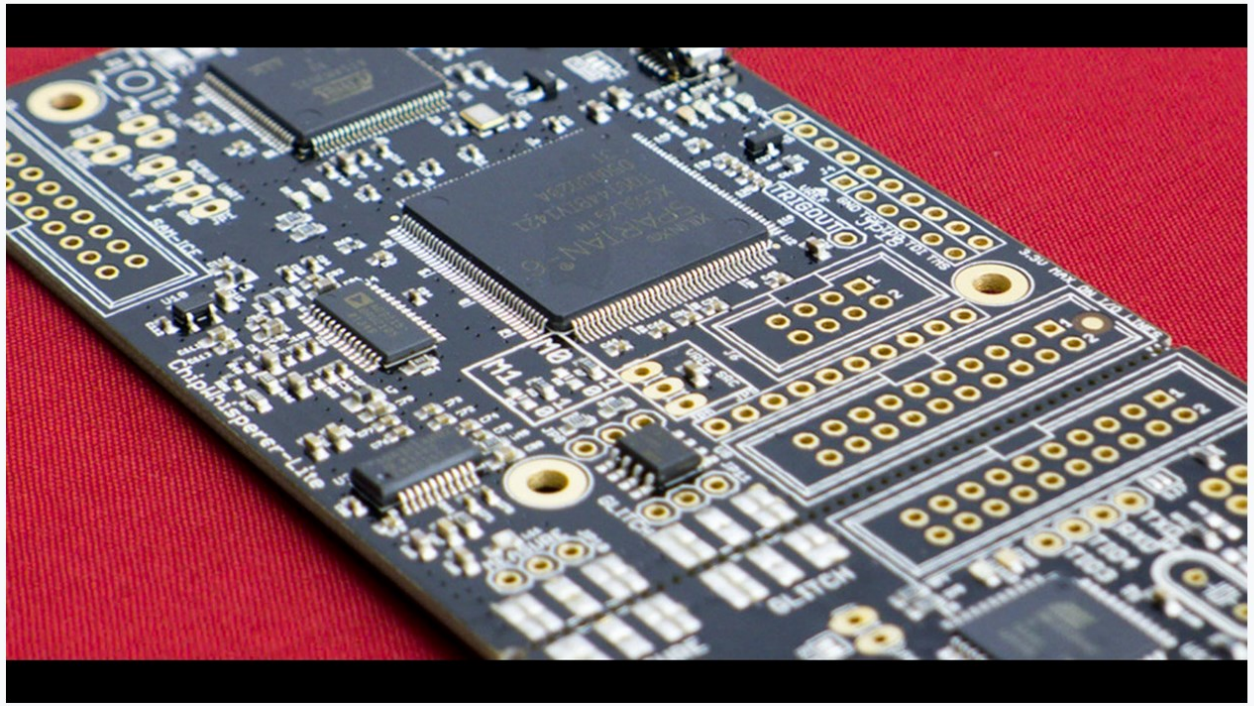
Brute-Forcing Lockdown Harddrive PIN Codes

Colin O'Flynn

blackhat USA 2016

About Me

ChipWhisperer-Lite: A New Era of Hardware Security Research



Embedded security - is it an oxymoron? Learn the truth through a series of hands-on labs targeting computer and electrical engineers.

Created by

Colin O'Flynn

331 backers pledged $88,535 CAD to help bring this project to life.

Types of Secure Drives

Previous Work

Joffrey Czarny & Raphaël Rigo

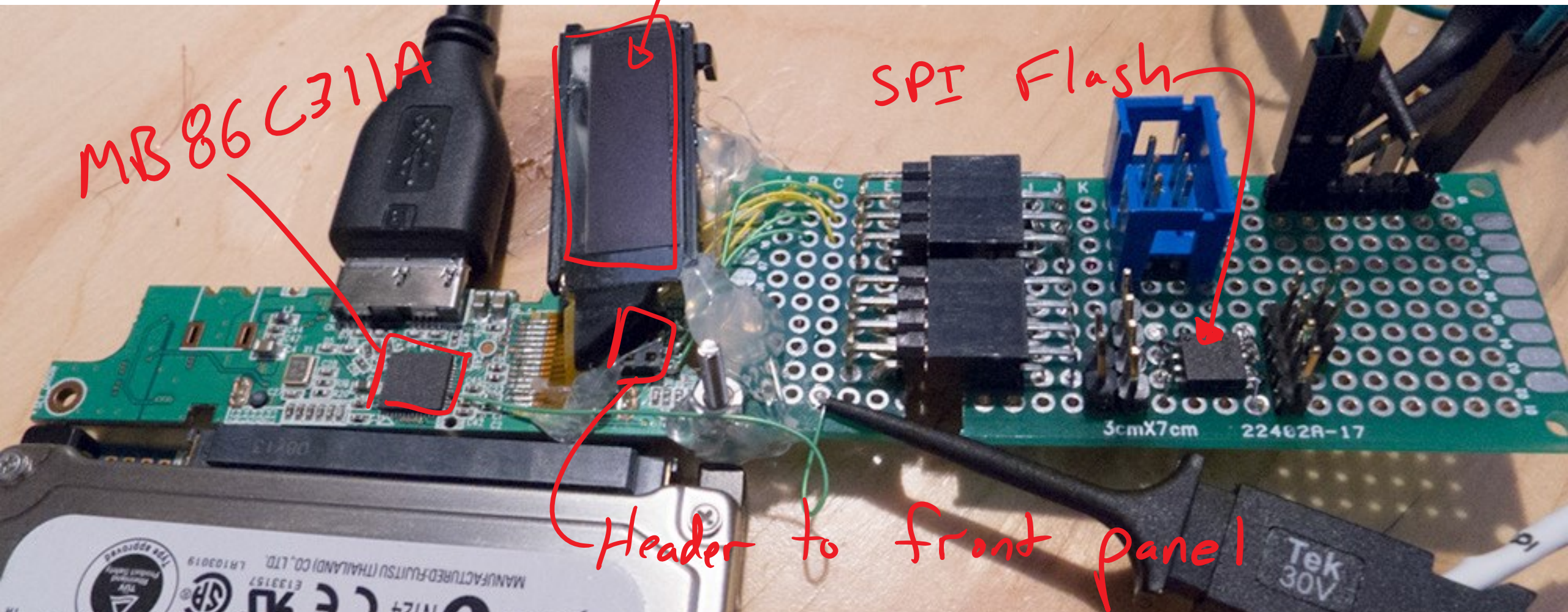
Presentation at Hardwear.io:

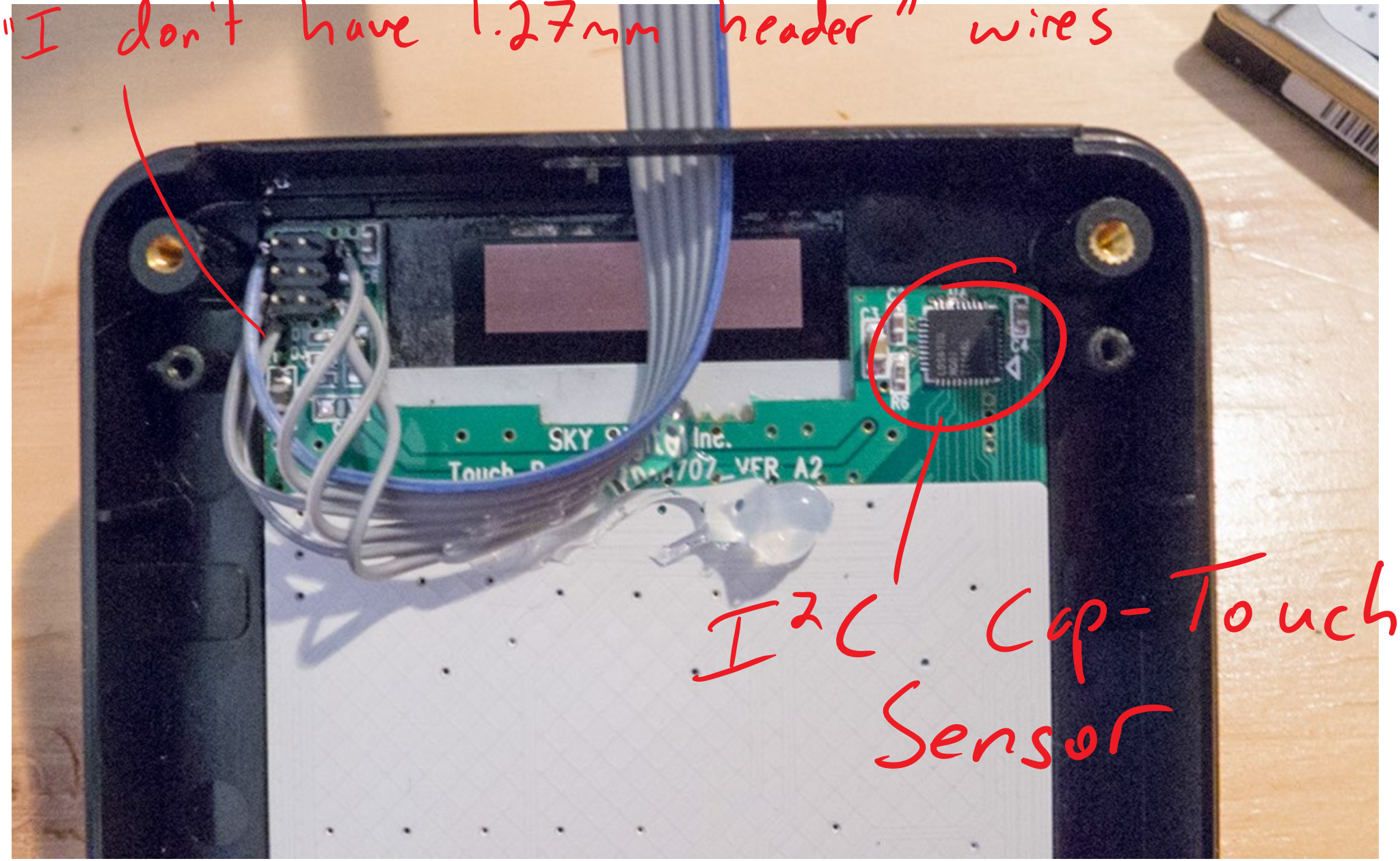
http://hardwear.io/wp-content/uploads/2015/10/Slide-hardware_re_for_software_reversers-By-Czarny-Rigo.pdf

Lots of details in paper:

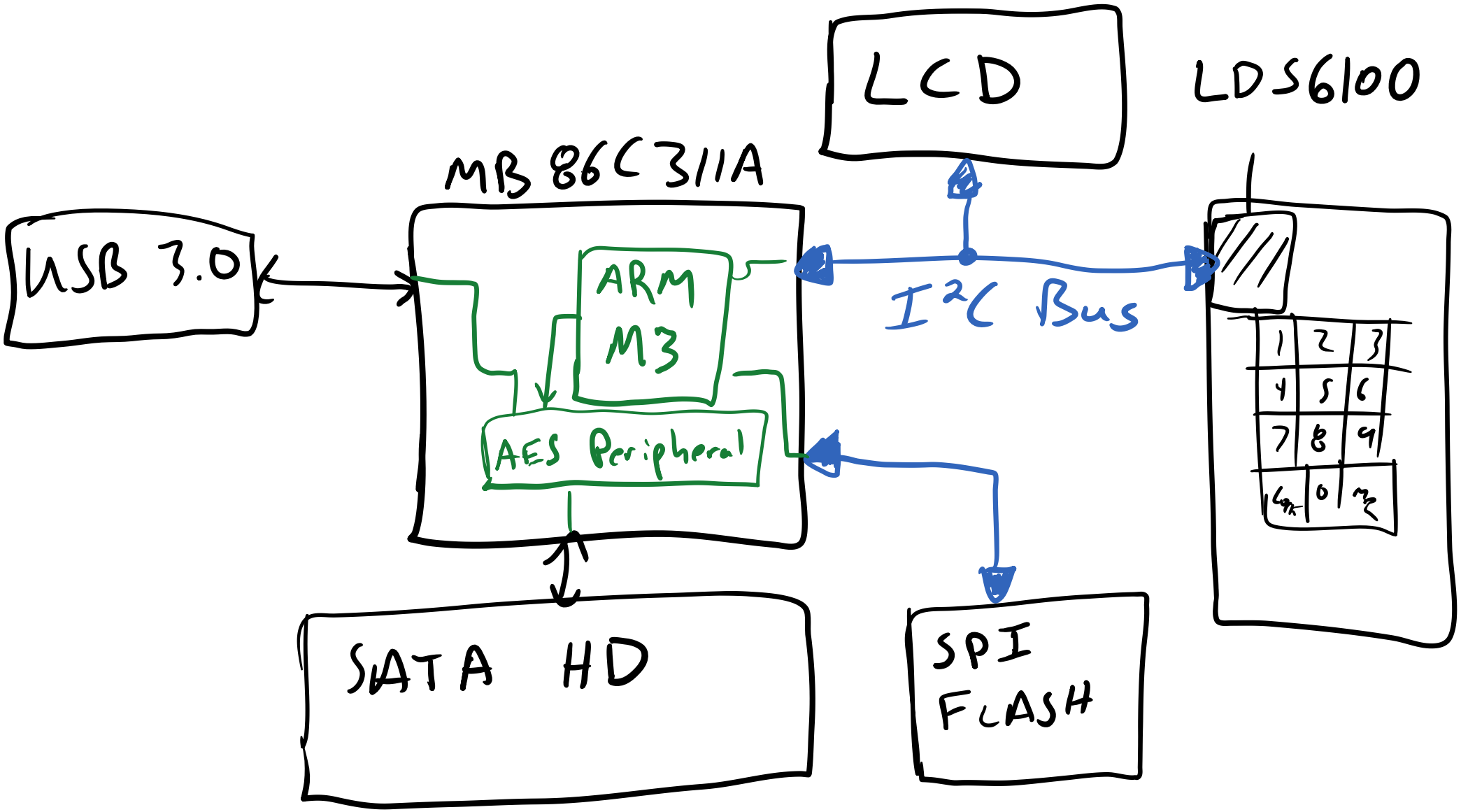
https://www.sstic.org/media/SSTIC2015/SSTIC-actes/hardware_re_for_software_reversers/SSTIC2015-Article-hardware_re_for_software_reversers-czarny_rigo.pdf


BRUTE
FORCING


LCD
MB86C311A
SPI Flash
Header to front panel
5cmX7cm
22462A-17
Tek
30V

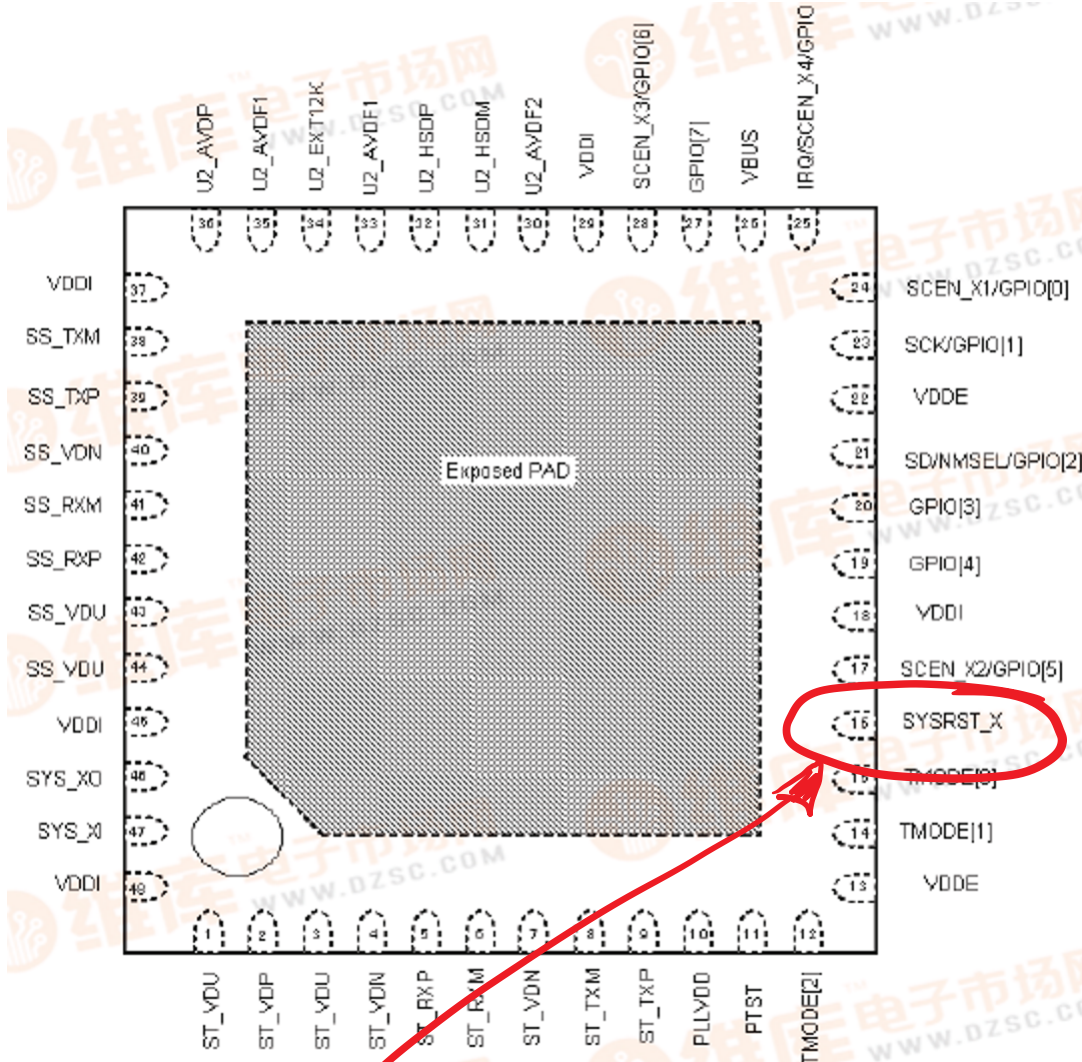

"I don't have 1.27mm header" wires
I²C Cap-Touch Sensor

Block Diagram

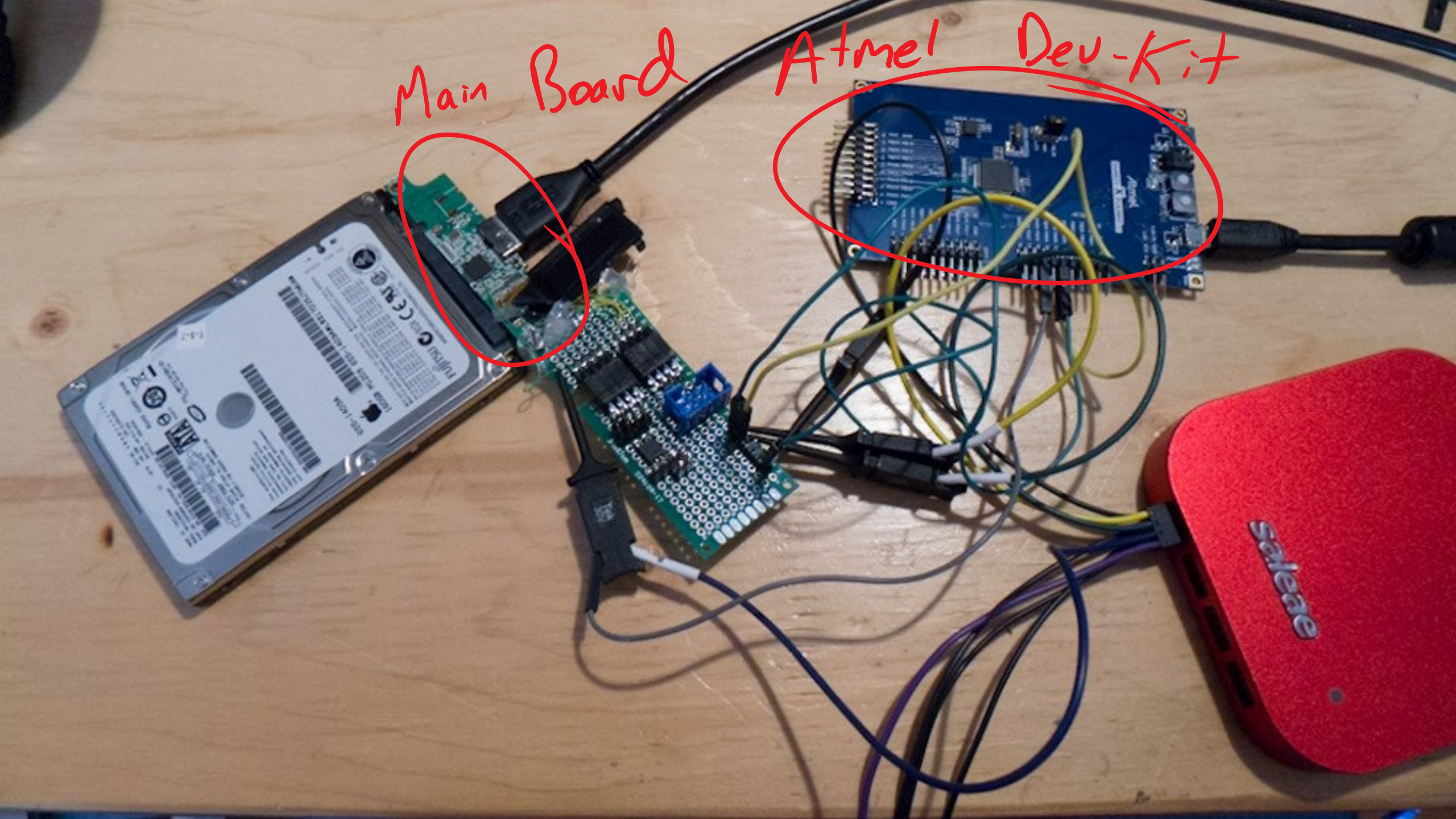


3. Turning on/off the power supply

item	Regulator		Pin
Turning on/ off the power supply	3.3V Power supply	Digital power supply for external IO	VDDE
		SATA 3.3V analog power supply	ST_VDP
		USB2.0 3.3V analog power supply	U2_AVDF1, U2_AVDB
Turning on/off the power supply	1.2V Power supply	Analog power supply for PLL	PLLVDD
		Digital power supply for internal core	VDDI
		SATA 1.2V analog power supply	ST_VD, ST_VDU
		USB3.0 1.2V analog power supply	SS_VDN、SS_VDU
		USB2.0 1.2V analog power supply	U2_AVDF2, U2_AVDP

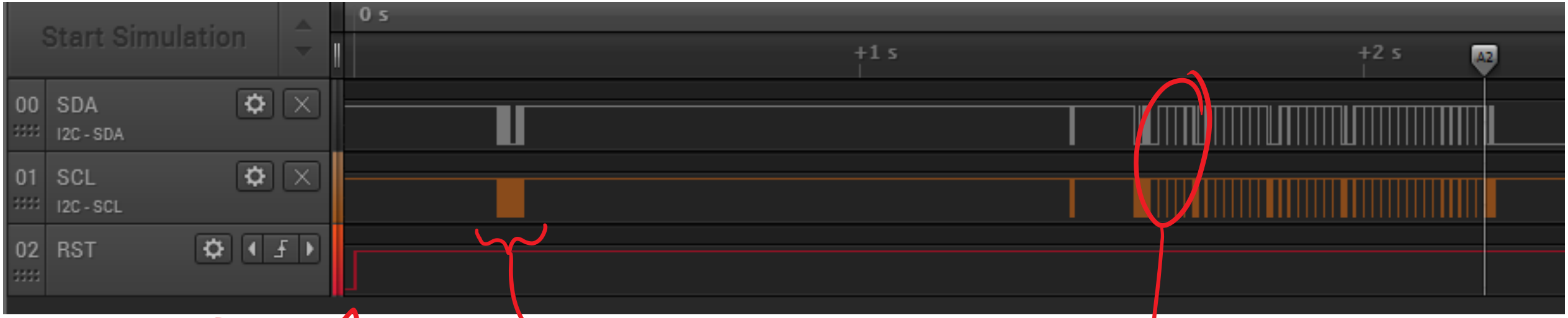


Can we bypass delay?


Main Board
Atmel Dev-Kit
saleae

1. Wait for I²C poll
2. Feed in button presses.
3. Check response.

https://github.com/colinoflynn/hddkeyboard-spoof-demo/



Reset released

Initilization of LCD

Polling for buttons

FINDINGS

1. Time-out not stored in NVM.
   ↳ Can reset chip to bypass waits.
2. "Reasonable" entry speeds enforced
   ↳ ~0.5s for 4 digits
3. Boot involves about 1.5s delay.

Results

- 4 digit pin takes 2.23 s

Pin Length	Guess Time (worst case)
4	6.2 hrs
5	2.6 days
6	26 days
7	260 days

You purchased this item on Dec 23 2015.
View this order

Satechi LockDown USB 3.0 Super-Speed 256-bit Encrypted Portable External 2.5" SATA Hard Drive Enclosure

by Satechi

4.5 out of 5 stars 13 customer reviews

Currently unavailable.

We don't know when or if this item will be back in stock.

- Quality OLED screen enables easy password input, menu navigation, HDD info and more
- Incorporates capacitance touch panel numeric keypad - Unlock the drive with your own unique 4 to 8 digit password
- USB 3.0 Super-Speed (5Gbps) driver also supports USB 2.0, ARM based 32bit Controller
- Measures (W) 3.1 x (H) 5 x (D) 0.5 inches / 3.25 ounces (w/o HDD)


CHECKING
CHIPS

OK! How SECURE ARE THESE?

Previous work by Czarny & Rigo:

8 Conclusion

Starting with no information, we managed, in full black box, to have a good understanding of the way this encrypted drive enclosure works.

While we know the crypto design is a fail, because all the encryption related data is stored on the drive itself, with no enclosure dependent secret, we were unable to actually exploit it.

SWAP IC FOR VIRGIN

→ Device worked.

↳ NO fuse bits or other secret specific to manufacture.

Compare Bins

Two versions of program for Zalman drive.

(This is from Czarny & Rigo)

zalman\fw\VE400_firmware_1-37(FAT).bin

```
0000 0000: 32 FF 1F 81 8E F2 DE E2  34 76 A0 2B 77 CC 03 E3  2 .üÄ=îõ 4vá+w╠.õ
0000 0010: 92 A9 6E 62 37 E5 8E 48  EA 5D 58 97 8D 5E A5 E6  Æ®nb7õÄH Ô]Xùì^Ñµ
0000 0020: 84 2E 01 00 42 00 00 00  AA C8 01 A7 67 D6 C1 18  ä...B...  ¬╚.ºgí┴.
0000 0030: B0 4D 92 1B 28 0C 38 8F  AB D1 0E 18 68 5C 21 D1  ▒MÆ.(.8Å ½Ð..h\!Ð
0000 0040: EC F8 0B D7 6D CB 4F 13  05 8A C3 32 F7 6E 21 38  ý°.Îm╥O. .è├2.n!8
0000 0050: 2A C1 36 9E 5A 1C 6B BF  78 F2 EC BC 7B 5F A1 6C  *┴6×Z.k┐ x=ý╝{_íl
0000 0060: 2C A4 5A B1 6F 1B 80 E2  53 7E A4 FC FA 67 00 18  ,ñZ▒o.Çô S~ñ³·g..
0000 0070: 50 63 63 D0 83 D1 E3 9A  F5 D6 16 88 C2 19 67 F1  PccðâÐÓÜ §í.ê┬.g±
0000 0080: A8 77 7F A4 C1 76 3D A9  4C 6C 0B 76 1E 3E 18 BA  ¿wo╡┴v=® Ll.v.>.‖
0000 0090: 67 F2 42 55 FC C0 91 A8  10 83 E1 7B DF E6 26 E0  g=BU³└æ¿ .âß{■µ&ó
0000 00A0: 33 44 4B 90 50 A3 55 32  7C 39 12 0A 89 19 DF E0  3DKÉPúU2 ¦9..ë.▀ó
0000 00B0: F2 00 30 07 CB AC EC EA  B1 E9 DA 9C C6 02 0E 55  =.0.╥¾ýÛ ▒ú┌£ã..U
0000 00C0: 14 CC 19 19 6D 31 85 41  A7 3F 92 15 9F F2 CA 77  .╠..m1àA º?Æ.ƒ=╩w
0000 00D0: 90 D0 BF 77 63 E8 0A 85  88 14 2E 49 3E 22 F5 05  Éð┐wcÞ.à ê..I>"§.
0000 00E0: 96 B0 C1 3A 93 23 4C 51  7C 7A BB CD C3 19 10 7F  û▒┴:ô#LQ ¦z╗=├..⌂
0000 00F0: B2 8F 34 59 B7 0E B4 F2  75 43 10 D5 5B 22 7D 86  ▒Ä4Yà.┤= uC.┴["}å
0000 0100: 0E 93 D1 03 43 37 BB D1  1C C9 DF 95 EC 7C 73 37  .ôÐ.C7╗Ð .╔▀òý¦s7
0000 0110: 83 90 A9 EF 89 A1 2B 12  BB 52 38 C2 0B 66 8F DC  âÉ®´ëí+. ╗R8┬.fÄ▄
0000 0120: C5 3C 47 D6 9B 97 4F F1  3A 01 87 DC C6 50 18 95  ┼<GíøùO± :.ç▄ãP.ò
0000 0130: D7 0E 75 E0 17 83 32 A0  19 3D 46 5A DC 44 88 DF  Î.uó.â2á .=FZ▄Dê▀
0000 0140: E4 D0 84 89 86 FC 9B BD  FA D7 F1 BE C5 79 EF C4  õðäëå³ø¢ ·î±¥┼y´−
0000 0150: 96 2D D2 5C 5C F4 4C E8  24 83 93 CB 12 B1 18 04  û−Ê\\¶LÞ $âô╥.▒..
0000 0160: 94 BD 16 44 49 C3 54 36  76 A6 4A D1 5D 4C BE E0  öç.DI├T6 v¦JÐ]L¥ó
0000 0170: FF 60 7D 96 D3 DD 9C C7  9A 69 C0 60 C7 7F EB 8F  ˙`}ûÓ¦£Ã Üi└`Ã⌂ùÅ
0000 0180: DE F1 0E CB 7F C9 55 28  D7 23 7E 1F 98 10 00 4D  ì±.╥⌂╓U( Î#~.ÿ..M
0000 0190: 53 8D CF 14 50 32 6C 6E  82 C6 E1 06 2B C6 22 B4  Sì¤.P2ln éãß.+ã"┤
0000 01A0: 8A 23 ED EB F4 46 0F 15  02 EF 45 0A 77 59 A3 9B  è#ýÙ¶F.. .´E.wY£¢
0000 01B0: 21 54 1F 5B DD 68 6D 07  F3 A3 77 AD 73 99 8E 70  !T.[¦hm. ¾úw¡sÖÄp
```

zalman\fw\VE400_firmware_1-37(NTFS).bin

```
0000 0000: 01 ED 10 47 A0 70 D3 BE  A4 B0 6B DE 49 0E 09 67  .ý.GápÉ¥ ñ▒k╥I..g
0000 0010: E1 87 DB B4 37 87 4B 5A  4B 6D 28 F0 9E 88 21 CD  ßç▄┤7çKZ Km(−xê!=
0000 0020: 94 2E 01 00 02 00 00 00  AA C8 01 A7 67 D6 C1 18  ö.......  ¬╚.ºgí┴.
0000 0030: CC 4D 92 1B 54 0C 38 8F  D7 D1 0E 18 14 5C 21 D1  ╠MÆ.T.8Å ÎÐ...\!Ð
0000 0040: 90 F8 0B D7 6D CB 4F 13  05 8A C3 32 F7 6E 21 38  É°.Îm╥O. .è├2.n!8
0000 0050: 2A C1 36 9E 26 1C 6B BF  04 F2 EC BC 7B 5F A1 6C  *┴6×&.k┐ .=ý╝{_íl
0000 0060: 50 A4 5A B1 13 1B 80 E2  4F 7E A4 FC F6 67 00 18  PñZ▒..Çô O~ñ³÷g..
0000 0070: 4C 63 63 D0 83 D1 E3 9A  F9 D6 16 88 36 1A 67 F1  Lcc⌂âÐÓÜ ¨í.ê6.g±
0000 0080: BC 77 7F A4 3D 75 3D A9  B0 6F 0B 76 EA 3D 18 BA  ╝wo╡=u=® ▒o.vÛ=.‖
0000 0090: 6B F2 42 55 E8 C0 91 A8  6C 83 E1 7B A3 E6 26 E0  k=BUÞ└æ¿ lâß{úµ&ó
0000 00A0: 27 44 4B 90 2C A3 55 32  00 39 12 0A F5 19 DF E0  'DKÉ,úU2 .9..§.▀ó
0000 00B0: 8E 00 30 07 A7 AC EC EA  A5 E9 DA 9C BA 02 0E 55  ä.0.º¾ýÛ ÑÚ┌£‖..U
0000 00C0: 00 CC 19 19 71 31 85 41  B3 3F 92 15 9F F2 CA 77  .╠..q1àA ∫?Æ.ƒ=╩w
0000 00D0: EC D0 BF 77 6F E8 0A 85  88 14 2E 49 3E 22 F5 05  ýð┐woÞ.à ê..I>"§.
0000 00E0: 96 B0 C1 3A 93 23 4C 51  7C 7A BB CD C3 19 10 7F  û▒┴:ô#LQ ¦z╗=├..⌂
0000 00F0: B2 8F 34 59 B7 0E B4 F2  75 43 10 D5 5B 22 7D 86  ▒Ä4Yà.┤= uC.┴["}å
0000 0100: 0E 93 D1 03 43 37 BB D1  1C C9 DF 95 EC 7C 73 37  .ôÐ.C7╗Ð .╔▀òý¦s7
0000 0110: 83 90 A9 EF 89 A1 2B 12  BB 52 38 C2 67 66 8F DC  âÉ®´ëí+. ╗R8┬gfÄ▄
0000 0120: 29 3C 47 D6 9B 97 4F F1  3A 01 87 DC C6 50 18 95  )<GíøùO± :.ç▄ãP.ò
0000 0130: D7 0E 75 E0 17 83 32 A0  19 3D 46 5A DC 44 88 DF  Î.uó.â2á .=FZ▄Dê▀
0000 0140: E4 D0 84 89 86 FC 9B BD  FA D7 F1 BE C5 79 EF C4  õðäëå³ø¢ ·î±¥┼y´−
0000 0150: 96 2D D2 5C 5C F4 4C E8  24 83 93 CB 12 B1 18 04  û−Ê\\¶LÞ $âô╥.▒..
0000 0160: 94 BD 16 44 49 C3 54 36  76 A6 4A D1 5D 4C BE E0  öç.DI├T6 v¦JÐ]L¥ó
0000 0170: FF 60 7D 96 D3 DD 9C C7  9A 69 C0 60 C7 7F EB 8F  ˙`}ûÓ¦£Ã Üi└`Ã⌂ùÅ
0000 0180: DE F1 0E CB 7F C9 55 28  D7 23 7E 1F 98 10 00 4D  ì±.╥⌂╓U( Î#~.ÿ..M
0000 0190: 53 8D CF 14 50 32 6C 6E  82 C6 E1 06 2B C6 22 B4  Sì¤.P2ln éãß.+ã"┤
0000 01A0: 8A 23 ED EB F4 46 0F 15  02 EF 45 0A 77 59 A3 9B  è#ýÙ¶F.. .´E.wY£¢
0000 01B0: 21 54 1F 5B DD 68 6D 07  F3 A3 77 AD 79 99 8E 70  !T.[¦hm. ¾úw¡yÖÄp
```

Stream Cipher

Keystream : E7 1A F1 37

⊕ ⊕ ⊕ ⊕

Plain text : 2C 1F 09 1A ....

Cipher text : CB 05 F8 2D

Compare Bins

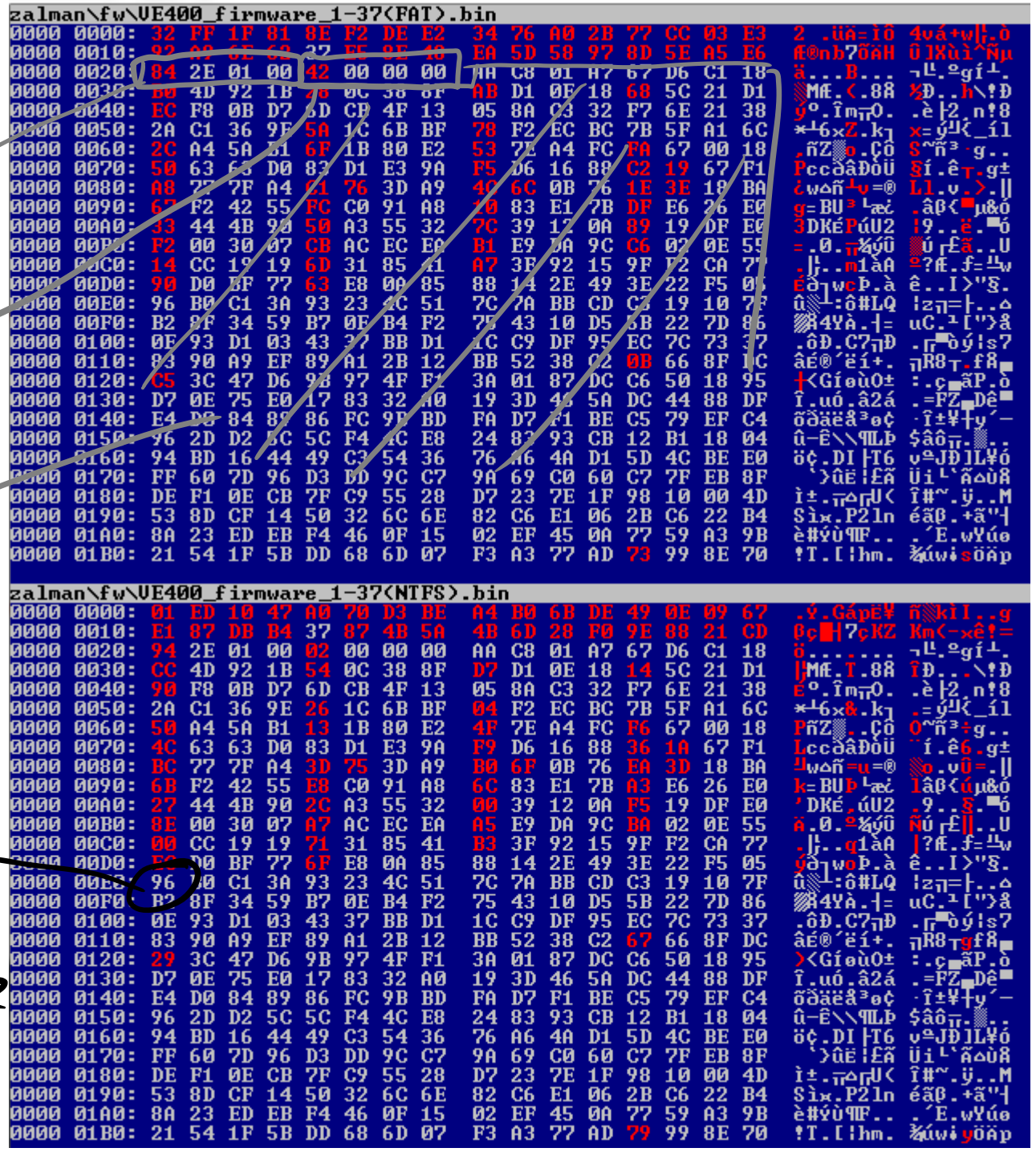
File Length

Flag?

Code

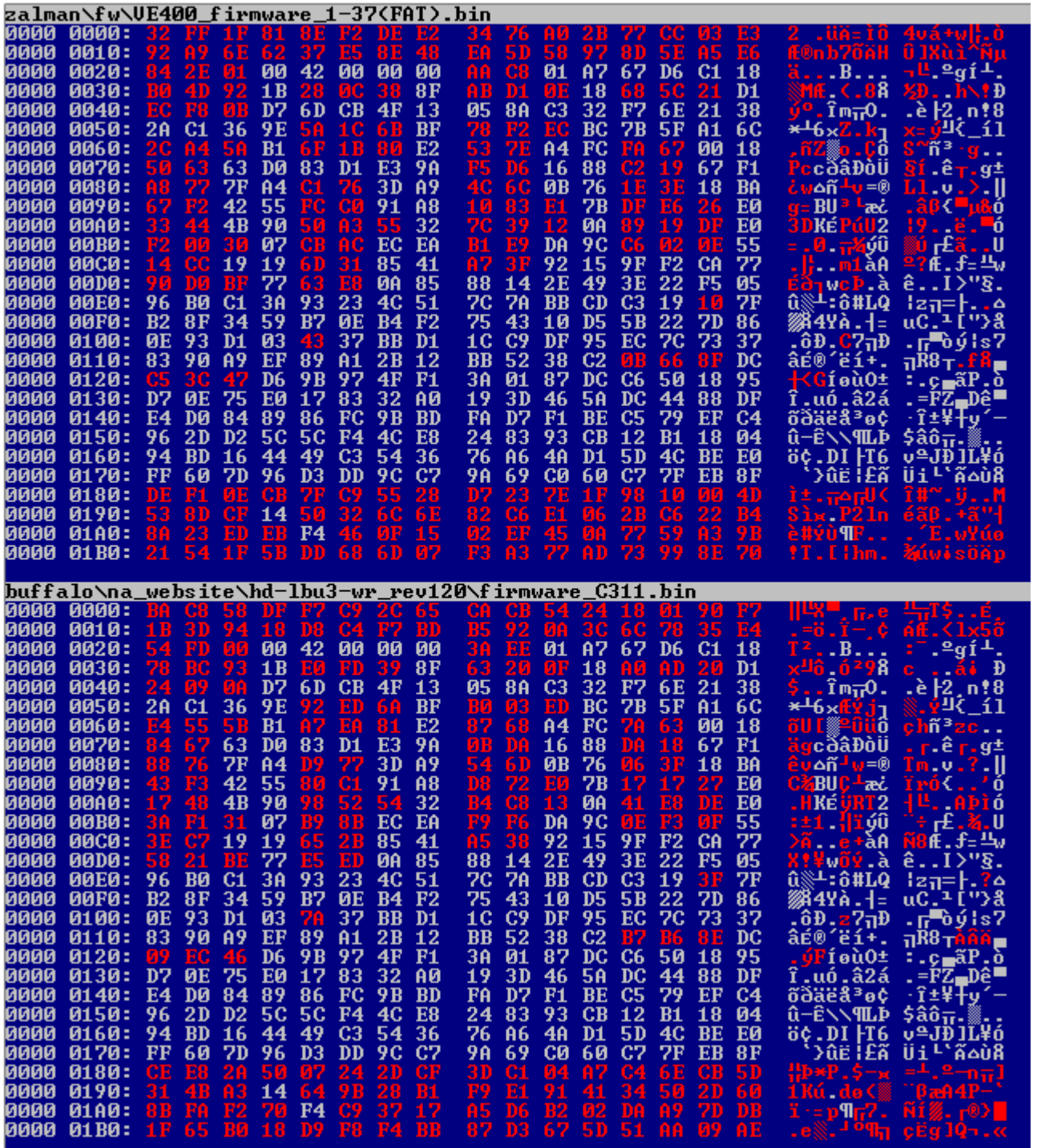
No difference in encrypted data

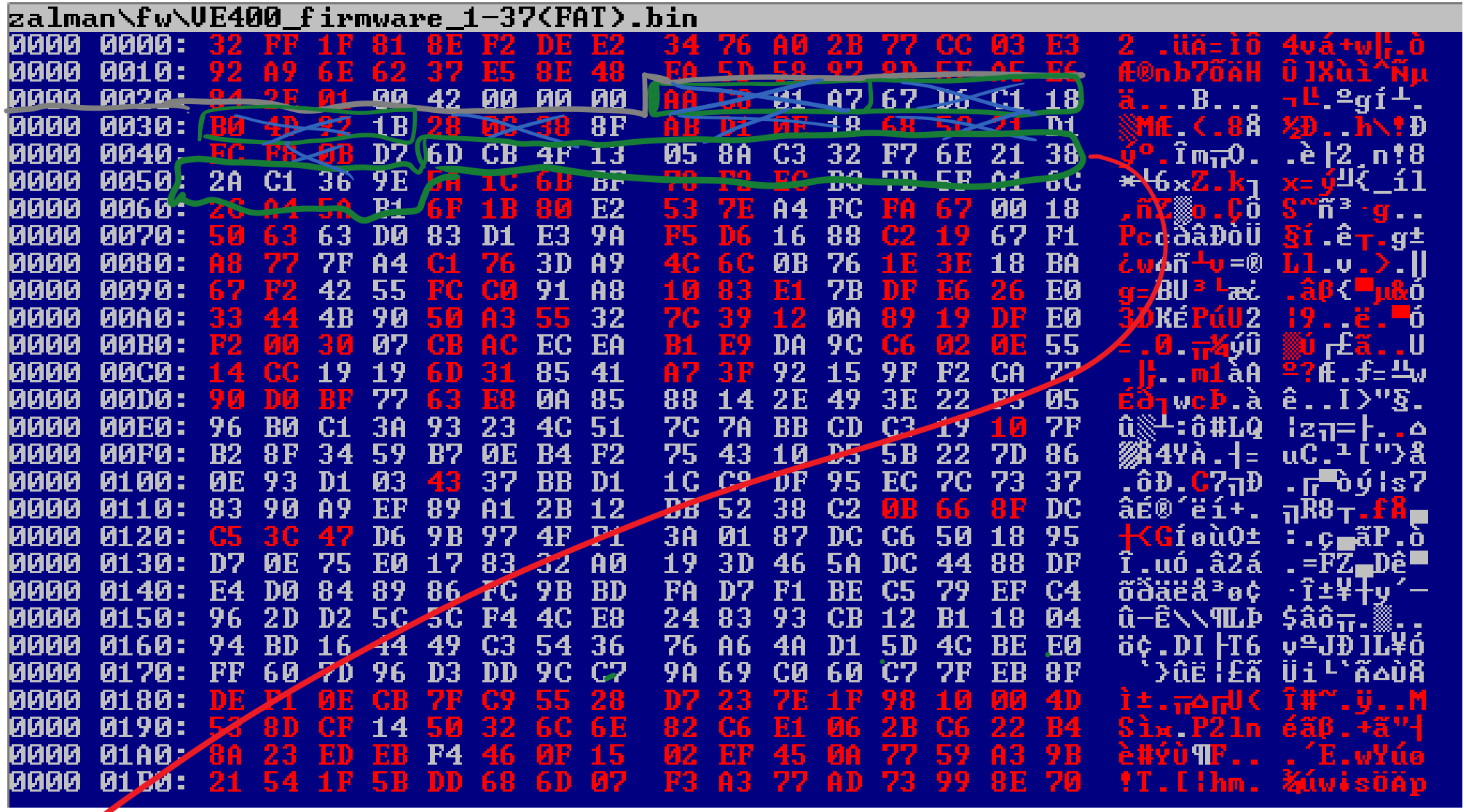
⇒ No diff in source at that byte.



Zalman →

Buffalo →
(other manufacture)





Blocks 8, 9, 10, 11 same (Block = 32 bits)

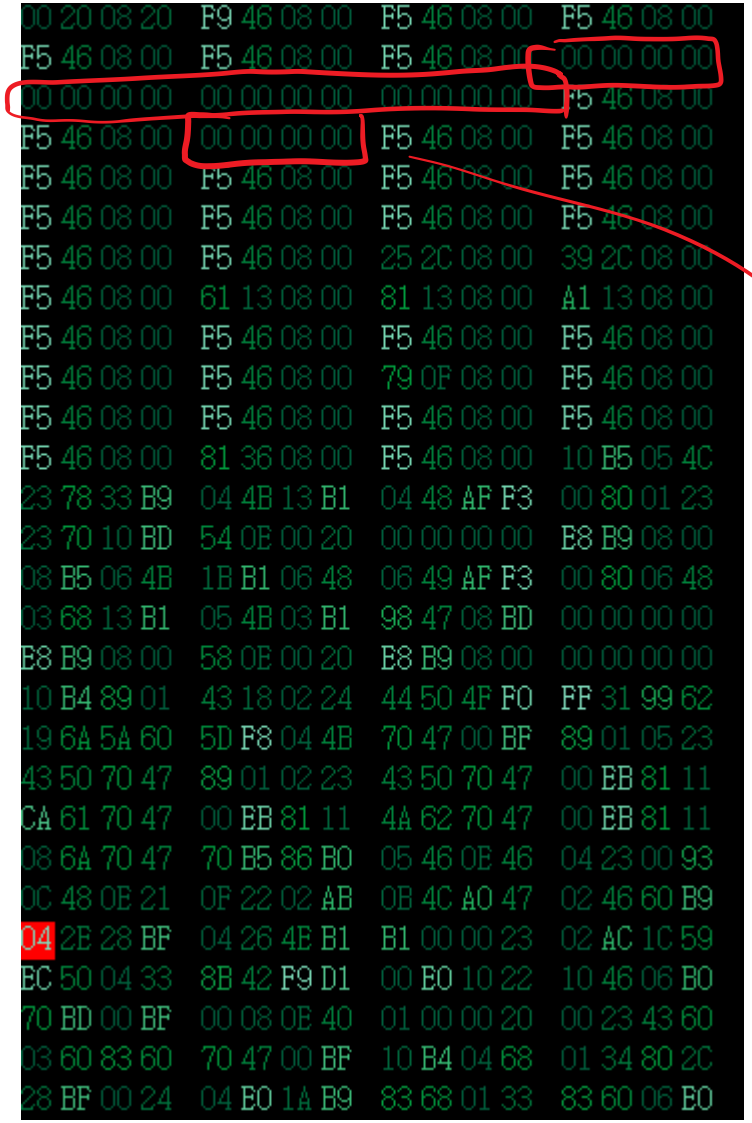


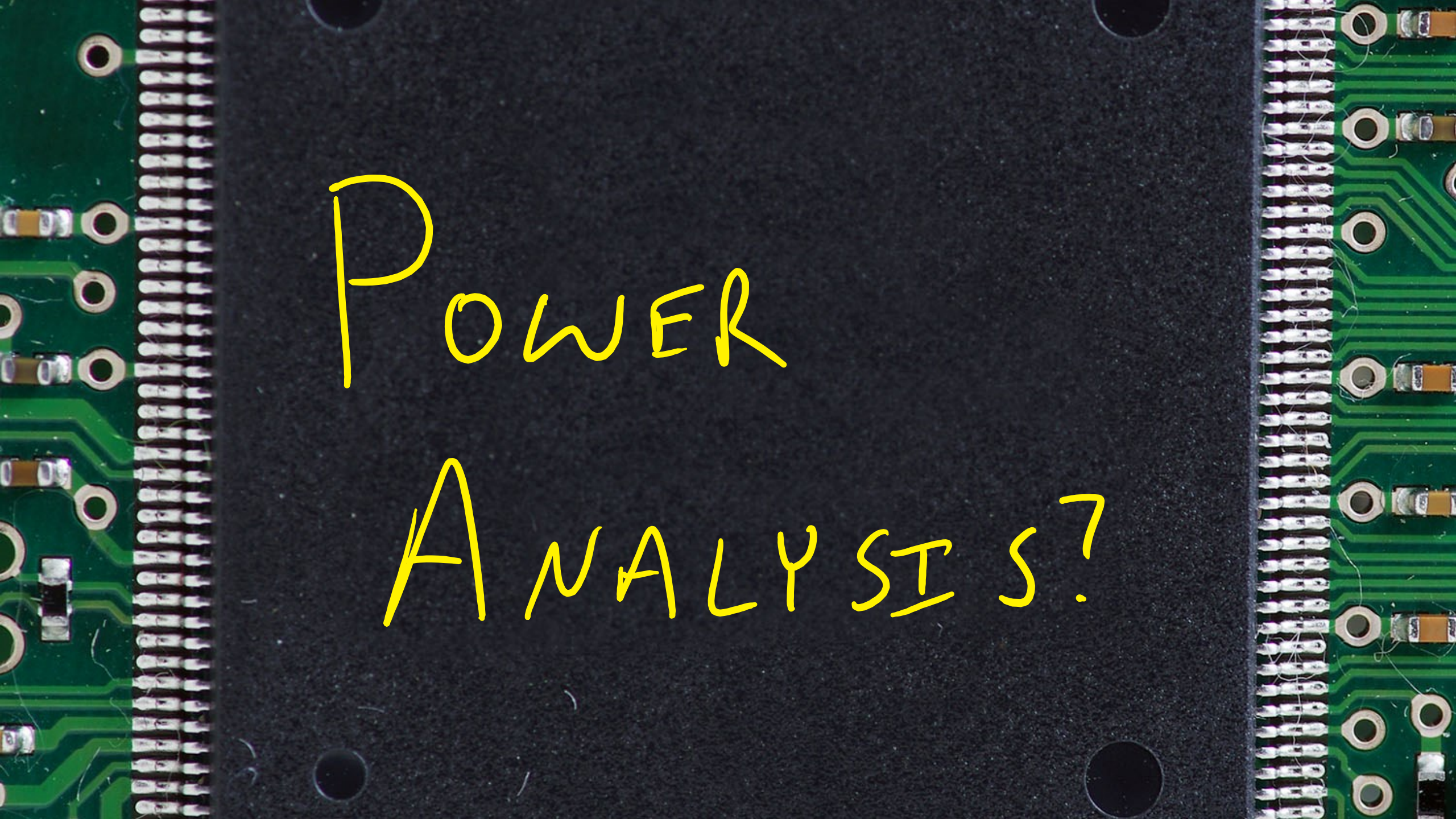
Figure 2.2. Vector table

Exception number	IRQ number	Offset	Vector
16+n	n	0x0040+4n	IRQn
.	.	.	.
.	.	.	.
.	.	.	.
18	2	0x004C	IRQ2
17	1	0x0048	IRQ1
16	0	0x0044	IRQ0
15	-1	0x0040	Systick
14	-2	0x003C	PendSV
13		0x0038	Reserved
12			Reserved for Debug
11	-5		SVCall
10		0x002C	
9			Reserved
8			
7			
6	-10		Usage fault
5	-11	0x0018	Bus fault
4	-12	0x0014	Memory management fault
3	-13	0x0010	Hard fault
2	-14	0x000C	NMI
1		0x0008	Reset
		0x0004	Initial SP value
		0x0000	

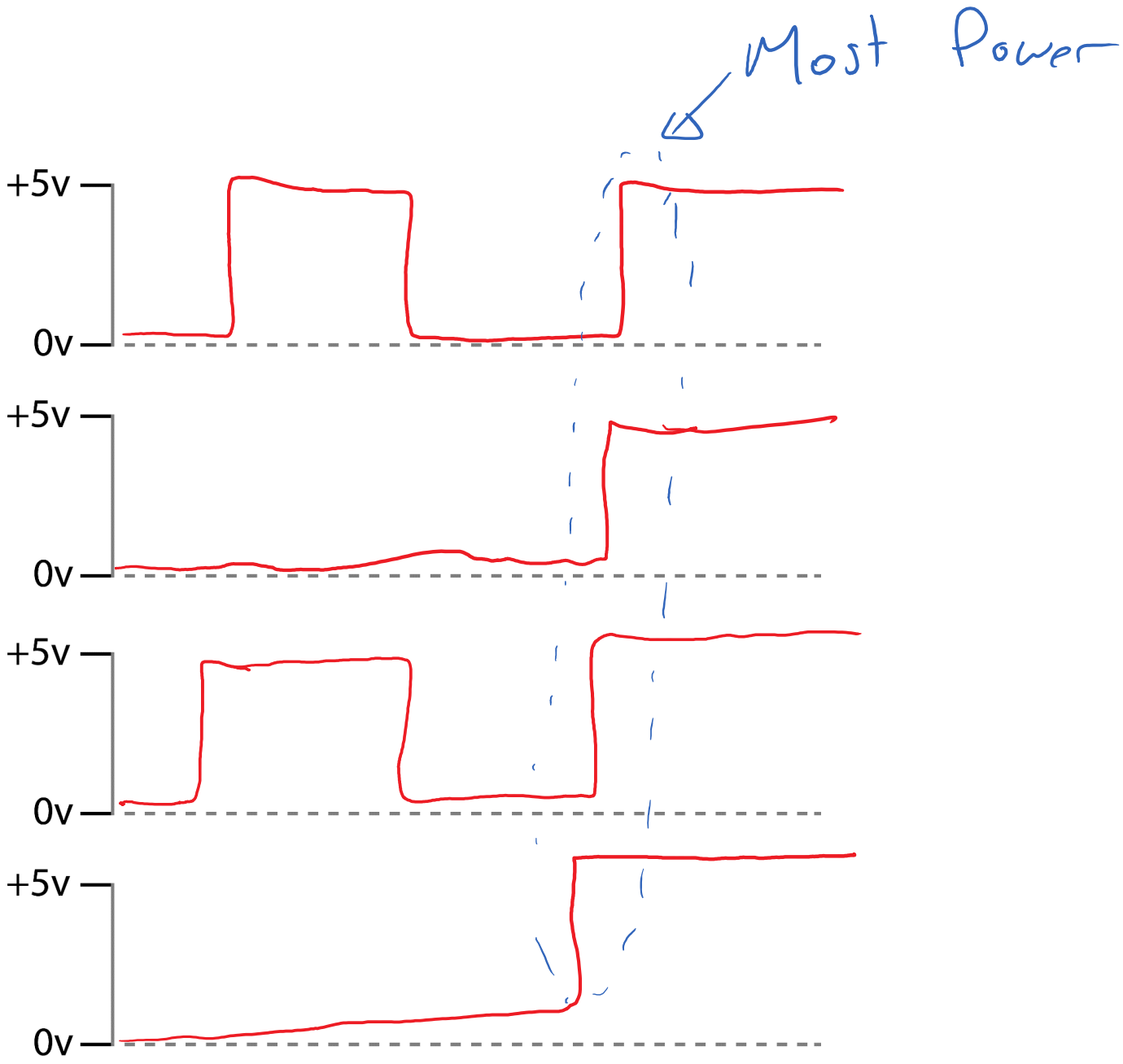
On system reset, the vector table is fixed at address 0x00000000.

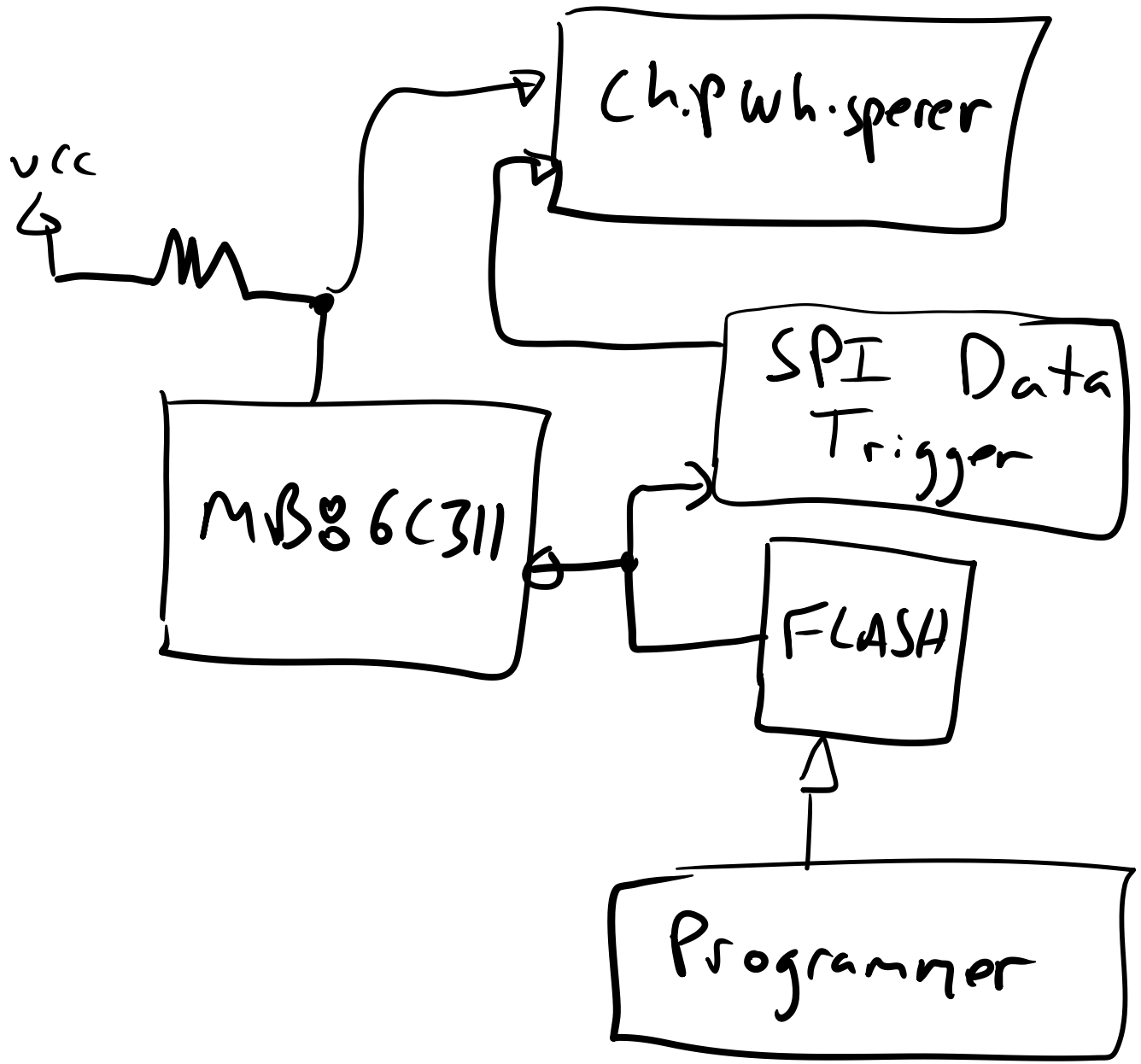
Knowledge?

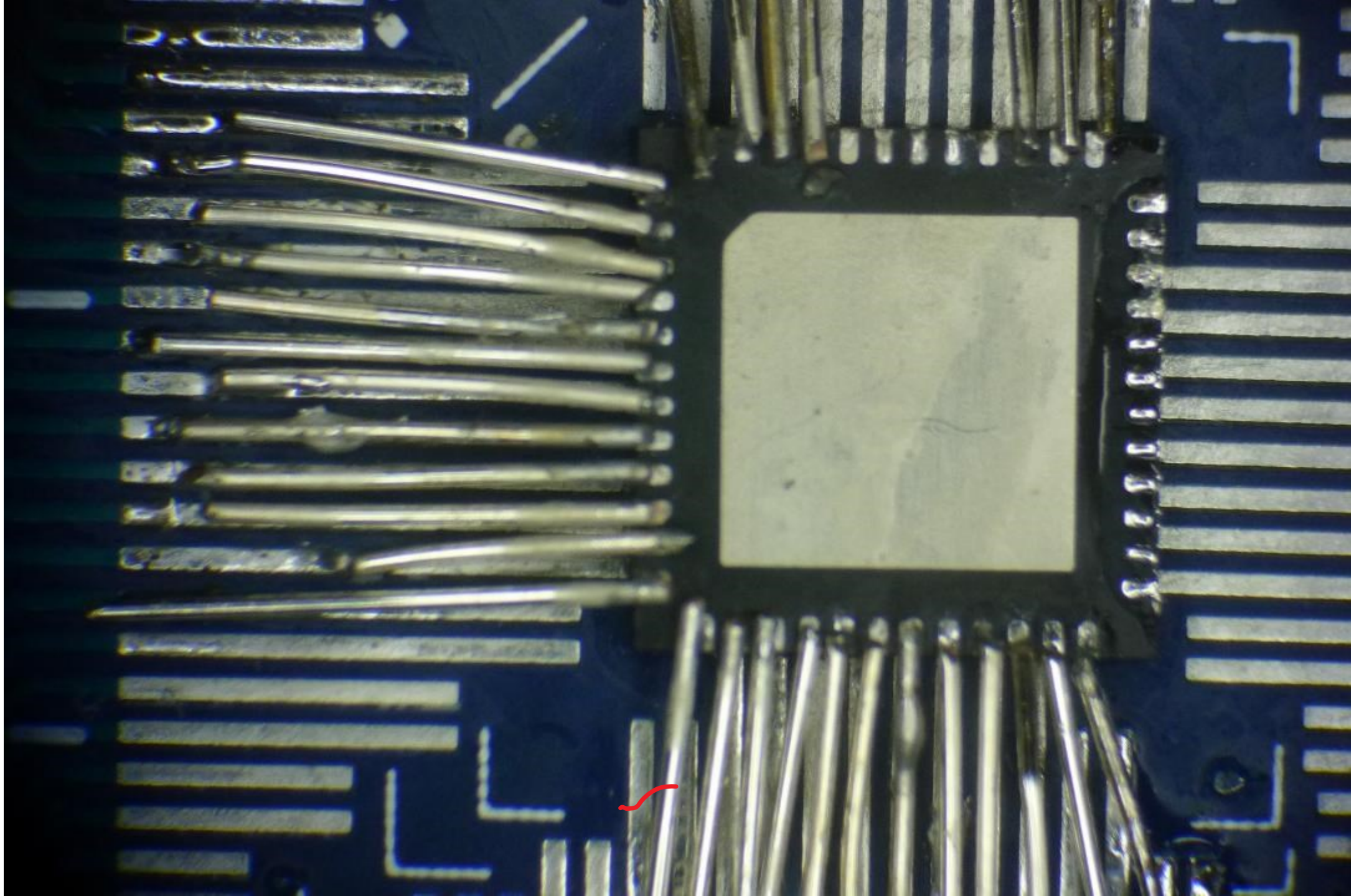
- Direct ARM Cortex M3 code, with stream cipher.
- Break stream cipher, all is lost.

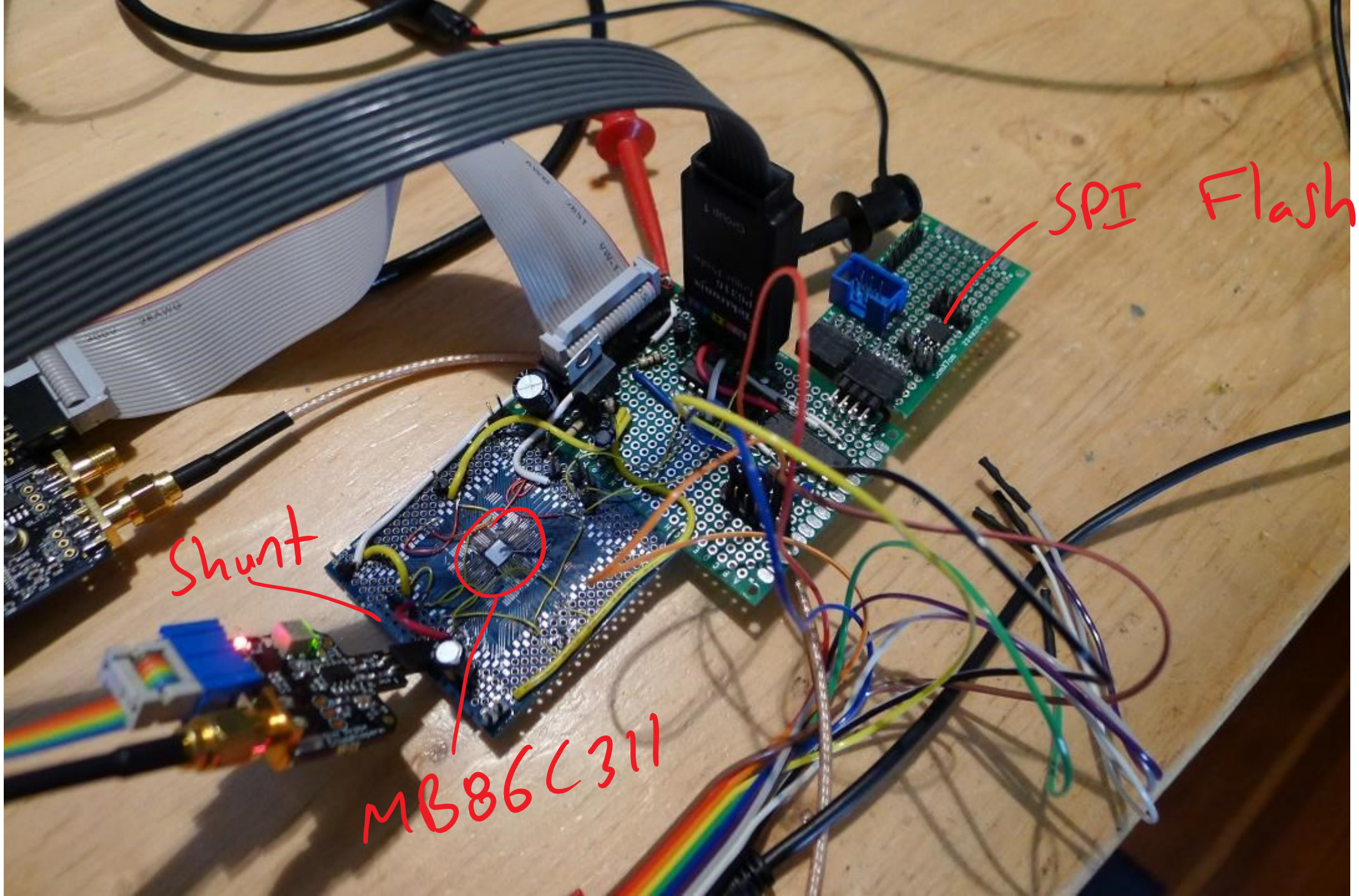

Power
Analysis?

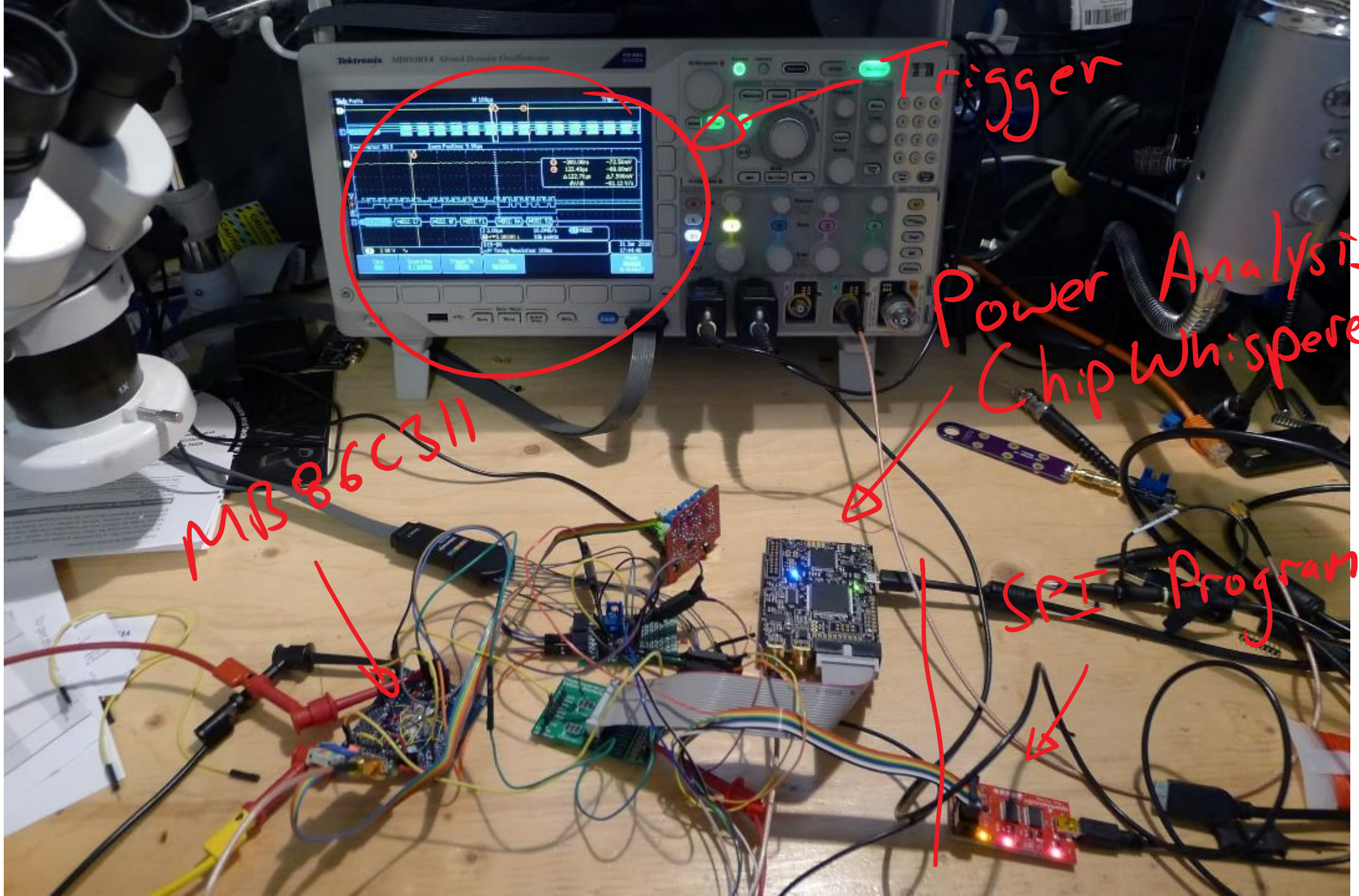
Data Busses...




Chipwhisperer
vcc
SPI Data Trigger
MSP430F311
FLASH
Programmer

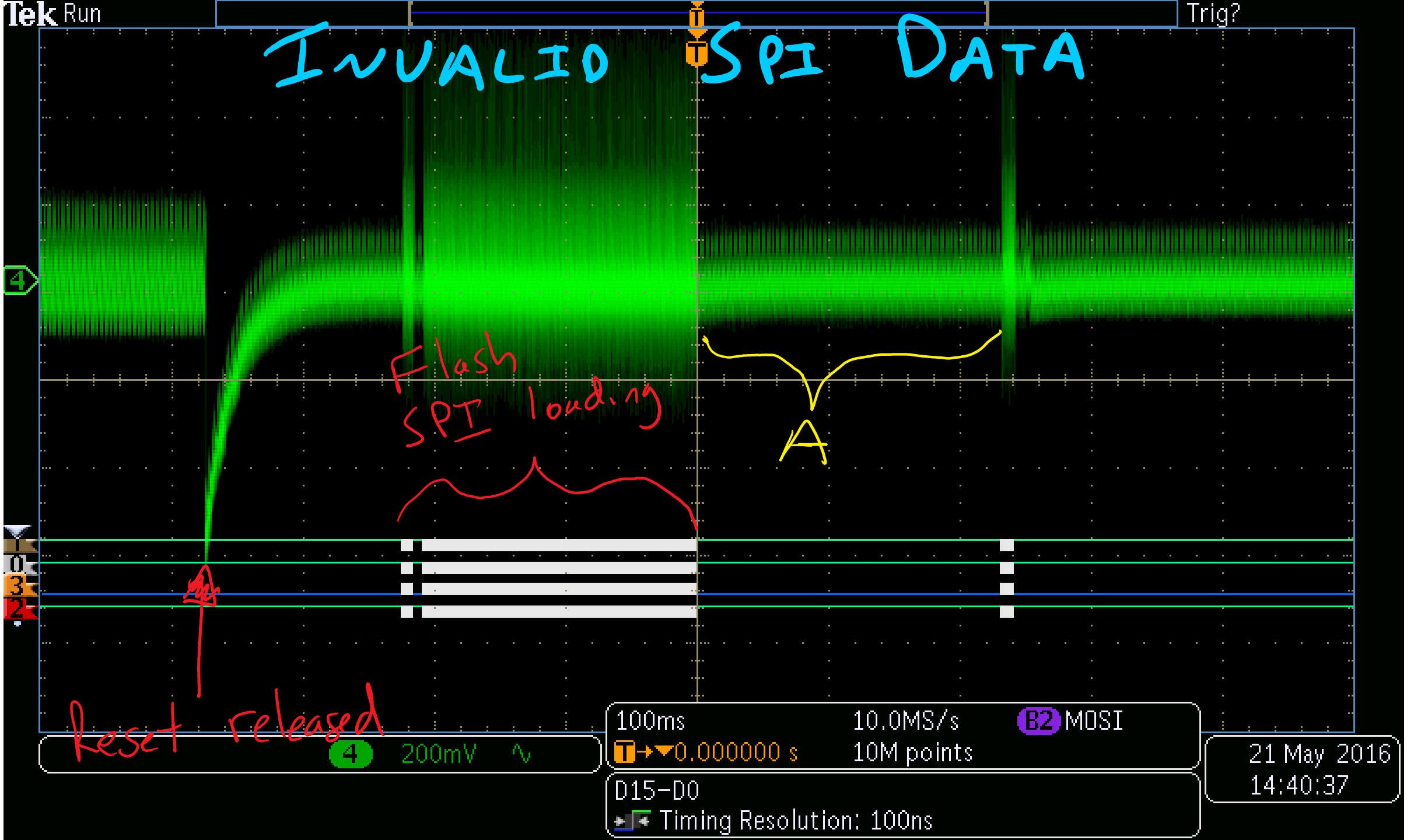


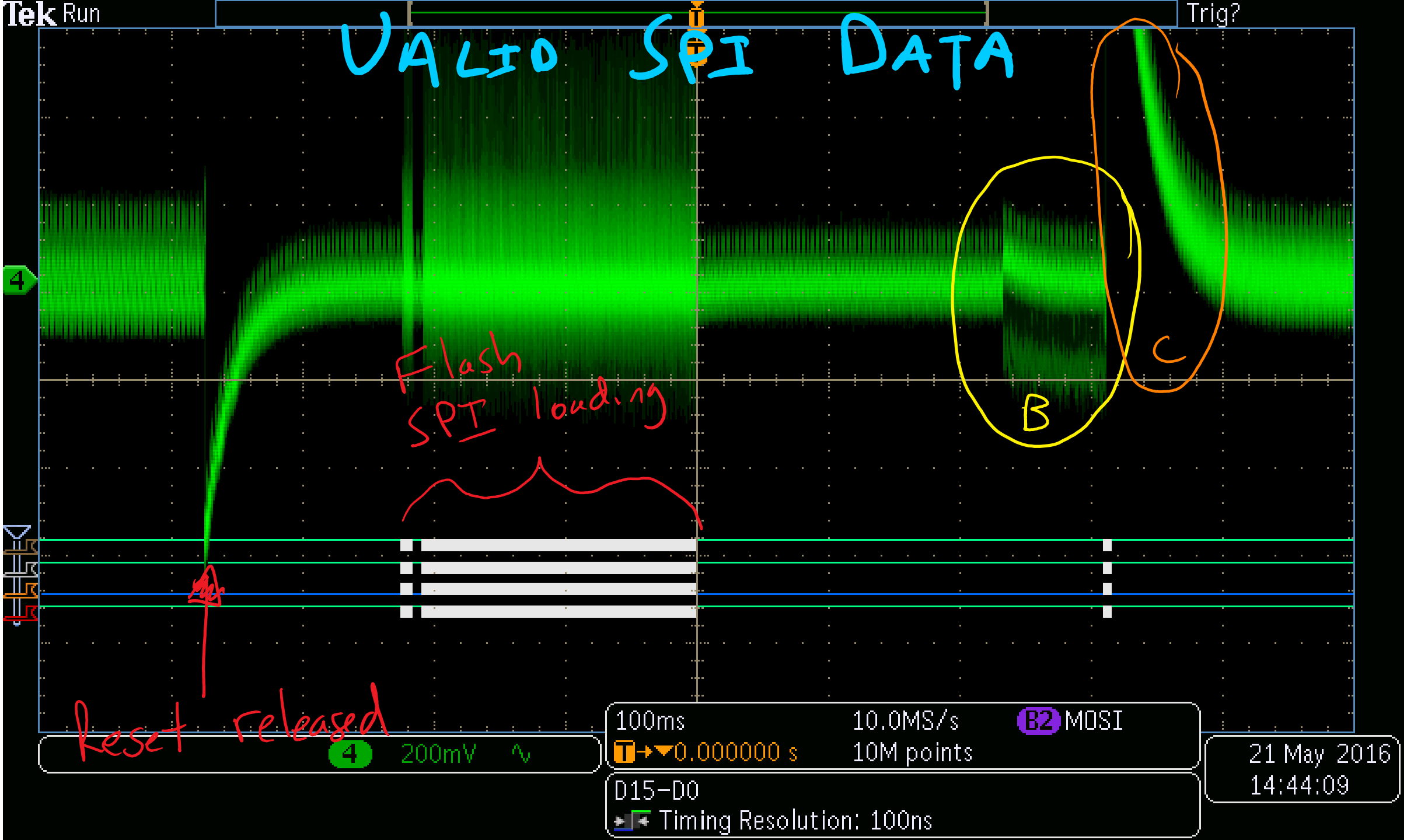

SPI Flash
Shunt
MB86C311


Trigger
Power Analysis
ChipWhisperer
MB86C311
SPI Programmer

Power Analysis

1. Program SPI Flash with pattern.
2. Reset MB86711, will reload FLASH.
3. Capture power at different places


Tek Run
Trig?
Invalid SPI Data
Flash SPI loading
A
reset released
4
200mV
100ms
10.0MS/s
B2 MOSI
0.000000 s
10M points
21 May 2016
14:40:37
D15-D0
Timing Resolution: 100ns


Tek Run
Trig?
Valid SPI Data
Flash SPI loading
B
C
Reset released
4
200mV
100ms
10.0MS/s
B2 MOSI
T 0.000000 s
10M points
21 May 2016
14:44:09
D15-D0
Timing Resolution: 100ns

Comparison

Invalid = Changed a few bytes

Valid = Original Image

1. Both complete operation "A"
2. Only valid code completes "B"
3. Assume "C" is switching to operating code, maybe powering on SATA etc?

LEARNINGS

As  when code valid

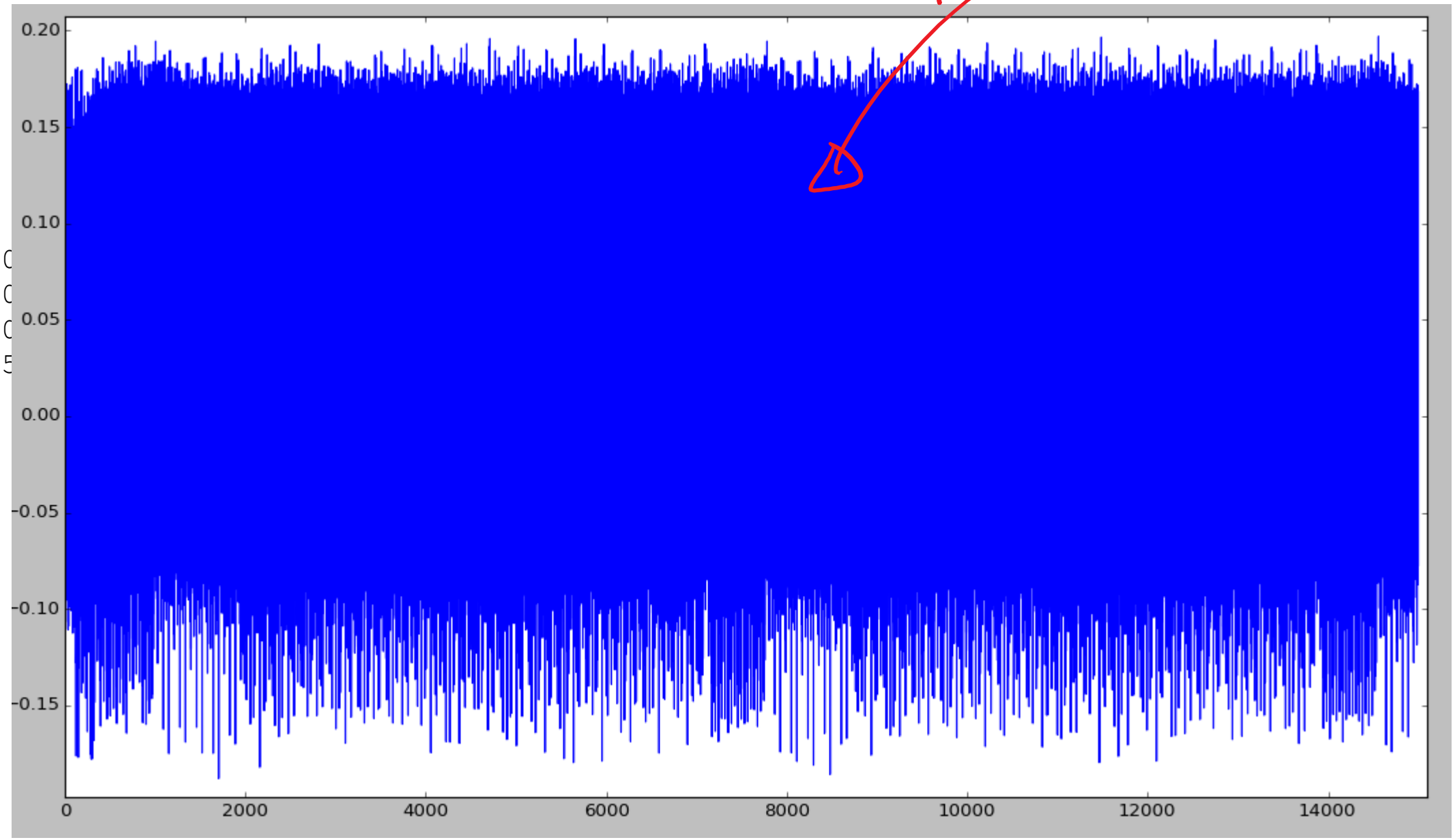
 when code invalid

 = Validation (Hash?)


Tek Run
Trig?
Experiment with "known" section where plaintext = 0's
SPI(MOSI)
I: B9
MOSI: 6B
MOSI: 1D
MOSI: E7
MOSI: FF
MOSI: FF
MOSI: FF
MOSI: FF
2.00µs
2.50GS/s
B1 MOSI
0.000000 s
1M points
D15-D0
Timing Resolution: 2.00ns
2 Feb 2016
14:22:31
Type
Bus
Source Bus
B1 (SPI)
Trigger On
MOSI
Data
6B 1D E7h
Mode
Normal
& Holdoff

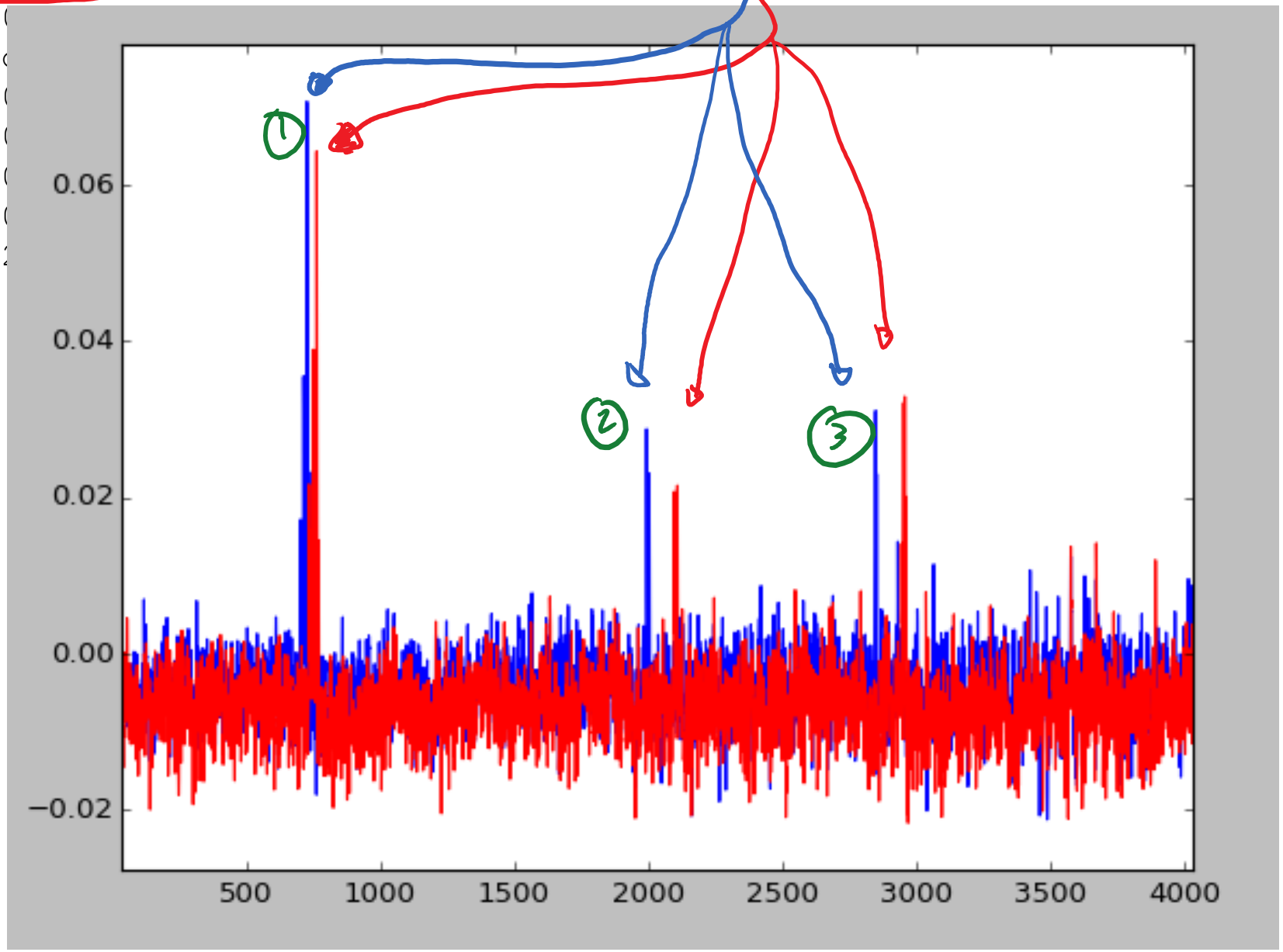
```
0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1:  ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00
2:  00 00 00 00 ff ff ff ff 00 00 00 00 00 00 00 00
3:  ff 00 00
4:  00 ff 00
5:  00 00 ff
6:  00 00 00
7:  70 0e 75
8:  8f f1 8a
9:  00 00 00
10: 00 00 00
11: 00 00 00
12: 00 00 00
13: a3 3c f5
```

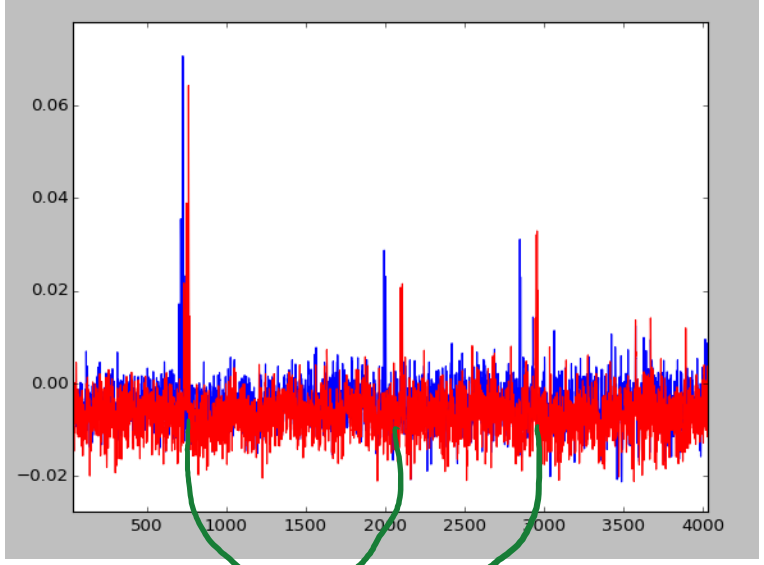
Raw power trace



```
 0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 1: ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00
 2: 00 00 00 00 ff ff ff ff 00 00 00 00 00 00 00 00
 3: ff 00 00 00 (
 4: 00 ff 00 00 (
 5: 00 00 ff 00 (
 6: 00 00 00 ff (
 7: 70 0e 75 ad (
 8: 8f f1 8a 52 (
 9: 00 00 00 00 2
10: 00 00 00 00
11: 00 00 00 00
12: 00 00 00 00
13: a3 3c f5 54
```

Difference Traces

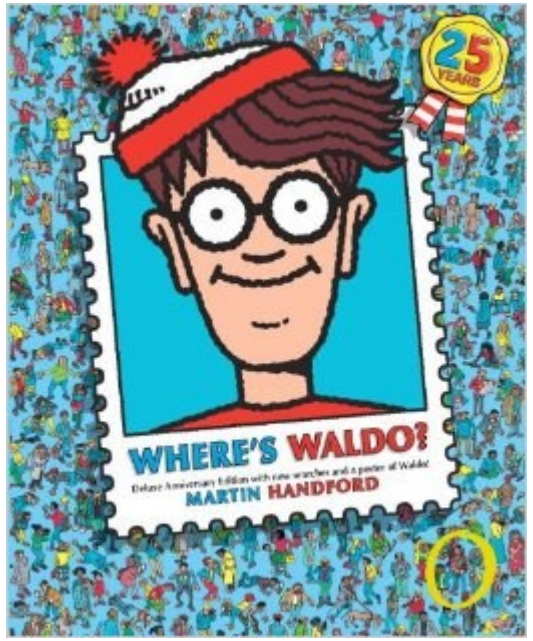




Multiple Manipulations

↳ Some sort of feedback?

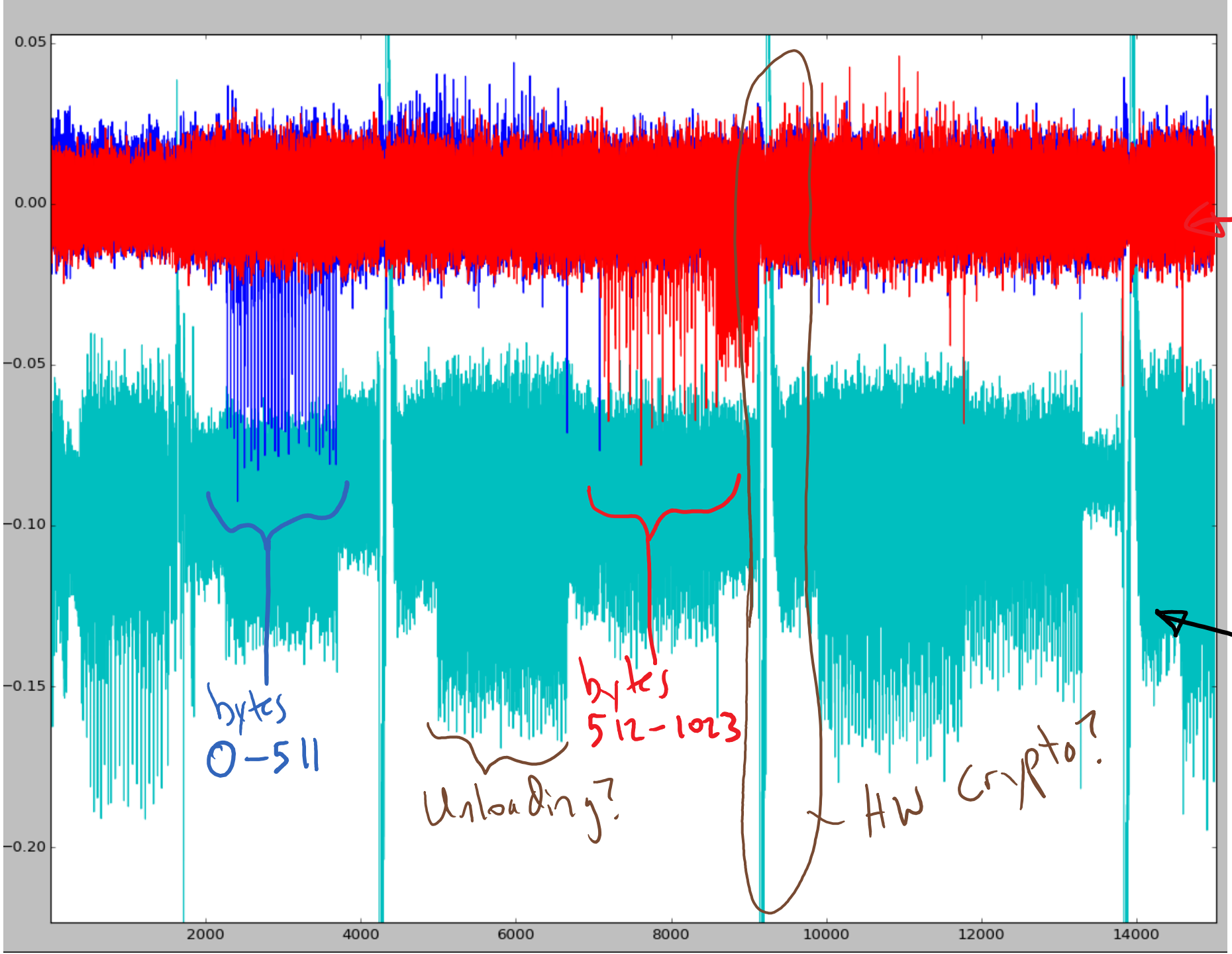
FINDING HASH




```python
#Simple python script to try a bunch of hashes
import hashlib
raw_data = open('firmware_C311.bin', 'rb').read()
known_hash = raw_data[0:32]

test_hash = hashlib.algorithms_available

for startoffset in range(0x20, 0x28, 4):
    #Trim bytes off end, seems to be some header at end?
    for endoffset in range(0, -1024, -1):
        firmware = raw_data[startoffset:endoffset]
        for t in test_hash:
            h = hashlib.new(t)
            h.update(firmware)
            dig = h.hexdigest()

            #This example uses known hash, NOT read from start of file
            #If searching in middle, do:
            #if "77badb" in dig:
            if dig.startswith("77badb"):
                print dig
                print "***HASH FOUND?***"
                print "start=0x%02x, end=0x%02x, type=%s"%(startoffset, endoffset, t)
```

Build Signatures

```python
import hashlib

fd = open('lockdown_hdd1.bin', 'rb')
orig_flash = fd.read()
fd.close()

#As an example - write 0xff's in blanks
programmed_file = [0xff]*16

newdata = orig_flash[:]
newdata = [ord(d) for d in newdata]
for i in range(0x2044, 0x2044+16):
    newdata[i] = programmed_file[i-0x2044]

#Calculate new hash
newdata_temp = ''.join([chr(j) for j in newdata])
h = hashlib.new("sha256")
h.update(newdata_temp[0x2020:0x170F8])
shash = h.hexdigest()

#Add hash to file
i=0
for t in range(0, len(shash), 2):
    newdata[0x2000+i] = int(shash[t:(t+2)], 16)
    i += 1

#Save file
newdata = ''.join([chr(j) for j in newdata])

fd = open('lockdown_temp.bin', 'wb')
fd.write(newdata)
fd.close()
```



- 128 words per “block”
  - 512 bytes

Difference traces

Decryption power trace

FIRMWARE ATTACKS

1) Can cause arbitrary code to be decrypted.
2) Can move interrupt/reset vectors quazi-blindly.
3) If could determine either (a) part of stream cipher or (b) part of code (init routines?) may be able to build short "dumper".

Hard Drive Summary

1. No secrets inside silicon
2. Secure and normal variants of MB86C311 use same FW protection.

∴ Any company using these devices could decrypt this FW.

Hard Drive Summary

3. Once F.W. is known, all is lost. Possible to brute-force password in < 1s.

4. Various attack vectors to FW itself due to flaws.

Questions?

Colin O'Flynn

@colinoflynn

oflynn.com

newae.com

coflynn@newae.com