Internet of Tchotchke*

*urban dictionary: *A small piece of worthless crap, a decorative knick knack with little or no purpose*

Hack.lu 2015 – Paul Rascagneres

Me: Paul Rascagnères
Twitter account: @r00tbsd

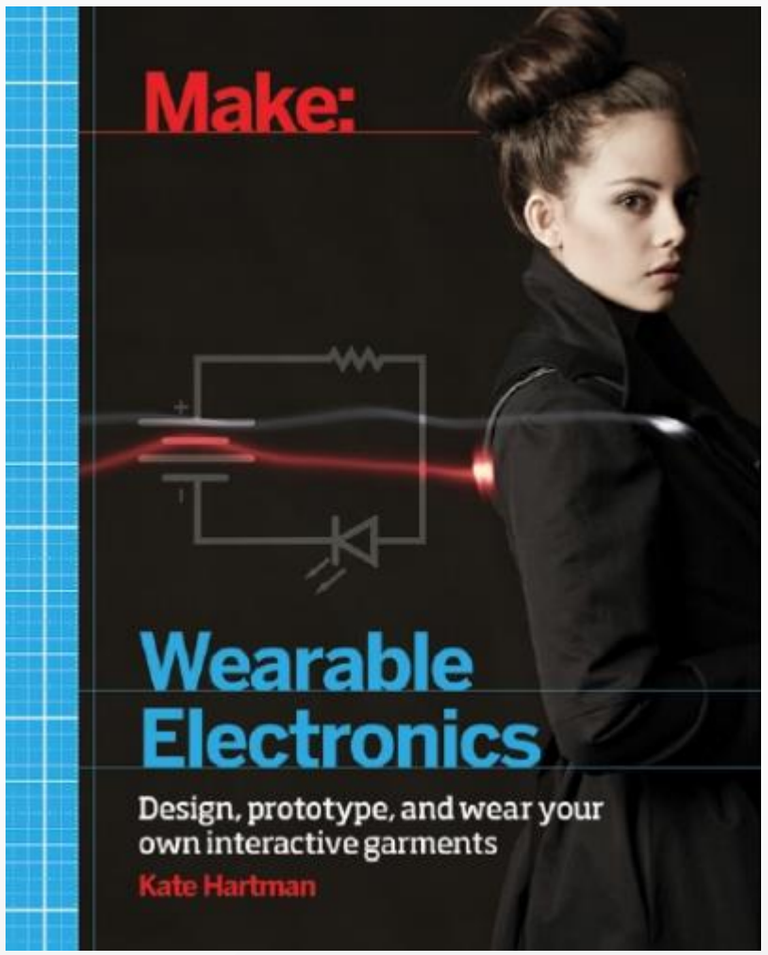
Senior threat researcher at CERT SEKOIA
Author of the French book
"Malwares - Identification, analyse et eradication"
(ISBN: 978-2746079656)

Co-Organizer of Botconf (2-4 December – Paris)

Located in our offices in Luxembourg & Paris

The beginning…

My hack.lu 2014 CTF prize:

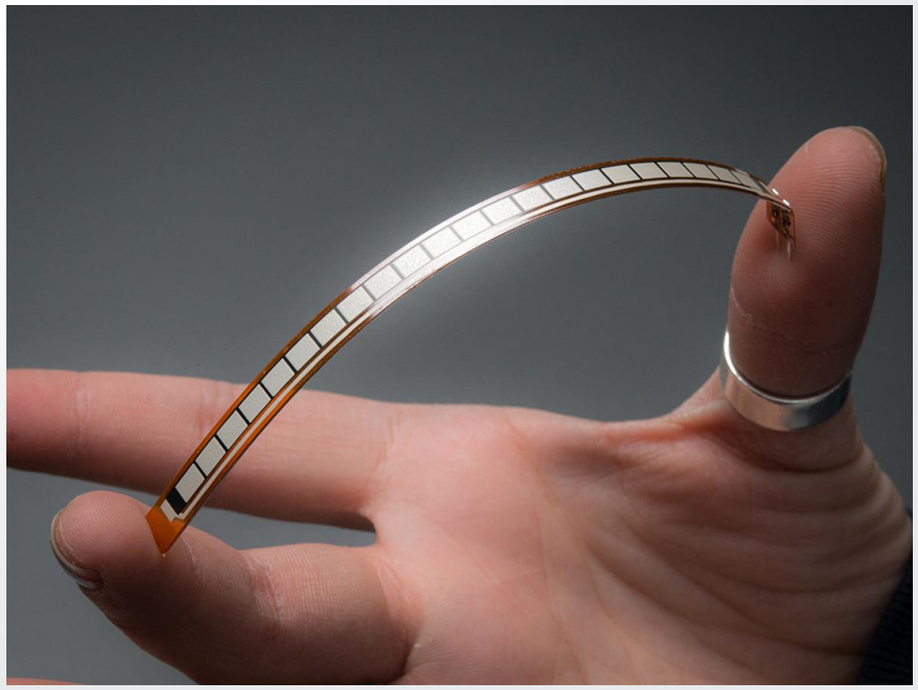


Thanks to @adulau, a new world appear to me…

Amazing sensors

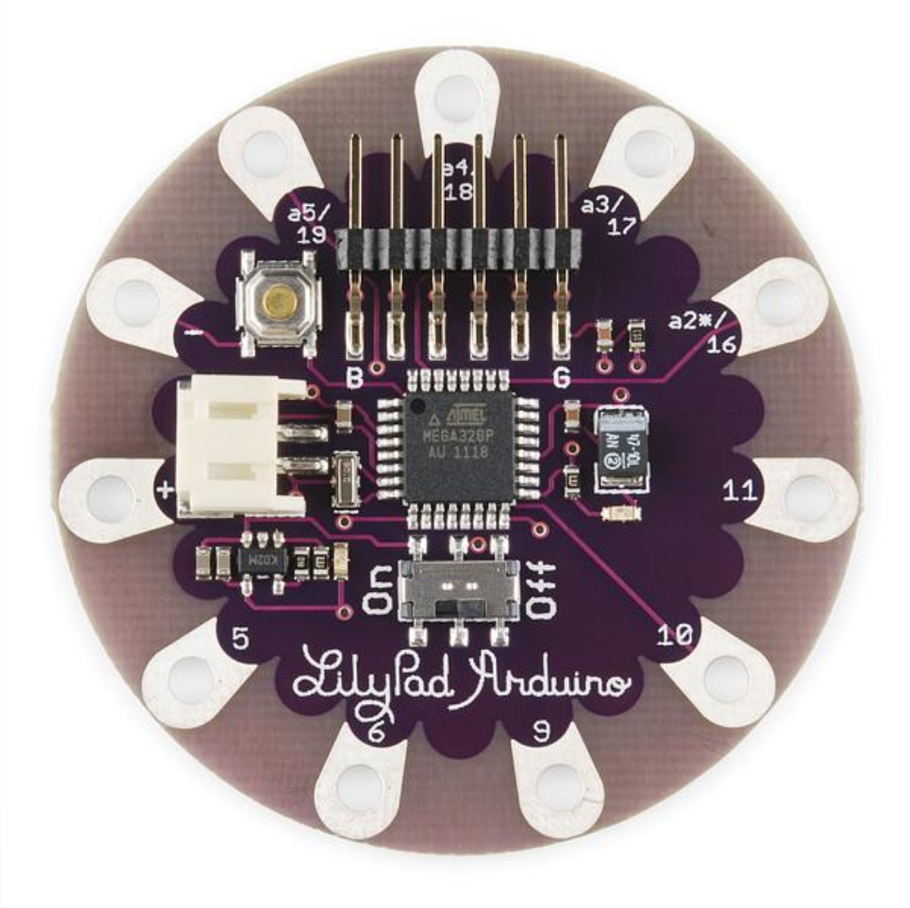
Humidity & temperature

Flex



Amazing Arduino

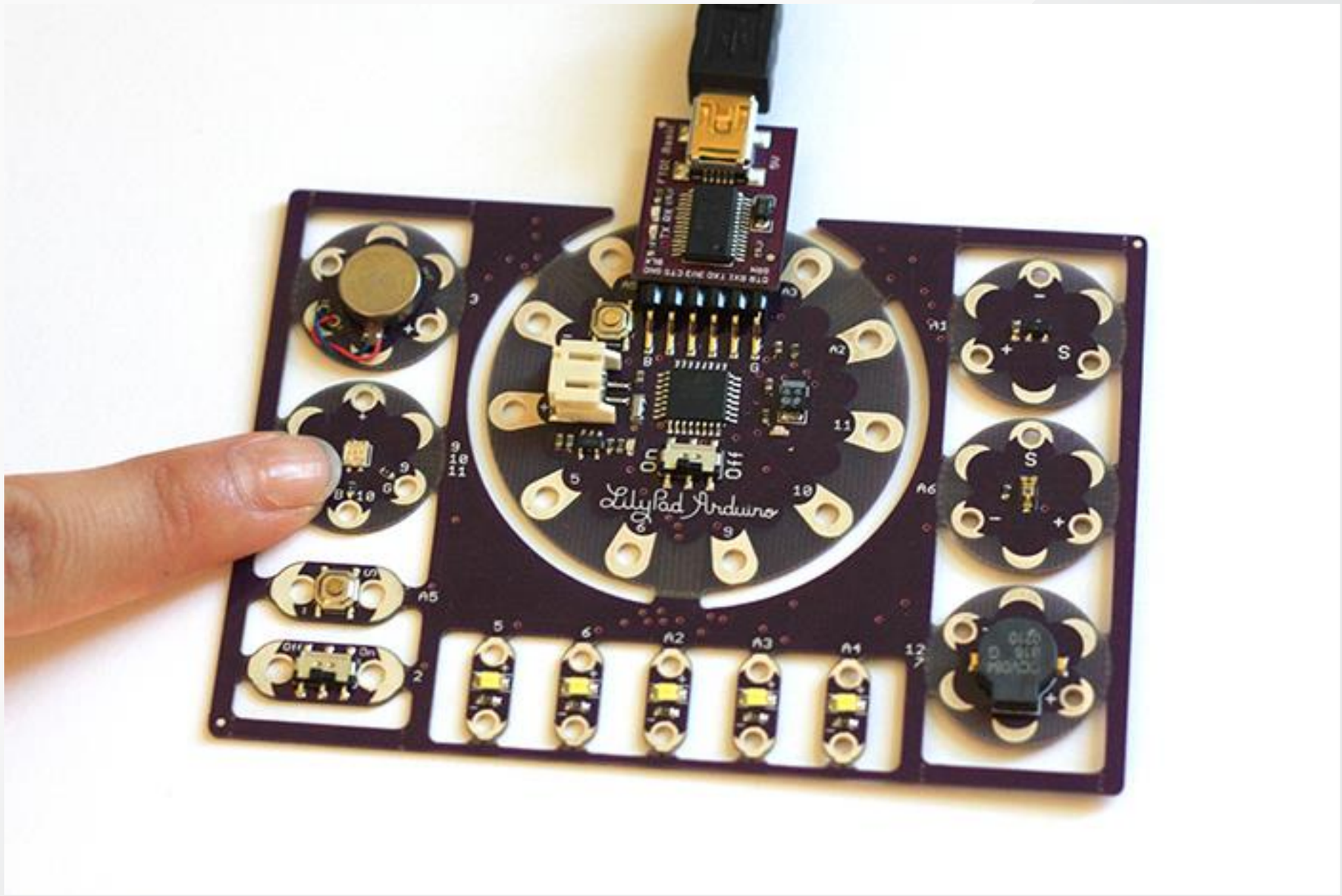
Lilypad



Amazing Arduino

- Button
- Buzzer
- Resistor

...



Amazing misc

Conductive thread

Amazing connected device
Because it better when it’s connected!!

Bluetooth module

I mixed all the elements…

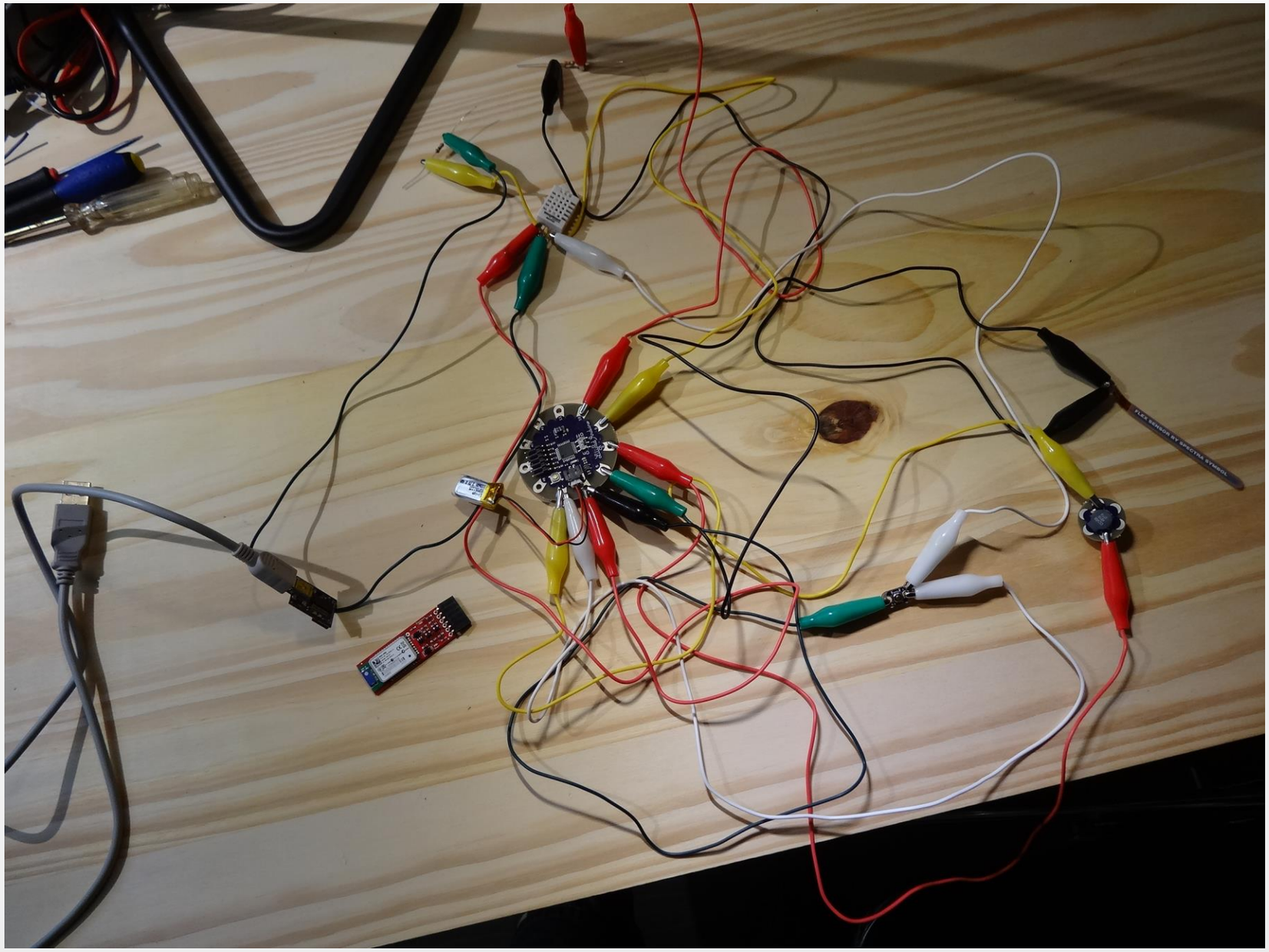
…

and I got a positive emulsion

…

!!! THE IDEA !!!

The first POC



The first connected underwear of the world

SHIP'R'ED

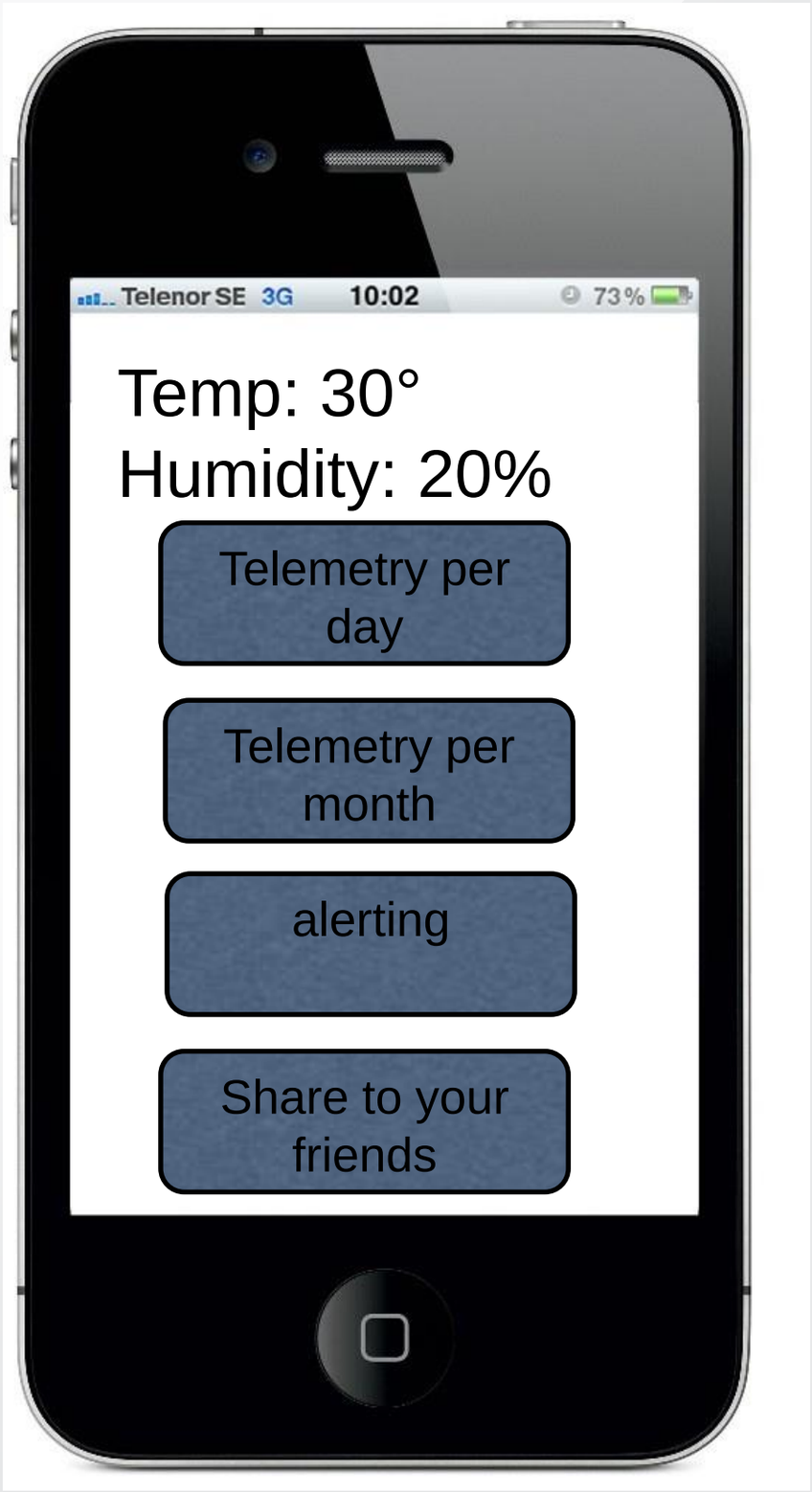
- Humidity & temperature sensor
- Flex sensor to have telemetry of the underwear usage
- Buzzer to alert the user
- Radare2 support
- Bluetooth

Limited hack.lu edition!!

The first connected underwear of the world

- GSM app*
- Cloud telemetry*
- Share to your friends*
- More more more

*Not ready yet

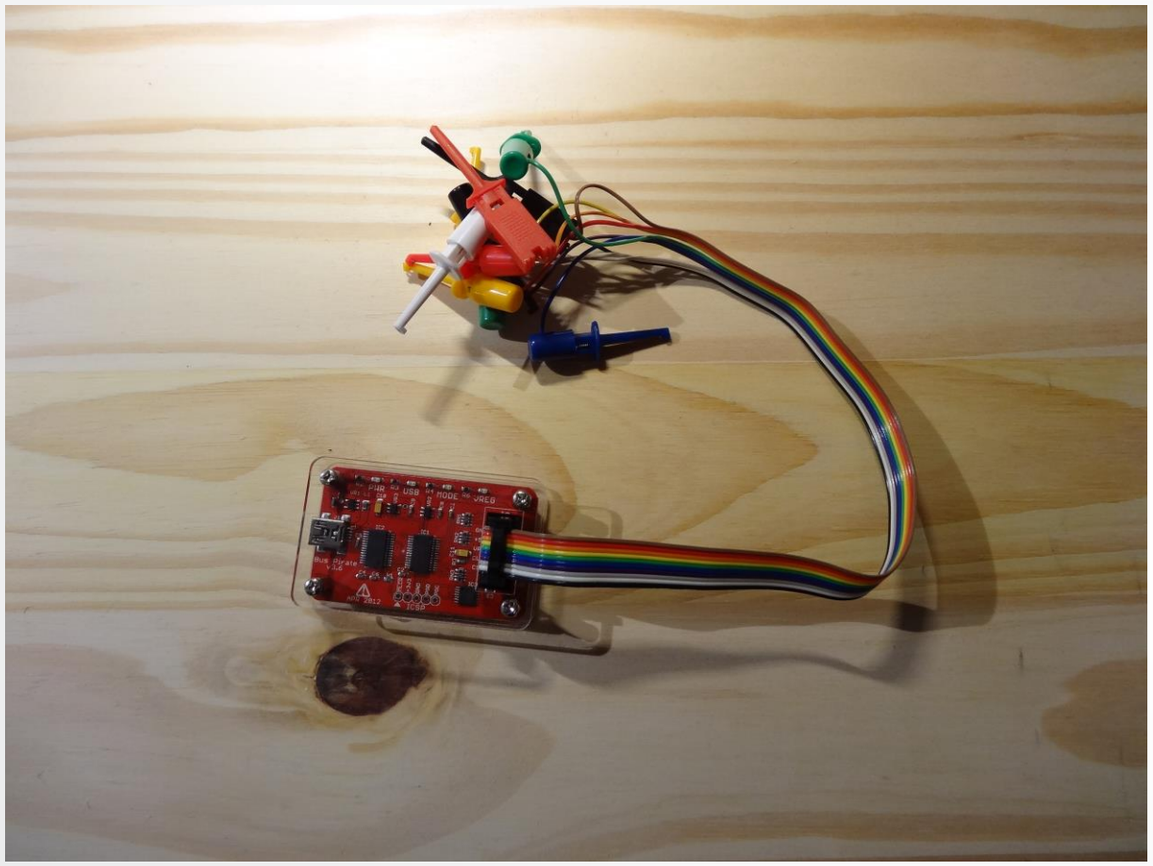


seriousLevel ++;

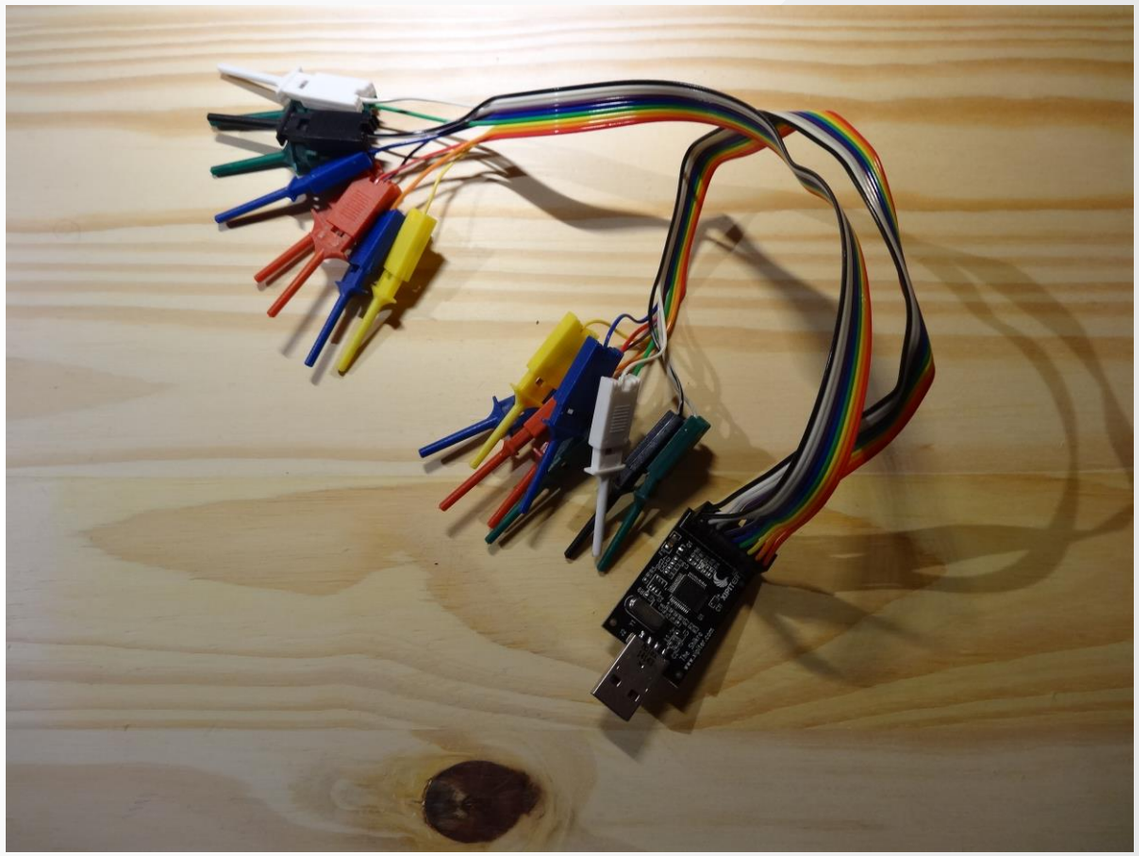
Hardware and Embedded system reverse engineering

Part 1: mandatory hardware

Bus pirate | Shikra (and not Shakira)

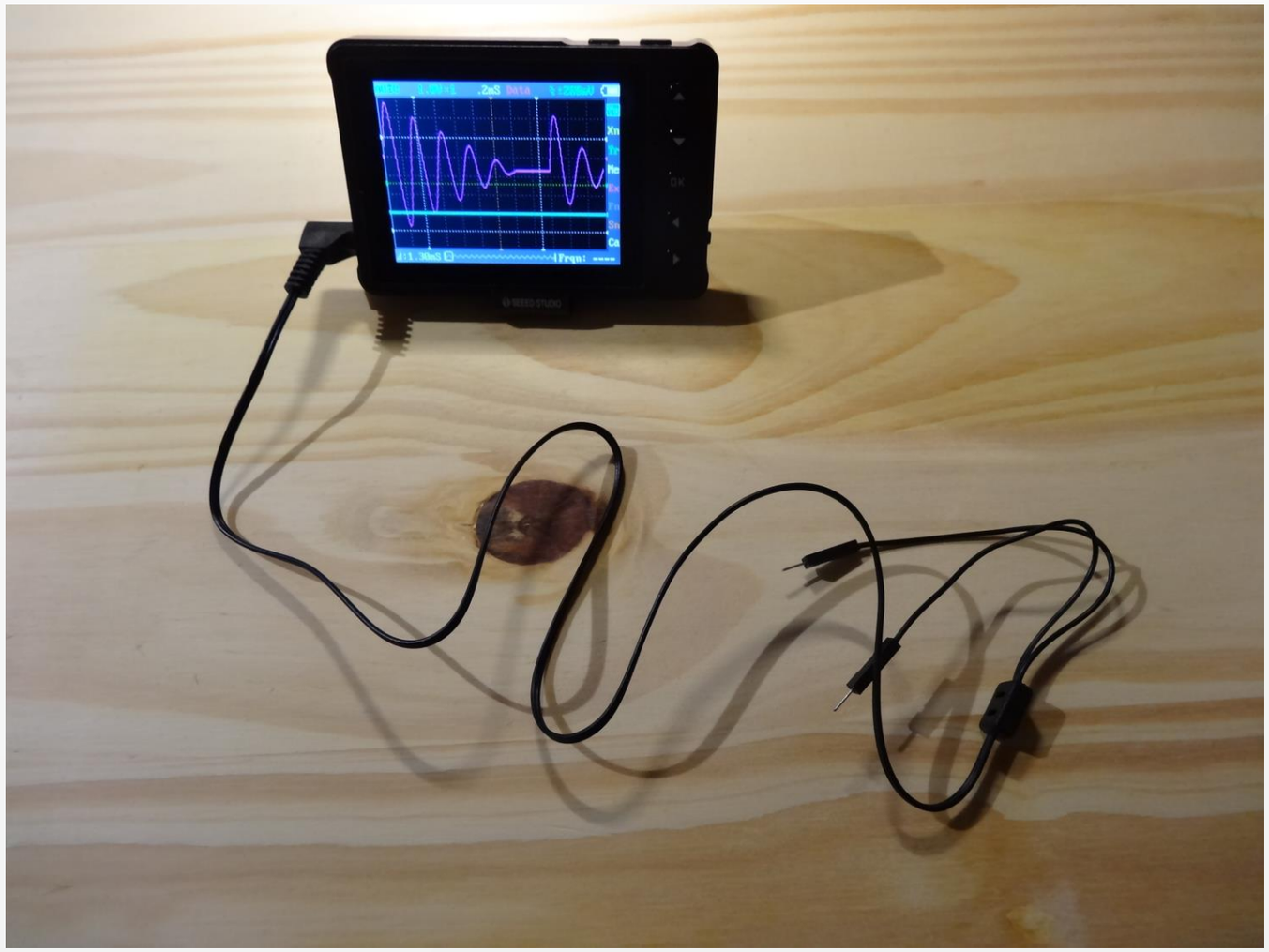


- 1-2-3 wire
- I2C
- SPI
- JTAG
- Asynchronous serial
- …

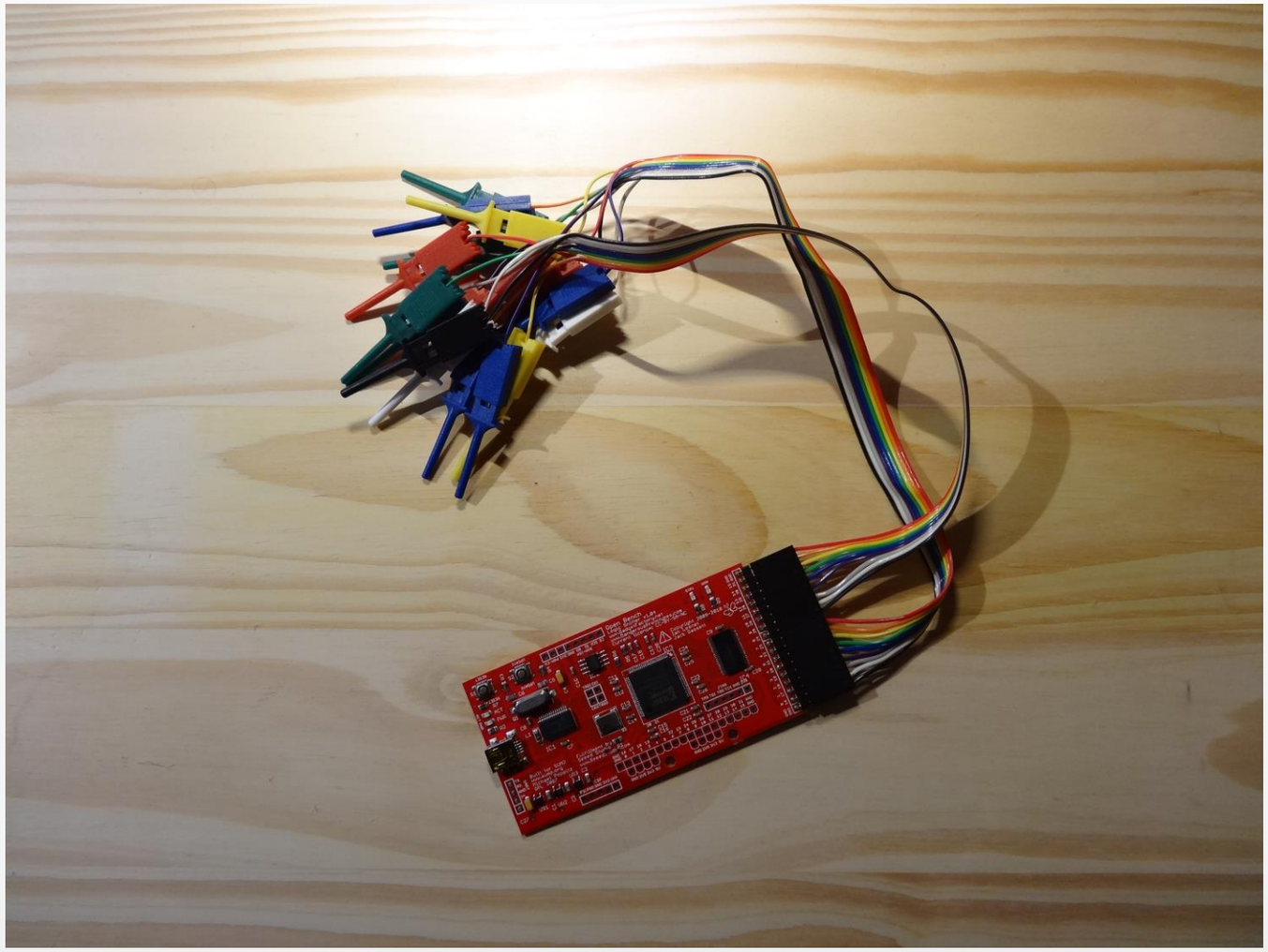


- I2C
- SPI
- JTAG
- UART
- GPIO

Oscilloscope: Nano DSO

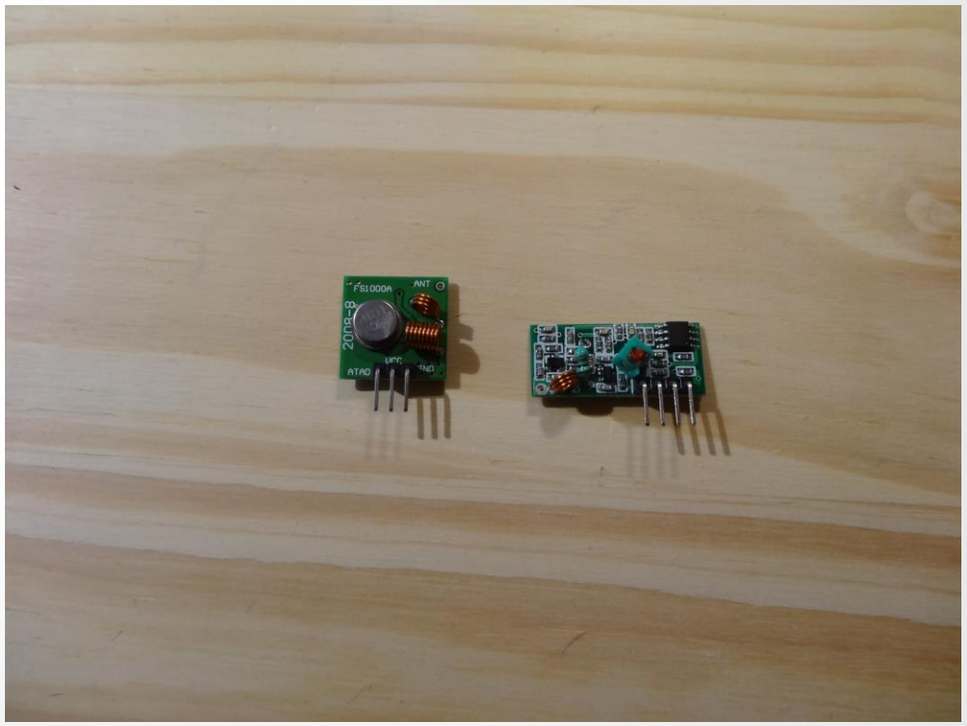


Open Bench Logic Sniffer



Solder station + misc

HackRF | 433Mhz receiver/transmitter



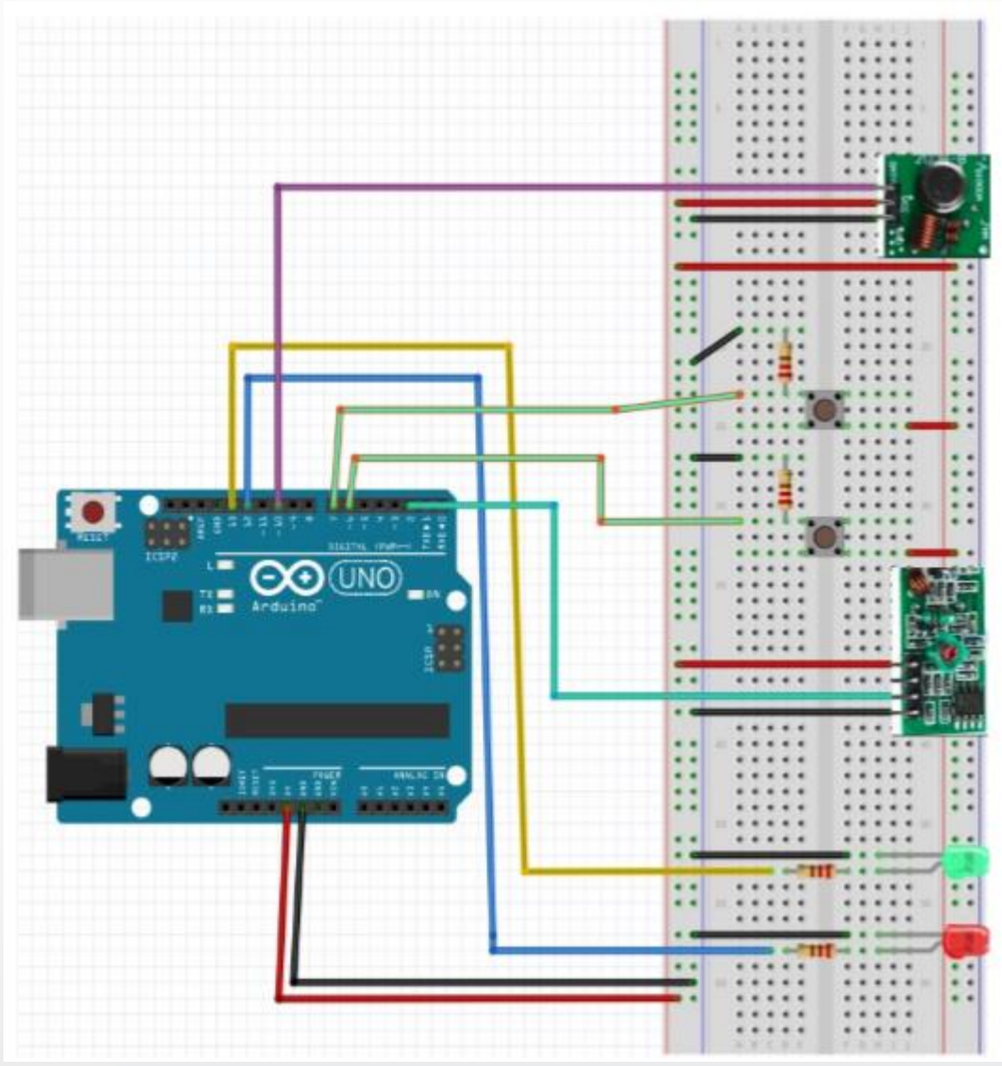
Part 2: few hacks

RF hacking case 1: wireless door bell

Complete analysis: https://bitbucket.org/rootbsd/433mhz-ask-signal-analysis/

RF hacking case 1: wireless door bell

Magicbox Arduino schema



RF hacking case 1: wireless door bell

Magicbox (scan/log/replay 433Mhz)

RF hacking case 1: wireless door bell

Magicbox works on:
- a lot of wireless door bell
- few garage

RF hacking case 2: wireless sextoy

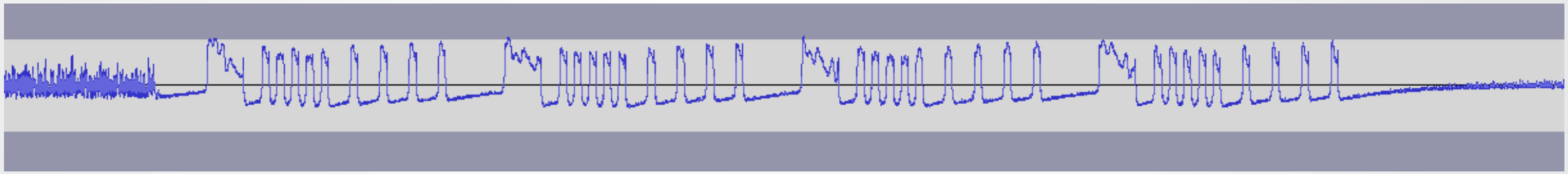
Vibrating Egg Genius Secret Vibes Dorcel

RF hacking case 2: wireless sextoy

Vibrating Egg Genius Secret Vibes Dorcel

RF hacking case 2: wireless sextoy
Vibrating Egg Genius Secret Vibes Dorcel

Signal to switch on the toy.
The signal is always the same for this model (no rolling code)



RF hacking case 2: wireless sextoy

Vibrating Egg Genius Secret Vibes Dorcel

the first **RVE** of the world (**Remote Vibration Execution**)

DEMO

Amazon product description:

*“a large remote range of over 30 meters for **public** pleasure”*

RF hacking case 2: wireless sextoy
Vibrating Egg Genius Secret Vibes Dorcel

MITRE exchange to obtain a CVE:

-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA256

> The remote controller uses radio frequency at 433Mhz, the signal to
> switch on the toy is static and can be easily replay. This
> vulnerability allows an attacker to arbitrary switch on and vibrate
> this model of device few meters away.

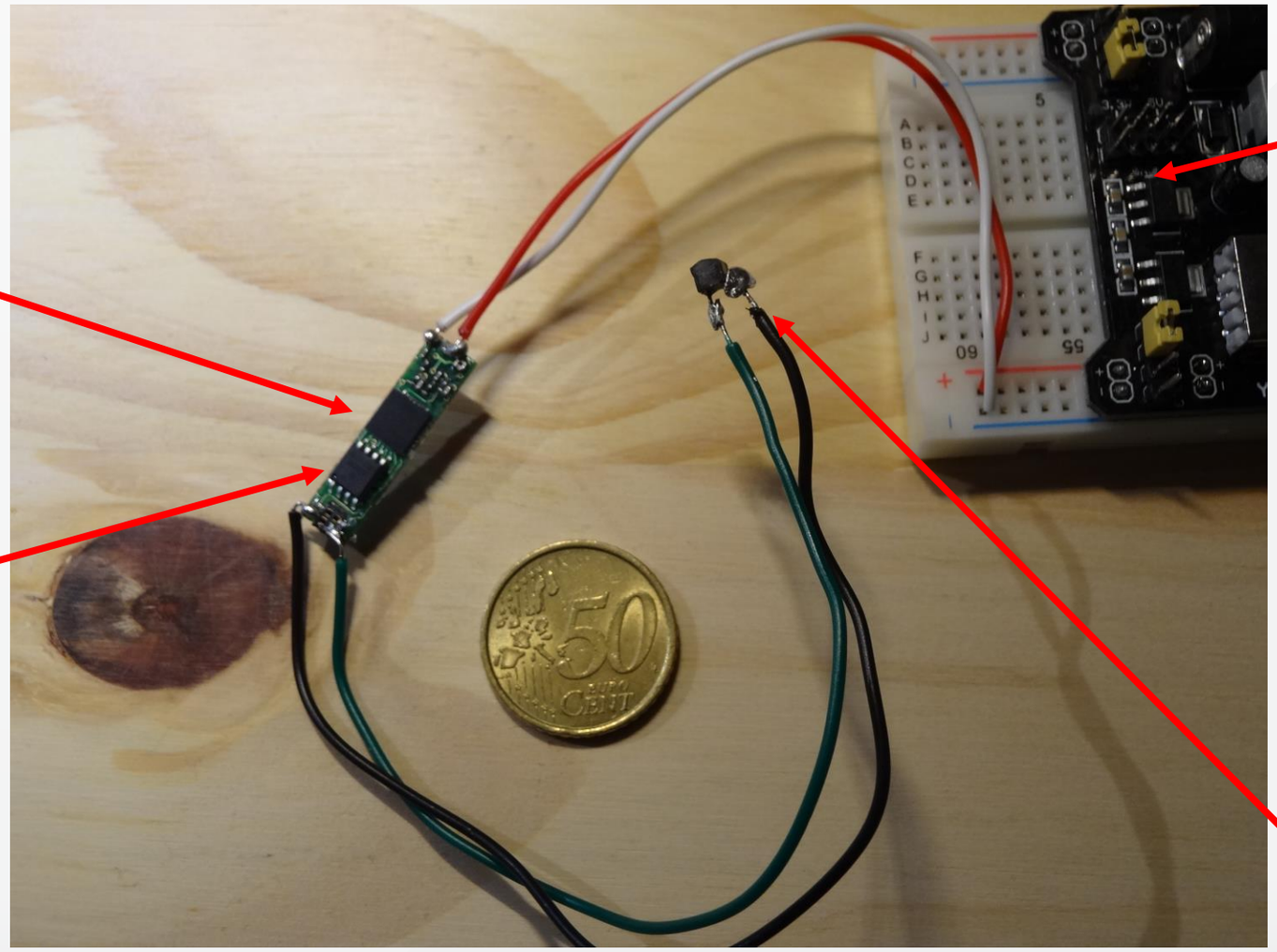
Thank you for contacting the CVE project about your research; however, the finding is outside the scope of CVE. We are categorizing it as a situation in which there is (or was) an opportunity to introduce an additional security feature. The reported behavior of "static and can be easily replay" appears to be reasonable as an initial design, based on the reported exploitation methodology and the threat likelihood.

- --
CVE assignment team, MITRE CVE Numbering Authority
M/S M300
202 Burlington Road, Bedford, MA 01730 USA
[PGP key available through http://cve.mitre.org/cve/request_id.html]
-----BEGIN PGP SIGNATURE-----

hacking case 3: ATM skimmer

Power

Microcontroller

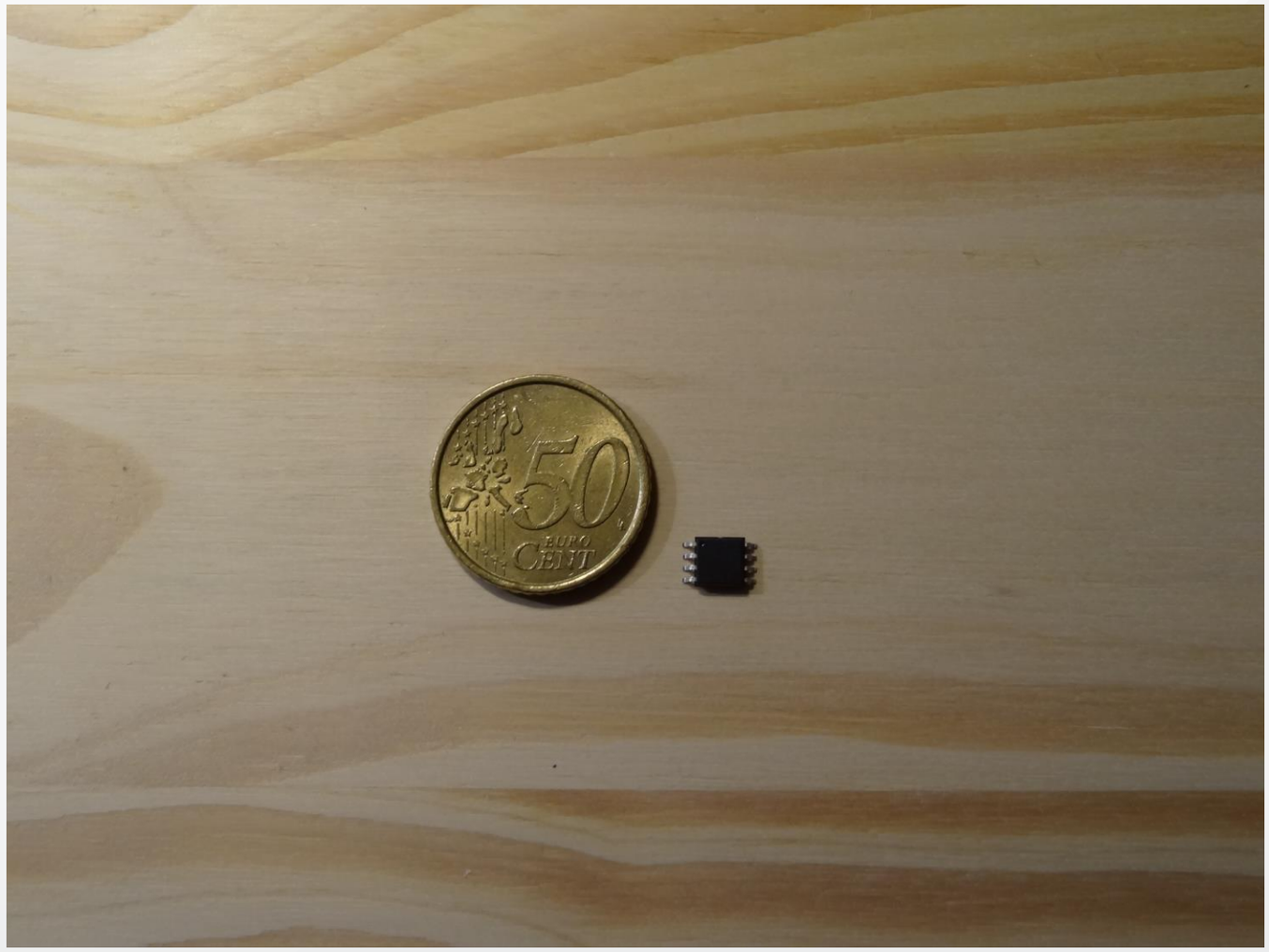


DataFlash

Magnetic reader

hacking case 3: ATM skimmer

AT45DB321D: DataFlash



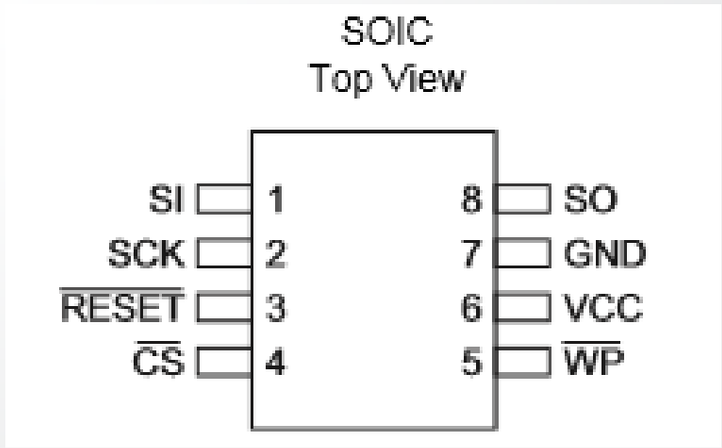
hacking case 3: ATM skimmer

AT45DB321D:
- Atmel dataflash 32Mb
- **SPI** compatible

SPI: Serial Peripheral Interface
Supported by Bus Pirate & Shikra

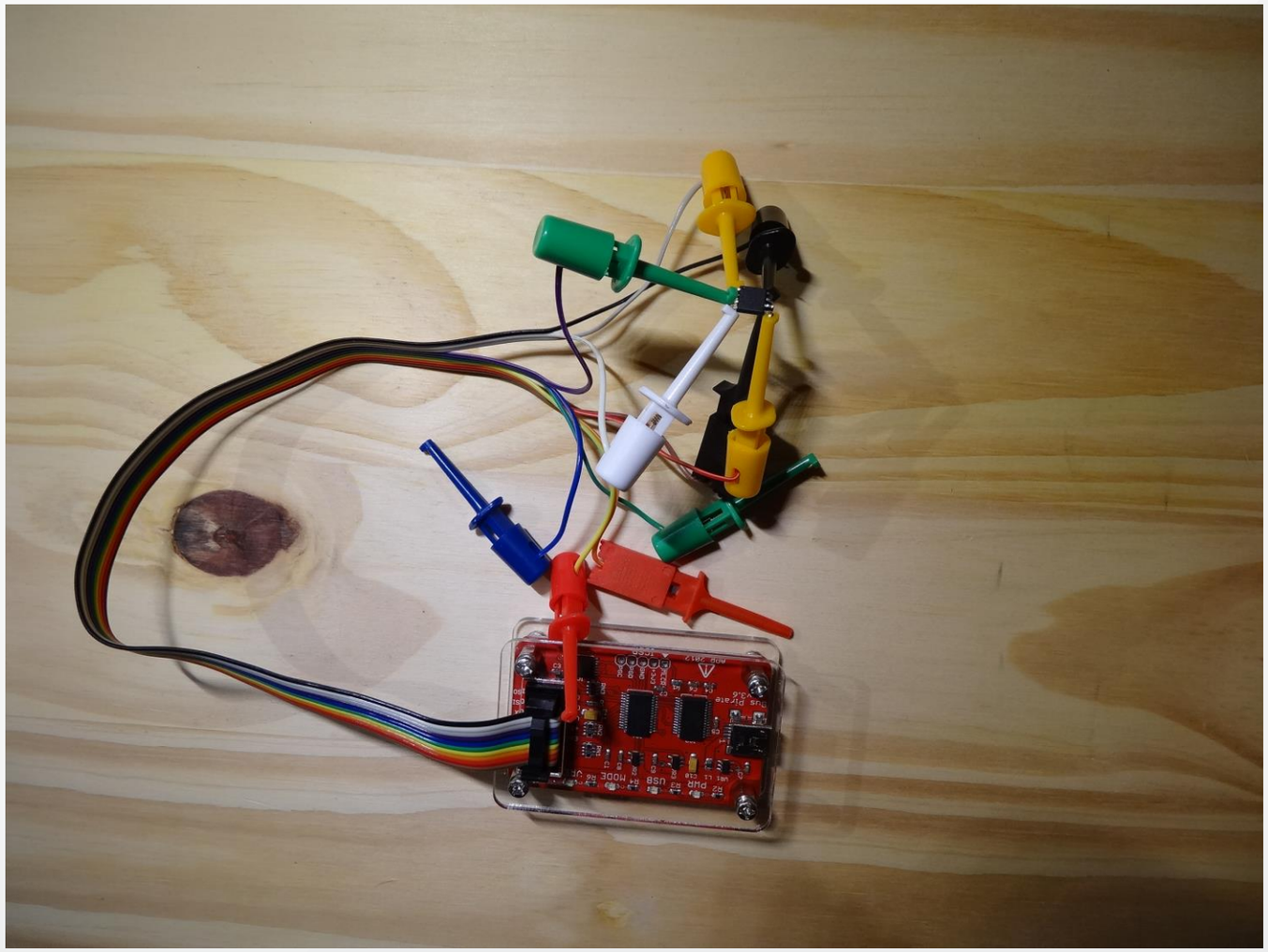
4 PINs:

Bus Pirate	AT45DB321D
- MOSI	→ SI
- MISO	→ SO
- CLOCK	→ SCK
- CS	→ CS



hacking case 3: ATM skimmer

AT45DB321D: bus pirate connection



hacking case 3: ATM skimmer

AT45DB321D:
- flashrom support AT45DB*

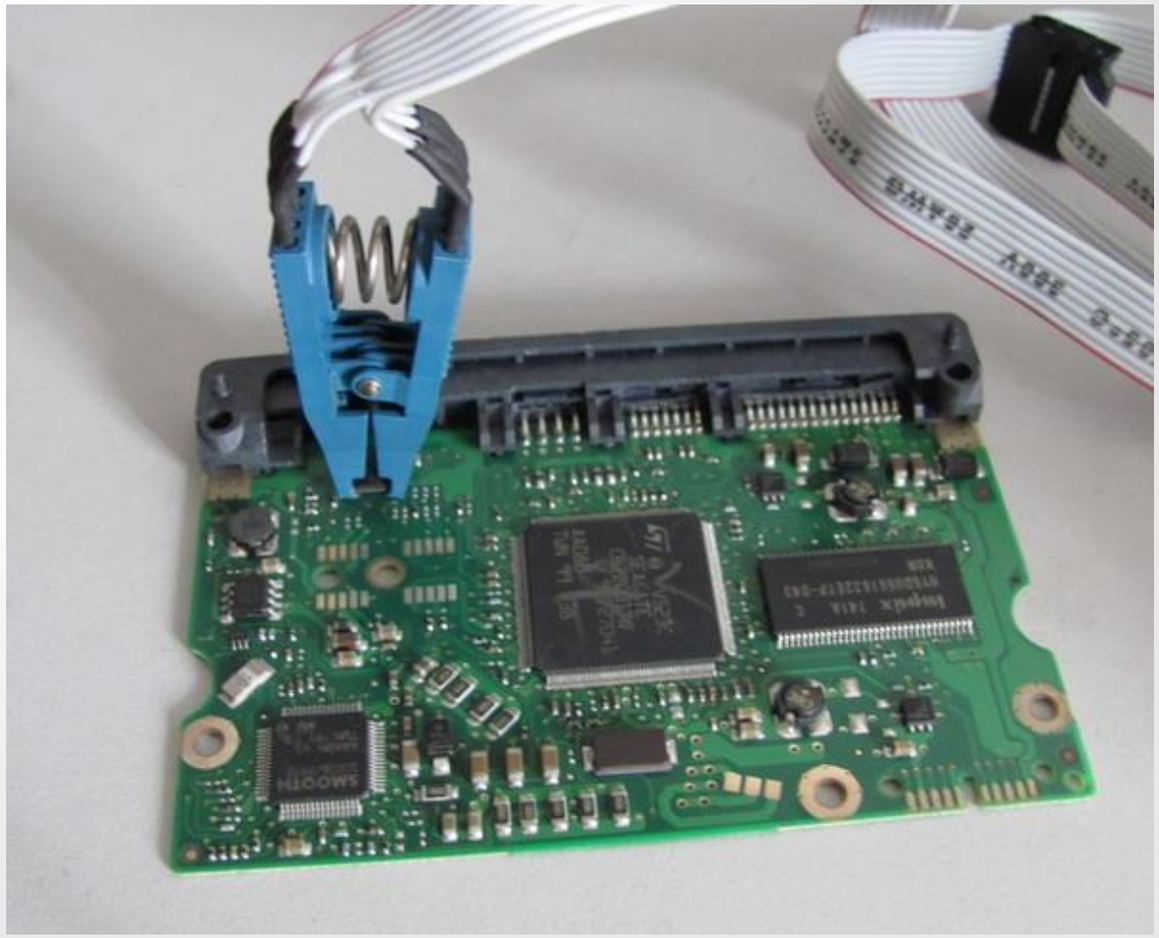
```
$ flashrom -p buspirate_spi:dev=/dev/ttyUSB0,spispeed=1M -r
data.raw
```

The stolen credit card numbers are directly available in the data.raw file.

hacking case 3: ATM skimmer

SPI quick note:

- BIOS can be dump and write thanks to SPI
- Here is the connector

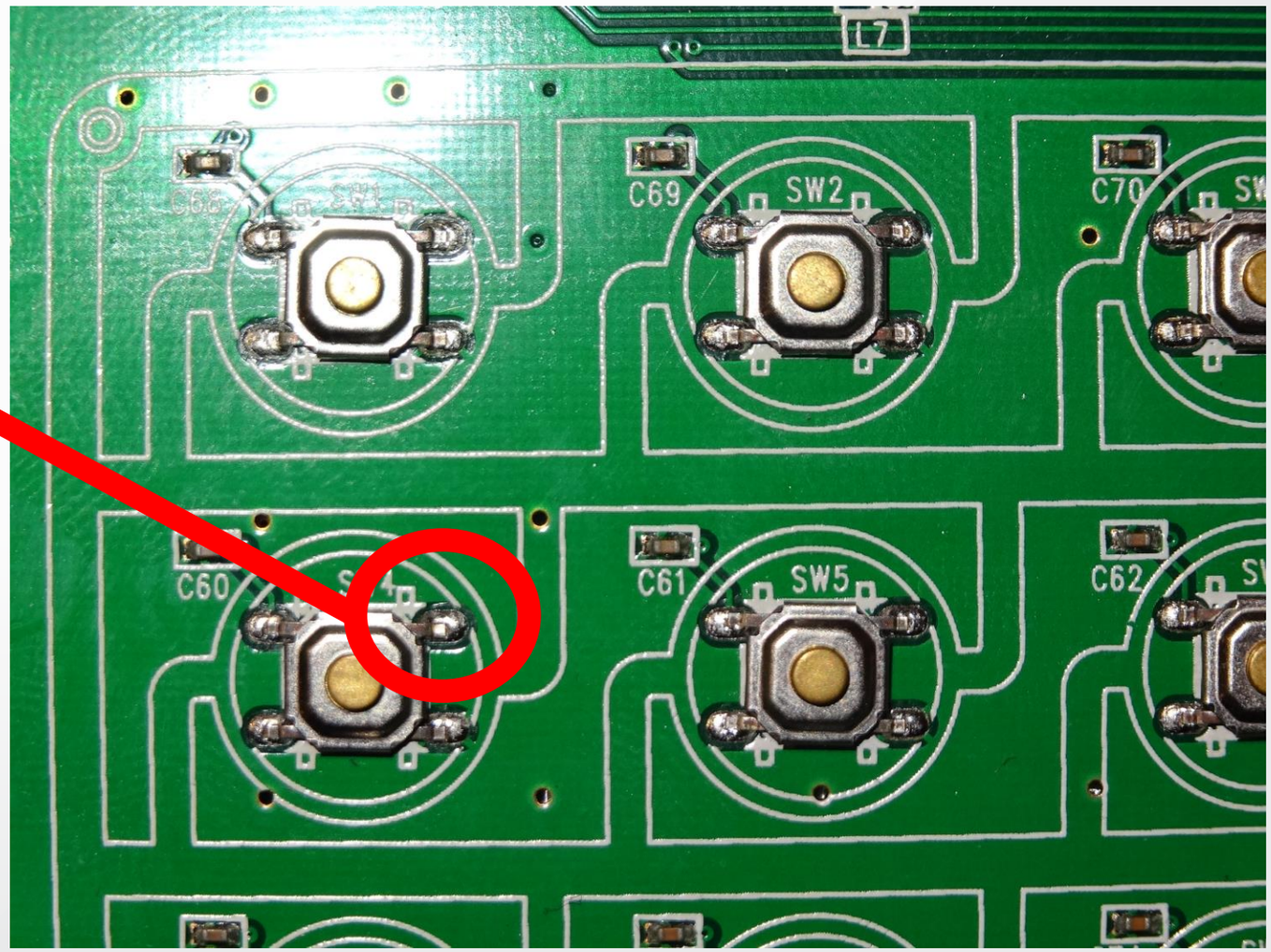


hacking case 4: encrypted hard drive

hacking case 4: encrypted hard drive

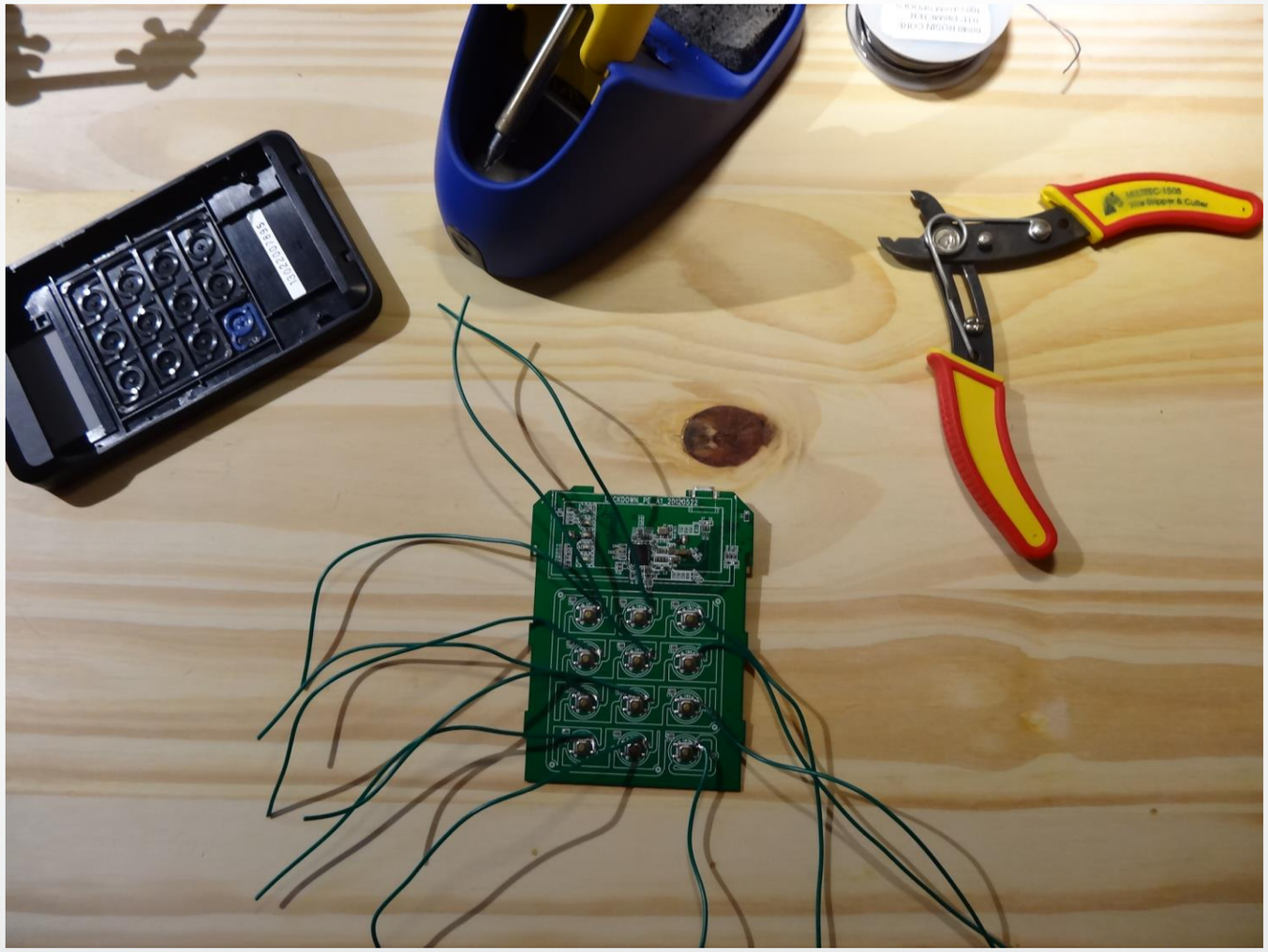
hacking case 4: encrypted hard drive

This PIN switch from 3.3v to 0v when the button is pushed



hacking case 4: encrypted hard drive

Ugly hack: hardware keylogger



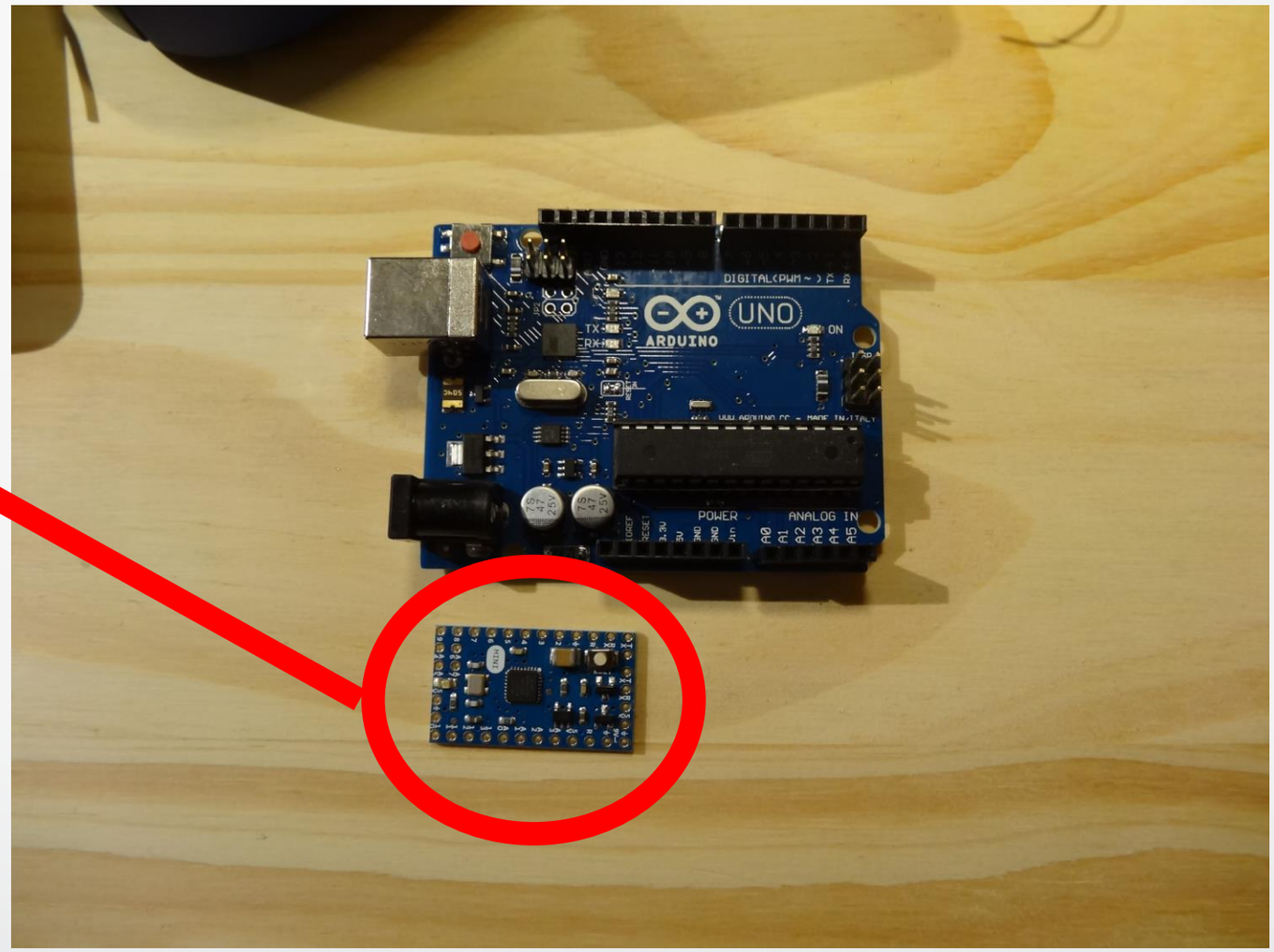
hacking case 4: encrypted hard drive

Ugly hack: hardware keylogger

hacking case 4: encrypted hard drive

Ugly hack: hardware keylogger

Arduino mini

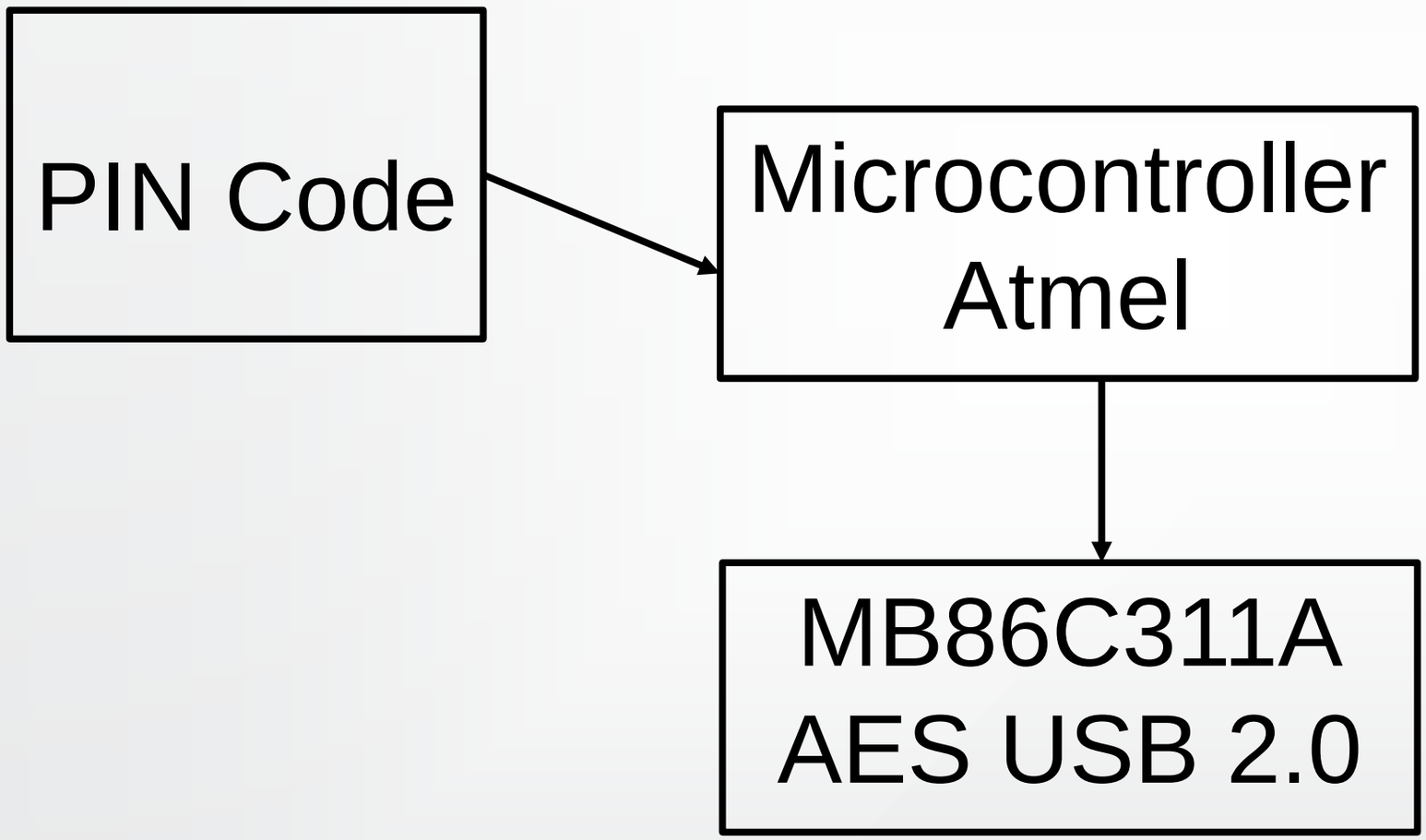


hacking case 4: encrypted hard drive

Ugly hack: hardware keylogger

hacking case 4: encrypted hard drive

Software approach: How work the encrypted hard drive?



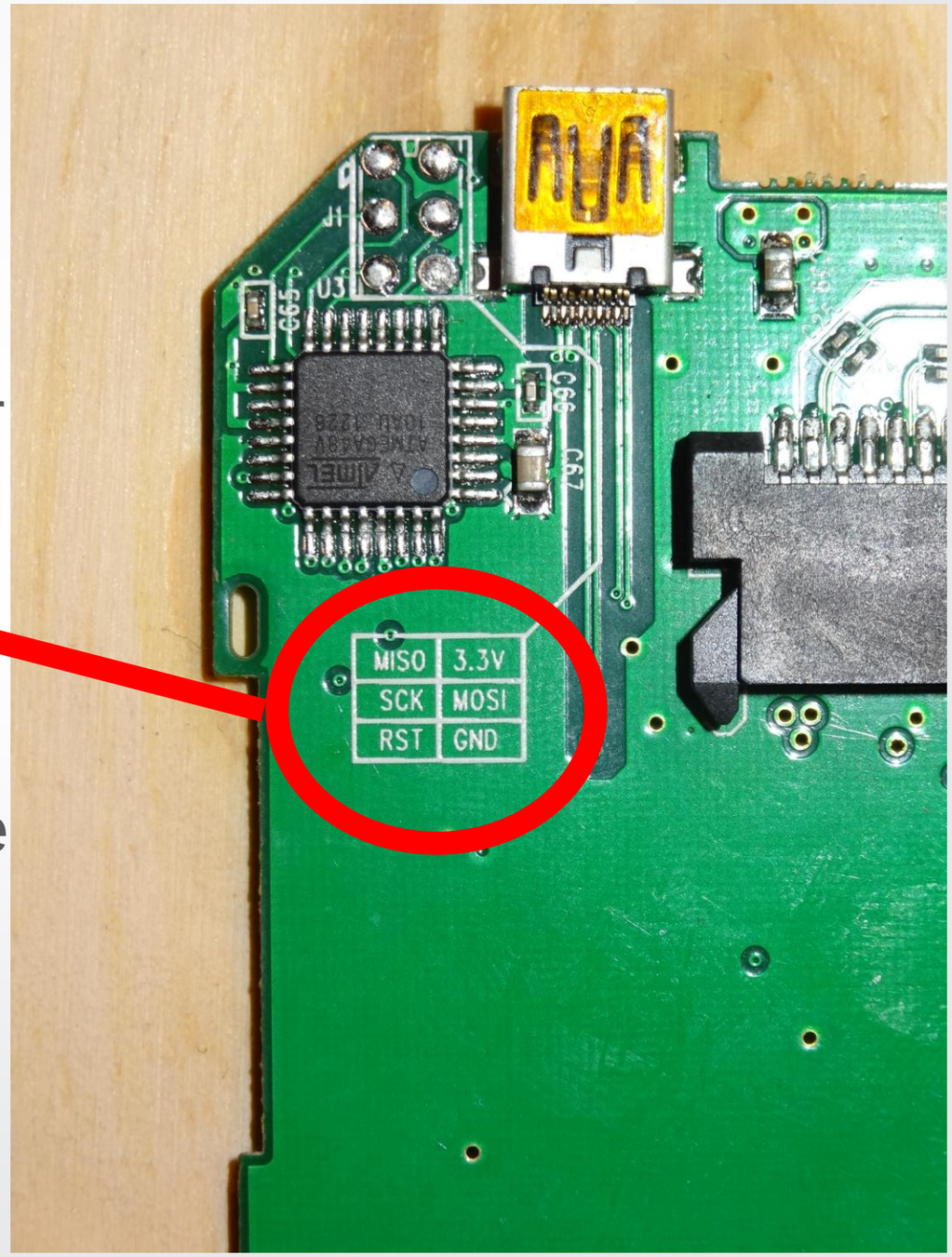
hacking case 4: encrypted hard drive

Software approach
How work the encrypted hard drive?

SPI connector to the Atmel Microcontroller

So we can dump & update the
firmware of the microcontroller
between the PIN pad and the AES module

Goals:
- reverse the firmware
- update it to force the PIN 0000
- no success for the moment :’(



hacking case 5: Firmware & IDA Pro

IDA Pro support a lot of architectures… (don’t forget to choose the good one)
Test with Arduino ATmega328

Firmware dump:

```
paul@lab:~$ avrdude -p m328p -P /dev/ttyACM1 -c arduino -U flash:r:flash.bin:r

avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.00s

avrdude: Device signature = 0x1e950f
avrdude: reading flash memory:

Reading | ################################################## | 100% 4.20s

avrdude: writing output file "flash.bin"

avrdude: safemode: Fuses OK (H:00, E:00, L:00)

avrdude done.  Thank you.
```

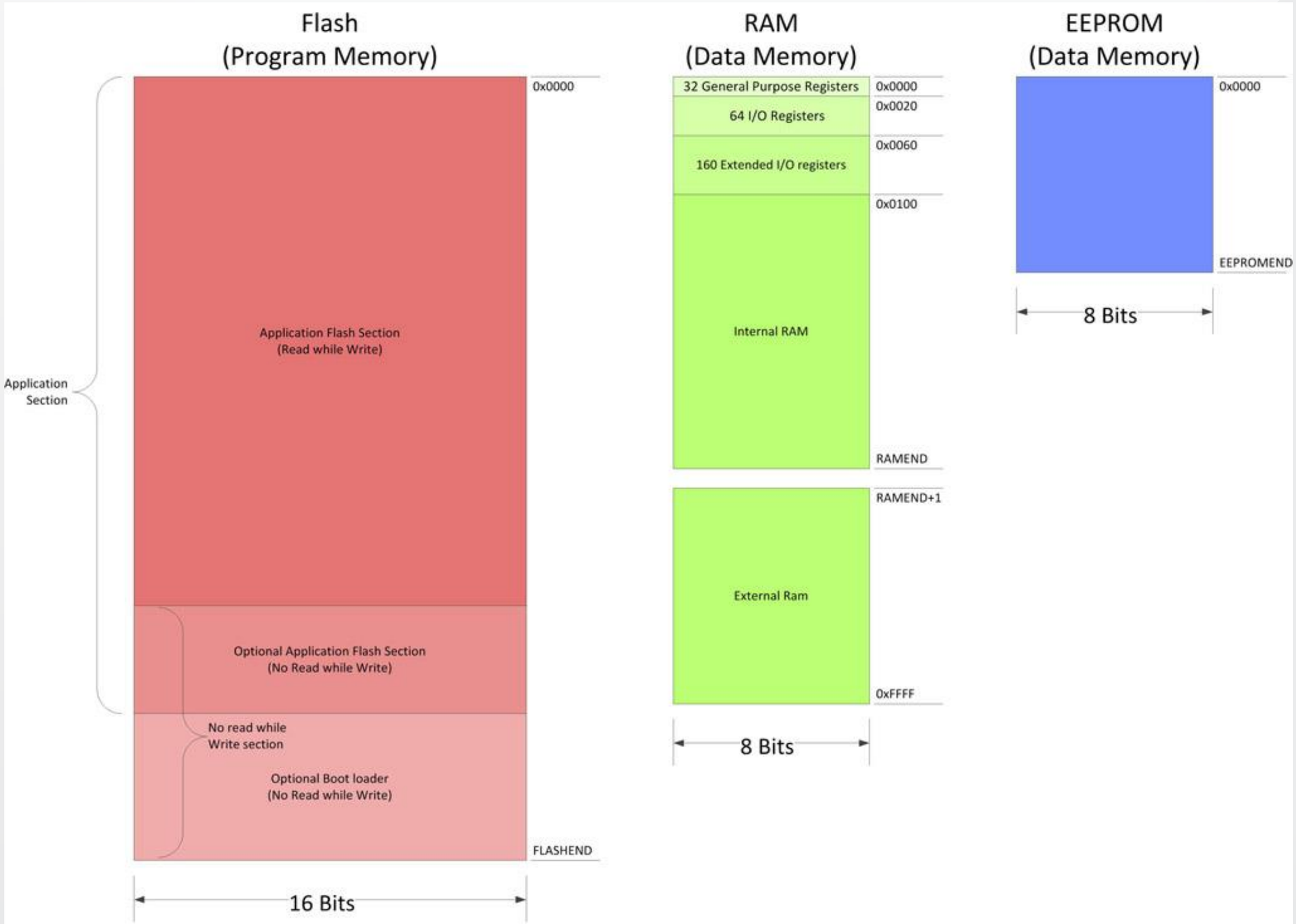
hacking case 5: Firmware & IDA Pro

IDA Pro support a lot of architectures…
Test with Arduino ATmega328

Harvard architecture specificity:

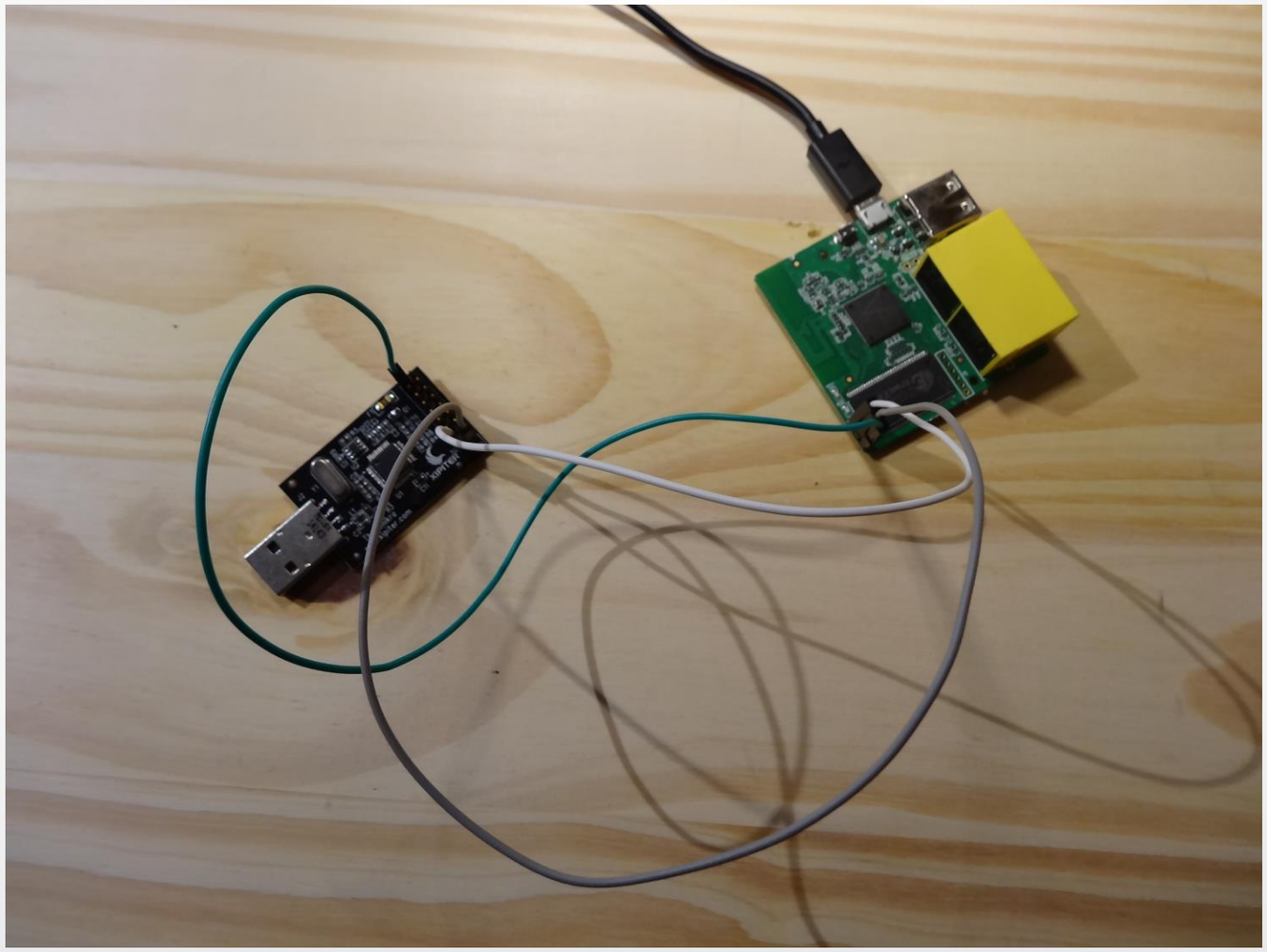
The variables are copied in the RAM and not used directly from the Flash. So the variables in IDA Pro point to unallocated memory (RAM area not mapped). We need to create a new segment of RAM with the data.

hacking case 5: Firmware & IDA Pro



hacking case 6: Linux embedded device

UART (Universal Asynchronous Receiver Transmitter) Shikra connection



hacking case 6: Linux embedded device

UART (Universal Asynchronous Receiver Transmitter)

```
paul@lab:~$ picocom -b 115200 -p n -d 8 /dev/ttyUSB0

Terminal ready
d$�����4�
*****************************************
*      U-Boot 1.1.4  (Sep 25 2014)      *
*****************************************
AP121 (AR9331) U-Boot for GL-iNet
DRAM:  64 MB
FLASH: Winbond W25Q128 (16 MB)
LED on during eth initialization...

 _______  _          _________ _        _______ _________
(  ____ \( \         \__   __/( (    /|(  ____ \\__   __/
| (    \/| (            ) (   |  \  ( || (    \/   ) (
| |      | |            | |   |   \ | || (__       | |
| | ____ | |            | |   | (\ \) ||  __)      | |
| | \_  )| |            | |   | | \   || (         | |
| (___) || (____/\ _ ___) (___| )  \  || (____/\   | |
(_______)(_______/(_)\_______/|/    )_)(_______/   )_(
Hit any key to stop autobooting:  0
Device calibrated. Booting ...
Booting image at: 0x9F020000
   Image name:   OpenWrt r42853
   Image type:   MIPS Linux Kernel Image (lzma compressed)
   Data size:    1113636 Bytes = 1.1 MB
   Load address: 0x80060000
   Entry point:  0x80060000
Uncompressing kernel image... OK!
Starting kernel...
[    0.000000] Linux version 3.10.49 (alzhao@alzhao-ubuntu) (gcc version 4.8.3 (OpenWrt/Linaro GCC 4.8-
2014.04 r42801) ) #6 Thu Oct 9 18:20:47 HKT 2014
[    0.000000] bootconsole [early0] enabled
[    0.000000] CPU revision is: 00019374 (MIPS 24Kc)
```

hacking case 7: Windows 10 IoT

Windows 10 IoT on a Raspberry PI 2

Classic FAT32/NTFS partition:

```
paul@lab:~$ fdisk -l output.img

Disk output.img: 7730 MB, 7730495488 bytes
1 heads, 63 sectors/track, 239660 cylinders, total 15098624 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xae420040

     Device Boot      Start         End      Blocks   Id  System
output.img1   *        4096      135167       65536    c  W95 FAT32 (LBA)
output.img2          147456     8904703     4378624    7  HPFS/NTFS/exFAT
output.img3         8904704    10133503      614400    c  W95 FAT32 (LBA)
output.img4        10133504    15099391     2482944    5  Extended
output.img5        10133632    10135679        1024   70  DiskSecure Multi-
Boot
output.img6        10141696    15098623     2478464    7  HPFS/NTFS/exFAT
```

hacking case 7: Windows 10 IoT

Windows 10 IoT on a Raspberry PI 2

```
paul@lab:~$ ls -l /mnt1
total 16
lrwxrwxrwx 1 root root   216 juil. 10 22:13 CrashDump -> /mnt1//.NTFS-
3G/Volume{2bc78352-273b-11e5-80c0-441ea1418544}
lrwxrwxrwx 1 root root   216 juil. 10 22:13 Data -> /mnt1//.NTFS-3G/Volume{2bc78353-
273b-11e5-80c0-441ea1418544}
drwxrwxrwx 1 root root     0 juil. 10 22:13 EFI
lrwxrwxrwx 1 root root   216 juil. 10 22:13 EFIESP -> /mnt1//.NTFS-
3G/Volume{2bc7834c-273b-11e5-80c0-441ea1418544}
drwxrwxrwx 1 root root     0 juil. 10 22:14 IoTApps
drwxrwxrwx 1 root root     0 juil. 10 22:14 ProgramData
drwxrwxrwx 1 root root     0 juil. 10 22:14 Program Files
drwxrwxrwx 1 root root     0 juil. 10 22:13 Program Files (x86)
drwxrwxrwx 1 root root     0 juil. 10 22:13 PROGRAMS
drwxrwxrwx 1 root root 12288 juil. 10 22:14 RDBG
drwxrwxrwx 1 root root     0 juil. 10 22:13 System Volume Information
drwxrwxrwx 1 root root     0 juil. 10 22:14 Users
drwxrwxrwx 1 root root  4096 juil. 10 22:14 Windows


paul@lab:~$ file /mnt1/Windows/System32/NETSTAT.EXE
/mnt1/Windows/System32/NETSTAT.EXE: PE32 executable (console) ARMv7 Thumb, for MS
Windows
```

What’s next?
In real life?

Real cases of hardware or embedded system hacks

- CISCO SYNful Knock: malicious OS
- MalwareTech SBK: A bootkit capable of surviving reformat (hard drive firmware hacking)
- NSA hard drive firmware hacking
- Payment terminal fake firmware
- Hacking Team BIOS compromise
- …

But It’s not something new:

- “Greek wiretapping case of 2004-2005”: the legal wiretaps (lawful interceptions) of a Ericsson AXE is compromised in order to silently intercept communications. A rogue is put on a firmware. 6500 lines of code written in the PLEX programming language…

The future… Yes I'm medium

- What's next step?
- Common practice today or tomorrow?
- Can you trust your hardware?
- Your appliances?
- Your routers?
- Your gateways?
- Your magic sandbox (to compromised a sandbox would be ironic)?
- Have you got the internal skills to deal with this kind of malicious code?
- How to detect it?
- Does hardware is the next trend?
- …

Thank you for your attention.

Questions or awkward silence?