
000F31F8	00	00	00	00	00	00	00	01	00	00	96	00
000F3204	00	00	00	00	00	00	00	00	00	01	00	00
000F3210	96	00	00	00	00	00	00	00	00	00	00	01
000F321C	00	00	96	00	00	00	00	00	00	00	00	00
000F3228	00	01	00	00	96	00	00	00	00	00	00	00
000F3234	00	00	00	01	00	00	96	00	00	00	00	00
000F3240	00	00	00	00	00	01	00	00	96	00	00	00
000F324C	00	00	00	00	00	00	00	01	00	00	96	00

Raw binary data from KTAG extraction

1479051111.967139	can0	18EBFF00	08	04	AF	00	03	03	6C	00
1479051111.976171	can0	18EBFA00	01	45	1A	A0	00	00	04	26
1479051111.986092	can0	18EBFA00	02	1E	FA	15	9B	85	8A	12
1479051111.996095	can0	18EBFA00	03	F6	01	D2	79	DC	00	00
1479051112.006143	can0	18EBFA00	04	96	00	00	00	00	00	00
1479051112.016150	can0	18EBFA00	05	00	00	00	00	01	00	00
1479051112.026145	can0	18EBFA00	06	96	00	00	00	00	00	00
1479051112.026713	can0	18EBFF00	09	04	01	66	00	04	01	94
1479051112.036134	can0	18EBFA00	07	00	00	00	00	01	00	00
1479051112.046148	can0	18EBFA00	08	96	00	00	00	00	00	00
1479051112.056151	can0	18EBFA00	09	00	00	00	00	01	00	00
1479051112.066164	can0	18EBFA00	0A	96	00	00	00	00	00	00
1479051112.076132	can0	18EBFA00	0B	00	00	00	00	01	00	00
1479051112.086140	can0	18EBFA00	0C	96	00	00	00	00	00	00
1479051112.086713	can0	18EBFF00	0A	04	03	01	B9	04	04	03
1479051112.096128	can0	18EBFA00	0D	00	00	00	00	01	00	00
1479051112.106137	can0	18EBFA00	0E	96	00	00	00	00	00	00
1479051112.116171	can0	18EBFA00	0F	00	00	00	00	01	00	00

Logged CAN traffic data, filtered for J1939 Transport Protocol messages

Interpreter Python Source Code Snippet

Attribution *Data*

```
with open(filename,'rb') as binfile:
    raw_data = binfile.read()

#Extract run time, occurrence distance, and temp of 3 records
ECM_run_time1 = struct.unpack(">L",raw_data[record1:record1+4])[0]/4
Occurrence_distance1 = struct.unpack(">L",raw_data[record1+20:record1+24])[0]/205.99605
Temp1 = (((((struct.unpack(">L",raw_data[record1+18:record1+22])[0]) & 0xFFFF0000 )>> 16)/128)*9/5)+32

ECM_run_time2 = struct.unpack(">L",raw_data[record2:record2+4])[0]/4
Occurrence_distance2 = struct.unpack(">L",raw_data[record2+20:record2+24])[0]/205.99605
Temp2 = (((((struct.unpack(">L",raw_data[record2+18:record2+22])[0]) & 0xFFFF0000 )>> 16)/128)*9/5)+32

ECM_run_time3 = struct.unpack(">L",raw_data[record3:record3+4])[0]/4
Occurrence_distance3 = struct.unpack(">L",raw_data[record3+20:record3+24])[0]/205.99605
Temp3 = (((((struct.unpack(">L",raw_data[record3+18:record3+22])[0]) & 0xFFFF0000 )>> 16)/128)*9/5)+32
```

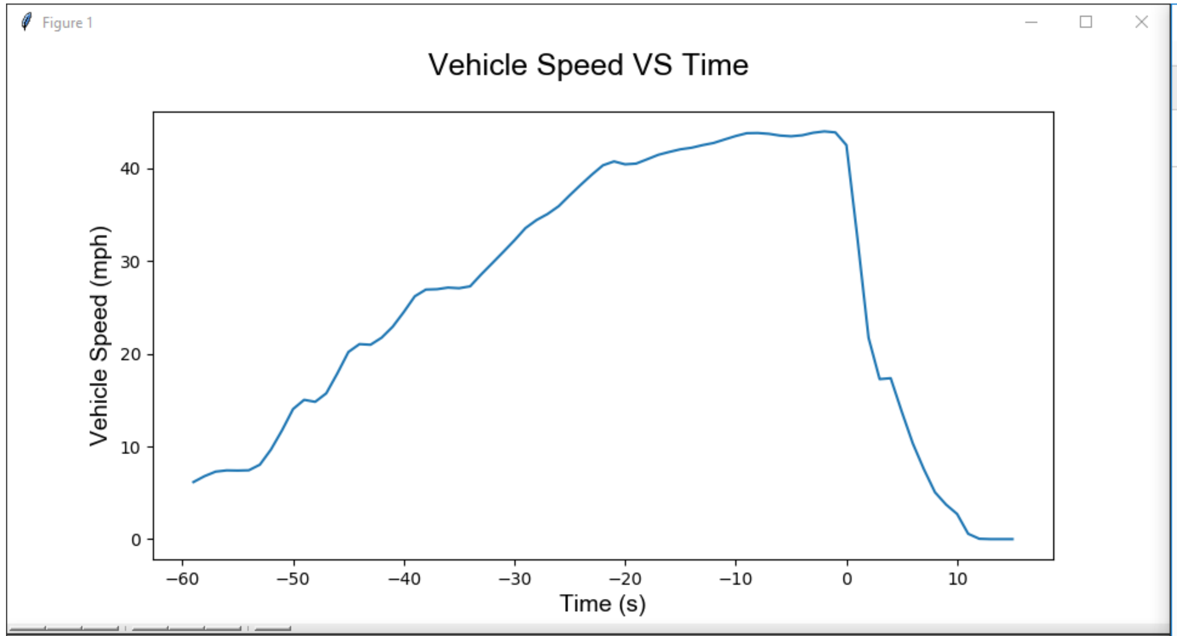
Attribution Result

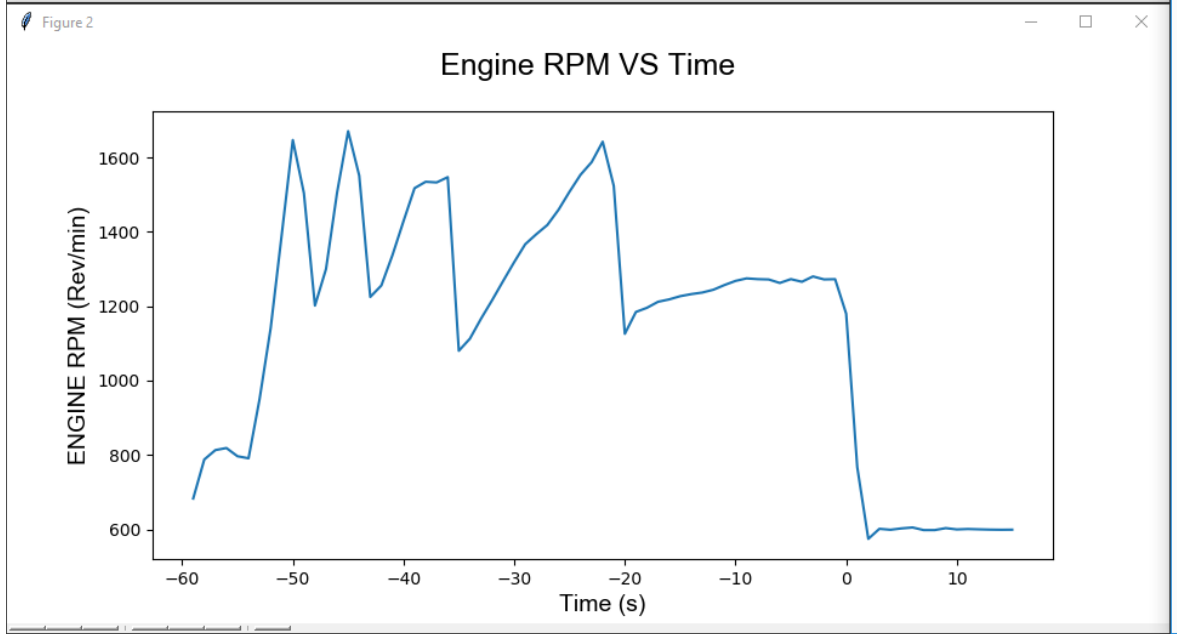
```
Record 1 ECM occurrence run time (s): 10437516.0
Record 1 ECM occurrence distance (mile): 109323.35838478456
Record 1 ECM occurrence air temperature (F): 71.9234375

Record 2 ECM occurrence run time (s): 29454947.0
Record 2 ECM occurrence distance (mile): 250785.41069112733
Record 2 ECM occurrence air temperature (F): 55.3015625

Record 3 ECM occurrence run time (s): 6186154.0
Record 3 ECM occurrence distance (mile): 66872.2371516104
Record 3 ECM occurrence air temperature (F): 117.359375
```

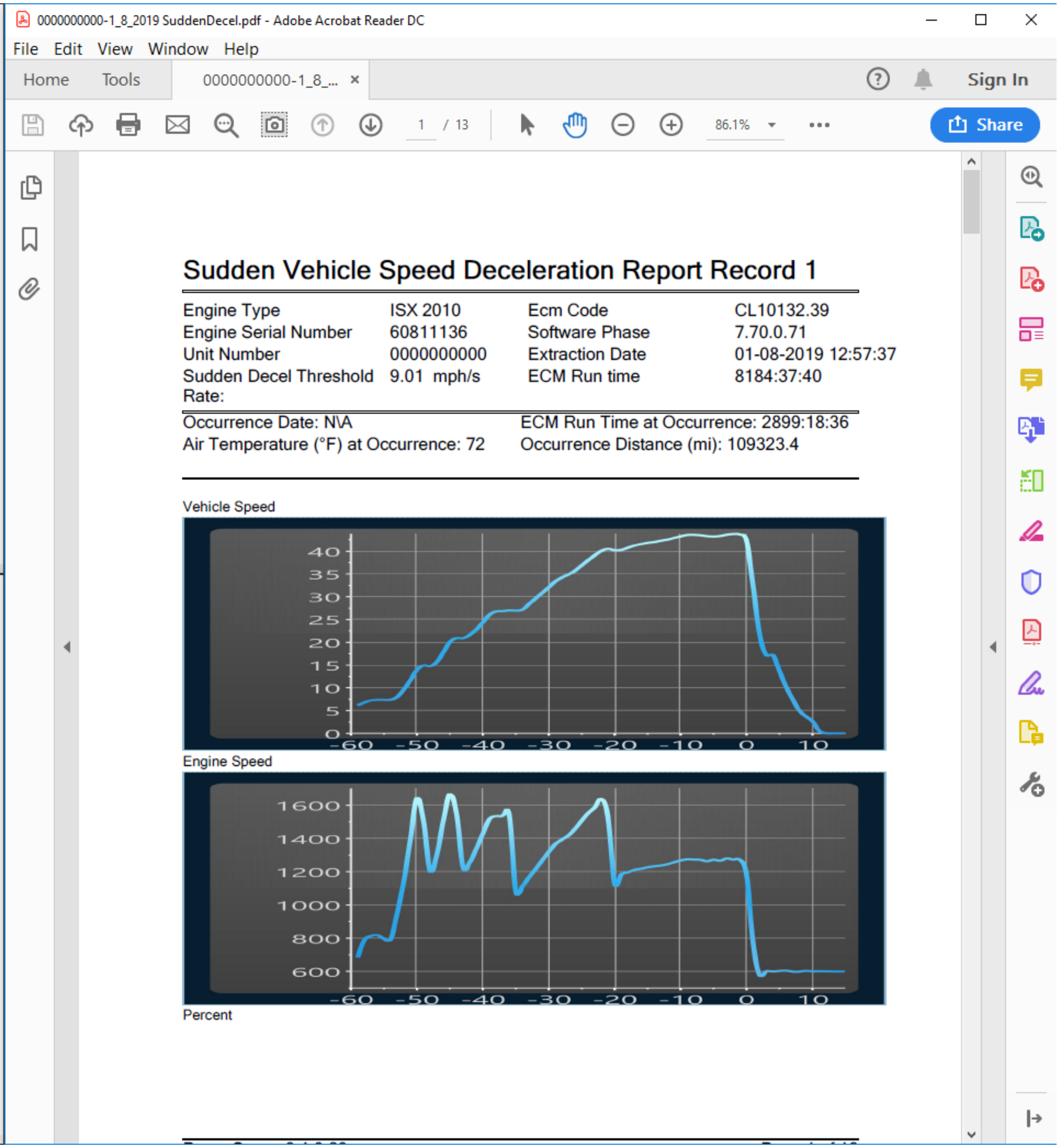
Python Code for Sudden Decel Records

```
#Parsing through the raw data
for i in range(int((record_size-header_size)/block_size)):
    speed1 = ((struct.unpack(">L",raw_data[record1+24+i*block_size : record1+28+i*block_size])[0]) & 0x0000FFFF )/102.455
    rpm1 = ((struct.unpack(">L",raw_data[record1+28+i*block_size : record1+32+i*block_size])[0]) & 0x0000FFFF )/8

    speed2 = ((struct.unpack(">L",raw_data[record2+24+i*block_size : record2+28+i*block_size])[0]) & 0x0000FFFF )/102.455
    rpm2 = ((struct.unpack(">L",raw_data[record2+28+i*block_size : record2+32+i*block_size])[0]) & 0x0000FFFF )/8

    speed3 = ((struct.unpack(">L",raw_data[record3+24+i*block_size : record3+28+i*block_size])[0]) & 0x0000FFFF )/102.455
    rpm3 = ((struct.unpack(">L",raw_data[record3+28+i*block_size : record3+32+i*block_size])[0]) & 0x0000FFFF )/8

    time = -59+i
```

Graphical Result From Binary Interpretation







Sudden Vehicle Speed Deceleration Report Record 1

Engine Type	ISX 2010	Ecm Code	CL10132.39
Engine Serial Number	60811136	Software Phase	7.70.0.71
Unit Number	0000000000	Extraction Date	01-08-2019 12:57:37
Sudden Decel Threshold Rate:	9.01 mph/s	ECM Run time	8184:37:40

Occurrence Date: N\A
Air Temperature (°F) at Occurrence: 72
ECM Run Time at Occurrence: 2899:18:36
Occurrence Distance (mi): 109323.4

Vehicle Speed

Engine Speed

Percent

Data Analysis: Missing Data

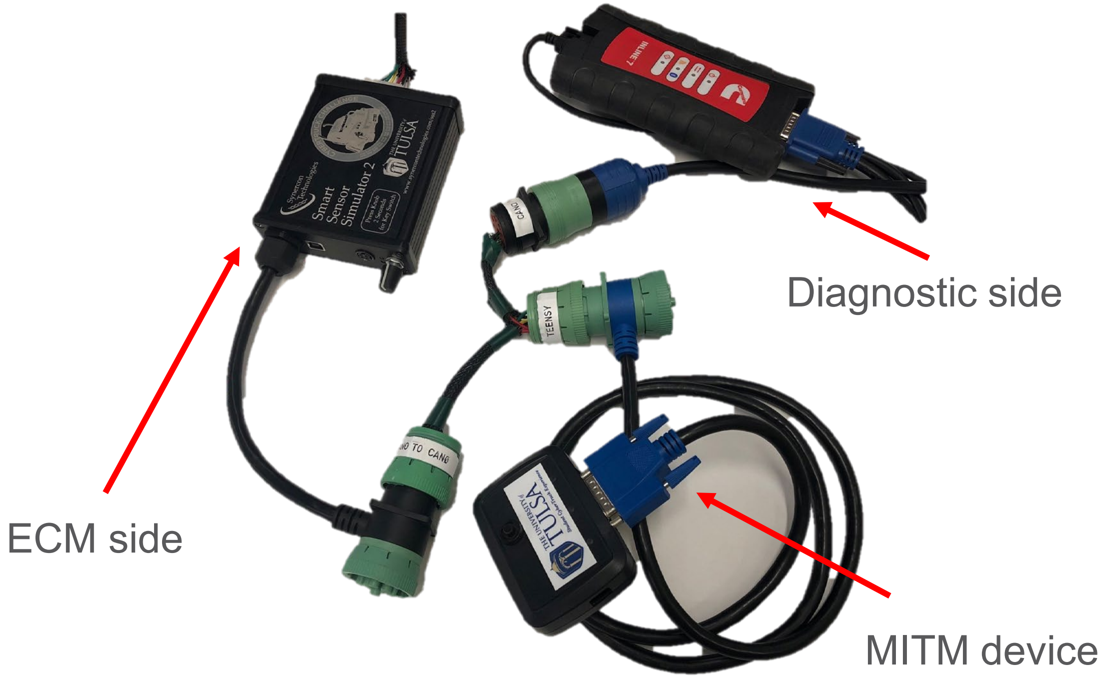
- Reverse Engineered only 8 bytes in the 14 byte block.
  - What do other bytes mean?
  - Which one is the brake status, clutch status, etc.?

Record 2

Time (Seconds)	Vehicle Speed (mph)	Engine Speed (rpm)	Engine Load (%)	Throttle (%)	Brake Status	Clutch Status	Cruise Status	Lamp Status
11	6	840	14.6	32.3	-	On	-	-
12	6	1071	38.6	45.5	-	-	-	[illegible]
13	8	1235	23.5	38.3	-	-	-	-
14	8	1287	0.0	8.0	-	-	-	-
15	8	952	0.0	22.8	-	On	[illegible]	-

Man In The Middle Attack

- Man In The Middle (MITM) attack for reverse engineering
    - Sudden deceleration data sent from the ECM will be modified by the MITM device to observe how the it changes the record



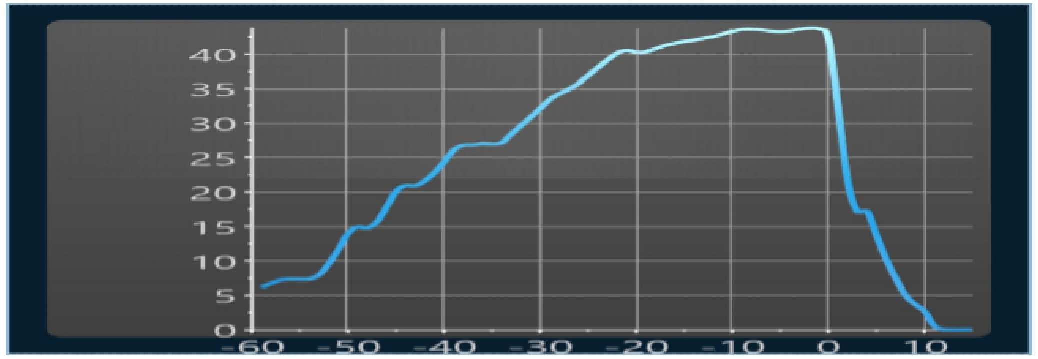
Machine-In-the-Middle (MITM) Attack

Sudden Vehicle Speed Deceleration Report Record 1

Engine Type	ISX 2010	Ecm Code	CL10132.39
Engine Serial Number	60811136	Software Phase	7.70.0.71
Unit Number	0000000000	Extraction Date	04-03-2019 04:37:51
Sudden Decel Threshold Rate:	N/A	ECM Run time	8227:56:32

Occurrence Date: N\A ECM Run Time at Occurrence: 2899:18:36
Air Temperature (°F) at Occurrence: 72 Occurrence Distance (mi): 109323.4

Vehicle Speed



Record 1

Time (Seconds)	Vehicle Speed (mph)	Engine Speed (rpm)	Engine Load (%)	Throttle (%)	Brake Status	Clutch Status	Cruise Status	Lamp Status
-59	6	683	34.8	30.1	-	-	-	-
-58	7	788	33.6	32.2	-	-	-	-
-57	7	813	21.9	27.5	-	-	-	-

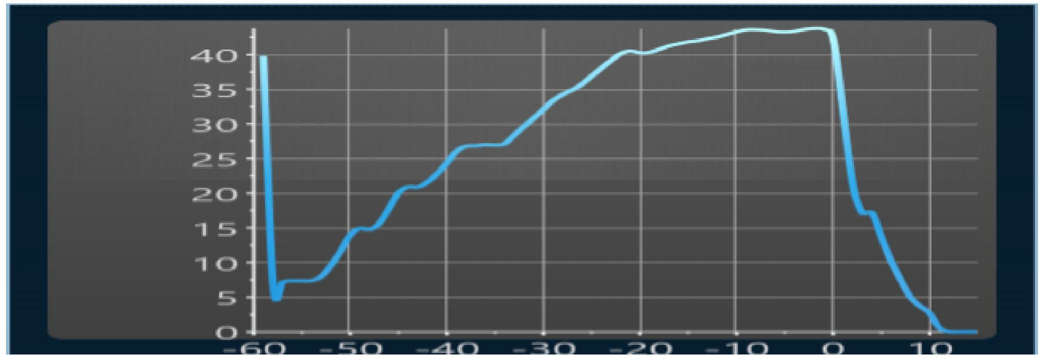
Genuine Data

Sudden Vehicle Speed Deceleration Report Record 1

Engine Type	ISX 2010	Ecm Code	CL10132.39
Engine Serial Number	60811136	Software Phase	7.70.0.71
Unit Number	0000000000	Extraction Date	04-03-2019 02:08:09
Sudden Decel Threshold Rate:	N/A	ECM Run time	8225:27:34

Occurrence Date: N\A ECM Run Time at Occurrence: 2899:18:36
Air Temperature (°F) at Occurrence: 72 Occurrence Distance (mi): 109323.4

Vehicle Speed

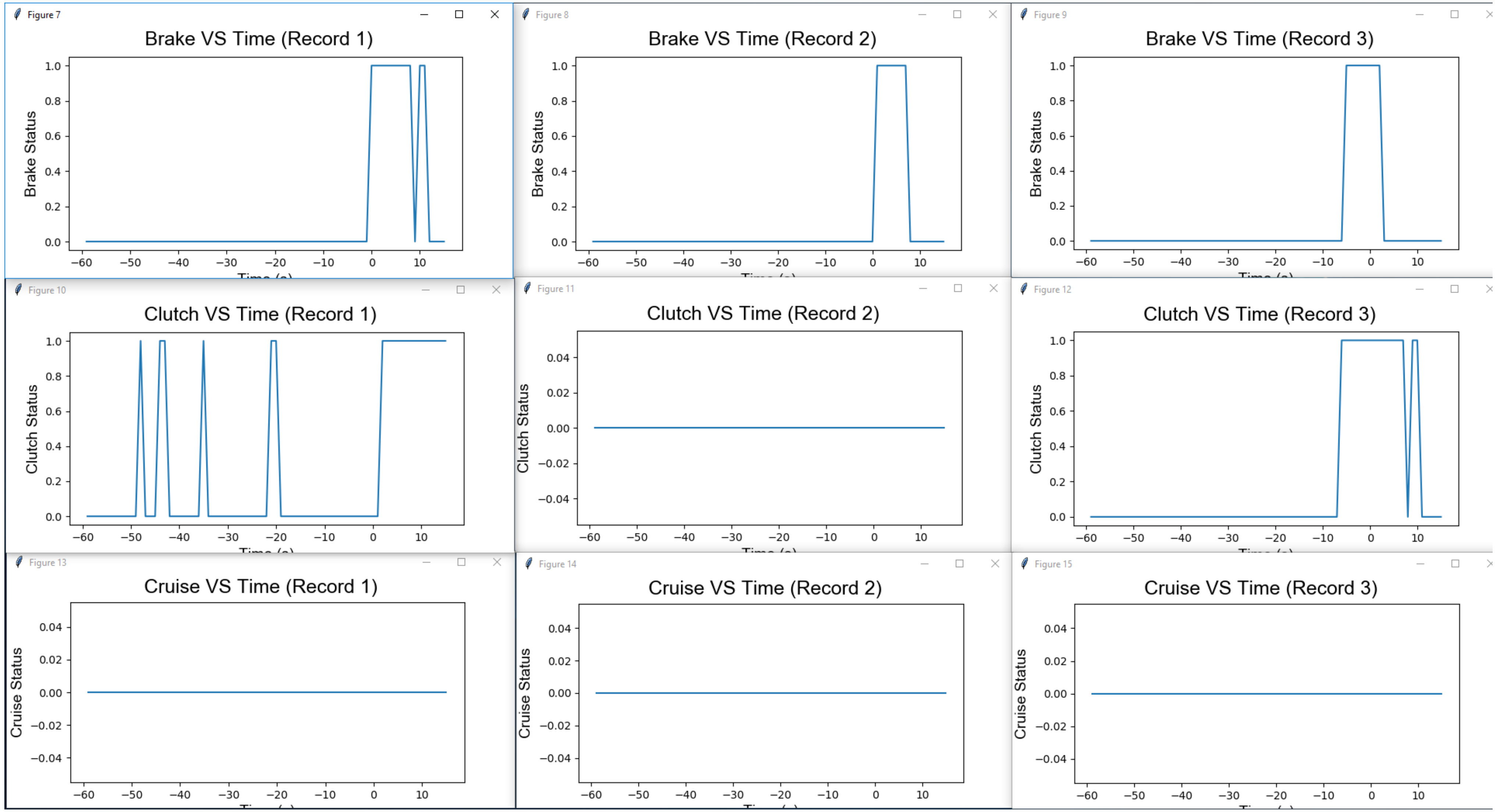


Record 1

Time (Seconds)	Vehicle Speed (mph)	Engine Speed (rpm)	Engine Load (%)	Throttle (%)	Brake Status	Clutch Status	Cruise Status	Lamp Status
-59	40	683	34.8	30.1	-	-	-	-
-58	7	788	33.6	32.2	-	-	-	-
-57	7	813	21.9	27.5	-	-	-	-

Attack Data

Brake, Clutch, Cruise Results



Data Manipulation

Forensic Soundness and Protecting Against

Editing Data in Hex

HxD - [C:\Users\dailyadmin\OneDrive - University of Tulsa\Research\Summer 2019\Florida Trip\back up - attaccked.MPC]

File Edit Search View Analysis Tools Window Help

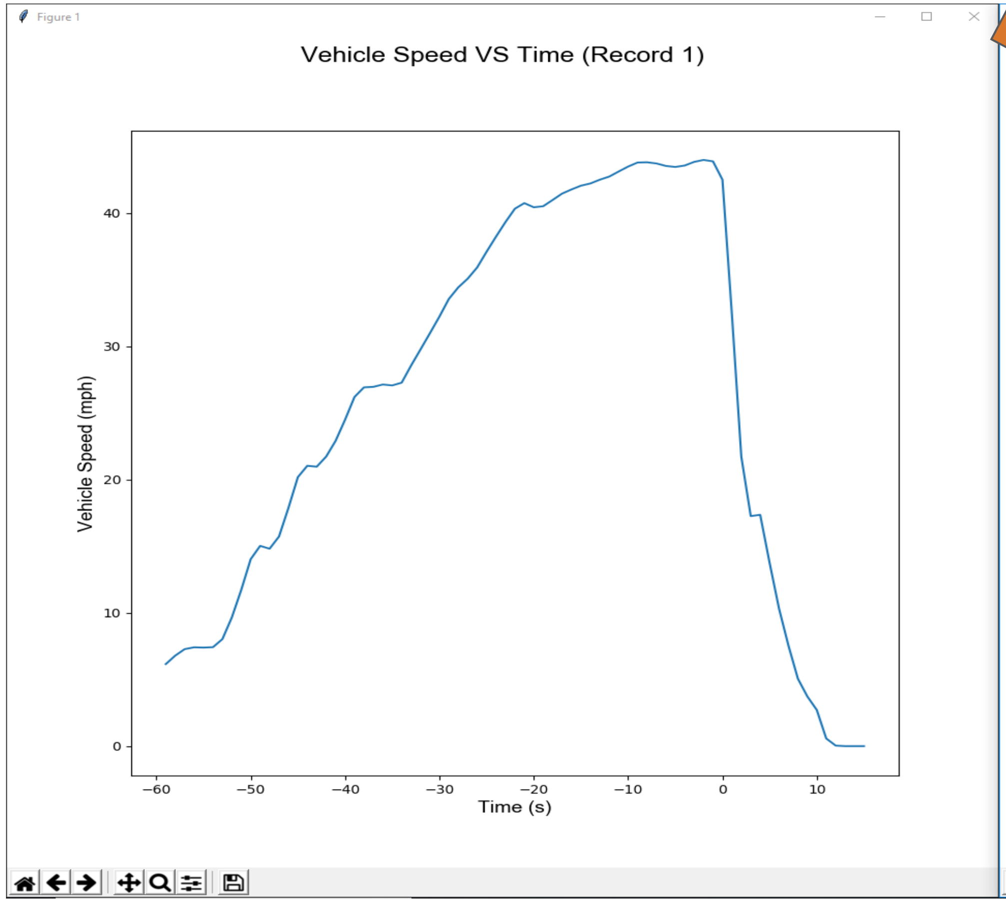
16 | Windows (ANSI) | hex

back up - attaccked.MPC

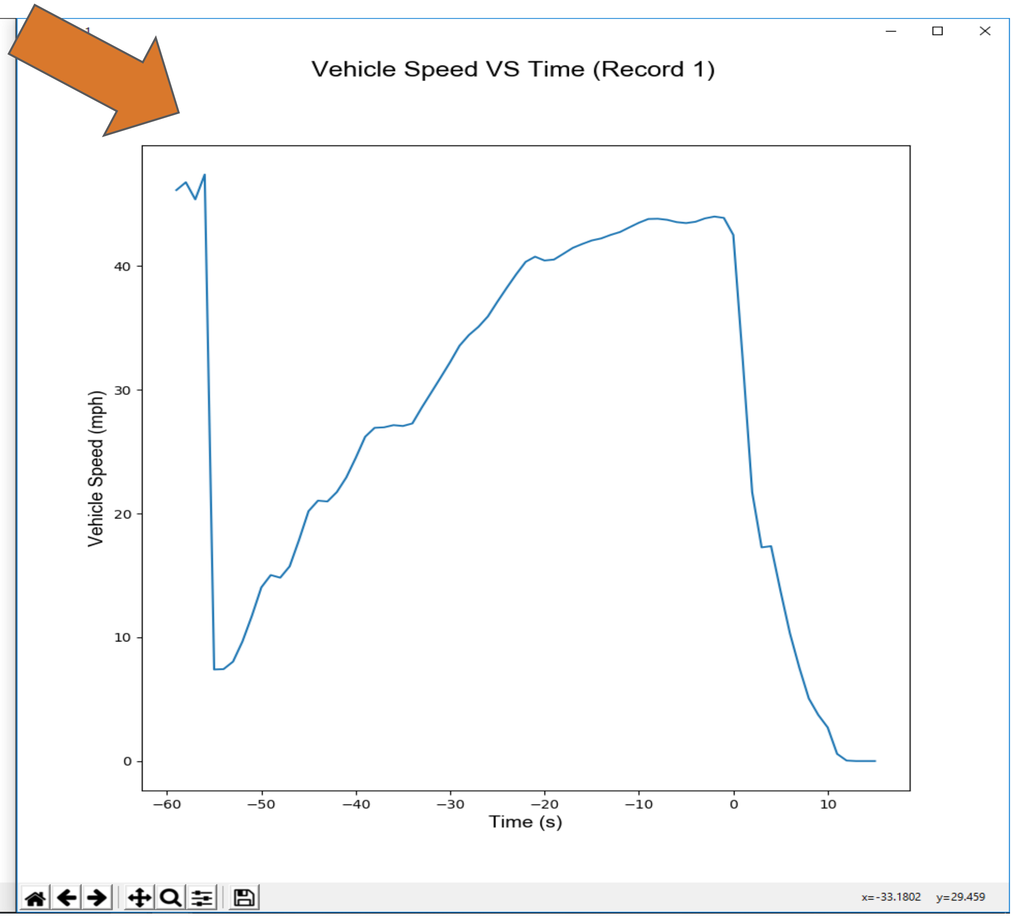
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00009510	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00009520	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00009530	1B	EB	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.ë..............
00009540	00	FF	FF	FF	02	7D	0E	30	FF	FF	FF	FF	01	00	00	01	.ÿÿÿ.}.0ÿÿÿÿ....
00009550	01	01	01	00	01	01	0B	17	01	57	A1	74	00	00	13	77	.........W¡t...w
00009560	00	00	15	58	00	00	1E	15	22	C4	00	00	00	00	13	B8	...X...."Ä.....¸
00009570	00	00	18	9F	00	00	20	35	21	98	00	00	00	00	13	2A	...Ÿ.. 5!˜.....*
00009580	00	00	19	69	00	00	1B	76	15	E0	00	00	00	00	13	F8	...i...v.à.....ø
00009590	00	00	19	96	00	00	16	0D	0A	8C	00	00	00	00	02	F6	...–.....Œ.....ö
000095A0	00	00	18	E5	00	00	0F	B9	00	00	00	00	00	00	02	F9	...å...¹.......ù
000095B0	00	00	18	B9	00	00	1D	EC	1A	90	00	00	00	00	03	37	...¹...ì.......7
000095C0	00	00	1D	B7	00	00	2C	7D	30	0C	00	00	00	00	03	DE	...·..,}0......Þ
000095D0	00	00	23	A9	00	00	33	0C	27	10	00	00	00	00	04	B2	..#©..3.'......²
000095E0	00	00	2B	85	00	00	3F	23	30	70	00	00	00	00	05	9E	..+…..?#0p.....ž
000095F0	00	00	33	76	00	00	40	94	2B	5C	00	00	00	00	06	04	..3v..@”+\......
00009600	00	00	2F	02	00	00	00	8E	00	00	00	00	00	00	05	EE	../....Ž.......î
00009610	00	00	25	8D	00	01	13	FB	00	00	00	00	00	00	06	4C	..%....û.......L

Manipulating Speed Records

Original Authentic Record



Manipulated Record



Forensic Soundness

- **Meaning** is a term that denotes confidence in the interpretation of extracted evidence data.
- **Error Detection** denotes processes for detecting or predicting errors in the forensic process.
- **Transparency** means the forensic process is documented, known, and verifiable.
- **Expertise** is required for investigators examining digital data.
- **Tampering detection** involves processes to evaluate if this has occurred.

Forensic Soundness: Applied

Tamper Detection

- Digitally sign the file with a secure hash algorithm

Goal: Verify mathematically the data has not been changed from its original form.

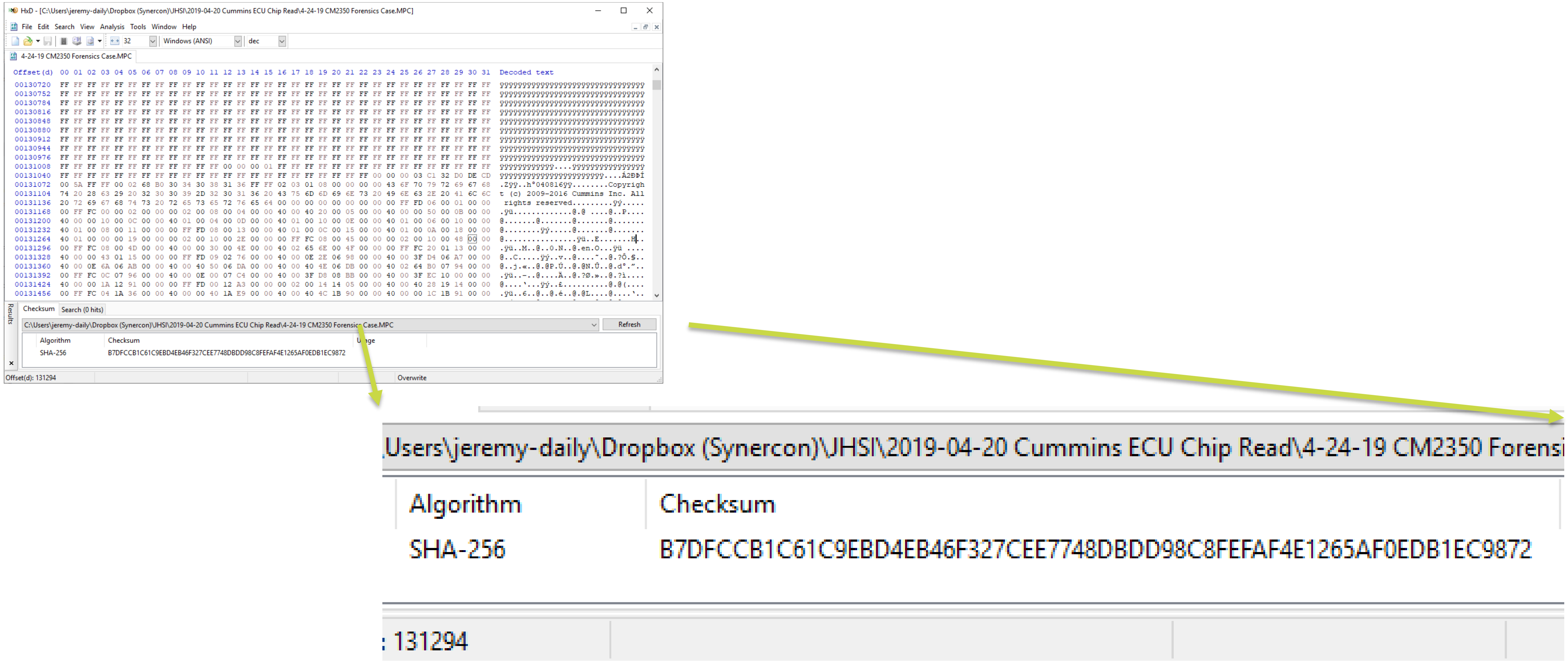
- Calculated hash values match each time file is used

Challenge: How to compute the original hash?

Cryptographic Hashing

- Secure Hash Algorithm (SHA-256)
  - Calculates a unique 256-bit number based on file contents
  - File contents cannot be determined from the hash
  - Unique for each file (i.e. little chance of a “collision”)
- Windows right click utilities:
  - https://www.nirsoft.net/utils/hash_my_files.html
  - https://www.tenforums.com/tutorials/78681-add-file-hash-context-menu-windows-8-10-a.html

Calculate Hash with Hex Editor Utility



Python Source Code: Do the hashes match?

C:\Users\jeremy-daily\Dropbox (Synercon)\JHSI\2019-04-20 Cummins ECU Chip Read\Cummins2350BinaryAnalysis.py - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help

Cummins2350BinaryAnalysis.py

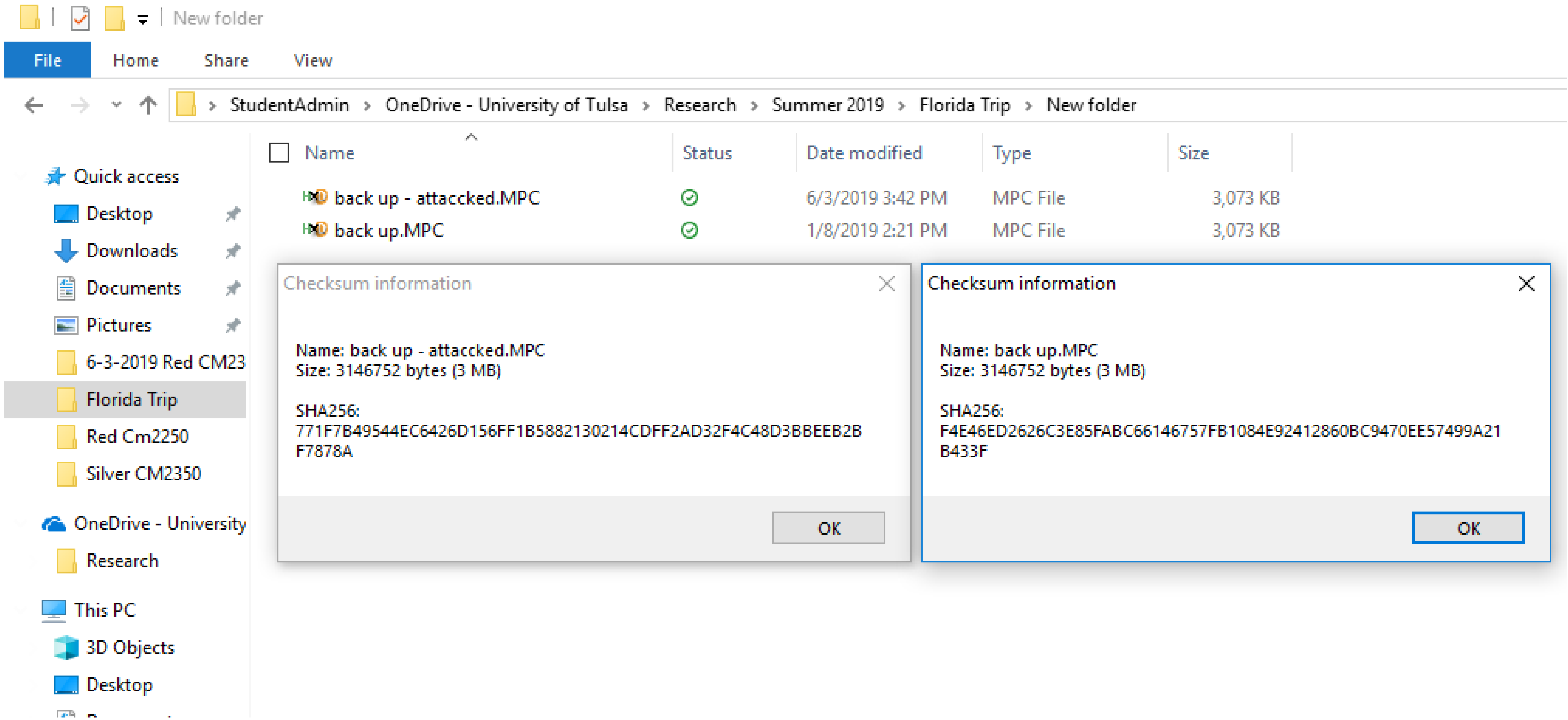
```python
from cryptography.hazmat.backends import default_backend
from cryptography.hazmat.primitives import hashes

#Calculate Hash Digest from file
with open("4-24-19 CM2350 Forensics Case.MPC",'rb') as file_pointer:
    binary_file_contents = file_pointer.read()
new_hash_digest = hashes.Hash(hashes.SHA256(), backend=default_backend())
new_hash_digest.update(binary_file_contents)
digest_string = "".join("{:02X}".format(c) for c in new_hash_digest.finalize())
print(digest_string)

#Compare to the old file
with open("SHA 256.txt",'r') as sha_file:
    original_hash_value = sha_file.read()
print(original_hash_value)
print(digest_string == original_hash_value)
```

```
B7DFCCB1C61C9EBD4EB46F327CEE7748DBDD98C8FEFAF4E1265AF0EDB1EC9872
B7DFCCB1C61C9EBD4EB46F327CEE7748DBDD98C8FEFAF4E1265AF0EDB1EC9872
True
[Finished in 0.3s]
```

Yes

Line 1, Column 1 Tab Size: 4 Python

Hash Values Do Not Match



Recommended Practice


ISB 6.7
220 HP
CPL
4569
6654

Recommended Practice

Take photos or video of opening the ECU.

Document video creation time.

Created at 9:09

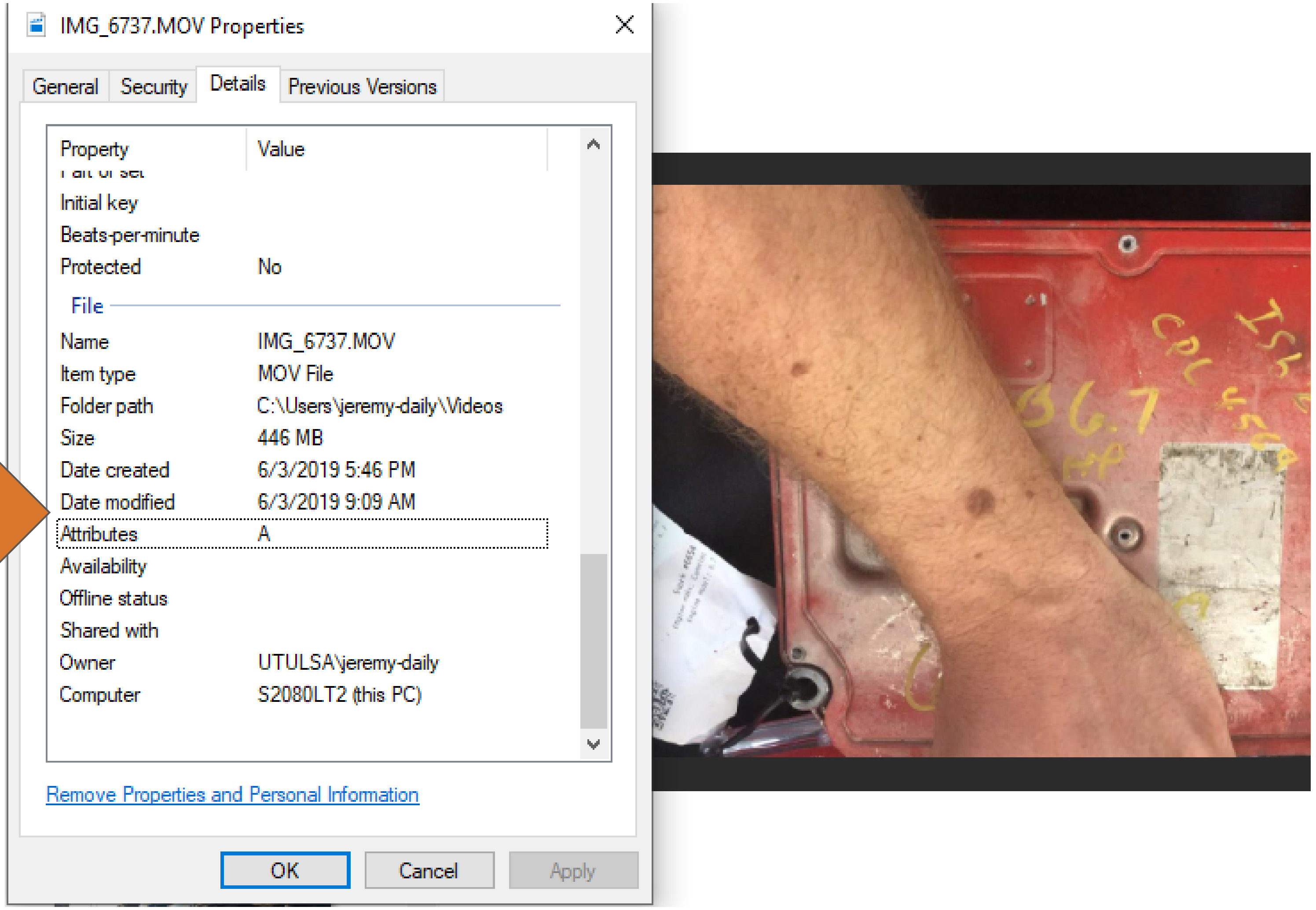
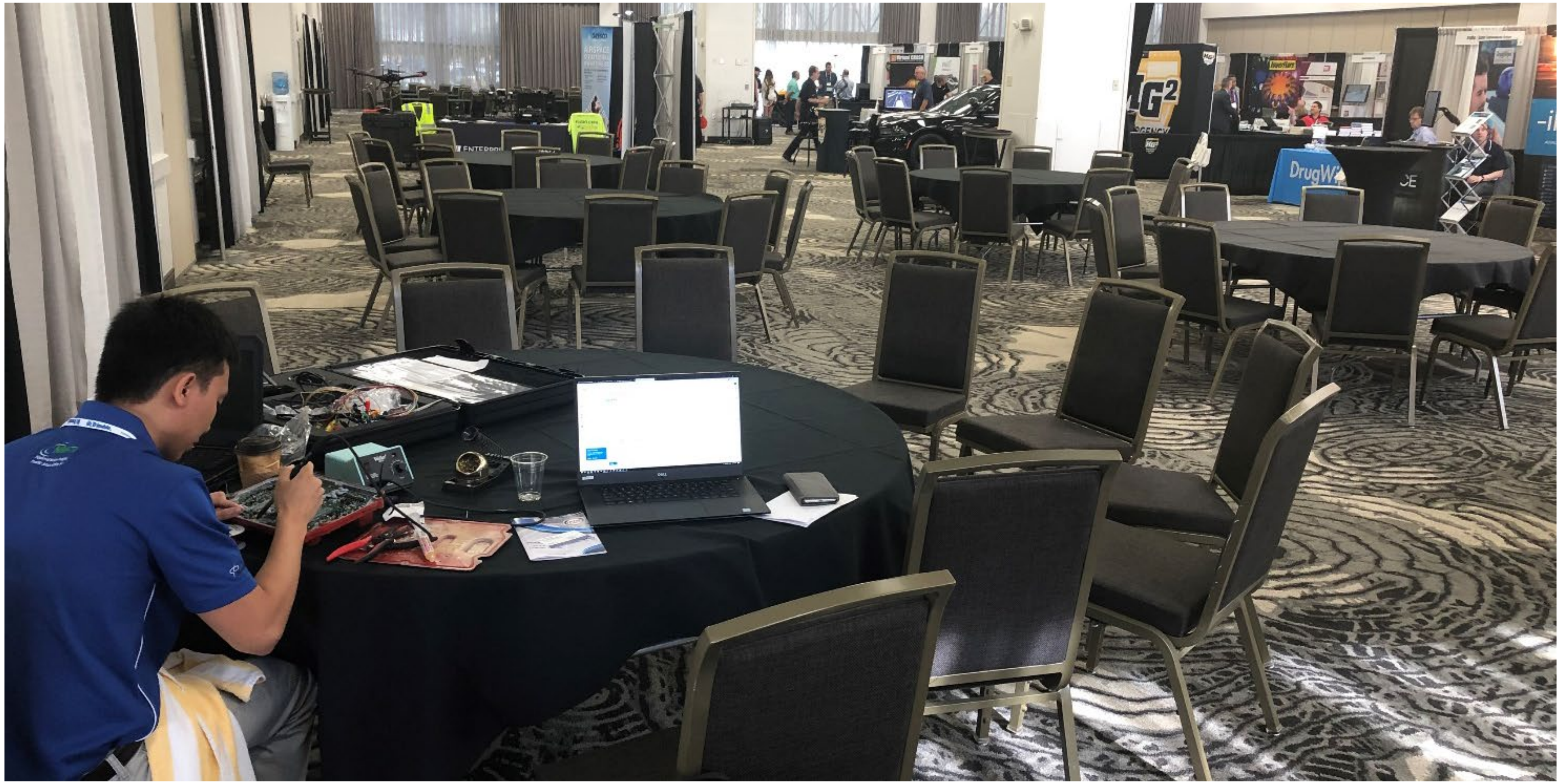
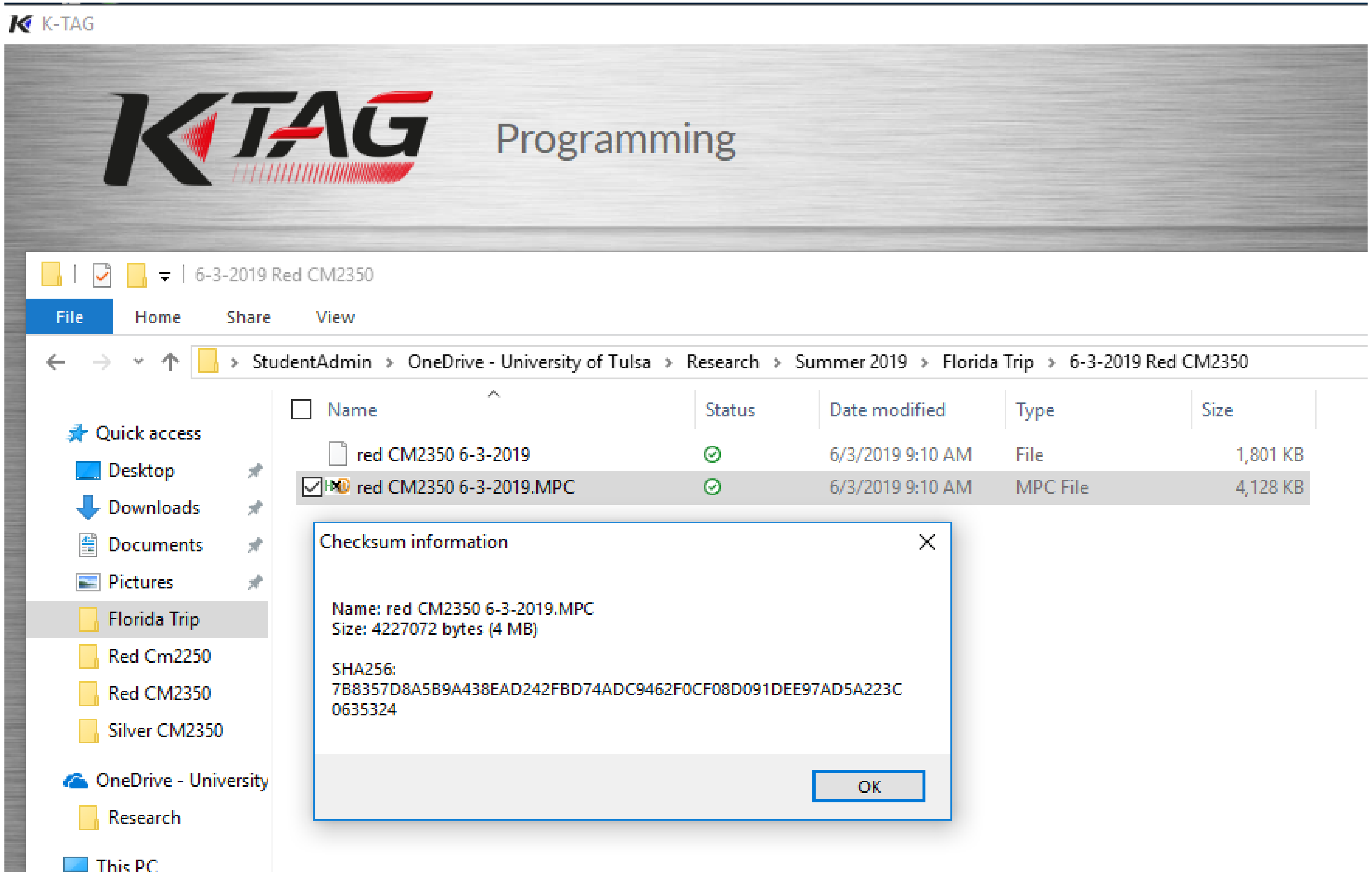


Image the ECU



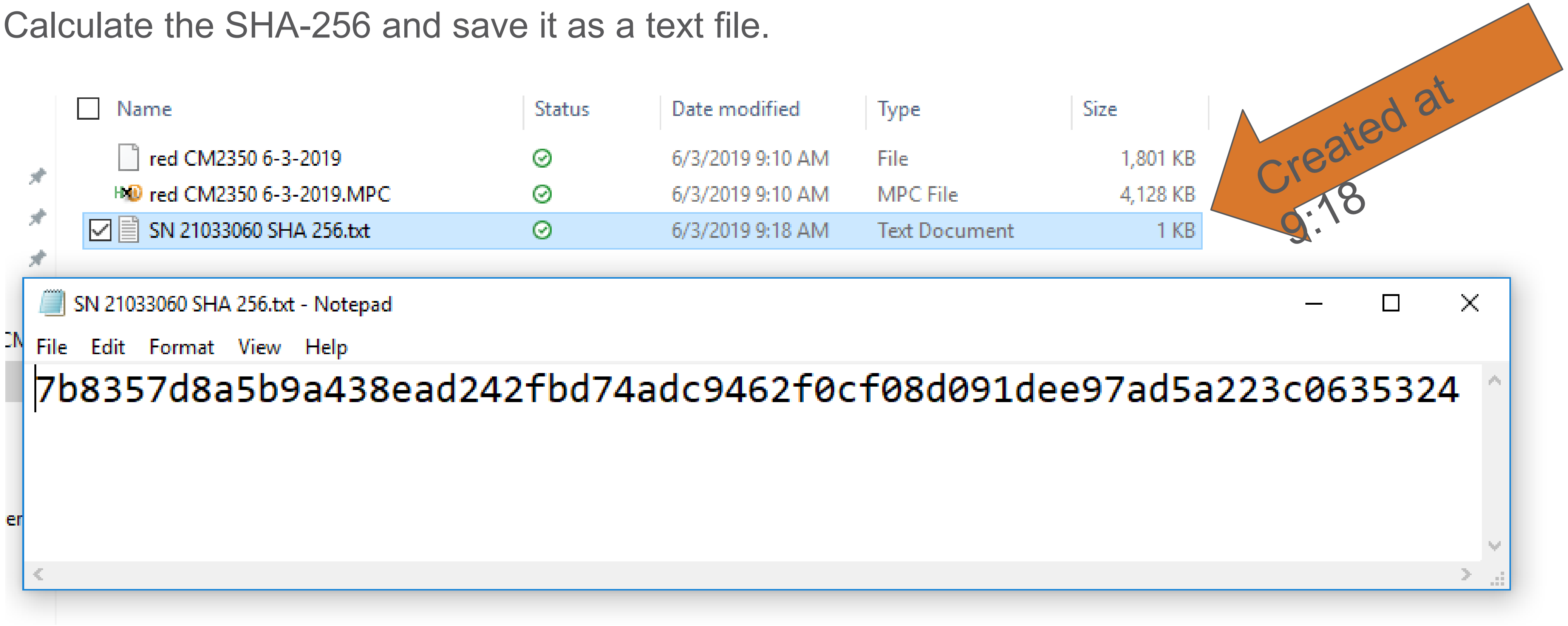
Hash the data after Extraction



Recommended Practice

Immediately after imaging the chip memory:

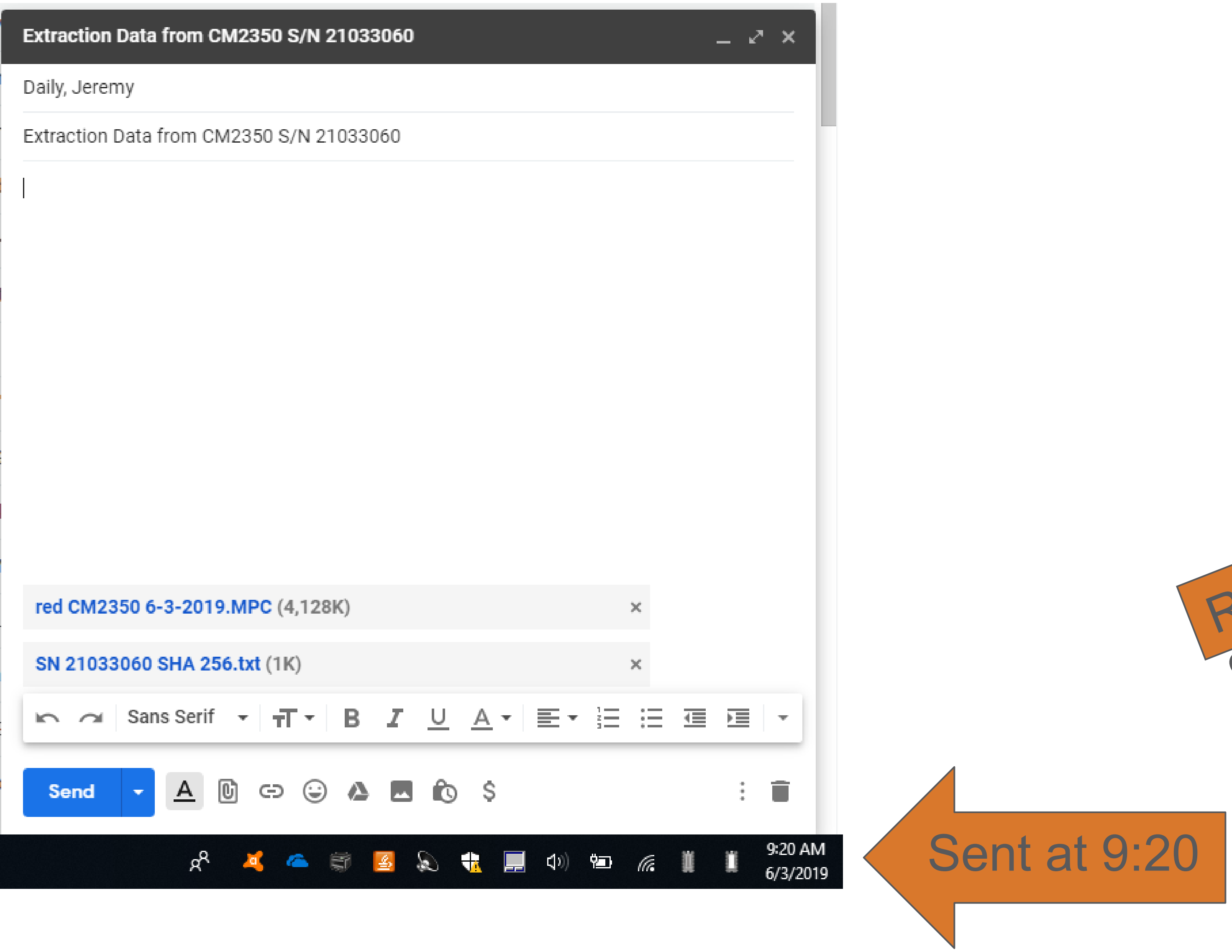
– Calculate the SHA-256 and save it as a text file.

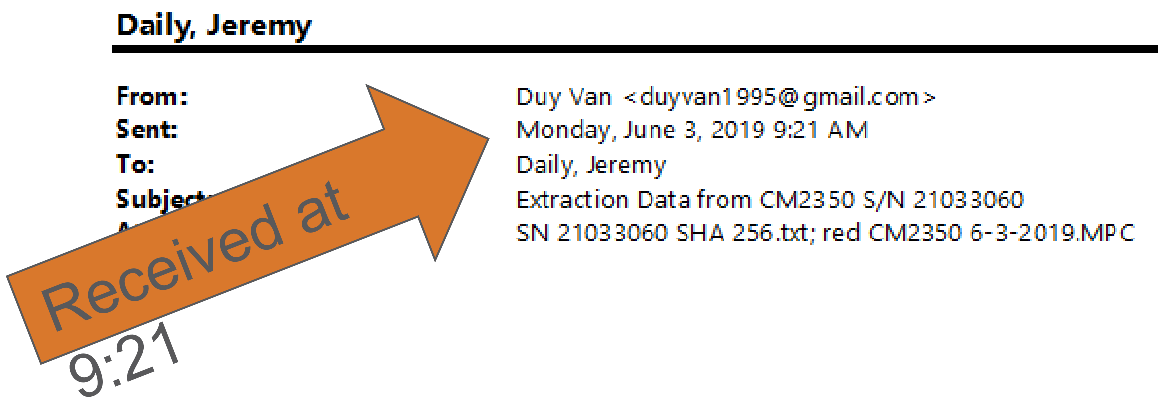


Recommended Practice

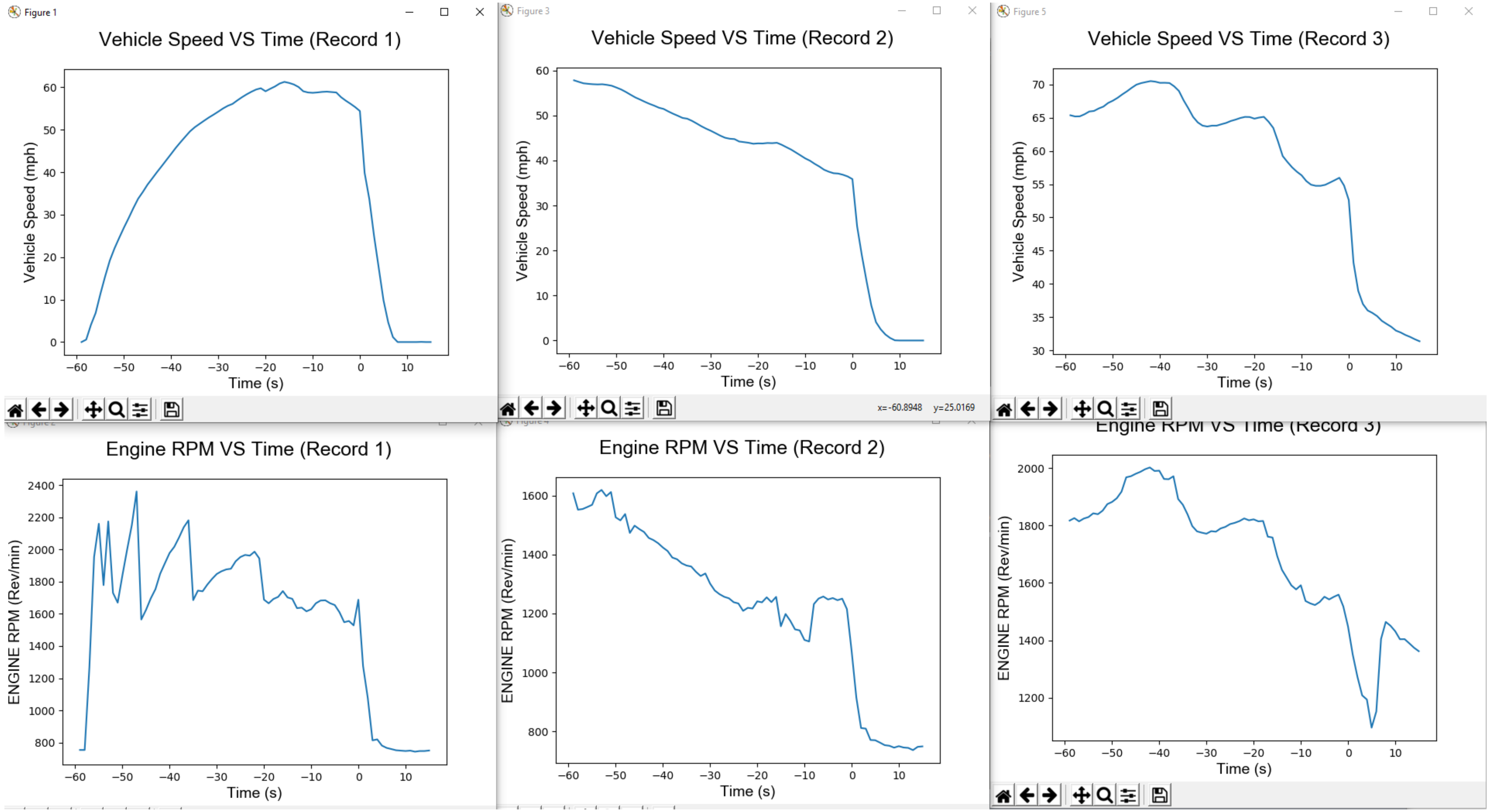
E-Mail the SHA-256 digest value to a trusted party

Establishes a record with a timestamp when the SHA was calculated





After Data Preservation, Analyze the New Data



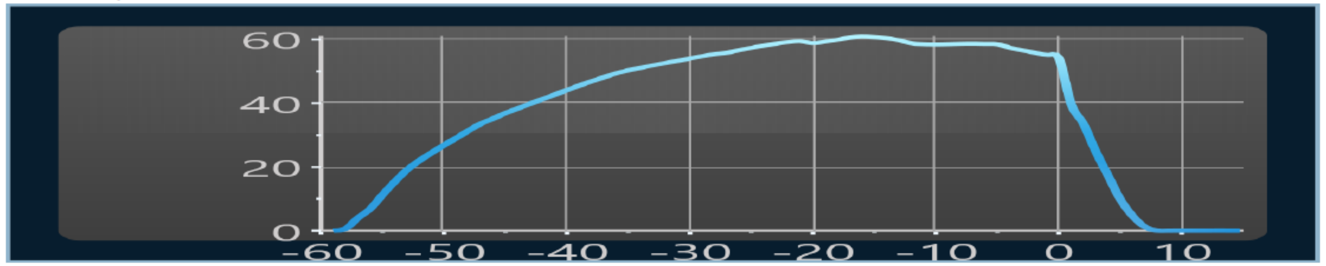
Confirm with PowerSpec

Vehicle Sudden Deceleration Report Record 1

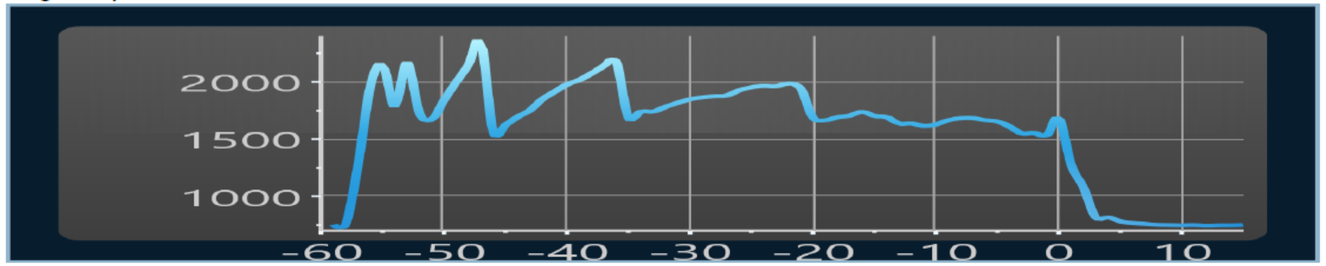
Engine Type	B6.7 2017	Ecm Code	HC80013.14
Engine Serial Number	74109625	Software Phase	22.60.70.3
Unit Number	**********	Extraction Date	06-03-2019 09:46:00
Sudden Decel Threshold Rate:	9.01 mph	ECM Run time	1578:40:56

Occurrence Date: N\A
Air Temperature (°F) at Occurrence: 91
ECM Run Time at Occurrence: 1287:52:34
Occurrence Distance (mi): 37785.1

Vehicle Speed



Engine Speed



Percent

Summary and Conclusions

- ECUs may be broken in a crash and not communicate with vehicle networks.
- Board level access provides opportunity to image chips through JTAG programming ports.
- Chip level access reads memory bearing chips directly
- Resulting binary data from Cummins ECUs has Sudden Deceleration Records to decode.
- A man-in-the-middle can help decode the data.
- Data can be manipulated with a hex editor.
- A Secure Hash Algorithm and tight timelines are critical to ensuring forensic verifiability.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1715409. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Autonomous Systems Forensics

Autonomous systems are on the road now

Overview

- What is the ATMA
- Data Overview
- Data Samples
- Key Challenges
- Recommendations
- Summary

Autonomous Truck Mounted Attenuator

https://royaltruckandequipment.com/truck_bodies/autonomous-tma/

https://royaltruckandequipment.com/truck_bodies/autonomous-tma/

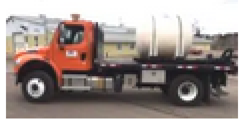
The autonomous TMA system can be used in a nearly unlimited variety of applications. Examples include highway maintenance operations including plowing, sweeping, line striping, and water trucks. The ATMA follower system can be configured to follow almost any vehicle as the leader.

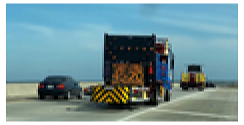
In addition to live deployments, many DOTs have worked with universities to conduct live testing and evaluation of the system. These published reports show the potential for the ATMA to bring us closer toward zero deaths.

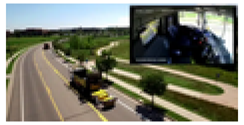

Tennessee DOT

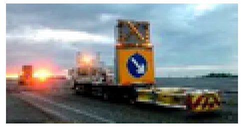

North Dakota DOT


Missouri DOT


Minnesota DOT


Florida DOT


Colorado DOT


Colas U.K. DOT


Caltrans DOT

Leader – Follower System

CDOT Paint Striper as leader

Autonomous Truck Mounted Attenuator as follower

Leader – Follower System

Photo by
John Daily

Automated Truck Mounted Attenuator

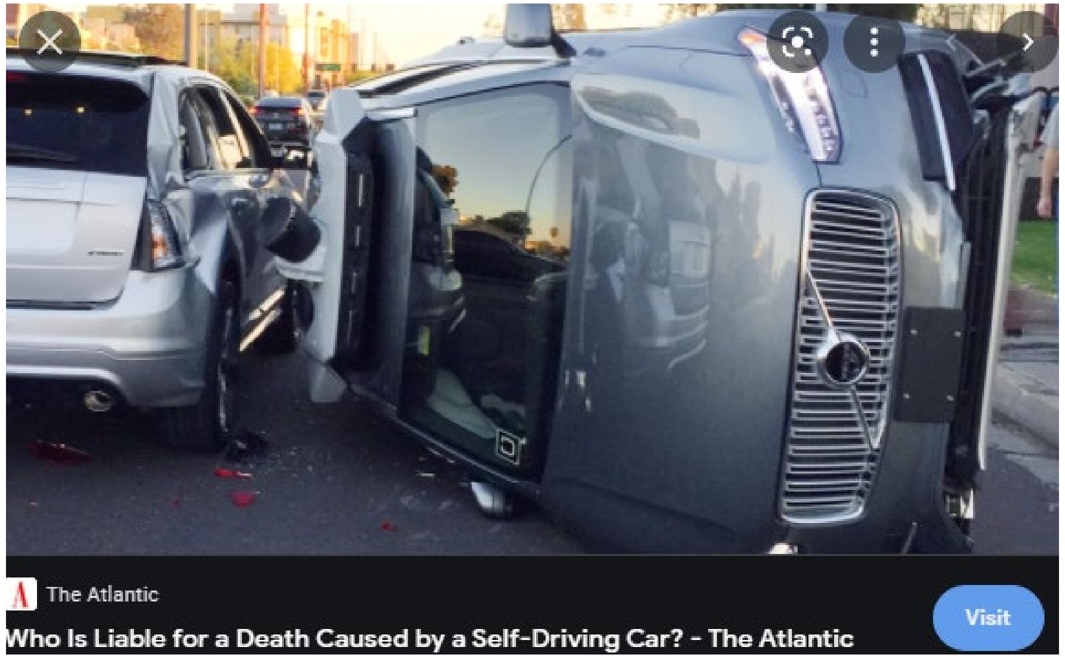
- Leader sends GPS Based Breadcrumbs
- Follower seeks a following distance along the breadcrumb path

Questions:

- What type of forensic log data exists for these systems?
- Is the data reliable? (Available and accurate)

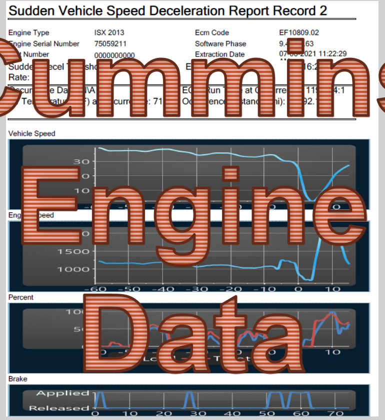
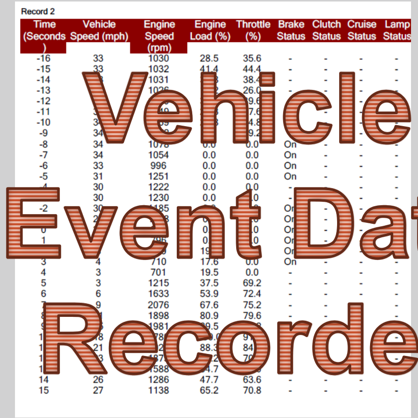
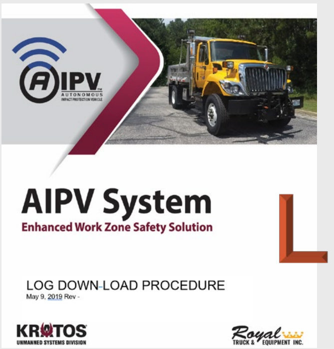
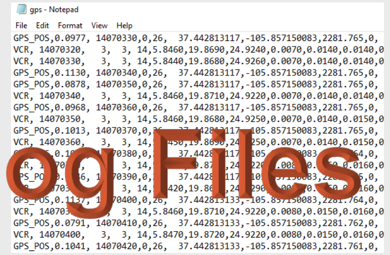
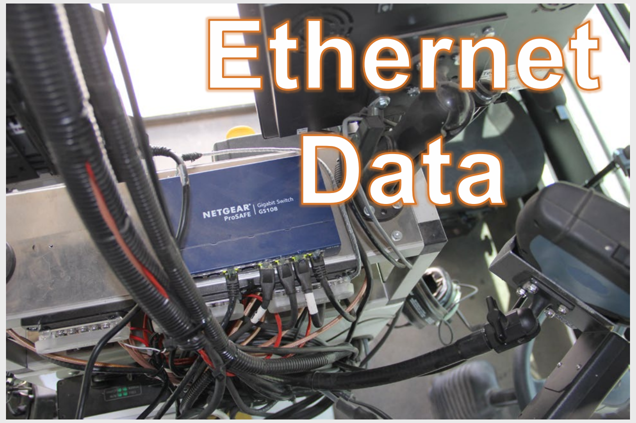
Motivation


FAST FAILS
AUTONOMOUS


The Atlantic
Who Is Liable for a Death Caused by a Self-Driving Car? - The Atlantic
Visit


After
Before
India.com
Google self-driving car fails public test, causes accident after crashing into public transit bus (Watch video) | India.com
Visit

ATMA Data Sources

Source	Leader	Follower
Source		
Native Vehicle *J1939*	 Cummins Engine Data  Vehicle Event Data Recorder	 Brake Data Recorder
Autonomy Solution	  Log Files 	 Ethernet Data

ATMA Onboard Data Overview

- Data Sources:
  - Native Vehicle data – Available via J1939 standard interface. Includes log files for engine ECU, transmission, and brake controller
  - ATMA specific log files – Log files generated from ATMA specific commands and navigation information

Native vehicle Data Sample:

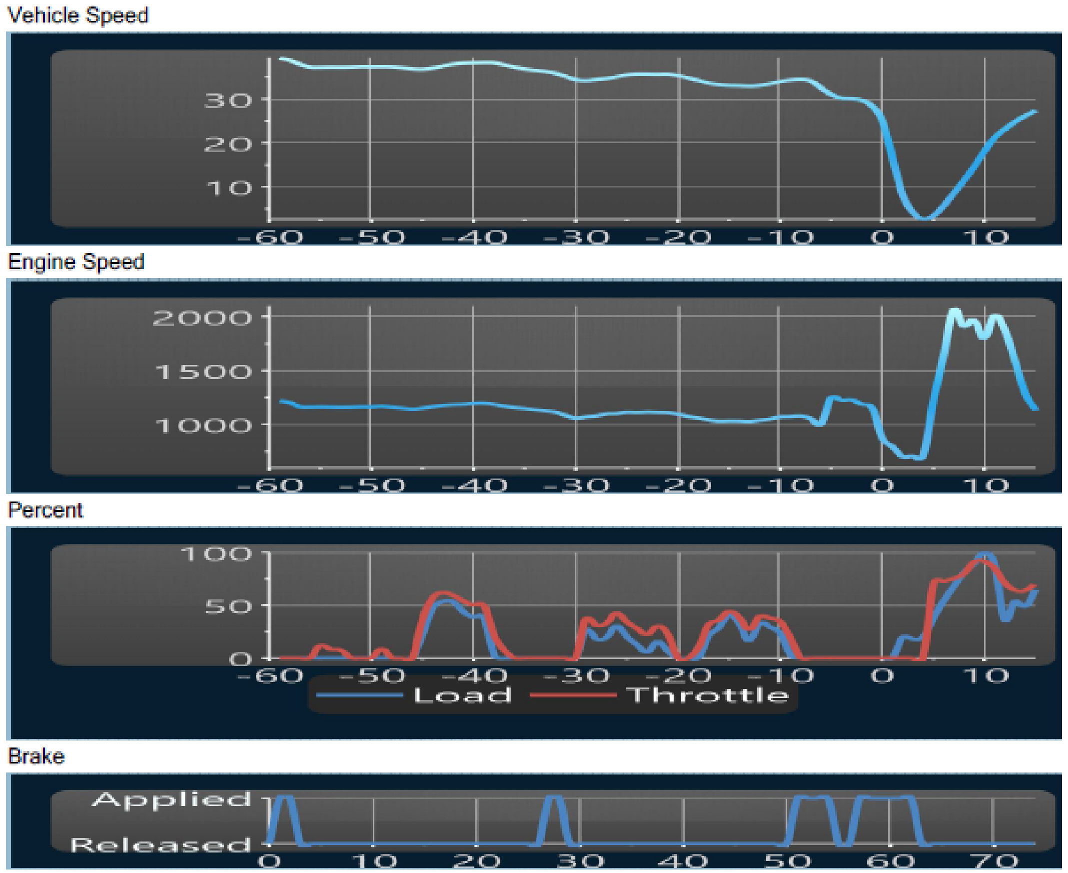
- Data captured through the leader's diagnostic port using Cummins Powerspec software
- Data collection performed on 6 July 2021
- Shows leader vehicle events
  - Brake
  - Accelerator pedal
  - Vehicle Speed
  - Engine RPM
- Common for incident response

Sudden Vehicle Speed Deceleration Report Record 2

Engine Type	ISX 2013	Ecm Code	EF10809.02
Engine Serial Number	75059211	Software Phase	9.40.17.63
Unit Number	0000000000	Extraction Date	07-06-2021 11:22:29 AM
Sudden Decel Threshold Rate:	7.00 mph/s	ECM Run time	1216:20:12

Occurrence Date: N\A — ECM Run Time at Occurrence: 1190:44:1

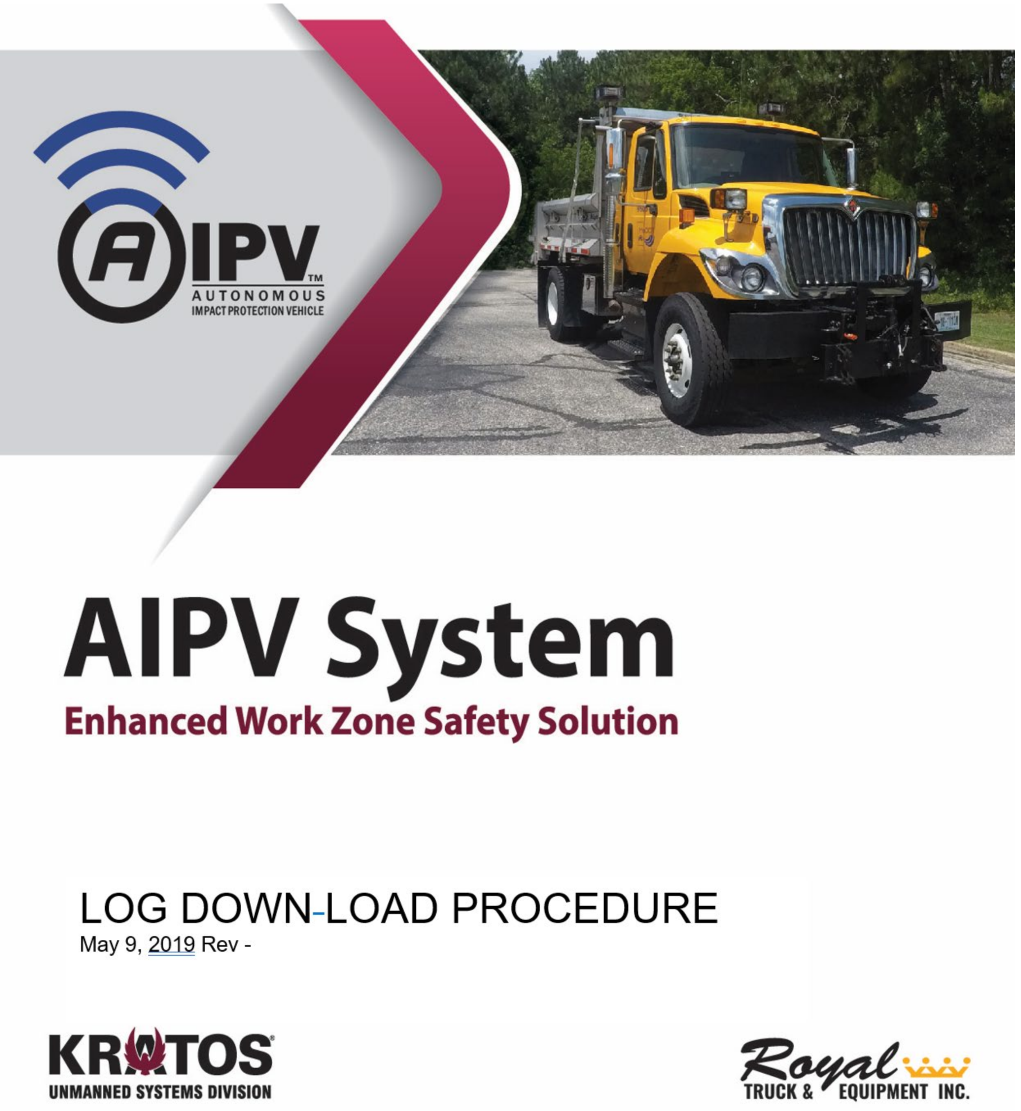
Air Temperature (°F) at Occurrence: 71 — Occurrence Distance (mi): 20092.1



Record 2

Time (Seconds)	Vehicle Speed (mph)	Engine Speed (rpm)	Engine Load (%)	Throttle (%)	Brake Status	Clutch Status	Cruise Status	Lamp Status
-16	33	1030	28.5	35.6	-	-	-	-
-15	33	1032	41.4	44.4	-	-	-	-
-14	33	1031	32.8	38.4	-	-	-	-
-13	33	1026	15.2	26.0	-	-	-	-
-12	33	1039	33.6	39.6	-	-	-	-
-11	33	1049	29.3	37.6	-	-	-	-
-10	34	1069	23.8	34.8	-	-	-	-
-9	34	1072	3.1	19.2	-	-	-	-
-8	34	1078	0.0	0.0	On	-	-	-
-7	34	1054	0.0	0.0	On	-	-	-
-6	33	996	0.0	0.0	On	-	-	-
-5	31	1251	0.0	0.0	On	-	-	-
-4	30	1222	0.0	0.0	-	-	-	-
-3	30	1230	0.0	0.0	-	-	-	-
-2	30	1185	0.0	0.0	On	-	-	-
-1	28	1158	0.0	0.0	On	-	-	-
0	25	872	0.0	0.0	On	-	-	-
1	17	796	0.0	0.0	On	-	-	-
2	8	699	19.9	0.0	On	-	-	-
3	4	710	17.6	0.0	On	-	-	-
4	3	701	19.5	0.0	-	-	-	-
5	3	1215	37.5	69.2	-	-	-	-
6	6	1633	53.9	72.4	-	-	-	-
7	9	2076	67.6	75.2	-	-	-	-
8	11	1898	80.9	79.6	-	-	-	-
9	15	1981	89.5	91.2	-	-	-	-
10	18	1780	100.0	91.6	-	-	-	-
11	21	2023	88.3	84.8	-	-	-	-
12	23	1875	35.2	70.8	-	-	-	-
13	25	1588	54.7	64.0	-	-	-	-
14	26	1286	47.7	63.6	-	-	-	-
15	27	1138	65.2	70.8	-	-	-	-

ATMA Log Data Sample:

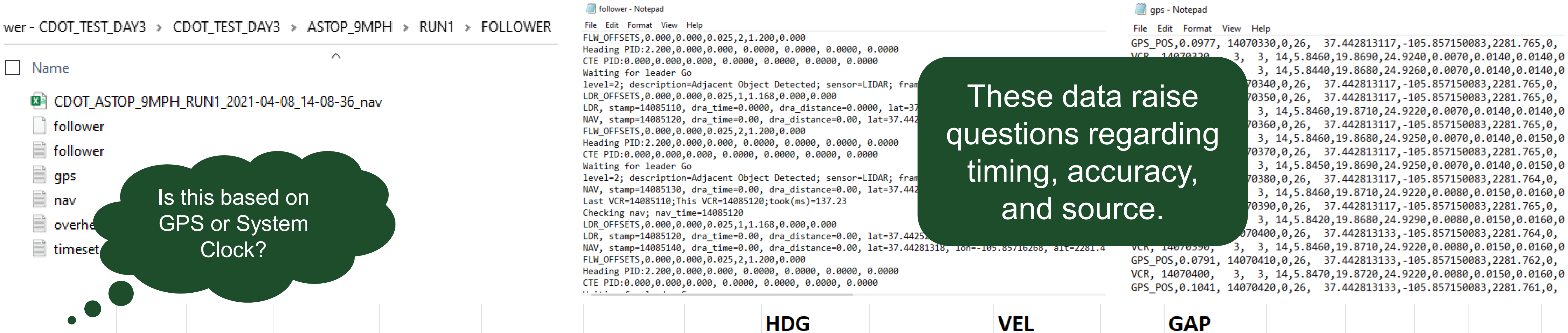


Data Headers available in the .CSV log files downloaded from the ATMA Limon System in January 2022 testing

Table 1. Data Headers from ATMA CSV Log File

Vehicle	Crumb	Stamp	Lat	Lon	Alt	Heading	Heading (Desired)	Velocity
Velocity (Desired)	Gap	Gap (Desired)	#of Satellites	Valid	CTE	Accel	Steer	State

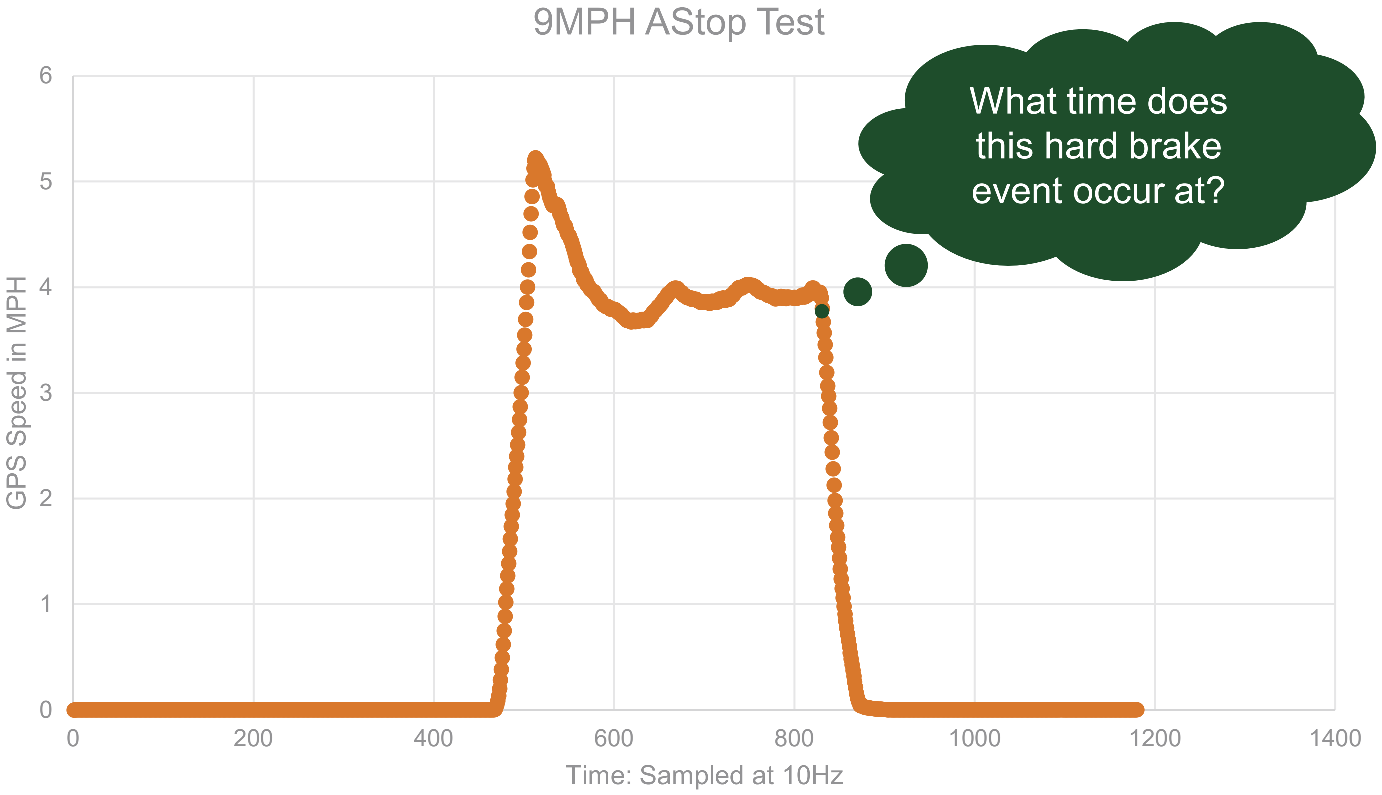
ATMA Log Data Sample:

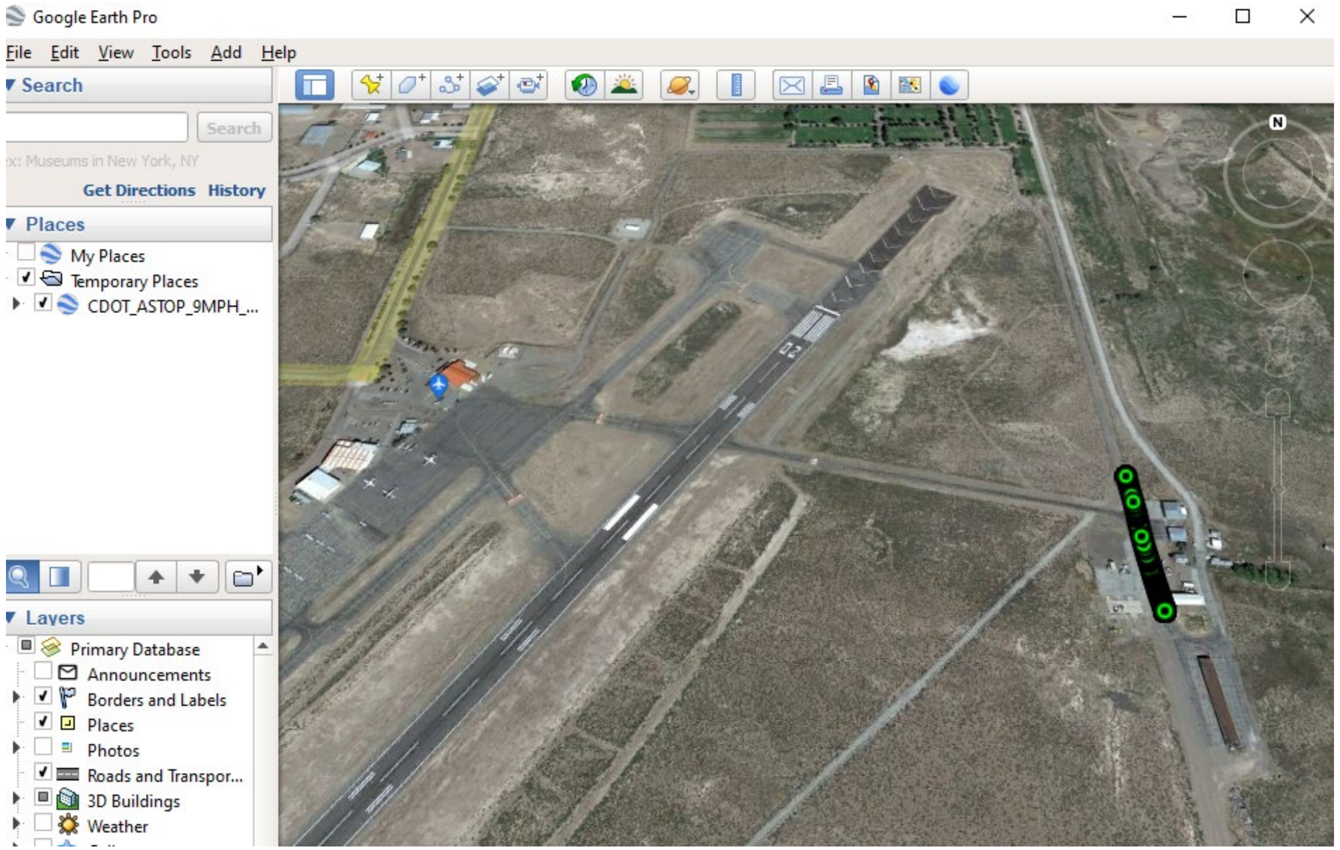


TIME	VEH	CRUMB	STAMP	LAT	LON	ALT	HDG	HDG (Desired)	VELOCITY	VEL (Desired)	GAP	GAP (Desired)	#SATS	VALID	CTE	ACCEL	STEER	STATE
38:17.0	FLW	457	20381700	39.27197	-103.7308	1632.171	285.99	286.153	17.69	17.91	48.88	30.5	27	1	0.41	53.59	-3.95	RUN
38:17.1	LDR	476	20381690	39.27213	-103.7315	1630.346	286.4	17.79	286.3	400	77							
38:17.1	FLW	457	20381710	39.27198	-103.7308	1632.173	285.86	286.143	17.67	17.82	48.89	30.5	27	1	0.42	52.99	-4.27	RUN
38:17.2	LDR	477	20381700	39.27213	-103.7315	1630.347	286.41	17.77	286.8	401	77							
38:17.2	FLW	457	20381720	39.27198	-103.7309	1632.18	285.84	286.146	17.85	17.78	48.9	30.5	27	1	0.43	50.74	-4.33	RUN
38:17.2	LDR	478	20381710	39.27213	-103.7315	1630.34	286.39	17.94	285.6	402	77							
38:17.3	FLW	457	20381730	39.27198	-103.7309	1632.175	285.87	286.15	18.08	17.85	48.9	30.5	27	1	0.43	49.3	-4.26	RUN
38:17.3	LDR	479	20381720	39.27214	-103.7315	1630.339	286.33	17.9	286.3	403	77							
38:17.4	FLW	457	20381740	39.27198	-103.7309	1632.184	285.86	286.148	18.16	17.92	48.9	30.5	27	1	0.42	49.25	-4.18	RUN
38:17.5	LDR	480	20381730	39.27214	-103.7316	1630.341	286.31	17.82	287.1	404	77							
38:17.5	FLW	457	20381750	39.27198	-103.7309	1632.189	285.8	286.153	18.19	17.86	48.89	30.5	27	1	0.43	48.29	-4.33	RUN

ATMA Data Analysis (Initial):

- Cursory Analysis of the logs generates the following:
  - Scatter plot analysis of the GPS Speed vs Timestamp in Excel
  - Google Earth can import .csv or .kml file data and plot a breadcrumb trace of Lat vs Long





- NOTE: Kratos has provided a Python script for automating some of this analysis but at this time further investigation and troubleshooting is being performed on its use.
  - Of specific interest is a method to combine leader and follower logs to generate a combined location breadcrumb path trace

Key Challenge: Data Attribution

- Incident Response to a crash for traditional heavy vehicles has well defined procedures
  - Law Enforcement and investigators are familiar with procedures to download vehicle log files via the standard J1939 interfaces
- Procedures for incident response to autonomous heavy vehicles are in their infancy
  - ATMA specific log data download procedures are new and unfamiliar for law enforcement and investigators
    - Area for Improvement 3.2 from the Tabletop: Streamline and provide data download and handling procedures

- **Issue:** Lack of Common Data Fields Between Powertrain Data Logs and ATMA Data Logs
  - Powertrain data does NOT include a Timestamp – uses Engine Hours for time reference
  - Attribution can also be made using odometer readings
  - ATMA specific logs do NOT include any vehicle network data
    - Wheel Speed should correlate to GPS speed
    - At low speeds wheel speed may be more accurate
    - Currently, these signals are not synchronized

Table 1. Data Headers from ATMA CSV Log File

Vehicle	Crumb	Stamp	Lat	Lon	Alt	Heading	Heading (Desired)	Velocity
Velocity (Desired)	Gap	Gap (Desired)	#of Satellites	Valid	CTE	Accel	Steer	State

Powerspec Data Header

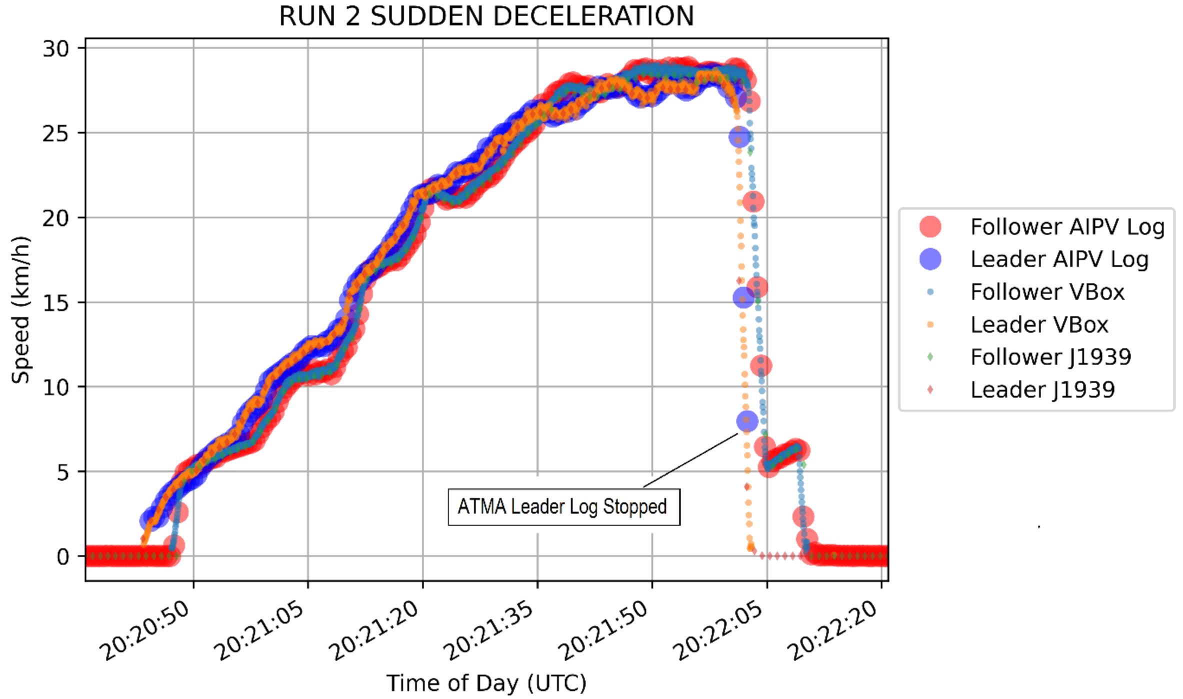
Time (Seconds)	Vehicle Speed (mph)	Engine Speed (rpm)	Engine Load (%)	Throttle (%)	Brake Status	Clutch Status	Cruise Status	Lamp Status

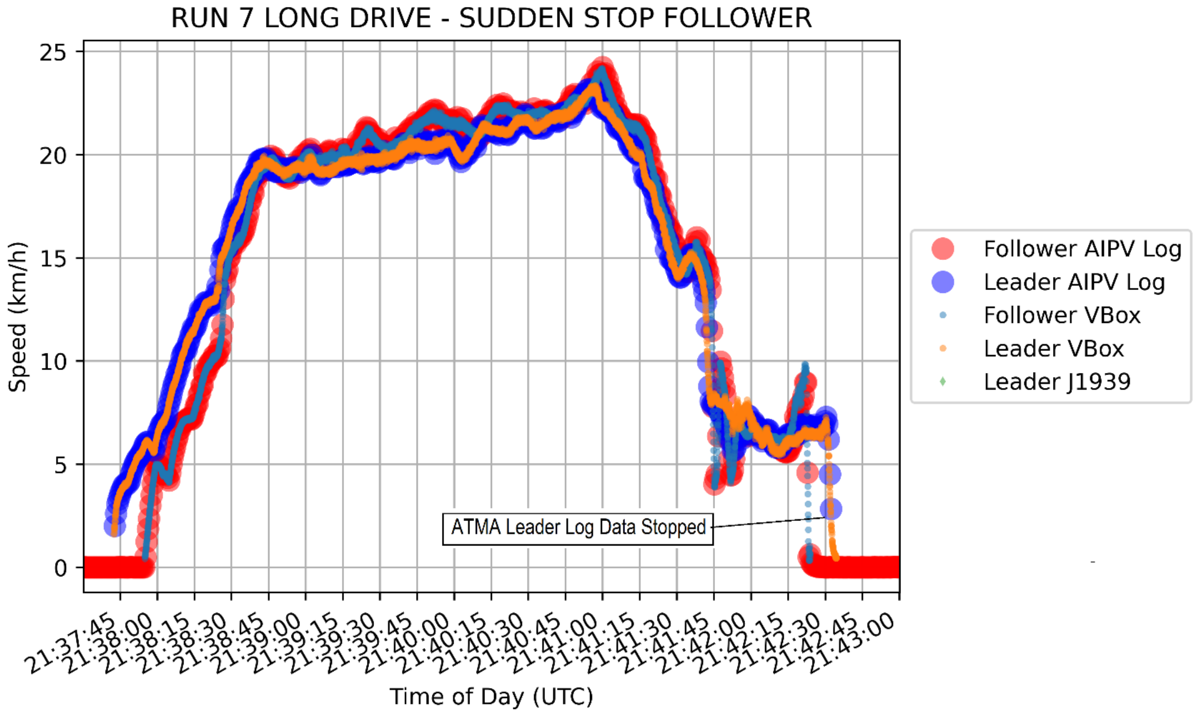
NOTE: This time is counting down to Captured Event, not a GPS Time

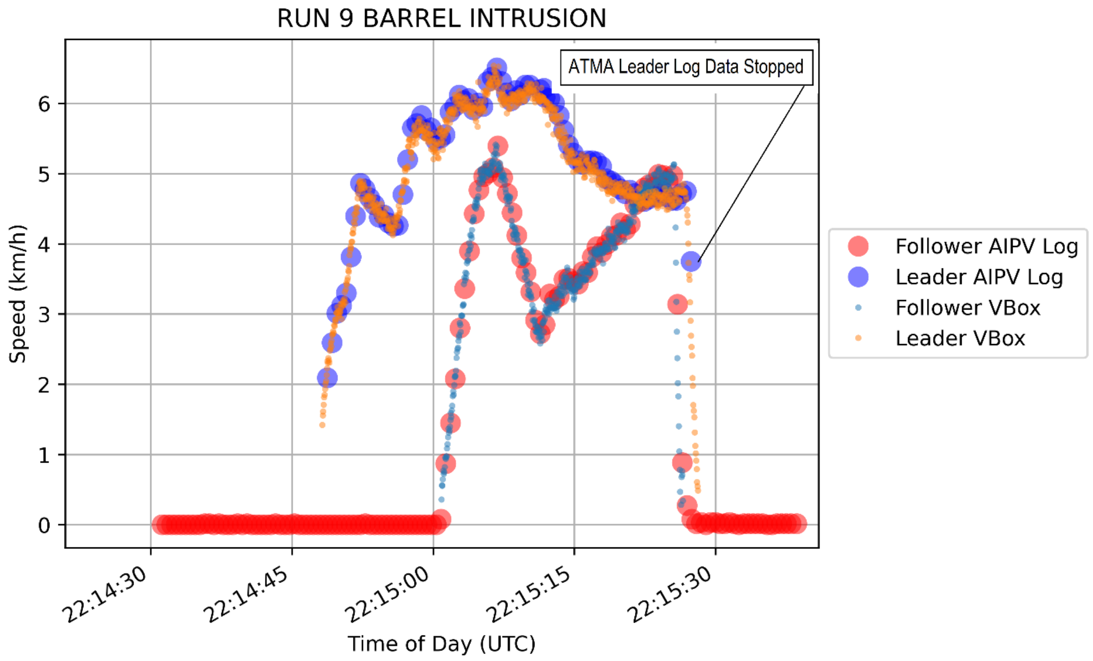
CSU ATMA Data Collection Testing 13 Jan 2022

- Test 1 – Sudden Deceleration – Leader Initiated Heavy Braking
- Test 2 – Sudden Deceleration – ESTOP Leader
- Test 3 – Sudden Deceleration – ESTOP Follower
- Test 4 – Uncommanded Sudden Stop from Follower
- Test 5 – Sudden Deceleration – Barrel Intrusion

Test Result Data Samples







Data and Security Recommendations:

Recommendation 1: Improve data attribution -- Add vehicle specific data sources to ATMA specific log to facilitate direct comparison to vehicle network data logs.

Recommendation 2: Produce clear and concise ATMA data retrieval procedures – In accordance with the Tabletop After Action Report, Area for Improvement 3.2 highlights the need for clear data download and handling procedures, written for DOT personnel, law enforcement, and other investigators.

Recommendation 3: Deliver a user friendly ATMA log analysis tool – current ATMA log file analysis is cumbersome and does not have a GUI or user-friendly interface to generate plots, charts, and tables of relevant data for incident response.

Recommendation 4: Clarify roles for support to interpret log files – Are there any contractual obligations for ATMA vendor to support the gathering or interpreting log files after an incident?

Recommendation 5: Modify the ATMA logging system to continue to log after an ESTOP– resting showed the ATMA logs terminate upon the loss of ignition power in an ESTOP. The data during the commanded Estop hard braking event is critical information in the case of an accident

Recommendation 1: Improve Data Attribution

- Currently ATMA logs do not include vehicle specific CAN Bus data from J1939.
  - This is challenging for correlating powertrain data to ATMA specific information

- SAE Standard J3197-Automated Driving System Data Logger provides specific recommended minimum data logger data elements for automated driver systems
- Proposed Solution: Add powertrain data to ATMA specific log
  - Can be accomplished via a Data Diode from the J1939 CAN Network to broadcast UDP packets. A data diode would preserve security through physical restriction of a one- way data flow

Sudden Vehicle Speed Deceleration Report Record 1

Engine Type	ISX 2013	Ecm Code	EF10809.02
Engine Serial Number	75059211	Software Phase	9.40.17.63
Unit Number	0000000000	Extraction Date	01-13-2022 01:07:10
Sudden Decel Threshold Rate:	7.00 mph/s	ECM Run time	1363:46:32

Occurrence Date: N\A
Air Temperature (°F) at Occurrence: 78
ECM Run Time at Occurrence: 1295:32:7
Occurrence Distance (mi): 22176.7

E	VEH	CRUMB	STAMP	LAT	LON	ALT	HDG	HDG (Desired)	VELOCITY	(Desired)	GAP	GAP (Desired)	#SATS	VALID	CTE	ACCEL	STEER	STATE
3:17.0	FLW	457	20381700	39.27197	-103.7308	1632.171	285.99	286.153	17.69	17.91	48.88	30.5	27	1	0.41	53.59	-3.95	RUN
3:17.1	LDR	476	20381690	39.27213	-103.7315	1630.346	286.4	17.79	286.3	400	77							
3:17.1	FLW	457	20381710	39.27198	-103.7308	1632.173	285.86	286.143	17.67	17.82	48.89	30.5	27	1	0.42	52.99	-4.27	RUN
3:17.2	LDR	477	20381700	39.27213	-103.7315	1630.347	286.41	17.77	286.8	401	77							
3:17.2	FLW	457	20381720	39.27198	-103.7309	1632.18	285.84	286.146	17.85	17.78	48.9	30.5	27	1	0.43	50.74	-4.33	RUN
3:17.2	LDR	478	20381710	39.27213	-103.7315	1630.34	286.39	17.94	285.6	402	77							
3:17.3	FLW	457	20381730	39.27198	-103.7309	1632.175	285.87	286.15	18.08	17.85	48.9	30.5	27	1	0.43	49.3	-4.26	RUN
3:17.3	LDR	479	20381720	39.27214	-103.7315	1630.339	286.33	17.9	286.3	403	77							
3:17.4	FLW	457	20381740	39.27198	-103.7309	1632.184	285.86	286.148	18.16	17.92	48.9	30.5	27	1	0.42	49.25	-4.18	RUN
3:17.5	LDR	480	20381730	39.27214	-103.7316	1630.341	286.31	17.82	287.1	404	77							
3:17.5	FLW	457	20381750	39.27198	-103.7309	1632.189	285.8	286.153	18.19	17.86	48.89	30.5	27	1	0.43	48.29	-4.33	RUN
3:17.5	LDR	481	20381740	39.27214	-103.7316	1630.342	286.31	17.86	286.5	406	76							
3:17.6	FLW	457	20381760	39.27199	-103.7309	1632.191	285.84	286.154	18.07	17.84	48.87	30.5	27	1	0.43	49.35	-4.33	RUN
3:17.7	LDR	482	20381750	39.27214	-103.7316	1630.344	286.31	17.84	285.7	407	76							
3:17.7	FLW	457	20381770	39.27199	-103.7309	1632.194	285.85	286.156	18.15	17.85	48.86	30.5	27	1	0.44	48.68	-4.39	RUN
3:17.7	LDR	483	20381760	39.27214	-103.7316	1630.344	286.34	17.82	286.2	408	76							
3:17.8	FLW	457	20381780	39.27199	-103.7309	1632.202	285.88	286.155	18.22	17.83	48.85	30.5	27	1	0.45	47.77	-4.35	RUN
3:17.8	LDR	484	20381770	39.27215	-103.7316	1630.341	286.34	17.94	286.6	409	76							
3:17.9	FLW	457	20381790	39.27199	-103.7309	1632.206	285.84	286.154	18.21	17.88	48.83	30.5	27	1	0.44	48.43	-4.41	RUN
3:17.9	LDR	485	20381780	39.27215	-103.7316	1630.343	286.35	18.02	286.7	410	76							
3:18.0	FLW	457	20381800	39.27199	-103.7309	1632.204	285.85	286.155	18.27	17.98	48.82	30.5	27	1	0.44	48.81	-4.33	RUN
3:18.1	LDR	486	20381790	39.27215	-103.7316	1630.343	286.4	18.01	285.6	411	76							
3:18.1	FLW	457	20381810	39.272	-103.7309	1632.205	285.97	286.149	18.05	18.01	48.81	30.5	26	1	0.44	51.33	-4.14	RUN
3:18.2	LDR	487	20381800	39.27215	-103.7316	1630.342	286.39	18.06	286	412	76							
3:18.2	FLW	457	20381820	39.272	-103.7309	1632.219	285.94	286.162	18.09	18.03	48.81	30.5	26	1	0.44	51.11	-4.22	RUN
3:18.3	LDR	488	20381810	39.27215	-103.7316	1630.339	286.4	18.09	286	413	76							
3:18.3	FLW	457	20381830	39.272	-103.731	1632.225	285.95	286.155	18.02	18.07	48.81	30.5	27	1	0.43	52.26	-4.11	RUN
3:18.4	LDR	489	20381820	39.27216	-103.7316	1630.34	286.43	18.18	286.2	414	76							
3:18.4	FLW	457	20381840	39.272	-103.731	1632.223	285.99	286.146	18.26	18.13	48.81	30.5	27	1	0.43	50.47	-3.94	RUN
3:18.5	LDR	490	20381830	39.27216	-103.7316	1630.34	286.42	18.24	285.3	416	75							
3:18.5	FLW	457	20381850	39.272	-103.731	1632.216	285.97	286.149	18.21	18.21	48.79	30.5	27	1	0.44	51.69	-4.13	RUN
3:18.6	LDR	491	20381840	39.27216	-103.7316	1630.337	286.42	18.26	285.8	417	75							
3:18.6	FLW	457	20381860	39.27201	-103.731	1632.223	285.94	286.15	18.09	18.25	48.8	30.5	27	1	0.44	53.35	-4.18	RUN
3:18.6	LDR	492	20381850	39.27216	-103.7317	1630.338	286.45	18.24	286.3	418	75							
3:18.7	FLW	457	20381870	39.27201	-103.731	1632.23	285.99	286.146	18.09	18.25	48.8	30.5	27	1	0.43	53.38	-4.04	RUN

SAE INTERNATIONAL

SURFACE VEHICLE RECOMMENDED PRACTICE

J3197™

Issued 2020-04
Revised 2021-07

Superseding J3197 APR202

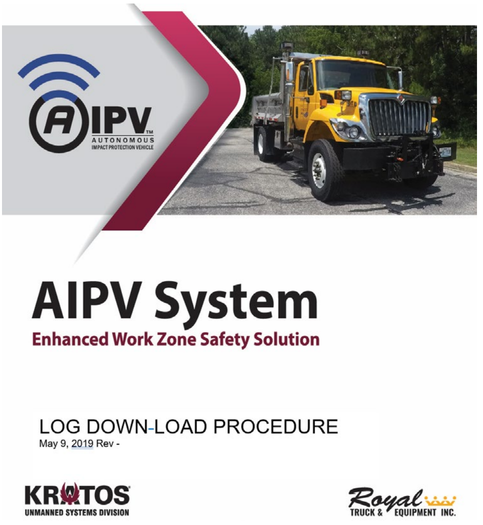
Automated Driving System Data Logger

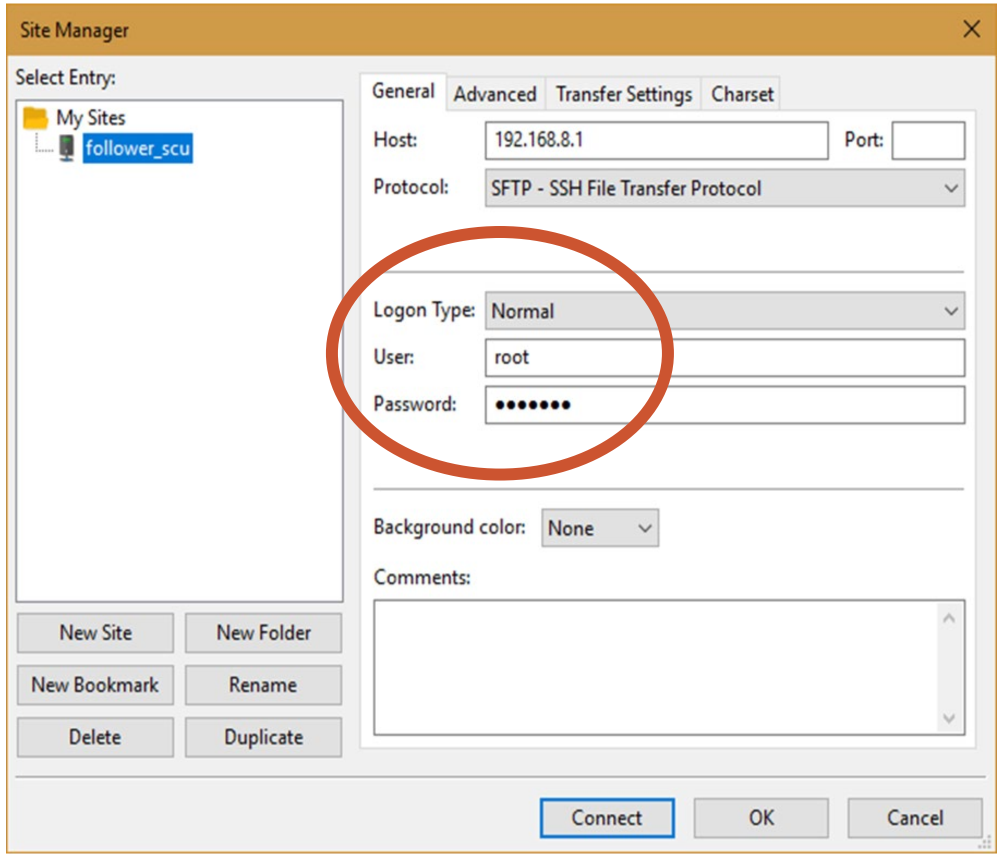
Recommendation 2: Produce Clear and Concise ATMA Data Retrieval Procedures

- Current ATMA log file download procedures are complex and not readily available in the vehicles.
  - Procedure is written for engineers with familiarity in Linux systems
  - Few users are trained on how to accomplish ATMA log file download
  - Current download process uses “root” permission which is a known security vulnerability (See Security Engineering by Ross Anderson).
  - From the Tabletop AAR Area for Improvement 3.2

  “Participating agencies acknowledged the lack of protocols stating what to do with the data that is extracted from the vehicle and how to manage it for further investigation or storage.”

- Procedures should be explicit and clear for download and handling of the data to ensure the integrity of the data if an incident occurs
  - This process will likely mimic existing vehicle data downloads but needs to be explicitly defined for ATMA


AIPV System
Enhanced Work Zone Safety Solution
LOG DOWN-LOAD PROCEDURE
May 9, 2019 Rev -
KRATOS
UNMANNED SYSTEMS DIVISION
Royal TRUCK & EQUIPMENT INC.

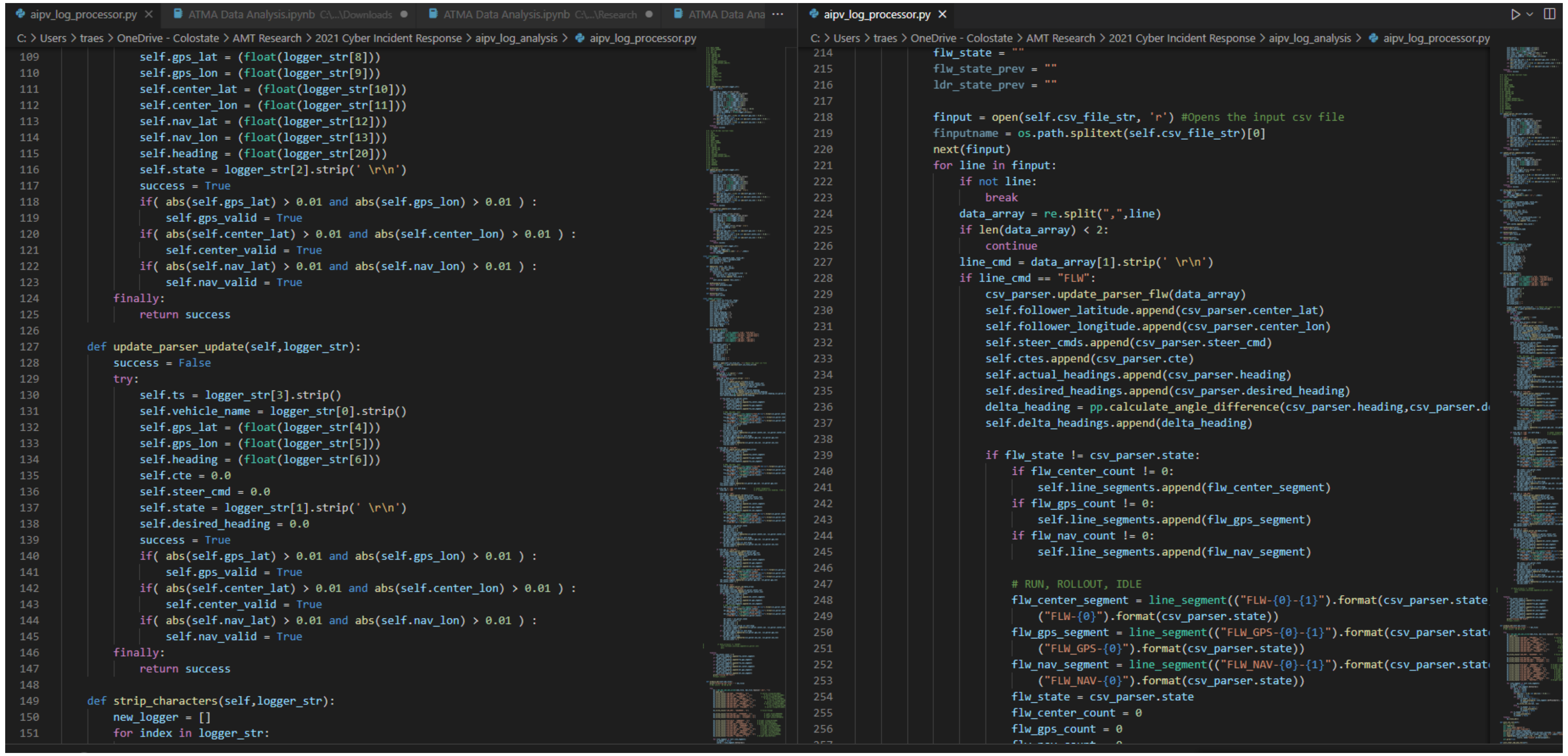


Recommendation 3: Deliver a user friendly ATMA log analysis tool

- Current ATMA log analysis is either done with inefficient Excel analysis or a Python script.
  - Many users may not be familiar with python
    - The current Python script requires download of multiple Python add in packages and does not use a GUI

- A GUI based tool could be developed that would create a user-friendly interface to a script to ingest the data logs and produce tables and graphics of interest for an incident

- Log files should be digitally signed to prevent undetected manipulation

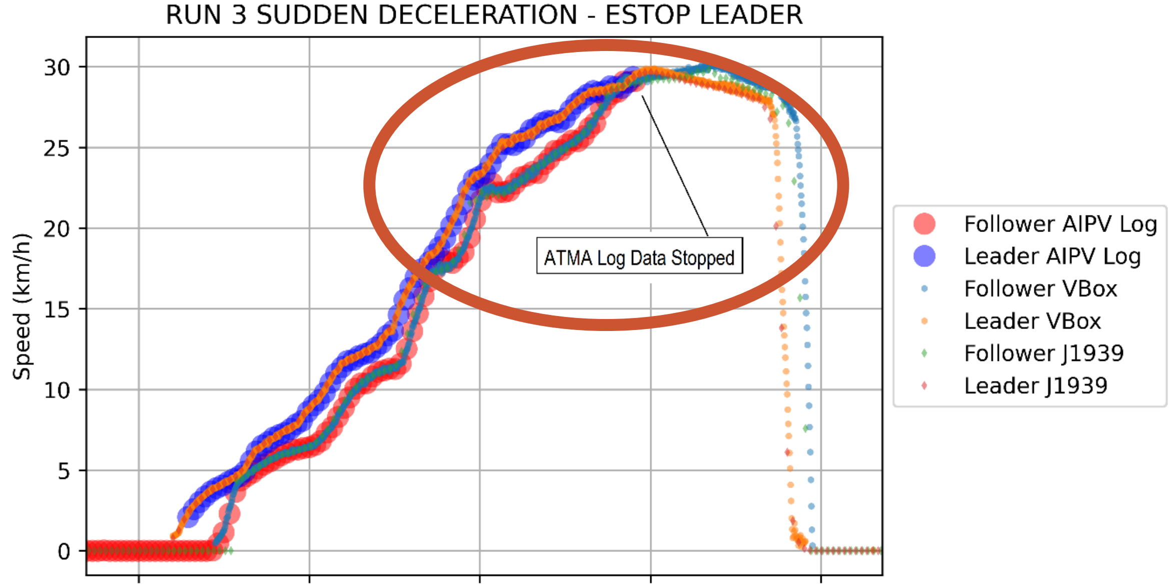


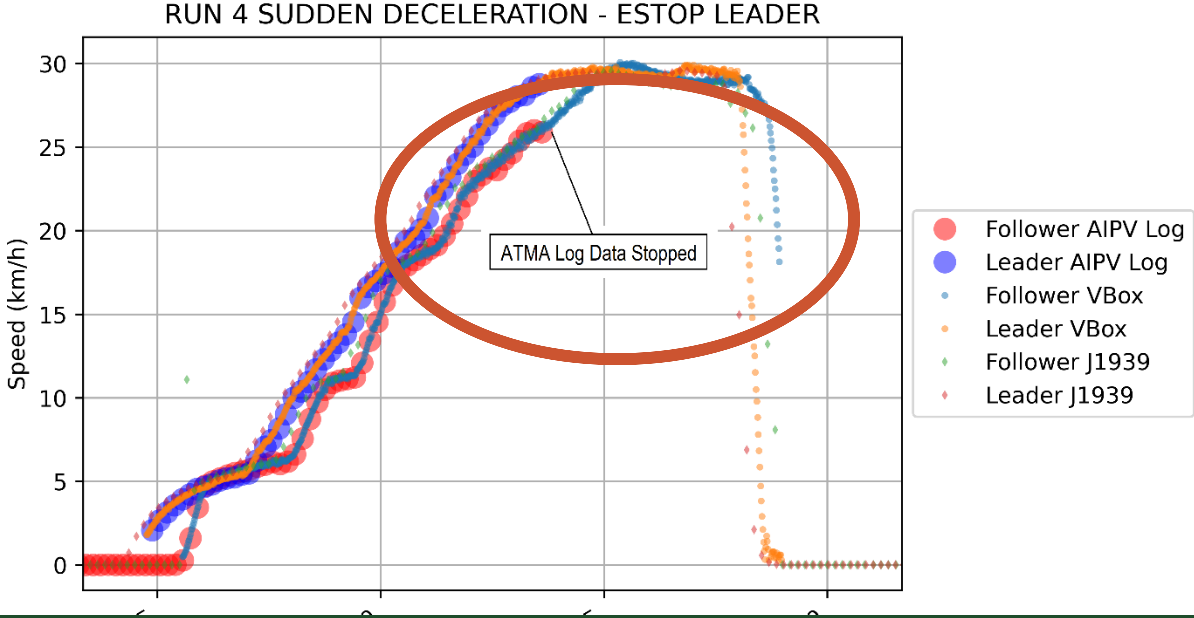
Recommendation 4: Clarify roles for support to interpret log files

- DOT acquisition documents should have specific requirements language of roles and responsibilities for
  - Log data collection
  - Secure log data storage
  - File download procedures
  - Data interpretation and analysis
  - Training for incident response activities
- Understand how to perform these incident response functions without the ATMA vendor (i.e. practice)

Recommendation 5: Modify the ATMA logging system to continue to log after an ESTOP

- CSU tests Executed multiped ESTOP events- during each one the Kratos recording system abruptly terminated after the ESTOP was commanded
  - ESTOP cuts ignition power to the follower vehicle
  - Kratos designed logger to require ignition power to record data
- Data between the commanded ESTOP and the vehicle actually coming to a stop is critical data in the event of an accident





Questions?

Colorado State University