

A Lightbulb Worm?

A teardown of the Philips Hue.



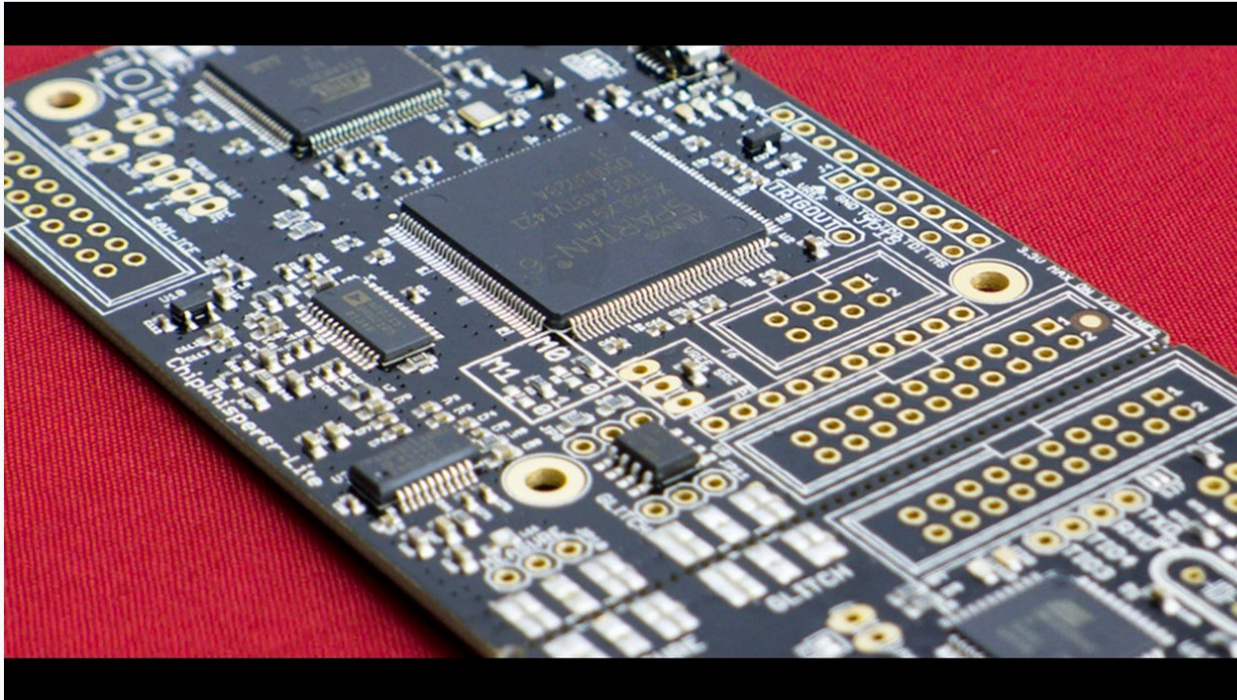
Colin O'Flynn

(with special appearance by Eyal Ronen)



ABOUT ME

ChipWhisperer-Lite: A New Era of Hardware Security Research



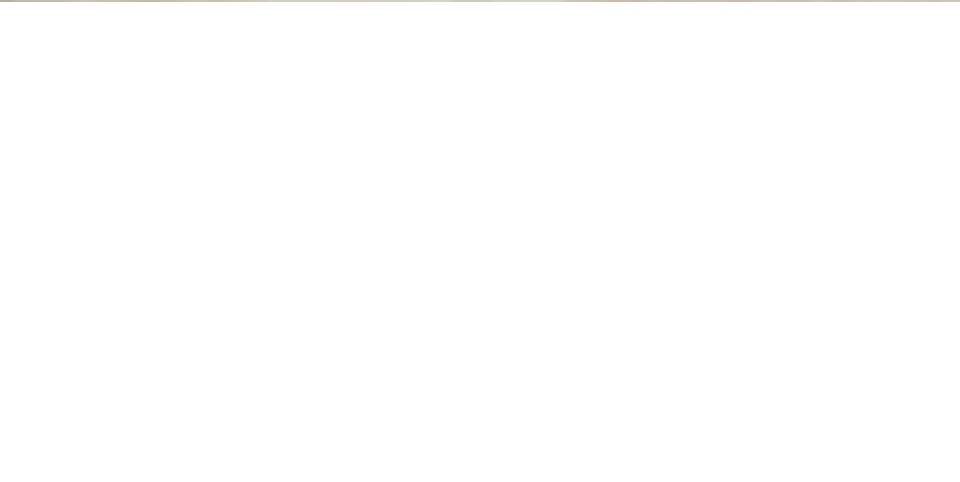
Embedded security - is it an oxymoron? Learn the truth through a series of hands-on labs targeting computer and electrical engineers.

Created by

Colin O'Flynn



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



HACKS?

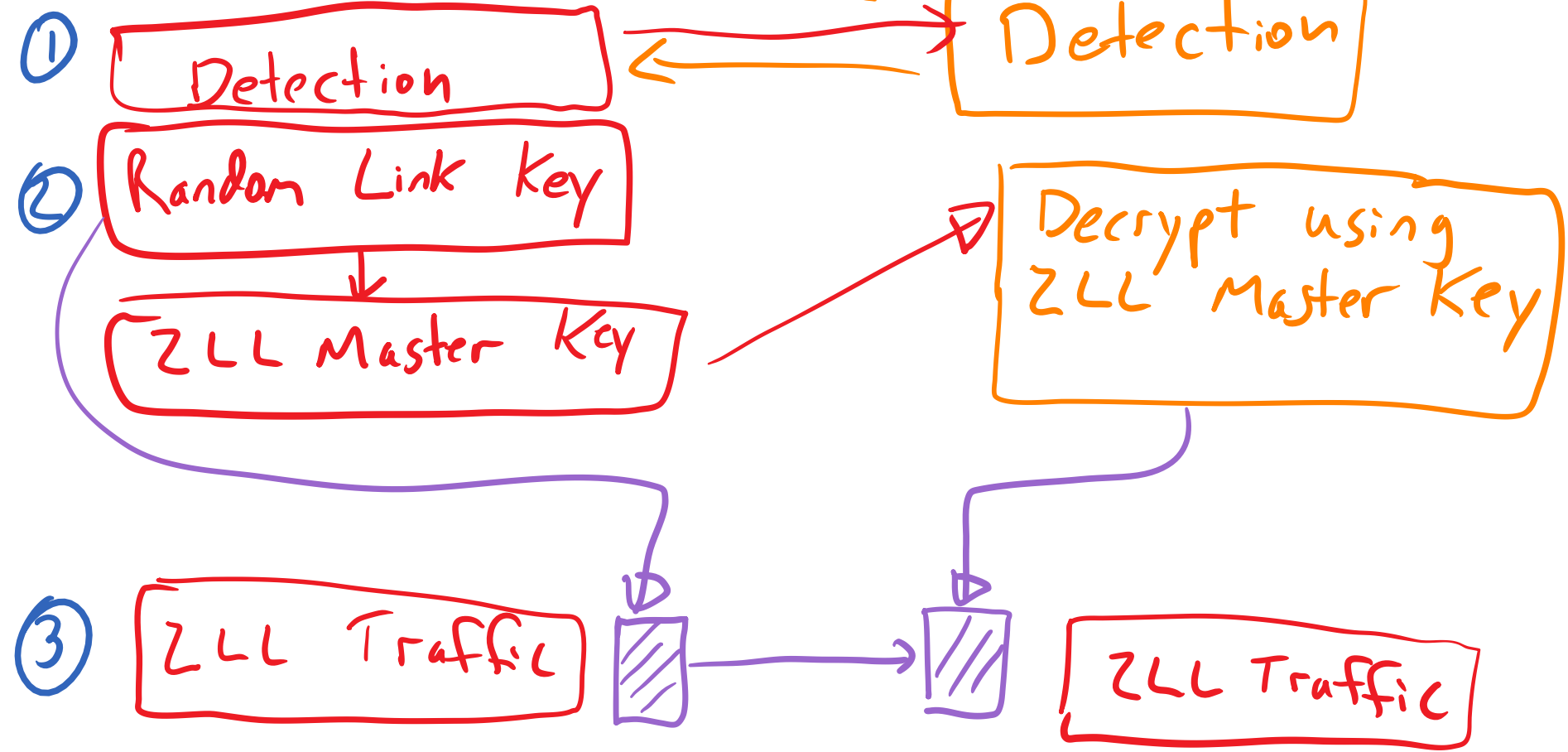
- ① Brick light-bulb by OTA firmware update.
- ② Move bulb onto unavailable network, or control bulb.
- ③ Hack into bridge, access ethernet.
- ④ Malware in bulbs to do #3?

Understanding ZLL

BRIDGE



New Bulb



LIGHT BULB THEFT

6.4.4 Stealing a Node

A node that is already part of a ZLL network can be taken or 'stolen' by another ZLL network using Touchlink (in which case, the stolen node will cease to be a member of its previous network). This transfer can only be performed on a node which supports one or more Lighting devices (and not Controller devices).

The node is stolen using an initiator in the new network, e.g. from a remote control unit. The 'stealing' process is as follows:

1. The initiator sends a Scan Request to nodes in its vicinity. The required function is:

eCLD_ZIICommissionCommandScanReqCommandSend()

2. A receiving ZLL node replies to the Scan Request by sending a Scan Response. The required function is:

eCLD_ZIICommissionCommandScanRspCommandSend()

3. The initiator receives Scan Responses from one or more nodes and, based on these responses, selects a node (containing a Lighting device) that is already a member of another ZLL network.
4. The initiator then sends a Reset To Factory New Request to the desired node. The required function is:
eCLD_ZIICommissionCommandFactoryResetReqCommandSend()
5. On receiving this request on the target node, the event `E_CLD_COMMISSION_CMD_FACTORY_RESET_REQ` is generated and the function **ZPS_eAplZdoLeaveNetwork()** should be called. In addition, all persistent data should be reset.
6. The node can then be commissioned into the new network by following the process in [Section 6.4.2](#) from Step3.

Reply based on signal power, should only work at short distances.

8.1.2 Channels

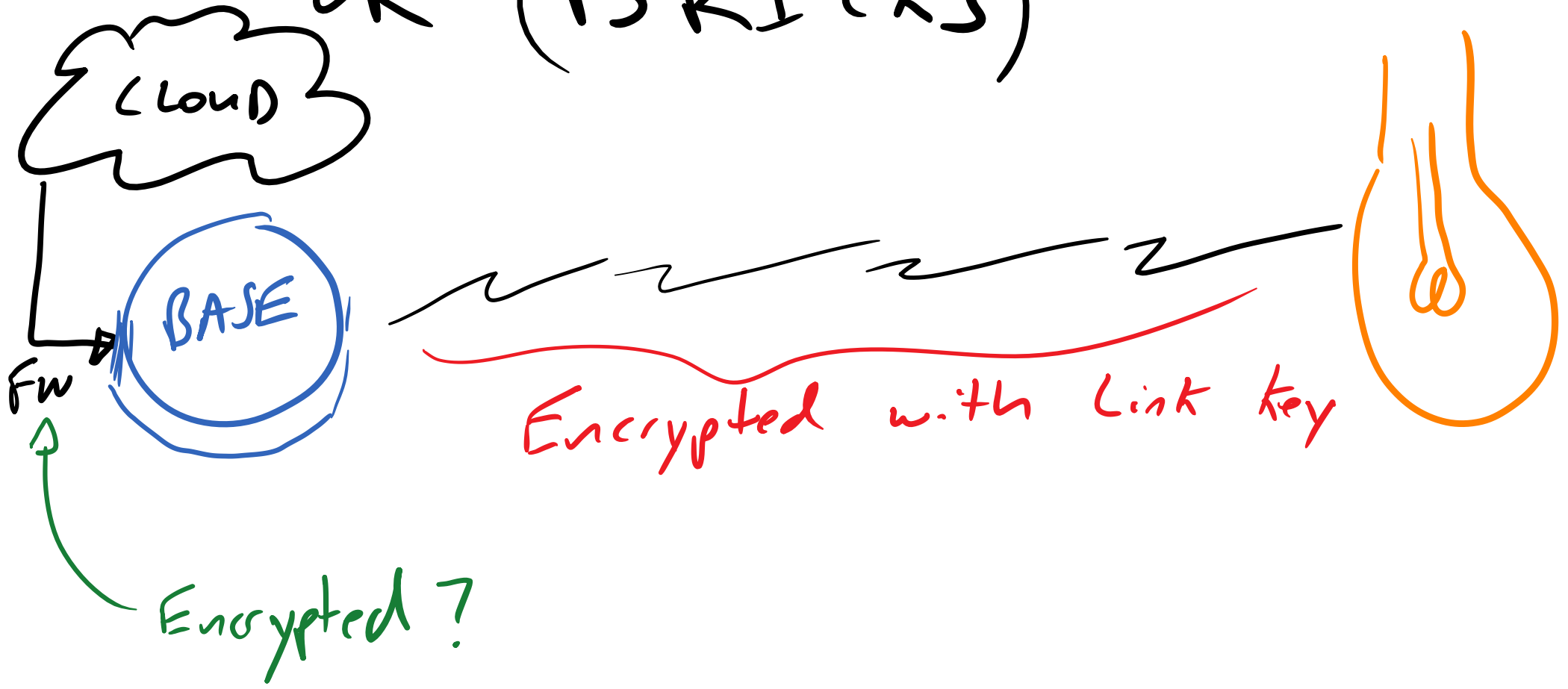
A ZLL device shall be able to operate on all channels available at 2.4GHz, numbered from 11 to 26. When operating on channel 26, the transmission power may be reduced in order to comply with FCC regulations.

Within this range, two sets of channels shall be defined. The *primary* ZLL channel set shall consist of channels 11, 15, 20 and 25 and shall be used in preference for commissioning and normal operations. The *secondary* ZLL channel set shall consist of channels 12, 13, 14, 16, 17, 18, 19, 21, 22, 23, 24 and 26 and can be used as a backup to allow the ZLL device to connect to a non-ZLL network.

Demo by Eyal Ronen

See <http://www.wisdom.weizmann.ac.il/~eyalro/>

LIGHT BULB MALWARE OR (BRICKS)




LIGHT BULB MALWARE

- 1) ZLL key leaked. We know it's possible to "steal" bulbs.
- 2) Custom FW on bulbs could turn bulb into "bridge" that searches for & steals nearby bulbs.
- 3) IF could cause other bulbs to perform OTA FW update → WORM

CHEAP

BULLBS

A close-up photograph of a hand holding a white LED light bulb. The bulb is held by the base, which is a standard E26 screw-in base. The main body of the bulb is white and has a frosted appearance. On the lower part of the white body, there is printed text in black ink. The background is a wooden surface, possibly a desk, with some blurred objects like a keyboard and a ruler visible on the left side.

800 Lumen

Hue white A19 9.5W 90mA
110-130Vac 50/60Hz
FCC ID: O3M9290011369X
IC: 10469A-1369X
Model: 9290011369



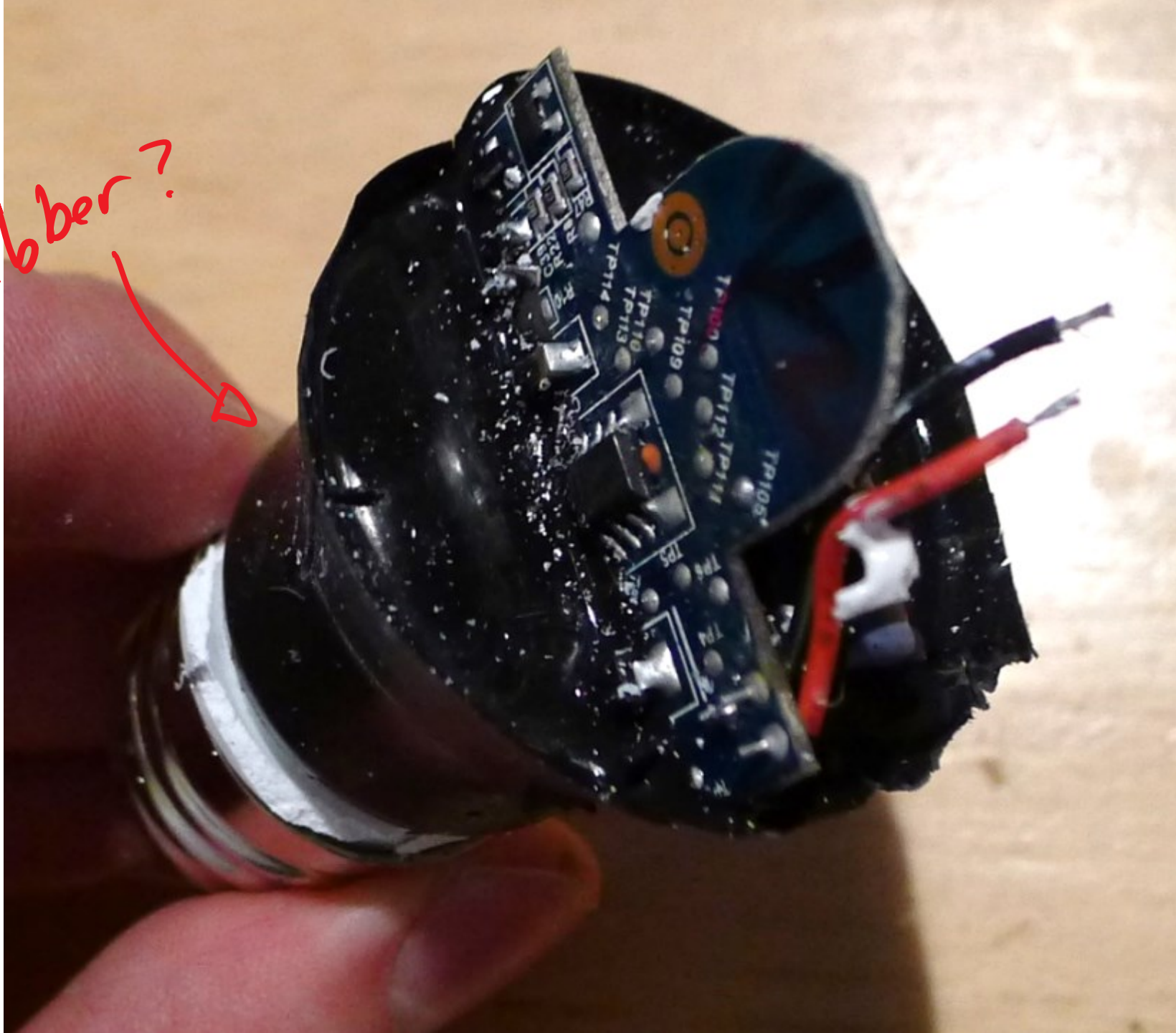
Heat sink

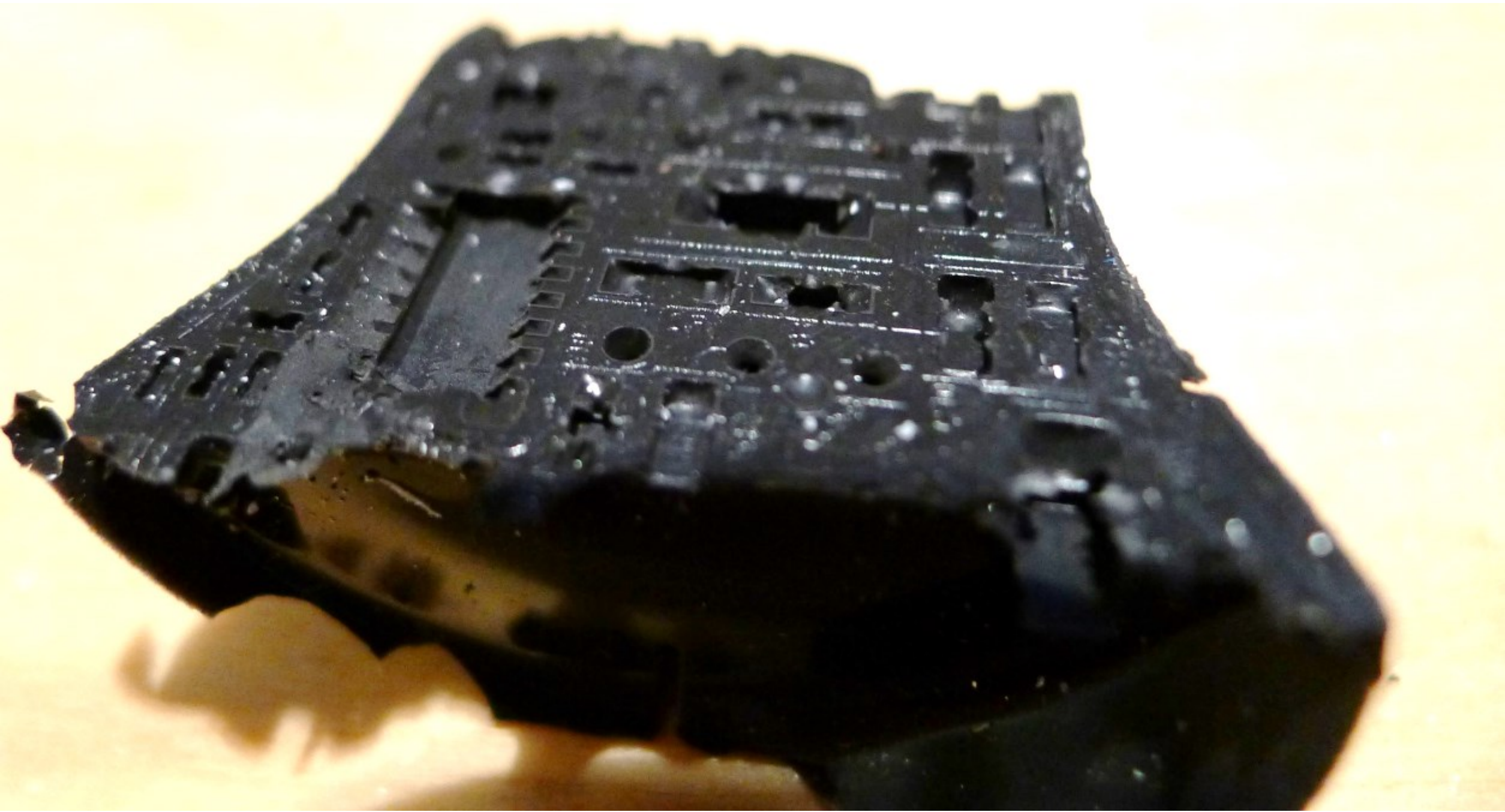
Antenna

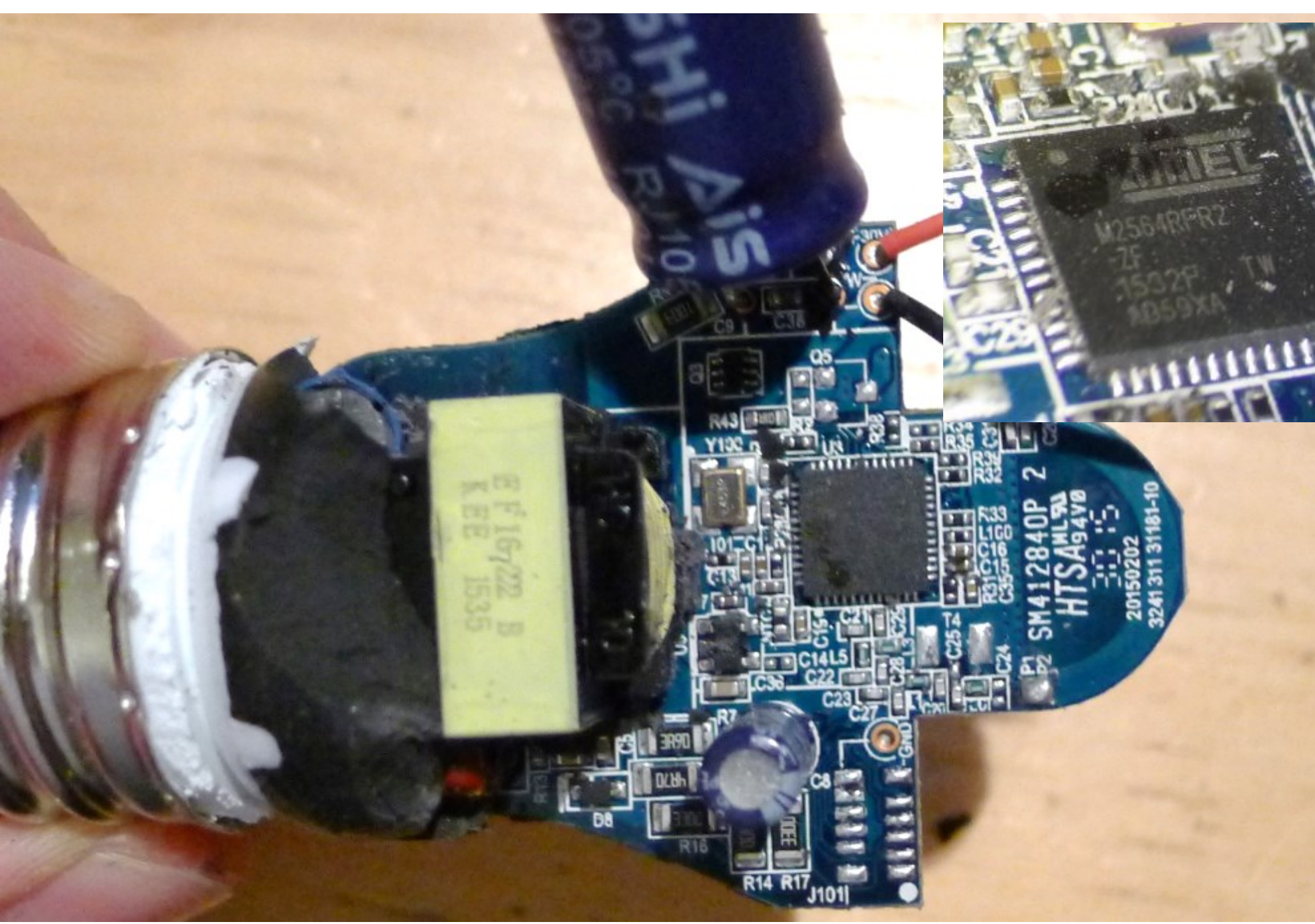
800 Lumen
Power: 8.5W
Voltage: 85-265V AC
Frequency: 50/60Hz
Color: Warm White
Beam Angle: 120°

SM412840P 2
HTSA
3015
3015
3015

Rubber? →

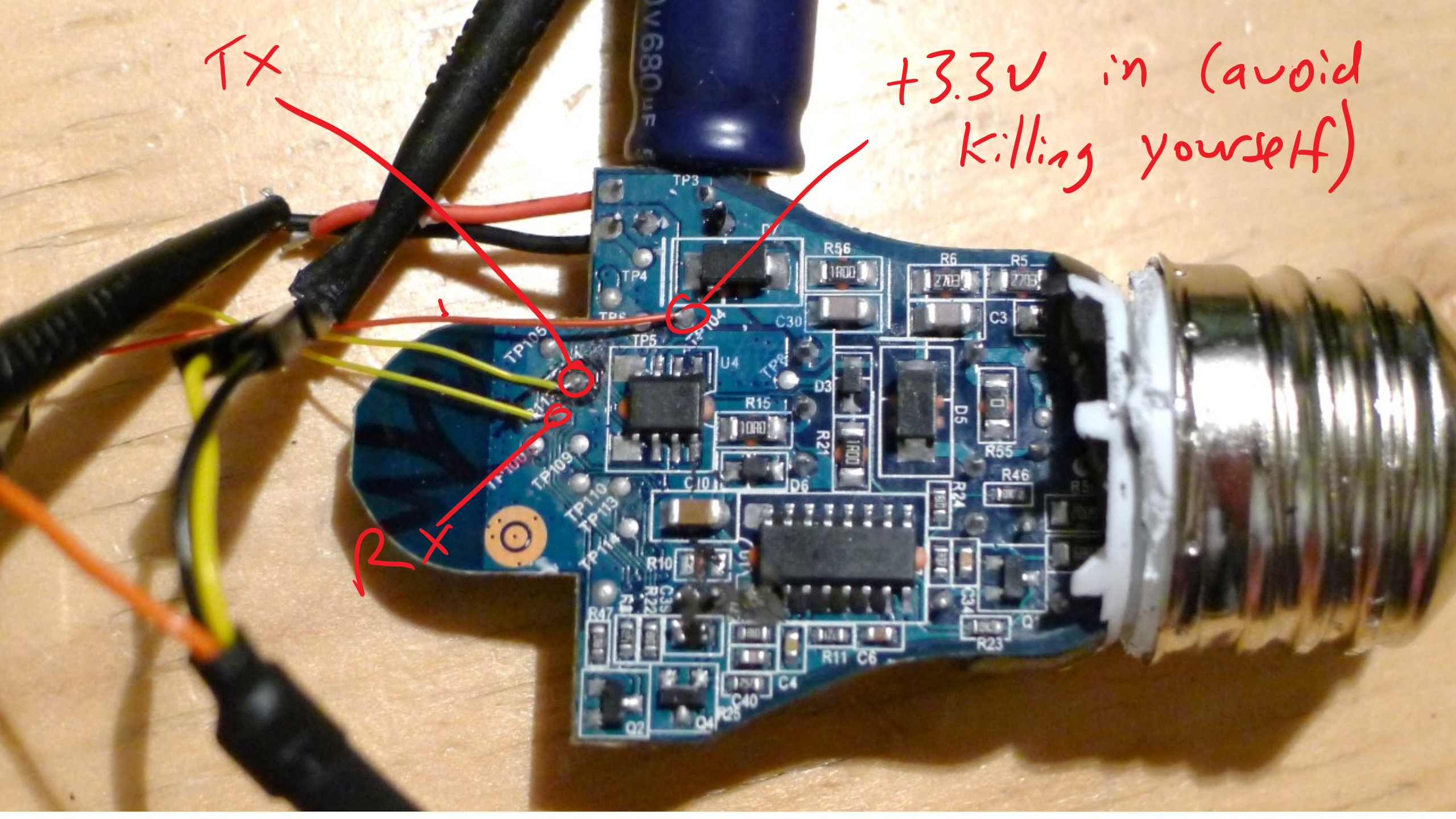






TX

+3.3V in (avoid
killing yourself)



Locked

```
[Log, Info, ConnectedLamp, MCUCR=0x00, LockBits=0xFC, LowFuse=0xF6, HighFuse=0x9A, ExtFuse=0xFE]
[Log, Info, ConnectedLamp, devsig=0x1EA803]
[Log, Info, S_DeviceInfo, Booting into normal mode...]
[Log, Info, S_DeviceInfo, DeviceId: Bulb_A19_DimmableWhite_v2]
[Log, Info, N_Security, LIB4.5.75]
[Log, Info, N_Security, KeyBitMask, 0x0012]
[Log, Info, ConnectedLamp, Platform version 0.41.0.1, package_ZigBee
117, package_BC_Stack 104, svn 26632]
[Log, Info, ConnectedLamp, Product version WhiteLamp-Atmel 5.38.1.15095, built
by LouvreZLL]
[Log, Info, A_Commissioning, Factory New at Ch: 11]
[TH, Ready, 0]
```

COM32 115200 bps, 8N1, no handshake

Settings

Clear

```
[00] YYYYYYY
[Log, Info, ConnectedLamp, MCUCR=0x00, LockBits=0xFC, LowFuse=0xF6, HighFuse=0x9A, ExtFuse=0xFE]
[Log, Info, ConnectedLamp, devsig=0x1EA803]
[Log, Info, S_DeviceInfo, Booting into normal mode...]
[Log, Info, S_DeviceInfo, DeviceId: Bulb_A19_DimmableWhite_v2]
[Log, Info, N_Security, LIB4.5.75]
[Log, Info, N_Security, KeyBitMask, 0x0012]
[Log, Info, ConnectedLamp, Platform version 0.41.0.1, package_ZigBee 117, package_BC_Stack 104, svn 26632]
[Log, Info, ConnectedLamp, Product version WhiteLamp-Atmel 5.38.1.15095, built by Louvre2LL]
[Log, Info, A_Commissioning, Factory New at Ch: 11]
[TH, Ready, 0]
[Sys, test, 1]
[SYS, Error, Incorrect format]
```

Working serial input too!

Tool: Atmel-ICE | Device: ATmega2564RFR2 | Interface: JTAG | Device signature: 0x1EA803

Fuse Name	Value
<input checked="" type="checkbox"/> BODLEVEL	1V8
<input checked="" type="checkbox"/> OCDEN	<input type="checkbox"/>
<input checked="" type="checkbox"/> JTAGEN	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> SPIEN	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> WDTON	<input type="checkbox"/>
<input checked="" type="checkbox"/> EESAVE	<input type="checkbox"/>
<input checked="" type="checkbox"/> BOOTSZ	2048W_1F800
<input checked="" type="checkbox"/> BOOTRST	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> CKDIV8	<input type="checkbox"/>
<input checked="" type="checkbox"/> CKOUT	<input type="checkbox"/>
<input checked="" type="checkbox"/> CKSEL_SUT	TOSC_1KCK_4MS1

JTAG
test points
(see w.p.)

Tool: Atmel-ICE | Device: ATmega2564RFR2 | Interface: JTAG | Device signature: 0x1EA803 | Target Voltage: 3.3 V

Lock Bit	Value
<input checked="" type="checkbox"/> LB	PROG_VER_DISABLED
<input checked="" type="checkbox"/> BLB0	NO_LOCK
<input checked="" type="checkbox"/> BLB1	NO_LOCK

See white-paper for JTAG pin-out connections.

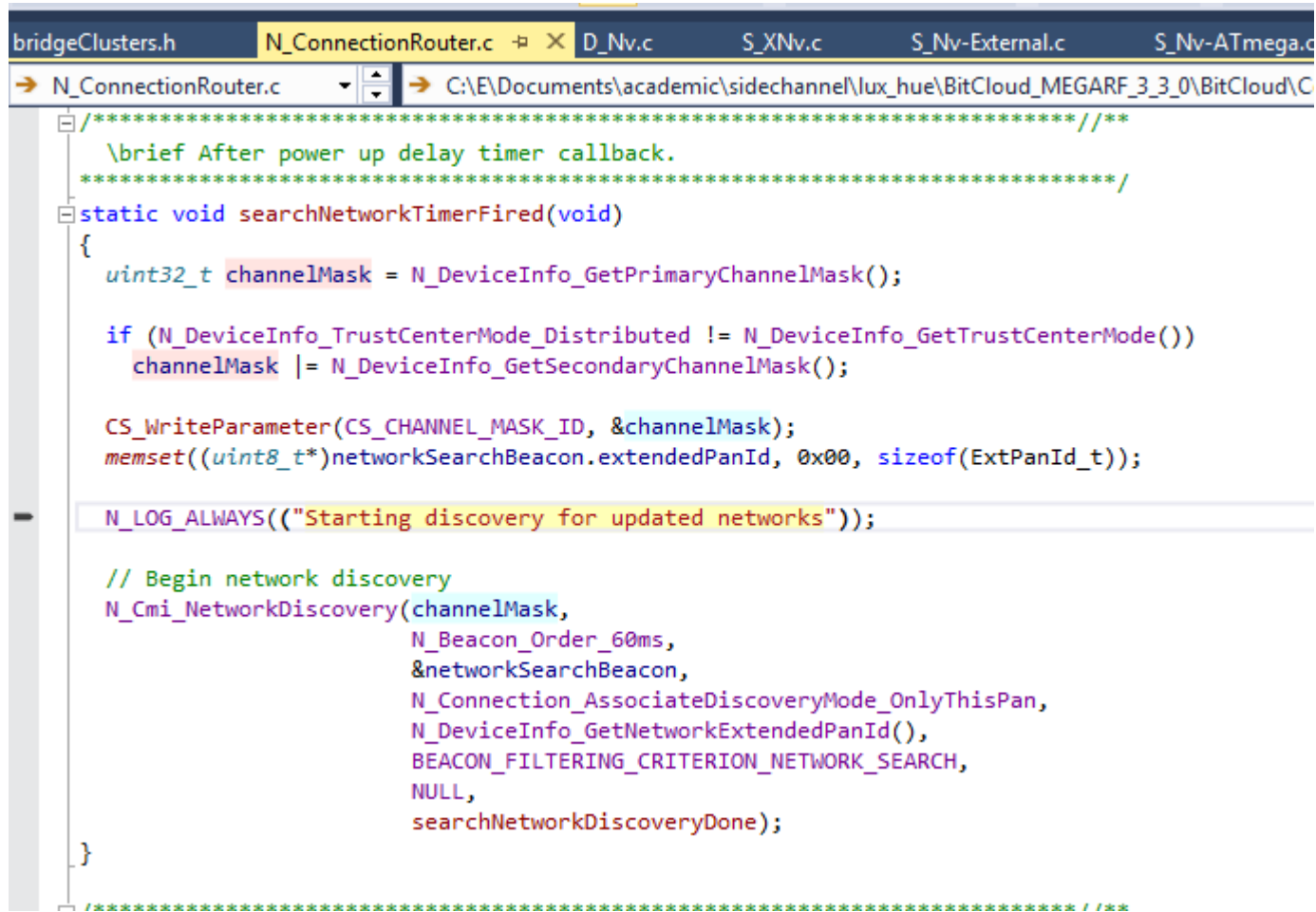
a. Hold SPI line low, notice ASSERT printed matches same name-types used (NVs)

b. Can find same print statements

[TH,Ready,0]

[Log,Info,N_Connection,Starting discovery for updated networks]

[Log,Info,N_Connection,Discovery for updated networks completed]



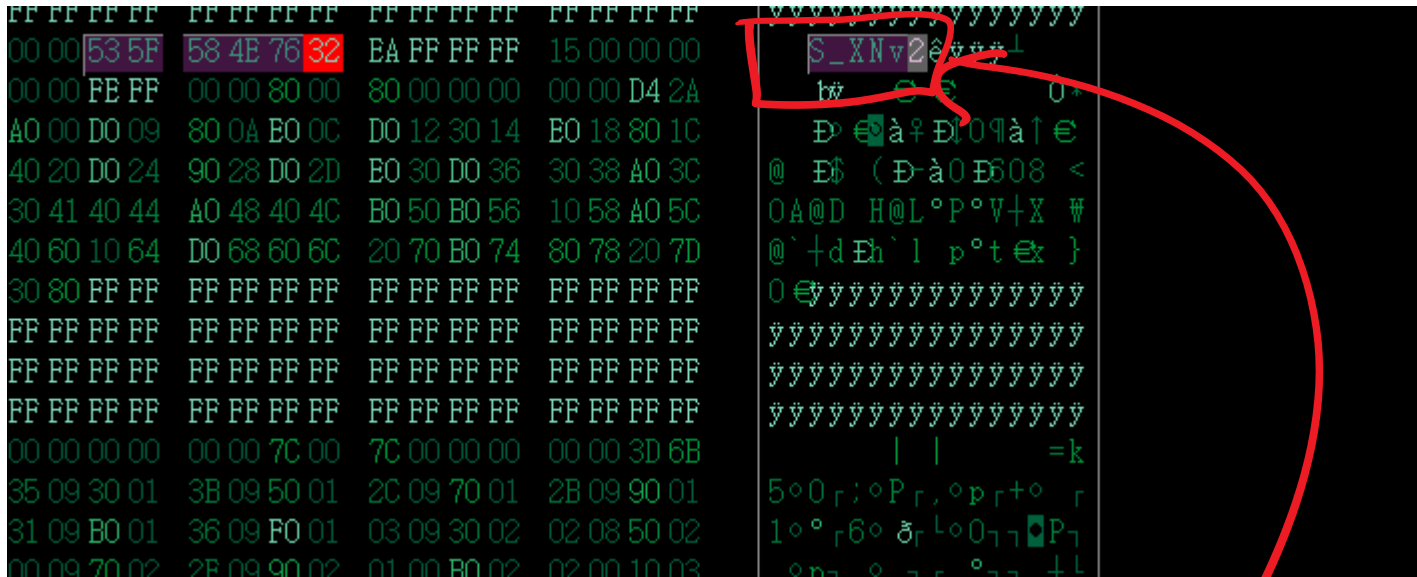
```
bridgeClusters.h  N_ConnectionRouter.c  D_Nv.c  S_XNv.c  S_Nv-External.c  S_Nv-ATmega.c
N_ConnectionRouter.c  C:\E\Documents\academic\sidechannel\lux_hue\BitCloud_MEGARF_3_3_0\BitCloud\C
/*****
 \brief After power up delay timer callback.
 *****/
static void searchNetworkTimerFired(void)
{
    uint32_t channelMask = N_DeviceInfo_GetPrimaryChannelMask();

    if (N_DeviceInfo_TrustCenterMode_Distributed != N_DeviceInfo_GetTrustCenterMode())
        channelMask |= N_DeviceInfo_GetSecondaryChannelMask();

    CS_WriteParameter(CS_CHANNEL_MASK_ID, &channelMask);
    memset((uint8_t*)networkSearchBeacon.extendedPanId, 0x00, sizeof(ExtPanId_t));

    N_LOG_ALWAYS(("Starting discovery for updated networks"));

    // Begin network discovery
    N_Cmi_NetworkDiscovery(channelMask,
        N_Beacon_Order_60ms,
        &networkSearchBeacon,
        N_Connection_AssociateDiscoveryMode_OnlyThisPan,
        N_DeviceInfo_GetNetworkExtendedPanId(),
        BEACON_FILTERING_CRITERION_NETWORK_SEARCH,
        NULL,
        searchNetworkDiscoveryDone);
}
/*****/
```



```
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00 00 53 5F 58 4E 76 32 EA FF FF FF 15 00 00 00 S_XNv2
00 00 FE FF 00 00 80 00 80 00 00 00 00 00 D4 2A t# 0 0 0+
A0 00 D0 09 80 0A E0 0C D0 12 30 14 E0 18 80 1C D à 09 à 1 €
40 20 D0 24 90 28 D0 2D E0 30 D0 36 30 38 A0 3C @ È ( D-à0È608 <
30 41 40 44 A0 48 40 4C B0 50 B0 56 10 58 A0 5C 0A@D H@L°P°V+X #
40 60 10 64 D0 68 60 6C 20 70 B0 74 80 78 20 7D @`+dÈh`l p°t ex }
30 80 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00 00 00 00 00 00 7C 00 7C 00 00 00 00 00 3D 6B | | =k
35 09 30 01 3B 09 50 01 2C 09 70 01 2B 09 90 01 5°0 r;°P r;°p r+° r
31 09 B0 01 36 09 F0 01 03 09 30 02 02 08 50 02 1°° r6° ð r L°0 077 P7
00 09 70 02 2B 09 90 02 01 00 B0 02 02 00 10 03 0 r 0 0 0 0 0 0 +k
```

```
/** 16 byte sector header used in flash located at the start of the active sector. */
typedef struct SectorHeader_t
{
    /** Is this sector active. Written with 0x0000 at the end of the compact operation. */
    uint16_t isActive;
    /** Signature to detect valid sectors. Must have the value "S_XNv2". */
    uint8_t signature[6];
    /** Counter, decreased each time a new sector becomes the active sector. */
    uint32_t sequenceNumber;
    /** Parity bits for the sequenceNumber field = sequenceNumber ^ 0xFFFFFFFFFuL. */
    uint32_t sequenceParity;
} SectorHeader_t;
```

Damn.

December 03, 2014

Lamp software version: 66013452

- Related products are hue A19 and BR30 downlight bulbs and Friends of hue
- hue Tap range is extended if lamp in between Tap and bridge is powered
- Faster start-up when using the wall switch
- Bug fixes and stability improvements

NOT

Atmel based.

BRIDGE

1.0

BRIDGE 1.0 HACKING

Bridge

Dumb hub

Run APP



firmwareupdate_ethernet_bridge_around1206time.pcapng [Wireshark 1.8.0 (SVN Rev 43431 from /trunk-1.8)]

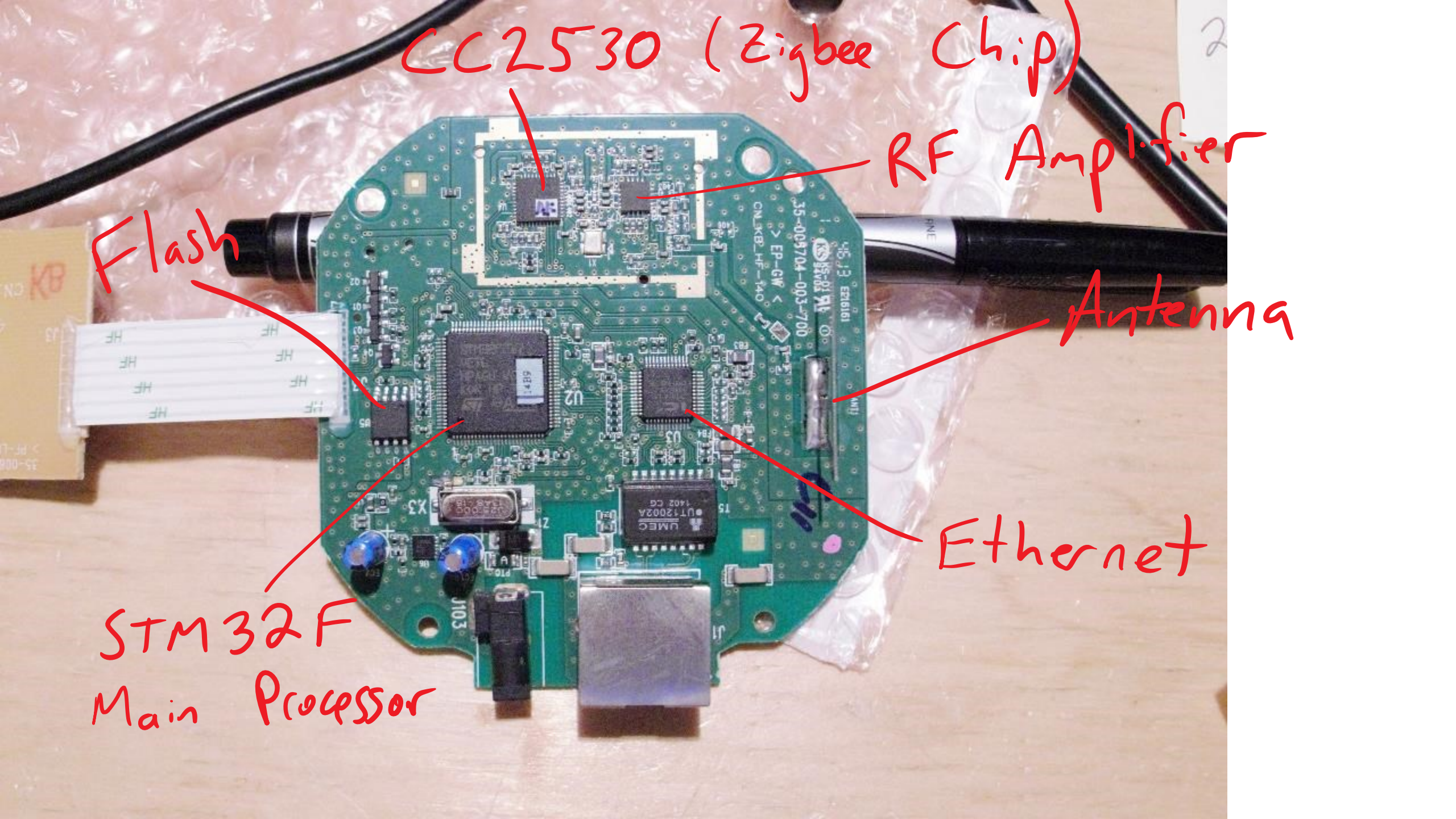
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
8500	1171.694544000	192.168.0.23	5.79.62.93	TCP	60	49640 > http [FIN, ACK] Seq=1623 Ack=873 win=1808 Len=0
8501	1171.694545000	192.168.0.23		DNS	79	standard query 0xaf13 A fds.cpp.philips.com
8502	1171.759431000		192.168.0.23	DNS	172	standard query response 0xaf13 CNAME e4f.edgesuite.net CNAME a1049.g2.akamai.net A 173.237.125.64 A 173.237.125.64
8503	1171.759433000	192.168.0.23	173.237.125.64	TCP	60	49641 > http [SYN] Seq=0 win=2144 Len=0 MSS=536
8504	1171.769461000	173.237.125.64	192.168.0.23	TCP	64	http > 49641 [SYN, ACK] Seq=0 Ack=1 win=14600 Len=0 MSS=1460 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
8505	1171.769464000	192.168.0.23	173.237.125.64	TCP	60	49641 > http [ACK] Seq=1 Ack=1 win=2144 Len=0
8506	1171.769465000	192.168.0.23		HTTP	260	GET /firmware/BSB001/1030262/firmware_rel_cc2530_encrypted_stm32_encrypted_01030262_0012.fw HTTP/1.1
8507	1171.779553000	173.237.125.64	192.168.0.23	TCP	64	http > 49641 [ACK] Seq=1 Ack=207 win=15544 Len=0 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
8508	1171.808458000	5.79.62.93	192.168.0.23	TCP	64	http > 49640 [ACK] Seq=873 Ack=1624 win=3230 Len=0 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
8509	1171.972258000	173.237.125.64	192.168.0.23	TCP	590	[TCP segment of a reassembled PDU]

http://xxx/firmware/HUE0100/66013452/ConnectedLamp-Target_0012_13452_8D.sbl-ota

http://xxx/firmware/BSB001/1030262/firmware_rel_cc2530_encrypted_stm32_encrypted_01030262_0012.fw



CC2530 (Zigbee Chip)

RF Amplifier

Antenna

Ethernet

Flash

STM32F
Main Processor



20706 105C 60V VW-1 -HF-
20706 105C 60V VW-1 -HF-
20706 105C 60V VW-1 -HF-
20706 105C 60V VW-1 -HF-

SW1

35-0087G4
> EP-GW <

TP1 TP6 TP33 TP46 TP43 TP44 TP8
TP41 TP24 TP25 TP45 TP38
TP22 TP23 TP26 TP40 TP34
TP21 TP19 TP16 TP27 TP37 TP36 TP2
TP7 TP32 TP35 TP5 TP30 TP34 TP3
TP17 TP18 TP20 TP39 TP49
TP4 TP9 TP28 TP31 TP10 TP29
TP15

Output from CC2530

```
[Log,Info,S_DeviceInfo,Booting into normal mode...]  
[Log,Info,S_DeviceInfo,DeviceId: IpBridge]  
[Log,Info,N_Security,LIB4.4.52]  
[Log,Info,N_Security,KeyBitMask,0x0012]  
[Log,Info,A_Bridge,Platform version 0.25.0,package_ZigBee 8720,package_Z_Stack  
8720,built by LouvreZLL]  
[Log,Info,A_Bridge,Product version 5.7.1,SmartBridge 11393,built by LouvreZLL]  
[Bridge,Version,5.7.1,SmartBridge 11393,built by LouvreZLL]  
[Bridge,GroupRange,0x5357,0x5367]  
[Log,Info,D_Led,dc 16]  
[Bridge,NetworkSettings,False,0xB163,26DF52A183D85889,11,0,S=0x0001]  
[Log,Info,A_Bridge,NwkAddr: 0x0001, Ch: 11, Pan: 0xB163, NwkUpdId: 0,  
ExtPanID:26:DF:52:A1:83:D8:58:89]  
[Log,Info,D_Led,dc 16]  
[TH,Ready,0]  
[Connection,A]  
[Connection,GetAddress,L=00:17:88:01:01:07:BF:FC,S=0x0001.0]  
[Bridge,StoreGroupRange,0]  
[Log,Info,N_ConnectionRouter,Startup network discovery...]
```

Input to CC2530

```
[Connection,GetAddress]  
[Bridge,StoreGroupRange,0x5357,0x5367]  
[Zcl,S,S=0x0002.11,6,0000000000]  
[Routing,ClearEntry,1]  
[Routing,SendMtoRR,True]  
[Zcl,S,S=0x0003.11,6,0001000000]  
[Routing,ClearEntry,2]  
[Routing,SendMtoRR,True]  
[Zcl,S,S=0x0002.11,6,0002000000]  
[Zcl,S,S=0x0003.11,6,0003000000]  
[Zcl,S,S=0x0002.11,6,0004000000]
```

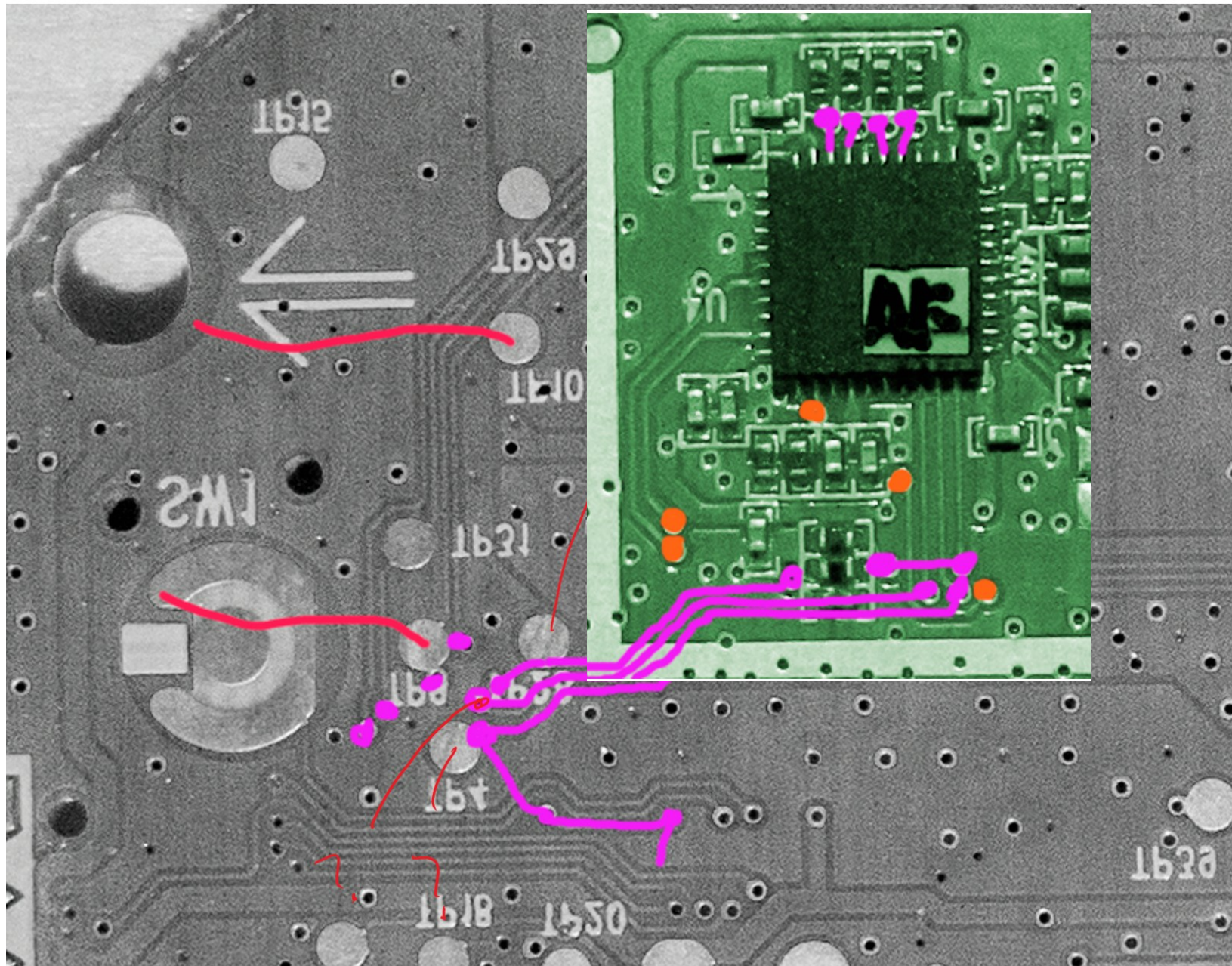
BYPASS

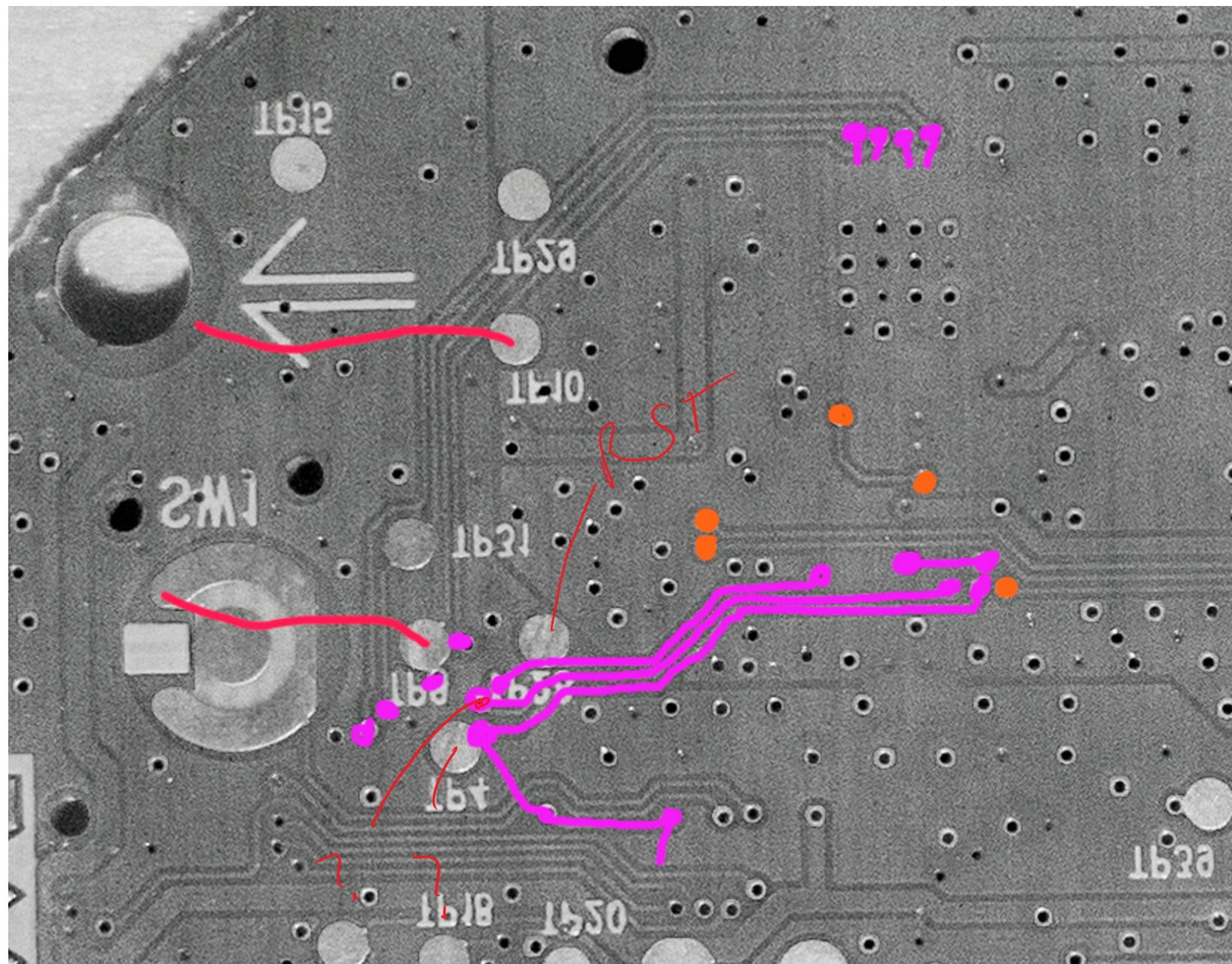
ZLL

KEY?

```
[Zc],s,s=0x0004.11,6,001a000000,64]
[Zc],s,s=0x0004.11,8,001b000000,64]
[Zc],s,s=0x0004.11,768,001c000100004002400300040007000800,64]
[Zc],s,s=0x0004.11,25,190d05000b1000018c340042ae0100002b97098fac0320f2f31e3c8fd035a34097da5018feb50a2e8b40d3678aa57c866a47122020a3a86220a25c93,64]
[Zc],s,s=0x0004.11,25,190e05000b1000018c340042d90100002b2d3d4e5d25c0622af60856c62900d59f71b104541e744b3657ebc32286f3e635474145d3189f7deca60cd9,64]
[Zc],s,s=0x0003.11,6,001d000000,64]
[Zc],s,s=0x0004.11,25,190f05000b1000018c340042040200002b92c84eb5d02416e5153d8aa6a944b0dd7c9796547fa4f63793ea06c100f2c3293c87a425cd5279a8765d3d,64]
[Zc],s,s=0x0003.11,8,001e000000,64]
[Zc],s,s=0x0004.11,25,191005000b1000018c3400422f0200002ba0b3d5e50a0e550e48f25a6125d1aea4fca962453c7f718f05ec20c7875e799ae71b45cd7fc74b3e436094,64]
[Zc],s,s=0x0004.11,25,191105000b1000018c3400425a0200002b2956b4f0014e777aa0ba92c6cb8ed7ddd6d67c114bd4,96d5e03f65105ab62da87dac1c7d344e73ea4c901,64]
[Zc],s,s=0x0002.11,6,001f000000,64]
[Zc],s,s=0x0004.11,25,191205000b1000018c340042850200002be080f0a5152a9d4c0f7eed933a2a3262474106f57b2947d1c44121c326e1c8bfbaea0a925ed58e5a9290a1,64]
a691ef28438fee5be91305000b1000018c340042b00200002b965b229d29d5cf2c0f7eed933a2a3262474106f57b2947d1c44121c326e1c8bfbaea0a925ed58e5a9290a1,64]
[Zc],s,s=0x0004.11,25,191405000b1000018c340042db0200002ba14b4f7686df97989d0371c2c4357733cd9e9361bc90de747f9ec249c2fe86b90f2430595cc5ba87bde7c0,64]
[Routing,SendMtoRR,True]
[Zc],s,s=0x0004.11,25,191505000b1000018c340042060300002be67a19318a005a40204ccbc0126951982709f080f5806e33d478efd8dcda9e79303ad662ddcfa822316b03,64]
```

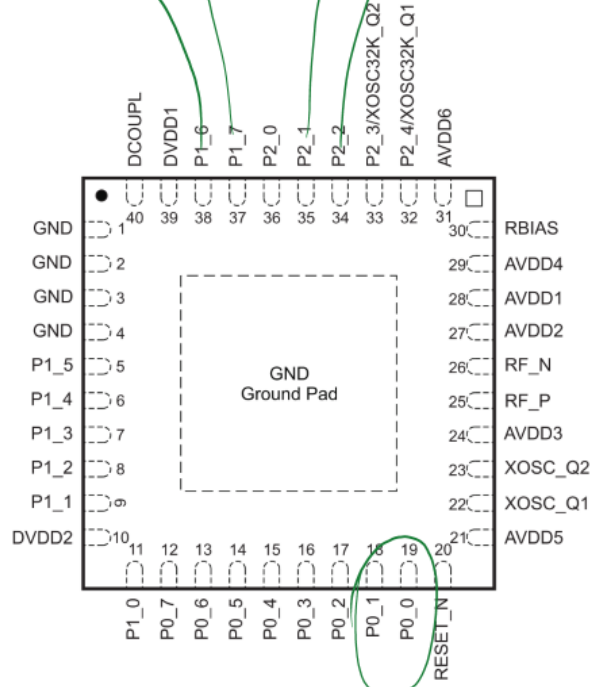
B3 D5 E5 0A	0E 55 0E 48	F2 5A 61 25	D1 AE A4 FC	°Öã	ÞUÞHòZa%Ñ@cü
A9 62 45 3C	7F 71 8F 05	EC 20 C7 87	5E 79 9A 88	@bE<Dq	i Ç#yšç
1B 45 CD 7E	C7 4B 3E 43	00 94 29 5C	B4 F0 01 1B	<EíDÇK>C`")V´ð-N	
77 7A A0 BA	92 C6 CB 8E	D7 DD D6 D6	7C 11 4B D4	wz	°ÆŽ>ÿÖÖ ◀KÔ
29 6D 5E 03	F0 51 05 AB	62 DA 87 DA	C1 C7 D3 44	9m^LòQ	<bú þÁÇÓD
E7 3E A4 C9	01 E0 80 F0	A5 15 2A 9D	4C 0F 7E ED	ç>æÉrà	€±* LQ~í
93 3A 2A 32	62 47 41 06	F5 7B 29 47	D1 C4 41 21	"	*2bGA-ø{}GÑAA!
C3 26 E1 C8	BF BA EA 0A	92 5E D5 8E	5A 92 90 A1	Ã&áÈ	¿æ' ^ÖZZ' i
96 5B 22 9D	29 D5 CF 2E	C8 1C 7E B2	7D 98 84 96	-[")Öí.È ~²}~, -
DC 79 66 A9	7D 4E E7 41	B2 67 E4 CB	C1 C1 B2 BA	Üvf@}	NçA²gäÉÁÁ² °





TX1 TX1
 PP DC } Rebug

CC2530
 RHA Package
 (Top View)



P0076-02

RST

Serial Data During Boot load

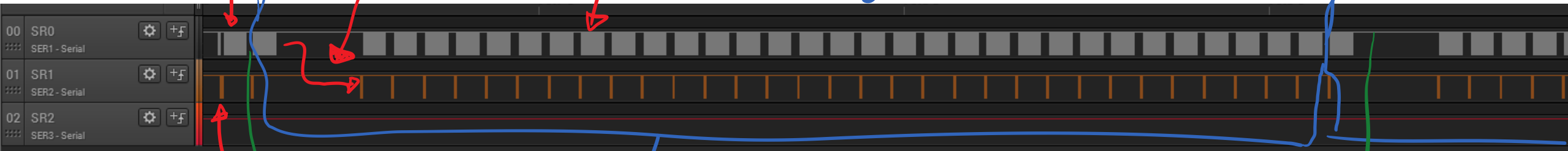
② Data

③

Larger Delay during page erase

Each data block = 64 bytes

Second Page



① Sign on

A

One page = 512 Bytes

B

Extracting Keys from Second Generation Zigbee Chips

Travis Goodspeed
1933 Black Oak Street
Jefferson City, TN, USA
travis@radiantmachines.com

ABSTRACT

First generation Zigbee chips were SPI slaves with no internal processing beyond cryptographic acceleration. Extracting a key was as simple as spying on the SPI transactions. The second generation chips, typified by the CC2430 from Texas Instruments and the EM250 from Ember, contain both a microcontroller and a radio, making the SPI sniffing attack all but irrelevant. Nevertheless, both chips are vulnerable to local key extraction. This paper describes techniques for doing so, focusing on the CC2430 as the EM250 has no protection against outside access. Recommendations are made for defending CC2430 firmware by using compiler directives to place sensitive information in flash memory, rather than in RAM. All Chipcon radios with 8051 cores released prior to the publication of this paper are expected to be vulnerable.

Keywords

Zigbee, CC2430, EM250, System on a Chip (SoC)

1. GENERATIONS

First generation Zigbee chips, such as the CC2420, were simply digital radios with SPI interfaces and a bit of hardware-accelerated cryptography. They could not run a Zigbee stack themselves, but rather relied upon an external microcon-

troller cores were added for convenience, not security, as will be explained below.

The third generation of chips will include more powerful microprocessors and—hopefully—a lot more security. The offering from Texas Instruments is the CC430 family, based upon the MSP430X2 processor. Ember will be using the Arm Cortex M3 in its EM300 series. These chips are out of scope for this paper, as they are not yet commercially available. Also, Freescale's line of radios have not yet been examined by the author, but they will be in the near future.

2. CONCERNING THE EM250

The Ember EM250 contains a 16-bit XAP2b microprocessor from Cambridge Consultants Ltd.[3] Debugging support is provided by that firm's proprietary SIF protocol, with hardware and software available only through Ember. SIF itself is a variant of JTAG.

While the datasheet and various piece of marketing literature claim “The EM250 employs a configurable memory protection scheme usually found on larger microcontrollers.”, this refers not to a debugging fuse or bootloader password, but rather to protection from accidental self-corruption of memory. This is in the form of Application/System separation, allowing the EmberZNet stack to defend certain regions

Good Things

- ZLL master key not in regular SRAM

- Tried AES-128 CBC to decrypt boot loader image, where key = {every possible 16-byte block}

↳ No success, key not in SRAM?

```

..\hue_lux_zll\srandump\bootloadersram_8192_firstframe.bin
0000 0000: 48 B5 7E CE 55 F0 B1 4E 49 57 4F B0 13 9E 7E B4 Jã^?R-?N IWO...x^?
0000 0010: F3 6C D4 74 9E 3E B9 64 F5 4E E0 57 FE B3 BD 89 %1Êt>x>ild SNÓWm|çë
0000 0020: 3E D7 AF 25 B3 87 BF F7 4C 9F BF 7A 0F 9D 2B 7F >I>»|çj Lf1z.0+o
0000 0030: D5 B8 AF FC FF E4 C7 5D 6B 4F 48 9C 7C AC BE F3 ±@»³ õã] kOHÉ!%?%
0000 0040: 97 01 58 FF 00 00 FF FF 00 EE 07 A7 FF 66 00 C7 ù.X .. -° f.ã
0000 0050: F7 00 00 06 00 00 E9 09 FF 01 FA 00 76 03 00 00 .....ú. ..çv..ç
0000 0060: 7E 01 3C 05 07 FA 04 07 FA 04 B0 D2 F1 3B 00 6E ^.<..... ^±;n
0000 0070: 06 FF 00 FA 04 00 FA 04 00 5E 07 F6 0D 01 01 01 .....^ ÷
0000 0080: 01 2A 00 01 01 00 66 CB 15 12 16 15 12 33 03 7E .*...fπ ..3.~
0000 0090: 80 87 74 01 F6 22 FC 87 DB 09 42 96 94 73 46 5C Ççt.÷"³ç ■.Büösf\
0000 00A0: 16 FE 01 00 81 00 80 2A 00 01 01 00 66 CB 15 12 ■...ü.ç* ..fπ..
0000 00B0: 16 15 12 33 03 7E 09 2D 4A 27 D6 3C 49 6D B2 53 ...3.~ç- J'í<ImS
0000 00C0: 80 9E B7 CC 57 E1 95 A3 1A 1A 80 54 E1 01 28 83 Ç>à|AM0óú ..çtø.Çã
0000 00D0: DA 24 B5 7E 4B AD 45 37 9A 52 E5 85 98 10 13 F1 Çá^?KíE7 ÉR0àÿ..±
0000 00E0: FD 86 E8 CD 30 32 C0 00 F6 00 00 00 00 00 00 00 ²ãp=02!! ÷
0000 00F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000 0100: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000 0110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000 0120: 00 01 00 00 42 00 01 01 00 FA 0D 00 01 01 00 66 ...B... ..f
0000 0130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000 0140: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000 0150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000 0160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000 0170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000 0180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000 0190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000 01A0: 00 00 00 00 00 00 00 00 00 00 01 37 90 52 E5 .....7ÉR0
0000 01B0: 85 98 10 13 F1 FD 86 E8 CD 30 32 CA 66 00 01 00 àÿ..±²ãp =02!f...

```

Rx Buffer
 Rx CRC

```

..\hue_lux_zll\srandump\bootloadersram_8192_firstpage.bin
0000 0000: 97 B9 94 8C 26 7B C1 53 31 42 DC 01 29 61 E4 AF ù|òì&<¹S 1B_í)ãö>>
0000 0010: FD B7 8F CF F7 F2 AF 94 6F FF 3E DC 69 F7 B4 76 ²A8x.=>ö o >_i |v
0000 0020: E3 DC B6 D9 BB B8 FA F8 E7 9D C1 EF FE EA FB D7 ò_ã¹_ü0-° p0¹_ü¹_i
0000 0030: AC 7F 53 4B 5E AB 58 8C DF 39 D7 93 1E 7E DE D6 %òSK^?xi #?iô.~i i
0000 0040: CE 6A 58 FF 00 00 FF FF 00 EE 07 A7 FF 66 00 D7 !!jX .. -° f.î
0000 0050: F7 00 00 06 00 00 E9 09 FF 00 F5 84 00 EB 00 52 .....ú. ..Sã.ù.R
0000 0060: 4B 06 DC 04 07 F5 04 07 F5 04 42 0D 7F 4F 00 6E K...S.. S.Bi△o.n
0000 0070: 06 4E 00 F5 04 00 F5 04 00 5E 07 E4 0D 01 01 01 ^..S..S.. ^..õ....~
0000 0080: 01 2A 00 01 01 00 66 CB 15 12 16 15 12 33 03 7E .*...fπ ..3.~
0000 0090: 80 87 74 01 F6 22 FC 87 DB 09 42 96 94 73 46 5C Ç[7kãã.p @.HÜ²$>
0000 00A0: 5F FE 01 00 81 00 80 FF FF FF FF FF FF FF FF ■...ü.ç
0000 00B0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0000 00C0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0000 00D0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0000 00E0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0000 00F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....õ.....
0000 0100: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000 0110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000 0120: 00 01 00 00 42 00 01 22 00 FA 0D 00 01 01 00 66 ...B... ..f
0000 0130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000 0140: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000 0150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000 0160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000 0170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000 0180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000 0190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000 01A0: 00 00 00 00 00 00 00 00 00 00 01 37 90 52 E5 .....7ÉR0
0000 01B0: 85 98 10 13 F1 FD 86 E8 CD 30 32 CA 66 00 00 00 àÿ..±²ãp =02!f...

```

Tx Buffer
 Rx CRC
 Page #

??

TX BUFFER ATTACK

```
for(uint8 i=0; i < data-to-send; i++) {  
    uart_write(tx_buf[i]);  
}
```

Tx Buffer

90:	80	87	74	01	F6	22	FC	87	DB	09	42	96
A0:	16	FE	01	00	81	00	80	2A	00	01	01	00
B0:	16	15	12	33	03	7E	80	2D	4A	27	D6	3C
C0:	80	FE	B7	CC	57	11	95	A3	1A	1A	80	54
D0:	00	04	0E	7E	40	00	4E	27	00	50	FE	0E

The Tx Buffer (rows A0 and B0) contains the data: 16 FE 01 00 81 00 80 2A 00 01 01 00. The overflowed data (row C0) is circled in blue and contains: 80 FE B7 CC 57 11 95 A3 1A 1A 80 54. The overflowed data is labeled 'RX BUF' in green.

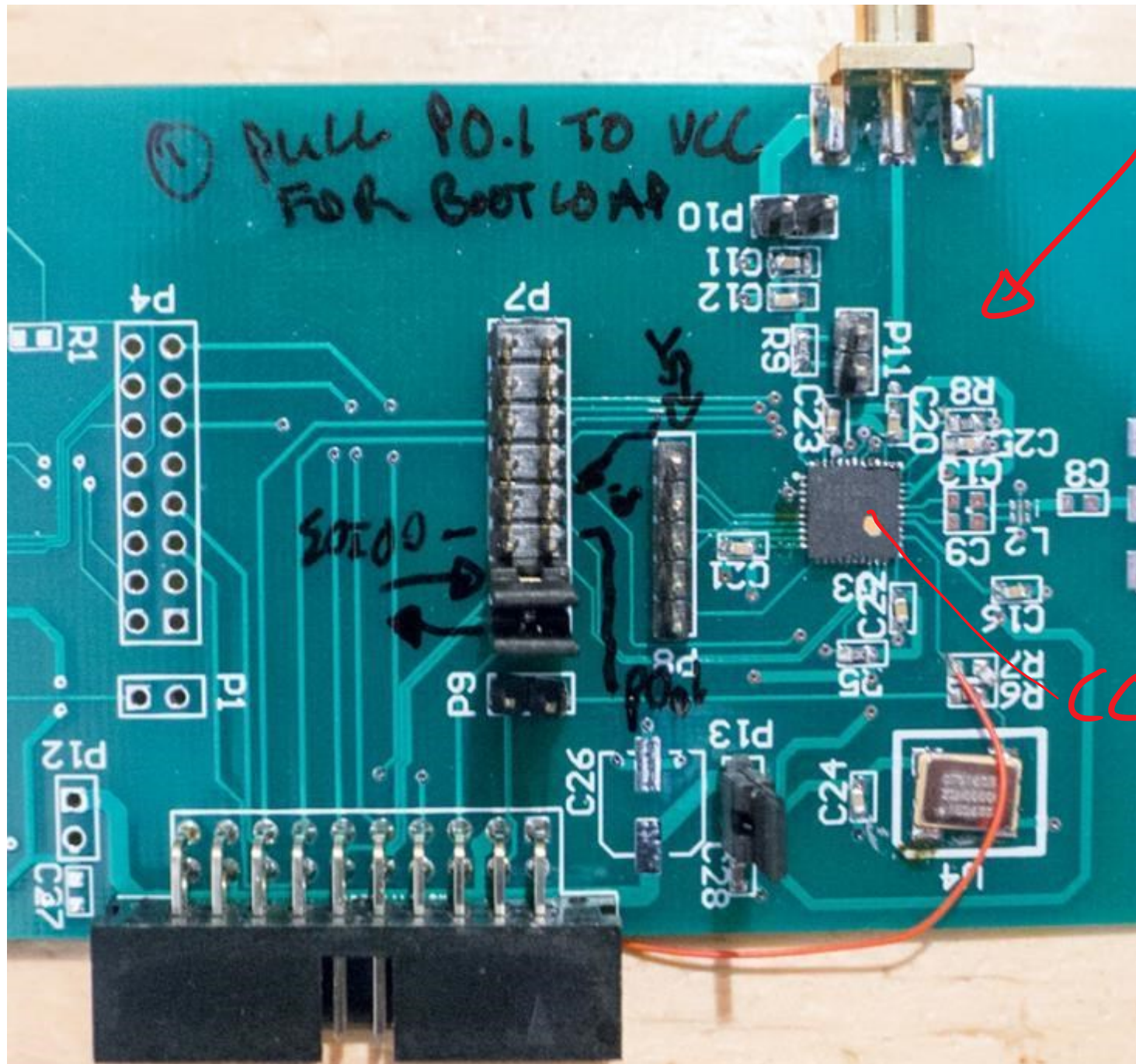
TX BUFFER ATTACK

```
for(uint8 i=0; i < data-to-send, i++) {  
    uart_write(tx_buf[i]);  
}
```

Glitch Attack!

Tx Buffer

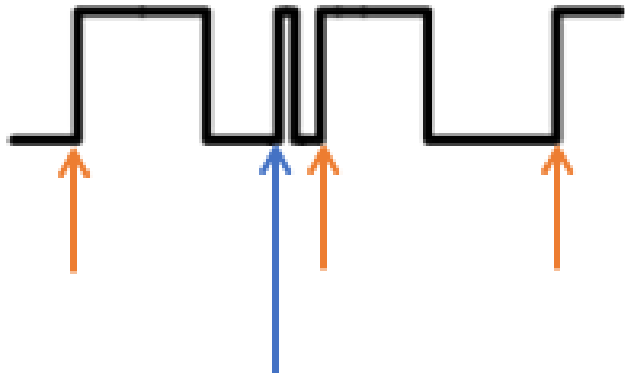
90:	80	87	74	01	F6	22	FC	87	DB	09	42	96
A0:	16	FF	01	00	81	00	80	2A	00	01	01	00
B0:	16	15	12	33	03	7E	80	2D	4A	27	D6	3C
C0:	80	9E	B7	CC	57	11	95	A3	1A	1A	80	54
D0:	00	24	BE	7E	4B	00	4E	27	00	52	FE	9E

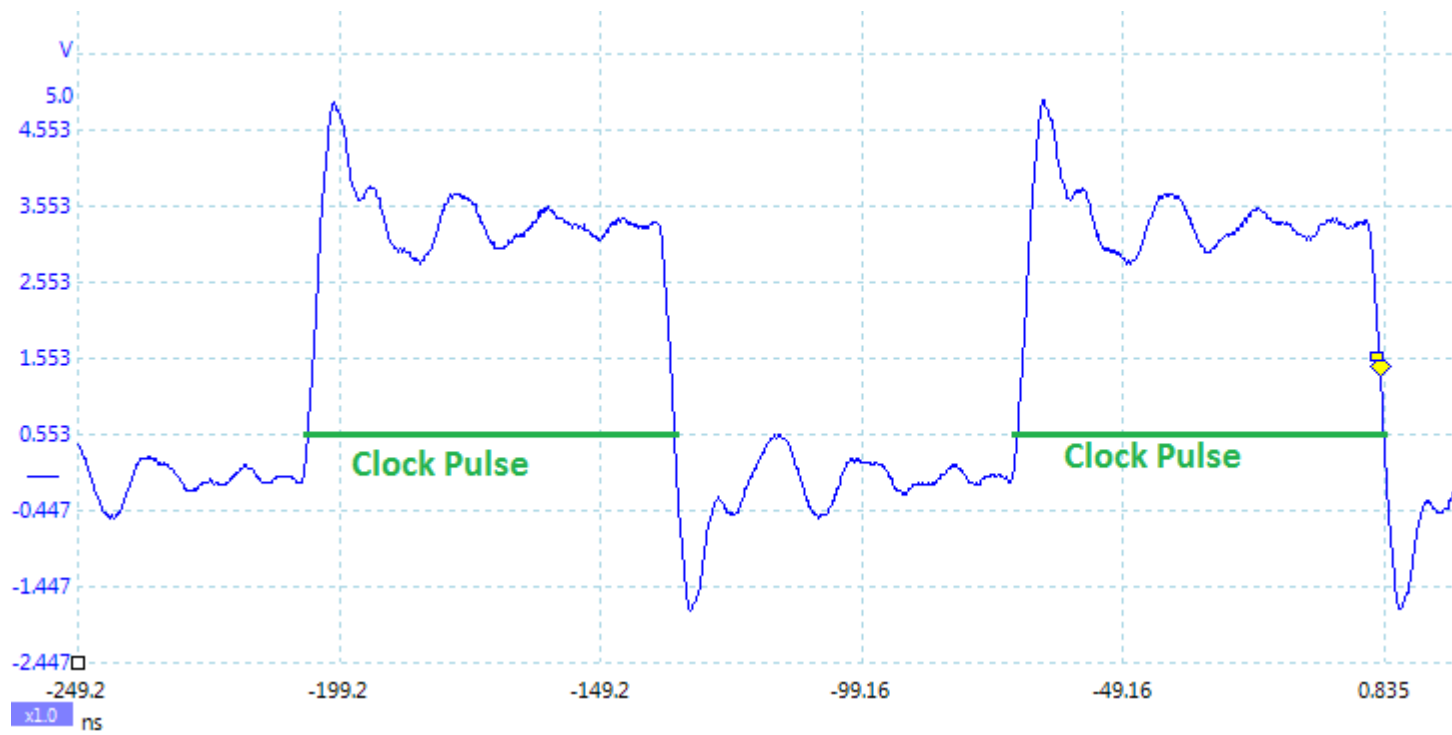


Custom PCB

C2530 from Bridge

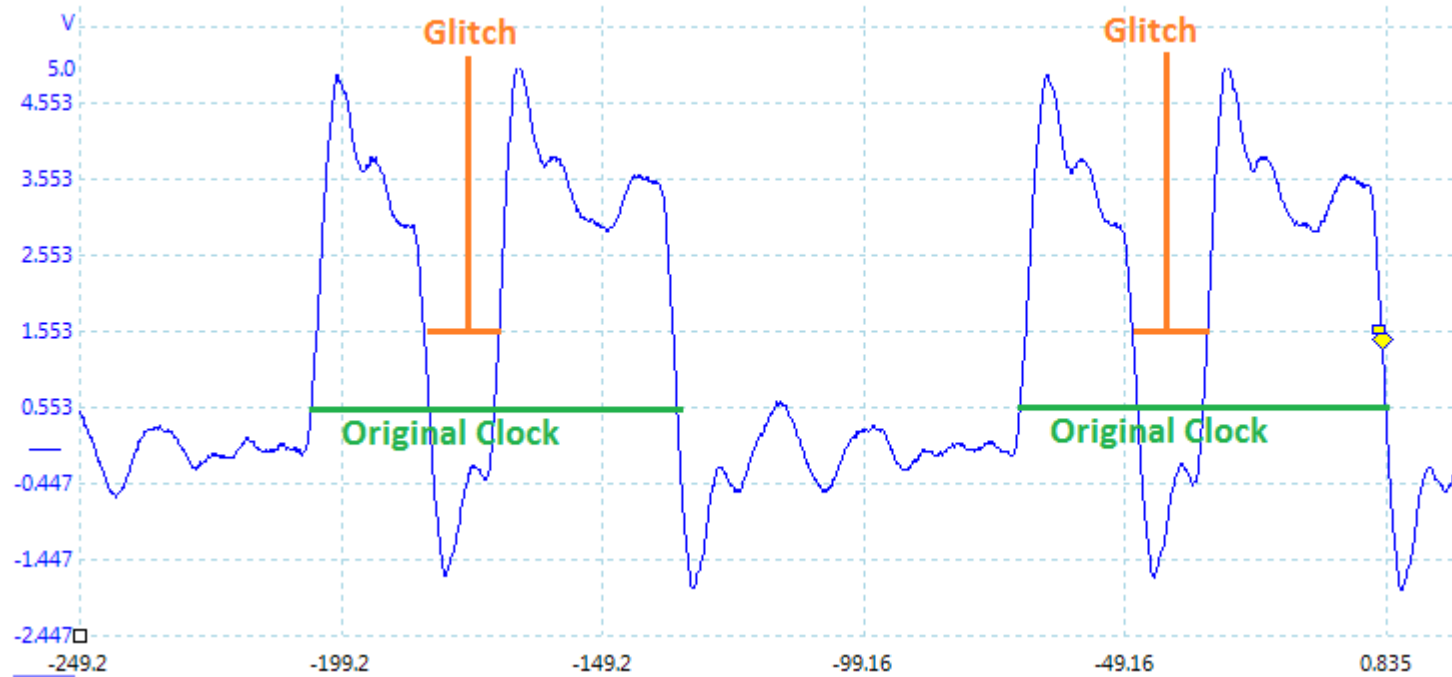
Clock Glitching





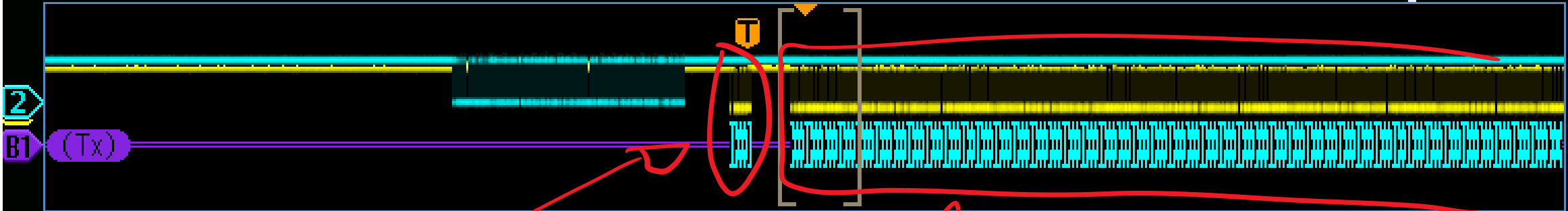
**Original
Clock**

7.37 MHz



**Width = 10%
Offset = +15%**

**Clock XORd
with Glitch**

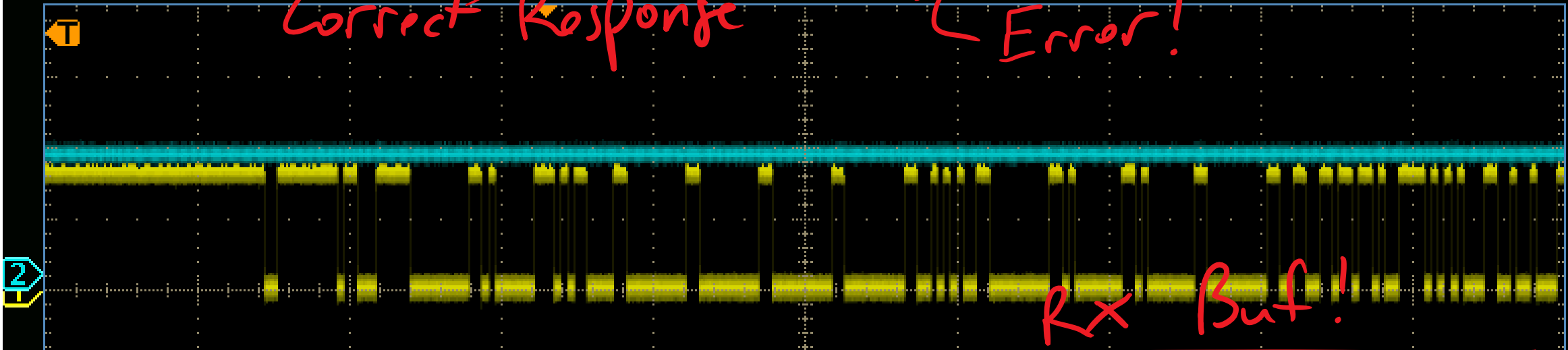


Zoom Factor: 20 X

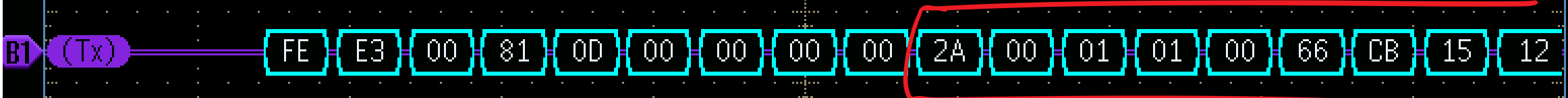
Zoom Position: 1.84ms

Correct Response

Error!



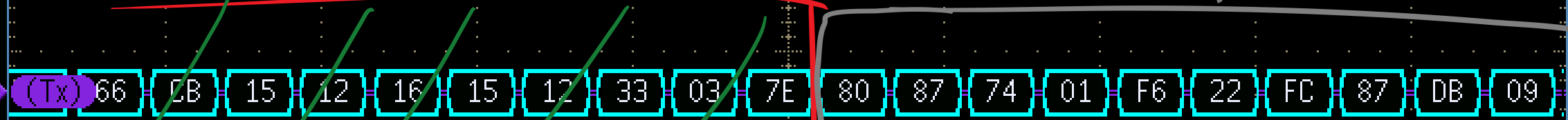
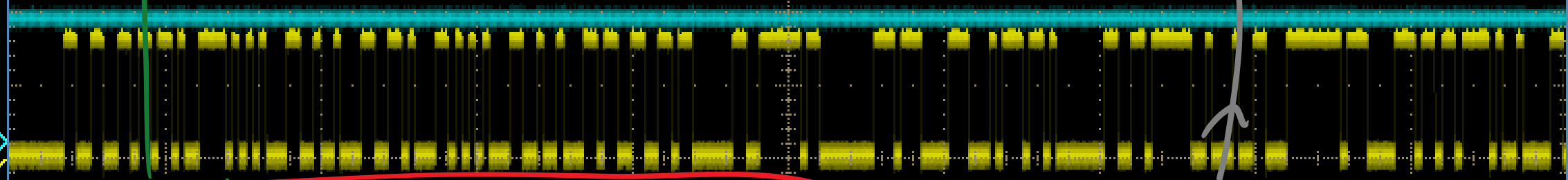
Rx Buf!



1 2.00 V R_{W} 2 2.00 V R_{W} Z 200 μ s 250MS/s B1 Tx Data
 1.50000ms 10M points
 Bus Search events found: 0
 3 Apr 2016 15:38:03

Search On Search Type Bus Source Bus B1 (RS-232) Search For Tx Data Data 4A 27h

0000 0070:	06	FF	00	FA	04	00	FA	04	00	5E	07	FC	00	01	01	01		
0000 0080:	01	2A	00	01	01	00	66	CB	15	12	16	15	12	33	03	7E	*	f
0000 0090:	80	87	74	01	F6	22	FC	87	DB	09	42	96	94	73	46	5C	ç	t.
0000 00A0:	16	FE	01	00	81	00	80	2A	00	01	01	00	66	CB	15	12	.	ü.
0000 00B0:	16	15	12	33	03	7E	80	2D	4A	27	D6	3C	49	6D	B2	53	.	f
0000 00C0:	80	9E	B7	CC	57	E1	95	A3	1A	1A	80	54	E1	01	28	83	ç	À
0000 00D0:	DA	24	B5	7E	4B	AD	45	37	90	52	E5	85	98	10	13	F1	ç	À
0000 00E0:	FD	86	E8	CD	30	32	CA	00	F6	00	00	00	00	00	00	00	ç	À
0000 00F0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
0000 0100:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
0000 0110:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		



7.200us 250MS/s 10M points (Tx) 66 CB 15 12 16 15 12 33 03 7E 80 87 74 01 F6 22 FC 87 DB 09

1 2.00 V Bw 2 2.00 V Bw 1.50000ms

Bus Search events found: 0

3 Apr 2016 15:38:15

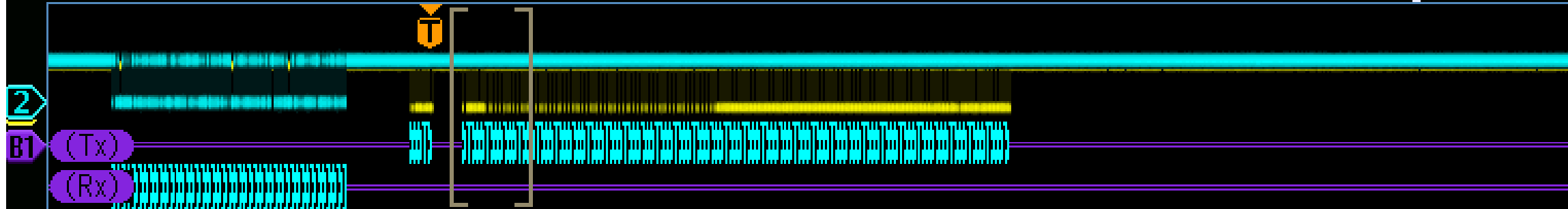
Search On

Search Type Bus

Source Bus B1 (RS-232)

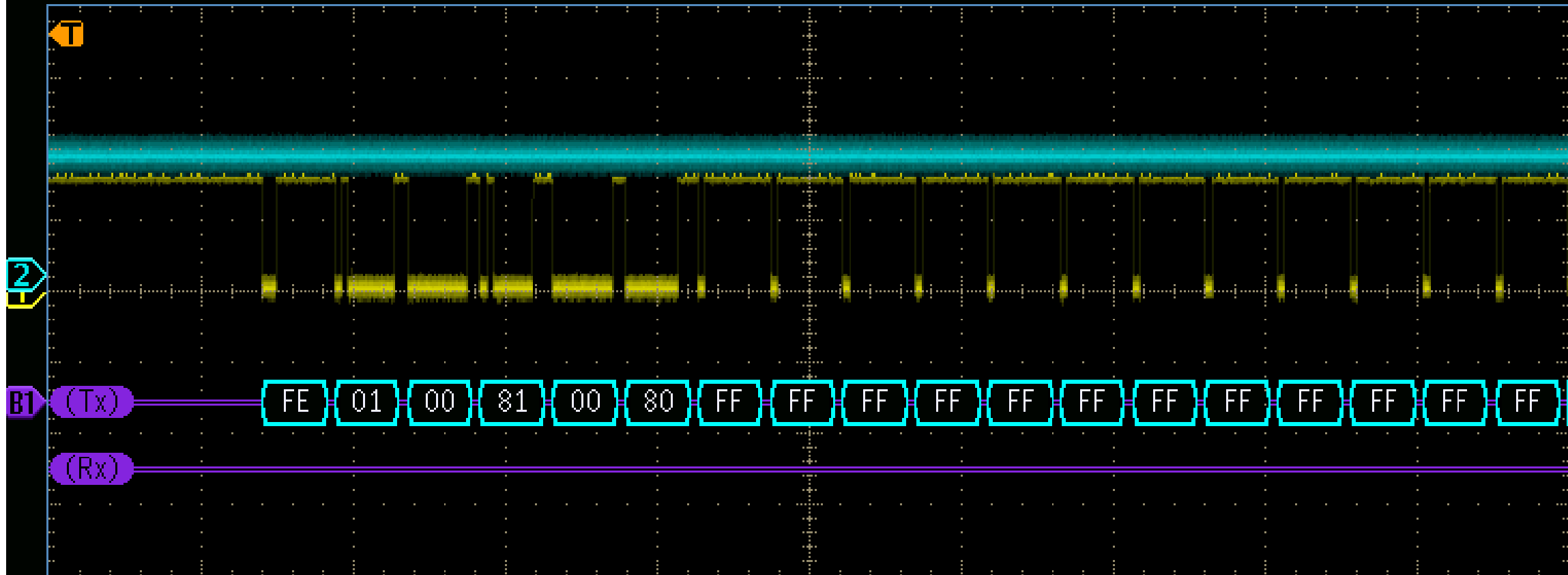
Search For Tx Data

Data 4A 27h



Zoom Factor: 20 X

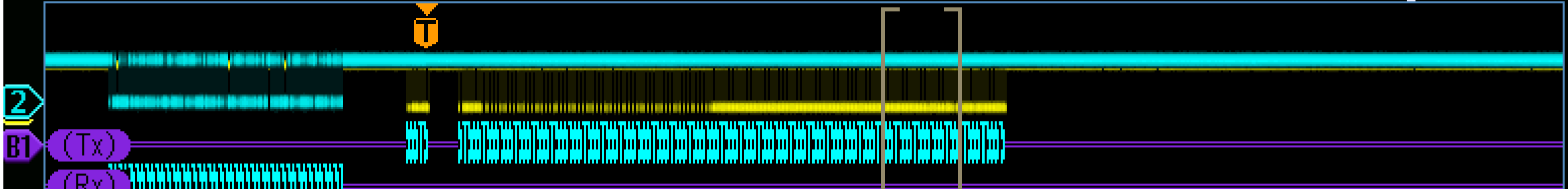
Zoom Position: 1.62ms



1 2.00 V B_w 2 2.00 V B_w

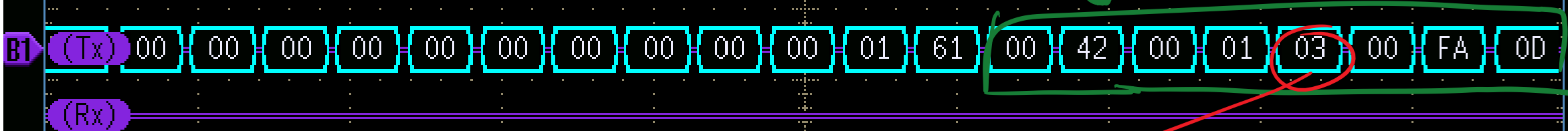
Z 200µs 125MS/s B1 Tx Data
T 25.00 % 5M points

3 Apr 2016
17:38:22



Zoom Factor: 20 X

0000	00D0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF		
0000	00E0:	FF	FF	FF	FF	FF	FF	FF	00	E4	00	00	00	00	00	00	00		õ
0000	00F0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
0000	0100:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
0000	0110:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
0000	0120:	00	01	00	00	42	00	01	22	00	FA	0D	00	01	01	00	66	B	"
0000	0130:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
0000	0140:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
0000	0150:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
0000	0160:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
0000	0170:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
0000	0180:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
0000	0190:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
0000	01A0:	00	00	00	00	00	00	00	00	00	00	00	01	37	90	52	E5		7ÉRõ
0000	01B0:	85	98	10	13	F1	FD	86	E8	CD	30	32	CA	66	00	00	00	àü	..±²&þ =02µf



Targetted page 3

Appears section of SRAM
is erased after use.

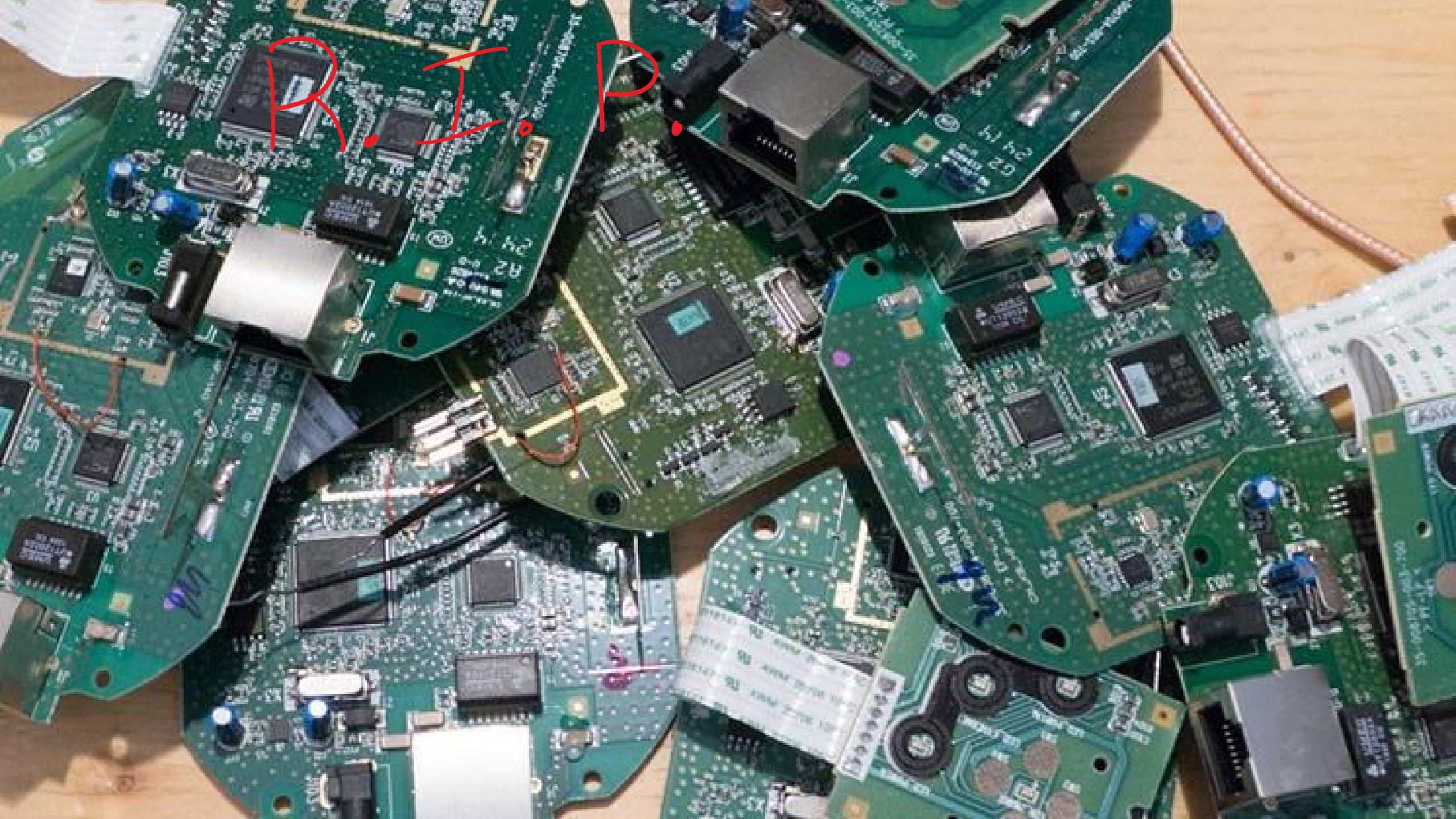
↳ This is good practice!

↳ May be possible with more
glitches.

Glitch Attacks To Firmware

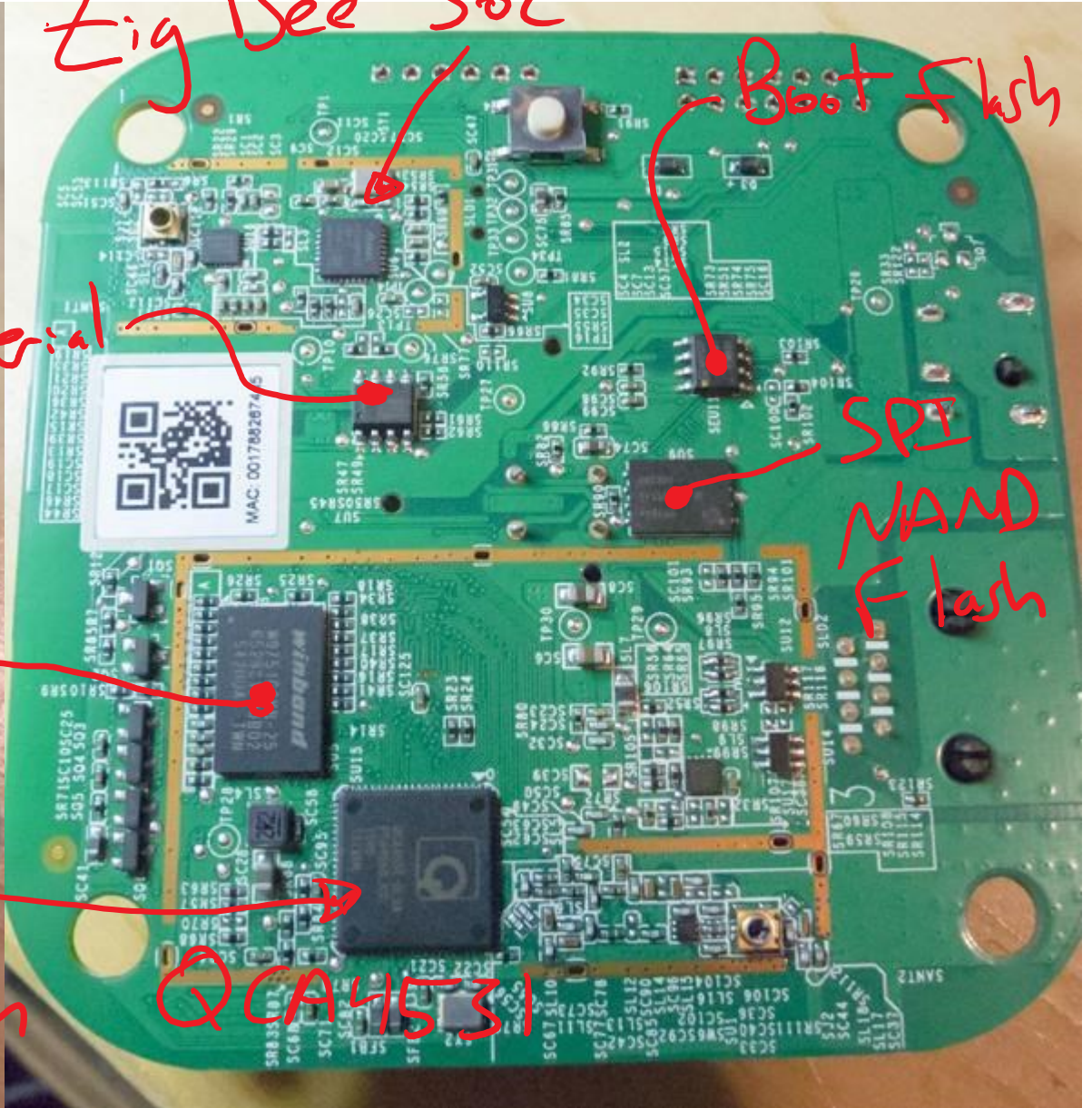
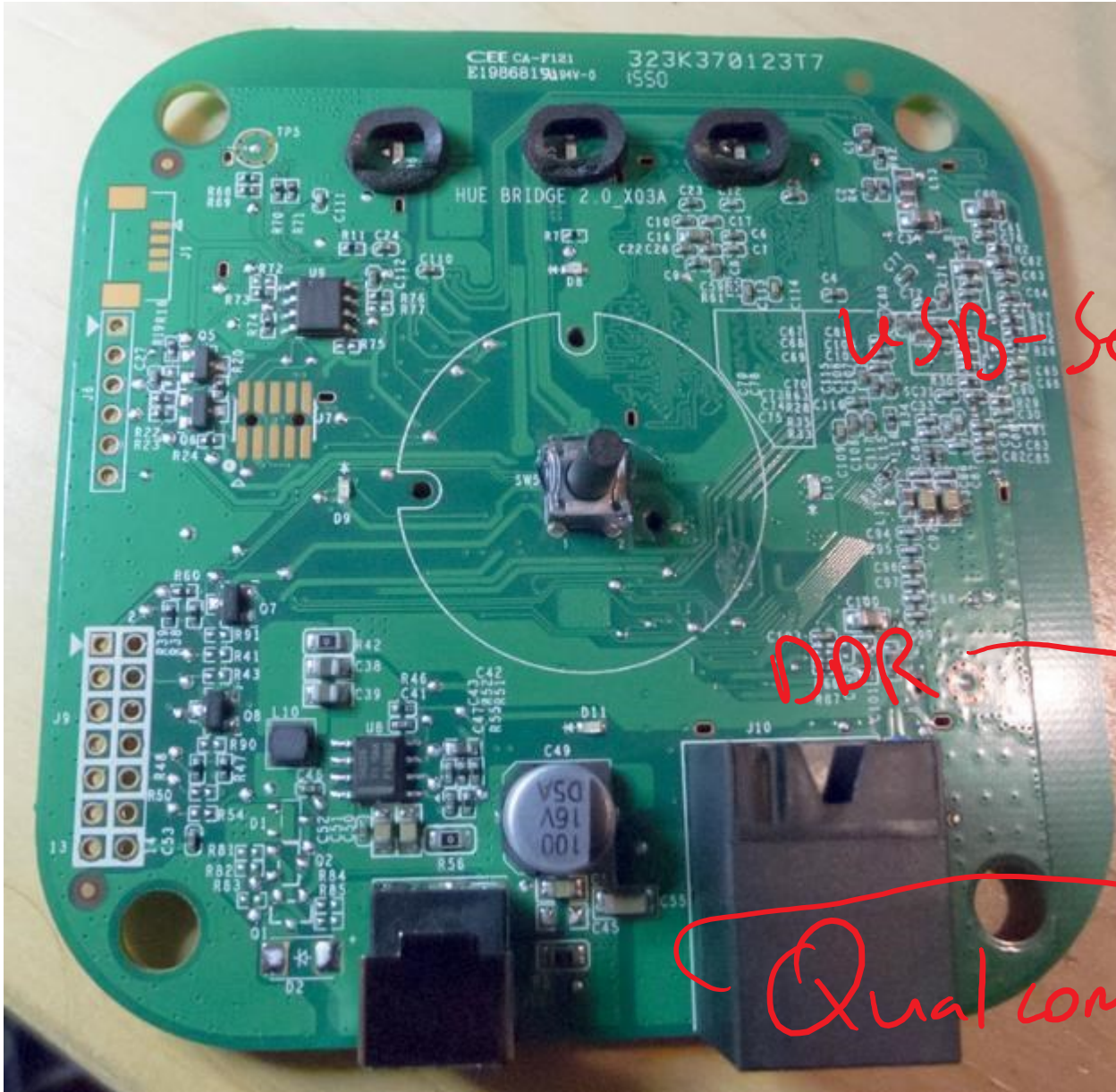
- Appears we can use glitching to dump SRAM.
- Careful timing required to get decrypted data.

R.I.P.



BRIDGE

2.0



Zig Bee Soc

Boot Flash

USB-Serial

SPI NAND Flash

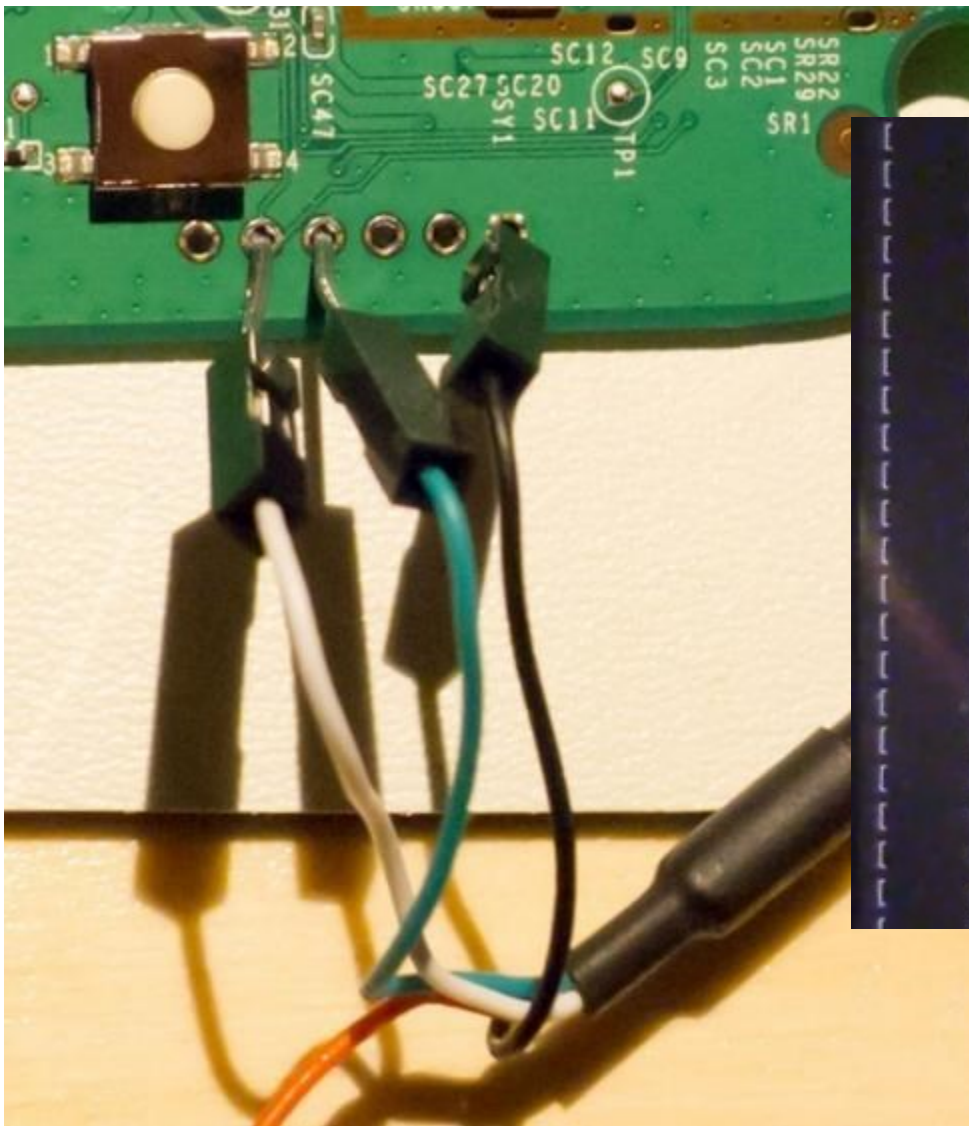
DDR

Qualcom

QCA4531

HACKING
TOOLS

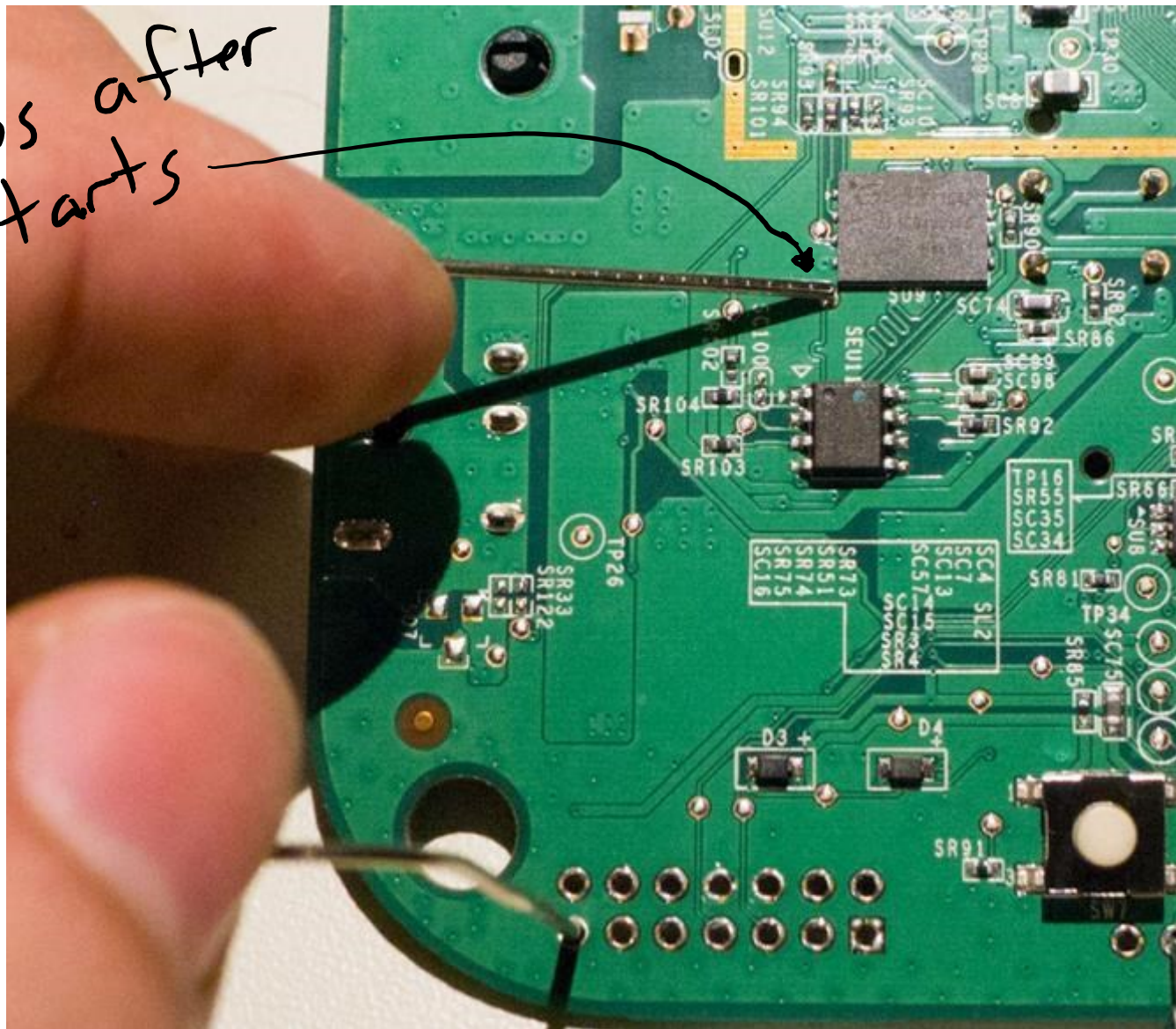




```
[ 0.600000] io scheduler noop registered
[ 0.600000] io scheduler deadline registered (default)
[ 0.610000] Serial: 8250/16550 driver, 1 ports, IRQ sharing disabled
[ 0.630000] serial8250.0: ttyS0 at MMIO 0x18020000 (irq = 11, base_
[ 0.640000] console [ttyS0] enabled
[ 0.640000] console [ttyS0] enabled
[ 0.650000] bootconsole [early0] disabled
[ 0.650000] bootconsole [early0] disabled
[ 0.660000] m25p80 spi0.0: found gd25d40, expected m25p80
[ 0.670000] m25p80 spi0.0: gd25d40 (512 Kbytes)
[ 0.670000] 4 cmdlinepart partitions found on MTD device spi0.0
[ 0.680000] Creating 4 MTD partitions on "spi0.0":
[ 0.680000] 0x000000000000-0x0000000040000 : "u-boot"
[ 0.690000] 0x0000000040000-0x0000000060000 : "u-boot-env"
[ 0.690000] 0x0000000060000-0x0000000070000 : "reserved"
[ 0.700000] 0x0000000070000-0x0000000080000 : "art"
[ 0.710000] nand: device found, Manufacturer ID: 0xc8, Chip ID: 0xb
[ 0.720000] nand: Giga Device GD5F1GQ4U 1G 3.3V 8-bit
[ 0.730000] nand: 128MiB, SLC, page size: 2048, OOB size: 128
[ 0.730000] Scanning device for bad blocks
[ 0.900000] Bad eraseblock 768 at 0x000006000000
[ 0.900000] Bad eraseblock 776 at 0x000006100000
```

<https://www.youtube.com/watch?v=hi2D2MnwiGM>
Or: <http://www.oflynn.com>

Short
boot ~3s after
starts



<https://www.youtube.com/watch?v=hi2D2MnwiGM>
Or: <http://www.oflynn.com>

```
eth1: 00:17:88:24:15:8e
```

```
athrs27_phy_setup ATHR_PHY_CONTROL 0 :1000
```

```
athrs27_phy_setup ATHR_PHY_SPEC_STAUS 0 :10
```

```
athrs27_phy_setup ATHR_PHY_CONTROL 1 :1000
```

```
athrs27_phy_setup ATHR_PHY_SPEC_STAUS 1 :10
```

```
athrs27_phy_setup ATHR_PHY_CONTROL 2 :1000
```

```
athrs27_phy_setup ATHR_PHY_SPEC_STAUS 2 :10
```

```
athrs27_phy_setup ATHR_PHY_CONTROL 3 :1000
```

```
athrs27_phy_setup ATHR_PHY_SPEC_STAUS 3 :10
```

```
eth1 up
```

```
eth0, eth1
```

```
Qualcomm Atheros SPI NAND Driver, Version 0.1 (c) 201
```

```
ath_spi_nand_ecc: Couldn't enable internal ECC
```

```
Setting 0x181162c0 to 0x4b97a100
```

```
Hit any key to stop autoboot: 0
```

```
** Device 0 not available
```

```
ath> █
```


Use "mkpasswd"

```
ath> setenv bootdelay 3  
ath> printenv security
```

*****COPY THE DEFAULT VALUE THAT WAS PRINTED & SAVE THIS SOMEWHERE*****

```
ath> setenv security '$5$wbgtEC1iF$ugIfQUoE7SNg4mplDI/7xdfLC7jXoMAkupeMsm10hY9'  
ath> printenv security  
security=$5$wbgtEC1iF$ugIfQUoE7SNg4mplDI/7xdfLC7jXoMAkupeMsm10hY9  
ath> saveenv  
ath> reset
```

<https://www.youtube.com/watch?v=hi2D2MnwiGM>

<http://colinoflynn.com/?p=706>

- Master binary seems to “do it all” (webserver, parsing requests, etc.)
at `/usr/sbin/ipbridge`
- FW Update routine at `/usr/sbin/swupdate`
 - References AES-CBC-256 decryption routine, which references encryption key
at `/home/swupdate/certs/enc.k`
 - Two different bridges used same AES key (not really a big deal, as we already have unencrypted binaries since we have root).

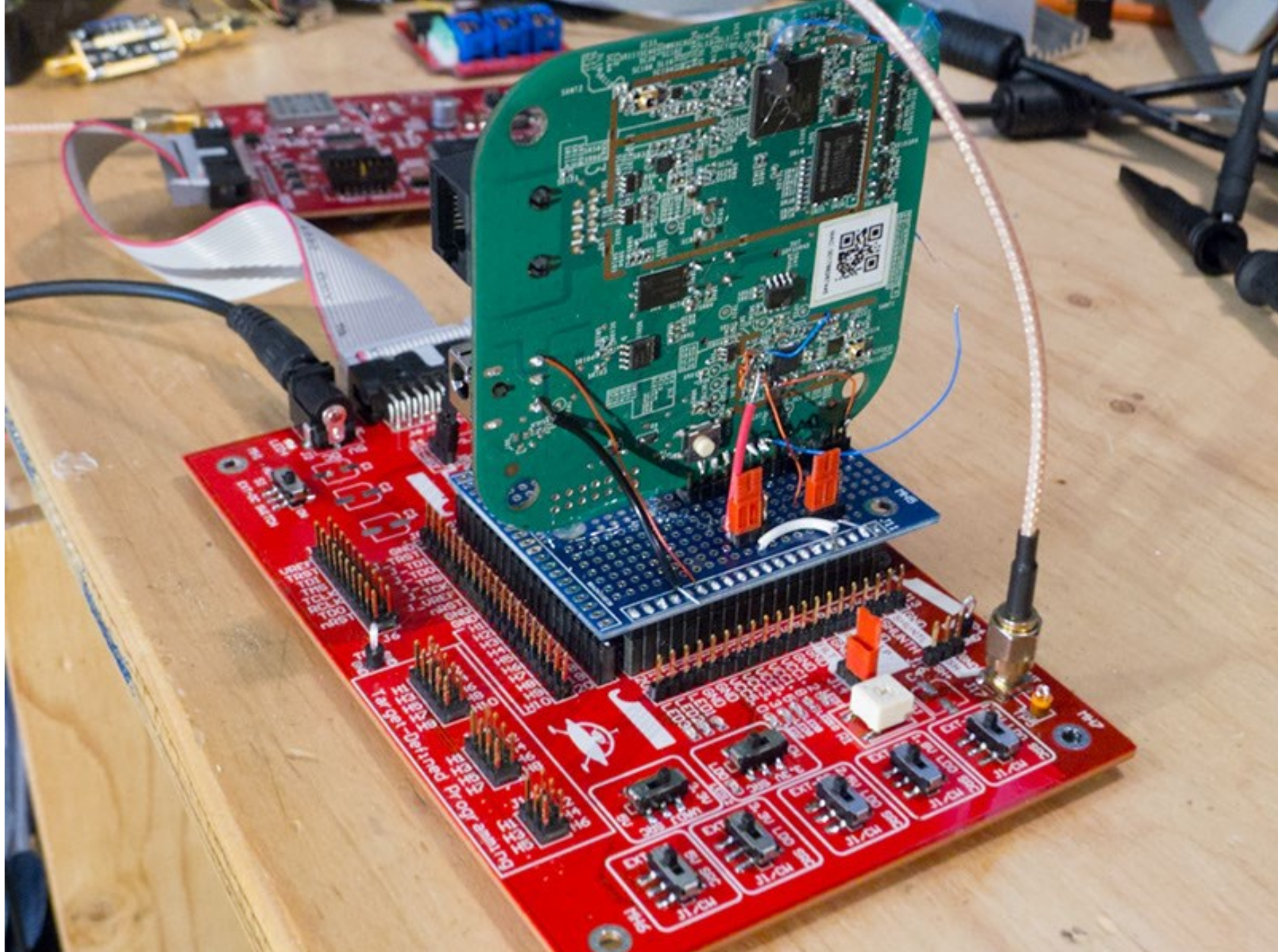


TX/RX

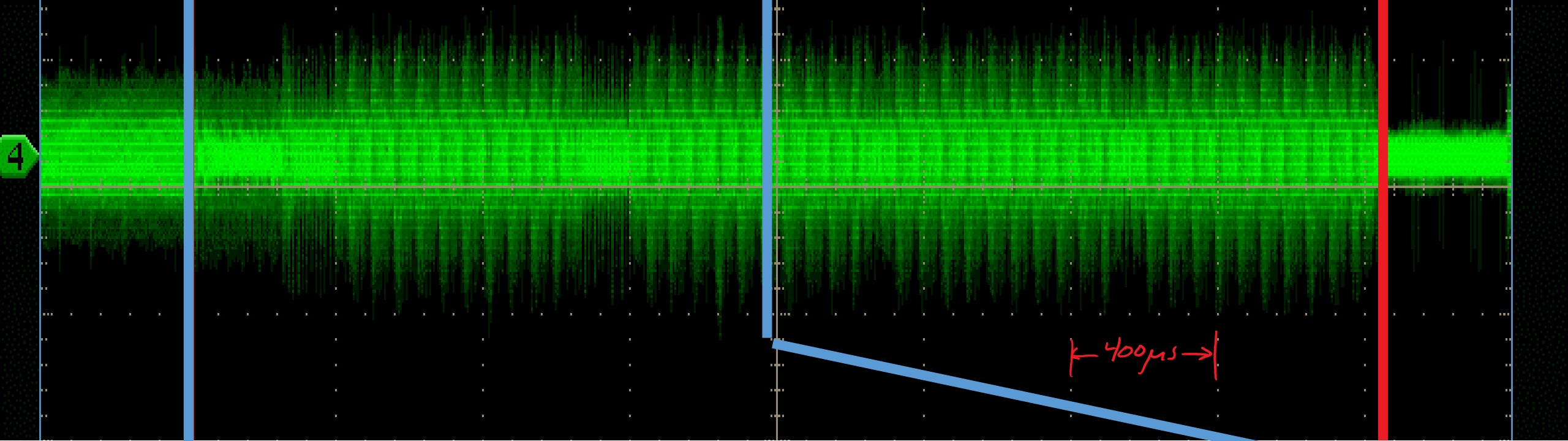
nRST

Power Analysis

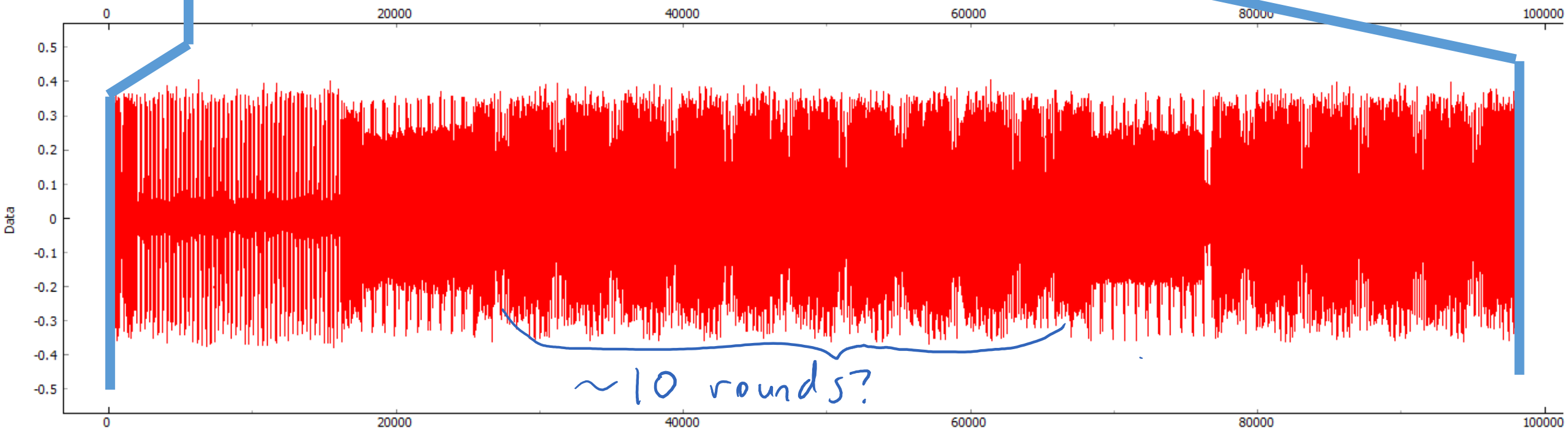
CLK-IN



4

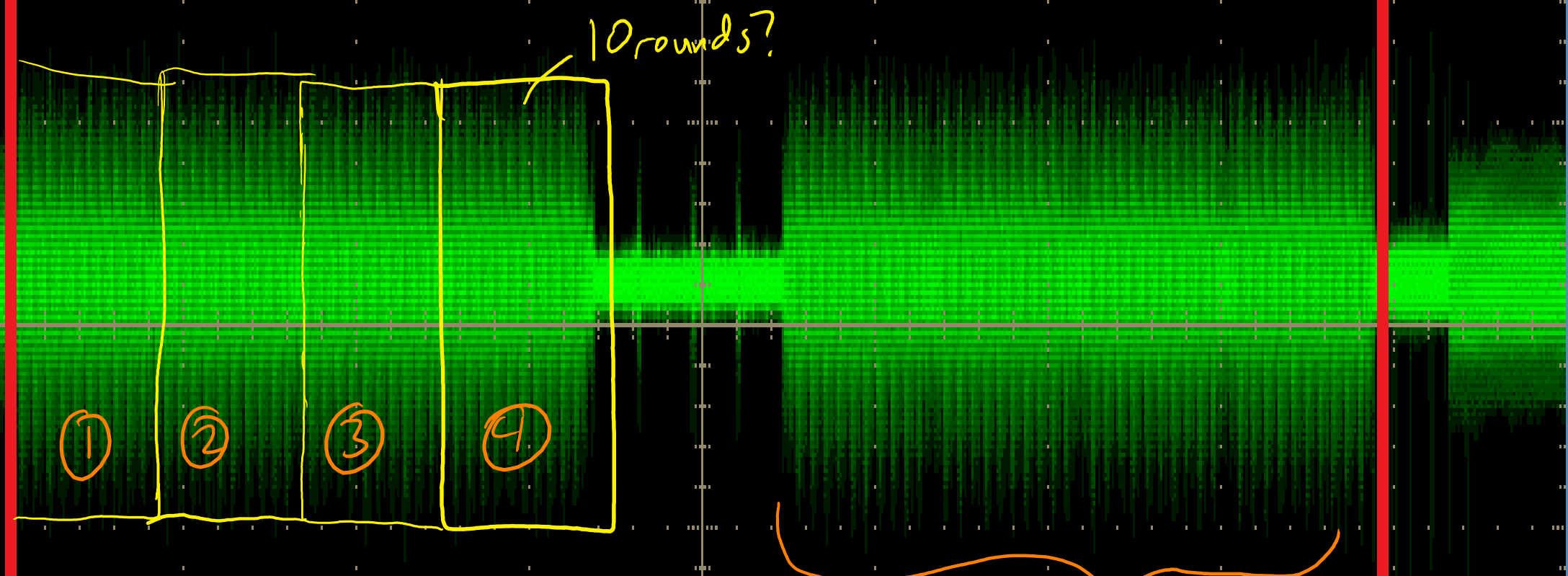


← 400μs →



~ 10 rounds?

4



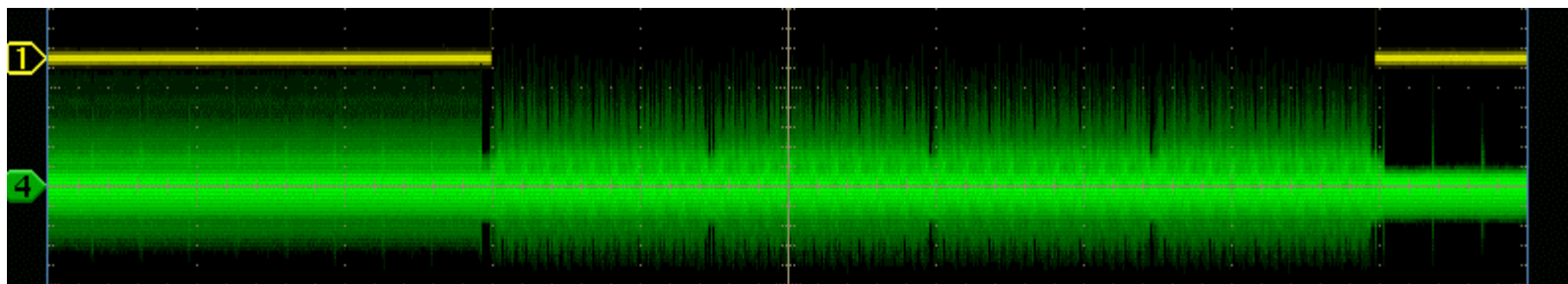
① - ④ = AES-128?
would explain 64 bytes.

Again!?
↳ Maybe signature?

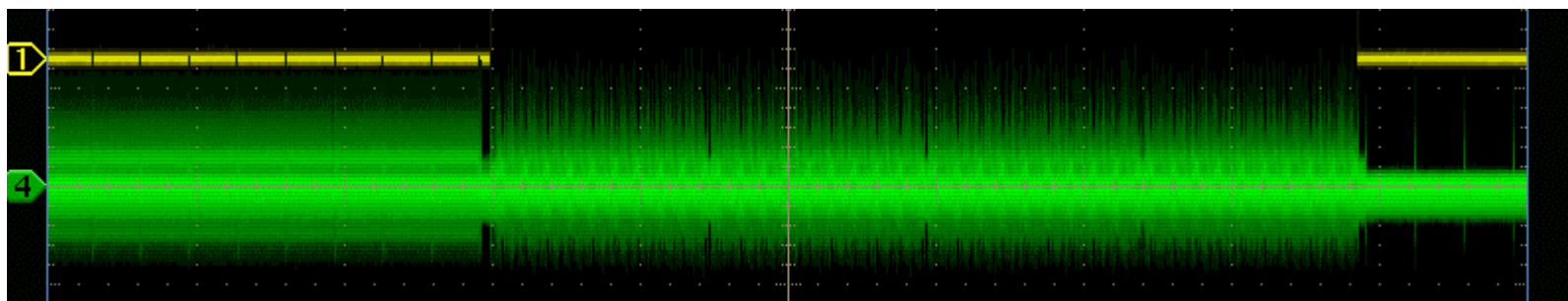
800 μ s/div

Previous slide: power signature of first 64-byte block sent (sign-on info?).
This slide: Power signature for remaining 64-byte blocks (delay varies).

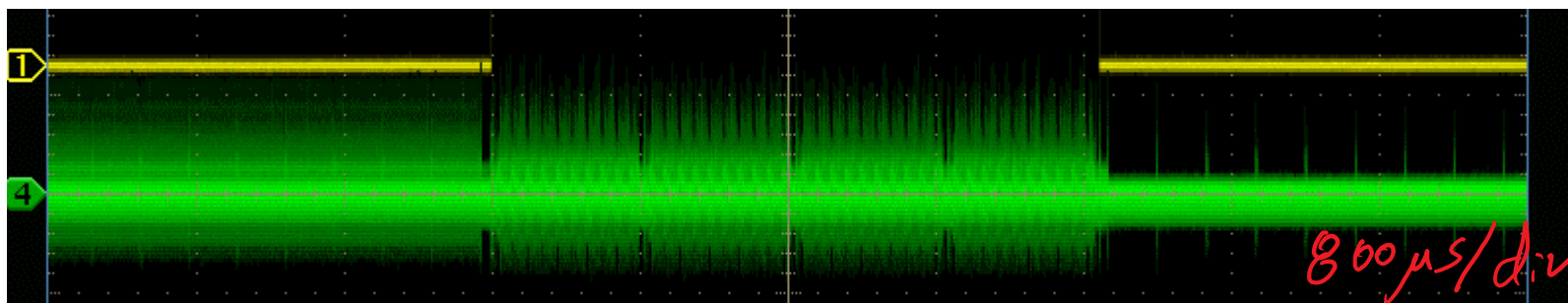
- SAM D20 Xplained Pro (109)
 - SAM D21 Xplained Pro (232)
 - 8MHz Oscillator Calibration Application - SAM D21 Xplained Pro
 - ADP example application - SAM D21 Xplained Pro
 - AES Software Library Demo - SAM D21 Xplained Pro
 - Alert Notification Client Application - SAM D21 Xplained Pro



ECB



CBC



CTR

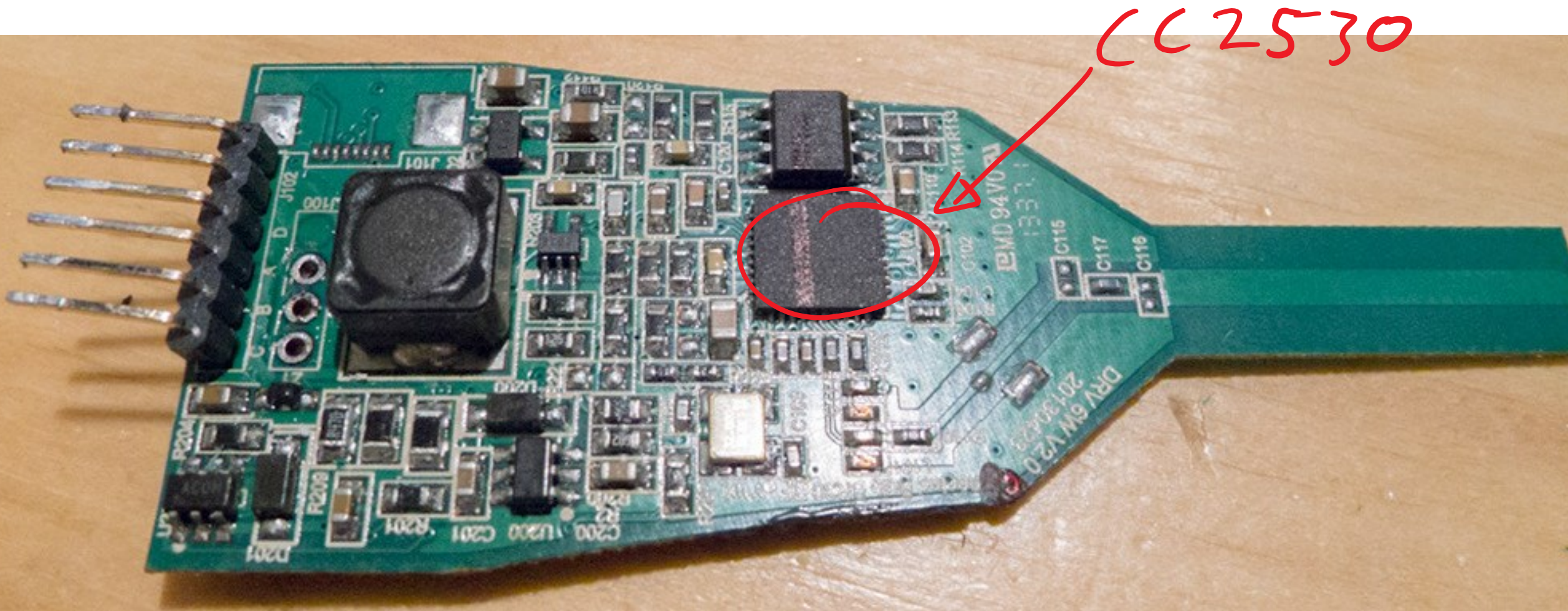
64 BYTE DECRYPTION

BR30

DOWN LIGHT

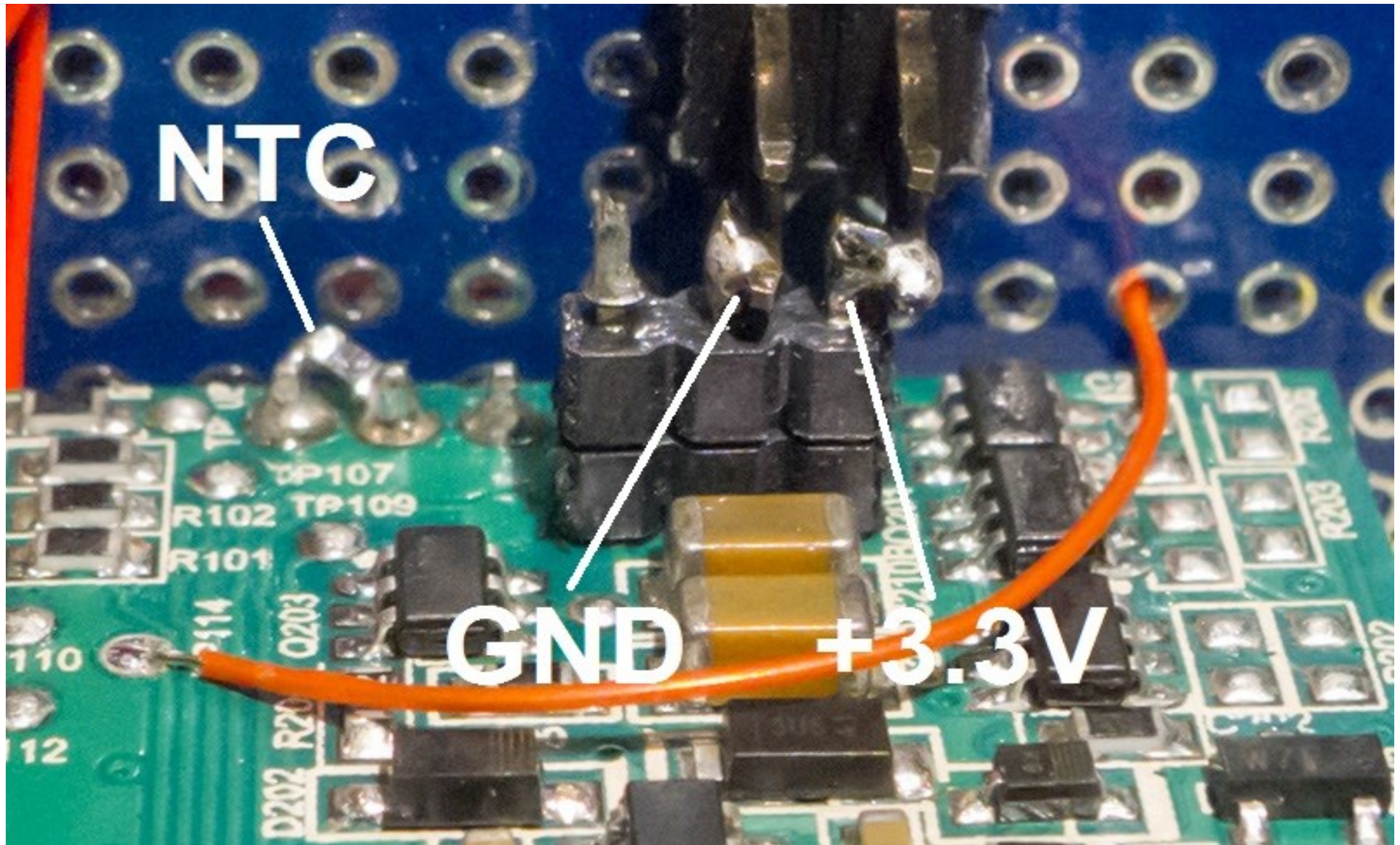
CC 2530 Based





CC2530



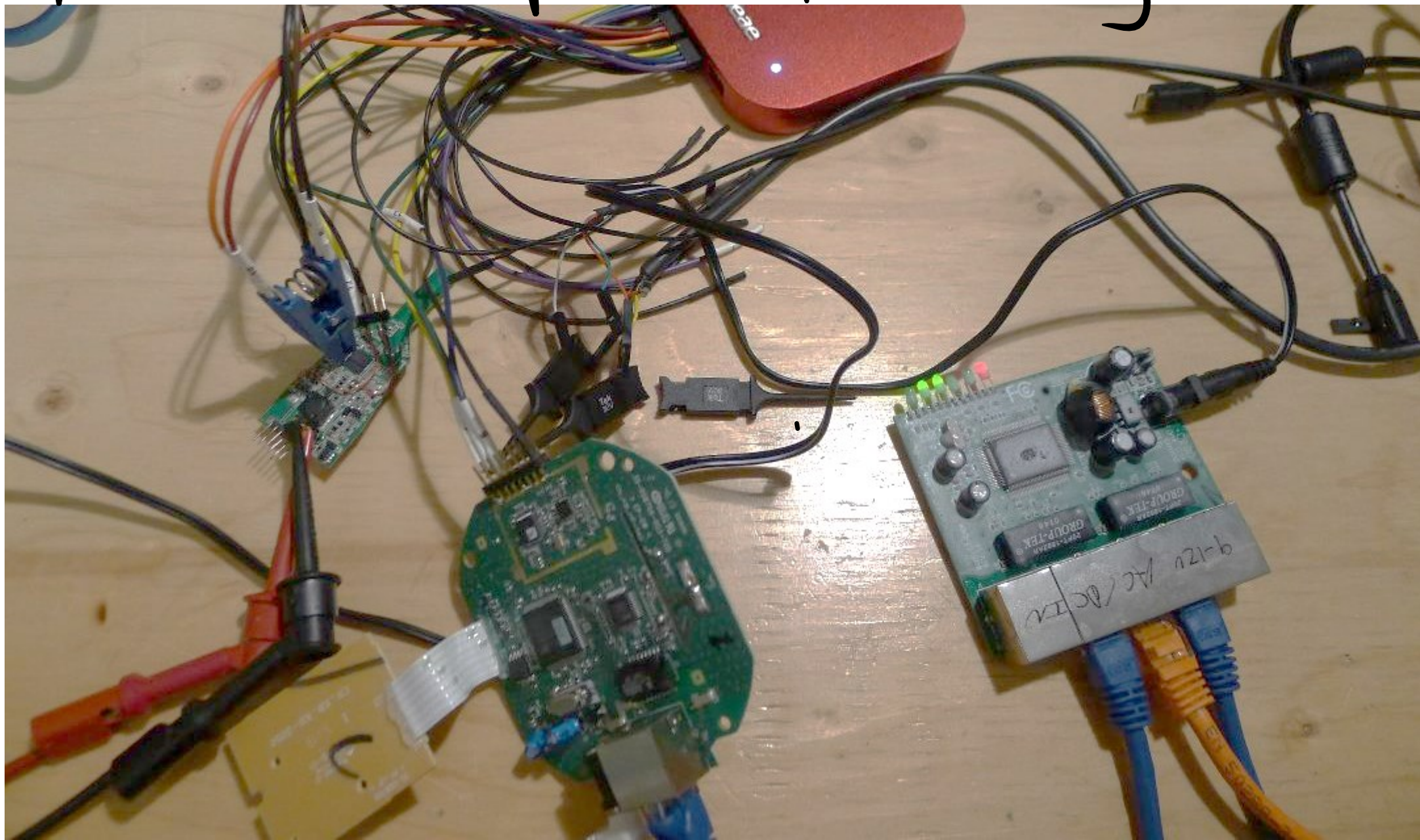


NTC

GND

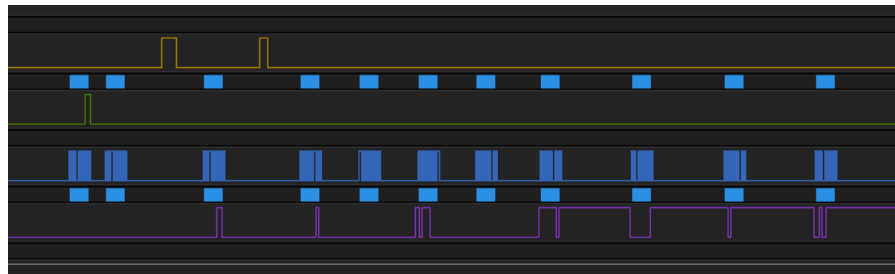
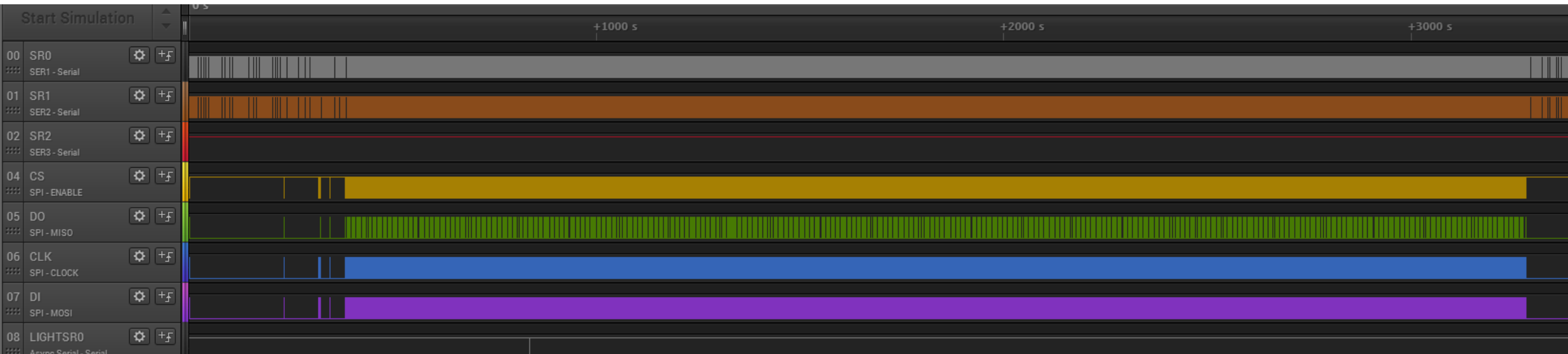
+3.3V

Firmware Update Hacking

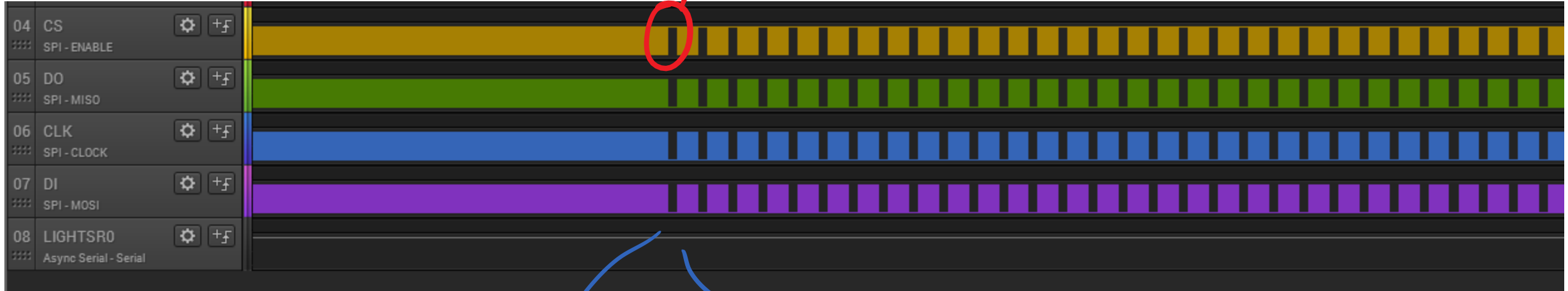


Using Salae Pro-16

↳ Capture 8MHz SPT Traffic



Page Erase



Pass #1

Pass #2

Flag

MiniPro v6.50

File(F) Select IC(S) Project(P) Device(D) Tools(V) Help(H) Language(L)

Select IC: MX25L4006E @SOP8

IC Information (No Project opened):
ChipType: EEPROM ChkSum: 0x04D1 B457
IC Size: 0x80000 Bytes

Product Identification: ChipID: C2 20 13

Set Interface: 40P adapter ICSP port ICSP_VCC Enable

Buff select: Code Memo Config

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
000000:	53	42	4C	31	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	SBL1.....
000010:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
000020:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
000030:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
000040:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

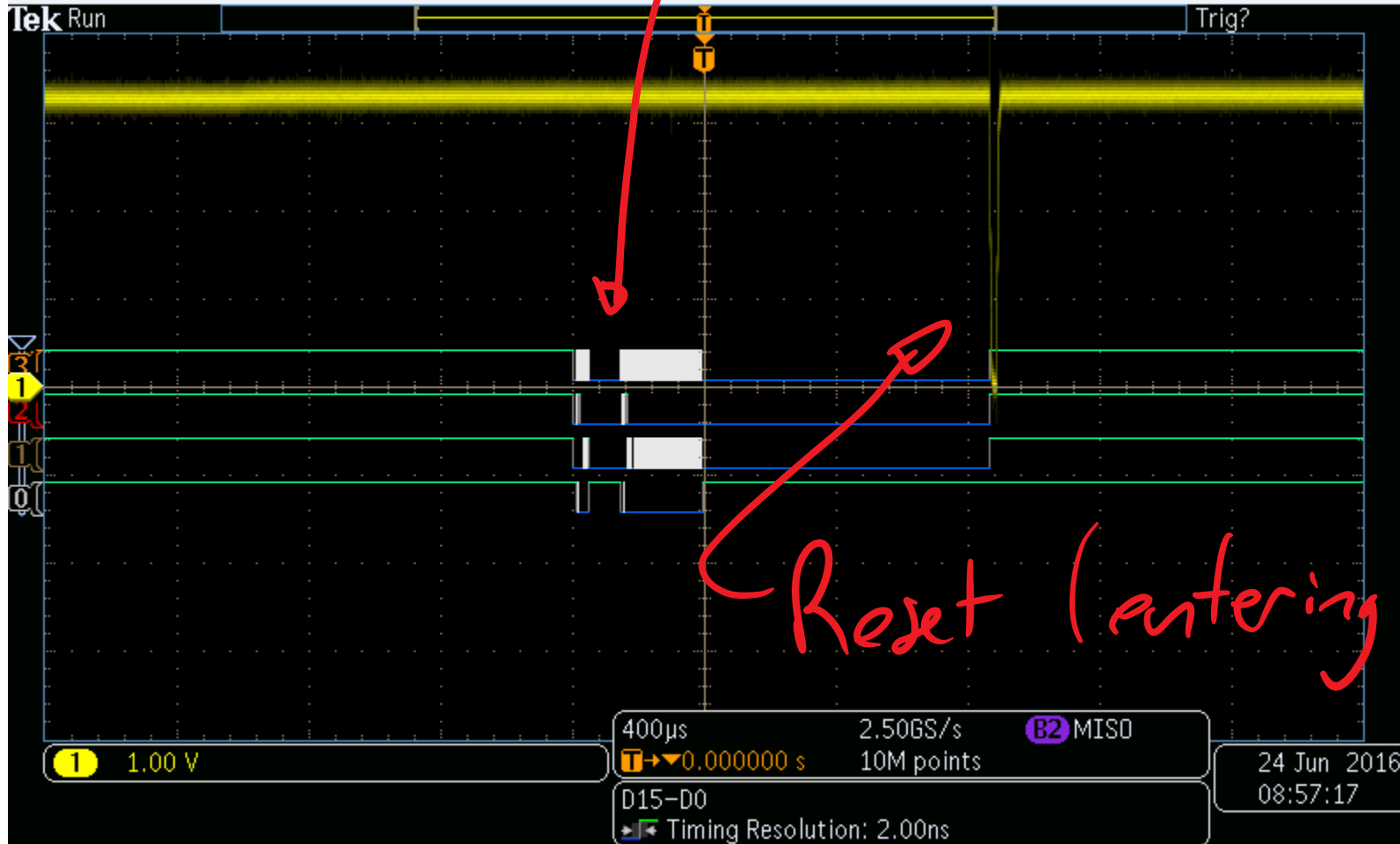
First block sent

000780:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
000790:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0007A0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0007B0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0007C0:	2A	00	01	00	00	66	52	14	10	02	17	30	39	03	EF	40	*....fR....09..@
0007D0:	2E	37	0B	25	EC	C0	47	65	CB	E1	1E	0E	74	F7	A1	14	.7.%..Ge....t...
0007E0:	EE	6B	58	B5	2F	F3	0D	83	68	12	67	71	4C	7A	75	20	.kX./...h.gqLzu
0007F0:	4D	08	E0	74	95	54	CE	AB	23	72	2B	80	AB	46	46	CD	M..t.T..#r+..FF.
000800:	77	CF	AC	2E	8C	58	9E	75	8C	1D	77	43	D5	A2	28	5C	w....X.u..wC..(\
000810:	4E	94	CC	F9	C8	C5	5B	62	E7	09	8B	E3	6A	3A	0C	07	N.....[b....j:..
000820:	86	27	80	7A	76	91	90	AA	1E	8F	40	FD	35	96	CC	C0	.'..zu.....@.5...
000830:	BF	53	2D	F0	88	7E	28	ED	F3	B7	96	AF	65	8C	8A	1D	.S-..~(.....e...
000840:	D6	8B	07	49	EE	8C	B7	49	54	D9	D9	62	94	62	65	0C	...I...IT..b.be.
000850:	99	E4	B8	4A	CE	17	26	28	A8	FF	F3	4C	48	45	B0	A0	...J..&(...LHE..
000860:	2E	29	3D	2A	4E	1D	40	42	C3	8A	9D	E0	D6	6E	47	98	.)=*N.@B.....nG.
000870:	D3	42	47	CF	29	EC	BC	88	CB	FB	35	15	CD	DB	8A	FE	.BG.).....5.....

SRAM Dump

↳ DURING BOOTLOAD

That block from previous page.



Reset (entering debug)

00000000	A7 7B 8B 33 11 A4 C9 33 84 A2 DE 32 5C DA E4 B0	§ { 1 3 ◀ * É 3 l e b 2 \ Ó ä °
00000010	EA 67 DE CF DF 6B 06 5E EF 41 2F 9E BE 7F 66 AE	ê g p l B k - ^ i A / l % l f @
00000020	A7 FA CB BB F6 FA B0 3C 17 FB 34 F9 9B F4 90 FB	§ ú È » ù ° < l ù 4 ù l ô ù
00000030	EE FF FD 77 8F F3 7B 76 DF 9E 79 63 84 EB FA B3	i ÿ ý w ö l f y c l y c l *
00000040	F9 6D C5 F6 F0 5B EF 00 00 FF FF 00 7D 01 B9 EF	ù m Ä ö ö [i ÿ ý } i i
00000050	66 00 D9 E7 00 00 06 0E 00 78 03 FF FA E6 00 30	ù ç - l y ú a 0
00000060	F0 0E C4 1C 00 4C F8 07 E6 00 30 F0 0E C4 1C 46	ö ð Ä l y æ ö ö } A F
00000070	F8 DE FA 9B BE 00 8C FA 00 E6 EF C2 EF A5 00 E6	o b ú l % l ú æ i Â i ¥ ö
00000080	07 2A 00 01 00 00 66 52 14 10 02 17 30 39 03 EF	• * f R q l j 0 9 l i
00000090	40 2E 37 0B 25 EC C0 47 65 CB E1 1E 0E 74 F7 A1 14 EE 6B 58 B5 2F F3 0D	@ l l ç - n - , ô H Y 7 * +
000000A0	9C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	l
000000B0	00 00 00 00 00 00 00 00 00 2A 00 01 00 00 66 52	* f R
000000C0	14 10 02 17 30 39 03 EF 40 2E 37 0B 25 EC C0 47	¶ l + 4 9 4 i @ . 7 1 % i Å G
000000D0	65 CB E1 1E 0E 74 F7 A1 14 EE 6B 58 B5 2F F3 0D	e È á ð t ÷ l ð l X µ l ó
000000E0	83 68 12 67 71 4C 7A 75 20 4D 08 E0 74 95 54 CE	l h l i y l L z u M Q à l i l i
000000F0	7B 23 72 2B 80 AB 46 46 CD F4 3E 66 00 53 42 4C	« # r + l « F l i ô > f S B L
00000100	81 FF 00 00 00 00 00 E8 FB 01 02 03 01 09 5E B5	l ý e ù 7 ^ µ
00000110	A9 68 FE F8 BF 6B FB 79 F5 1F 89 8F B5 F9 E2 7E	@ f b o è x y 7 l µ ù â ~
00000120	F3 CD 44 5D 3B B9 FB 1B FA 60 DD FF DF 1F 72 FE	ó í D] ; ' l + ö ÿ ÿ r b
00000130	17 99 3A 97 DE FF 4E 33 78 7D 06 DE C7 71 AD DF	l i : l b ý N 3 x } - P Ç q - B
00000140	CE BE B2 EB 4C D4 CE 05 0E BD BB 6C BD 23 D4 BF	i % * è L Ô i ð % » i % # Ô ÷
00000150	1D 75 BE F5 E6 FF 95 BB DD C3 BD 11 DF F0 DF 88	u % ö æ ý l » Ý Ä % B ö B l
00000160	8E 89 7F FD 51 11 E7 DA 7E 7C AC 4B D4 AF 7D 5C	l l l ý Q ◀ ç Ó ~ - K O ~ } \
00000170	74 CD A9 EE DF 29 C7 BB 8D B8 EF BB FE D1 BD F3	t í @ i B] Ç » , i » b N X ó
00000180	A5 F9 5C 57 A6 B4 96 FF B0 7F BD 7E D1 7F AA 4F	¥ ù \ W l ' i ý ° l % ~ Ñ i ÷ O
00000190	F6 FF FE 3D 6F FB A9 F7 F4 0F 6E 7D 66 E2 FB C5	ö ý ð = o û @ ÷ ô ð n } f â û Å
000001A0	EB 9E FF 7F F6 B1 3D 0F BA 3C 5E 6F CA 65 4D CB	è l ý l ö ± = ð ° < ^ o È e M È
000001B0	B1 8F FD F8 FB 4F F7 A3 F9 0F ED 38 FD 55 D1 ED	± ý o û O ÷ £ ù ð í 8 ý U Ñ í
000001C0	53 1B 87 9D 92 AE DF 95 F2 BB 54 1E DB 28 7B 5D	S + l ' @ B l ð » T Ô ({ }
000001D0	FB EC EB AF D6 2E E1 63 BD B3 37 D3 AE 83 FB B2	û i è ~ Ö . á c % * 7 Ó @ i û ^
000001E0	76 3D 3D CB 31 BF DA 15 67 E7 6E DF EB 7F 4D 5C	v = = È l ÷ Ó l g ç n B è l M \
000001F0	4C 7F 28 F7 4F DE 25 91 FF 2A E5 03 7E ED D7 77	L l (÷ O b % ' ý * á l ~ i x w
00000200	81 C7 B8 DC DA B7 AF 7C 2F 6A B5 F9 31 62 14 FC	Ç , Ü Ó . ~ / j µ ù l b ¶ ü

Block

First 16 bytes of block


what is this 16 bytes?

SRAM

Dump

ILU? Signature?

Address of SPI??



SECURITY
CONCLUSIONS

① Huge risk to Philips if worm designed.

② Good security practices in place to prevent this:

- Encrypted FW
- Signed FW (Linux only)
- Keep Keys out of SRAM
- Clear memory when done.

③ Trade-offs May cause future problems:

- Same key decrypts FW updates across many devices.
- ZLL master key leak opens up lamp-stealing.
- Huge Linux binary does a lot, vulns?
- See White Paper for more!

Eyal Ronen

<http://www.wisdom.weizmann.ac.il/~eyalro/>

Colin O'Flynn

@colinoflynn

oflynn.com

newae.com

coflynn@newae.com

Also read the
w.p. for details.