OWASP
Open Web Application Security Project

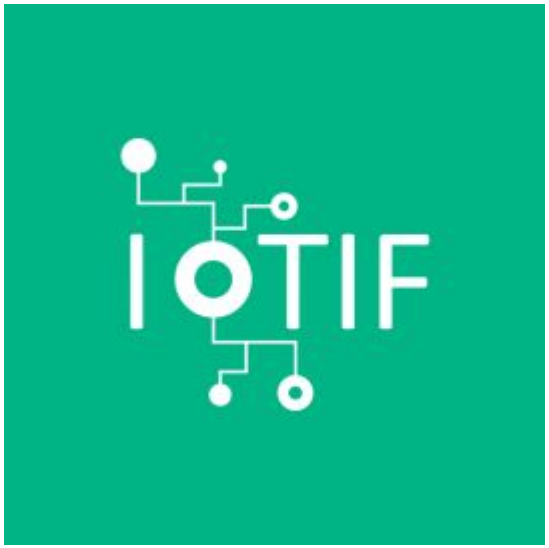
Exploring the IoT attack surface

Breaking || Liberating the brave new IoT world.

doc.dr.sc. Tonimir Kišasondi

Sponsored by

http://iot.foi.hr



Why should i listen to you and not go out for coffee?

Well, IoT is the next big hotness…

Everyone and their grandma is crowdfunding the next big smart thing, and it runs GNU/Linux or has a arduino in it!

- It’s like the 90s of INFOSEC, but with gadgets and the startup mentality!

IoT? What is that?

The Internet of Things (IoT) is the network of physical objects—devices, vehicles, buildings and other items—embedded with electronics, software, sensors, and network connectivity that enables these objects to collect and exchange data.

… creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit...

IoT...

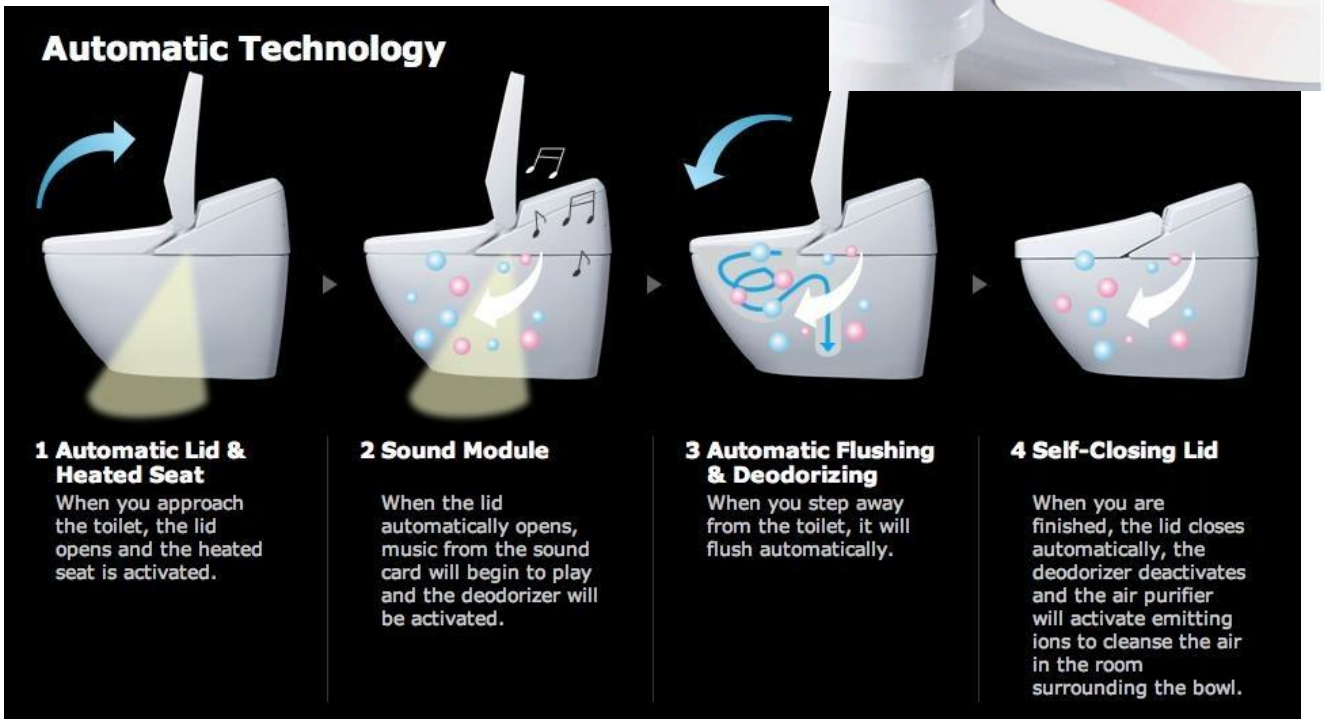

Bluetooth
My SATIS


wash ass


warm
ass


dry ass


Smart toilet cover


Automatic Technology
1 Automatic Lid & Heated Seat
When you approach the toilet, the lid opens and the heated seat is activated.
2 Sound Module
When the lid automatically opens, music from the sound card will begin to play and the deodorizer will be activated.
3 Automatic Flushing & Deodorizing
When you step away from the toilet, it will flush automatically.
4 Self-Closing Lid
When you are finished, the lid closes automatically, the deodorizer deactivates and the air purifier will activate emitting ions to cleanse the air in the room surrounding the bowl.

Trustwave SpiderLabs research:

http://www.forbes.com/sites/kashmirhill/2013/08/15/heres-what-it-looks-like-when-a-smart-toilet-gets-hacked-video/

Source:
http://www.symantec.com/connect/blogs/how-my-tv-got-infected-ransomware-and-what-you-can-learn-it

The FBI Can Neither Confirm Nor Deny Wiretapping Your Amazon Echo

Matt Novak

Yesterday 5:00pm · Filed to: SURVEILLANCE

198 17

An Amazon Echo, which the FBI can neither confirm nor deny has ever been hacked during an investigation (Gizmodo)

JEDINSTVENA METODA

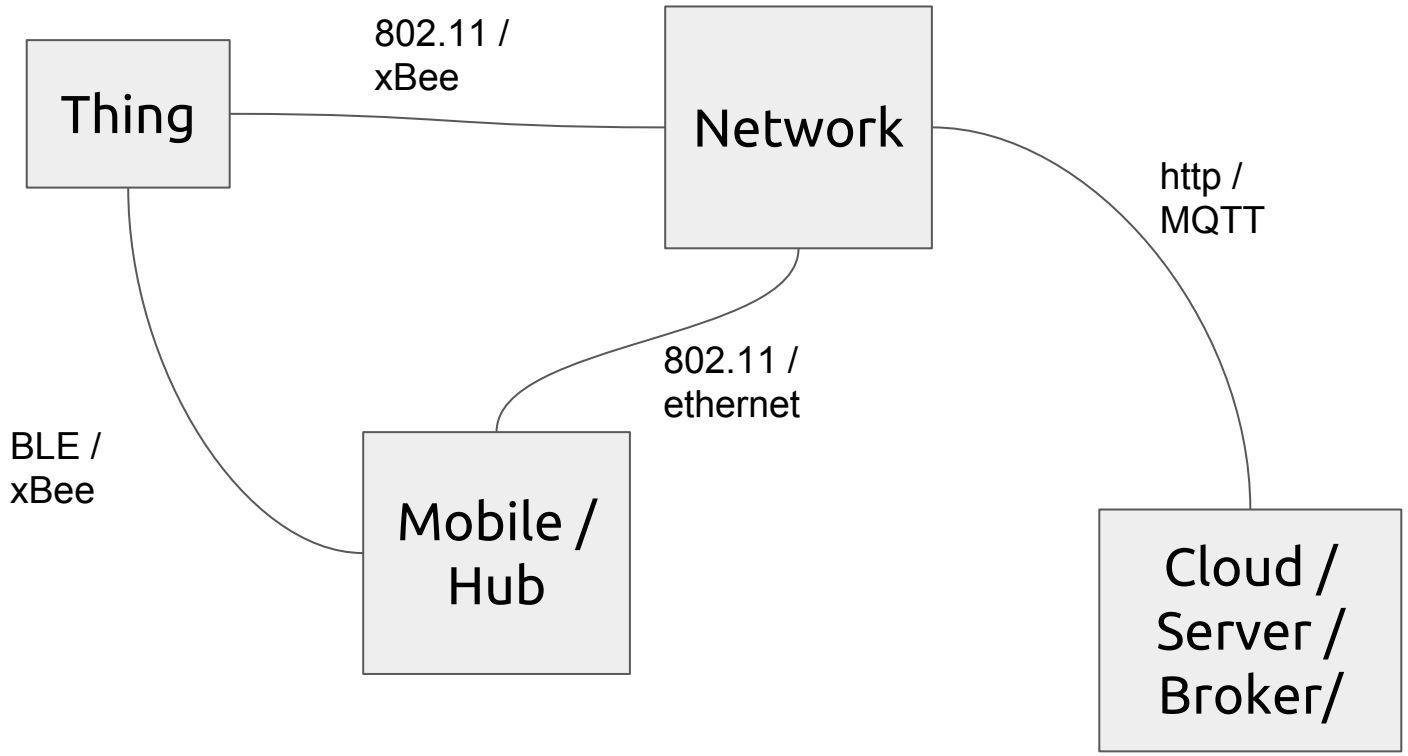
Kako je spriječena velika pljačka Zabe: Pink Panthere su u stanu prisluškivali pametnim TV-om

AUTOR: Dušan Miljuš OBJAVLJENO: 12.05.2016. u 13:12

Spektakularna pljačka poslovnice Zagrebačke banke, koja se trebala dogoditi 2. lipnja prošle godine, zbog koje se sudi skupini optuženih koja se naziva i balkanskim ogrankom Pink Panthera, spriječena je zahvaljujući prisluškivanju putem daljinskog aktiviranja snimača u smart televizoru, jedinstvenoj tehnici tajnog nadzora koju su koristili hrvatski policajci.

Prvookrivljeni **Dejan Kostić** i skupina optuženika snimani su u dnevnom boravku unajmljenog stana u Novom Zagrebu tako što su policajci putem IP adrese aktivirali snimač na pametnom Samsung televizoru koji je zabilježio razgovor osumnjičenika

Architecture of IoT ecosystem (simplified)



OWASP IoT Top 10

I1 Insecure Web Interface

I2 Insufficient Authentication/Authorization

I3 Insecure Network Services

I4 Lack of Transport Encryption/Integrity Verification

I5 Privacy Concerns

I6 Insecure Cloud Interface

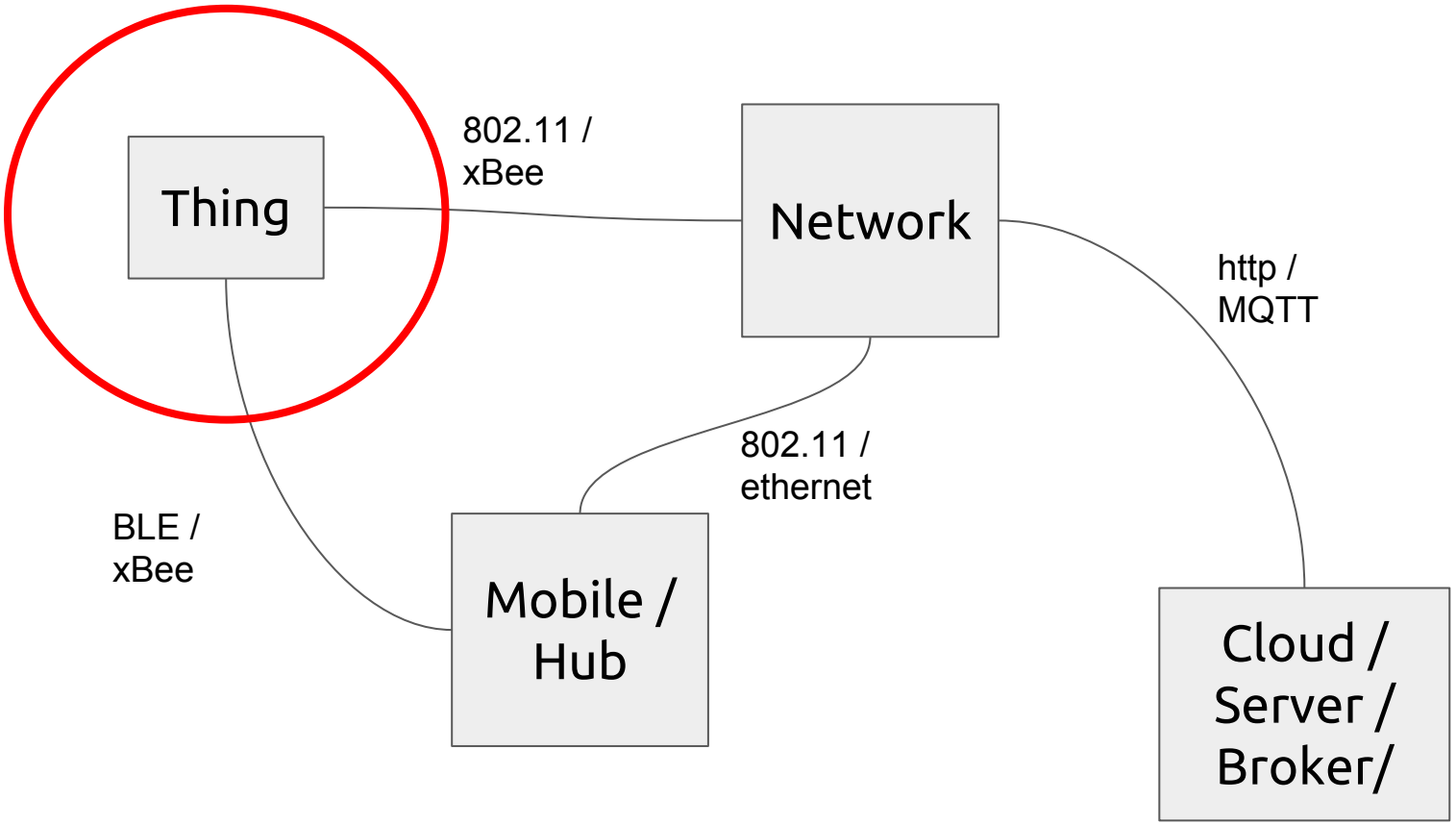
I7 Insecure Mobile Interface

I8 Insufficient Security Configurability

I9 Insecure Software/Firmware

I10 Poor Physical Security

Architecture of IoT ecosystem (simplified)



So? What's the attack surface on the thing?

1. Obtaining system access with physical access
    a. UART, JTAG?
2. Obtaining firmware, filesystem or local data storage/images
    a. In-Situ, JTAG, eMMC, USB, WTF
3. Analyzing firmware images
4. Software and firmware vulnerabilities

What i won’t cover:

1. Software analysis (Well, a little bit anyway...)
2. Radio / RF analysis
3. A lot of other stuff…

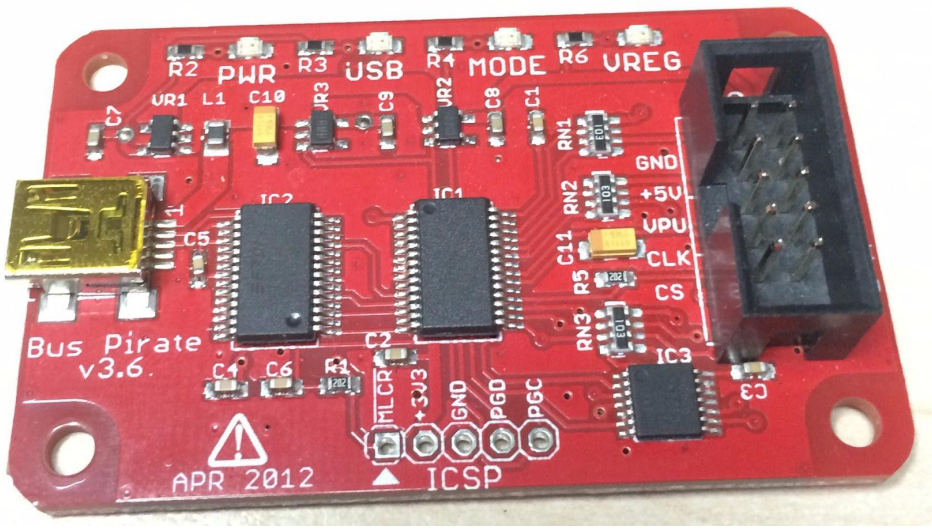
We are looking for easy victories...

Why am i doing this?

It’s simple: because i can :)

1) How do you measure security of a thing?
2) Do we have a testing methodology for embedded/IoT?
3) Is the IoT top 10 relevant?

OK, what do i need?



BusPirate / Shikra - 30$

FTDI cable -> 15ish

EZ-Hooks - 30$

…

SDR (HackRF, Ubertooth, YS1, RfCat...)

Jtagulator...

Why should i bother hacking my (smart blub|toilet |fridge)?

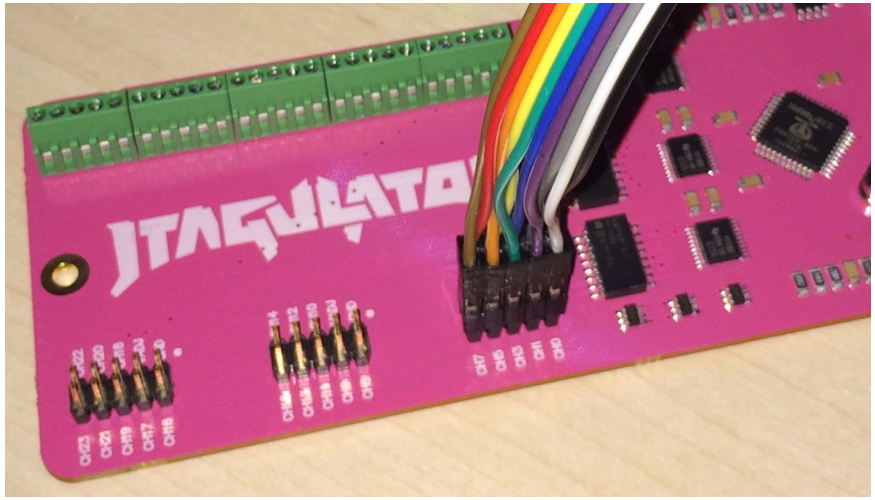
- Jailbreak or Security analysis

1) Factory set credentials like passwords
2) GPG signing keys and password in the firmware updates
3) Full access to binaries, source files for web applications
    - Usually written by EE devs.

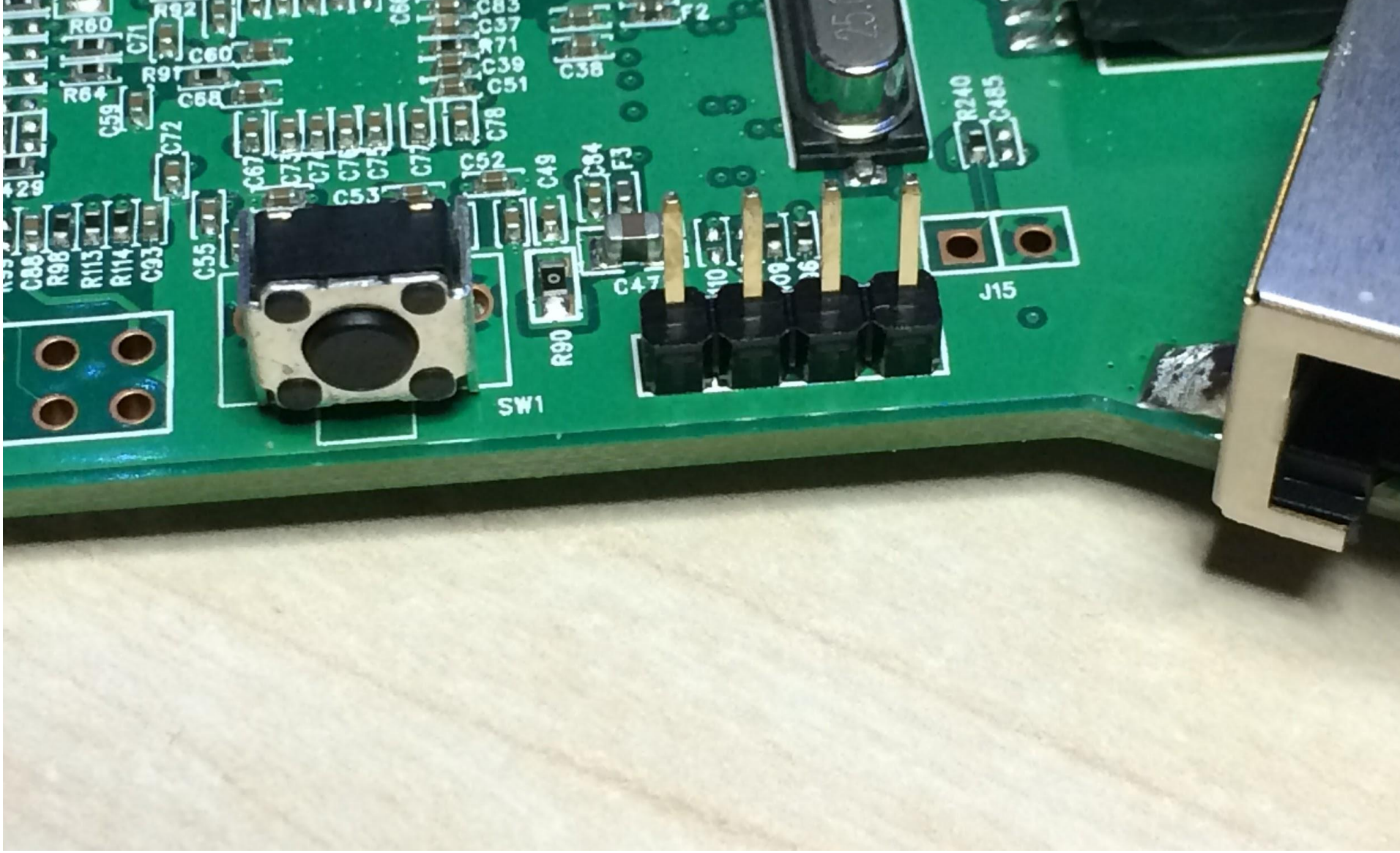
I have a cunning plan! -> Variant for noobs

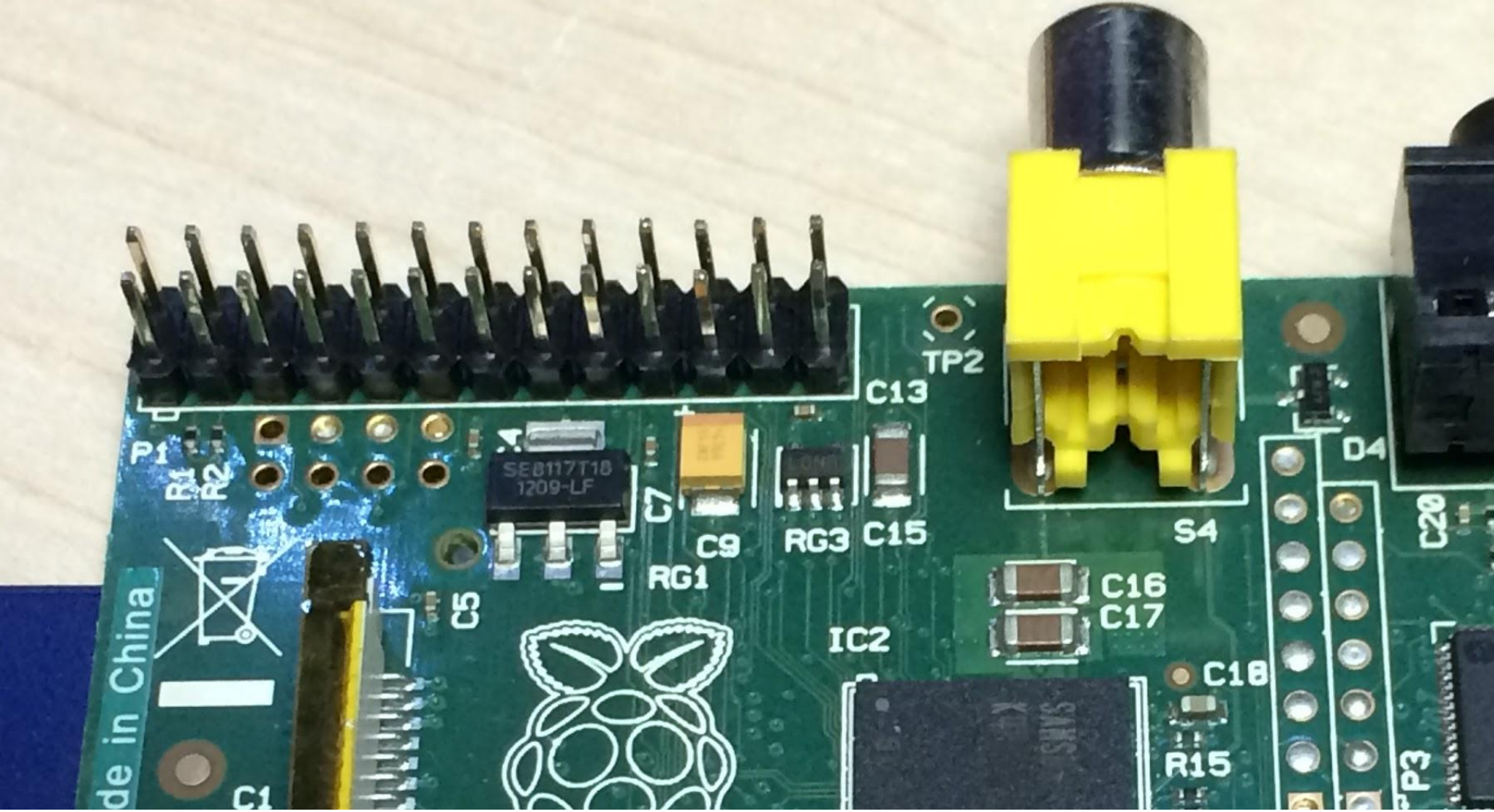
1) Find UART with serial console
2) Connect to UART (screen + buspirate)
3) Root shell
4) ???
5) Profit!

How to find UART ports?

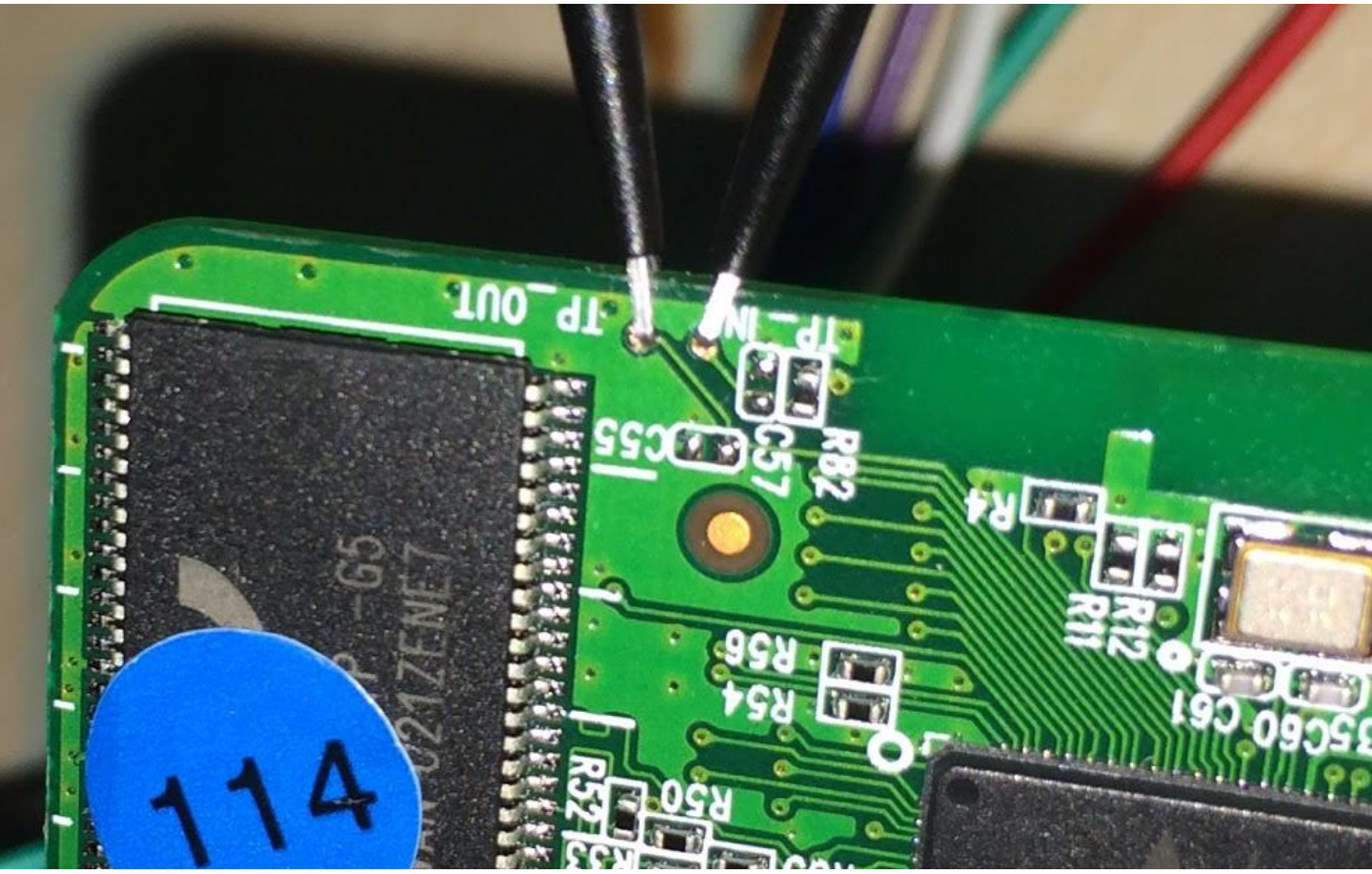


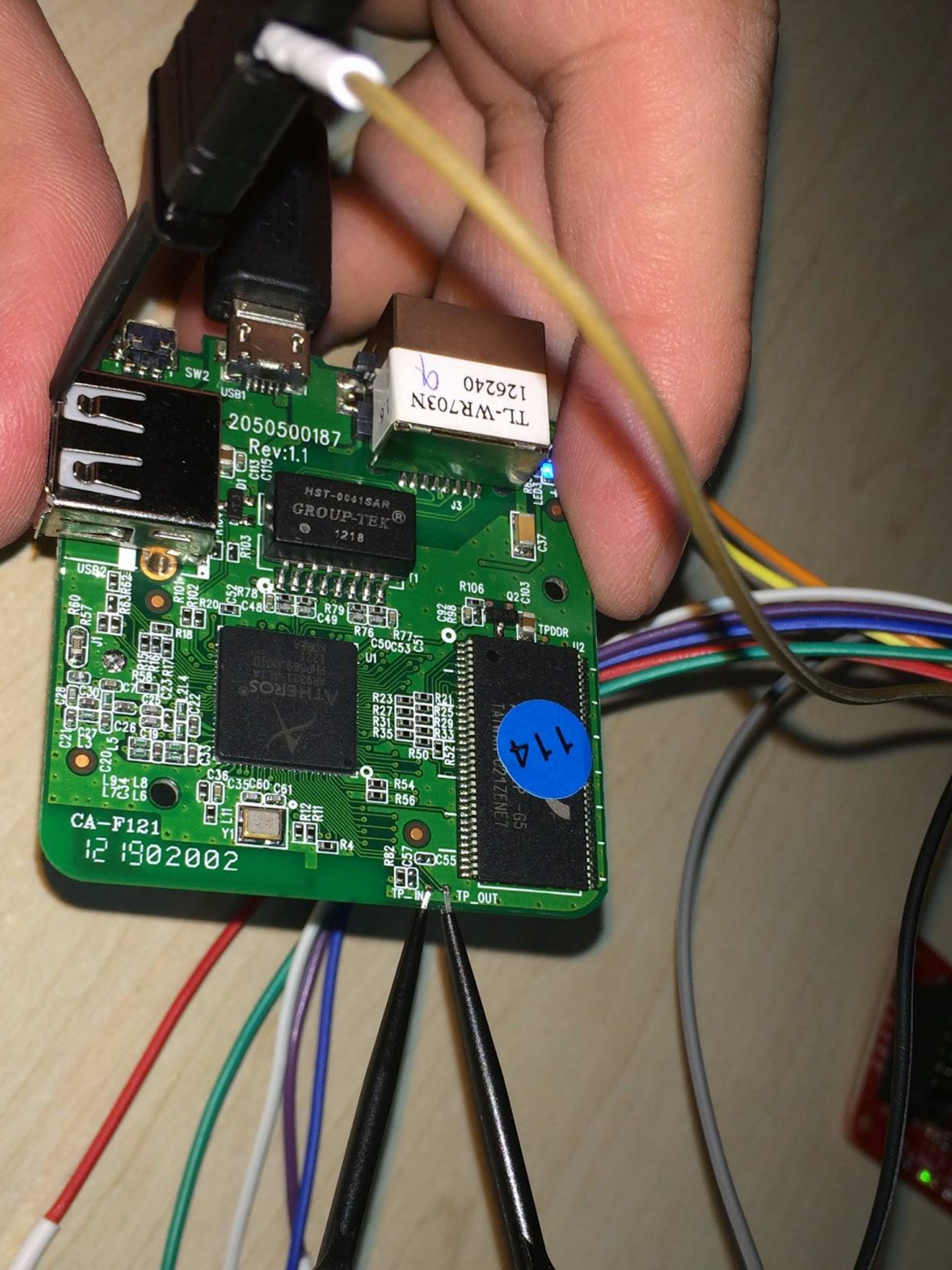
U303
MXIC
25L512C
MI-12G
C504
C261
RX
TX
3V
GND
U300
U308
Q301
L0307
TP310
SAMSUNG 034
K9F1208U0C
PCB0
TP303
TP305
TP307
TP309

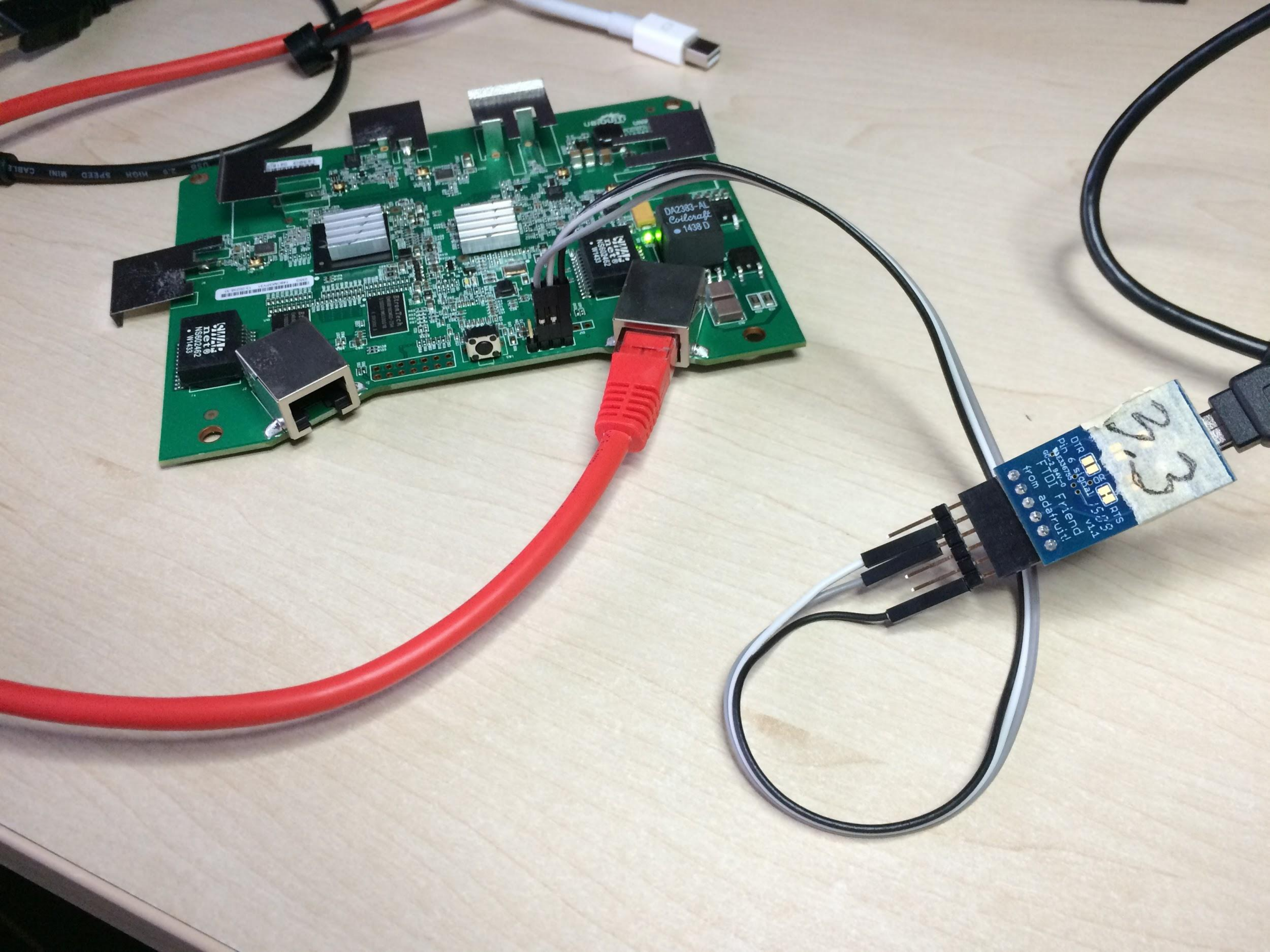

R60
C71
C60
R91
R64
C59
C68
C72
C67
C73
C74
C76
C75
C77
C52
C49
C53
C55
C88
R98
R113
R114
C93
SW1
R90
C47
C64
F3
C83
C37
R71
C39
C51
C38
F2
R240
C485
J15


TP2
C13
P1
R1
R2
SE8117T18
1209-LF
C7
C9
RG3
C15
S4
D4
C20
RG1
C5
C16
C17
IC2
C18
R15
P3
C1
de in China


GND
LEDW
LEDL
R44R46R48
R47
R45
R49


TP_OUT
TP_IN
C55
C57
R62
R4
R12
R11
R56
R54
R50
C61
114


TL-WR703N
126240
2050500187
Rev:1.1
HST-0041SAR
GROUP-TEK®
1218
SW2
USB1
USB2
J3
TPDDR
U1
U2
114
CA-F121
TP_IN
TP_OUT

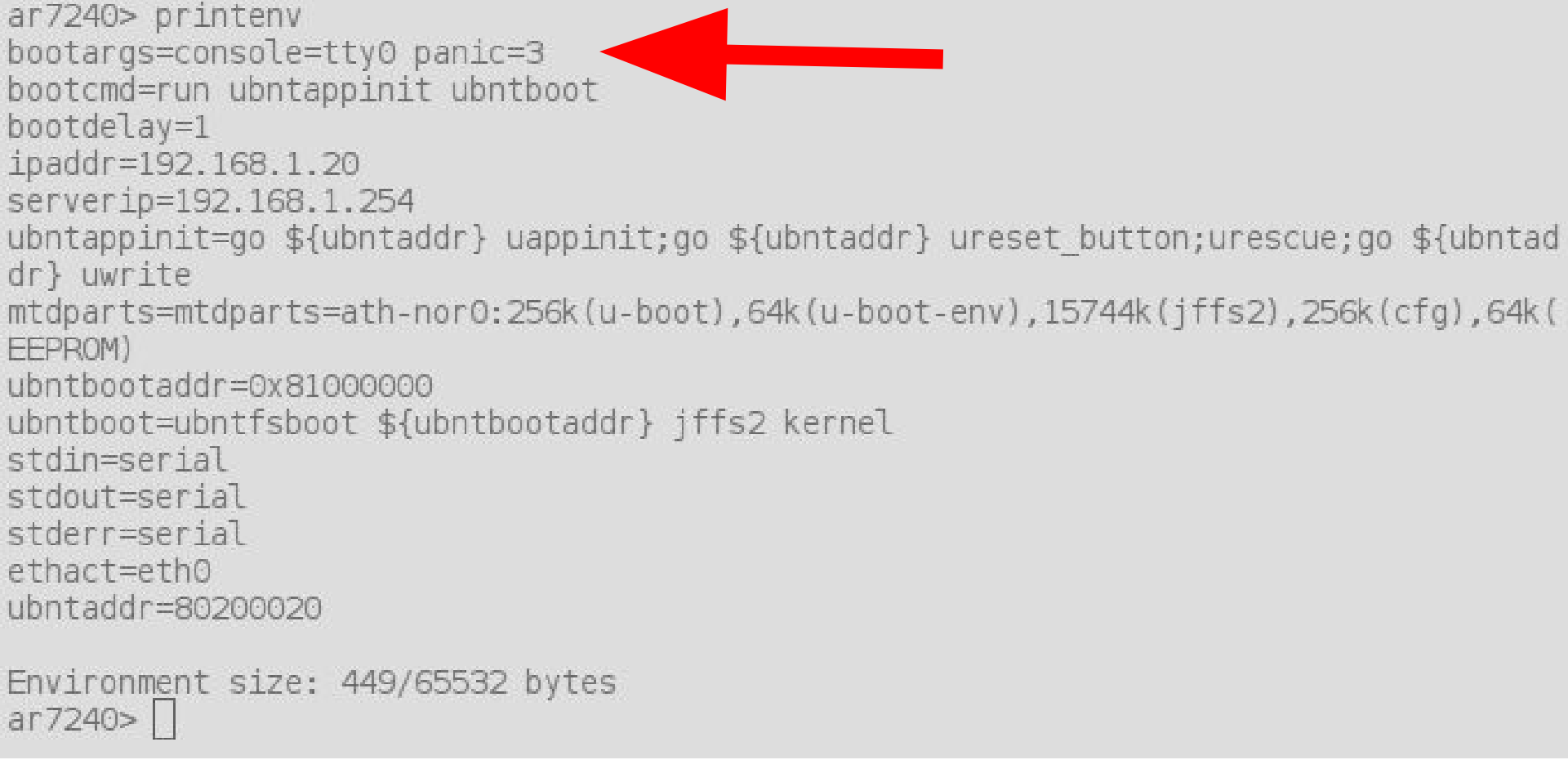

DA2383-AL
Coilcraft
1438 D
FTDI Friend
from adafruit!
3.3

```
U-Boot unifi-v1.5.2.206-g44e4c8bc (Aug 29 2014 - 18:02:07)

DRAM:  128 MB
Flash: 16 MB
PCIe WLAN Module found (tries: 1).
Net:   eth0: 04:18:d6:02:f4:2d
eth0
Setting 0xb8116290 to 0x20402d0f
Board: Copyright Ubiquiti Networks Inc. 2014
Hit any key to stop autoboot:  0
Board: Ubiquiti Networks AR9344 board (e507-31.2123.0030.0030)
UBNT application initialized
Scanning JFFS2 FS: . done.
## Booting image at 81000000 ...
   Image Name:   MIPS Ubiquiti Linux-2.6.32.33
   Created:      2015-01-19  22:16:17 UTC
   Image Type:   MIPS Linux Kernel Image (lzma compressed)
   Data Size:    4497416 Bytes =  4.3 MB
   Load Address: 80002000
   Entry Point:  80002000
   Verifying Checksum at 0x81000040 ...OK
   Uncompressing Kernel Image ... OK

Starting kernel ...

Booting Atheros AR934x
```


ar7240> printenv
bootargs=console=tty0 panic=3
bootcmd=run ubntappinit ubntboot
bootdelay=1
ipaddr=192.168.1.20
serverip=192.168.1.254
ubntappinit=go ${ubntaddr} uappinit;go ${ubntaddr} ureset_button;urescue;go ${ubntad
dr} uwrite
mtdparts=mtdparts=ath-nor0:256k(u-boot),64k(u-boot-env),15744k(jffs2),256k(cfg),64k(
EEPROM)
ubntbootaddr=0x81000000
ubntboot=ubntfsboot ${ubntbootaddr} jffs2 kernel
stdin=serial
stdout=serial
stderr=serial
ethact=eth0
ubntaddr=80200020
Environment size: 449/65532 bytes
ar7240>

```
setenv bootargs 'console=ttyS0,115200
init=/bin/sh'
saveenv
boot
```

Plan B -> NAND Glitch

No UART/Shell… Let’s try another approach...

Plan A -> Fetch a image from the vendor’s page

Plan B -> Intercept OTA update

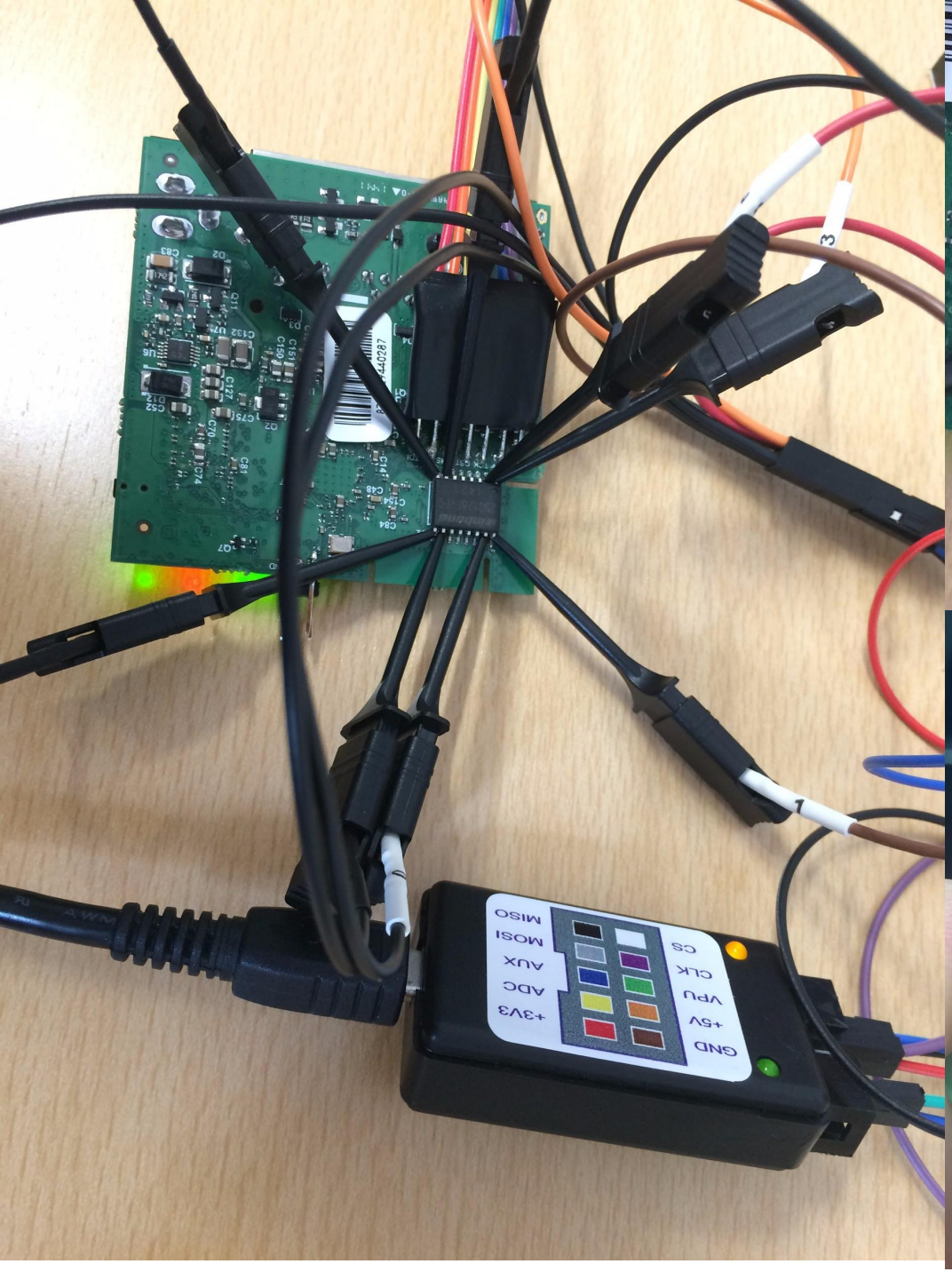
Plan C -> NVRAM dump (still under 30-100)

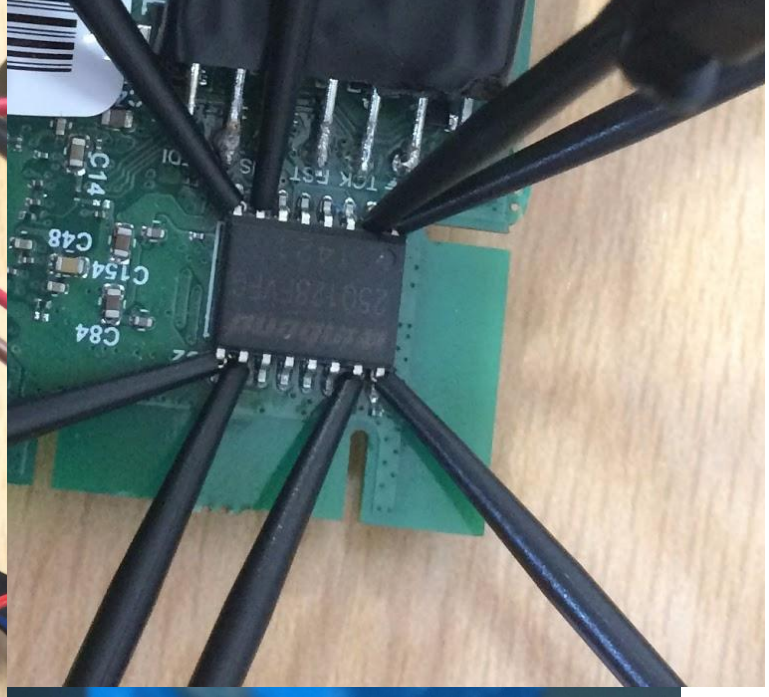
Plan C+½ -> Unsolder Chip, read in an adapter

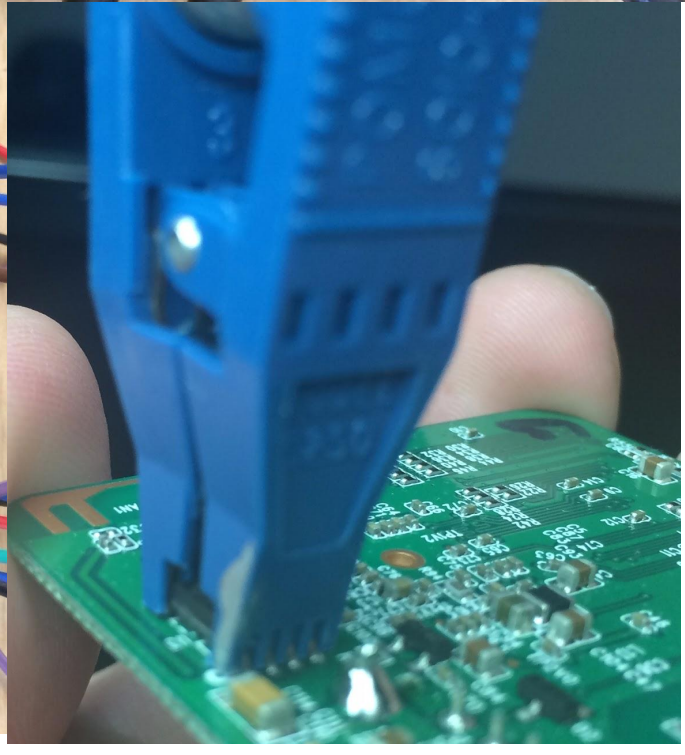
- Aliexpress is fun, they have all kinds of stuff...

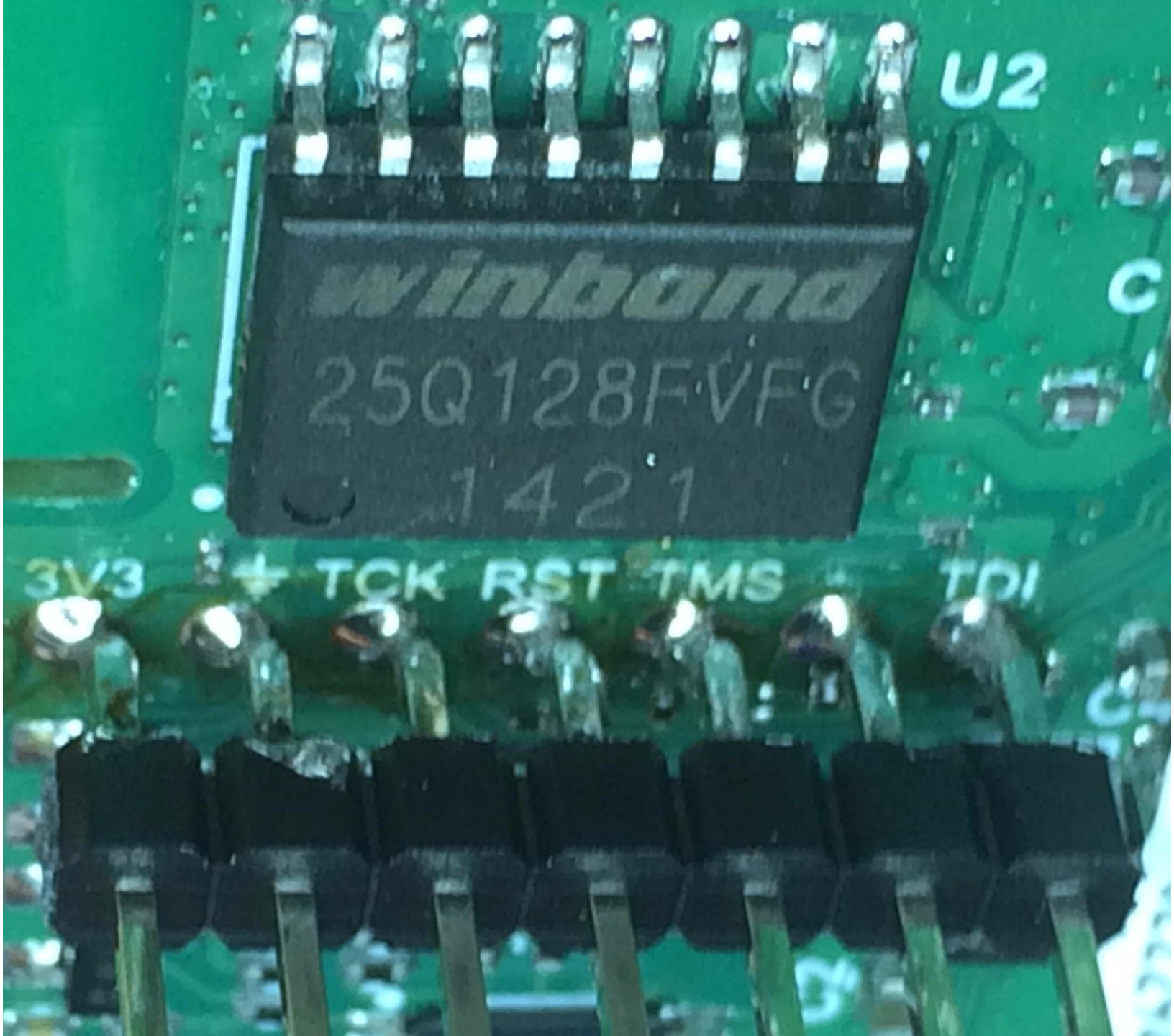
Plan C -> JTAG (30-150$)

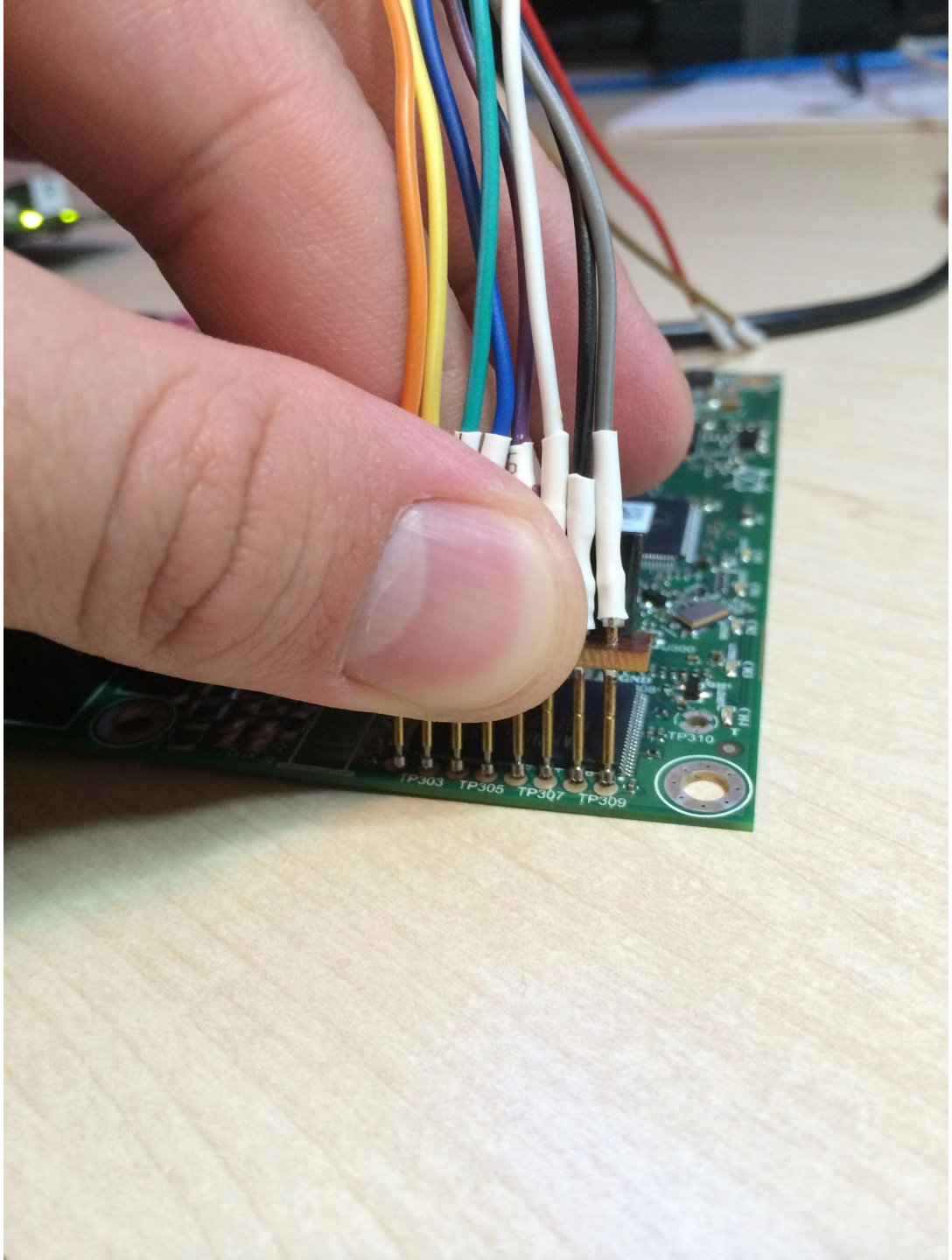
Plan Arduino -> avrdude -p m328p -P usb -c usbtiny -U flash:r:flash.bin:r


GND
+5V
VPU
CLK
CS
+3V3
ADC
AUX
MOSI
MISO






U2
winbond
25Q128FVFG
1421
3V3
TCK
RST
TMS
TDI


TP303
TP305
TP307
TP309
TP310


STATUS
RESET
CH23
CH22
CH21
CH20
CH19
CH18
CH17
CH16
CH15
CH14
CH13
CH12
CH11
CH10
CH9
CH8
CH7
CH6
CH5
CH4
CH3
CH2
CH1
CH0

Useful software tools

FlashRom - https://www.flashrom.org

OpenOCD - http://openocd.org/

QEMU - http://wiki.qemu.org/Main_Page

```
# flashrom -p
buspirate_spi:dev=/dev/ttyUSB0,spispeed=1M
```

```
# flashrom -p
buspirate_spi:dev=/dev/ttyUSB0,spispeed=1M -r
firmware-tikmap.bin
```

OK, i got a image, what now?

Binwalk - http://binwalk.org/

Sasquatch - https://github.com/devttys0/sasquatch

Firmwalker - https://github.com/craigz28/firmwalker

And:

~/Downloads > binwalk tik-map.bin

DECIMAL HEXADECIMAL DESCRIPTION

4096 0x1000 Squashfs filesystem, little endian, version 4.0, compression:xz, size: 9460162 bytes, 1173 inodes, blocksize: 262144 bytes, created: 2016-05-02 10:47:08

9465991 0x907087 Executable script, shebang: "/bin/bash"

9466452 0x907254 Executable script, shebang: "/bin/bash"

9466533 0x9072A5 ELF, 32-bit MSB MIPS64 executable, MIPS, version 1 (SYSV)

9496929 0x90E961 Unix path: /sys/devices/system/cpu

9501329 0x90FA91 ELF, 32-bit MSB MIPS64 executable, MIPS, version 1 (SYSV)

9578451 0x9227D3 xz compressed data

9578479 0x9227EF xz compressed data

10651796 0xA28894 xz compressed data

10748985 0xA40439 Unix path: /var/pdb/system/crcbin

```
~/Downloads > binwalk firmware\ \(1\).bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             Ubiquiti firmware header, header size: 264 bytes, ~CRC32: 0x6219681, ve
rsion: "MF.ar933x.v2.1.11.1309.150406.1423"
260           0x104           Ubiquiti partition header, header size: 56 bytes, name: "PARTu-boot", b
ase address: 0x00000000, data size: 0 bytes
184868        0x2D224         Certificate in DER format (x509 v3), header length: 4, sequence length:
 64
194660        0x2F864         CRC32 polynomial table, big endian
220616        0x35DC8         Ubiquiti end header, header size: 12 bytes, cumulative ~CRC32: 0x454E44
2E
229764        0x38184         Ubiquiti partition header, header size: 56 bytes, name: "PARTkernel", b
ase address: 0x00000001, data size: -2147475456 bytes
229828        0x381C4         uImage header, header size: 64 bytes, header CRC: 0x6A6C3610, created:
2015-04-06 21:29:56, image size: 1027674 bytes, Data Address: 0x80002000, Entry Point: 0x80002000, da
ta CRC: 0x491BBBE5, OS: Linux, CPU: MIPS, image type: OS Kernel Image, compression type: lzma, image
name: "MIPS Ubiquiti Linux-2.6.32.29"
229892        0x38204         LZMA compressed data, properties: 0x5D, dictionary size: 8388608 bytes,
 uncompressed size: 2992416 bytes
1257566       0x13305E        Ubiquiti partition header, header size: 56 bytes, name: "PARTrootfs", b
ase address: 0x00000002, data size: 0 bytes
1257630       0x13309E        Squashfs filesystem, little endian, version 4.0, compression:lzma, size
: 5977655 bytes, 1385 inodes, blocksize: 131072 bytes, created: 2015-04-06 21:29:59
```

```
~/Downloads > binwalk UVC.gen2.v3.2.0.57.a32ae49.160411.0337.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             Ubiquiti firmware header, header size: 264 bytes, ~CRC32: 0x14894E67, v
ersion: "UVC.A5S.v3.2.0.57.a32ae49.160411.0337"
260           0x104           Ubiquiti partition header, header size: 56 bytes, name: "PARTpri", base
 address: 0x00000001, data size: -1071611904 bytes
17128         0x42E8          gzip compressed data, maximum compression, from Unix, last modified: 20
16-04-11 03:37:29
2163012       0x210144        Ubiquiti partition header, header size: 56 bytes, name: "PARTlnx", base
 address: 0x00000002, data size: 0 bytes
2163076       0x210184        UBI erase count header, version: 1, EC: 0x0, VID header offset: 0x200,
data offset: 0x800
```

```
$ ls -alh /etc/init.d
total 52
drwxr-xr-x    2 root     root         904 May 12 12:11 .
drwxr-xr-x   17 root     root        3.0K May  1 21:14 ..
-rwxrwxrwx    1 root     root        2.2K Jul 24  2015 1S41wifi
-rwxrwxrwx    1 root     root        2.0K Jul 24  2015 S10udev
-rwxr-xr-x    1 root     root        6.7K Oct 29  2015 S11init
-rwxrwxrwx    1 root     root        6.4K Mar 18 10:38 S11init.bak
-rwxrwxrwx    1 root     root        1.4K Jul 24  2015 S12syscfg
-rwxrwxrwx    1 root     root        1.3K Jul 24  2015 S20urandom
-rwxrwxrwx    1 root     root        2.0K Jul 24  2015 S30dbus
-rwxrwxrwx    1 root     root         340 Jul 24  2015 S40network
-rwxrwxrwx    1 root     root         405 Jul 24  2015 S50app
-rwxrwxrwx    1 root     root        1.6K Jul 24  2015 S90demo
-rwxrwxrwx    1 root     root         776 Jul 24  2015 rcS
```

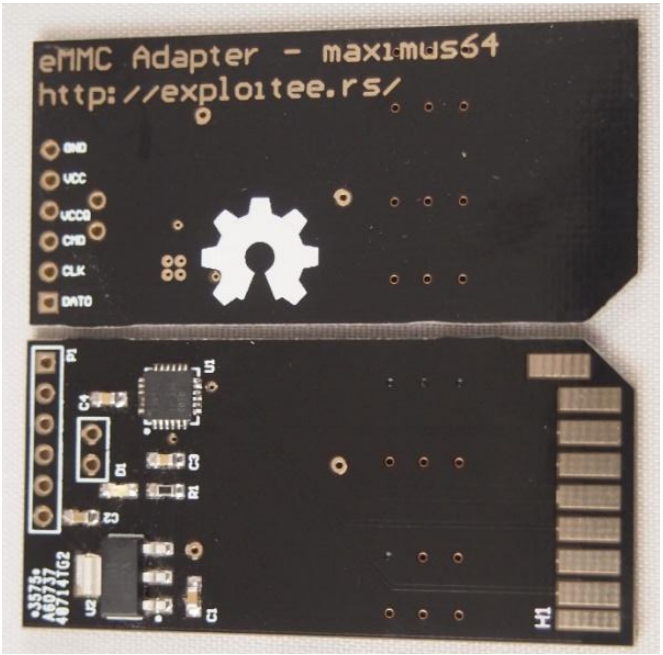
?master ~/4tools/firmwalker > ./firmwalker.sh ~/2dev/5learn/XXX
Search for password files
passwd
1/squashfs-root/etc/passwd
1/squashfs-root/usr/bin/passwd
shadow
1/squashfs-root/etc/shadow
*.psk
Search for Unix-MD5 hashes
/Users/tony/2dev/5learn/XXX/squashfs-root/etc/shadow:1mtjRWsdG$JOSdnKQhULmqnVajxi7LQ0
Search for SSL related files
*.pem
1/squashfs-root/etc/zxv10.pem
Search for configuration files
*.conf
1/squashfs-root/etc/ath/topology_ap.conf
1/squashfs-root/etc/ath/topology_sta.conf
1/squashfs-root/etc/ath/wpa-ap.conf
1/squashfs-root/etc/ath/wpa-sta.conf
1/squashfs-root/etc/inetd.conf
Search for ip addresses
ip addresses
0.0.0.0
127.0.0.1

LG Smart Refrigerator

(LFX31995ST)

1 - UART drops into root shell

2 - eMMC tapping



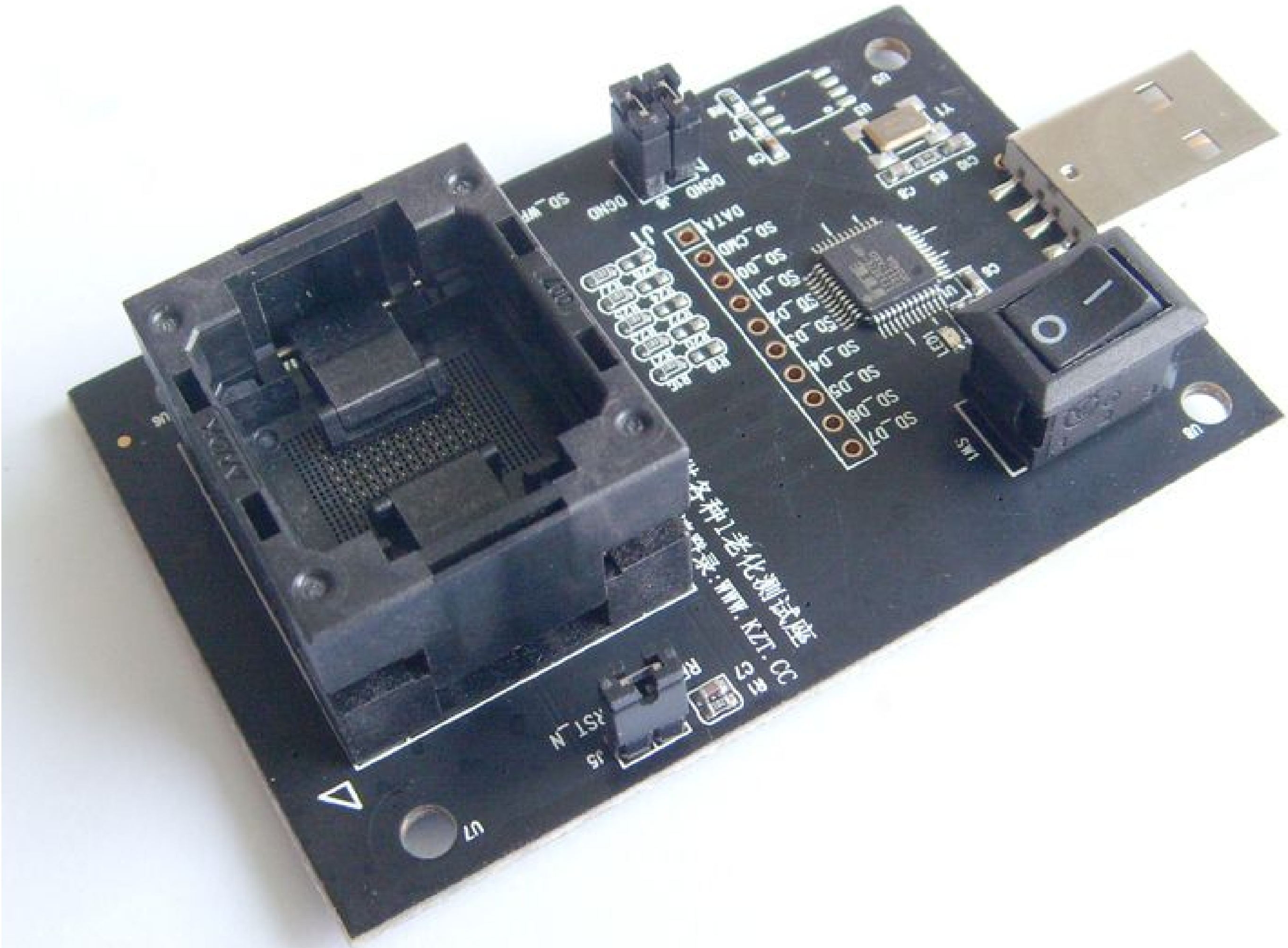
https://www.exploitee.rs/index.php/LG_Smart_Refrigerator_(LFX31995ST)

eMMC tapping

https://www.exploitee.rs/index.php/LG_Smart_Refrigerator_(LFX31995ST)

Other possibilities:

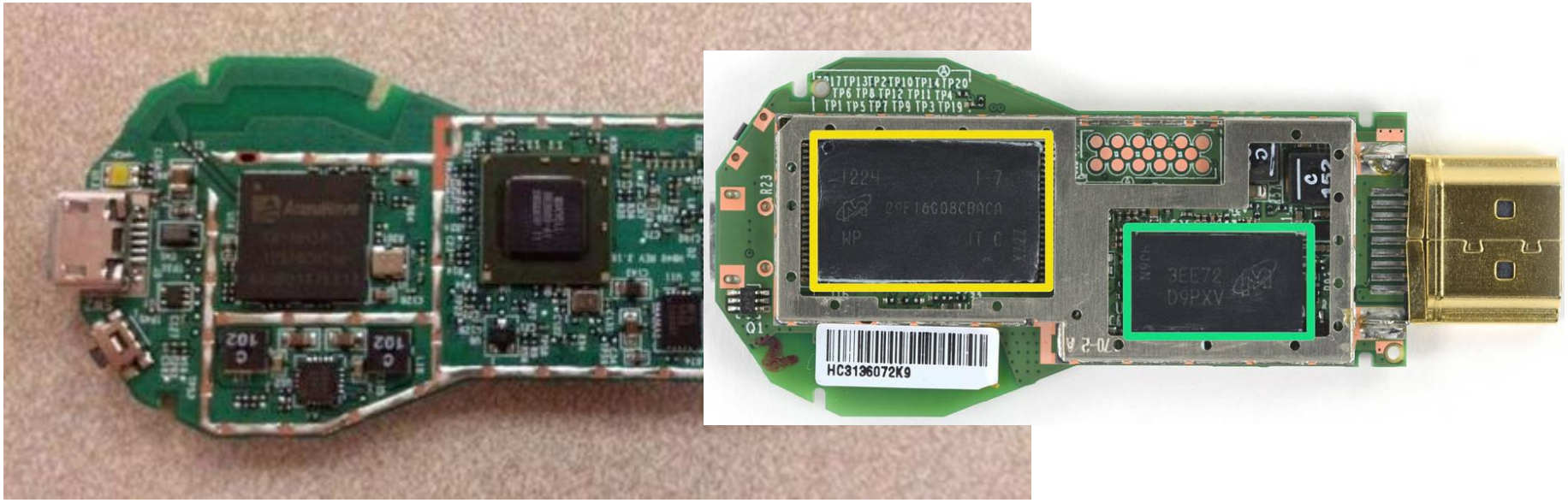
Unsolder the chip and throw it in a USB MS reader



Other possibilities:

Internal USB drive

Micro USB ports that are actually USB-OTG



Internal USB networks (think Moto RAZR / DROID)

Meh, hardware hacking, is there anything i can do from the script kiddie side?

Remember OWASP IoT Top 10? The most common vuln is? Insecure web interface…

- Guess what's the most common OWASP Top10 vuln? Command injection

The idea is simple - use the idea behind fuzzing:

- Crawl the webif
- Use a web vuln scanner
- Leave it overnight
- Profit!

You can find all kinds of interesting stuff…

Like remote RCE for some devices

http://<IP>/stainfo.cgi?ifname=ath0&sta_mac=00:11:22:33:44:55|<URLENCCMD>&mode=ap

Meh, why should i care?

- The smart device is just the implant you need in a network!
- Ideal pivot point
- Remember how hacking team got hacked?

A 0day in an embedded device seemed like the easiest option, and after two weeks of work reverse engineering, I got a remote root exploit. Since the vulnerabilities still haven't been patched, I won't give more details […]

- Phineas Fisher

When you start some lateral thinking…

Pretty much when you have LAN access, you have better chances of 0wning that network. (More on that if that's a windows domain)

- Responder
- Bettercap
- Backdoor factory
- BeeF framework

Write once, deploy your malware everywhere.

Also, code reuse...

So?

The IoT fad is coming, and from a security/privacy standpoint, we are not ready.

Well, own some of the stuff you own! You will be amazed what runs GNU/Linux ;)

Before buying, inform yourself if the vendor did some of the mortal security sins…

So, what’s the moral of this and open source?

If you develop FLOSS embedded things, take note:

“open by default” is broken, build security into the stuff you develop. Embedded people should get more security knowledge.

If some vendor wants to lock your device. There is always a way to free it. Just think… :)

Questions?

tonimir.kisasondi@foi.hr

@kisasondi

Thank you!