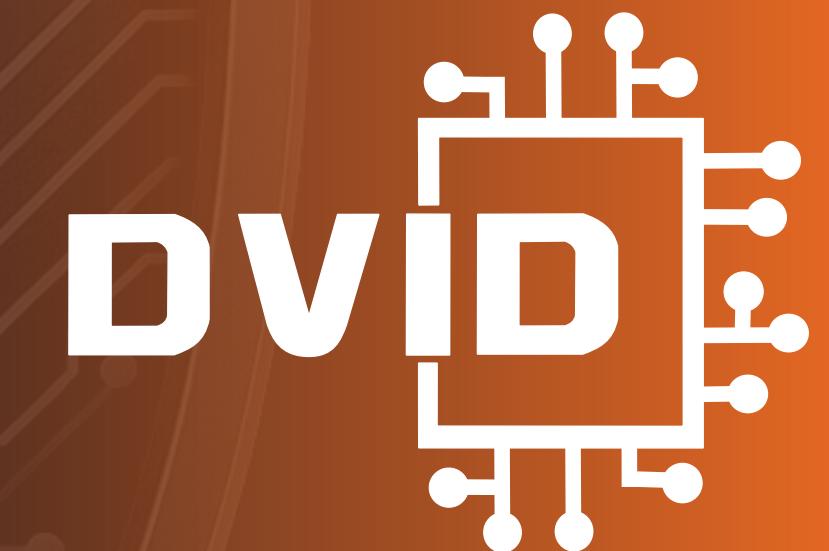


THCON 2025



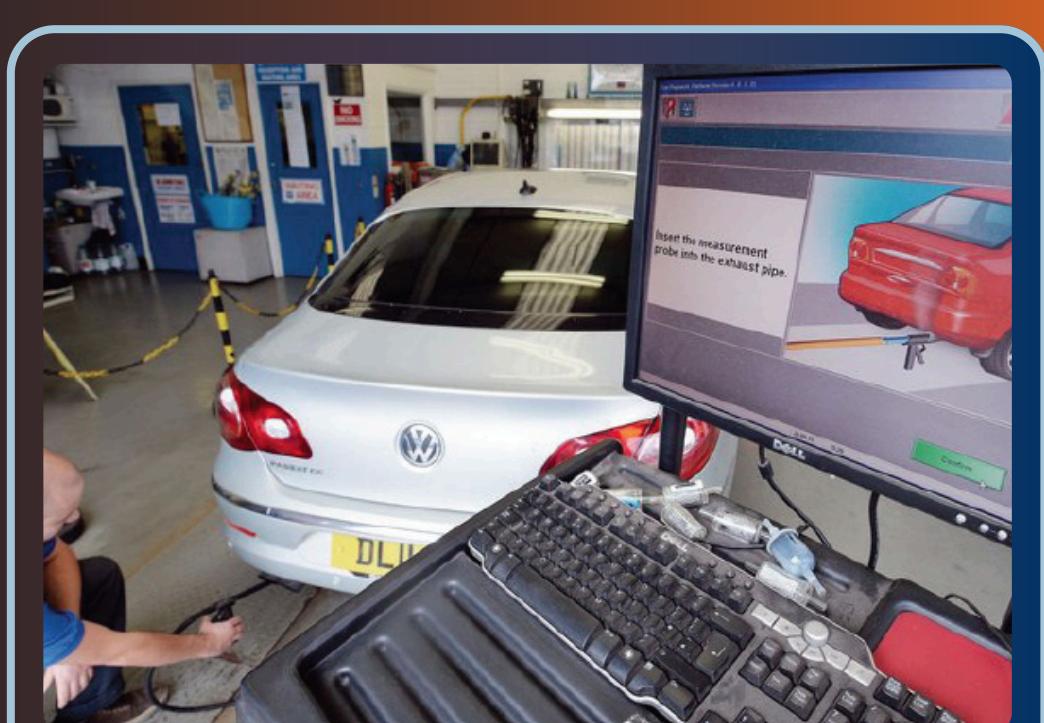
BATTLEFIELD



Pacemaker (2017)

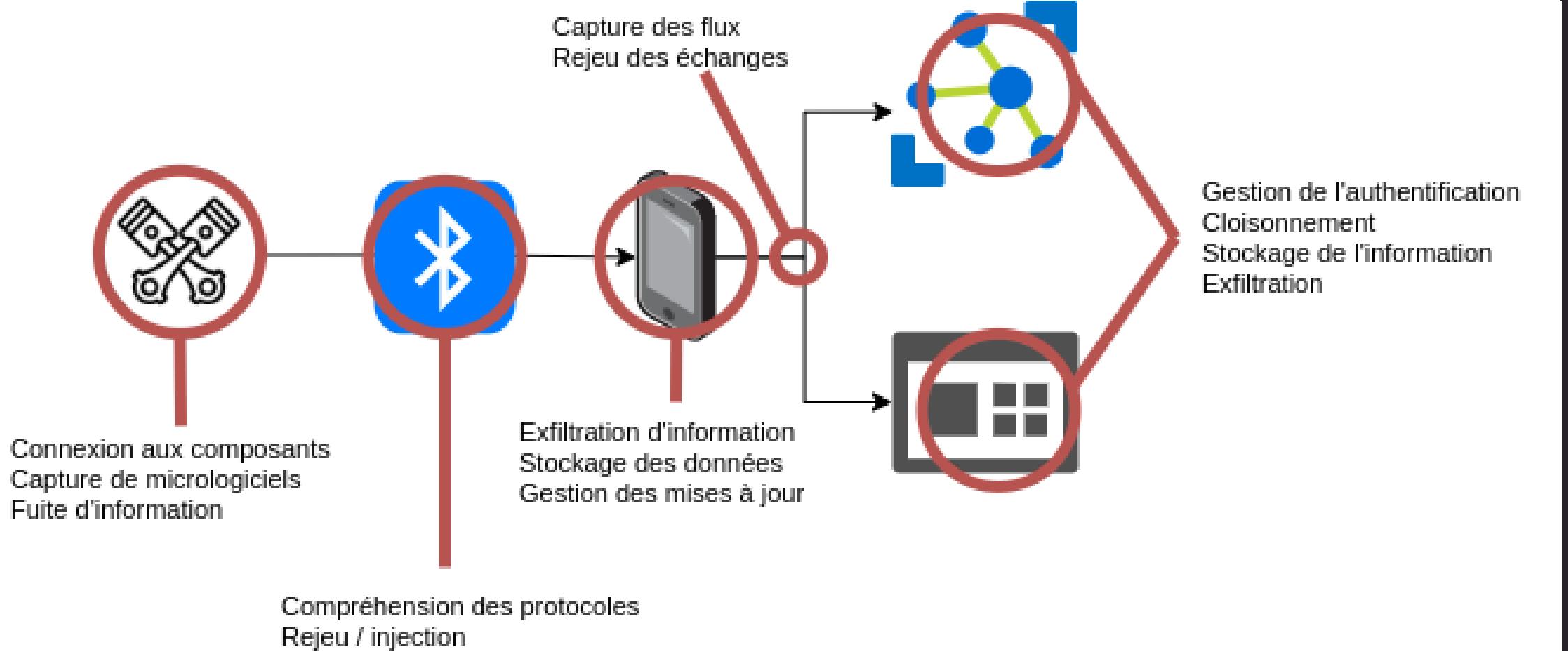


LG SmartThinQ (2016)



Volkswagen (2015)

ECOSYSTEM



30%
of hardware



20%
of cloud computing



20%
of data transport



30%
of only data

PRACTICAL

➤ Debug ports are opened



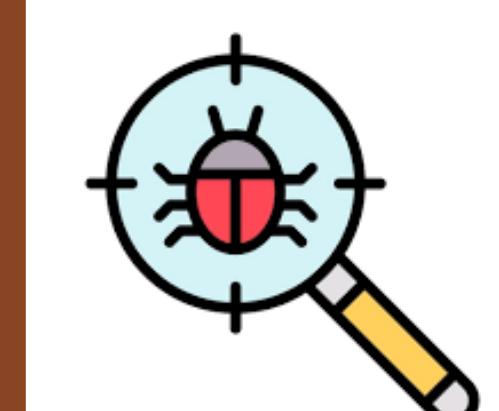
➤ Bootloader memory readable bytes per bytes

```
[ ] crc32 (NEW)
[*] md (NEW)
[*] memcmp (NEW)
[*] memcpy (NEW)
[*] memset (NEW)
[ ] memtest (NEW)
[!] memtester (NEW)
[ ] memory modify (mm) (NEW)
[*] mW (NEW)
```

➤ Firmware reconstruction from exact

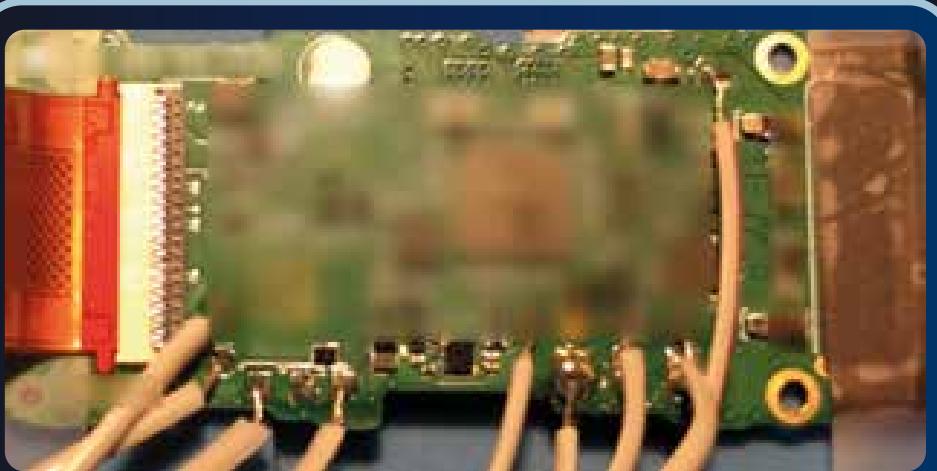
```
=> md.b 0x4000000 0x50
04000000: de ad be ef de
04000010: de ad be ef de
04000020: de ad be ef de
04000030: de ad be ef de
04000040: de ad be ef de
=>
```

➤ Binary analysis

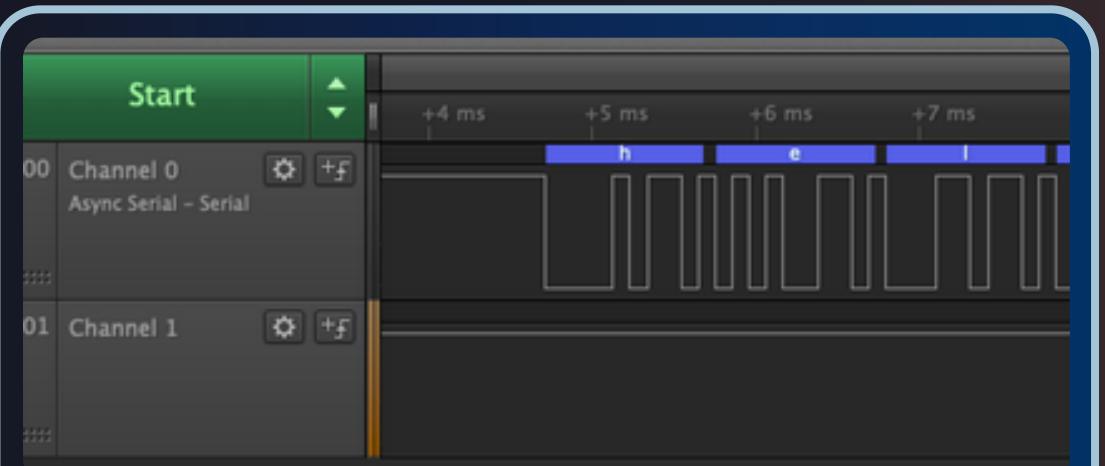


➤ Remote Code Execution finding

IRL



Soldering on PCB



Probing of signals



UART

```
=> md.b 0x4000000 0x50  
04000000: de ad be ef de  
04000010: de ad be ef de  
04000020: de ad be ef de  
04000030: de ad be ef de  
04000040: de ad be ef de  
=>
```

Memory display

gmbnomis/uboot-mdb-dump

1 3 92 24
Contributors Issues Stars Forks

Firmware recovery



Firmware emulation

```
memset(acStack152, 0x0200);  
sprintf(acStrack152, "echo %s>/tmp/%s", pcVar1);  
system(acStrack152)
```

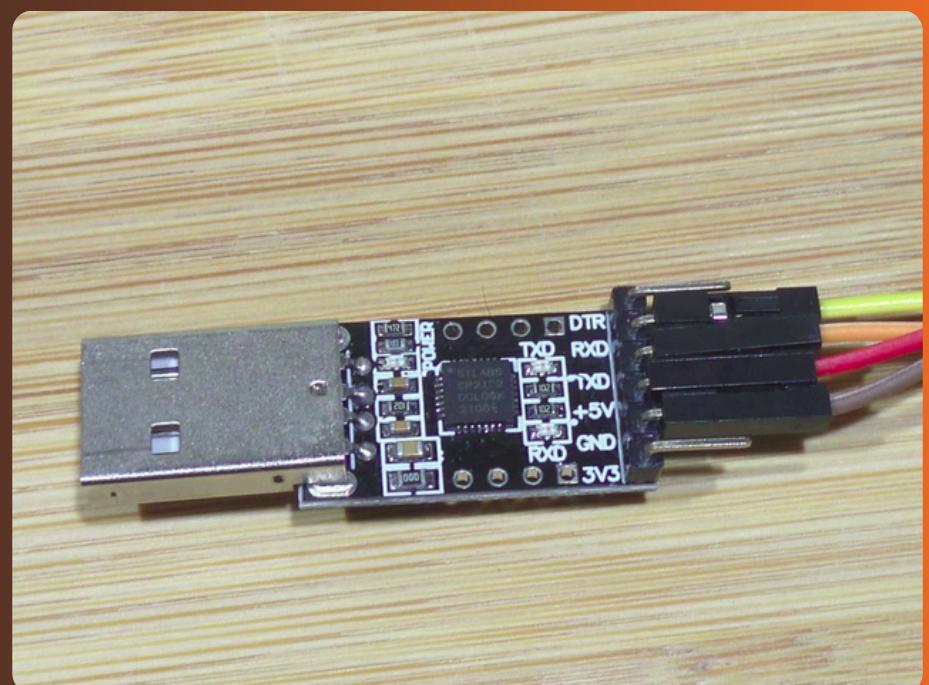
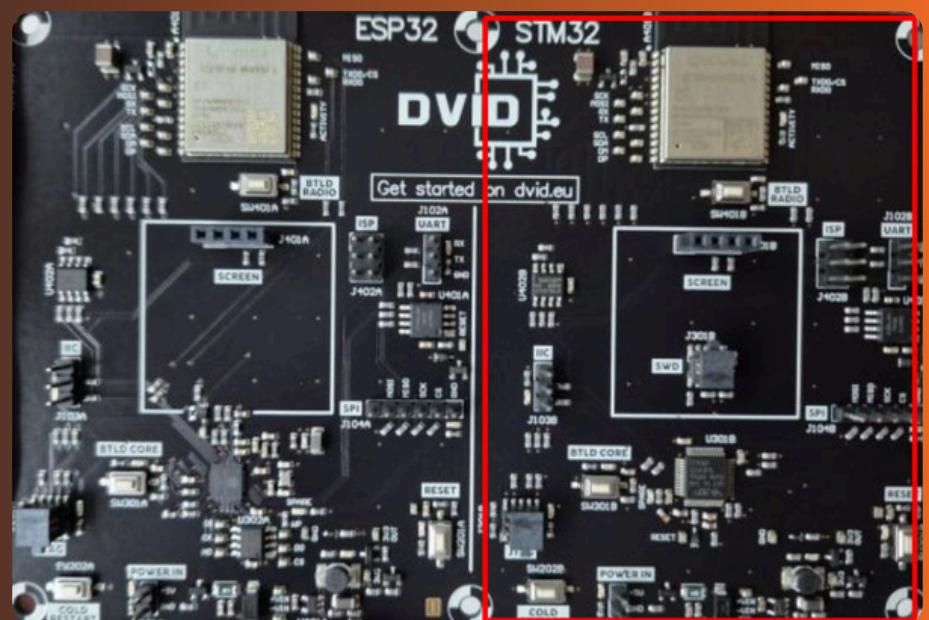
Firmware disassembly

LET'S PLAY YOURSELF

This workshop will focus on STM32 side (right) of the DVID board and explore UART protocol.

OBJECTIVE :

- Wire correctly UART to the STM32 MCU
- Dump firmware to local computer
- Analyze the firmware to find useful information
- Pass the password through UART dongle

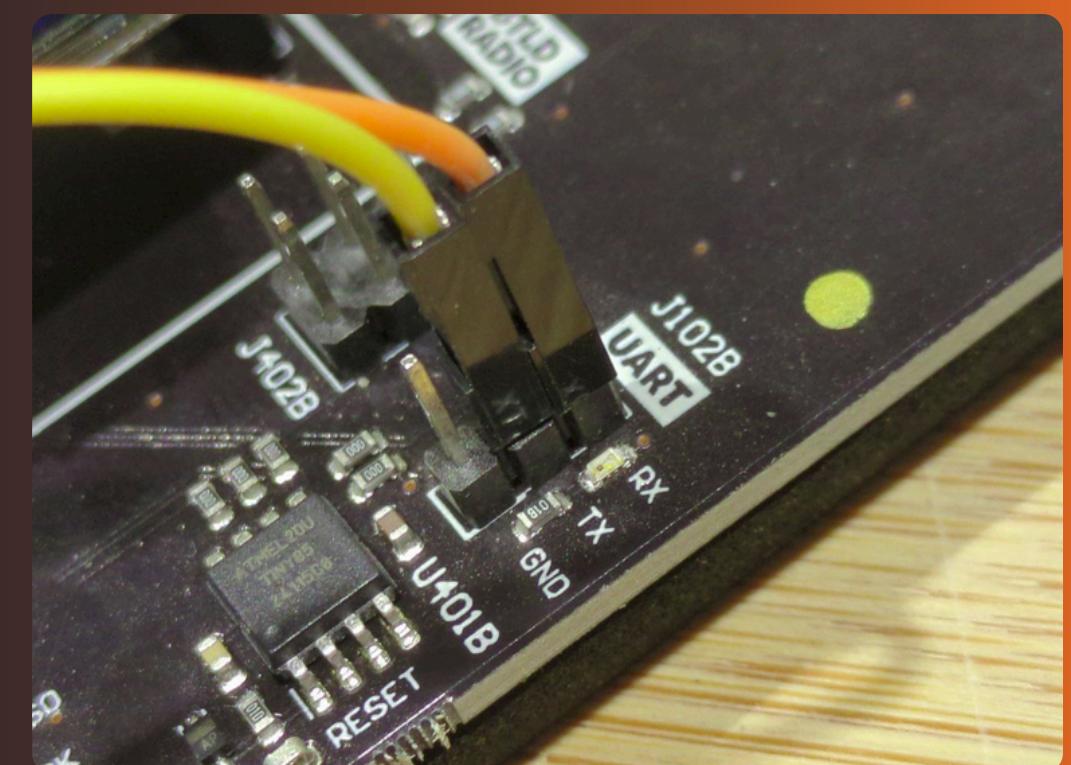
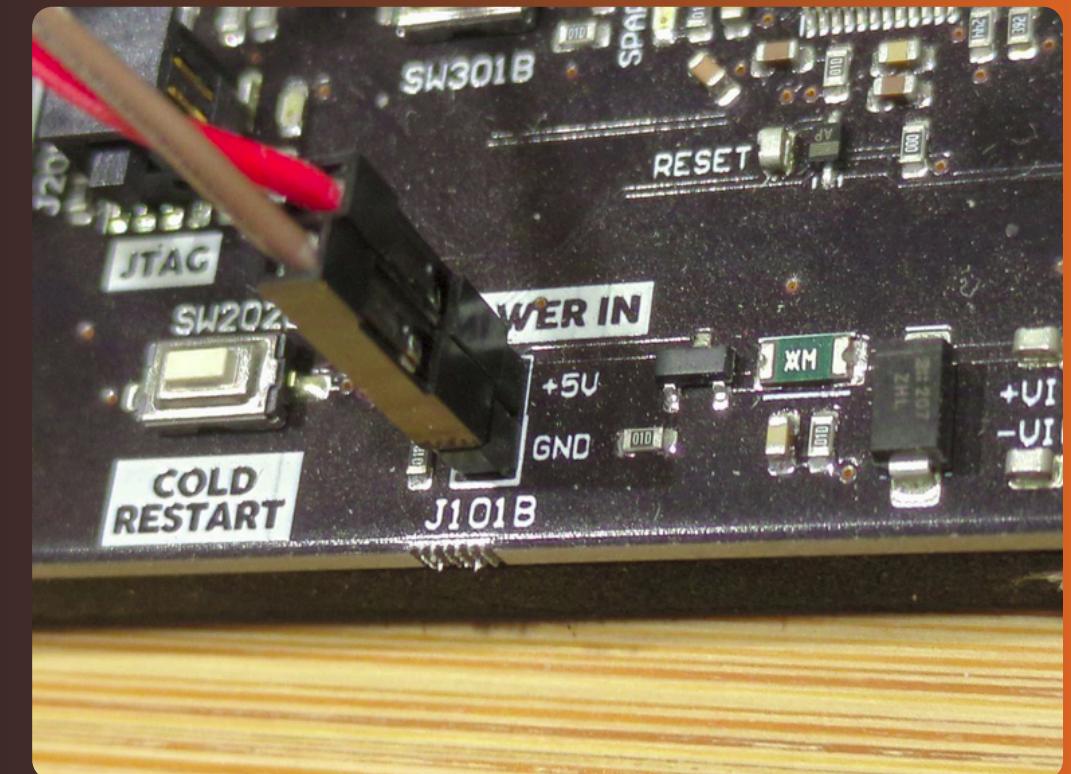
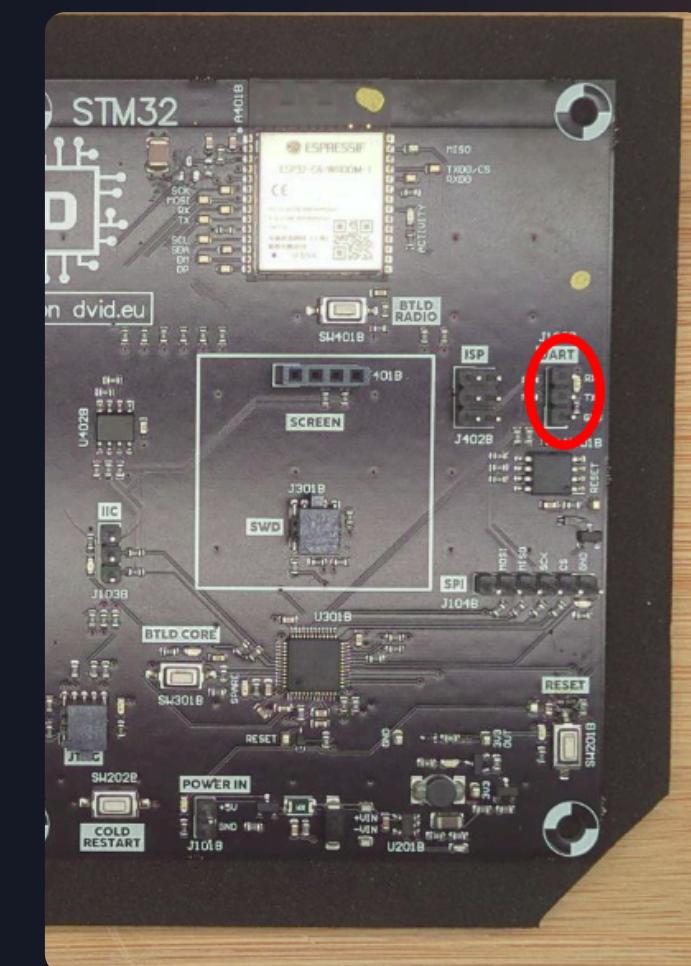
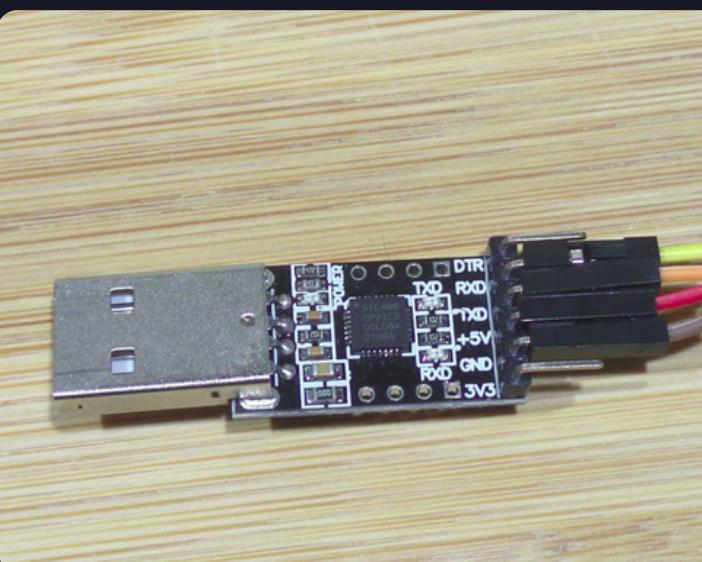


YOUR TURN - WIRE THE BOARD

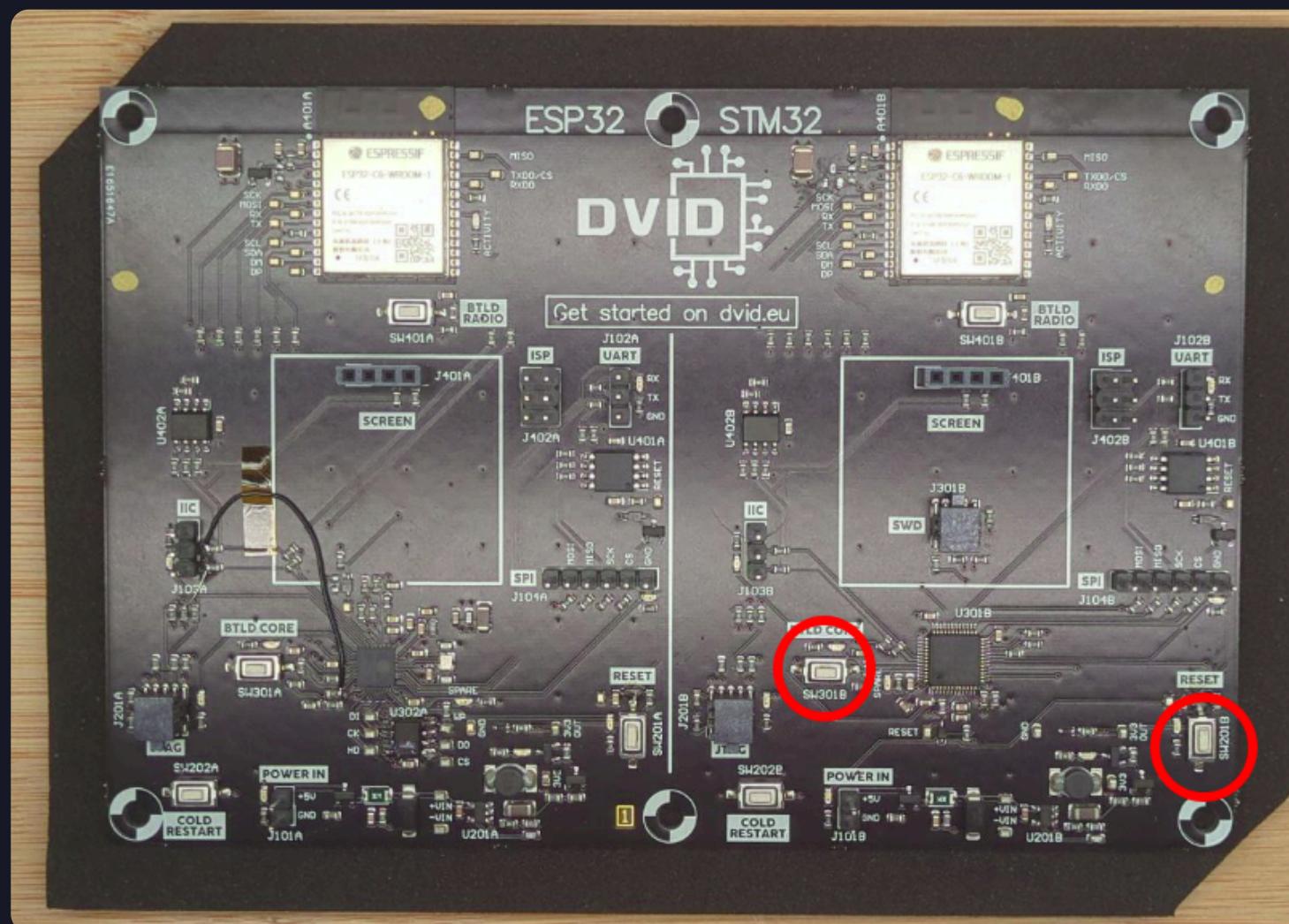
Locate the UART pins on DVID Board (STM32 side)

Remember : TX & RX are crossed wired

Press RESET button to start the board.



YOUR TURN - READ THE FLASH



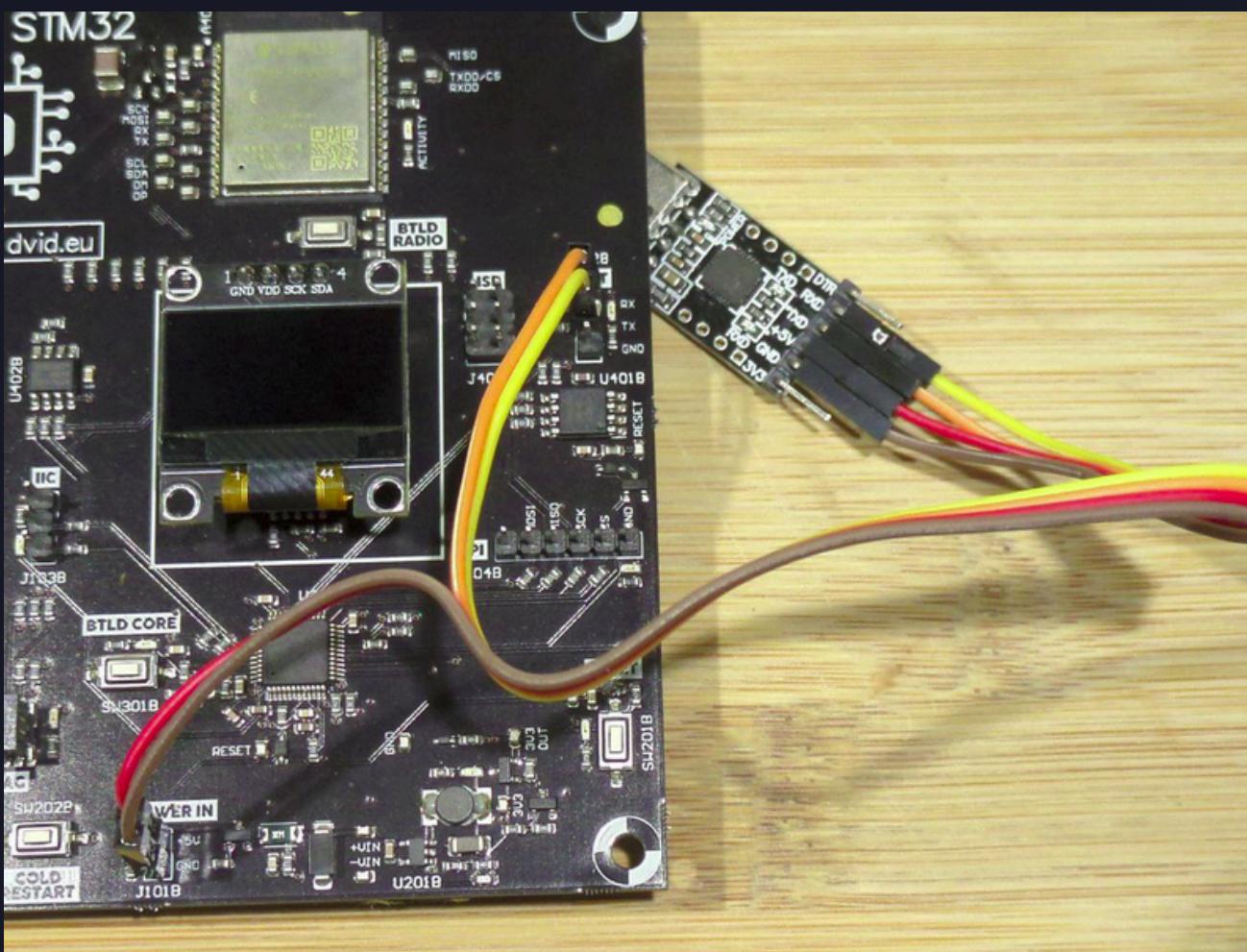
HOLD BTLDCore
HOLD & RELEASE RESET

strings test.bin | grep -i pass

Pass from UART ?
SUPERPASSWORD

Arduino_STM32/tools/linux64/stm32flash/stm32flash -b 115200 -r test.bin /dev/ttyUSB0

YOUR TURN - TALKING IN UART



```
Input: SUPERPASSWORD CR/LF Char delay: 0 ms Send file... Plain
[00:01:45:004] ko
[00:01:49:491] ko
[00:01:52:496] ok
```

Clear Hex output Logging to: /home/vulcainre0/cutecom.log
Device: /dev/ttyUSB0 Connection: 9600 @ 8-N-1

<https://packages.debian.org/fr/sid/amd64/cutecom>

PRACTICAL

- MQTT server accessible with certificate (clientId)
- jokers (#) usage is allowed
- Illegitimate access to other devices data
- Illegitimate access to \$SYS
- GDPR data leakage



IRL

▼ device

- ▼ 269054958815780

batt = 73
state = on
- ▼ 167024525700489

batt = 73
state = on

Device ID = CPAM number
(1 device / person = 1 unique identifier)



1	67	02	45	257	004	89	1 67 02 45 257 004 89	Retirer espaces
Homme	1967	Février	Département: Loire Région: Centre-Val de Loire (anciennement Centre) Pays: France	Commune: Pressigny-les-Pins Canton: Châtillon-Coligny Arrondissement: Montargis Code Postal: 45290	Ok	Ok		

Let's do some quick OSINT

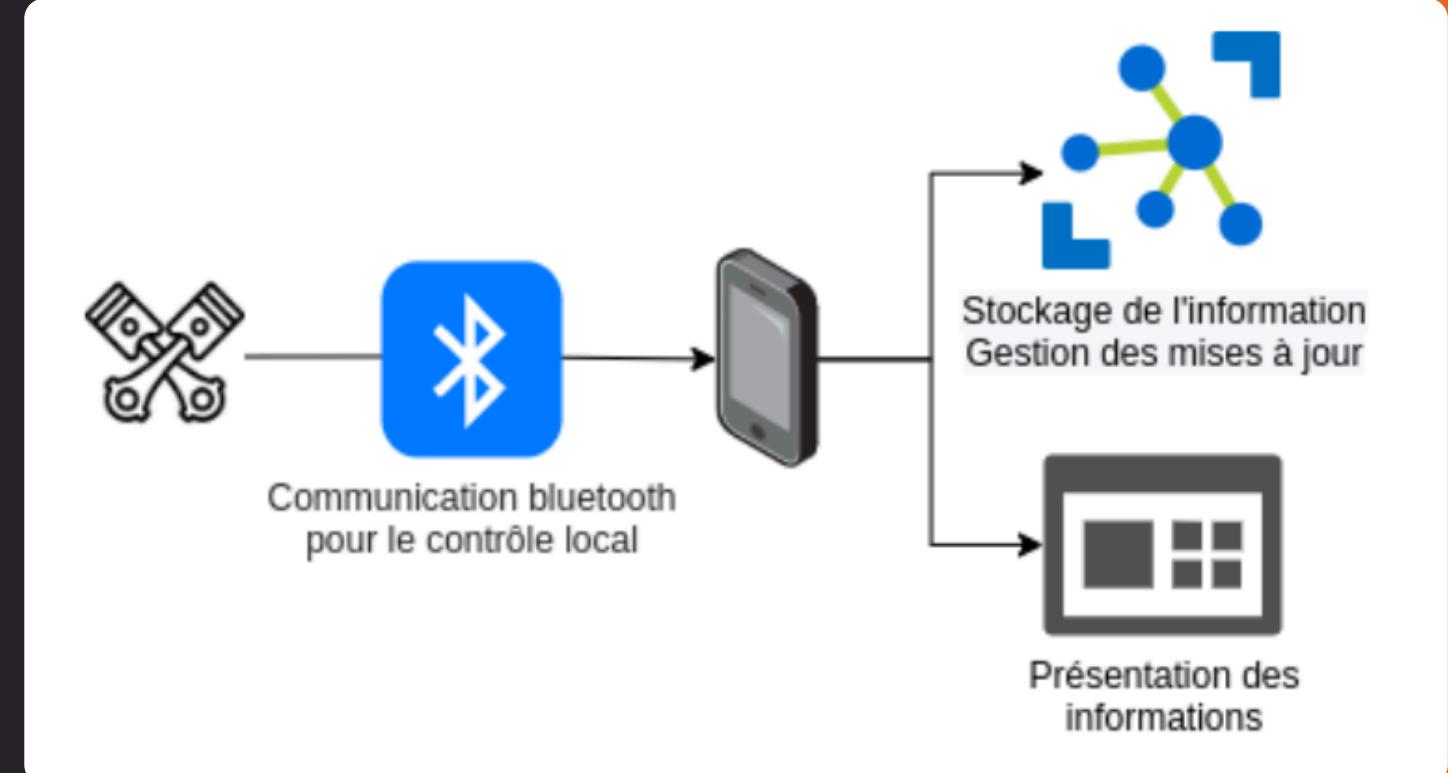
PRACTICAL

- Enroll the device using OTP
- Access to all devices on /devices endpoint
- Convert URL from /me to /device/UUID
- Combine /device/UUID with all devices UUID
- Impersonate device



PRACTICAL

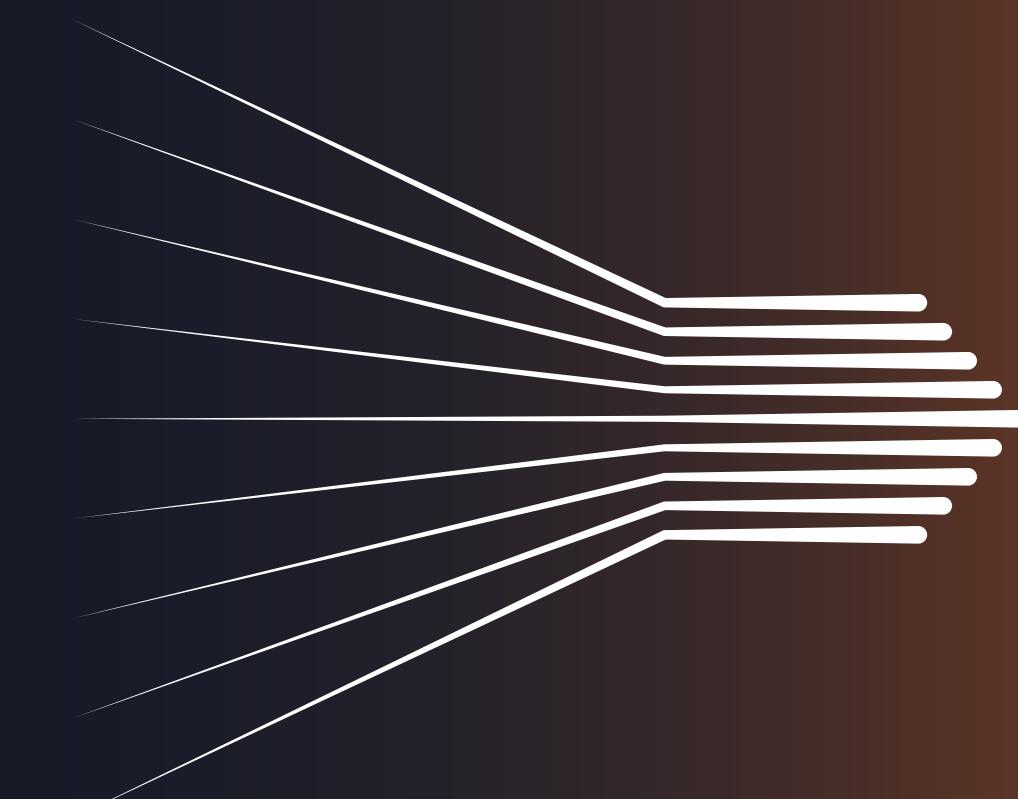
- Unit analysis of equipment
- No injection endpoint on Web app
- Discover how XSS can be injected
- MQTT enrollment allows custom data
- XSS exploit is executed on dashboard UI



Injection of an XSS load in the proprietary field during device enrollment, the payload is executed when the device proprietary is printed on the dashboard

SUMMARY

- Data produced on one side of the chain may only be consumed at the other side
- Debug code VS Production code
- Least privilege principle / features limits flexibility
- Unique identifiers can be personal and require protection
- Default passwords need to be updated



**How to anticipate
vulnerabilities
and learn how to
correct them
right from the
design phase**

WHAT IF GOING MORE FURTHER ?



Immersive experience directly inspired by the IoT world



Tools and techniques that can be reused directly in co



KEEP
CALM
AND
DON'T
SHOP

VULNERABLE IOT DEVICES

STAY “OPEN SOURCED”



<https://github.com/dvid-security/dvidv2-opensource>



Worlwide Open Cyber Security Association

- WOCSA (Worldwide Open Cyber Security Association) is a federation of **non profit association** with its HQ based in France and exists right now on **5 continents**.
- Cybersecurity is a public affair in which **everyone has to take a part**.
- WOCSA joins experts with other people **to take care of our digital life**.
- WOCSA mixes **international projects** with **local actions** based on:
 - a permanent work to develop proximity network,
 - focused on local actions,
 - unbounded collaboration as core way of work.





Worldwide Open Cyber Security Association

