

# Commercial Transportation: Truck Hacking



# Agenda

- ▶ 50 mins
- ▶ What is CT? Why does it matter?
- ▶ About
  - ▶ Trucks
  - ▶ Trailers
  - ▶ Maintenance
  - ▶ Distribution centers
  - ▶ Intermodal
- ▶ Vehicle Networks
  - ▶ J1939
  - ▶ J1708/J1587
  - ▶ J2497
- ▶ ‘Hacking’
  - ▶ CAN Attack Methods
  - ▶ What can you do?
  - ▶ What needs more?
  - ▶ Other ‘Hacking’
- ▶ Get Involved
- ▶ Tools
  - ▶ Examples
  - ▶ More
- ▶ Closing
  - ▶ Review
  - ▶ Call for Collaboration

# About Me



[linkedin.com/in/Ben0L0Gardiner](https://www.linkedin.com/in/Ben0L0Gardiner)



[github.com/BenGardiner](https://github.com/BenGardiner)

## ▶ **Ben Gardiner**

- ▶ Senior Cybersecurity Research Engineer contractor
- ▶ Experience: Embedded systems dev, RE
- ▶ CyberTruck™ Challenge Instructor
- ▶ DC HHV & CHV volunteer
- ▶ SAE volunteer

# About Commercial Transportation

# What is Commercial Transportation (CT)?

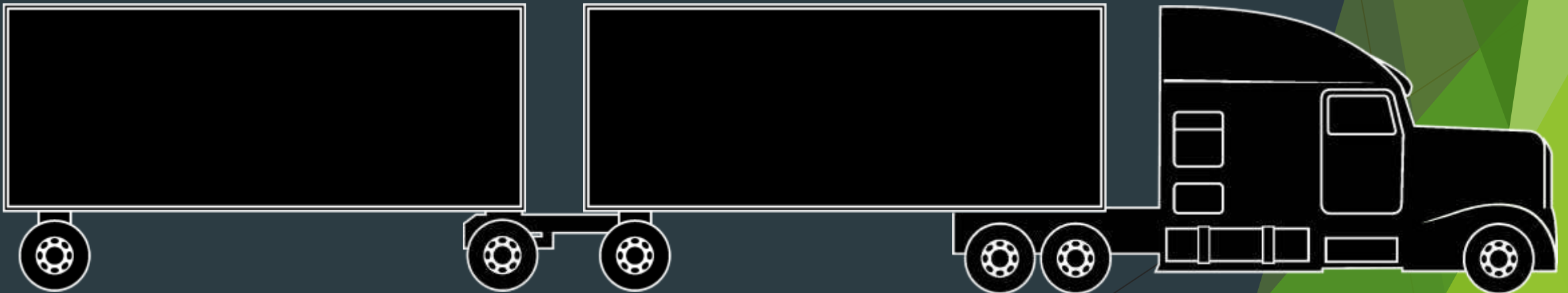
- ▶ All transportation of goods or people for business purposes (a large topic)



CC BY-SA kees torn

# Why does CT Security Matter (Trucking Specifically)?

- ▶ “If you bought it, it came on a truck”
- ▶ Truck problems are big problems for society. c.f. “[A week without Truck Transport](#)” at [iru.org](#)
- ▶ Safety issues with Trucks are all our Issues



# Why does CT Security Matter (All Modes)?

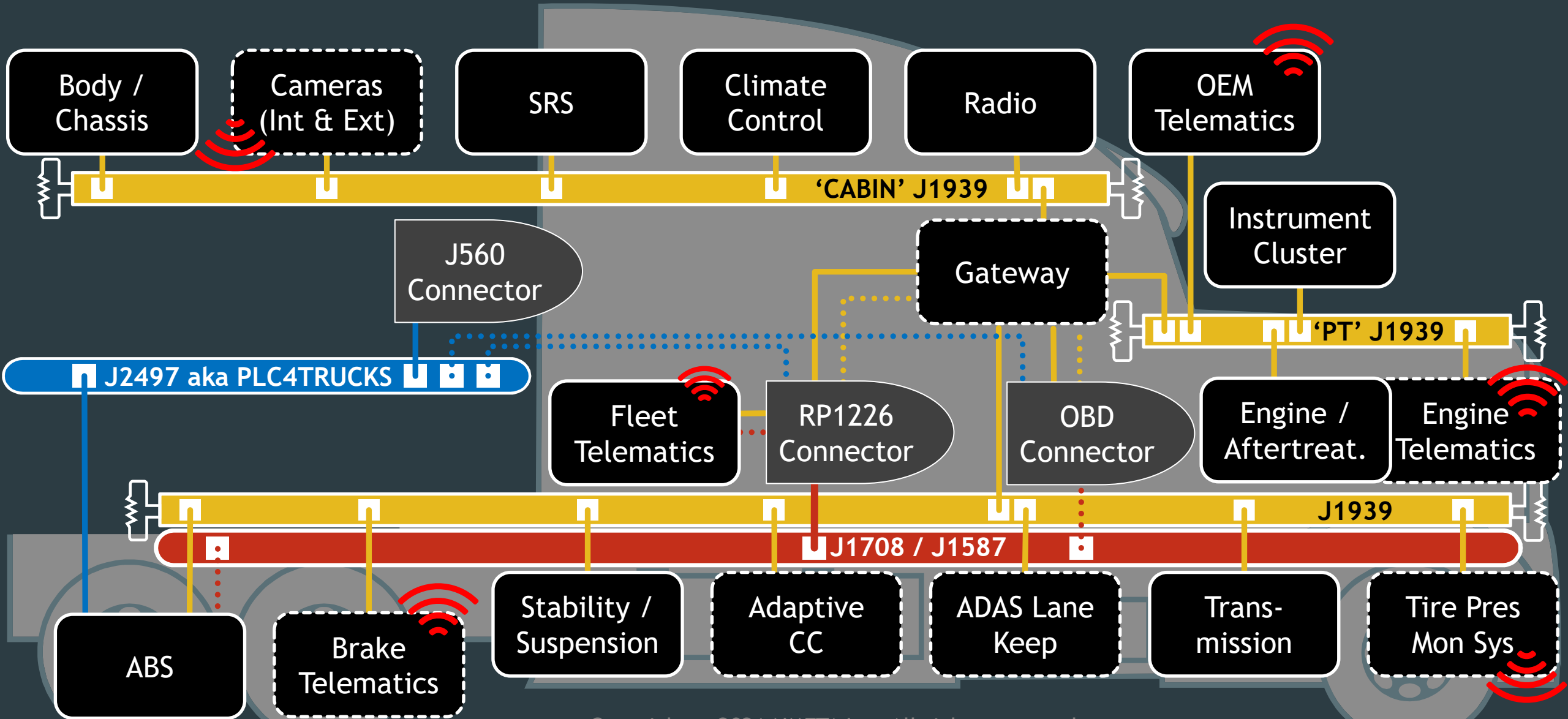
- ▶ The global supply chain links all of us
- ▶ All modes share technologies
  - ▶ e.g. the CanBusHack de-rate disablement abuse is applicable across modes [<https://ioactive.com/guest-urban-johnson-nmfta/>]



CC BY Vectors Market

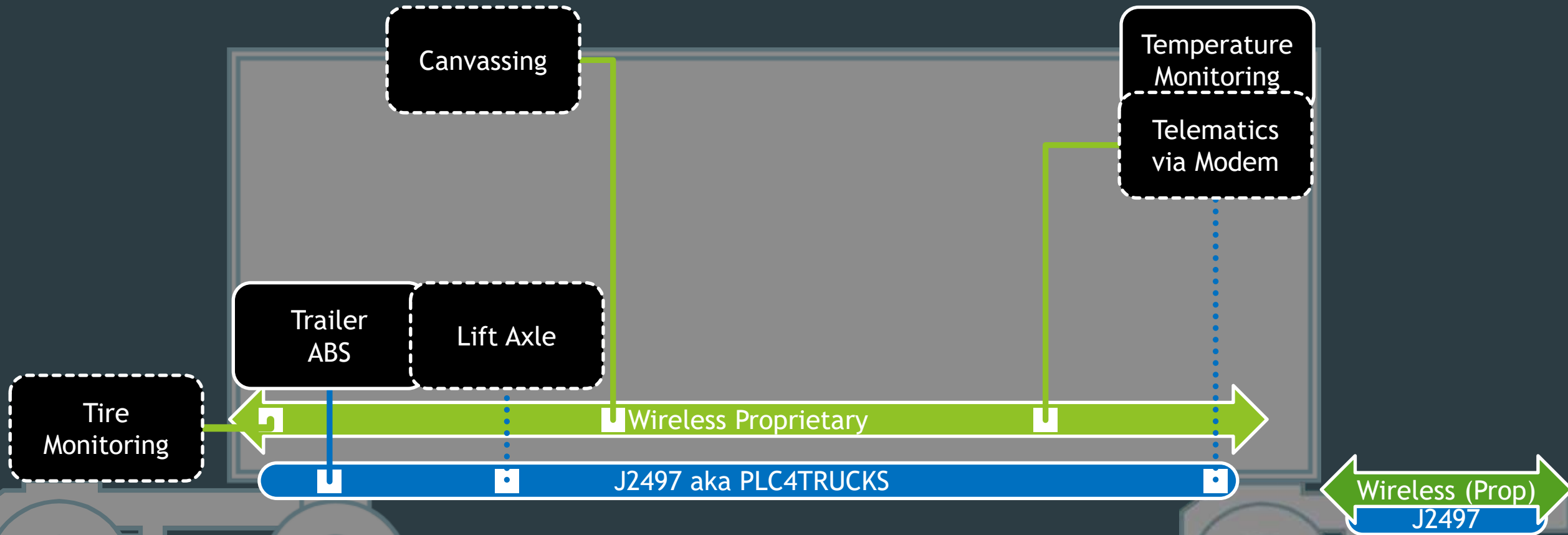
# About Trucks

aka Tractors aka Power Units aka 'The things that roll'.  
If it isn't moving then the fleet is losing money.



# About Trucking: Trailers

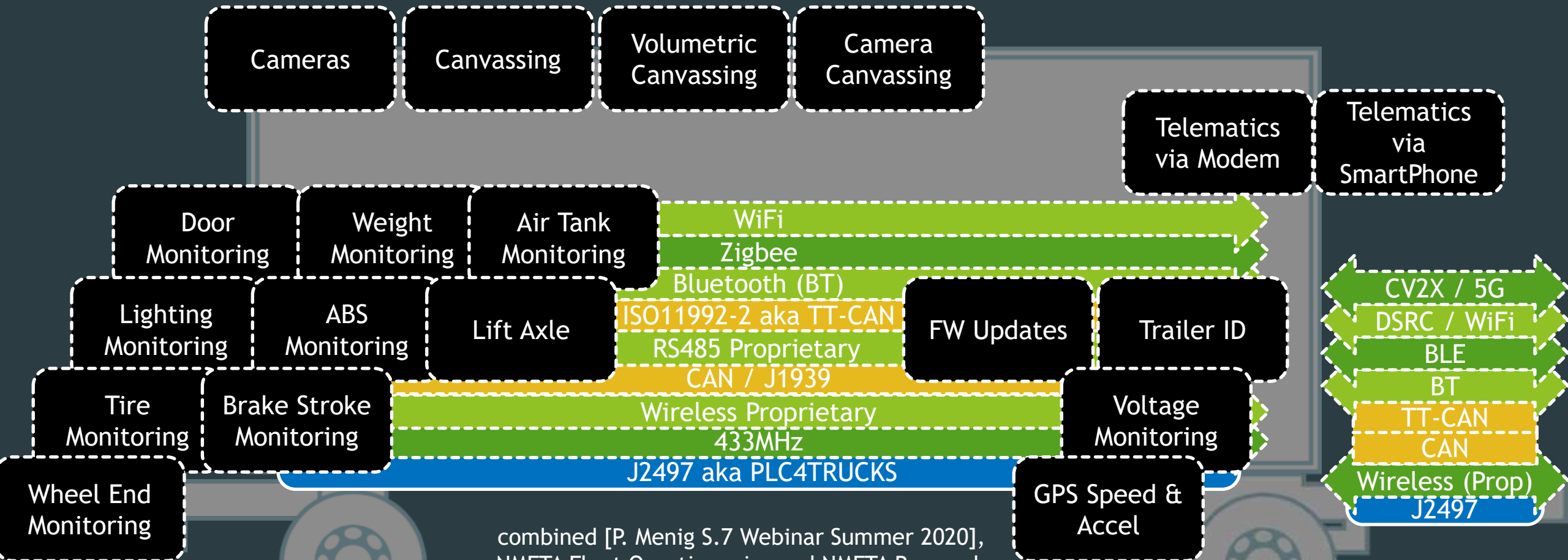
- ▶ The other things that roll
- ▶ Outnumber tractors (in North America)
- ▶ Many features today



based on NMFTA Fleet Questionnaire and NMFTA Research

# About Trucking: Trailers

► And more growing everyday



combined [P. Menig S.7 Webinar Summer 2020],  
NMFTA Fleet Questionnaire and NMFTA Research

# About Trucking: Maintenance

- ▶ Commercial Vehicles are owned for and by commercial motor freight carriers and leasing companies (e.g. Penske).

The fleets protect their investments with preventative maintenance.

- ▶ Tractors spend more time in a service center connected to diagnostics than any passenger car [haystack and sixvolts, [Cheap Tools for Hacking Heavy Trucks](#)]
- ▶ Diagnostics software is authorized to do lots of very powerful things including [Burakova, Hass, Millar, and Weimerskirch, [Truck Hacking: An Experimental Analysis of the SAE J1939 Standard](#)]:
  - ▶ disabling engine cylinders and
  - ▶ cycling ABS pressure valves
- ▶ Most diagnostics software is low-quality windows software



CC BY Oksana Latysheva, UA

# About Trucking: Distribution Centers

- ▶ Fleets make extensive use of distribution or service centers.
  - ▶ e.g. Less Than Truckload (LTL) -> '*terminals*'
- ▶ Trailers spend a lot of time here: either docked or parked
- ▶ Distribution centers have:
  - ▶ A lot of technology
  - ▶ (and a lot of attack surface)
  - ▶ that's a whole other topic



# About Trucking: Intermodal

- ▶ Some fleets make extensive use of *intermodal* aka ‘shipping containers’ which were designed to be able to go from the deck of a ship to a train or tractor-trailer and vice-versa
  - ▶ intermodal also includes the interchange of trailers between trucks and railroads, and trucks and barges or ships
- ▶ Some intermodal containers have networking interconnects to the vehicle networks on which they are being carried
- ▶ Many intermodal containers have their own telematics modems.  
[<https://seanews.co.uk/features/a-world-where-all-shipping-containers-are-smart-and-connected/>]



CC BY Joedamadman



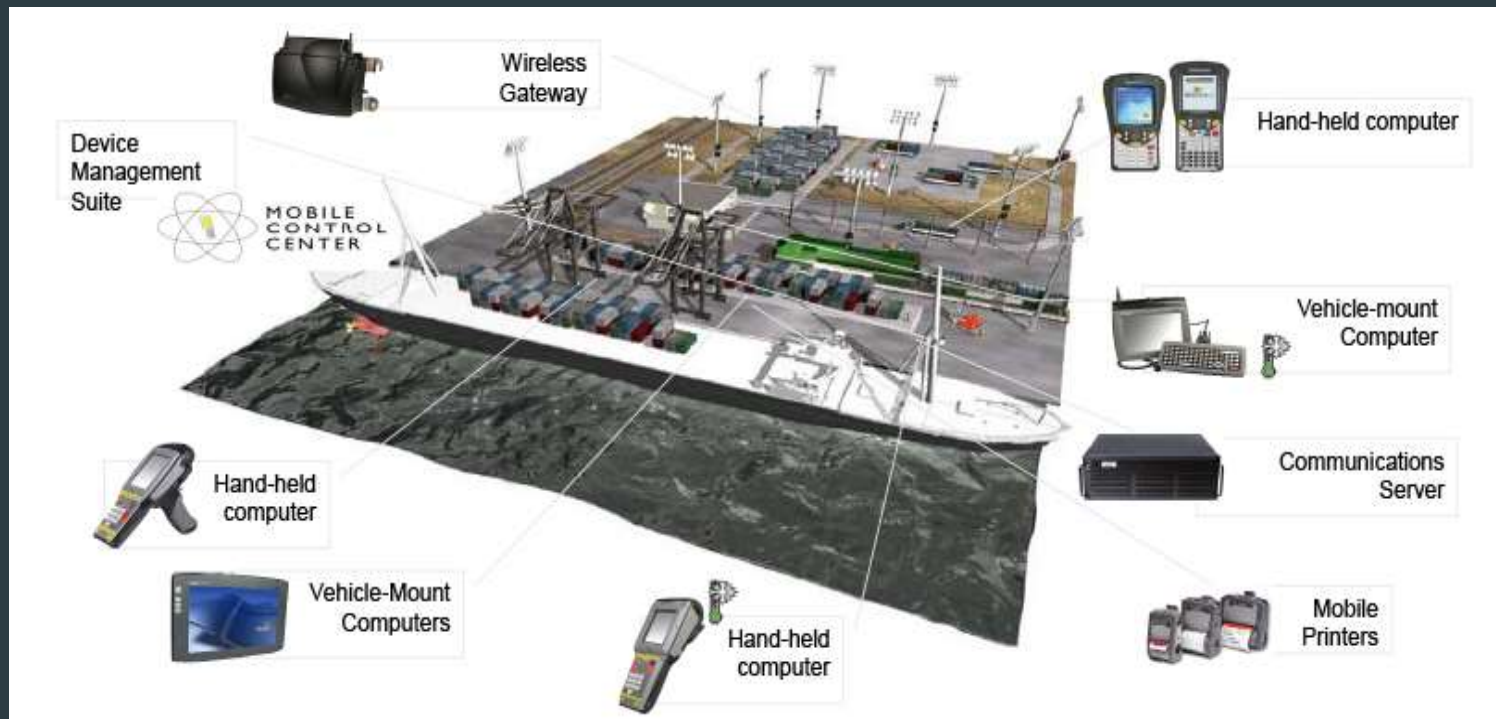
CC BY Doug Wertman



CC SA Photocopy

# About the Other Modes

- ▶ The modes share technologies
  - ▶ e.g. Ships use **J1939** -- there it is called NMEA
  - ▶ e.g. Trains use **J1939** too... Where there's a diesel engine there is **J1939**
- ▶ Also: all of the modes are bolting-on "Internet of Things" (IoT) 'stuff'

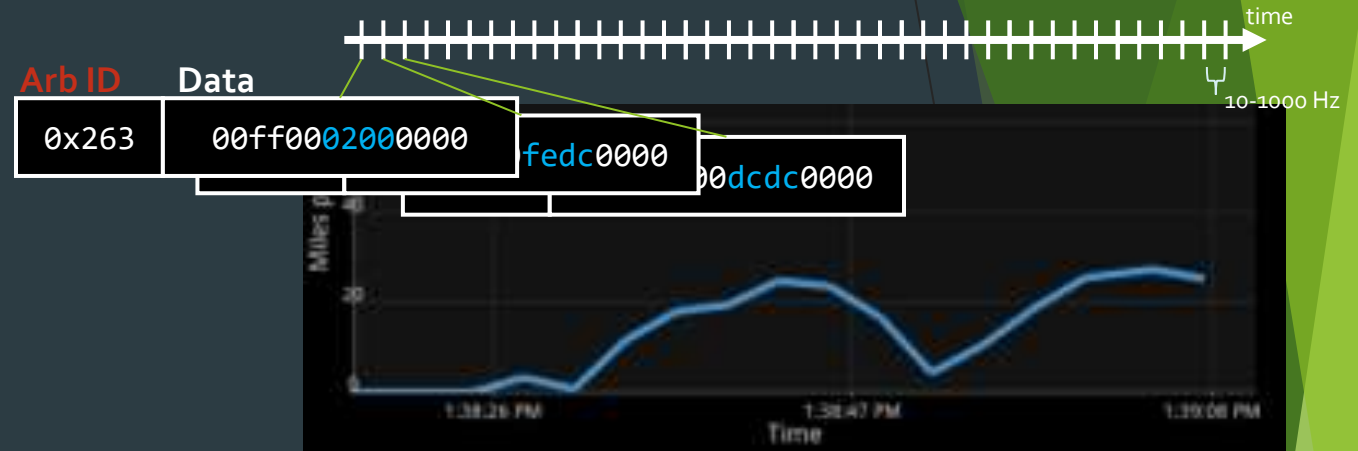


from: <http://www.teamliftss.com/workshop/port-intermodal-solutions/>

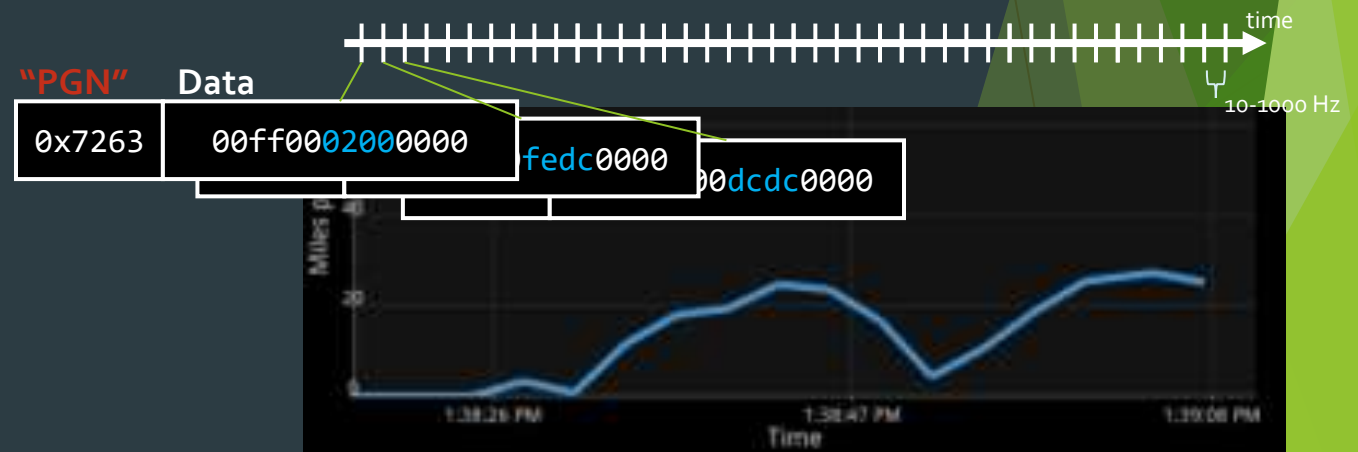
# Truck Vehicle Networks: J1939

# J1939 in relation to CAN in Passenger Cars

- ▶ Both: encoding time-varying signals into bitfield locations and diagnostics
- ▶ Passenger cars:
  1. *proprietary Arbitration ID*,
  2. *proprietary bitfield locations*,
  3. standard diagnostics (mostly)
- ▶ J1939:
  1. standard **PGNs** (mostly),
  2. standard **SPNs** (mostly),
  3. *proprietary diagnostics*



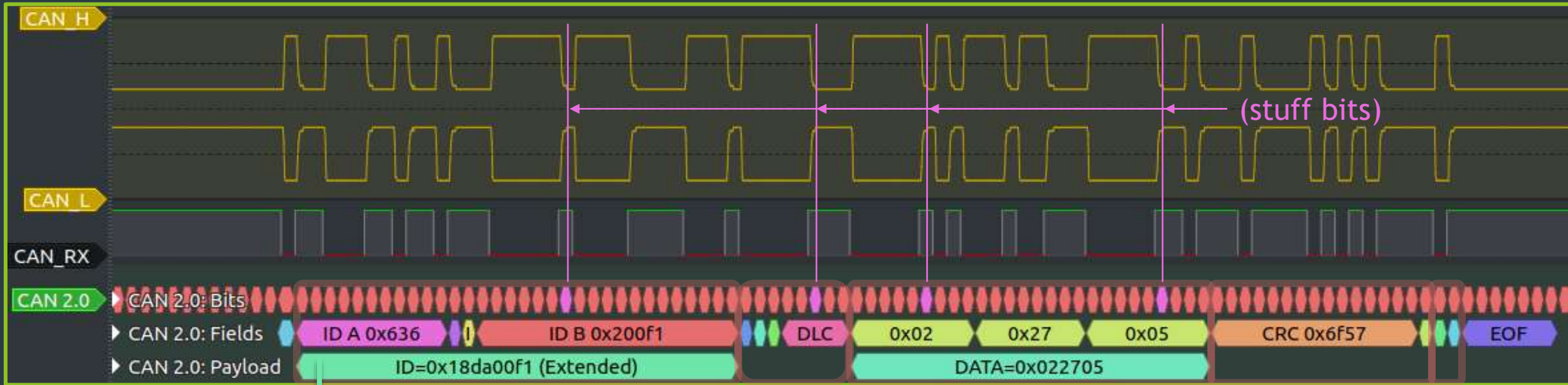
Ford OpenXC Platform 2017



Ford OpenXC Platform 2017

# J1939 Specifics: CAN Frames

sigrok with [kentindell/canhack](#) can2 decoder:



29bit Arbitration ID

Control Field

Data Field (8byte max)

Error Checking

ACK

	ID A											ID B																						
Arb ID bits (host):	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
for unicast:						PGN >> 8						Dest Addr																						
for bcast:						PGN																												
for all:	priority		res	page																			Source Addr											


# J1939 Features

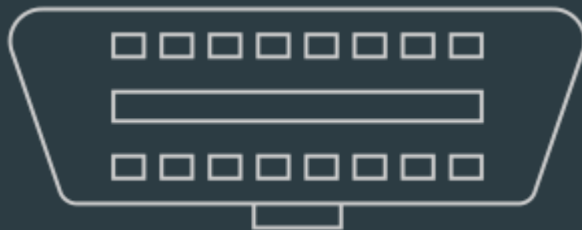
- ▶ Both **unicast** (PGNs < 0xF000) & **broadcast** (>= 0xF000)
- ▶ Transport fragmentation and reassembly (PGNs **0xEC00** and **0xEB00**)
- ▶ Address claiming (**0xEE00**)
- ▶ Request of PGNs (**0xEA00**)
- ▶ Proprietary messages:
  - ▶ destination-specific (propA **0xEF00**, propA2 **0x1EF00**) and
  - ▶ broadcast (propB0 **0xFF00-0xFFFF**, propB1 **0x1FF00-0x1FFFF**)
- ▶ Dump, reconfigure, reflash (☞ **'the fun stuff'**) is all protected by a challenge-response system called *Seed-Key Exchange*
  - ▶ over ISO 15765-2 aka *ISO-TP* for UDS (**0xDA00**)

- 
- ▶ For more details see Hannah Silva's CyberTruck Challenge™ 2021 Training [www.cybertruckchallenge.org/wp-content/uploads/2021/08/Truck-Networks-Print.pdf](http://www.cybertruckchallenge.org/wp-content/uploads/2021/08/Truck-Networks-Print.pdf)



# Finding J1939 (1/6)

- ▶ In-cab or On-Board Diagnostics J1939 connector
- ▶ Black or Green
- ▶ Some OEMs use the passcar OBD-II connector. 



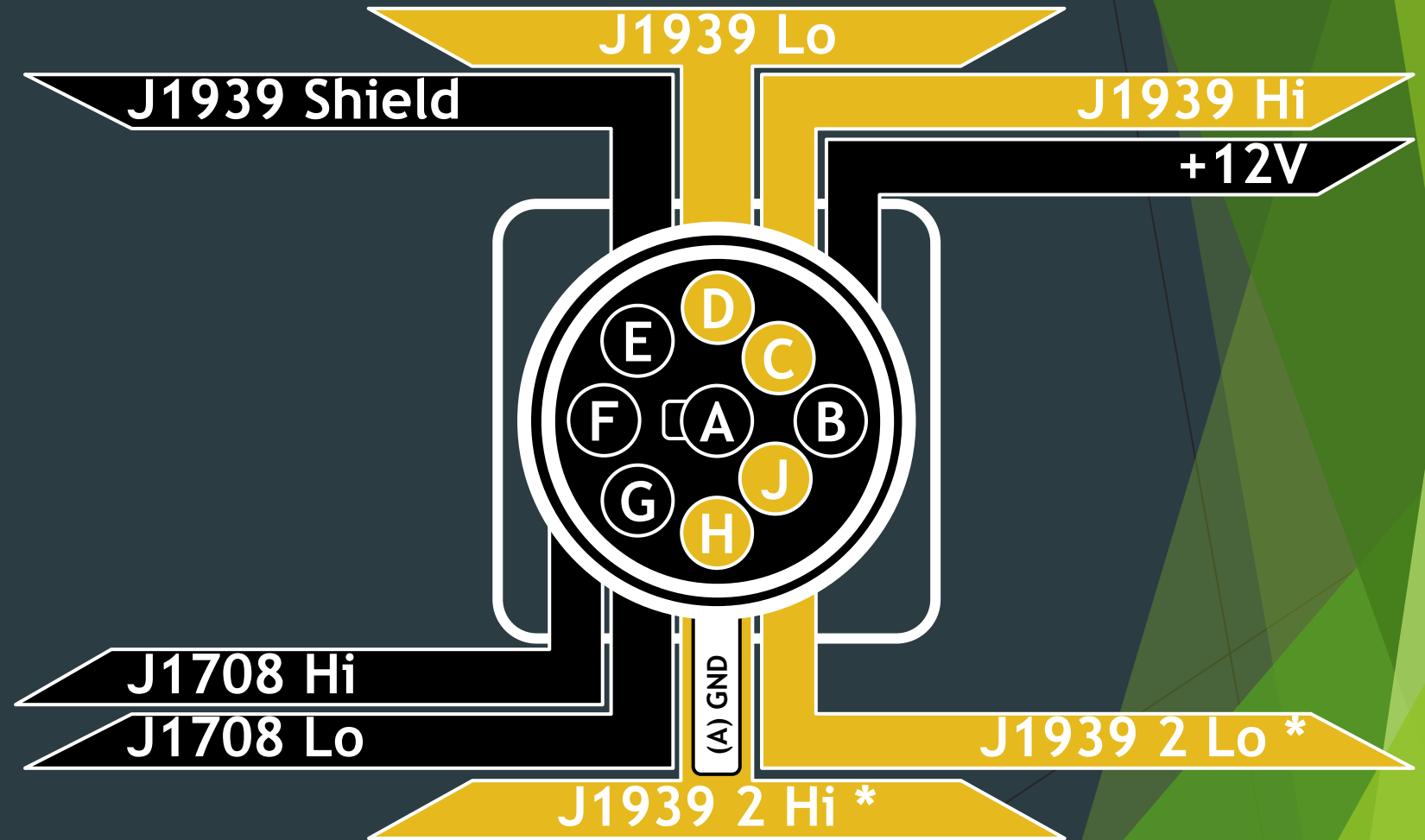
CC BY-SA Florian-schäffer



Dr. Jeremy Daily

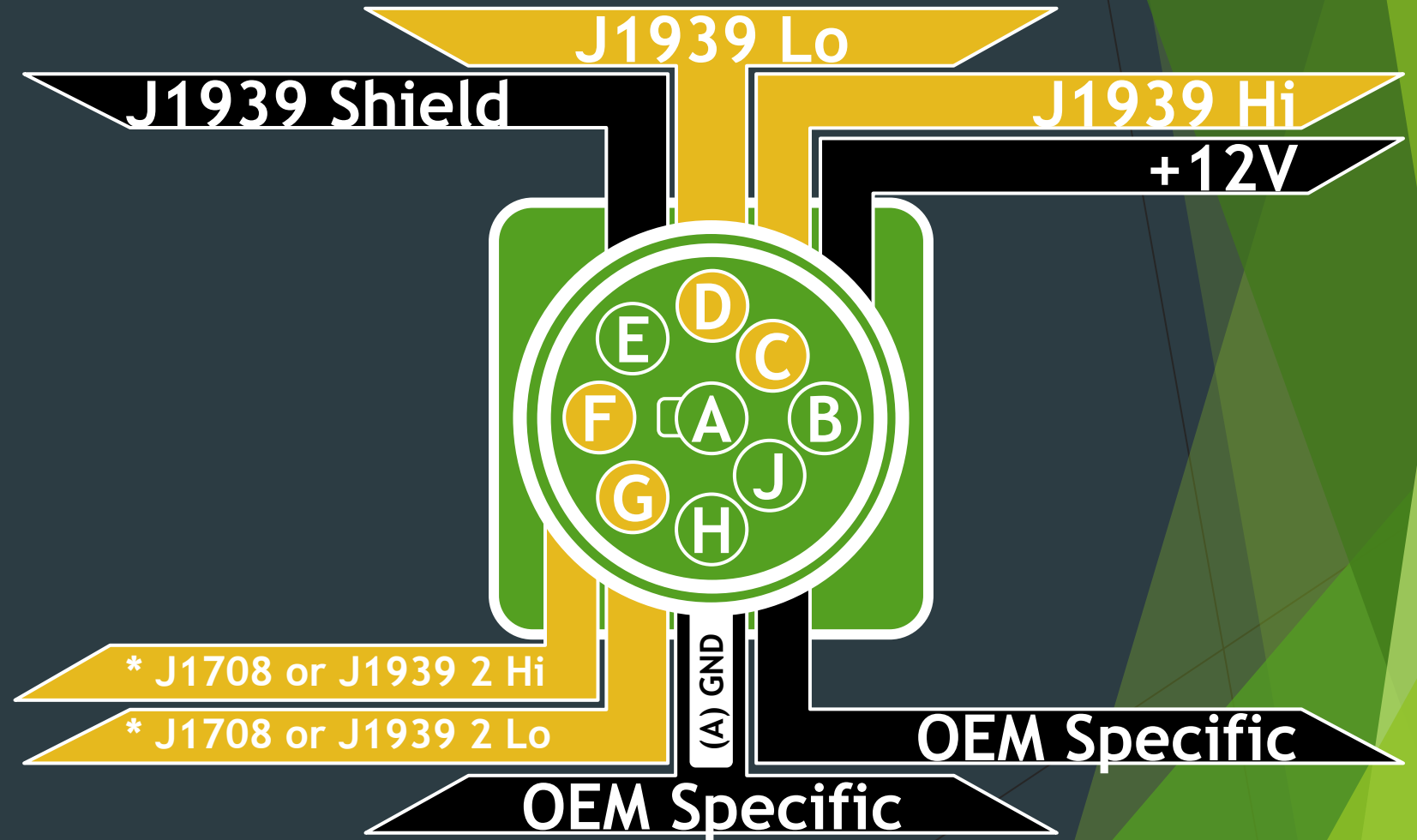
# Finding J1939 (2/6)

- ▶ On black socket
- ▶ (\* means optional)



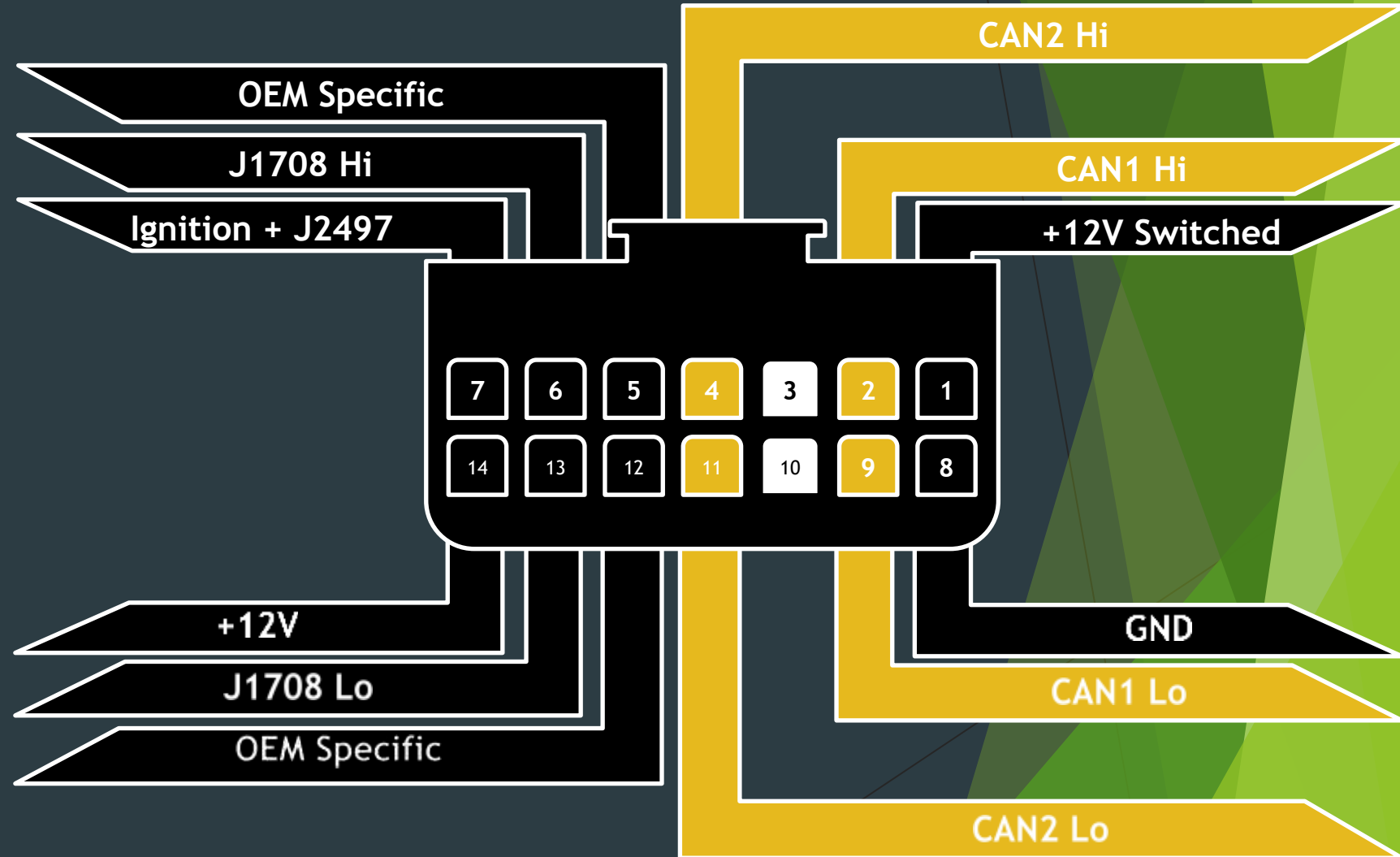
# Finding J1939 (3/6)

- ▶ On green socket




# Finding J1939 (4/6)

- ▶ On the RP1226 (Aftermarket/Telematics) Connector
- ▶ Found behind dash or in berth



# Finding J1939 (5/6)

- ▶ Other wires in the truck too...
  - ▶ ~6 separate CAN segments 

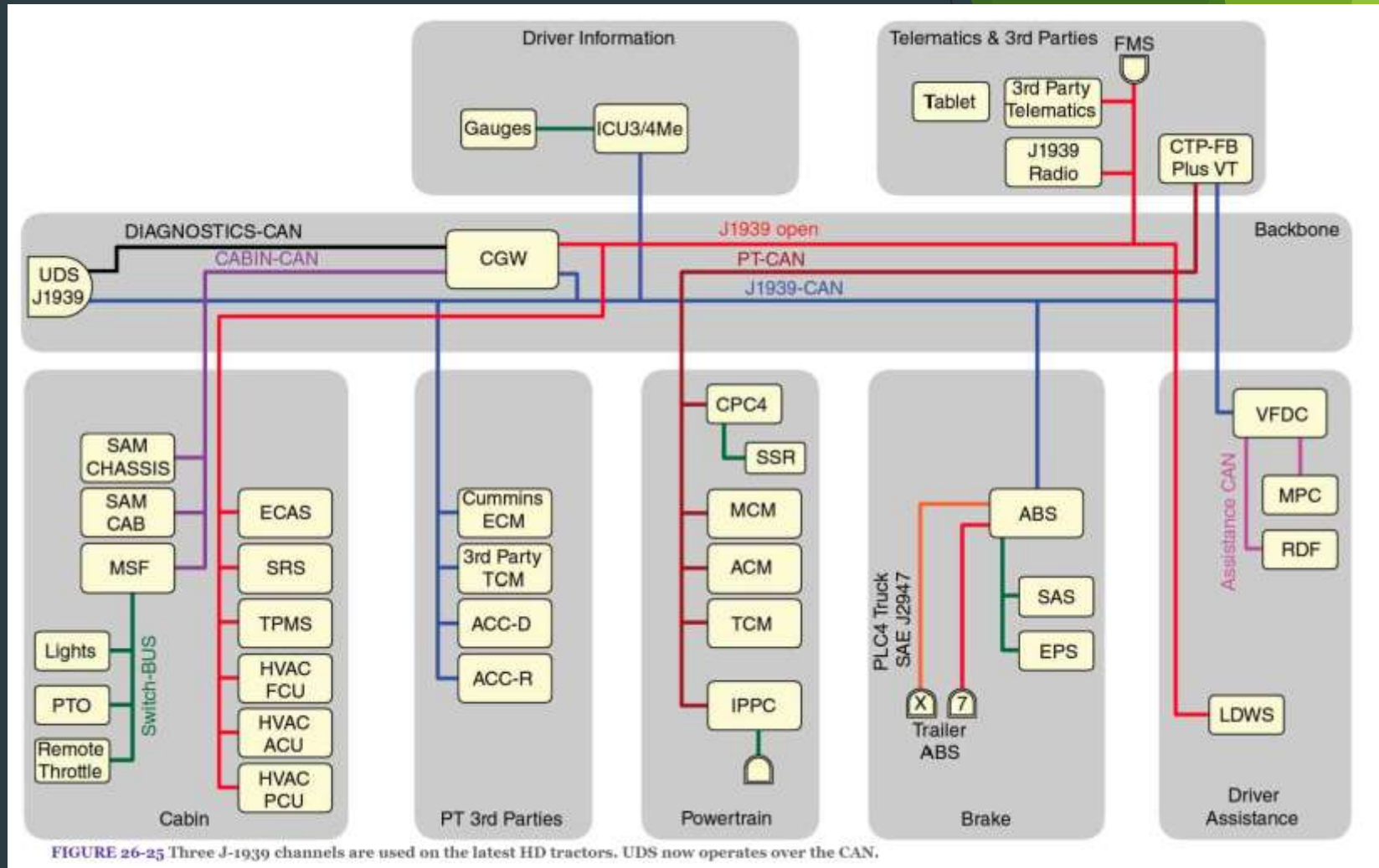
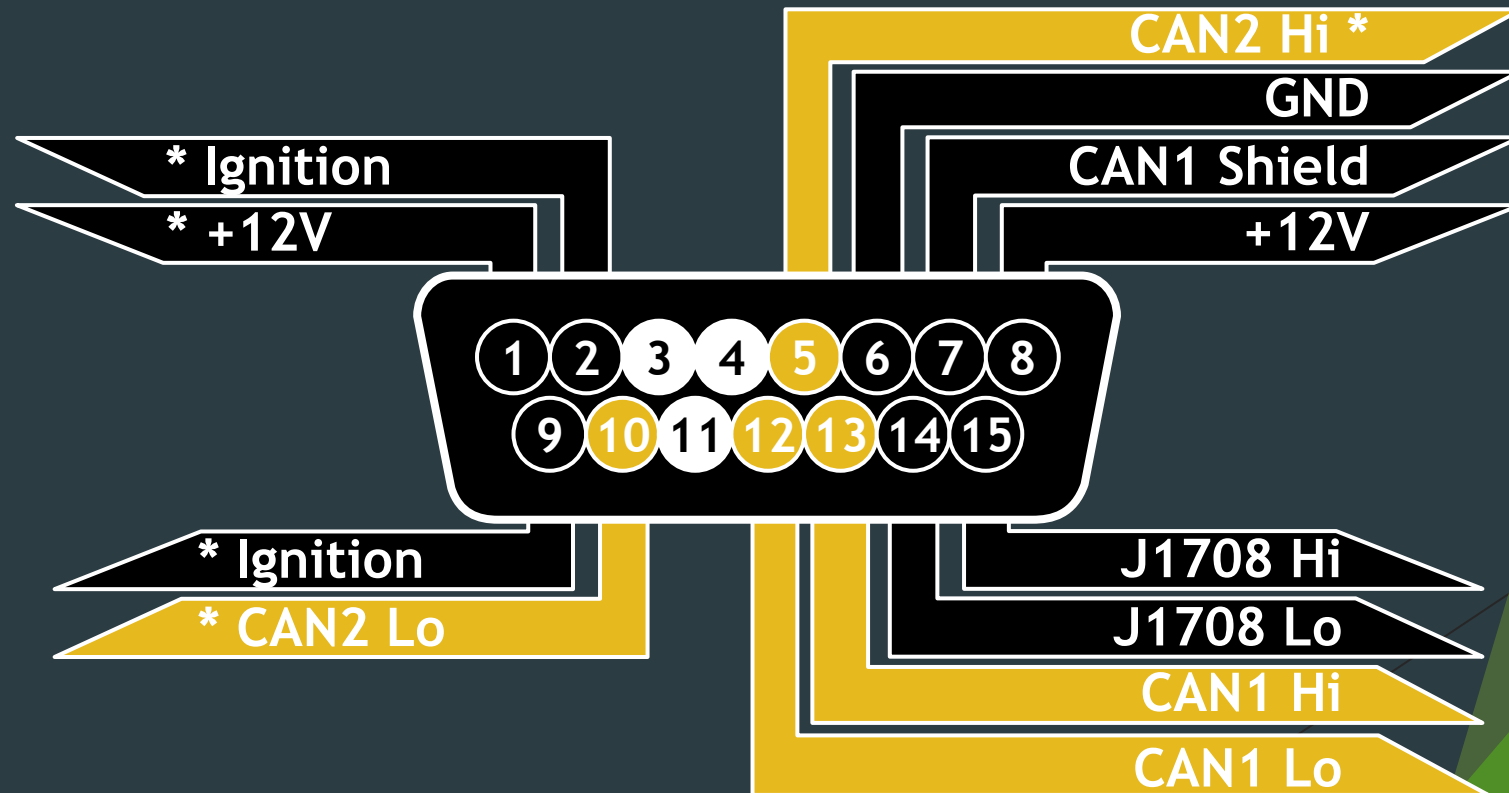


FIGURE 26-25 Three J-1939 channels are used on the latest HD tractors. UDS now operates over the CAN.

[Duffy, Owen C., and Gus Wright. *Fundamentals of Medium/Heavy Duty Commercial Vehicle Systems*: 2014 NATEF Edition. Jones & Bartlett Publishers, 2015]

## Finding J1939 (6/6)

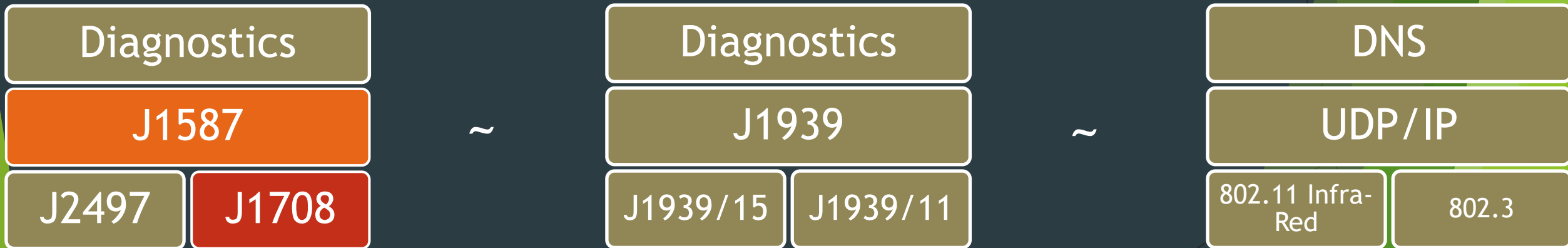
- ▶ On a DB-15 Connector (common in heavy vehicle cables)



# Truck Vehicle Networks: J1708/J1587

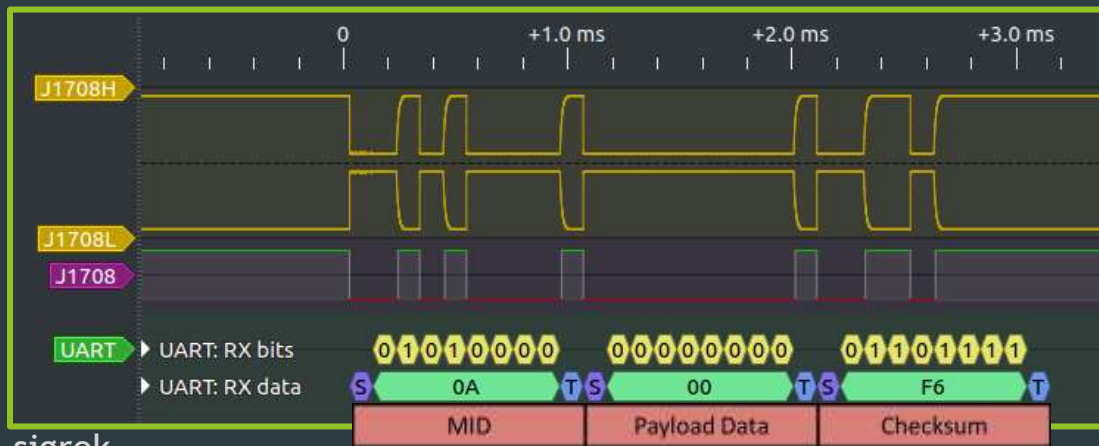
# J1708/J1587 Specifics (1/3)

- ▶ Predates J1939 by many years. Sometimes still found in the tractor. Always still found in the Trailer as **J2497** (more on that later).
- ▶ J1708/J1587 by analogy:



# J1708/J1587 Specifics (2/3)

- ▶ Has similar bus arbitration to CAN: lowest **first byte** wins.
- ▶ 9600bps / 8N1
- ▶ Very much like an RS-485 bus at physical layer
- ▶ Has RT constraints for framing and bus arbitration
- ▶ The **first byte** is like a source address: the **MID**
- ▶ Some noteworthy **MIDs** from the specs (**J1708/J1587/J2497**)
  - ▶ **111** is used for factory test
  - ▶ **128-255** are **defined by J1587**
  - ▶ **64 & 172** are off-board diagnostics
  - ▶ **48 & 153** are on-board diagnostics
  - ▶ **182** is off-board programming
  - ▶ **163** is ‘vehicle security’
  - ▶ **207** is for drivetrain bridge
  - ▶ **217 & 218** tractor & trailer bridges
  - ▶ **87** is for **J2497** active ABS event
  - ▶ **125** is for **J2497** identification
  - ▶ **10 (0x0a)** and **11 (0x0b)** are **J2497** lamp on/off



sigrok

# J1708/J1587 Specifics (3/3)

- ▶ signals are identified by a **PID** byte prepended to the signal
- ▶ can be **multiple PIDs** in one J1587 frame



12
34

 - byte

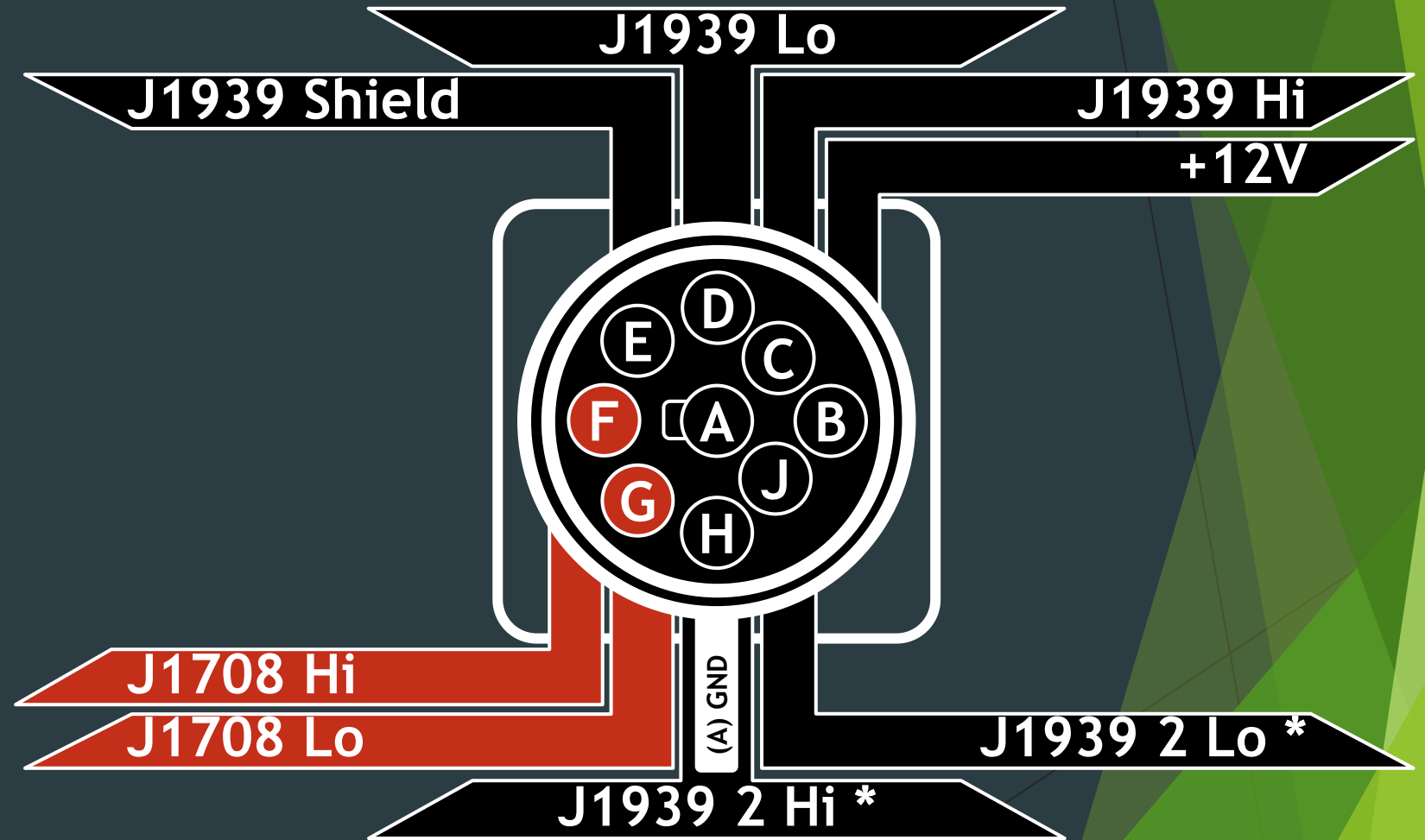
- ▶ **PIDs range: 0-1021**
  - ▶ **PIDs > 255** use multi-byte PID extension
- ▶ Decoding of PIDs is done by reference to the J1587 specification.
  - ▶ There are tools that can convert the SAE PDF into a database and do decode.

# J1587 Features

- ▶ Mostly **broadcast** (some **unicast**)
- ▶ Requests for data:
  - ▶ PID **0** (broadcast) / PID **128**: Component specific (unicast)
- ▶ Has fragmentation and reassembly (Frames *should be* less than 21 bytes if the vehicle is in motion)
  - ▶ PID **192**: ‘multisection’ parameter (broadcast)
  - ▶ PID **197** and **198**: transport protocol (unicast)
- ▶ ‘Standardized Free-Format Data’ requests on transport protocol
  - ▶ e.g. ‘Programmable Params’ / ‘Calibration’, ‘Executable Code’
- ▶ Proprietary messages: ‘**Data Link Escape**’ (unicast) 🖱️ ‘the fun stuff’
  - ▶ PID **254** and **510**
  - ▶ e.g. “**AC FE 80 F0 17**”  
is from MID **0xAC** to ‘MID’ **0x80**

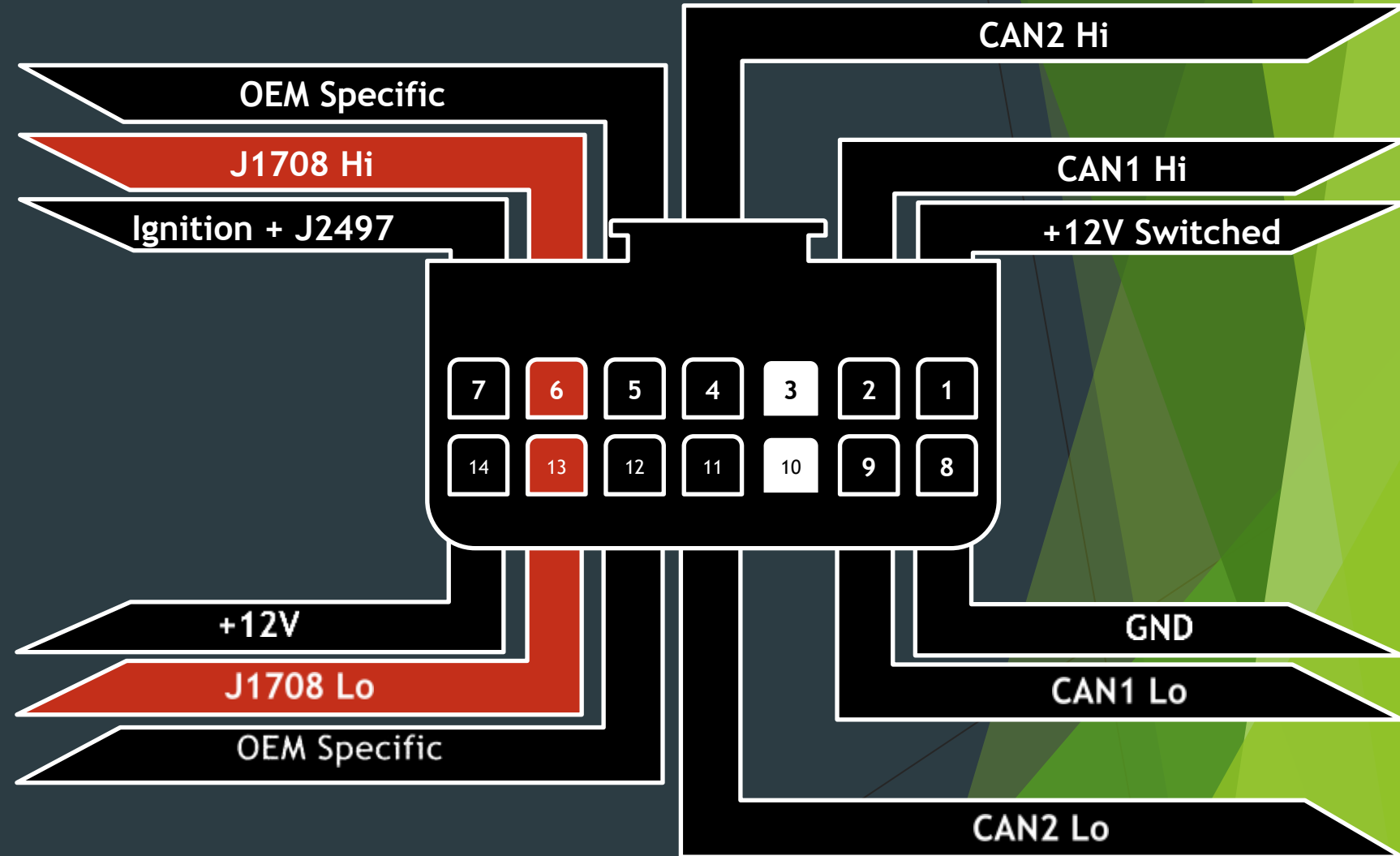
# Finding J1708/J1587 (1/3)

- ▶ Present on black socket
- ▶ Optional on green socket



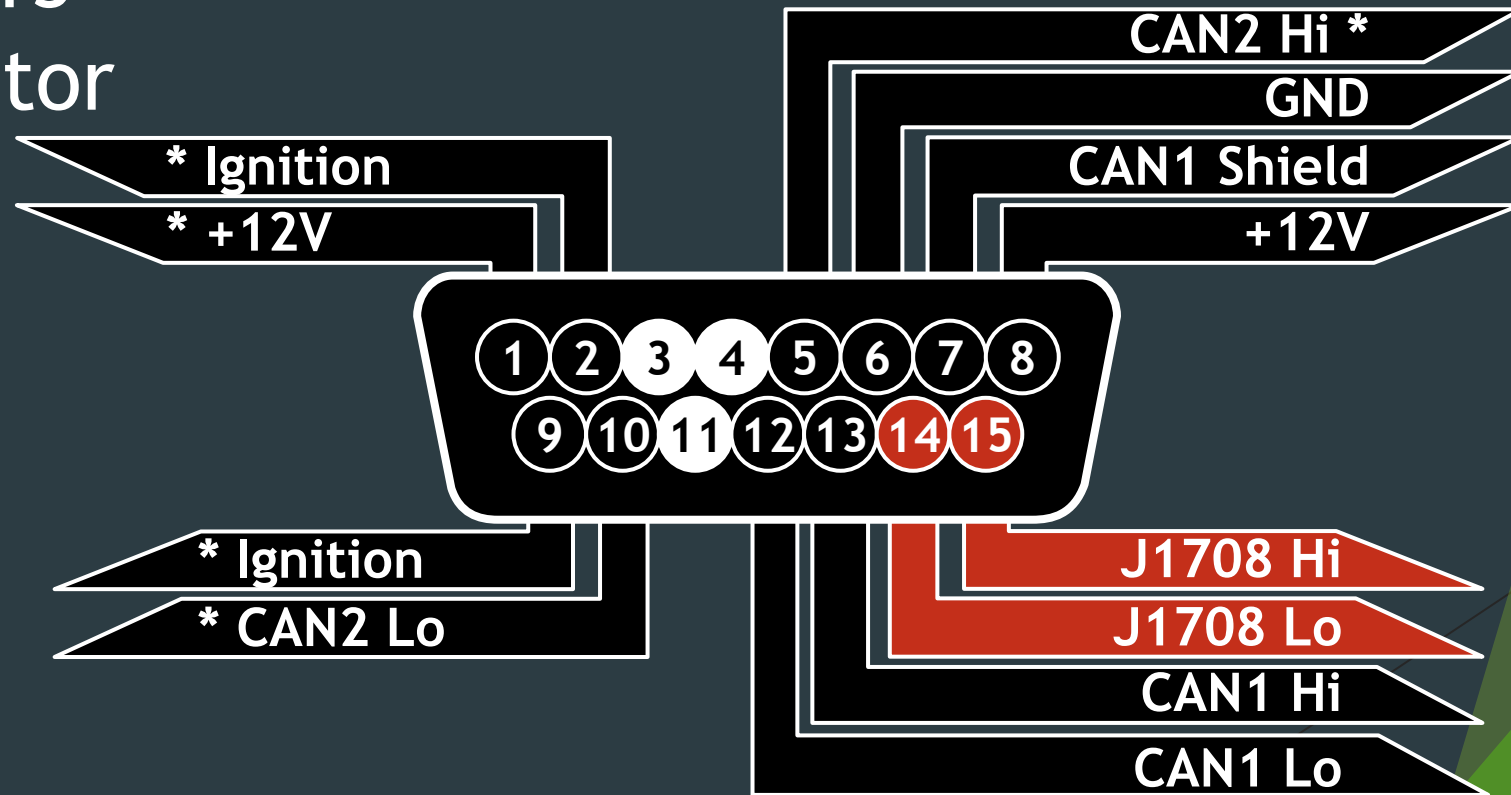
# Finding J1708/J1587 (2/3)

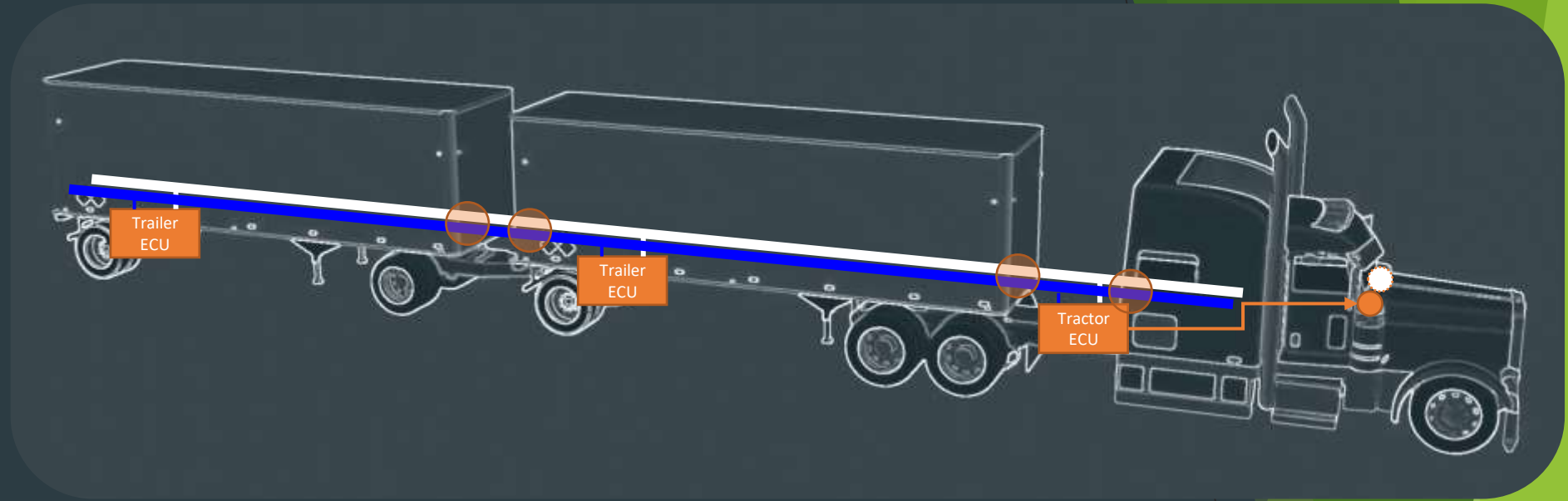
- ▶ On the RP1226 (Aftermarket/Telematics) Connector



# Finding J1708/J1587 (3/3)

## ► On DB-15 Connector





# Truck (and Trailer) Vehicle Networks: J2497

## J2497 Specifics (1/2)

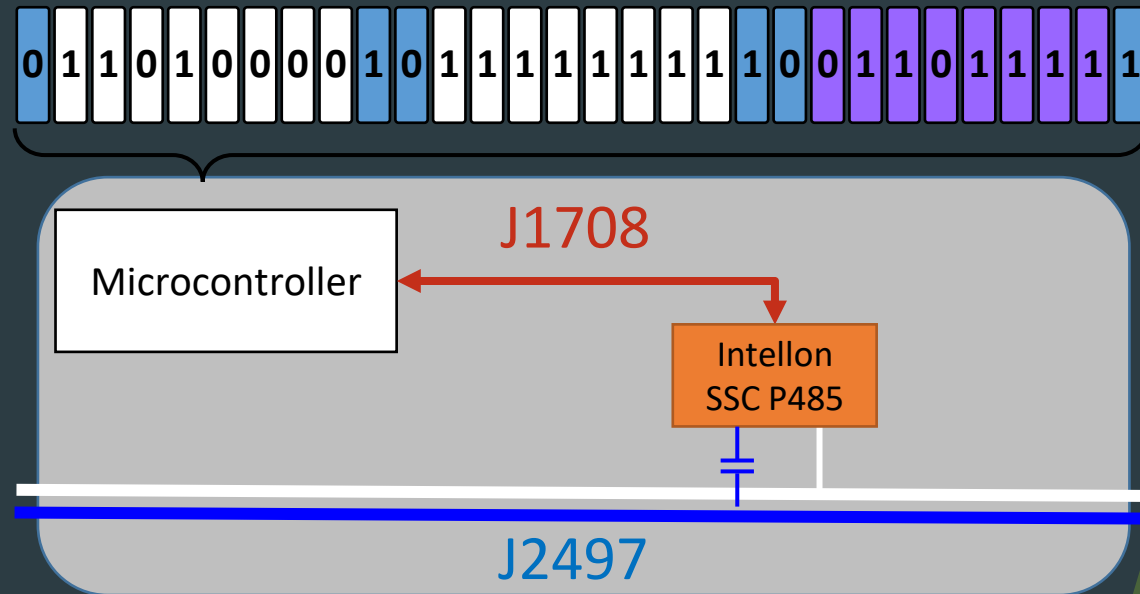
- ▶ Roughly speaking, it is “**J1708** over trailer power lines”
- ▶ a.k.a. *PLC4TRUCKS*
- ▶ Again by analogy:



# J2497 Specifics (2/2)

▶ J1708 ↔ J2497

▶ Implemented almost exclusively by the Intellon SSC P485 chip



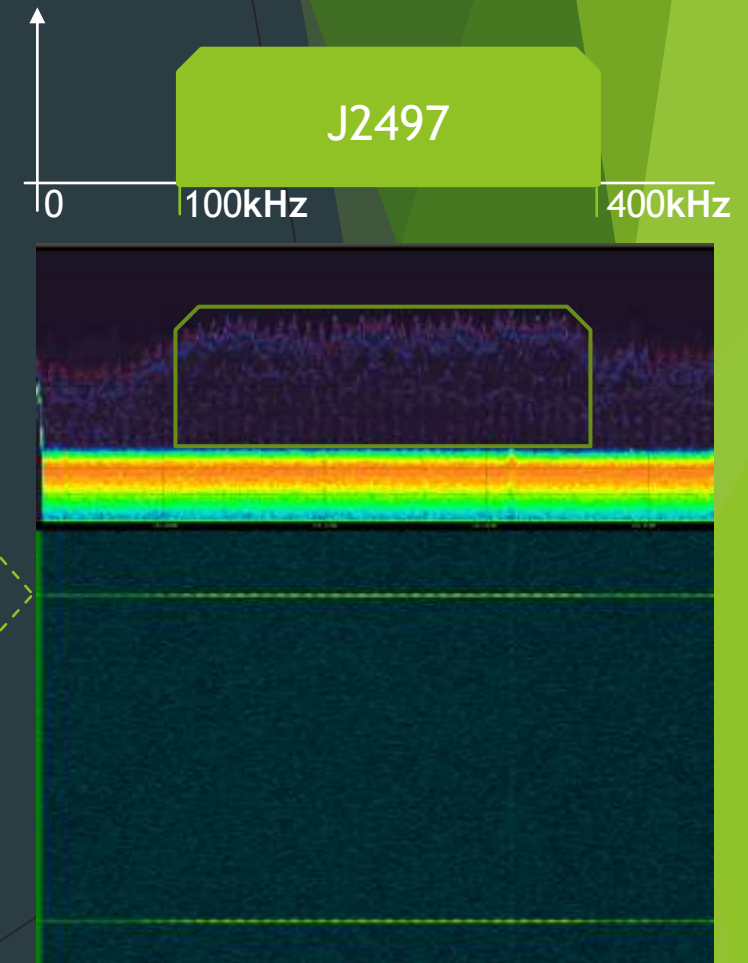
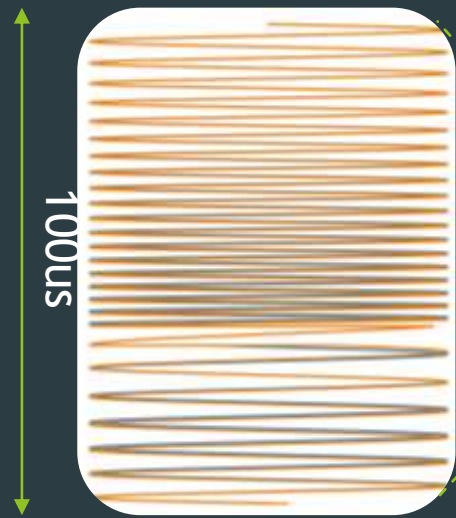
0 1 1 0 1 0 0 0 0 1 0 1 1 1 1 1 1 1 1 1 0 0 1 1 0 1 1 1 1 1

0 0 0 1 1 0 1 0 0 0 0 1 1 1 1 1 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 1 1 1 1 1 1 0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1



# J2497 Specifics (3/4)

- ▶ The chirps are 100us in duration, between 2.5 and 7V peak-peak
- ▶ The chirps sweep from 203KHz through 400KHz (63us) then to 100KHz (4us) and back to 203KHz (33us) to finish



# J2497 Specifics (4/4)

## Preamble

- Amplitude Shift Keying (ASK)
- Bit time 114us (14us silence after 100us chirp)
- Logic '0' = chirp present
- Initial symbols (1-2 logic '0')
- Start bit (logic '0')
- MID bits (duplicated in body)
- Stop bit (logic '1')



## Body

- Phase Shift Keying (PSK), 180deg difference
- Bit time 100us
- Logic '0' symbol is arbitrary per device, determined by the symbol transmitted in the preamble
- Sync symbols (5 logic '1')
- J1708 Body Bytes. MID followed by Data
  - Start bit (logic '0')
  - Data bits (8)
  - Stop bit (logic '1')
- J1708 Checksum Byte
  - Start bit (logic '0')
  - Checksum bits (8)
  - Stop bit (logic '1')
- Gap (0-4 logic '1') & End symbols (5 logic '1')




# J2497 Features

- ▶ Primary purpose is for **0a00** and **0bff** LAMP ON and LAMP OFF messages. But *there's more*:
- ▶ Has **all** the feature of **J1708/J1587** plus:
  - ▶ dynamic address (MID) claim (PID **4**)
  - ▶ data transfer *bridging* (PIDs **204** and **460**)
- ▶ Trailer brake diagnostic functions such as ABS air pressure valve cycling and ECU reconfiguration
- ▶ Some trailer brake ECUs have scripting languages programmable over J2497
- ▶ because of the added preamble/MID byte it is possible to create J2497 frames that override bus arbitration
  - ▶ e.g. a J2497 priority of maximum **00** and a J1708 priority of minimum **ff** which overrides all J2497 traffic but is received as MID **ff**
- ▶ Radiates enough energy to be read remotely at 6ft from trailer



# Finding J2497 (1/5)

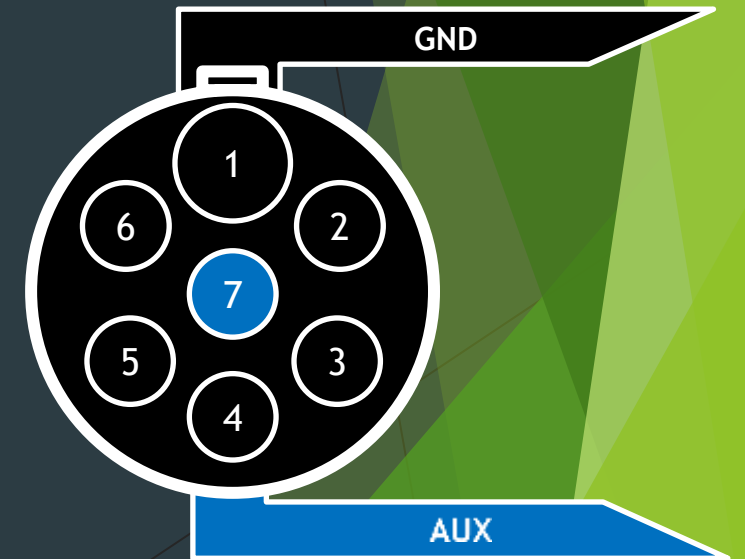
- ▶ Will always be on the power pin (**AUX**) of the trailer J560 connector  (at back of tractor / front of trailer)



CC BY-SA MobiusDaXter

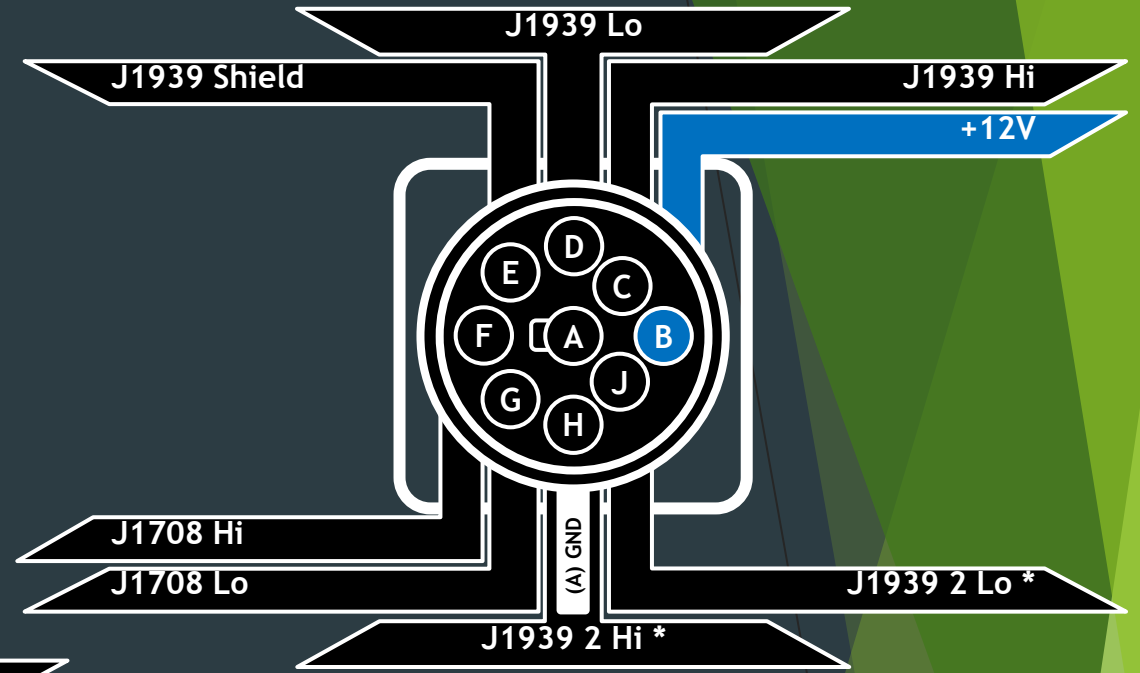
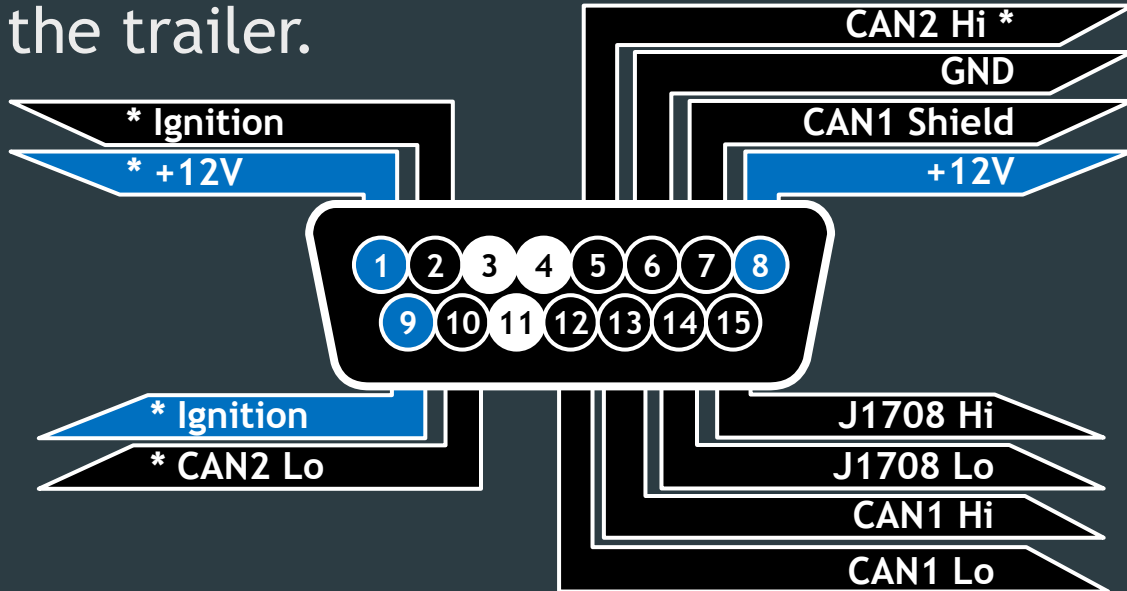


[www.ebay.ca/itm/Bendix-ABS-Trailer-Remote-Diagnostic-Unit-TRDU-PLC-Adapter-9-pin-Connection](https://www.ebay.ca/itm/Bendix-ABS-Trailer-Remote-Diagnostic-Unit-TRDU-PLC-Adapter-9-pin-Connection)



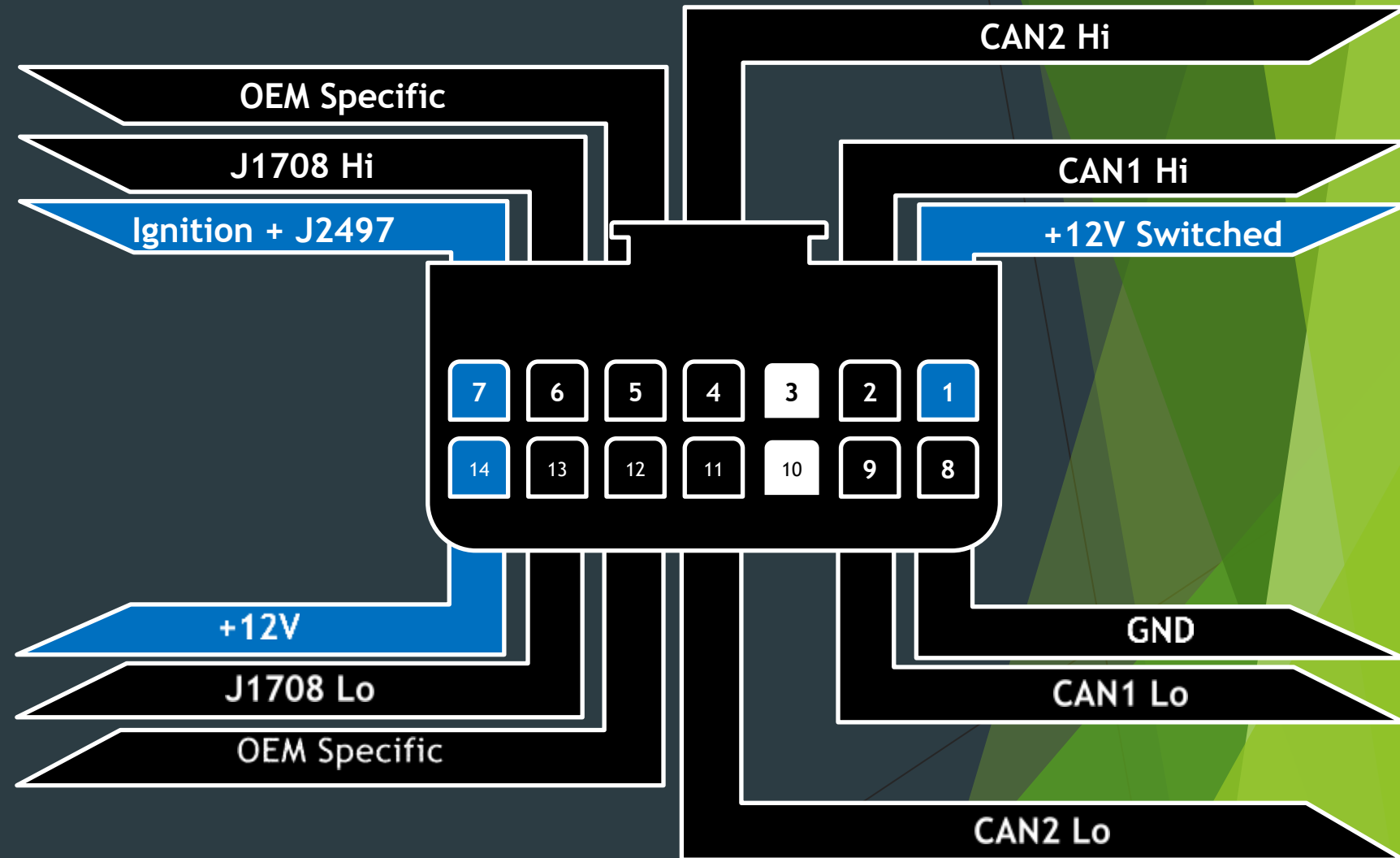
# Finding J2497 (2/5)

- ▶ Might be on the power pins of the diagnostics connector
- ▶ What you find could be filtered/segmented from the trailer.



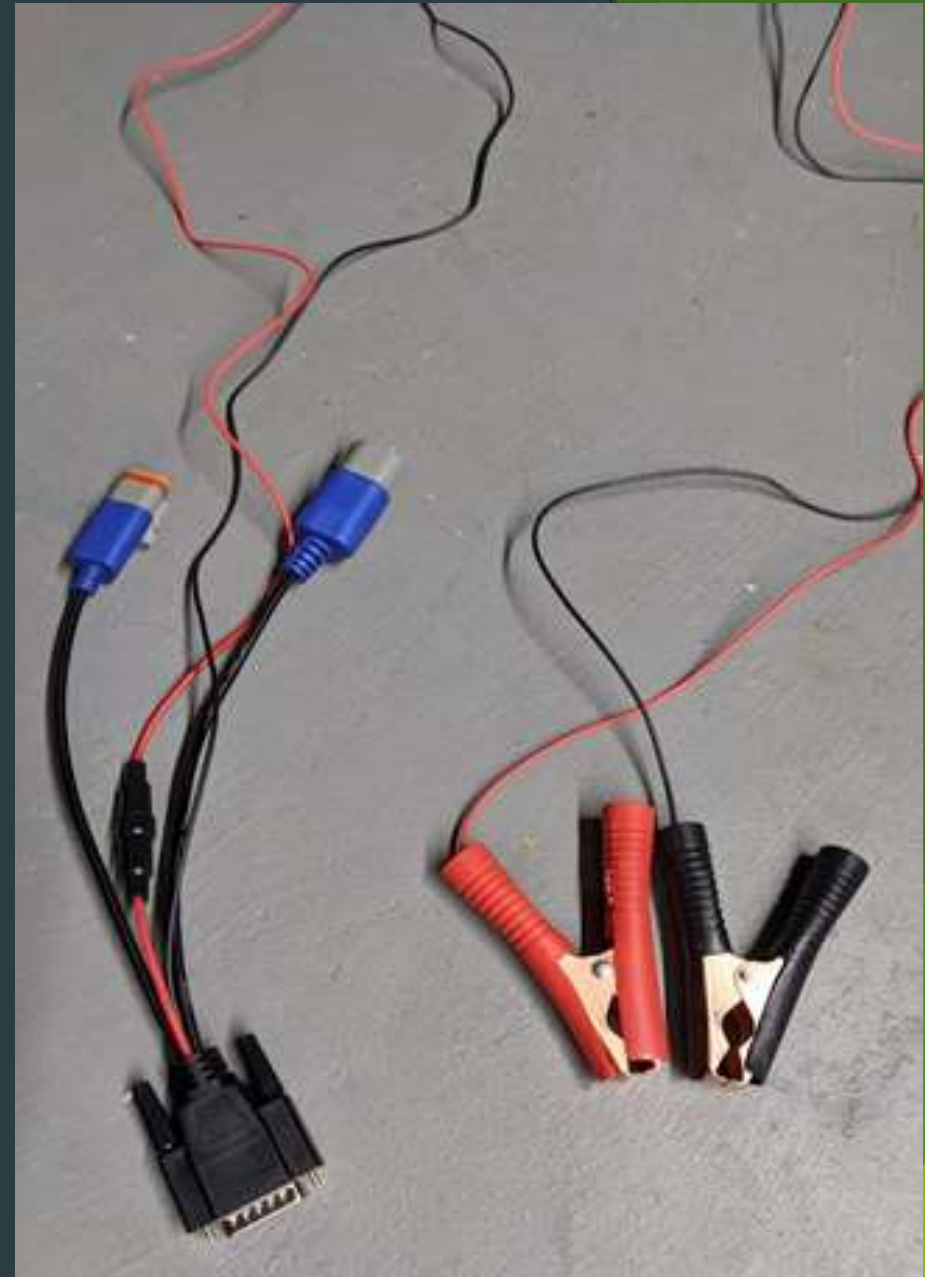
# Finding J2497 (3/5)

- ▶ Should be on  
the RP1226  
(Aftermarket/  
Telematics)  
Connector



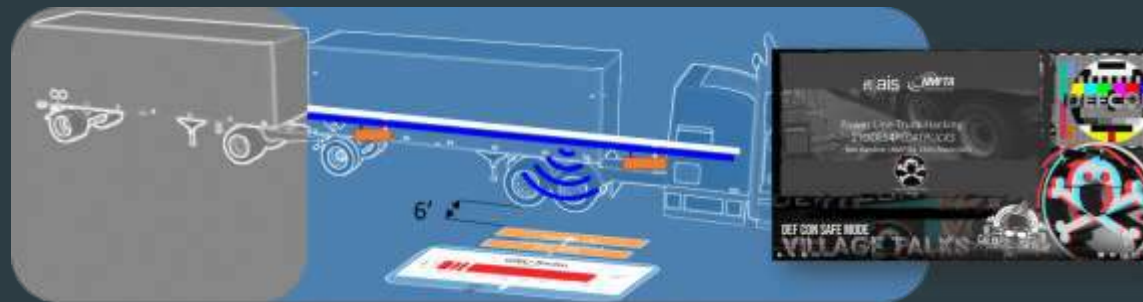
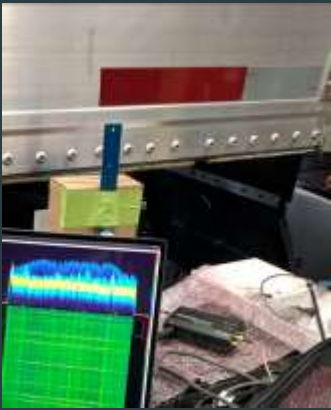
## Finding J2497 (4/5)

- ▶ Might be on the battery terminals -- but what you find could be filtered/segmented from the trailer.



# Finding J2497 (5/5)

- ▶ Might just radiate away from the trailer. [ICSA-20-219-01](#)



Chris Poore & Ben Gardiner. [Power Line Truck Hacking: 2TOOLS4PLC4TRUCKS](#)

- ▶ Might be writable via RF. [ICSA-22-063-01](#)



Chris Poore & Ben Gardiner. [Trailer Shouting, DEF CON 30](#) :

# Other Vehicle Networks

# Vehicle Networks: Intermodal

- ▶ **J1939** is found wherever there's a diesel engine
  - ▶ Probably also **J1708/J1587**
- ▶ **J2497** might be found on containers

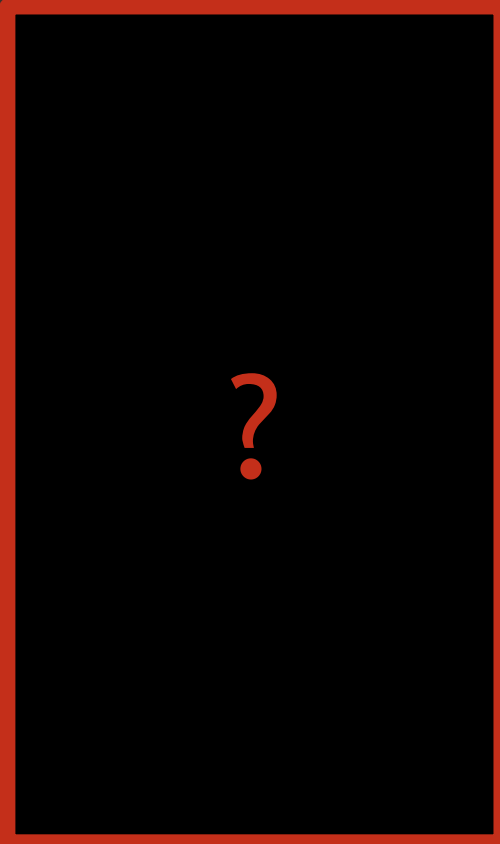
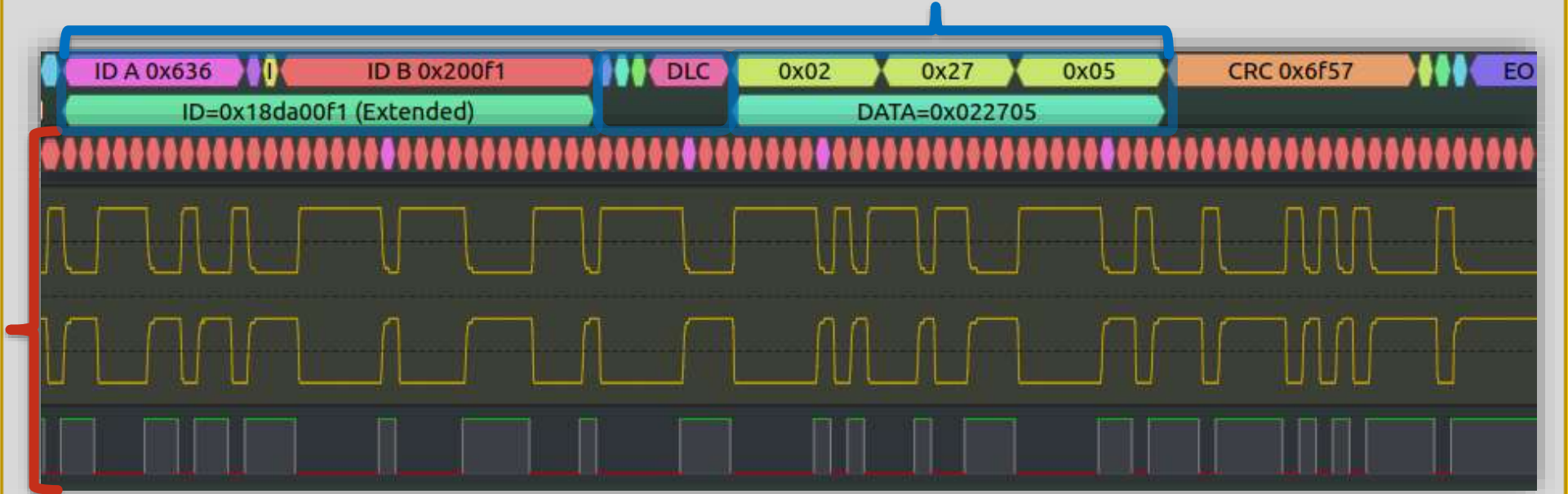
# Vehicle Networks: More

- ▶ LIN
- ▶ CAN-FD, CAN-HG
- ▶ Automotive Ethernet (BroadR-Reach)
- ▶ Much More Wireless

**‘HACKING’**

# CAN Attack Methods (below J1939)

	ALL e.g. Socket CAN	
Bus Flood	Y	<pre>while True:     sock.send(b' \x00\x00\x00\x00\x01\x00')</pre>
(Simple) Spoofing	Y	<pre>sock.send(b' \x18\xda\x00\xf1\x03\x02\x27\x05')</pre>



# CAN Attack Methods (below J1939)

	ALL e.g. Socket CAN	CAN Hack	CANT	CANHack by Dr. Ken Tindell @ CANIS CANT by b1tbane & ehntoo @ GRIMM --- Notes:
Bus Flood	Y			
(Simple) Spoofing	Y			← the means for nearly all the attacks discussed next
Bus-Off / Bus Killer		Y	Y	target an ECU and destroy its frames repeatedly with selective bit override Cho et. Al & Maggi, F. c.f. also ICS-ALERT-17-209-01
(ASAP) Spoofing		Y		takes advantage of bitbanging to ensure attack frame is entered into arbitration ASAP after the target frame
Double Receive		Y		make a transmitter double-send a frame, error is only visible to transmitter and every other node receives same frame twice
Freeze / Overload		Y	Y	send a number of overload frames after a target frame
Error Passive Spoofing / Data Replacer		Y	Y	put a target into error passive mode then put an attack frame in front of a target frame aka "bit smashing"
Janus [Tindell]		Y		create a custom bitstream for two sampling point values so that receivers configured to those sampling points will receive different frames
Bus Short			Y	(cyber paperclip mode): shorts CAN_H+CAN_L (requires analog switch)
NACK			Y	clobber ACK bit by asserting a recessive state on the bus (requires analog switch)
(Improved) Data Replacer			Y	" but can clobber also dominant bits (requires analog switch)

# CAN Attack Methods (below J1939)

	ALL e.g. Socket CAN	CAN Hack	CANT	CANHack by Dr. Ken Tindell @ CANIS CANT by b1tbane & ehntoo @ GRIMM --- Notes:
Bus Flood	Y			
(Simple) Spoofing	Y			← the means for nearly all the attacks discussed next
Bus-Off / Bus Kill		Y	Y	target an ECU and destroy its frames repeatedly with selective bit override Cho et. Al & Maggi, F. c.f. also ICS-ALERT-17-209-01
(ASAP) Spoofing		Y		takes advantage of bitbanging to ensure attack frame is entered into arbitration ASAP after the target frame
Double Receive		Y		make a transmitter double-send a frame, error is only visible to transmitter and every other node receives same frame twice
Freeze / Overload		Y	Y	send a number of overload frames after a target frame
Error Passive Spoofing / Data Replacer		Y	Y	put a target into error passive mode then put an attack frame in front of a target frame aka "bit smashing"
Janus [Tindell]		Y		create a custom bitstream for two sampling point values so that receivers configured to those sampling points will receive different frames
Bus Short			Y	(cyber paperclip mode): shorts CAN_H+CAN_L (requires analog switch)
NACK			Y	clobber ACK bit by asserting a recessive state on the bus (requires analog switch)
(Improved) Data Replacer			Y	" but can clobber also dominant bits (requires analog switch)

👍 J1708

👍 J1708

👍 J1708

👍 J1708

👍 J1708

# What can you do on Heavy Vehicle Networks (1/11)?

- ▶ Some examples that have been made public
- ▶ Each result is true only on specific model year builds of trucks

# What can you do (2/11)?

## J1939: Vehicle Disable / Limp by DEF Additive Message Manipulation

- ▶ [Jonson, Urban. *A comprehensive review of cyber security for heavy vehicles for the NMFTA membership.* (2015)  
[http://www.nmfta.org/documents/ctsrp/nmfta\\_heavy\\_duty\\_vehicle\\_cyber\\_security\\_whitepaper\\_v1.0.3.6.pdf](http://www.nmfta.org/documents/ctsrp/nmfta_heavy_duty_vehicle_cyber_security_whitepaper_v1.0.3.6.pdf)]

# What can you do (3/11)?

## J1939: Denial of ECUs

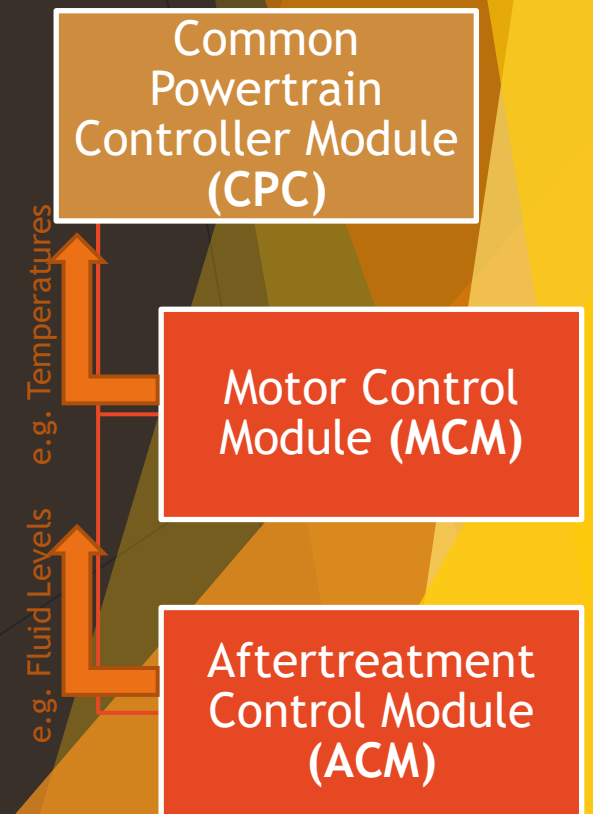
- ▶ [Mukherjee, Subhojeet & Shirazi, Hossein & Ray, Indrakshi & Daily, Jeremy & Gamble, Rose. (2016). *Truckers Beware: Practical Denial-of-service Attacks in Embedded Networks of Commercial Vehicles*.  
<http://www.cs.colostate.edu/dbsec/HeavyVehicle/wp-content/uploads/9783319498058-c2.pdf>]

# What can you do (4/11)?

## J1939: Vehicle Disable / Limp by Engine De-Rate Message Manipulation

### Research by CanBusHack

- ▶ [Jonson, Urban. *IOActive Guest Blog | Urban Jonson, Heavy Vehicle Cyber Security Program, NMFTA*. (2020). <https://ioactive.com/guest-urban-johnson-nmfta/>]
- ▶ More details available at [ctsrp.nmfta.org](https://ctsrp.nmfta.org)



# What can you do (5/11)?

## J1708/J1587: Malicious

## Misconfiguration of a Truck ECM

- ▶ [Haystack and sixvolts. *Cheap Tools for Hacking Heavy Trucks*. DEF CON 24 Car Hacking Village.  
<https://github.com/TruckHacking/DEFCON24/raw/master/Cheap-Tools-For-Hacking-Heavy-Trucks-Slides.pdf>]

# What can you do (6/11)?

## J1939: Instrument Cluster Override

- ▶ [Yelizaveta Burakova, Bill Hass, Leif Millar, and André Weimerskirch, *Truck Hacking: An Experimental Analysis of the SAE J1939 Standard*, Usenix WOOT, August 11-12, 2016, Austin, TX, USA.  
[http://www.weimerskirch.org/files/BurakovaEtAl\\_TruckHacking.pdf](http://www.weimerskirch.org/files/BurakovaEtAl_TruckHacking.pdf)]

# What can you do (7/11)?

## J1939: RPM control and Engine Brake Disable

- ▶ [Yelizaveta Burakova, Bill Hass, Leif Millar, and André Weimerskirch, *Truck Hacking: An Experimental Analysis of the SAE J1939 Standard*, Usenix WOOT, August 11-12, 2016, Austin, TX, USA.  
[http://www.weimerskirch.org/files/BurakovaEtAl\\_TruckHacking.pdf](http://www.weimerskirch.org/files/BurakovaEtAl_TruckHacking.pdf)]

# What can you do (8/11)?

## J1708/J1587: Disable Engine Cylinders

- ▶ [Yelizaveta Burakova, Bill Hass, Leif Millar, and André Weimerskirch, *Truck Hacking: An Experimental Analysis of the SAE J1939 Standard*, Usenix WOOT, August 11-12, 2016, Austin, TX, USA.  
[http://www.weimerskirch.org/files/BurakovaEtAl\\_TruckHacking.pdf](http://www.weimerskirch.org/files/BurakovaEtAl_TruckHacking.pdf)]

# What can you do (9/11)?

J1708/J1587 / J2497: Cycle ABS

Air Release Valves

- ▶ [Yelizaveta Burakova, Bill Hass, Leif Millar, and André Weimerskirch, *Truck Hacking: An Experimental Analysis of the SAE J1939 Standard*, Usenix WOOT, August 11-12, 2016, Austin, TX, USA.  
[http://www.weimerskirch.org/files/BurakovaEtAl\\_TruckHacking.pdf](http://www.weimerskirch.org/files/BurakovaEtAl_TruckHacking.pdf)]

# What can you do (10/11)?

## J2497: Remote Read and Write of Traffic

- ▶ [Poore, Chris & Gardiner, Ben. *Power Line Truck Hacking: 2TOOLS4PLC4TRUCKS*. DEF CON 28 SAFE MODE Car Hacking Village. <https://youtu.be/sf3JznYTo0I>]
- ▶ [Poore, Chris & Gardiner, Ben. *Trailer Shouting Talking PLC4TRUCKS Remotely with an SDR* DEF CON 30. [https://youtu.be/Na8K\\_fVEzQo](https://youtu.be/Na8K_fVEzQo)]

# What can you do (11/11)?

## Summary

Network	'Hacking'	Who
J1939	Vehicle disable/limp by DEF message manipulation	Jonson (NMFTA)
J1939	Denial of ECUs	Mukherjee et. al.
J1939	Vehicle disable/limp by de-rate message manipulation	Leale (CanBusHack)
J1708/J1587	Malicious misconfiguration of truck ECM	Haystack et. al.
J1939	Instrument Cluster override	Burakova et. al.
J1939	RPM control and engine brake disable	Burakova et. al.
J1708/J1587	Disable engine cylinders	Burakova et. al.
J1708/J1587 / J2497	Cycle ABS air release valves	Burakova et. al.
J2497	Remote read and write of traffic	Poore (AIS) et. al.

# What needs more 'Hacking'?

- ▶ Abuse of J1939 and/or J2497 features:
  - ▶ ECU Firmware Dumping and Re-flashing
  - ▶ Seed-Key Exchange
  - ▶ Interesting Body Control Functions
  - ▶ ADAS features
- ▶ Vehicle Network Gateways
  - ▶ Sometimes being introduced for security purposes
  - ▶ Always security relevant (pivoting)

# Other Truck & Trailer Hacking

- ▶ Like car hacking, this is also the “*Olympics of Hacking*”  
-- Will Caruanna
- ▶ All the usual IoT mobile, game hacking and RF stuff applies
  - ▶ Telematics (usually IoT)
  - ▶ Mobile (usually Android)
  - ▶ Diagnostics & Maintenance Tools (usually Windows)
  - ▶ RF (usually ͇\\_(\ツ)\\_/͇)

# Get Involved

# Get Involved: CyberTruck Challenge™

An event specifically to train students. Attended by industry and cybersecurity experts. >10K in free training per-person. Several Trucks present each year. Stipends available for students.

[www.cybertruckchallenge.org  
/participate/](http://www.cybertruckchallenge.org/participate/)



# Get Involved: Bench Setups

## ▶ Necessary

- ▶ If you are hacking on a truck then it isn't making the fleet any money

## ▶ You can build a Truck in a Box:

- ▶ see Haystack and sixvolts. [Cheap Tools for Hacking Heavy Trucks.](#)
- ▶ For the really gory details see Córcega, Jose L. DESIGN OF A FORENSICALLY NEUTRAL ELECTRONIC ENVIRONMENT FOR HEAVY VEHICLE EVENT DATA RECORDERS. Master's Thesis, University of Tulsa. 2015.



# Tools for Truck Hacking / Vehicle Networks

# Example: decoding J1939 with pretty\_j1939.py

[https://github.com/nmfta-repo/pretty\\_j1939](https://github.com/nmfta-repo/pretty_j1939)

```
# python pretty_j1939.py --candata --format --no-link --da-json=J1939db.json sample_candump.txt
(1604007053.845809) can1 1CECFF00#20220005FFE3FE00 ;
[...]
(1604007056.355739) can1 1CECFF00#20120003FFECFE00 ;
(1604007056.455709) can1 1CEBFF00#025834434A323933 ;
(1604007056.505740) can1 1CEBFF00#033134322AFFFFFF ; {
;         "PGN": "VI(65260)",
;         "DA": "All(255)",
;         "SA": "Engine #1( 0)",
;         "Vehicle Identification Number": "1XKAAP8X4CJ293142",
;         "Transport Data": "0x31584b414150385834434a3239333134322a"
; }
#
```

CAN Data from public logs at  
[engr.colostate.edu/~jdaily/J1939/candata.html](http://engr.colostate.edu/~jdaily/J1939/candata.html)

# Example: sending J1939 with SocketCAN on a TruckDuck (1/3)

- ▶ The TruckDuck <https://github.com/TruckHacking/TruckDuckHardware> by [sixvolt](#) & [haystack](#), @DEF CON 24. Later revisions 1.5 YEET and MEGA.



- ▶ Also remixed as the TruckCape by Dr. Daily: <https://oshpark.com/profiles/jeremy-daily> 

# Example: sending J1939 (2/3)

```
import socket
import struct

sock = socket.socket(socket.PF_CAN, socket.SOCK_RAW, socket.CAN_RAW)
sock.bind(("can0",))

can_id = 0x18EEFF00 # broadcast address claims
can_id |= socket.CAN_EFF_FLAG # Set the extended frame format bit

can_data = bytes.fromhex('06 03 BF 01 00 00 00 10') # ex address claim msg

can_dlc = min(len(can_data), 8)
can_packet = struct.pack("<1B3x8s", can_id, can_dlc, can_data[:can_dlc])
sock.send(can_packet)
```

From Dr. Daily's [github.com/SystemsCyber/TruckCapeProjects/](https://github.com/SystemsCyber/TruckCapeProjects/)  
Presentation on this topic available at [ctsrp.nmfta.org](https://ctsrp.nmfta.org)

# Example: sending J1939 Alternatives (3/3)

There's a whole bunch of other ways to send J1939 with Python

- ▶ Using Linux (since 5.4) `.CAN_J1939` e.g. via `cpython` (since 3.9)
  - ▶ Also how haystack's <https://github.com/TruckHacking/py-hv-networks> works (with backport)
- ▶ <https://github.com/juergenH87/python-can-j1939> is actively developed
  - ▶ Uses also `.CAN_RAW` on various python-can drivers
  - ▶ The API here is best suited to developing J1939 ECUs
- ▶ And CanCat and TruckDevil
  - ▶ No SocketCAN option

# Example: decoding J1708/J1587/J2497

with pretty\_j1587.py ► [github.com/ainfosec/pretty\\_j1587](https://github.com/ainfosec/pretty_j1587)

► Developed by Dan Salloum @ AIS

```
# echo ac#ff0189 | ./pretty_j1587.py -f -  
MSG: [0xac,0xff,0x1,0x89]  
MID 0xac (172): Off-board Diagnostics #1  
PID 0x101 (257): Cold Restart of Specific Component  
DATA: 0x89
```

```
# echo ac#f31089 | ./pretty_j1587.py -f -  
MSG: [0xac,0xf3,0x10,0x89]  
MID 0xac (172): Off-board Diagnostics #1  
PID 0xf3 (243): Component Identification  
DATA: 0x10, 0x89
```

J1587 Data from DEF CON 28 CHV CTF attempts by uhlox (Aug 2020)

# Example: sending J1708/J1587/J2497 with j1708send.py on a TruckDuck

- ▶ <https://github.com/TruckHacking/plc4trucksduck>

```
# j1708send.py --interface=plc 0a00
```

```
#
```

For more ways to send J2497 see: [Power Line Truck Hacking: 2TOOLS4PLC4TRUCKS](#)

# More Tools for Truck Hacking Vehicle Networks (1/8)

- ▶ e.g. TruckDevil, CanCat, canmatrix, gr-j2497, py-hv-networks, Nexiq USBLink, DGTech DPA4.

# TruckDevil (2/8)

- ▶ <https://github.com/LittleBlondeDevil/TruckDevil>
- ▶ By Hannah Silva
- ▶ For reading, decoding, logging and sending messages on **J1939**.
- ▶ Works with
  - ▶ the (very affordable) Macchina M2 via custom FW
  - ▶ and any SocketCAN device (via python-can)
- ▶ Being grown into an attack framework, starting with a fuzzer
- ▶ Training videos available at [ctsrp.nmfta.org](https://ctsrp.nmfta.org)

## CanCat (3/8)

- ▶ <https://github.com/atlasofd00m/CanCat>
- ▶ By atlas and also GRIMM CyPhy team
- ▶ Works with the M2 via custom FW
- ▶ includes **J1939** support

## canmatrix (4/8)

- ▶ <https://github.com/ebroecker/canmatrix>
- ▶ by Eduard Bröcker
- ▶ Converts between many CAN signal definitions, including support for **J1939**

## gr-j2497 (5/8)

- ▶ <https://github.com/ainfosec/gr-j2497>
- ▶ Developed by Chris Poore @ AIS
- ▶ MIT licensed
- ▶ Flow graphs with a custom block for reading **J2497/PLC4TRUCKS** traffic

## py-hv-networks (6/8)

- ▶ [github.com/TruckHacking/py-hv-networks/blob/master/hv\\_networks/J1587Driver.py](https://github.com/TruckHacking/py-hv-networks/blob/master/hv_networks/J1587Driver.py)
- ▶ Developed by haystack and sixvolts
- ▶ The core of the TruckDuck features
- ▶ A set of python libraries for send+receive of **J1939** and J1708/J1587 traffic

## Nexiq USBLink (7/8)

- ▶ A RP1210 Vehicle Diagnostic Adapter (VDA)
- ▶ One RP1210 VDA is necessary to use any OEM/supplier diagnostics packages
- ▶ This adapter is cheap and easy to find
- ▶ Has a DB15 connector for which there are many cables available for purchase as well

## DG Tech DPA4 (8/8)

- ▶ Another RP1210 compatible adapter
- ▶ Also has the most-useful DB-15 connector
- ▶ Drivers include a very useful data logging feature (called 'debug file' in the settings)

**CLOSING**

# Review

- ▶ Commercial Transportation is important to us (all)
- ▶ Trucks have 3 main types of vehicle networks:  
J1939, J1708/J1587, and J2497
  - ▶ Two (J1939 and J2497) are on all trucks in North America
  - ▶ Some are **shared** with **other modes** of transportation
- ▶ Talented/helpful people have published ~10 ‘**attacks**’ on vehicle networks
- ▶ There are plenty of opportunities for **more** (abuse aka ‘logic bugs’ in particular)
- ▶ There are a host of **free tools** for interacting with vehicle networks
- ▶ There are many ways for **you to get involved** !

# Collaboration / about the NMFTA CTSRP

The **NMFTA** Commercial Transportation Security and Research Program (**CTSRP**) funds and collaborates on a wide array of topics affecting commercial transport, e.g.

- ▶ vehicle security offense & defense (topics of this talk)
- ▶ backend systems security
- ▶ distribution and service center security
- ▶ mainframe & 'mid' (IBM z & IBM i series) security

If you are interested in collaborating on a particular project idea, reach out to us please: <https://ctsrp.nmfta.org/>

Thank You



# References and Resources

# References: Public Papers and Presentations for Truck Hacking

- ▶ Daily, Jeremy & Kongs, Andrew & Johnson, James & Corcega, Jose. (2015). *Extracting Event Data from Memory Chips within a Detroit Diesel DDEC v. 2015*. 10.4271/2015-01-1450. <https://synercontechologies.com/wp-content/uploads/2016/05/2015-SAE-Presentation-On-Chip-Level-Forensics.pdf>
- ▶ Mukherjee, Subhojeet & Shirazi, Hossein & Ray, Indrakshi & Daily, Jeremy & Gamble, Rose. (2016). *Truckers Beware: Practical Denial-of-service Attacks in Embedded Networks of Commercial Vehicles*. <http://www.cs.colostate.edu/dbsec/HeavyVehicle/wp-content/uploads/9783319498058-c2.pdf>
- ▶ Jonson, Urban. *A comprehensive review of cyber security for heavy vehicles for the NMFTA membership*. <http://www.nmfta.org/documents/hvcs/nmfta%20heavy%20duty%20vehicle%20cyber%20security%20whitepaper%20v1.0.3.6.pdf?v=1>
- ▶ Haystack and sixvolts. *Cheap Tools for Hacking Heavy Trucks*. DEF CON 24 Car Hacking Village. <https://github.com/TruckHacking/DEFCON24/raw/master/Cheap-Tools-For-Hacking-Heavy-Trucks-Slides.pdf>
- ▶ Yelizaveta Burakova, Bill Hass, Leif Millar, and André Weimerskirch, *Truck Hacking: An Experimental Analysis of the SAE J1939 Standard*, Usenix WOOT, August 11-12, 2016, Austin, TX, USA. [http://www.weimerskirch.org/files/BurakovaEtAl\\_TruckHacking.pdf](http://www.weimerskirch.org/files/BurakovaEtAl_TruckHacking.pdf)
- ▶ Córcega, Jose L. DESIGN OF A FORENSICALLY NEUTRAL ELECTRONIC ENVIRONMENT FOR HEAVY VEHICLE EVENT DATA RECORDERS. Master's Thesis, University of Tulsa. 2015.
- ▶ Jonson, Urban. *IOActive Guest Blog | Urban Jonson, Heavy Vehicle Cyber Security Program, NMFTA*. <https://ioactive.com/guest-urban-johnson-nmfta/>
- ▶ Thuen, Corey. *IOActive: Heavy Trucks and Electronic Logging Devices: What Could Go Wrong?* <https://act-on.ioactive.com/acton/attachment/34793/f-d8737079-c6eb-411c-94d1-f52ffc9df975/1/-/-/-/IOActive-ELoggingDeviceVulnerabilities.pdf>
- ▶ Salloum, Dan & Hayes, Thomas. *Before J1939: A J1708/J1587 Protocol Decoder*. DEF CON 28 SAFE MODE Car Hacking Village. <https://www.youtube.com/watch?v=hal-E4mProk>
- ▶ Poore, Chris & Gardiner Ben. *Power Line Truck Hacking: 2TOOLS4PLC4TRUCKS*. DEF CON 28 SAFE MODE Car Hacking Village. <https://www.youtube.com/watch?v=sf3JznYTo0I>

# References: Public Papers and Presentations for Vehicle Hacking

- ▶ Cho et al. *Error Handling of In-vehicle Networks Makes Them Vulnerable* (2016) [https://tagi98.github.io/files/publication/ktcho\\_busoff.pdf](https://tagi98.github.io/files/publication/ktcho_busoff.pdf)
- ▶ Maggi, Federico. *A Vulnerability in Modern Automotive Standards and How We Exploited It.* (2017) <https://documents.trendmicro.com/assets/A-Vulnerability-in-Modern-Automotive-Standards-and-How-We-Exploited-It.pdf>
- ▶ Brom, Tim and Johnson, Mitchell. *When CAN CANT* (2018) [https://www.youtube.com/watch?v=TRn\\_Rz2JIYQ](https://www.youtube.com/watch?v=TRn_Rz2JIYQ)
- ▶ Tindell, Ken. *CAN BUS: ATTACKS AND MITIGATIONS* (2020) <https://asrg.io/events/15/>
- ▶ Gardiner, Ben. *REMOTE WRITING TRAILER AIR BRAKES WITH RF* (2022) <https://act-on.ioactive.com/acton/attachment/34793/f-ab6fea27-1929-4ad1-8cd6-7d2094910f66/1/-/-/-/-/Trailer-Air-Brakes-RF-NMFTA.pdf>

# Links to More Resources for Truck Hacking

- ▶ CyberTruck Challenge™  
<https://www.cybertruckchallenge.org/participate/truck-challenges/>
- ▶ NMFTA CTSRP unrestricted <http://www.nmfta.org/pages/hvcs>
- ▶ NMFTA CTSRP access-controlled but still free:  
<https://hvcslistserv.nmfta.org/>
- ▶ NMFTA github org <https://github.com/nmfta-repo>
- ▶ DG Tech reference on heavy vehicle pinouts:  
[https://www.dgtech.com/wp-content/uploads/2016/04/Pinouts\\_ICR.pdf](https://www.dgtech.com/wp-content/uploads/2016/04/Pinouts_ICR.pdf)
- ▶ Vector References on Vehicle Networks  
<https://elearning.vector.com/?lang=en>
- ▶ Truck Hacking github org <https://github.com/TruckHacking>
- ▶ UTulsa github org <https://github.com/orgs/Heavy-Vehicle-Networking-At-U-Tulsa/teams>
- ▶ Colorado State github org <https://github.com/SystemsCyber>
- ▶ Colorado State Heavy Vehicle CAN data  
<https://www.engr.colostate.edu/~jdaily/J1939/candata.html>
- ▶ SAE J1939:  
[https://www.sae.org/publications/collections/content/j1939\\_dl/](https://www.sae.org/publications/collections/content/j1939_dl/)
- ▶ SAE J1939-DA:  
[https://www.sae.org/standards/content/j1939da\\_202012/](https://www.sae.org/standards/content/j1939da_202012/)
- ▶ SAE J1939-21:  
[https://www.sae.org/standards/content/j1939/21\\_201810/](https://www.sae.org/standards/content/j1939/21_201810/)
- ▶ SAE J1939-81:  
[https://www.sae.org/standards/content/j1939/81\\_201703/](https://www.sae.org/standards/content/j1939/81_201703/)
- ▶ SAE J1708:  
[https://www.sae.org/standards/content/j1708\\_200408/](https://www.sae.org/standards/content/j1708_200408/)
- ▶ SAE J1587:  
[https://www.sae.org/standards/content/j1587\\_201301/](https://www.sae.org/standards/content/j1587_201301/)
- ▶ SAE J2497:  
[https://www.sae.org/standards/content/j2497\\_201207/](https://www.sae.org/standards/content/j2497_201207/)