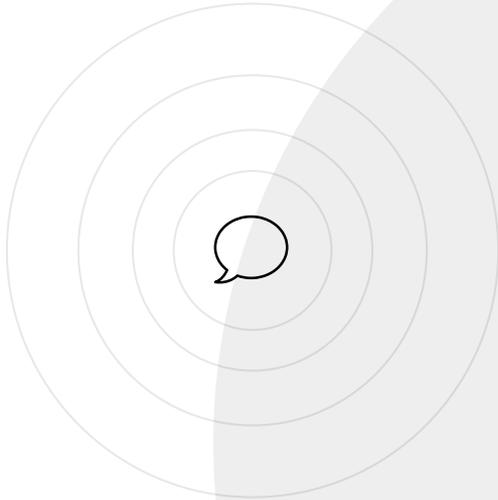




[TEKNIK]

PROBLEMS OF THE DIGITAL AGE: DIGITAL FORENSICS AND DATA RECOVERY



Vinny Dunne

Data Recovery, Digital Forensics and
Electronics lead technician.

Business owner.

Importance of DFIR and DR today

DFIR

- It is essential in the modern world, as criminals increasingly use technology to commit crimes.
- Its importance lies in the resolution of cybercrimes and the protection of sensitive data.

DATA RECOVERY

- It is essential in today's era, providing the crucial ability to restore lost information, ensuring operational continuity, and mitigating the consequences of events such as technical failures, accidental loss, or cyber attacks.
- Additionally, it is also applied in the investigation of security incidents, the recovery of lost data, and the protection of the integrity of digital information.





1

DFIR

**DIGITAL FORENSICS AND
INCIDENT RESPONSE**



Multidisciplinary field that employs techniques from computer science, law, and investigative procedures to collect, analyze, and preserve electronic evidence in order to solve and prevent cybercrimes

Objectives and scope

- Its main objectives include the collection of digital evidence, determination of responsibilities, and prevention of future incidents.
- Its scope covers areas such as cybercrime, computer security, and data recovery.



//

**El tiempo pone cambios.
Estamos entrando en la
época digital. Ni peor ni
mejor: distinta**

- José Mugica, Ex-presidente de Uruguay

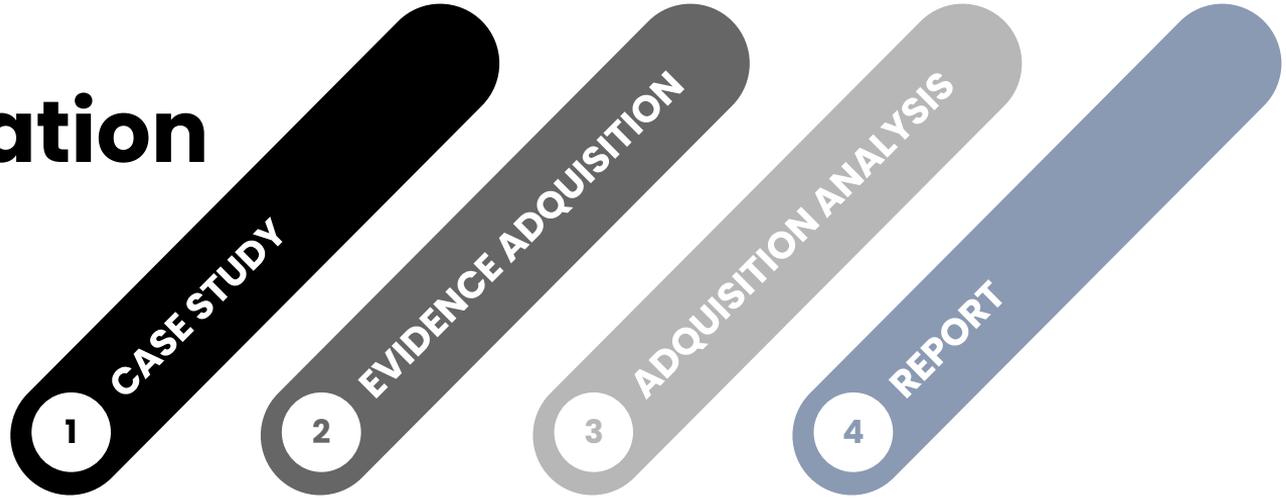


- Analysis of malware on smartphones and mobile phones (Anti-Malware Services).
- Investigation of routes and data storage in Remotely Piloted Aircraft (Drones).
- Identity theft or impersonation.
- Unfair competition investigation.
- Computer sabotage.
- Crimes against intellectual or industrial property.
- Discovery and disclosure of secrets, industrial espionage, etc.
- Investigation and analysis of computers involved in legal disputes or trials under chain of custody.
- Violation of confidentiality or company policies.
- Breach of privacy rights and communication rights in the workplace.
- Investigation of inappropriate computer use during working hours.
- Chat, file, and web browsing history, or any other form of electronic communications.
- Unauthorized access to computer systems.
- Investigation of data deletion, theft of computer data, and information leakage.
- Recovery of data stored on any device or in the cloud.
- Origin of email and instant messaging messages.
- Logs and traces of communications via email and instant messaging.
- Detection of hidden microphones and cameras (Electronic sweep services)



Scenarios

Forensic investigation process



1. Case study

- Main Pillar
- Understanding of the Legal Context
- Interviews with Affected Parties
- Identification of Potential Evidence
- Risk Assessment



2. Adquisition

- Collecting evidence
- Copies
- Mobiles: physical/logical
- Chain of custody → **HASH**
- Must be reproducible

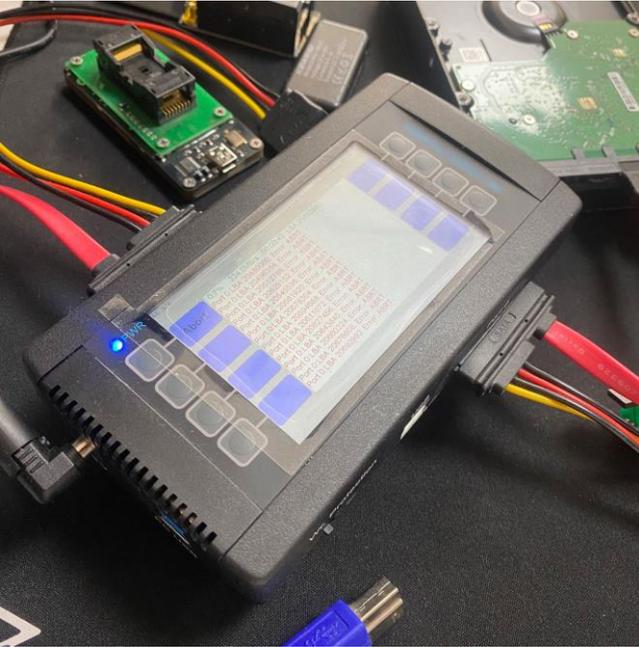




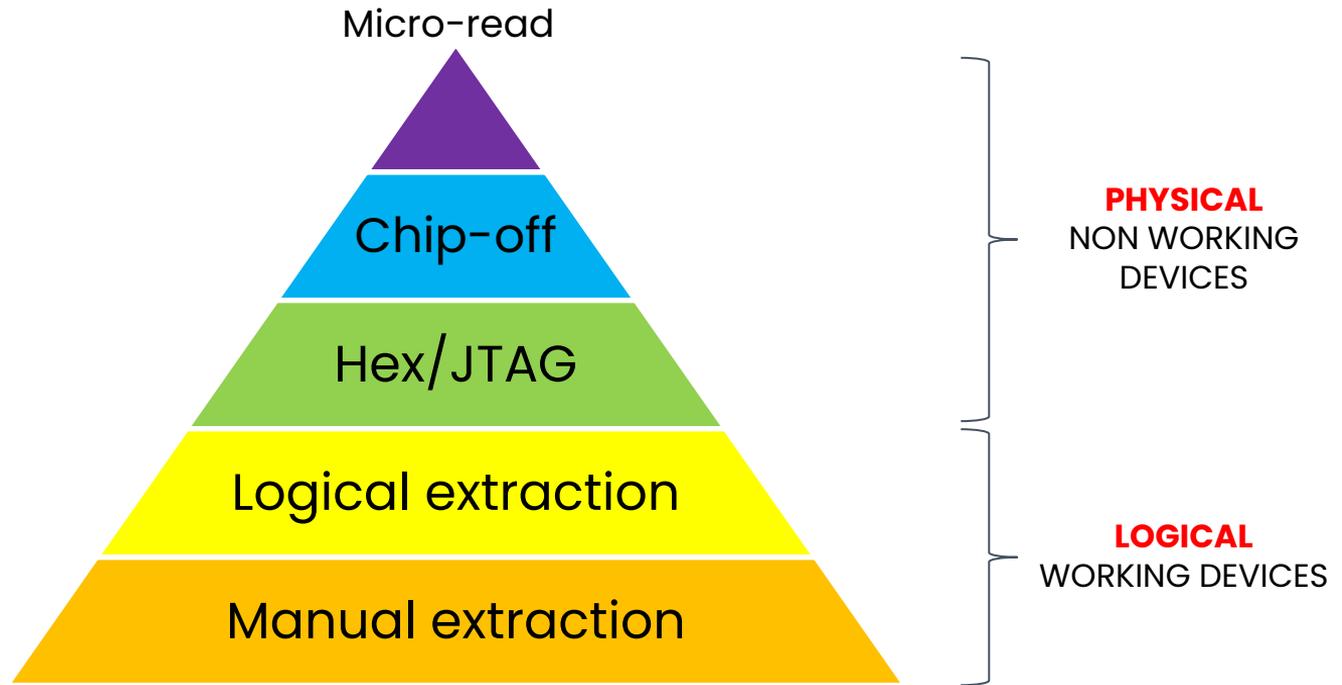
Forensic Workstation



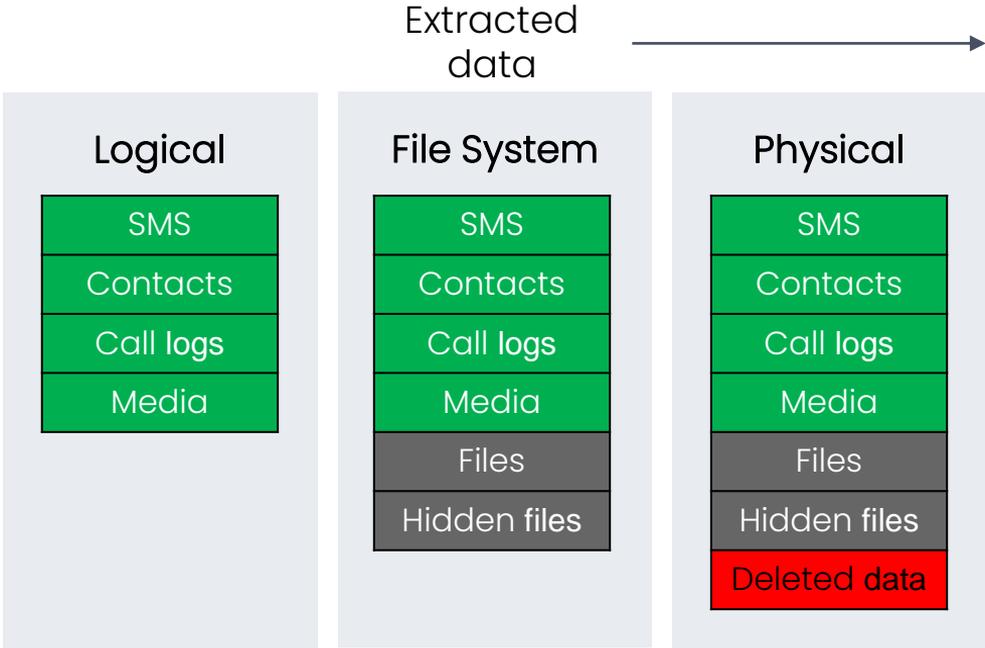
**POLICÍA MUNICIPAL
UDALTZAINGOA**
PAMPLONA · IRUÑA



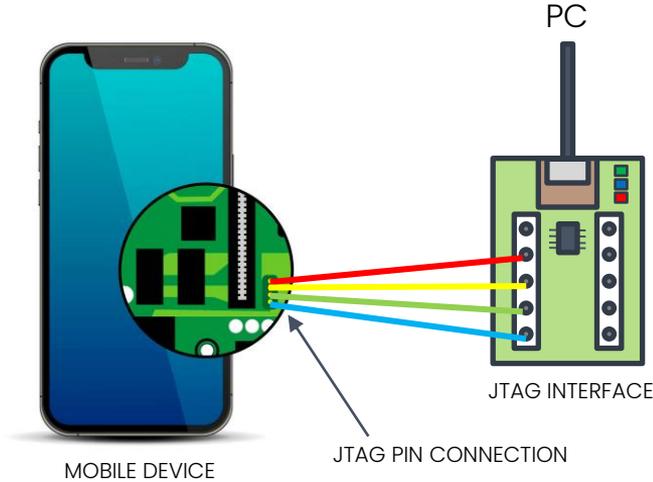
Portable solutions



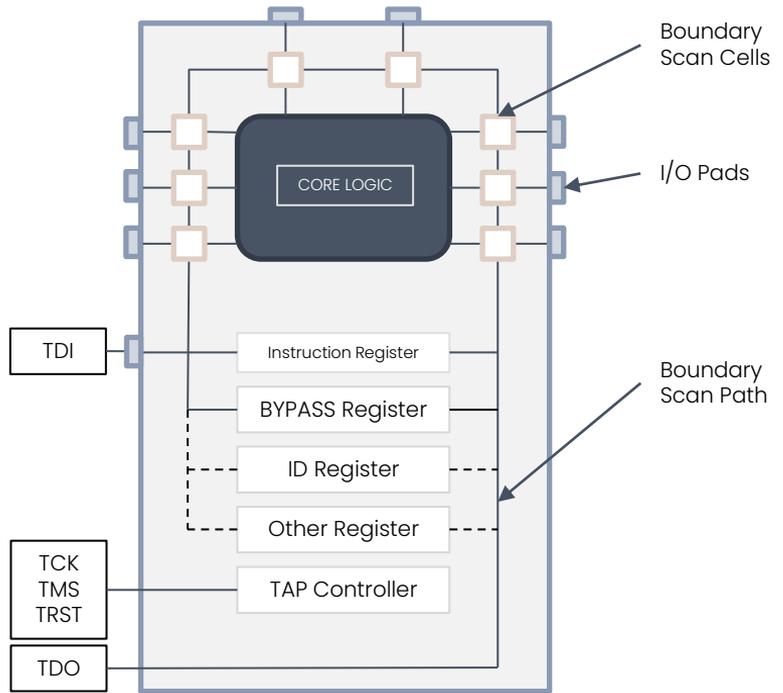
Extraction (mobile devices)



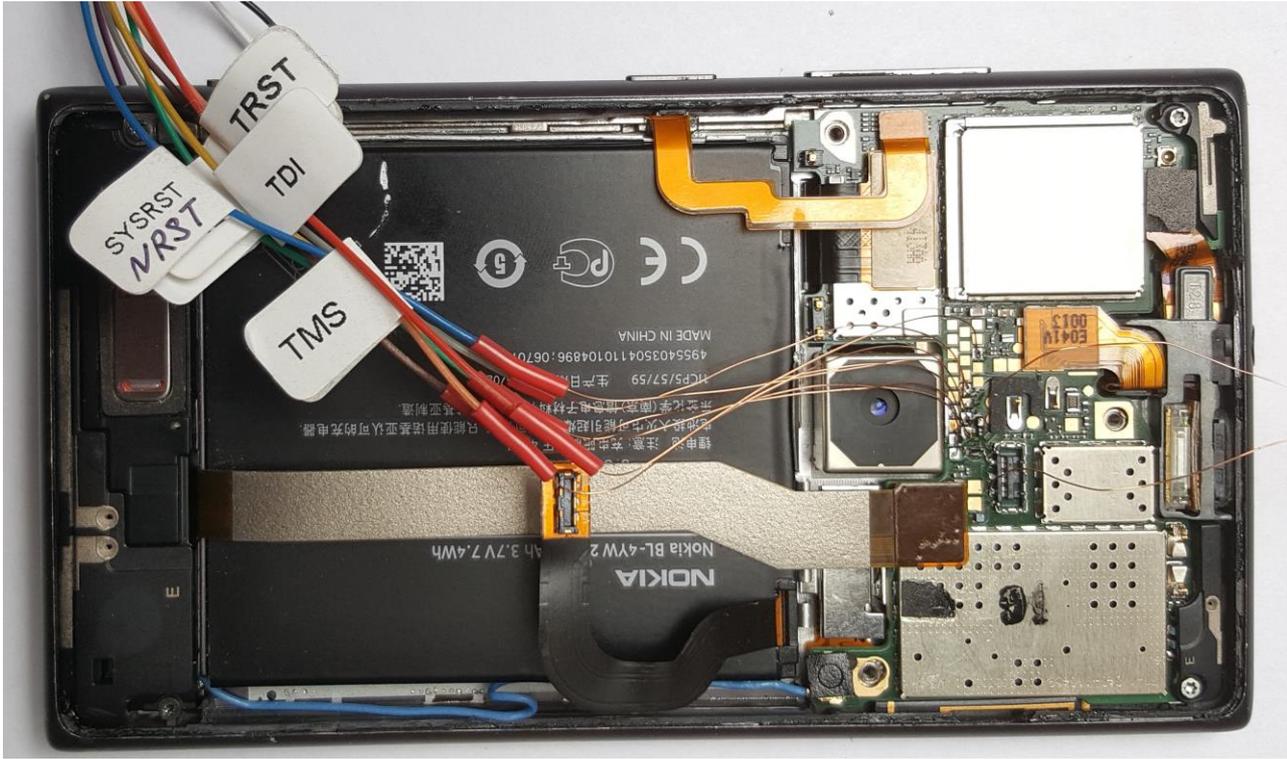
Extraction (mobile devices)



- | Required pins | Optional: |
|---|---|
| <ul style="list-style-type: none">• TDI• TDO• TCK• TMS | <ul style="list-style-type: none">• TRST• RTCK• GND• VCC |



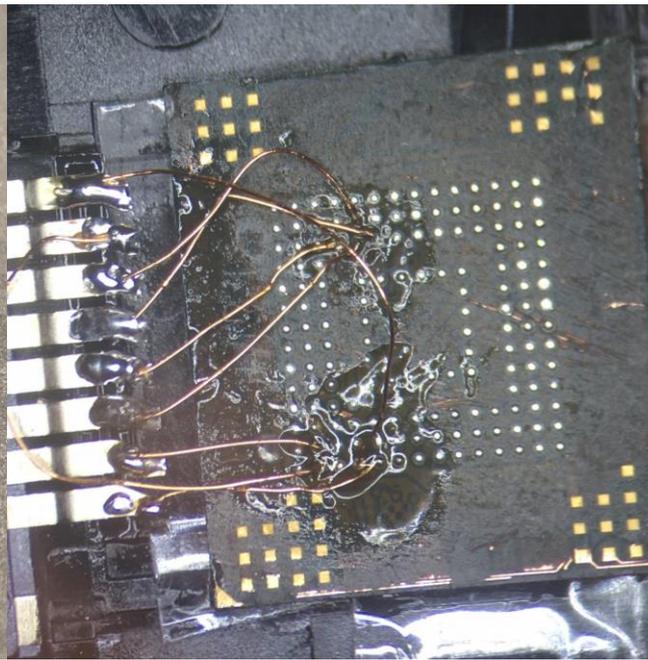
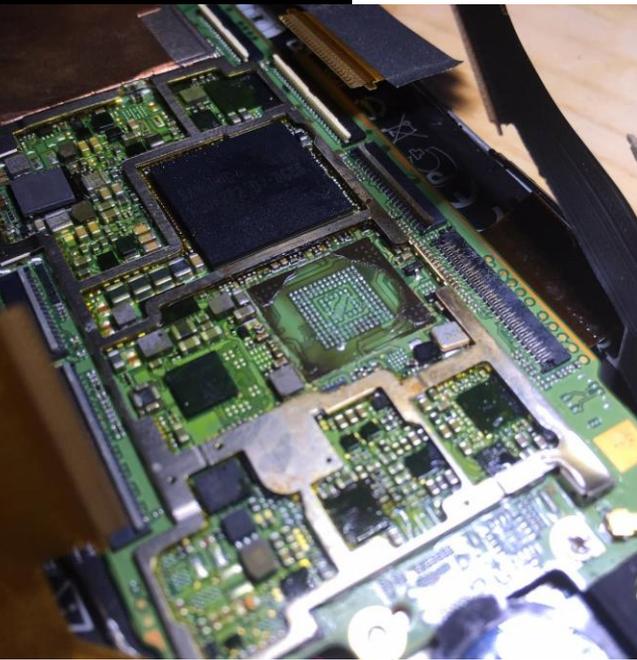
JTAG



Source: <packt>

JTAG

HTC M8 (eMMC)



Chip-off #01

Unbranded phone (eMCP)



Chip-off #02

It is the result of a cryptographic function through a mathematical algorithm that transforms any arbitrary block of data into a series of characters with a fixed length.

Regardless of the length of the input data, the output hash value will always have the same length

$$H : U \rightarrow M$$

$$x \rightarrow h(x)$$



HASH

HEX DUMP

```
00000000      54 45 4B 4E 49 4B      TEKNIK
```

MD5: 0ED4D6C6175378B65CE2E87B7220CBF3

SHA-1: CF9EDA0B9DE7B82E7B7C13F8ADE2DC91912D4CBD

SHA-256:

EE1A30EDB70A4FEF77213C31CDC604D326356FBE676C54F60C6263CFC7DEE5EE



HASH

3. Analysis

- Examine collected data
- Identify patterns
- Recover hidden information
- Data carving
- Reconstruction of events
- Establish timeline



- Almost all files have start and end identifiers or file signatures (headers/footers)
- Some have headers and no footers, or footer is not important for recovery
- Some of them have none at all, which can be an identifier (emails or TXT files)

JPG

FF D8 ...

FF D9

PDF

25 50 44 46 ...

25 25 45 4F 46

GIF

47 49 46 38 ...

00 3B

DOC

D0 CF 11 E0 A1 B1 1A E1...

File struc. Based.

DATA CARVING

Healthy image

```

20231207_220600.jpg 20231122_182726.jpg
Offset (h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Texto decodificado
00000000 FF D8 FF E1 74 BB 45 78 69 66 00 00 49 49 2A 00 0x00000000
00000010 08 00 00 00 0C 00 00 01 04 00 01 00 00 00 80 0F .....€
00000020 00 00 01 01 04 00 01 00 00 00 B8 08 00 00 0F 01 .....ž
00000030 02 00 08 00 00 0E 00 00 00 10 01 02 00 09 00 .....
00000040 00 00 A6 00 00 00 12 01 03 00 01 00 00 00 01 00 .....
00000050 00 00 1A 01 05 00 01 00 00 00 D2 00 00 00 1B 01 .....0
00000060 05 00 01 00 00 00 DA 00 00 00 28 01 03 00 01 00 .....Û (.....
00000070 00 00 02 00 00 00 31 01 02 00 0E 00 00 00 B0 00 .....1.....*
00000080 00 00 32 01 02 00 14 00 00 00 BE 00 00 00 13 02 .....2.....%
00000090 03 00 01 00 00 00 01 00 00 00 69 87 04 00 01 00 .....i#.....
000000A0 00 00 E2 00 00 00 BC 02 00 00 73 61 6D 73 75 6E .....ä...4...samsun
000000B0 67 00 53 4D 2D 47 39 38 38 42 00 00 47 39 38 38 g.SM-G988B..G988
000000C0 42 58 58 53 49 48 57 49 37 00 32 30 32 33 3A 31 BXXSIHWI7.2023:1
000000D0 31 3A 32 32 20 31 38 3A 32 37 3A 32 36 00 48 00 l:22 18:27:26.H.
000000E0 00 00 01 00 00 00 48 00 00 00 01 00 00 00 1C 00 .....H.....
000000F0 9A 82 05 00 01 00 00 00 78 02 00 00 9D 82 05 00 š,.....x.....
00000100 01 00 00 00 70 02 00 00 22 88 03 00 01 00 00 00 .....p...".
00000110 02 00 00 00 27 88 03 00 01 00 00 00 C4 09 00 00 ....."......Ä
00000120 00 90 07 00 04 00 00 00 30 32 32 30 03 90 02 00 .....0220....
00000130 14 00 00 00 38 02 00 00 04 90 02 00 14 00 00 00 .....8.....
00000140 4C 02 00 00 10 90 02 00 07 00 00 00 60 02 00 00 L.....'.....
00000150 11 90 02 00 07 00 00 00 68 02 00 00 01 92 05 00 .....h.....
00000160 01 00 00 00 80 02 00 00 02 92 05 00 01 00 00 00 .....€.....'.....
00000170 88 02 00 00 04 92 0A 00 01 00 00 00 90 02 00 00 ...../.....'.....
00000180 05 92 05 00 01 00 00 00 98 02 00 00 07 92 03 00 ...../.....'.....
00000190 01 00 00 00 02 00 00 00 09 92 03 00 01 00 00 00 ...../.....'.....
000001A0 00 00 00 00 0A 92 05 00 01 00 00 00 A8 02 00 00 ...../.....'.....
000001B0 90 92 02 00 04 00 00 00 34 39 34 00 91 92 02 00 ...../.....494.'...
000001C0 04 00 00 00 34 39 34 00 92 92 02 00 04 00 00 00 .....494.'...
000001D0 34 39 34 00 01 A0 03 00 01 00 00 00 01 00 00 00 .....494.'...
000001E0 02 A0 04 00 01 00 00 00 80 0F 00 00 03 A0 04 00 .....€.....
000001F0 01 00 00 00 B8 08 00 00 02 A4 03 00 01 00 00 00 .....H.....
00000200 00 00 00 00 03 A4 03 00 01 00 00 00 00 00 00 00 .....H.....
00000210 04 A4 05 00 01 00 00 00 A0 02 00 00 05 A4 03 00 .....H.....
00000220 01 00 00 00 19 00 00 00 06 A4 03 00 01 00 00 00 .....H.....
00000230 00 00 00 00 20 A4 02 00 00 00 00 00 B0 02 00 00 .....H.....
00000240 00 00 00 00 32 30 32 33 3A 31 31 3A 32 32 20 31 .....2023:11:22 1
00000250 38 3A 32 37 3A 32 36 00 32 30 32 33 3A 31 31 3A 8:27:26.2023:11:
00000260 32 32 20 31 38 3A 32 37 3A 32 36 00 2B 30 31 3A 22 18:27:26.+01:
00000270 30 30 00 00 2B 30 31 3A 30 30 00 00 DC 00 00 00 00...+01:00..Û...
00000280 64 00 00 00 01 00 00 00 0D 00 00 00 01 00 00 00 d.....
    
```

Editores especiales

Inspector de datos

Binary (8 bit) 11111111

Int8	lr.a: -1
UInt8	lr.a: 255
Int16	lr.a: -9985
UInt16	lr.a: 55551
Int24	lr.a: -9985
UInt24	lr.a: 16767231
Int32	lr.a: Inválido
UInt32	lr.a: Inválido
Int64	lr.a: Inválido
UInt64	lr.a: Inválido
LEB128	lr.a: Inválido
ULEB128	lr.a: Inválido
AnsiChar / char8_t	y
WideChar / char16_t	□
Punto de código UTF-8	Unidad de código no válida
Single (float32)	Inválido
Double (float64)	Inválido
OLETIME	Inválido
FILETIME	Inválido
DOS date	31/07/2088
DOS time	Inválido
DOS time & date	Inválido
time_t (32 bit)	Inválido
time_t (64 bit)	Inválido
GUID	Inválido
Disassembly (x86-16)	Inválido
Disassembly (x86-32)	Inválido
Disassembly (x86-64)	Inválido

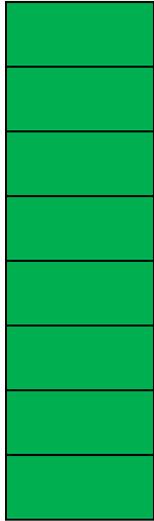
DATA CARVING

Empty image

The screenshot shows a hex editor window with two tabs: '20231207_220600.jpg' and '20231122_182726.jpg'. The main window displays a hex dump of the selected file. The first few bytes are 00 00 00 00, indicating an empty image. The 'Texto decodificado' column shows a series of dots representing the decoded text. To the right, the 'Editores especiales' window is open, showing the 'Inspector de datos' (Data Inspector) for the selected memory address. The inspector displays various data types and their values, such as Binary (8 bit) as 00000000, Int8 as 0, and time_t (32 bit) as 01/01/1970.

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Texto decodificado
00000000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000200	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000210	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000220	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000230	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000240	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000250	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000260	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000270	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000280	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

DATA CARVING



Hex table



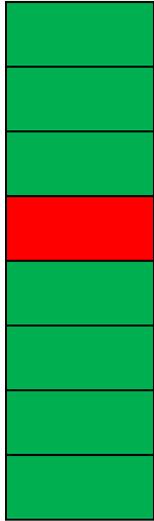
Healthy image



Hex table



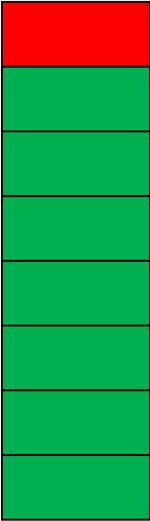
Barely damaged



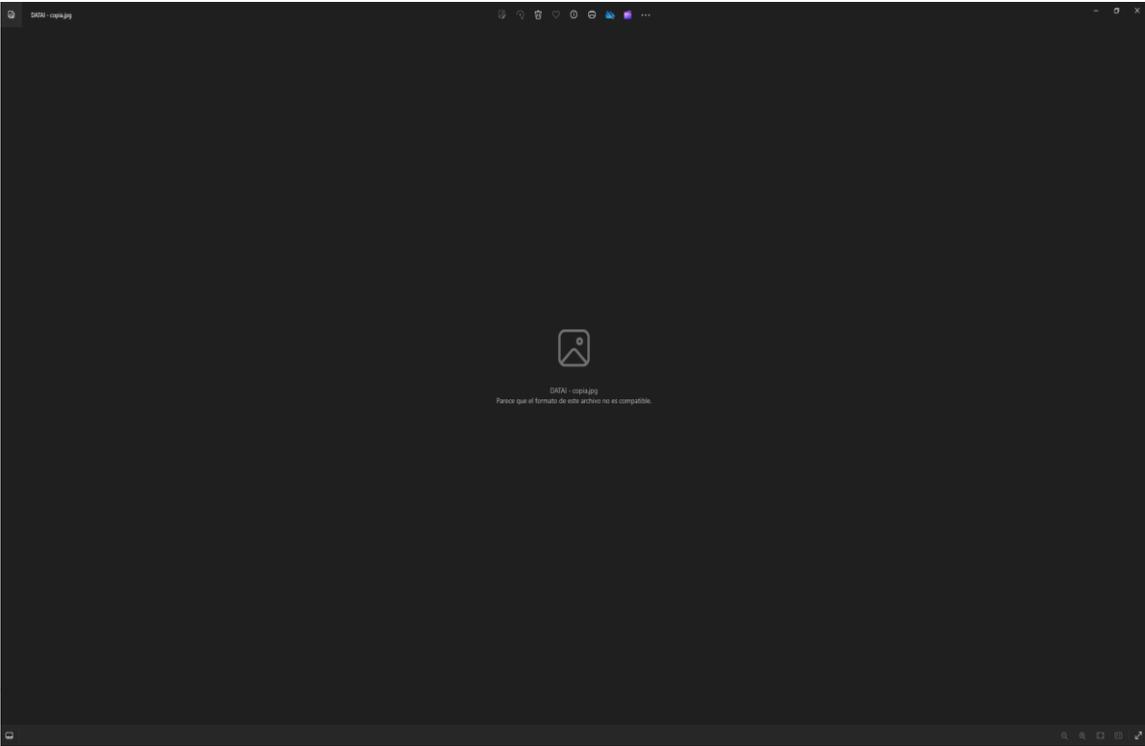
Hex table



Badly damaged



Hex table



Corrupt

Process list →

```

Progress: 0.00 Scanning layer_name using PdbSignatureScan
Progress: 100.00 PDB scanning finished
Volatility 3 Framework 2.0.2

PID PPID ImageFileName Offset(V) Threads Handles SessionId
Wow64 CreateTime ExitTime File output
4 0 System 0xbf0f64a63080 132 - N/A False 2
021-04-30 12:39:40.000000 N/A Disabled
108 4 Registry 0xbf0f64bc6040 4 - N/A F
alse 2021-04-30 12:39:38.000000 N/A Disabled
396 4 smss.exe 0xbf0f66967040 2 - N/A F
alse 2021-04-30 12:39:40.000000 N/A Disabled
492 484 csrss.exe 0xbf0f6adb6080 13 - 0 F
alse 2021-04-30 12:39:44.000000 N/A Disabled
568 484 wininit.exe 0xbf0f6b67a080 1 - 0 F

```

```

Volatility 3 Framework 2.0.2
Progress: 0.00 Scanning layer_name using PdbSignatureScan
Progress: 0.00 Scanning layer_name using PdbSignatureScan
Progress: 100.00 PDB scanning finished

User rid lmhash nthash
Administrator 500 aad3b435b51404eeaad3b435b51404ee 31d6cfe0d1
6ae931b73c59d7e0c089c0
Guest 501 aad3b435b51404eeaad3b435b51404ee 31d6cfe0d16ae931b7
3c59d7e0c089c0
DefaultAccount 503 aad3b435b51404eeaad3b435b51404ee 31d6cfe0d1
6ae931b73c59d7e0c089c0
WDAGUtilityAccount 504 aad3b435b51404eeaad3b435b51404ee 69
dbee1a98d4f53fbccb1fe5ce37c851
John Doe 1001 aad3b435b51404eeaad3b435b51404ee ecf53750b7

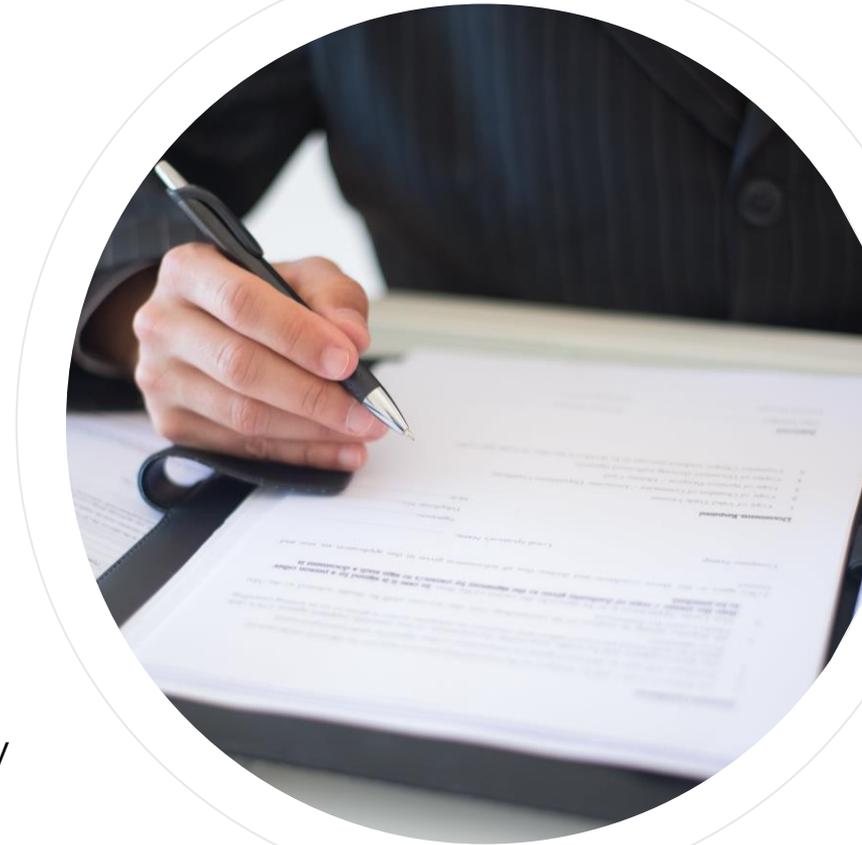
```

← User passwords

DATA CARVING (RAM)

4. Report

- UNE 197010/2015
- Objectives/scope
- Methodology
- Additional documentation
- Conclusions
- Clear, precise, and supported by solid evidence

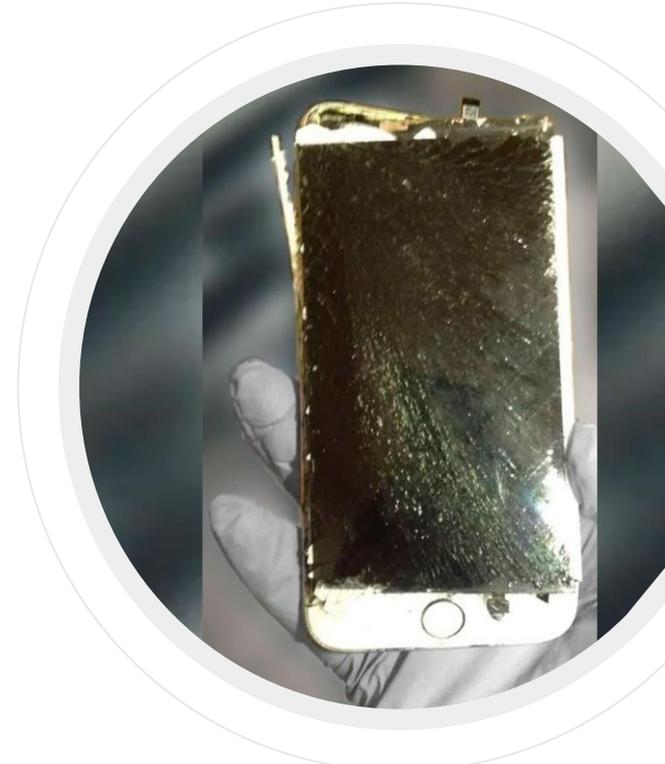


1

DIANA QUER

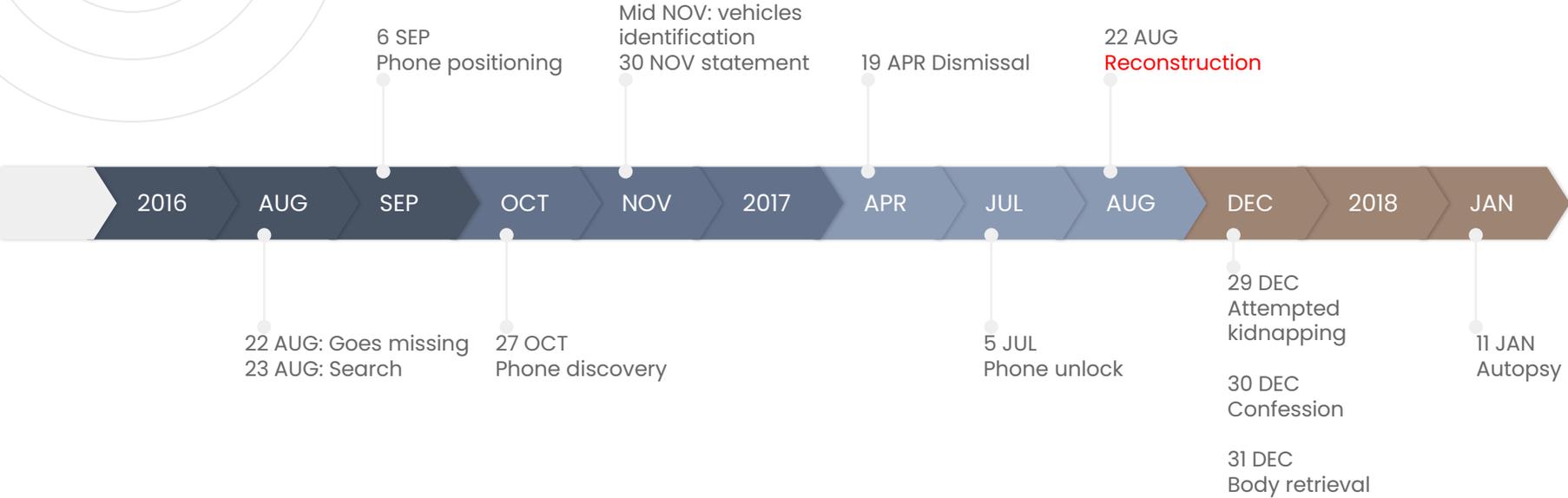


- Went missing 22nd August 2016
- Body retrieval 31st December 2017



iPhone 6
iOS 9

Chrono: Diana Quer



2

MARTA DEL CASTILLO



- Went missing 24th January 2009
- Her body has not been found yet

LAZARUS®



Motorola U9
Linux / Java

3

“LA MANADA”

- Events: 7th July 2016



LG G3S



iPhone 5



Vodafone Smart First 6



LG L9 II



Google Nexus 5



2

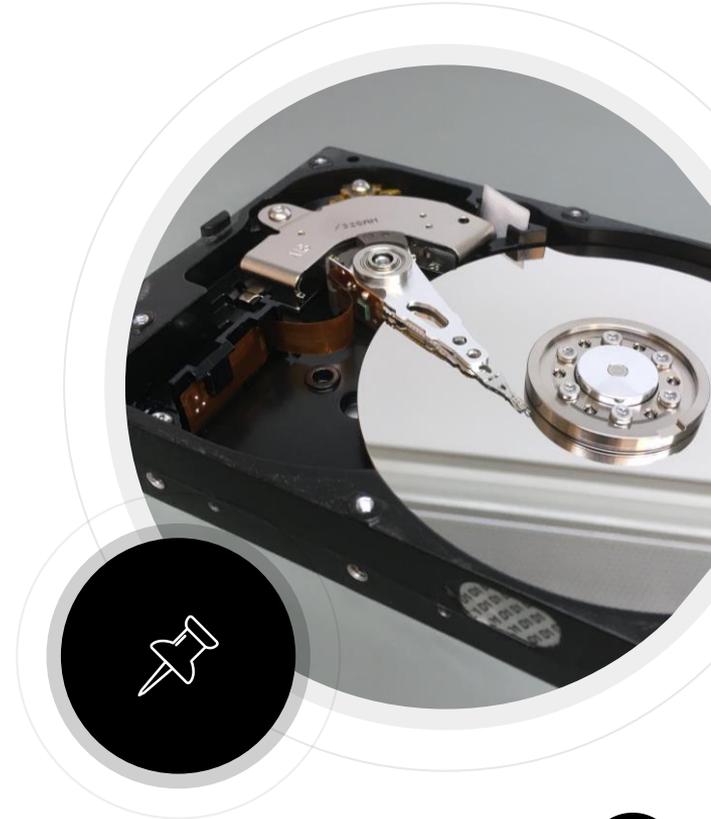
Data Recovery



Set of procedures used to access information stored on a digital medium that, for various reasons, is not accessible conventionally

Scenarios

- Accidental deletion
- Data corruption
- Disk formatting
- System failure
- Attacks, malware, virus/ransomware
- Physical damage to the device



Storage media

- Tapes
- HDD/SSD disks
- RAID/NAS/Servers systems
- USB drives
- Memory cards
- Mobile devices



Common cases

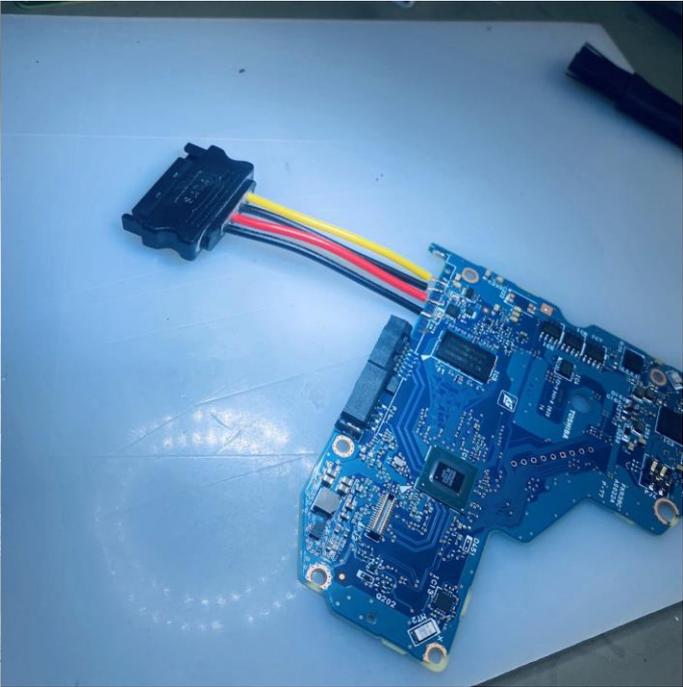
- Basic logical
- Advanced logical
- Electronic fault
- Media or firmware issues
- Physical

Problems

- + Density/+ Heads
- Helium drives
- Scarcity of old drives
- Searching for donors
- Time consuming



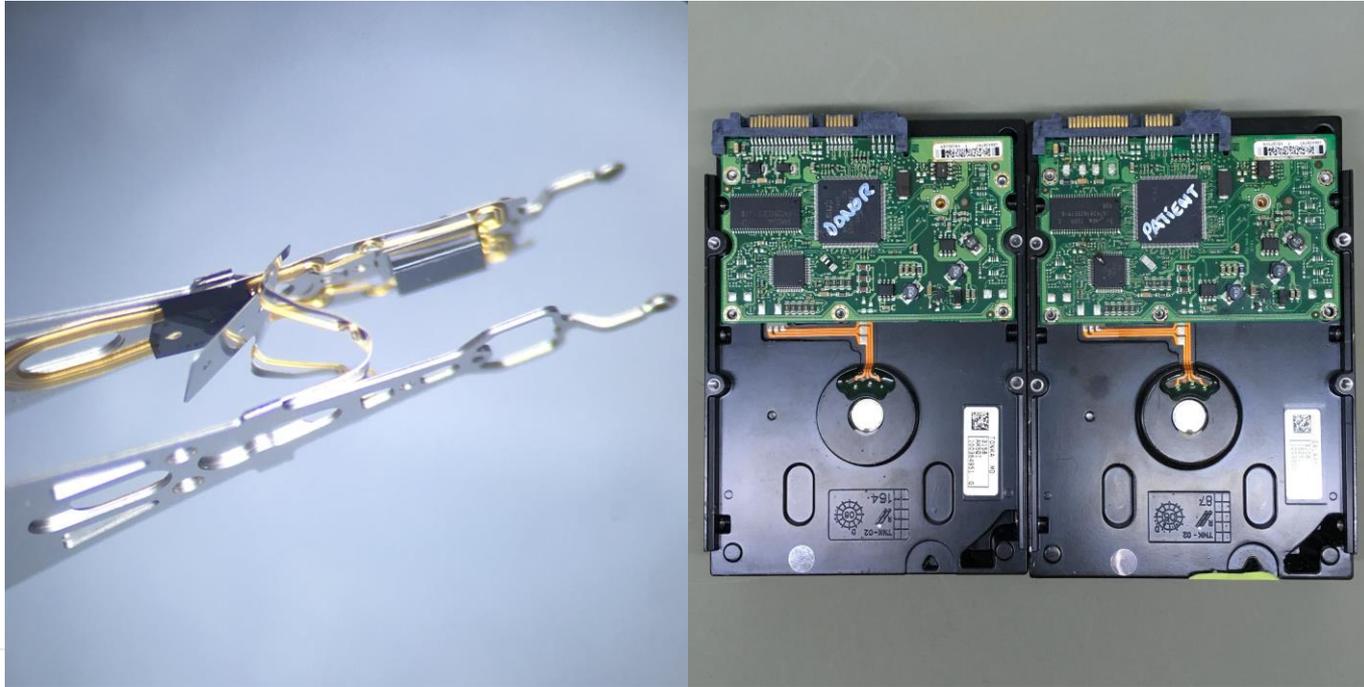
HDD



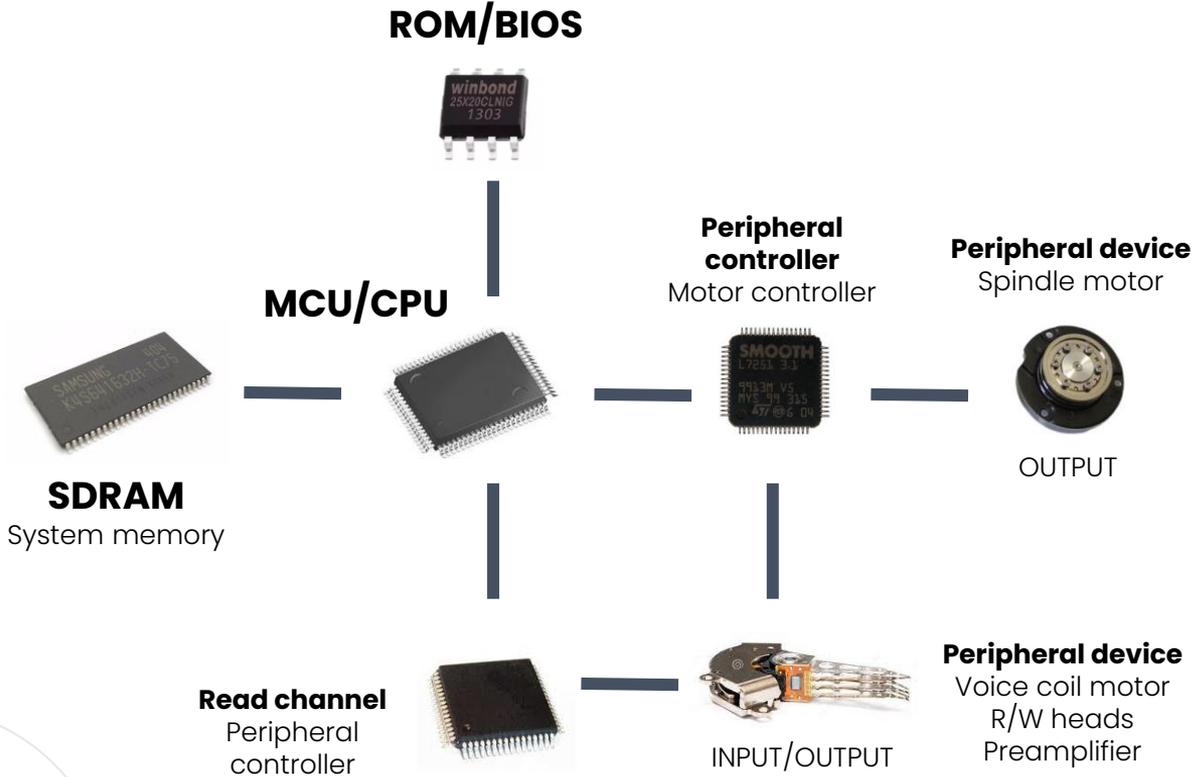
HDD (helium drive)



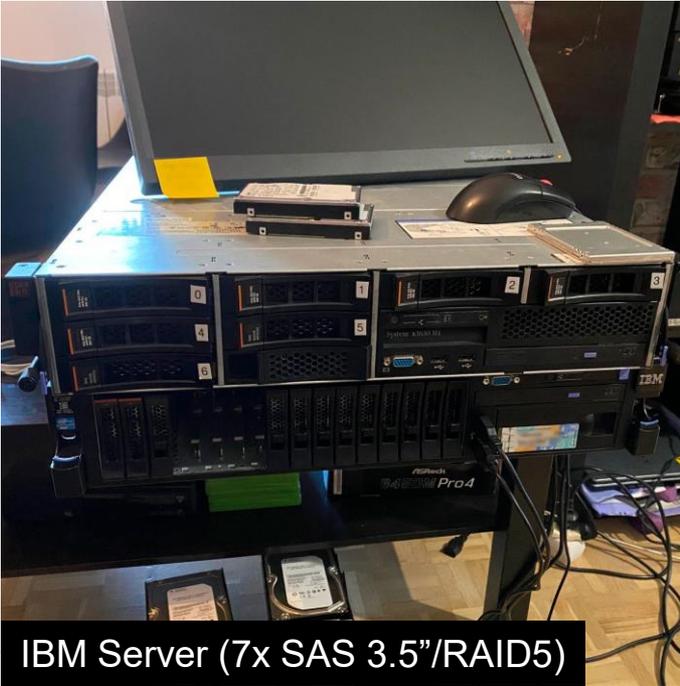
HDD (head swap)



HDD (head assy/PCB)



HDD (PCB)



RAID

Common cases

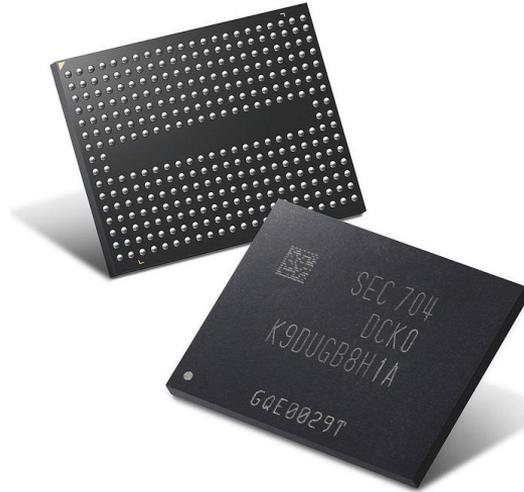
- Mild media issues
- Electronic, controller and/or memory degradation
- Physical

Problems

- No support
- Encrypted drives (SED)
- Scarcity of old drives
- Searching for donors



SSD



Chip-off method

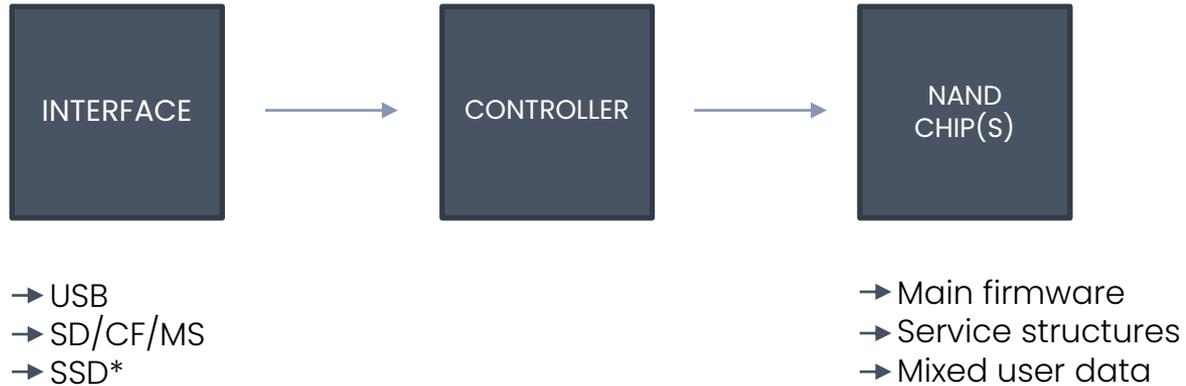
Flash devices



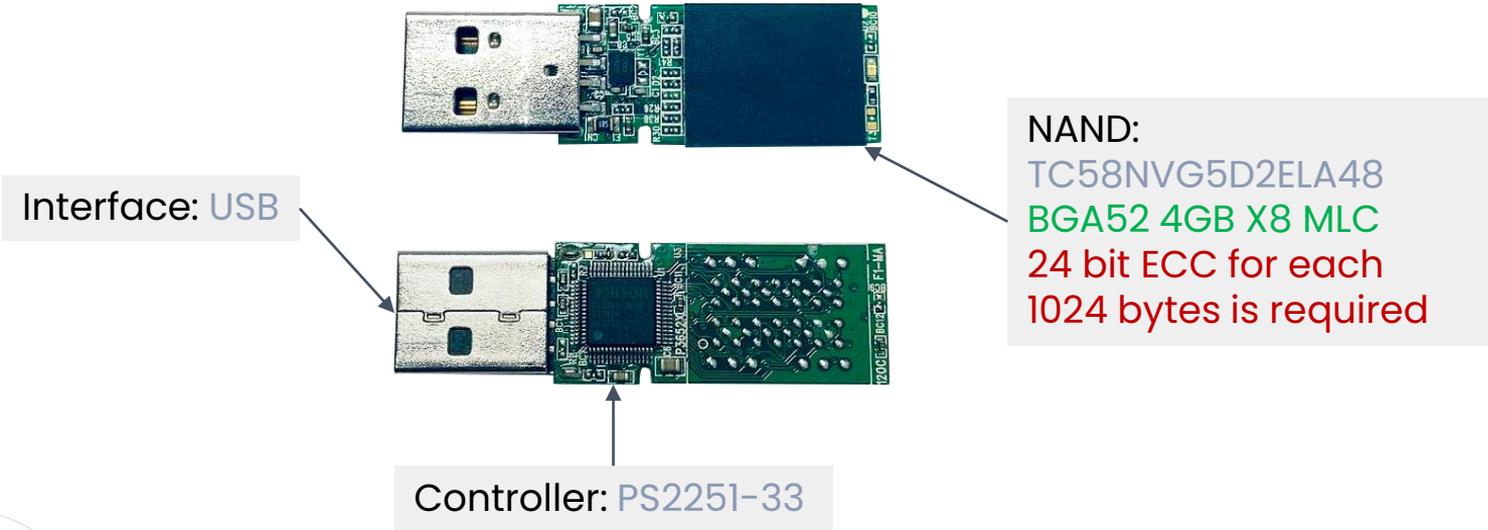
USB flash drive



1. Flash device structure
2. Types of chips
3. How data is stored
4. Bit errors
5. Encryption in flash drives
6. How this is usefull in DFIR
7. Steps during recovery



1. Flash device structure



SMT: surface mount technology

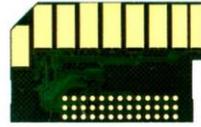
Source: ACELab®



USB

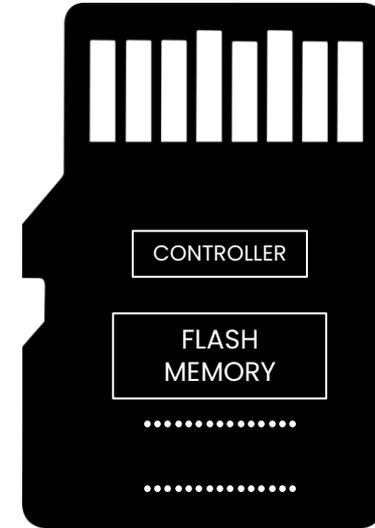


microSD



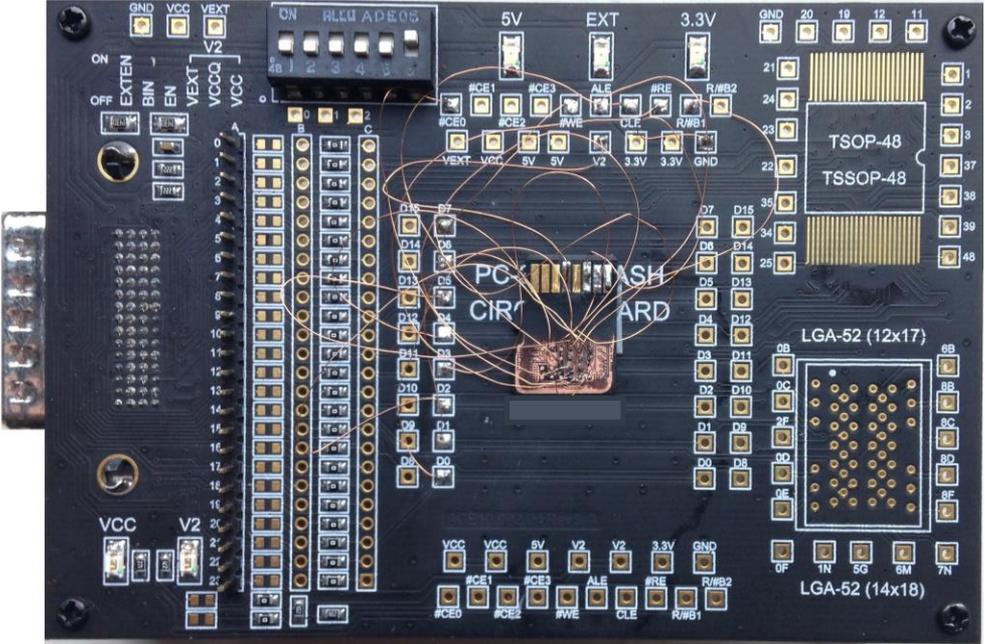
SD

- Unified structure of the NAND flash device
- All components integrated into a single chip



Technological pins

Monolith



Data recovery: NAND protocol

SLC: Single Level Cell

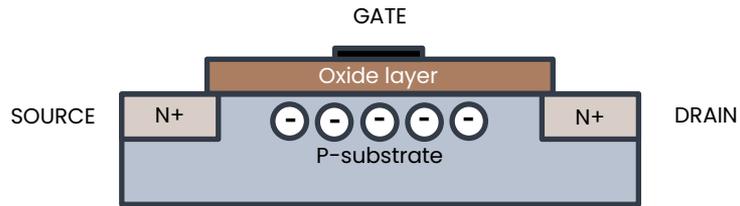
MLC: Multi Level Cell

TLC: Triple Level Cell

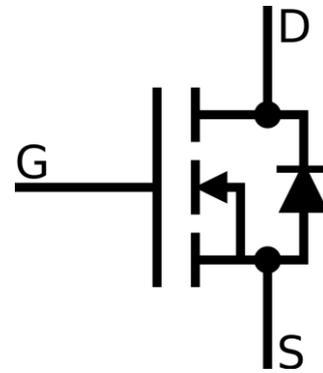
QLC: Quad Level Cell

SLC	0	1														
MLC	00	01	10	11												
TLC	000	001	010	011	100	101	110	111								
QLC	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

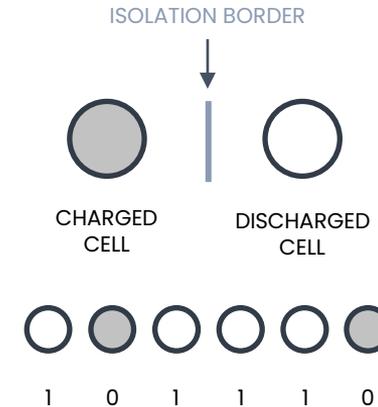
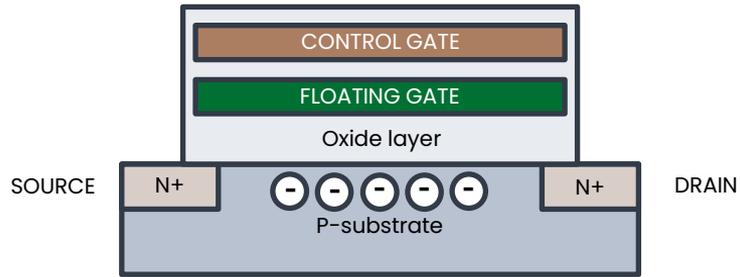
2. Types of chips



ON	1
OFF	0



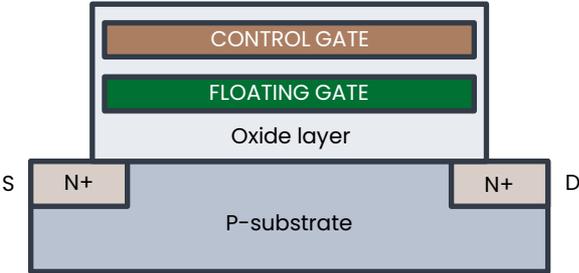
3. How data is stored: the N-mosfet



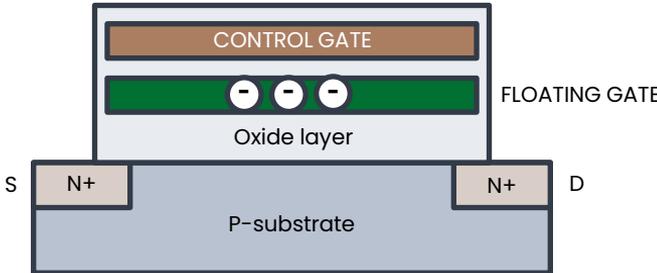
↓ V for Reading
↑ V for Erasing/Writing

- Degradation of isolation border
- Charge leakage

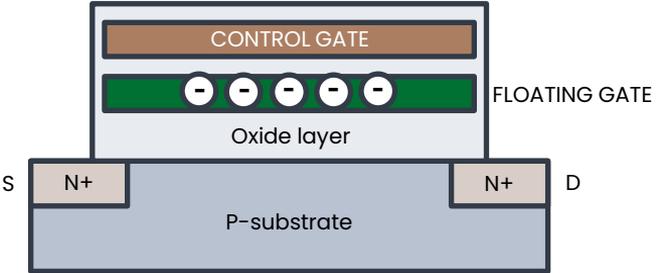
3. How data is stored (SLC)



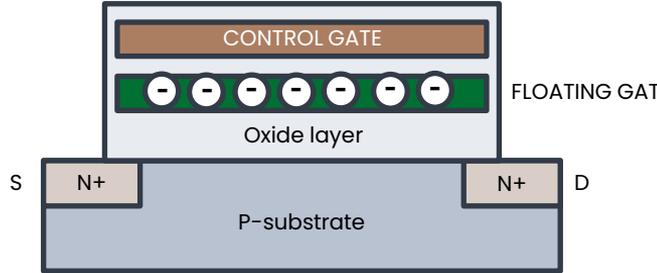
NO CHARGE



LIGHTLY CHARGED



MEDIUM CHARGE

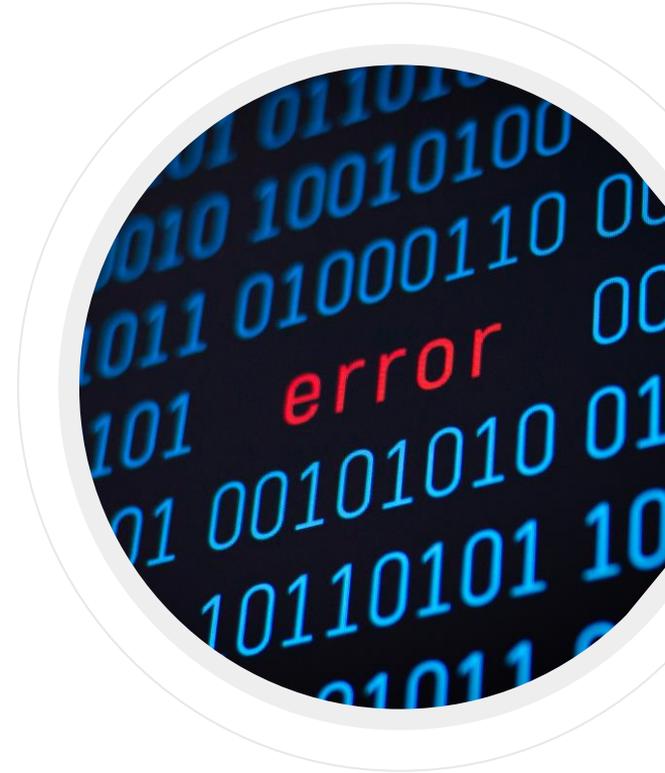


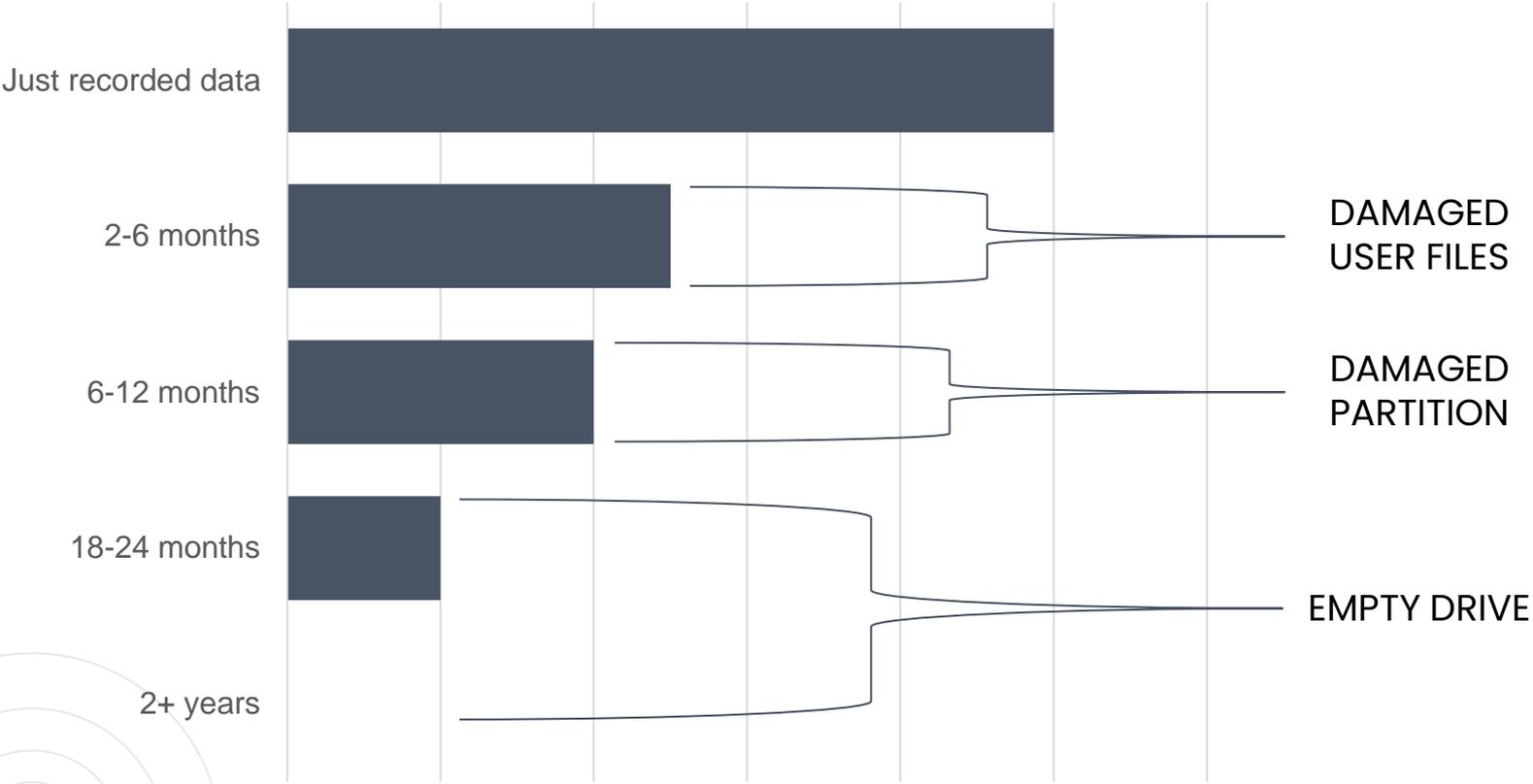
HIGHLY CHARGED

3. How data is stored (MLC)

4. Bit Errors in memory chips

- Bad quality NAND
- Charge leakage
- Damaged Temp/Volt table
- Compression of data
- Small process node
- Wear leveling/Block erase





Charge leakage

DEFAULT VALUES

Temp	VCC
-20°C	3V
20°C	3.3V
80°C	3.6V

NORMAL READING

POSSIBLE SHIFT

Temp	VCC
-20°C	3.3V
20°C	3.6V
80°C	3.9V

READING WITH ERRORS

Temperature/Voltage table

The screenshot shows a software interface for data analysis. At the top, it displays device information: 'Device: Model: SATABURN SB 56BMY1.3.5N (C) Mode: Make copy of object data. Chain - 1 [1] (LBA - 399 452 208 - 1293 417 510)'. Below this, it shows 'Capacity: 953.87 GB (2 000 493 264)' and 'Operation: Clipping - running LBA - 1 001 613 682[1]'. The main area is a large grid with a green background and a grey bottom section. The grid contains numerous small black markers, likely representing errors or data points. A tooltip 'LBA = 1 001 618 800' is visible over the grey section. At the bottom, there are several status indicators: 'LBA map', 'Log', 'Map', 'HEX', 'Structure', 'Status', 'Processes', 'Error register (Port 0)', 'SATA-II', and 'Power 5V', 'Power 12V', '6.8C USB'. The status indicators show various icons and values, such as '5V 0.10 A', '5.2V 12V', '13.4V 0.00 A', and '42.01°C 0'.

Reading errors

Freeze spray



Hot air



Used for error reading

5. Encryption in flash drives

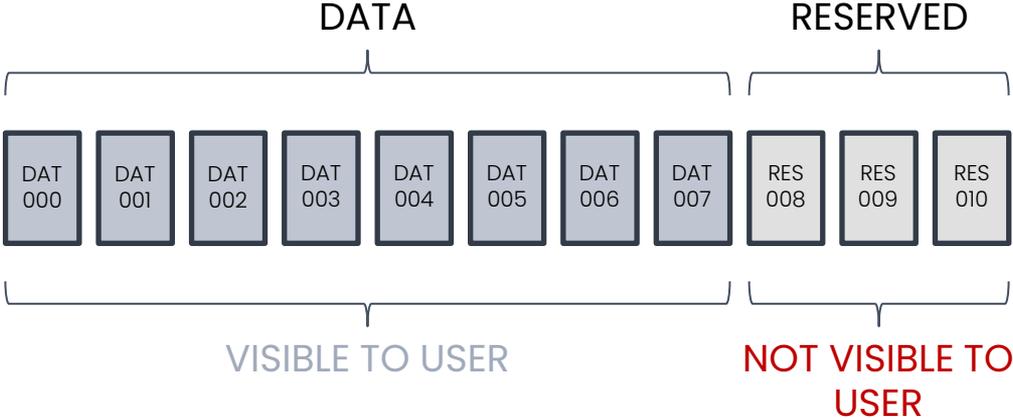
- Hardware encryption by Flash controller
- Software encryption on partition (Bitlocker/Filevault/other)
- XOR



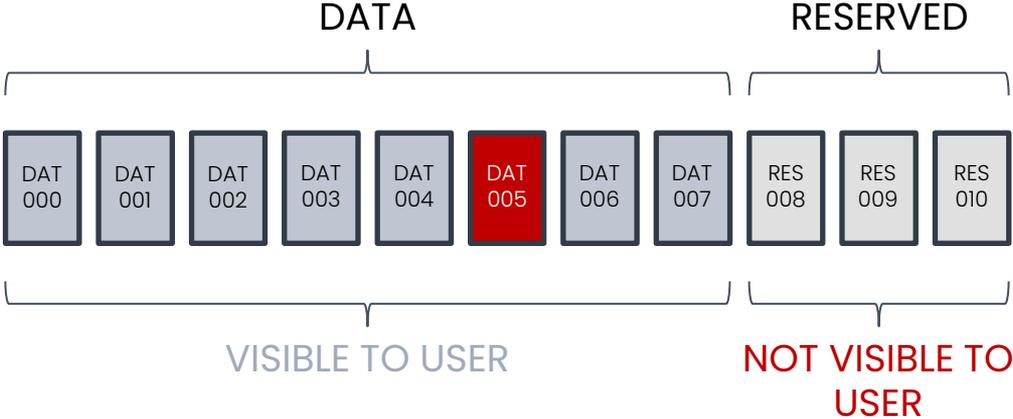
6. How is this useful in DF

- We have access to **unaddressed space** in the memory chip
- Deleted files or formatted drives can be recoverable

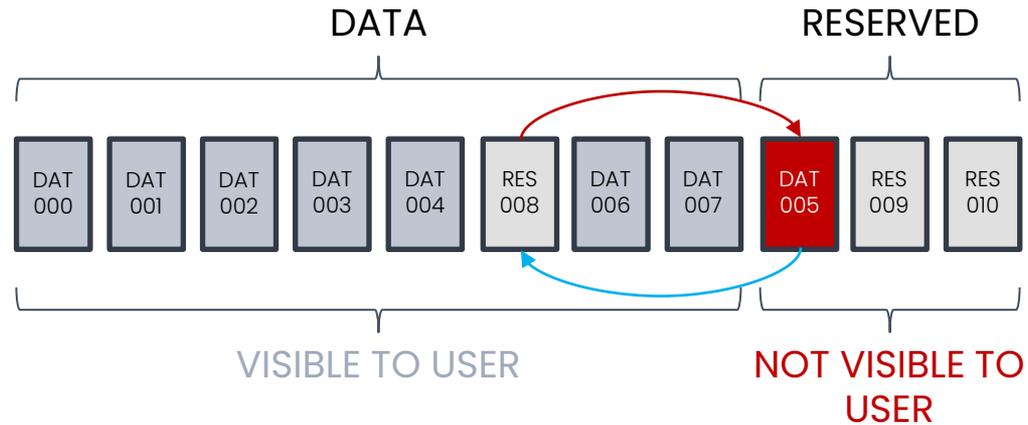




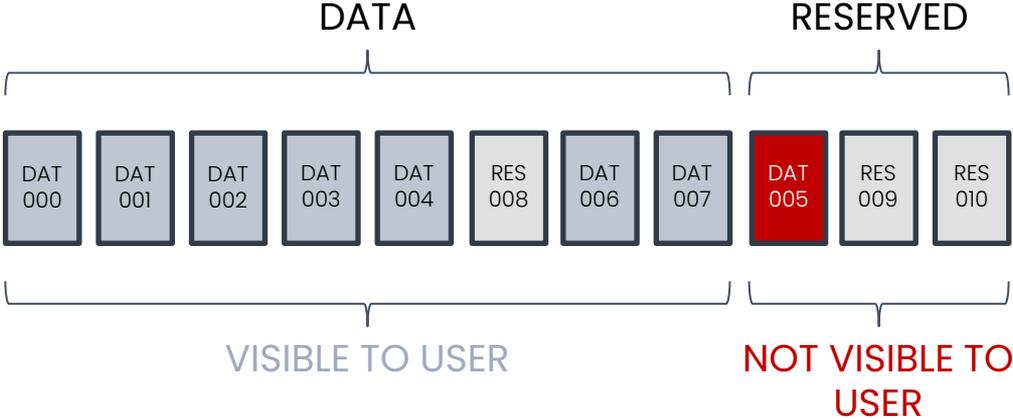
NAND based drive



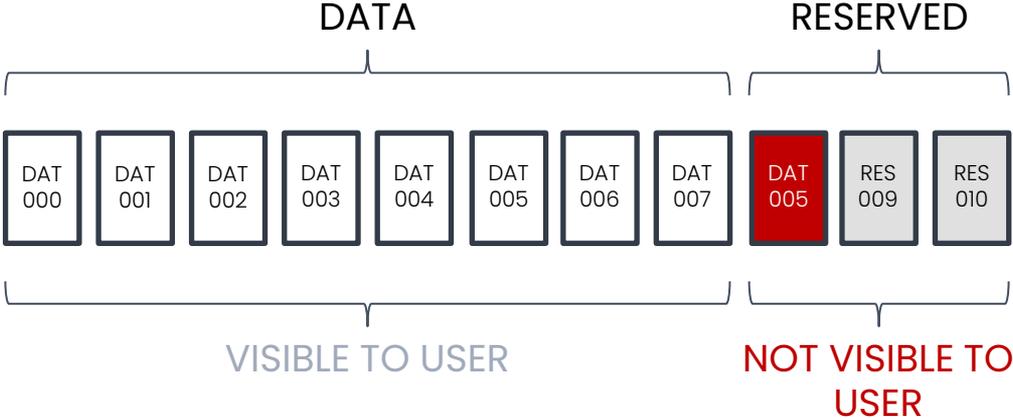
NAND based drive



NAND based drive

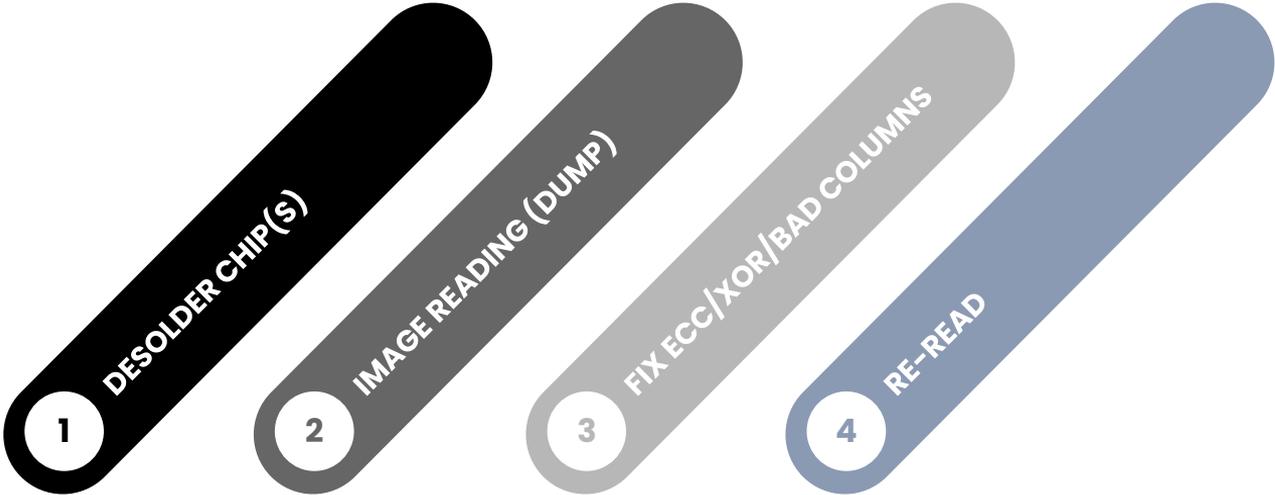


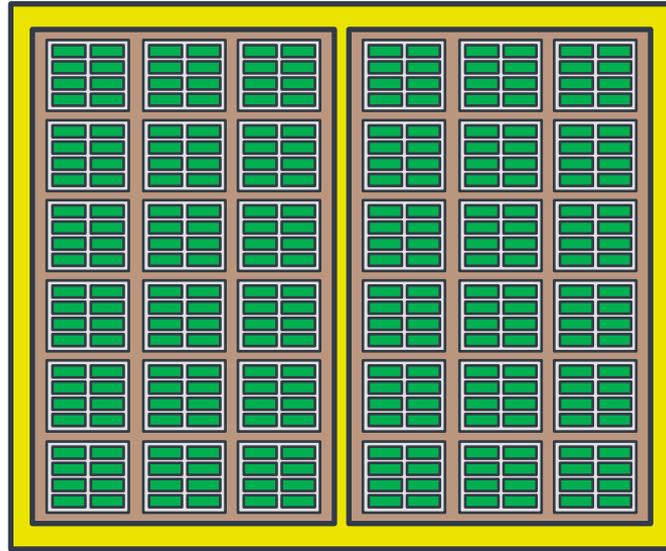
NAND based drive



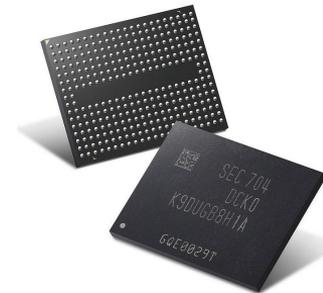
NAND based drive

Chip-off recovery



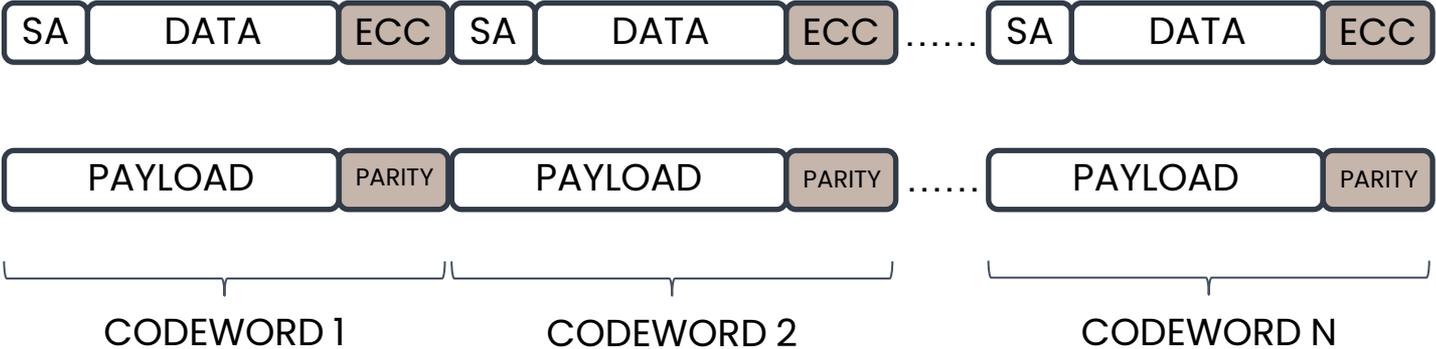


-  BLOCK
-  PAGE
-  PLANE
-  DIE



NAND flash die layout

SPARE AREA, DATA AREA, ECC AREA



CODEWORD: Controller model, Page size, ECC size, Number of Codewords

Page/ ECC structure (payload/parity)

Error Correction Codes (ECC) are algorithms used to add additional information (**parity bits**) to stored data.

These additional bits enable the detection and correction of errors when reading the data.

The **BCH ECC codes** are frequently utilized in flash storage devices. The BCH algorithm is customizable and comes with a specific set of parameters. These parameters are pre-configured in the controller's firmware and vary from one model to another. **If the controller becomes damaged**, the parameter information is lost; however, **ECC checksums persist** within each page of the NAND chip.



ECC

We use a BCH decoder designed to correct errors in data by utilizing the available ECC checksums. This ECC-based error correction algorithm can be implemented on the physical image after it has been extracted

This process is very time consuming

The logo consists of the letters 'ECC' in a bold, black, sans-serif font, centered within a series of four concentric, light gray circles of increasing radius.

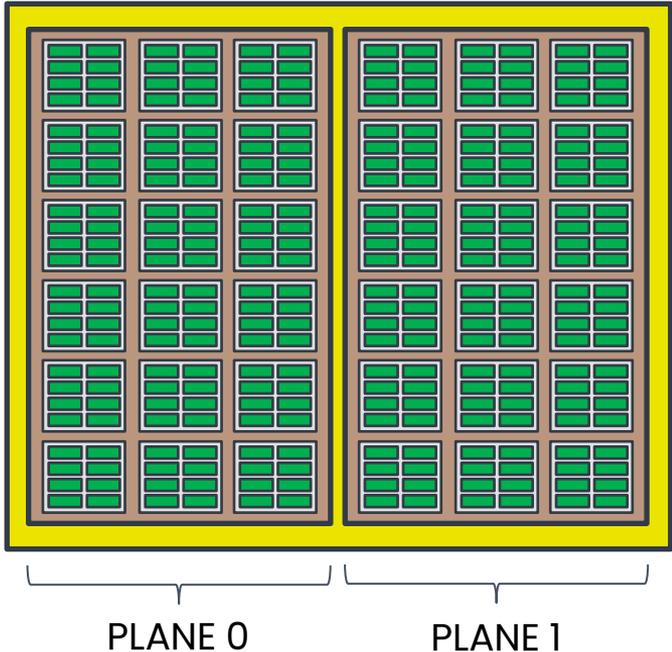
ECC

The screenshot shows the [TEKNIK] DataLab software interface. At the top, there is a toolbar with various analysis functions: XOR analyzer, Data area, Spare area, Data transformation, Page allocation, File system metadata, Find codewords, Codeword analysis, Correct dump, Reread dump, Save ECC map in file, Load ECC map from file, and Clean ECC map. Below the toolbar is a workflow diagram with three blocks: 'Reader' (0), 'Phy image' (0), and 'ECC' (0). The 'ECC' block has a warning icon. To the right of the workflow is a 'Parameters' panel with a search filter and several sections: 'Element' (Name: 0), 'Dump' (Length: 4529848320, Automatic str.: checked), 'ECC corrector' (Power: Off, Code words: Phison(P5)\PS2251-50-F., Page size: 8640, Use buffer: checked, Bytes rotate: unchecked, ECC map: unchecked).

Autodetect or search in database (if lucky)

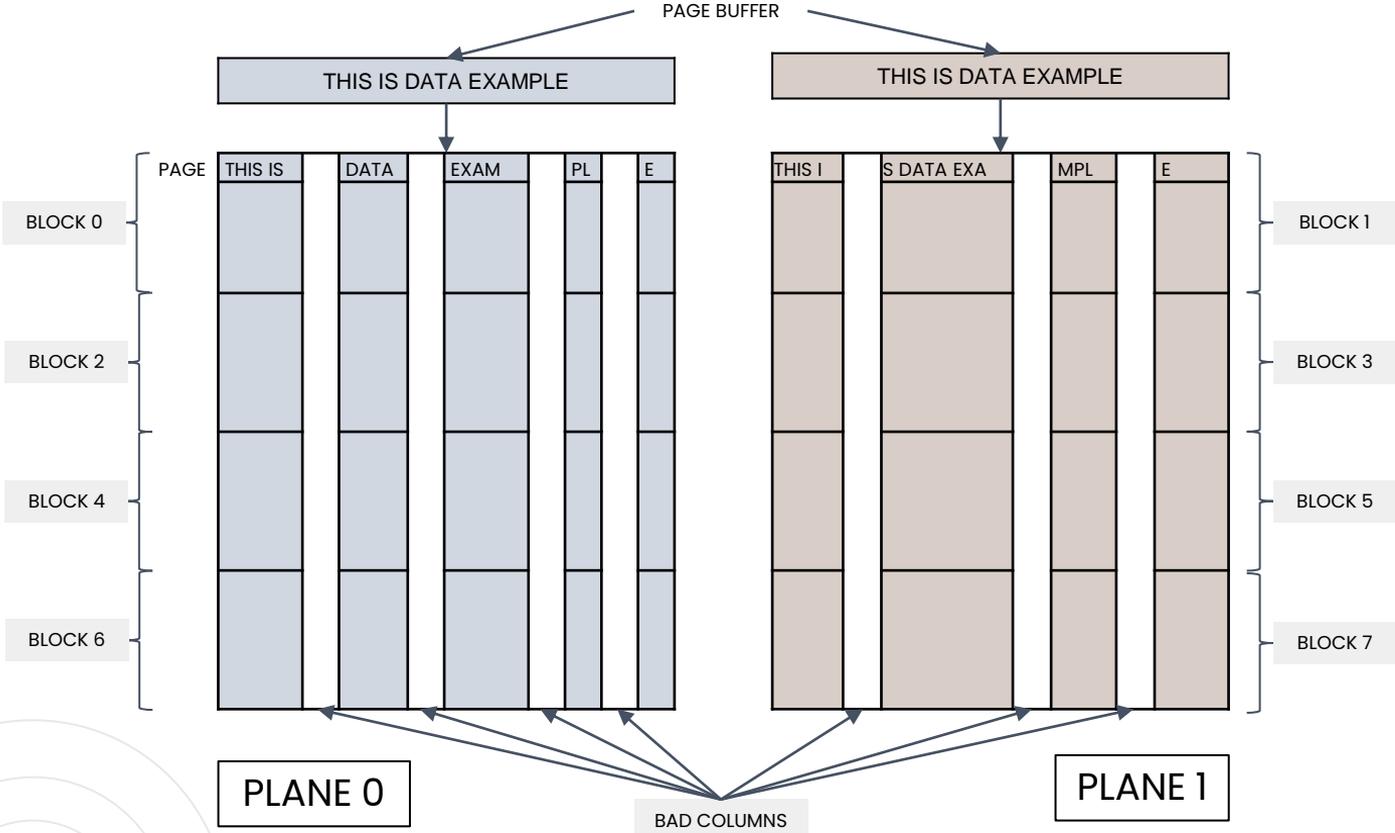
Phison controller

Page of flash memory

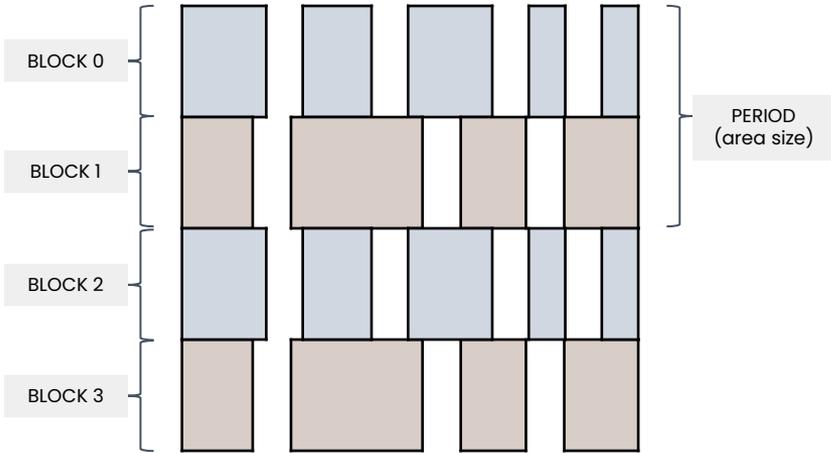


-  BLOCK
-  PAGE
-  PLANE
-  DIE

NAND flash die layout

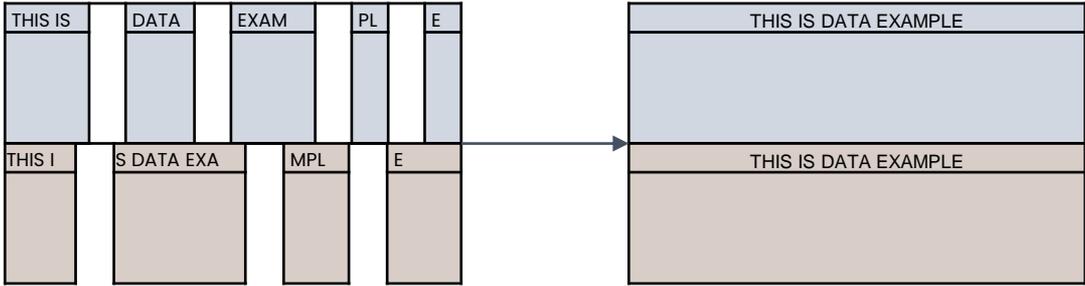


Bad columns



PHYSICAL IMAGE

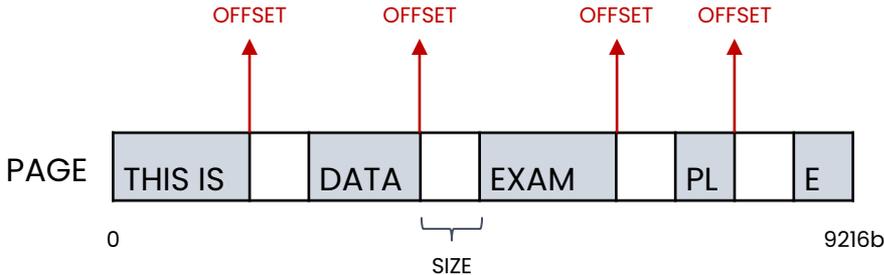
Bad columns



PHYSICAL IMAGE WITH BAD COLUMNS

PHYSICAL IMAGE WITHOUT BAD COLUMNS

Bad columns

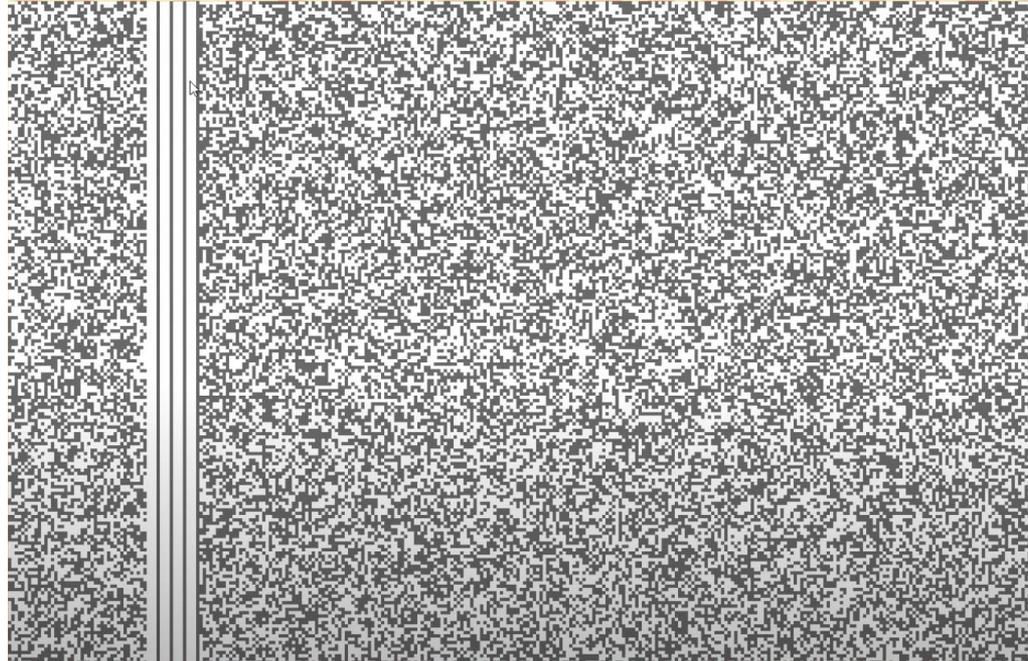


It's necessary to determine the location, size and number of Bad columns within one page of each plane. Location is offset of bad column from the beginning of page. Bad Column size expressed in bytes. Number determines number of offsets from the beginning of page, that must be added in order to cut Bad Column defects.

Bad columns



Bad columns



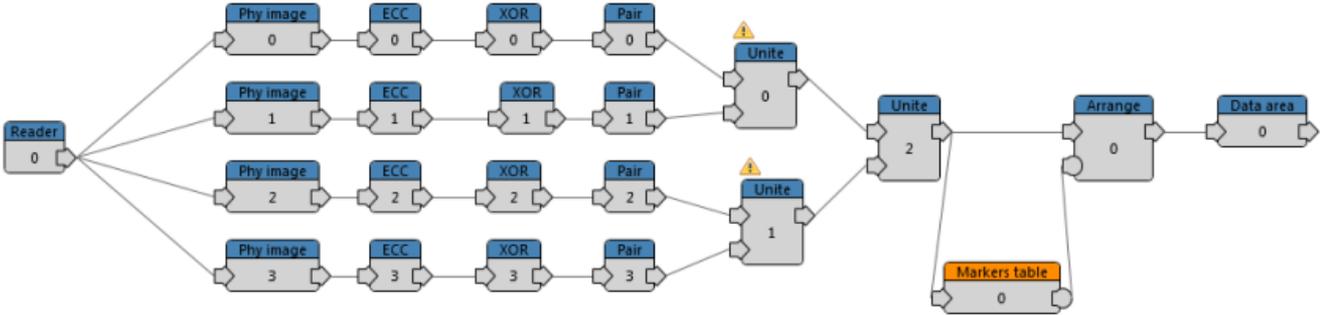
Bad columns

Modern flash controllers employ scrambling algorithms during the data recording process in flash memory. A typical implementation of a scrambler involves the generation of a unique scrambling (XOR) key by the controller, which is then combined with the data through an XOR gate/operation. This occurs for every data block/page before it is written into the flash memory.

Generally, the XOR key is specific to a particular controller model. However, there are instances where similar controllers may use different XOR keys, and different controllers may use the same key. Unlike encryption, **scrambling is not implemented for security purposes**; instead, it eliminates data patterns that modern NAND chips struggle to store effectively due to charge leakage from adjacent cells. **It serves as a data-integrity measure** rather than a security measure.



XOR scrambling



Assembly

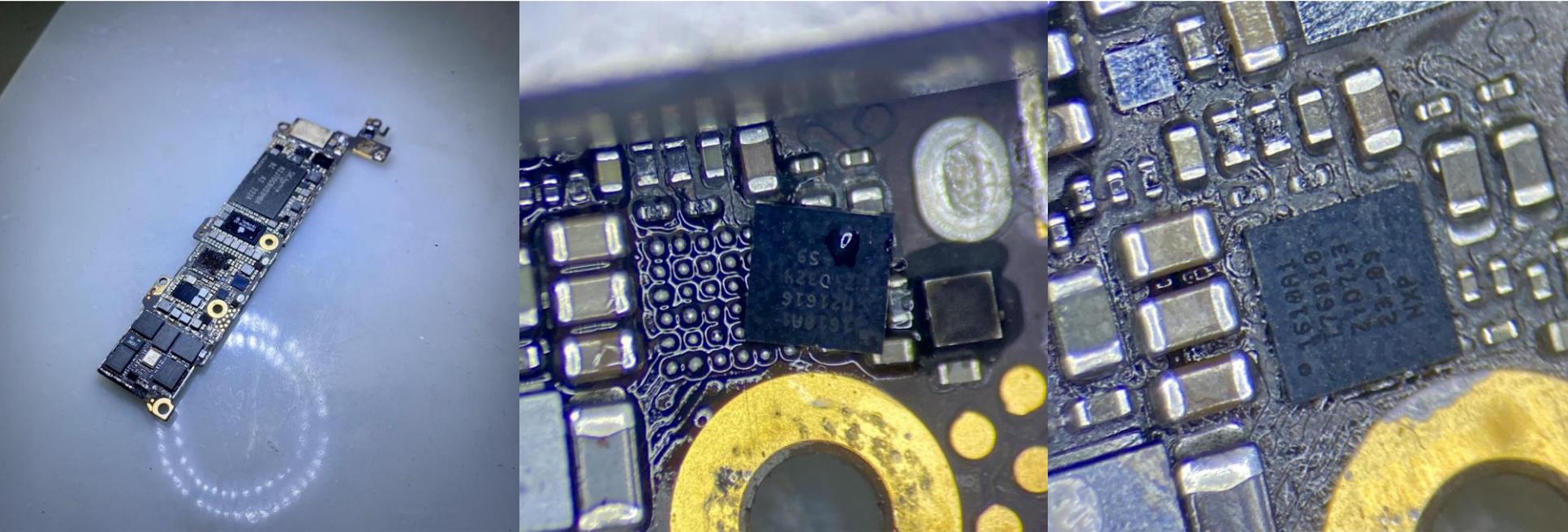


3

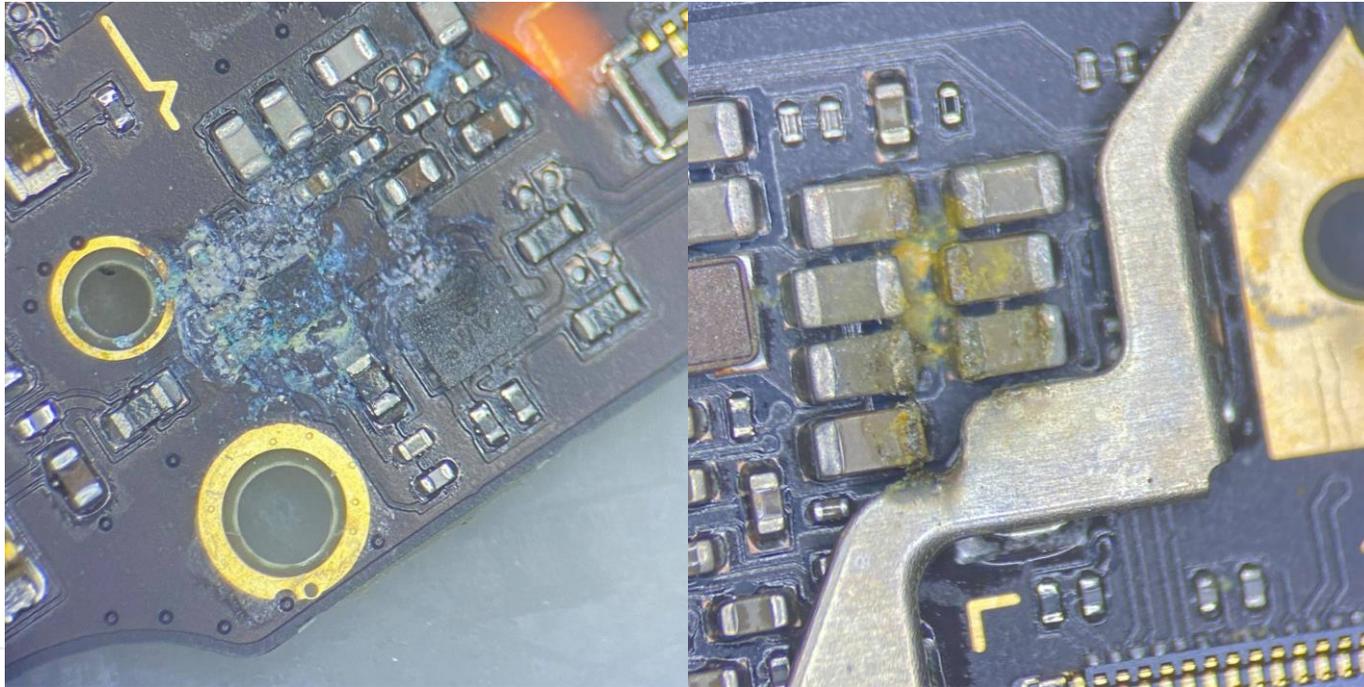
Extra Resources

Phones





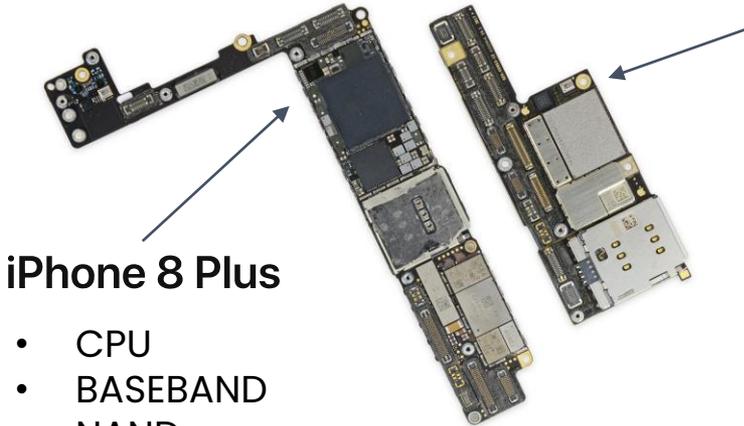
iPhone SE (liquid damage)



Xiaomi Mi A2 (liquid damage)



Xiaomi Mi A2 (EDL mode)



iPhone 8 Plus

- CPU
- BASEBAND
- NAND
- EEPROM

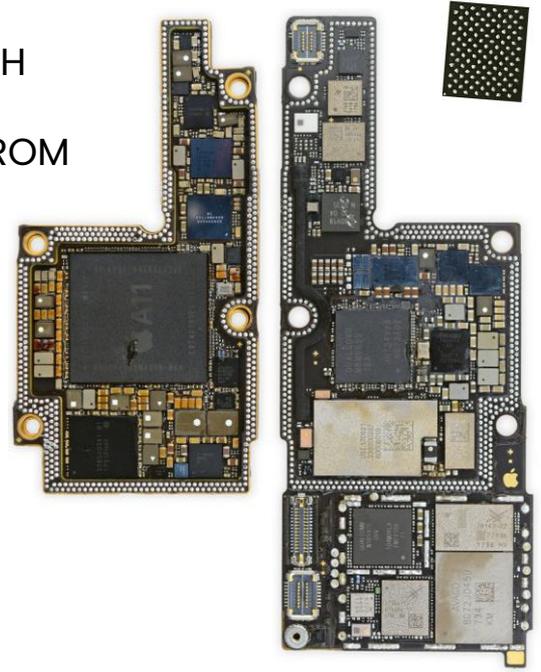
iPhone X

Bottom board:

- NFC
- WIFI/BLEETOOTH
- BASEBAND
- BASEBAND EEPROM

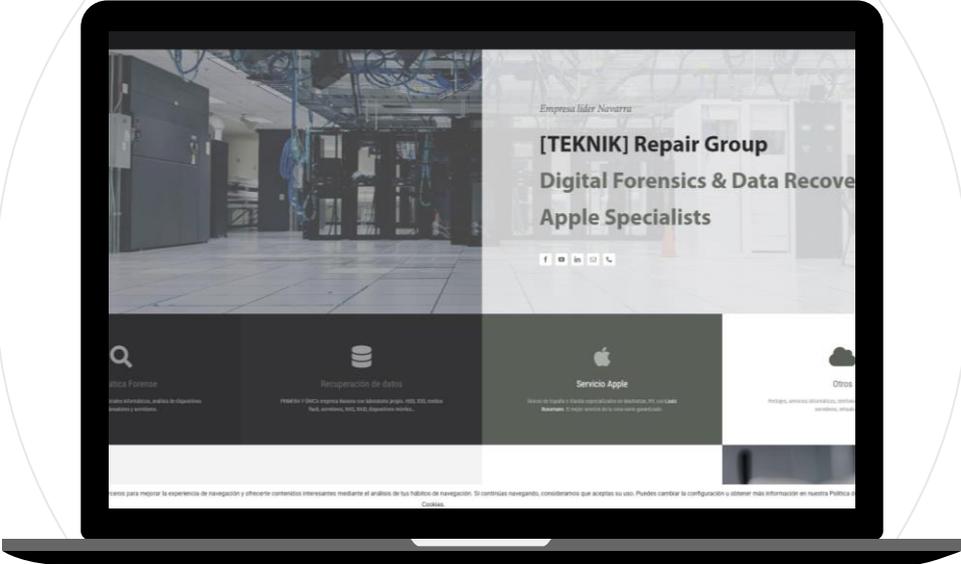
Top board:

- CPU
- LOGIC EEPROM
- NAND



Chip swap

Laptops



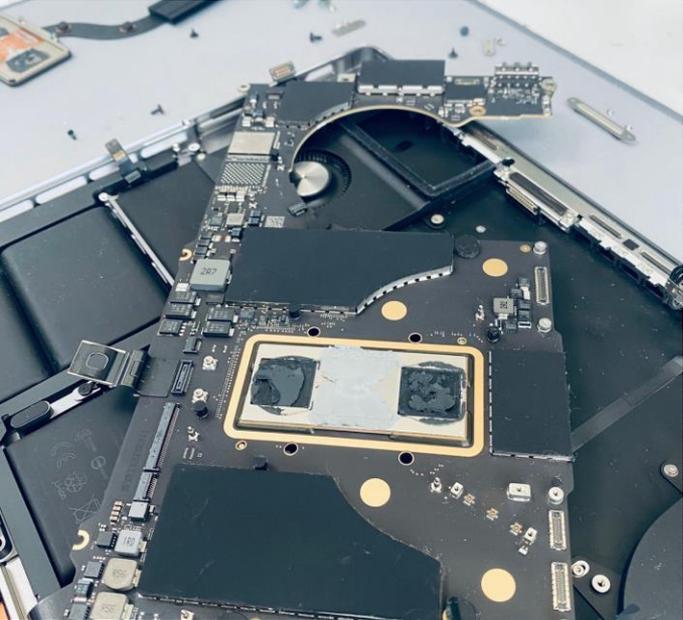


MacBook 13" 2018 liquid damage

MacBook



MacBook 12" 2015 | Dead CPU



MacBook Pro M1 2021

MacBook

Honorable mention: Louis Rossmann

- Pioneer right to repair activist
- **Joe Biden** signed an executive order directing the Federal Trade Commission to draft new regulations limiting device manufacturers' ability to restrict independent repairs of their products.
- In 2022, got the first bill passed in Colorado. In 2023, we saw two more bills pass in Minnesota & California.
- Has been supported by millions of people and the entire electronics community, including Apple co-Founder: **Steve Wozniak**





Thanks!

Any questions?

You can find me at

info@teknik.eus