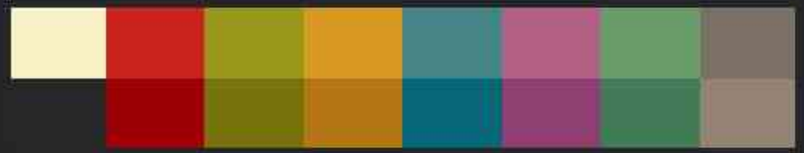
Automotive Security

WARCON.PL 2024

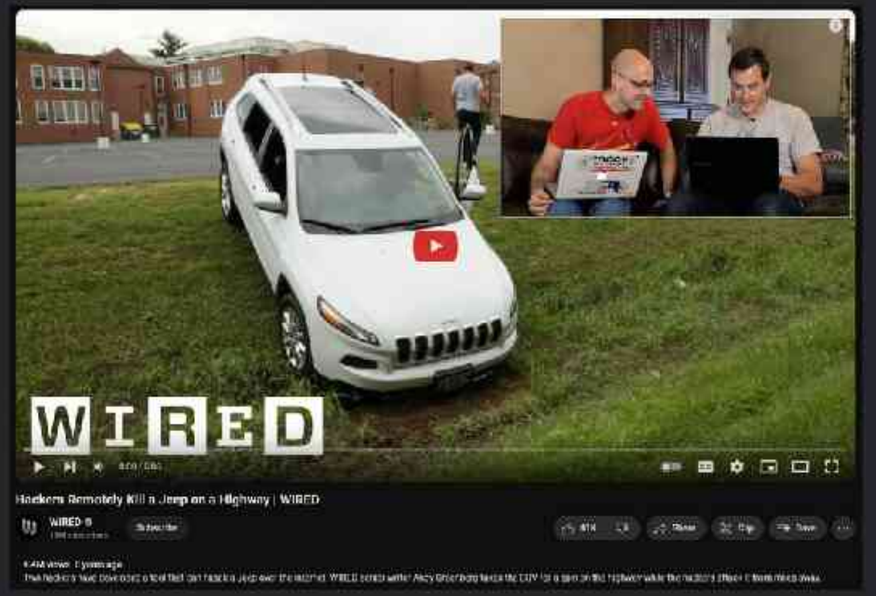
[~] whoami

djnn@penthertz.com

Occupation: intern @ penthertz u already know =)
Location: Paris
Favorite drink: Blond beer
Interests: Reverse-engineering, malware
Languages: C, Golang, Elixir, etc (learning Rust)
Contact: https://djnn.sh/pgp



[~] man voiture



ANDY GREENBERG SECURITY JUL 21, 2015 6:00 AM

Hackers Remotely Kill a Jeep on the Highway—With Me in It

I was driving 70 mph on the edge of downtown St. Louis when the exploit began to take hold.

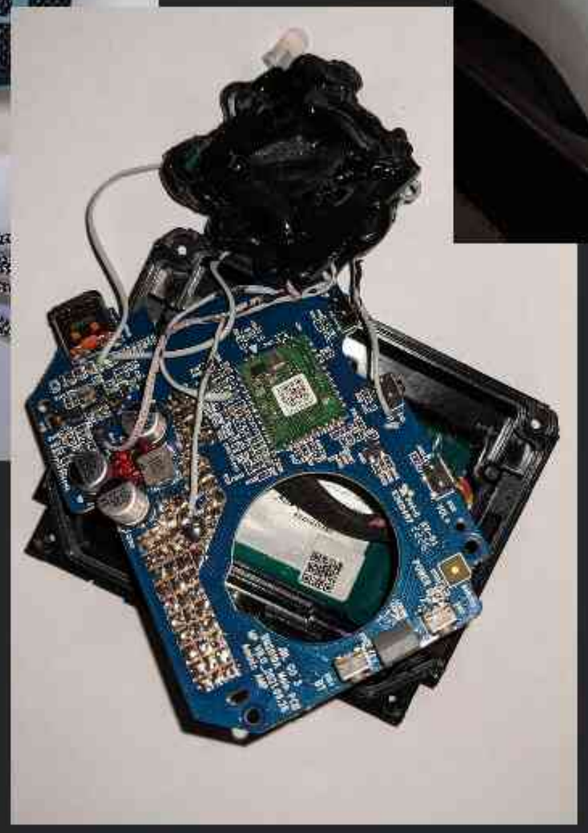
https://illmatics.com/Remote%20Car%20Hacking.pdf

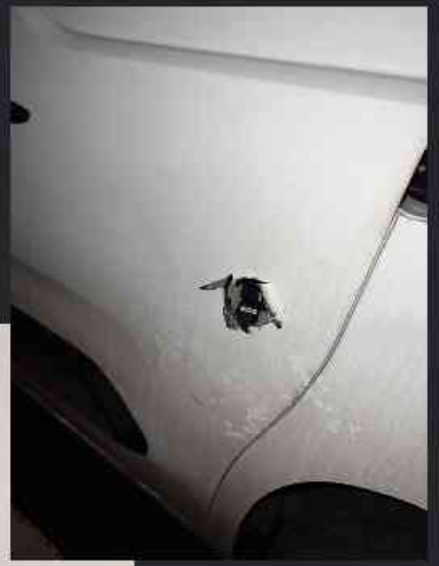
Gone in 61 seconds.

The keys were left near the front door.

7:54 PM · Oct 2, 2023 · **7.2M** Views

[~] man vroom-vroom





https://kentindell.github.io/2023/04/03/can-injection/

[~] vim vroom.txt

README Apache-2.0 license

[initial] 頭文字V

Initial V

Initial V is a BMW shifter that has been converted to a Bluetooth keyboard. In this repository, you'll find schematics and PCB designs, stl files, a Vim plugin, and client software for turning a BMW shifter in to a Bluetooth keyboard that can control Vim.

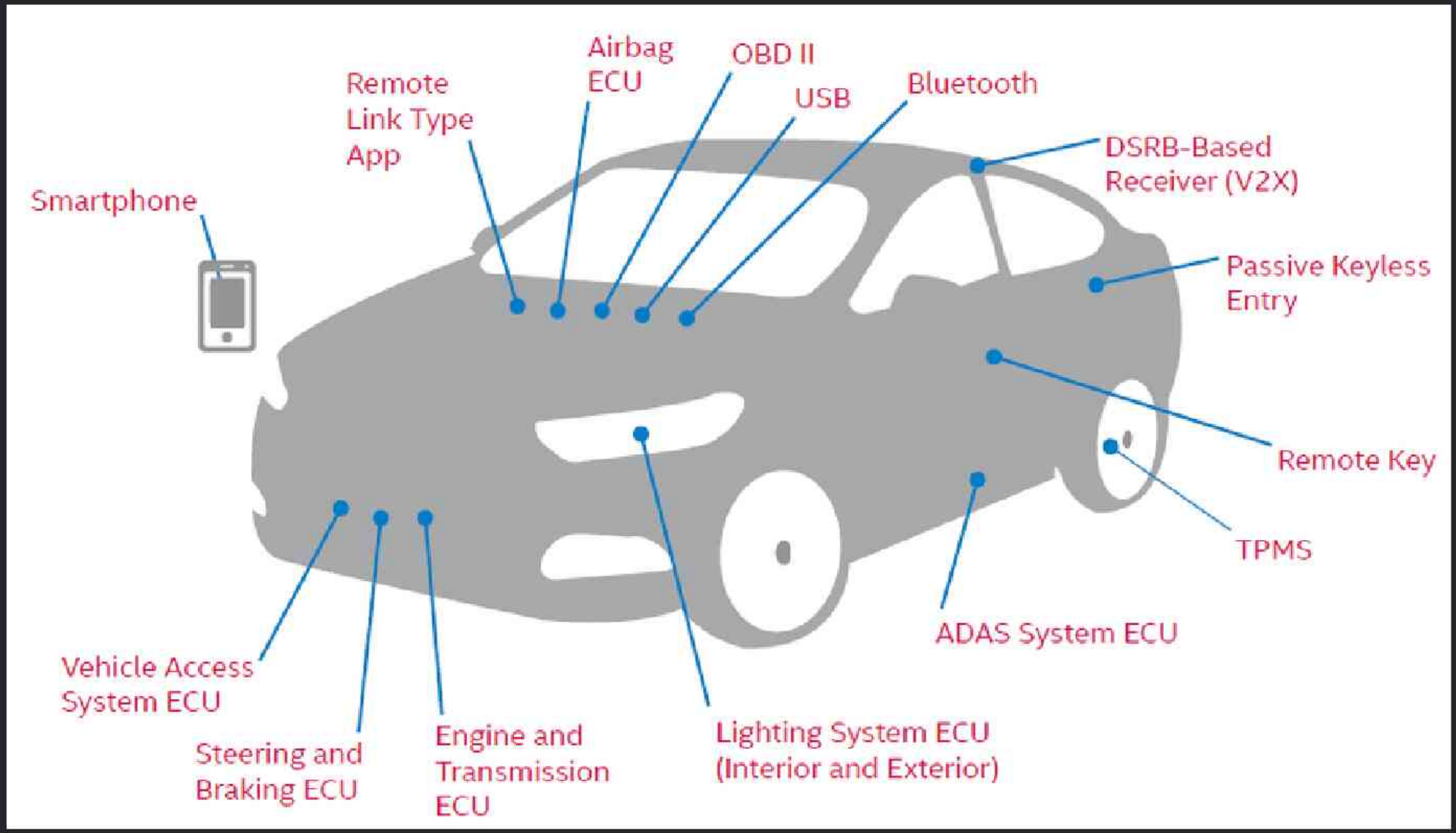
Think of this project as a very over engineered Vim clutch

Initial V is a Bluetooth Keyboard specialized for controlling Vim. The key presses sent depend on Vim's state. The table below describes the key presses for each handle position according to the state of the editor:

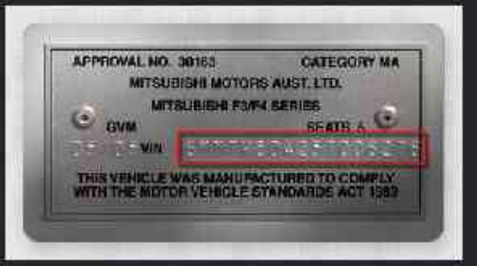
	Park	Up	Down	Double Up	Double Down	Move Left	Left Up	Left Down	Move Right (back to center)
Normal Mode (Drive)	`:w` on a modified buffer, `:wq` on unmodified buffer	Up key	Down key	`i`	`o`	`CTRL -V`	Up Key	Down key	`ESC`
Insert Mode (Neutral)	`ESC`	Up key	Down key	Page Up	Page Down				

"Drive" on the handle means "Normal Mode" in Vim. "Neutral" on the handle means "Insert Mode" in Vim. It's not possible to move the handle to the left when the handle is in Neutral mode, so there are no key combinations. I'm not sure what mode in Vim would map to Reverse on the handle, so there's no way to transition to Reverse at the moment.

Saving a buffer in Normal mode will put the handle in to the "Park" position. The Park position behaves the same way as Drive (Normal mode in Vim) except that if you hit Park again, it will exit Vim.


Smartphone
Remote Link Type App
Airbag ECU
OBD II
USB
Bluetooth
DSRB-Based Receiver (V2X)
Passive Keyless Entry
Remote Key
TPMS
ADAS System ECU
Vehicle Access System ECU
Steering and Braking ECU
Engine and Transmission ECU
Lighting System ECU (Interior and Exterior)

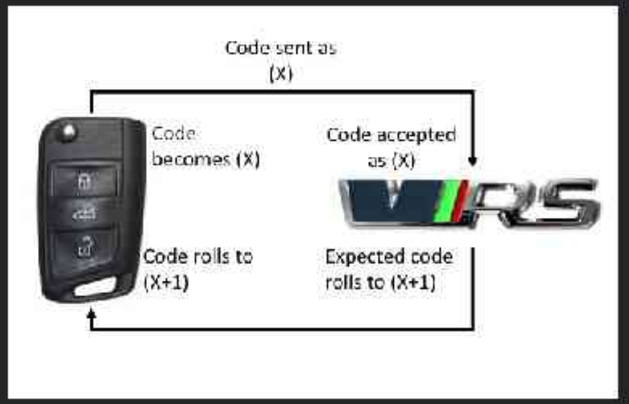
[~] apktool d deez_nuts.jar

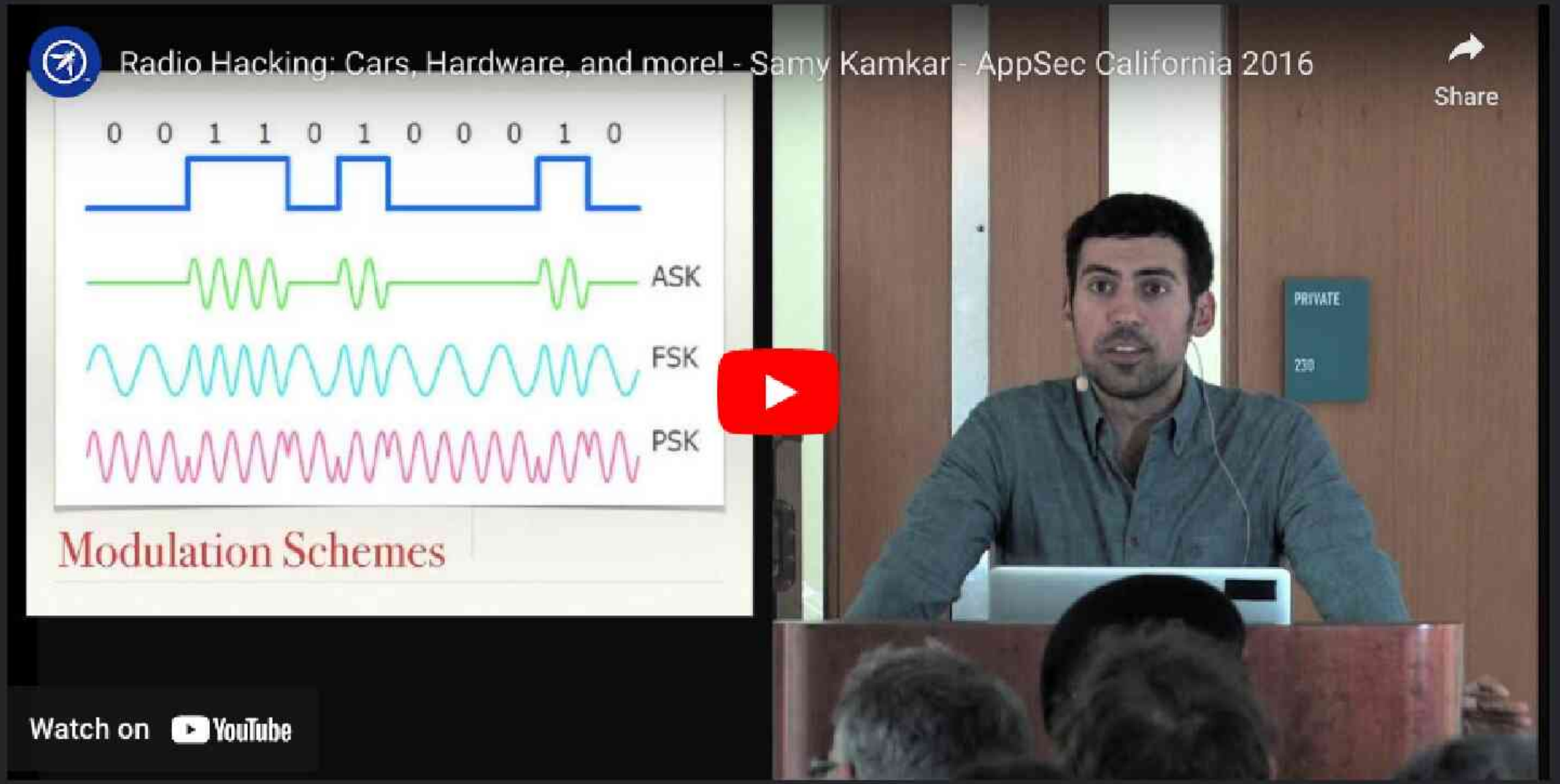


- remote startup
- open doors
- localisation
- ...

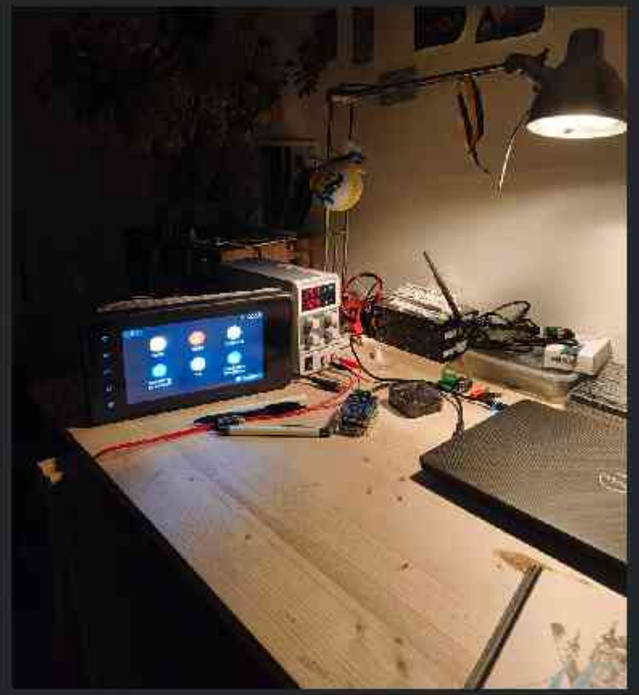
[~] tpms_rx --source rtlsdr

TPMS Frequencies:
300Mhz > f > 900Mhz




Radio Hacking: Cars, Hardware, and more! - Samy Kamkar - AppSec California 2016
Share
0 0 1 1 0 1 0 0 0 1 0
ASK
FSK
PSK
Modulation Schemes
PRIVATE
230
Watch on YouTube

[~] which IVI

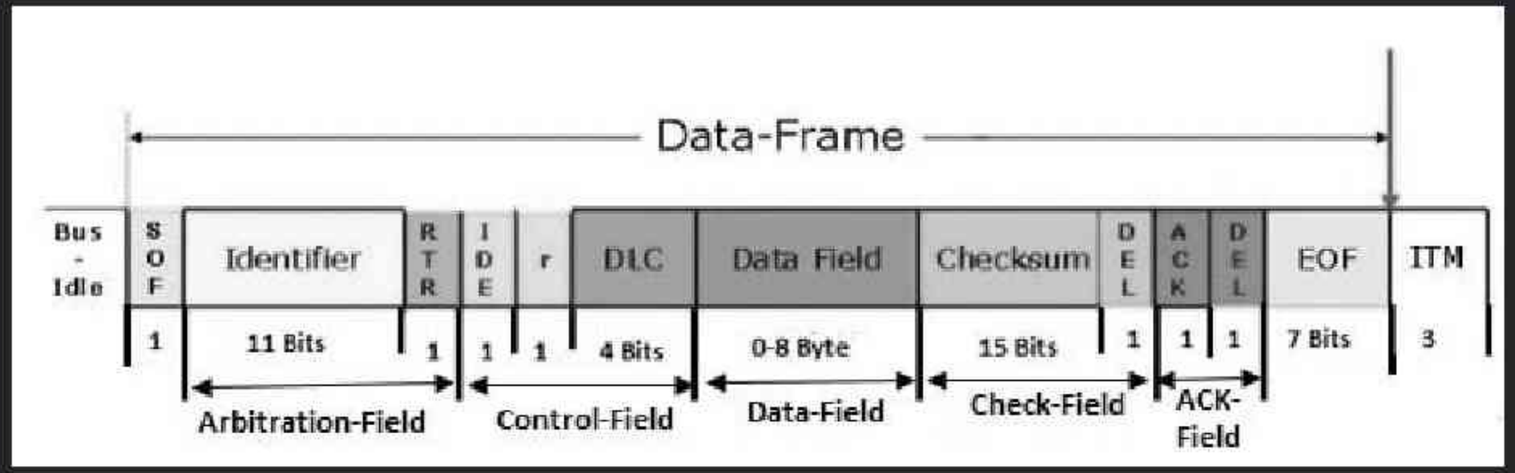


WiFi, Bluetooth, CAN, ...

https://hydrabus.com

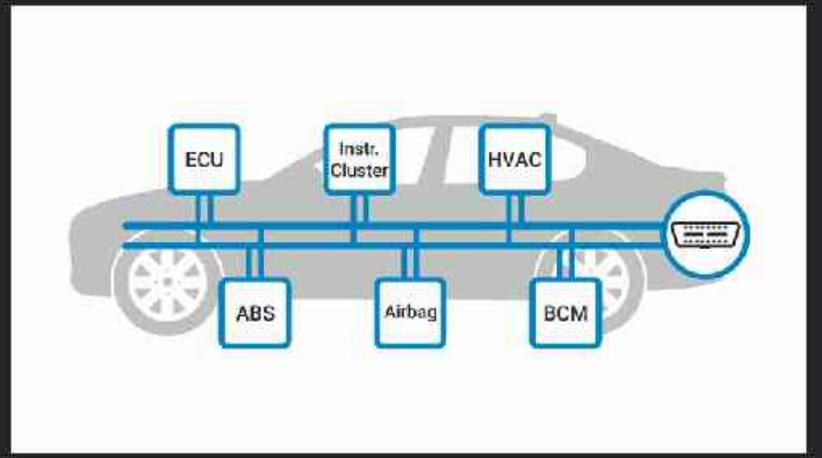
https://pinoutguide.com/Car-Stereo-Other/

```
[~] sudo modprobe vcan
```



Controller Area Network (CAN)

--> 1983 @ Bosch



[~] python3 trolling.py

```python
#!/bin/env python3

# pip install python-can
import can

bus = can.Bus()
while True:
  msg = can.Message(3, data=[0 for _ in range(8)])
  bus.send(msg)
```

[~] pip install socketcan-uds

Functional Unit	SID	Available in Default Session	Available for RoE	Has Sub-Function	Service Name	Mnemonic
Diagnostic and Communication Management	$10	✓		✓	Diagnostic Session Control	DSC
	$11	✓		✓	ECU Reset	ER
	$27			✓	Security Access	SA
	$28			✓	Communication Control	CC
	$3E	✓		✓	Tester Present	TP
	$83			✓	Access Timing Parameter	ATP
	$84				Secured Data Transmission	SDT
	$85			✓	Control DTC Setting	CDTCS
	$86	✓		✓	Response On Event	ROE
	$87			✓	Link Control	LC
Data Transmission	$22	✓			Read Data By Identifier	RDBI
	$23	✓			Read Memory By Address	RMBA
	$24	✓			Read Scaling Data By Identifier	RSDBI
	$2A		✓		Read Data By Periodic Identifier	RDBPI
	$2C	✓		✓	Dynamically Define Data Identifier	DDDI
	$2E	✓			Write Data By Identifier	WDBI
	$3D	✓			Write Memory By Address	WMBA
Stored Data Transmission	$14	✓			Clear Diagnostic Information	CDTCI
	$19	✓	✓	✓	Read DTC Information	RDTCI
Input Output Control	$2F		✓		Input Output Control By Identifier	IOCBI
Remote Activation of Routine	$31	✓	✓	✓	Routine Control	RC
Upload Download	$34				Request Download	RD
	$35				Request Upload	RU
	$36				Transfer Data	TD
	$37				Request Transfer Exit	RTE

Example:

UDS service identifiers (SIDs)

	UDS SID (request)	UDS SID (response)	Service	Details
Diagnostic and Communications Management	0x10	0x50	Diagnostic Session Control	Control which UDS services are available
	0x11	0x51	ECU Reset	Reset the ECU ("hard reset", "key off", "soft reset")
	0x27	0x67	Security Access	Enable use of security-critical services via authentication
	0x28	0x68	Communication Control	Turn sending/receiving of messages on/off in the ECU
	0x29	0x69	Authentication	Enable more advanced authentication vs. 0x27 (PKI based exchange)
	0x3E	0x7E	Tester Present	Send a "heartbeat" periodically to remain in the current session
	0x83	0xC3	Access Timing Parameters	View/modify timing parameters used in client/server communication
	0x84	0xC4	Secured Data Transmission	Send encrypted data via ISO 15764 (Extended Data Link Security)
	0x85	0xC5	Control DTC Settings	Enable/disable detection of errors (e.g. used during diagnostics)
	0x86	0xC6	Response On Event	Request that an ECU processes a service request if an event happens
	0x87	0xC7	Link Control	Set the baud rate for diagnostic access
Data Transmission	0x22	0x62	Read Data By Identifier	Read data from targeted ECU - e.g. VIN, sensor data values etc.
	0x23	0x63	Read Memory By Address	Read data from physical memory (e.g. to understand software behavior)
	0x24	0x64	Read Scaling Data By Identifier	Read information about how to scale data identifiers
	0x2A	0x6A	Read Data By Identifier Periodic	Request ECU to broadcast sensor data at slow/medium/fast/stop rate
	0x2C	0x6C	Dynamically Define Data Identifier	Define data parameter for use in 0x22 or 0x2A dynamically
	0x2E	0x6E	Write Data By Identifier	Program specific variables determined by data parameters
	0x3D	0x7D	Write Memory By Address	Write information to the ECU's memory
DTCs	0x14	0x54	Clear Diagnostic Information	Delete stored DTCs
	0x19	0x59	Read DTC Information	Read stored DTCs, as well as related information
	0x2F	0x6F	Input Output Control By Identifier	Gain control over ECU analog/digital inputs/outputs
	0x31	0x71	Routine Control	Initiate/stop routines (e.g. self-testing, erasing of flash memory)
Upload/ Download	0x34	0x74	Request Download	Start request to add software/data to ECU (incl. location/size)
	0x35	0x75	Request Upload	Start request to read software/data from ECU (incl. location/size)
	0x36	0x76	Transfer Data	Perform actual transfer of data following use of 0x74/0x75
	0x37	0x77	Request Transfer Exit	Stop the transfer of data
	0x38	0x78	Request File Transfer	Perform a file download/upload to/from the ECU
		0x7F	Negative Response	Sent with a Negative Response Code when a request cannot be handled

[~] gcc uds-psa.c -o trolling

UDS Frame	D0	D1	D2	D3.....Dn(Optional)
Seed –Request (Tool→ECU)	27	xx(Seed_Sunfunc)	Application specific Data	
Seed –Response (Tool←ECU)	67	xx(Seed_Subfunc)	Seed_Value[n]	
Key-Response (Tool→ECU)	27	zz (key_Subfunc)	Key_Value[n]	
Response (If Key Verified) (Tool←ECU)	67	zz (key_Subfunc)	Application specific Data	

main ▾ psa-seedkey-algorithm / algorithm.c

Ludwig Copyright notice

Code | Blame | 44 lines (37 loc) · 1.67 KB | Code 55% faster with GitHub Copilot

```c
/*
Copyright 2024, Ludwig V. <https://github.com/ludwig-v>
original algorithm by Wouter Bokslag & Jasor R. <https://github.com/prototux>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License at <http://www.gnu.org/licenses/> for
more details.

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.
*/

#include <inttypes.h>

// Transformation function with PSA not-so-secret sauce
int16_t transform(uint8_t data_msb, uint8_t data_lsb, uint8_t sec[])
{
    int16_t data = (data_msb << 8) | data_lsb;
    int32_t result = ((data % sec[0]) * sec[2]) - ((data / sec[0]) * sec[1]);
    if (result < 0)
        result += (sec[0] * sec[2]) + sec[1];
    return result;
}

// Challenge reponse calculation for a given pin and challenge
// Challenge (seed) is 4 bytes and pin (key) is 2 bytes
uint32_t compute_response(uint8_t pin[], uint8_t chg[])
{
    // Still hardcoded secrets
    int8_t sec_1[3] = {0x82, 0xCF, 0xAA};
    int8_t sec_2[3] = {0x81, 0x82, 0xAB};

    // Compute each 16b part of the response, with the twist, and return it
    int16_t res_msb = transform(pin[0], pin[1], sec_1) | transform(chg[0], chg[3], sec_2);
    int16_t res_lsb = transform(chg[1], chg[2], sec_1) | transform(res_msb >> 8, res_msb & 0xFF, sec_2);
    return (res_msb << 16) | res_lsb;
}
```

[~] git clone git@github.com:commaai/panda.git

master opendbc / tesla_can.dbc

```
225     BO_ 792 GTW_carState: 8 GTW
226      SG_ YEAR : 0|7@1+ (1,2000) [2000|2127] "Year" NEO
227      SG_ CERRD : 7|1@1+ (1,0) [0|1] "" NEO
228      SG_ MONTH : 8|4@1+ (1,0) [1|12] "Month" NEO
229      SG_ DOOR_STATE_FL : 12|2@1+ (1,0) [0|3] "" NEO
230      SG_ DOOR_STATE_FR : 14|2@1+ (1,0) [0|3] "" NEO
231      SG_ SECOND : 16|6@1+ (1,0) [0|59] "s" NEO
232      SG_ DOOR_STATE_RL : 22|2@1+ (1,0) [0|3] "" NEO
233      SG_ Hour : 24|5@1+ (1,0) [0|23] "h" NEO
234      SG_ DOOR_STATE_RR : 29|2@1+ (1,0) [0|3] "" NEO
235      SG_ DAY : 32|5@1+ (1,0) [0|31] "" NEO
236      SG_ MINUTE : 40|6@1+ (1,0) [0|59] "min" NEO
237      SG_ BOOT_STATE : 46|2@1+ (1,0) [0|3] "" NEO
238      SG_ GTW_updateInProgress : 48|2@1+ (1,0) [0|3] "" NEO
239      SG_ DOOR_STATE_FrontTrunk : 50|2@1+ (1,0) [0|3] "" NEO
240      SG_ MCU_factoryMode : 52|1@1+ (1,0) [0|1] "" NEO
241      SG_ MCU_transportModeOn : 53|1@1+ (1,0) [0|1] "" NEO
242      SG_ BC_headLightLStatus : 55|2@1+ (1,0) [0|3] "" NEO
243      SG_ BC_headLightRStatus : 57|2@1+ (1,0) [0|3] "" NEO
244      SG_ BC_indicatorLStatus : 59|2@1+ (1,0) [0|3] "" NEO
245      SG_ BC_indicatorRStatus : 61|2@1+ (1,0) [0|3] "" NEO
```

```
[~] sudo apt install python3 can-utils
```

Librairies Python

- python-can
- cantools
- scapy

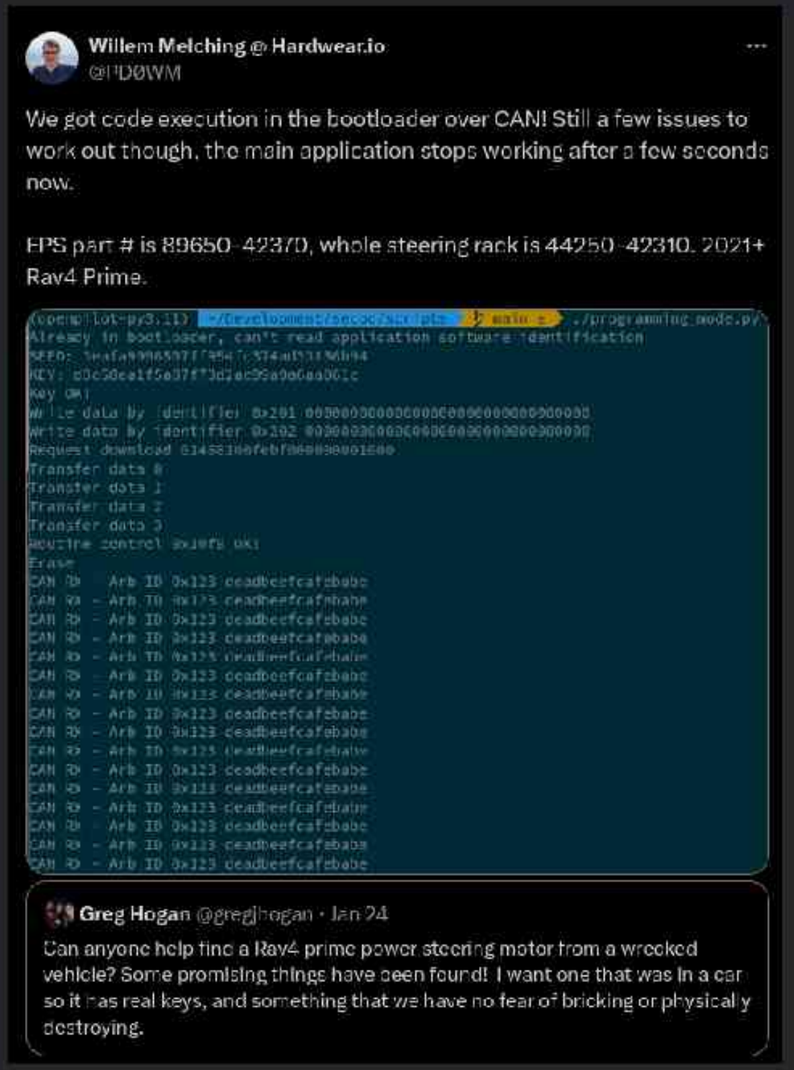
Support Linux in-kernel

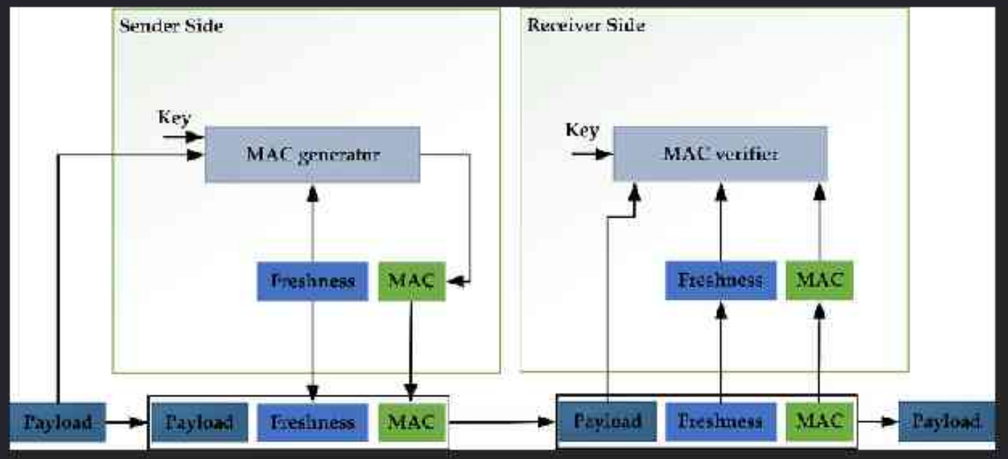
sudo modprobe vcan
sudo ip link add dev vcan0 type vcan
sudo ip link set up vcan0

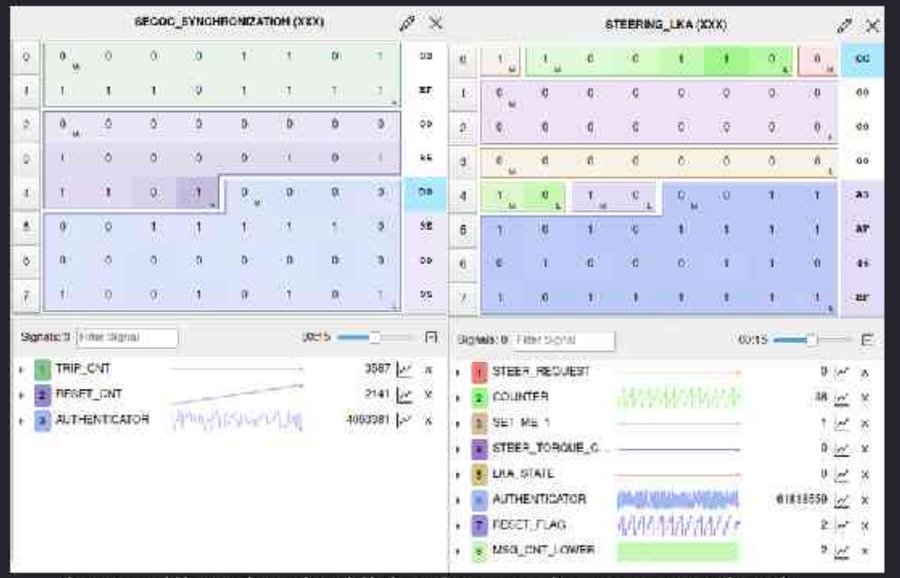
```python
import can

bus = can.Bus(channel='vcan0', interface='socketcan')
while True:
    msg = can.Message(arbitration_id=0xc0ffee, data=[id, i, 0, 1, 3, 1, 4, 1], is_extended_id=False)
    bus.send(msg)
```

[~] secure on-board communications







https://icanhack.nl/blog/secoc-key-extraction/

[~] other protocols & resources

- XCP (Debug / diagnostics)
- FlexRay (communication bus)
- SOME-IP (protocol over IP)
- ...

- **digital kaos**
- **motorcarsoft**
- **techniarabia**
- **autohacking**
- **msieur-lolo.fr**
- **dacianer**
- **medianav.ru**

Thanks :)

Contacts:

https://penthertz.com
https://djnn.sh

To go further:

- hardware reversing (side-channel attacks, JTAG, FCC-IDs)
- RF Hacking (Bluetooth, Digital Audio Broadcasting, RDS, 4G/5G, ...)
- Weaponzing logs (Bluetooth pairing -> DLT)
- MiTM opportunities (Firmware Over-the-air, ...)