Basic Firmware Extraction

Defcon HHV 2016 edition

Matt DuHarte

@Crypto_Monkey

https://github.com/CryptoMonkey

DISCLAIMER

All Standard disclaimers apply. This talk does not represent the views of my employers, associates, colleagues, or my dog. The information is provided to foster discussion of only the most ethical analysis of all the wide open firmware that surrounds us everyday.

Agenda

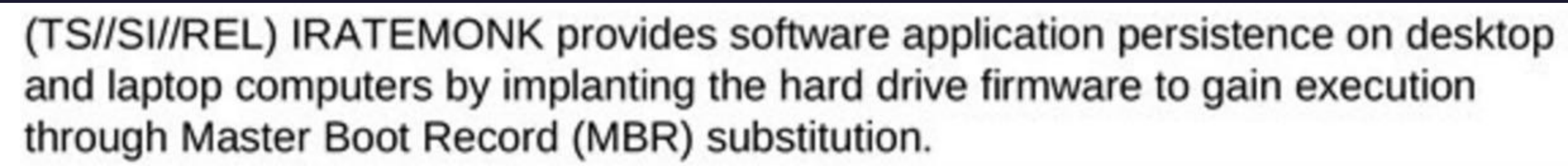
- Motivation
- Attack Pattern
- Extraction Protocols and Technology
- Firmware Analysis

Why This? Why Now?

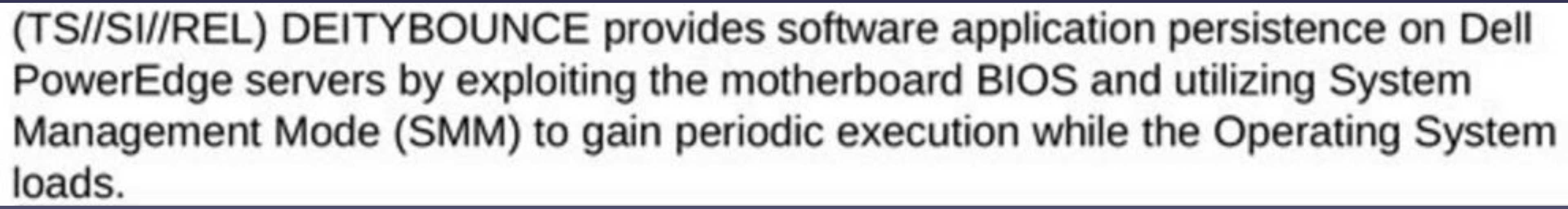
- Hardware cost and availability
  - Low cost, reliable hardware adapters readily available
  - Software for firmware analysis and image carving freely available
  - Increased demand for skills due to the proliferation of new technologies
- Not every small device has an exposed UART

NSA Programs In The News

- Several technologies revealed in the NSA ANT catalog appear to be persistent hardware trojans
- Two from https://nsa.gov1.info/dni/nsa-ant-catalog/



(TS//SI//REL) IRATEMONK provides software application persistence on desktop and laptop computers by implanting the hard drive firmware to gain execution through Master Boot Record (MBR) substitution.



(TS//SI//REL) DEITYBOUNCE provides software application persistence on Dell PowerEdge servers by exploiting the motherboard BIOS and utilizing System Management Mode (SMM) to gain periodic execution while the Operating System loads.

Trust Nobody

NSA spying may seem a distant problem to many of you but

- Shadow IT and BYOD
  - Who is auditing these systems?
  - Are they being audited?
  - Who could profit from the data passing through them?
- Who may be stealing your code?
  - American Superconductor vs. Sinoval Wind Group
    - http://fortune.com/2015/12/12/cybersecruity-amsc-cyber-espionage/
    - http://www.nbcnews.com/news/other/chinese-firm-paid-insider-kill-my-company-american-ceo-says-f6C10858966

Auditing Small Embedded Devices

- Stay on the right side of the law (IANAL) and ethics
  - Someday will expert witnesses be called on to
    - Prove that the software running a device has not been tampered with?
    - Prove that a company has done due diligence by updating infrastructure devices?
- Will your cyber-insurance policy cover you if your POS devices have not been updated?
- Does the system have any undisclosed access methods (backdoors or hard coded passwords) that threaten your security?

Technology Making Audits Simpler

- In the past many of the chips found in these devices could only be studied with expensive hardware sniffers
  - The target audience for those devices was engineers and designers not auditors
  - Our needs are much narrower and there are now low cost devices that meet the needs to access firmware easily
    - The first gen devices were very slow and unreliable

Pre-engagement Interactions

- What equipment you need is dictated by what architecture the manufacturer used for storing the firmware
- You customer should be able to tell you this for internal audits
- OSINT is a great source for information about unknown technologies

How much do you want to spend?

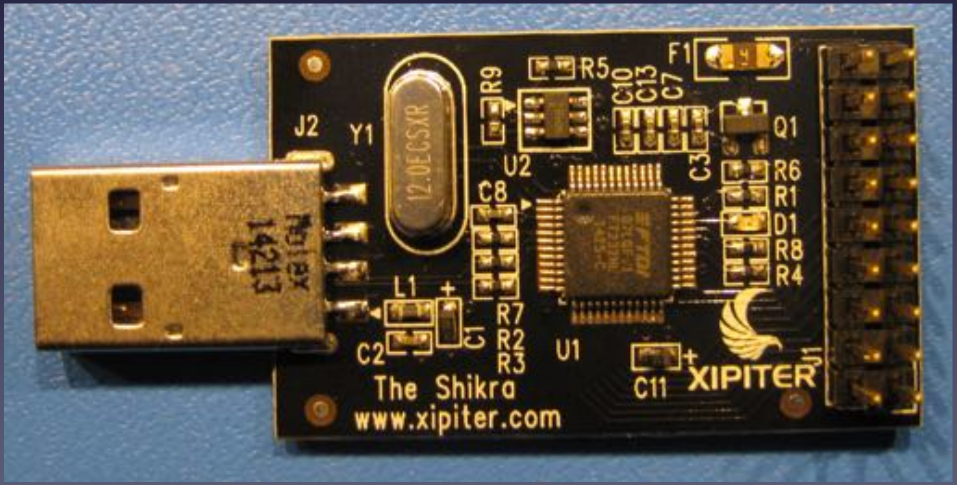
- Are you doing this for business purposes, as research or for fun?
- What technologies are you looking to study
  - What kinds of systems do your customers need you to analyze?
  - How tolerant are you or your employers to physical destruction of the device?

Building a Lab

- My slice of insanity

What You Really Need To Get Started

- A screwdriver
- A few jumper wires



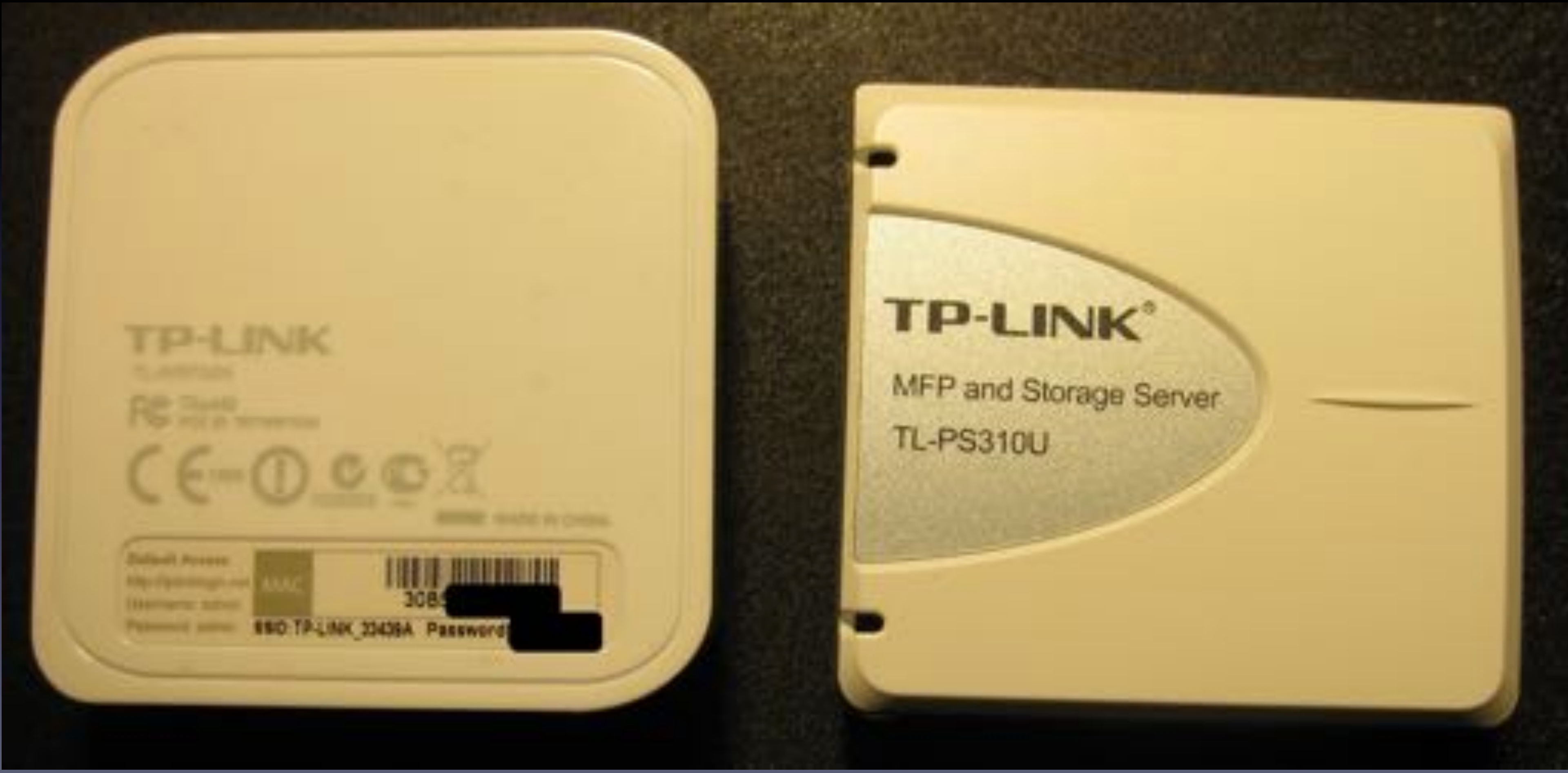
The Most Valuable Upgrade

- Of all the things we discuss, a good trinocular microscope will save your sanity
- It also makes your reports much more professional looking

Intelligence Gathering

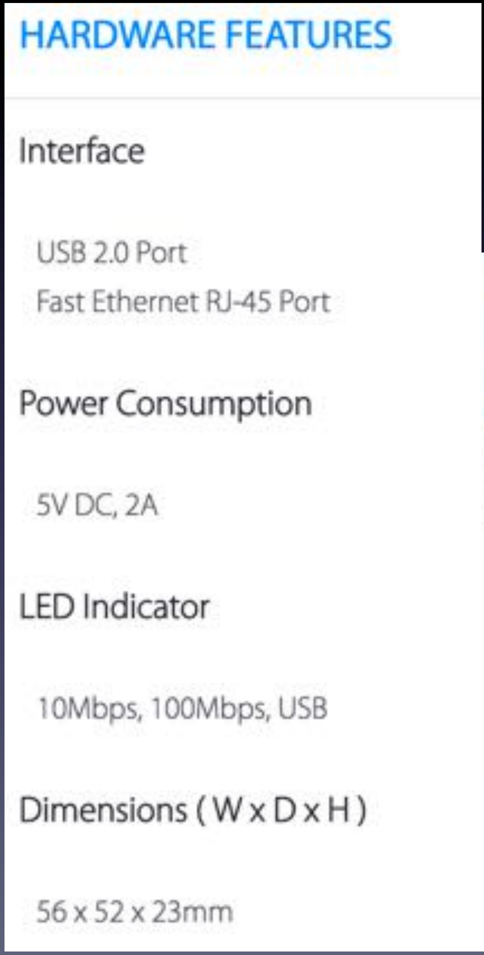
- Manufacturers will often use the same parts throughout their product line
- There can be odd parts that make it harder to extract firmware

Explored Devices



TP-Link TL-PS310U

- Single USB2.0 Port MFP and Storage Server
- http://www.tp-link.com/en/products/details/TL-PS310U.html



HARDWARE FEATURES

Interface

USB 2.0 Port
Fast Ethernet RJ-45 Port

Power Consumption

5V DC, 2A

LED Indicator

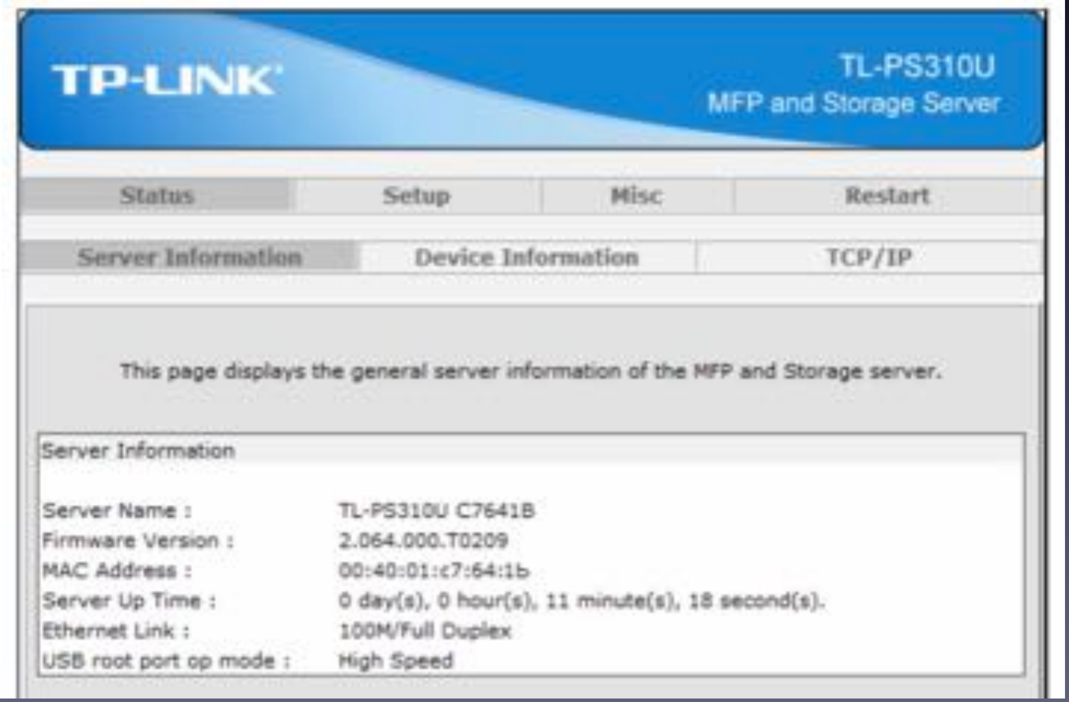
10Mbps, 100Mbps, USB

Dimensions (W x D x H)

56 x 52 x 23mm

According to the manual it has a GUI

http://www.tp-link.com/resources/document/TL-PS310U_V2_User_Guide_1910010947.PDF



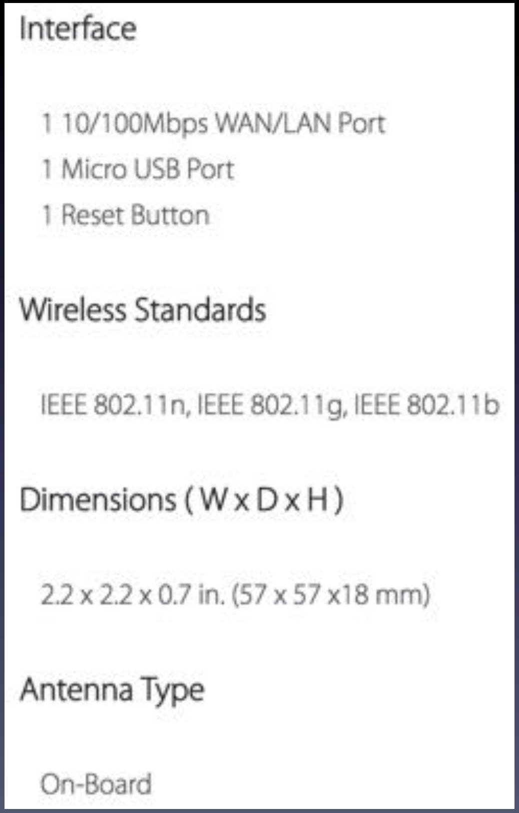
This page displays the general server information of the MFP and Storage server.

Server Information

Server Name :	TL-PS310U C7641B
Firmware Version :	2.064.000.T0209
MAC Address :	00:40:01:c7:64:1b
Server Up Time :	0 day(s), 0 hour(s), 11 minute(s), 18 second(s).
Ethernet Link :	100M/Full Duplex
USB root port op mode :	High Speed

TP-Link TL-WR702N

- 150Mbps Wireless N Nano Router
- http://www.tp-link.com/en/products/details/TL-WR702N.html



Interface

1 10/100Mbps WAN/LAN Port

1 Micro USB Port

1 Reset Button

Wireless Standards

IEEE 802.11n, IEEE 802.11g, IEEE 802.11b

Dimensions (W x D x H)

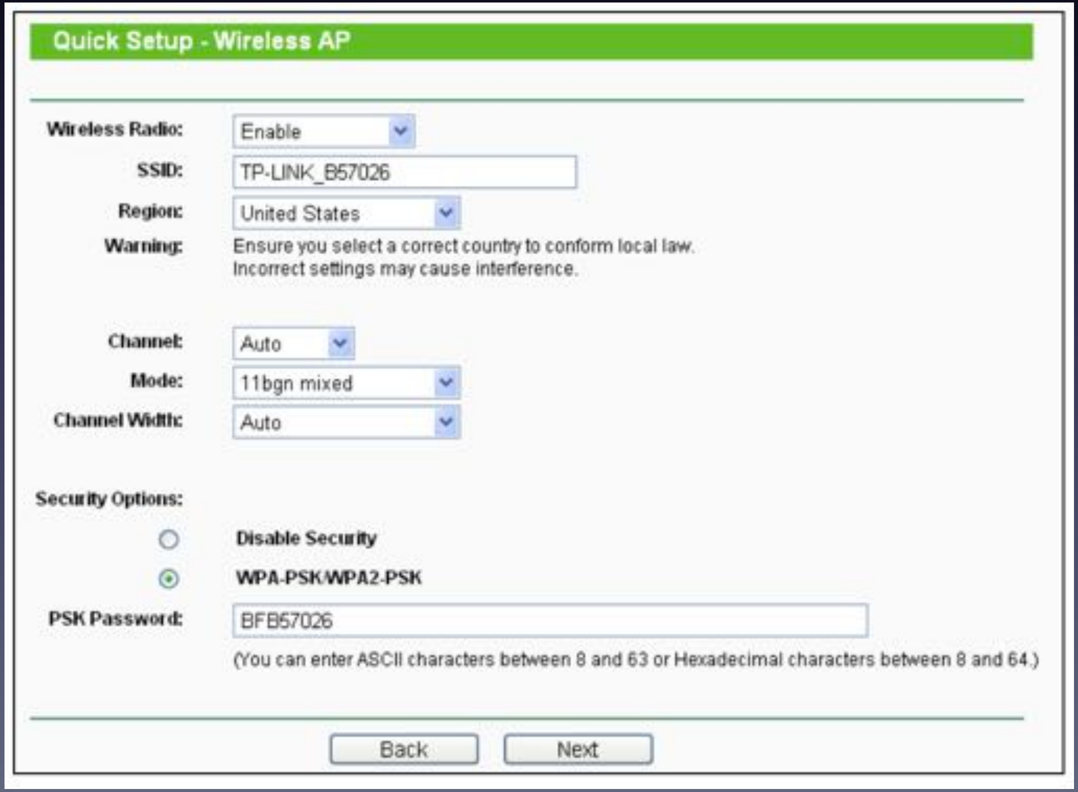
2.2 x 2.2 x 0.7 in. (57 x 57 x18 mm)

Antenna Type

On-Board

According to the manual it has a GUI

http://www.tp-link.com/res/down/doc/TL-WR702N_V1_UG.pdf



Quick Setup - Wireless AP

Wireless Radio:	Enable
SSID:	TP-LINK_B57026
Region:	United States
Warning:	Ensure you select a correct country to conform local law. Incorrect settings may cause interference.
Channel:	Auto
Mode:	11bgn mixed
Channel Width:	Auto

Security Options:

○ Disable Security

● WPA-PSK/WPA2-PSK

PSK Password: BFB57026

(You can enter ASCII characters between 8 and 63 or Hexadecimal characters between 8 and 64.)

Threat Modeling

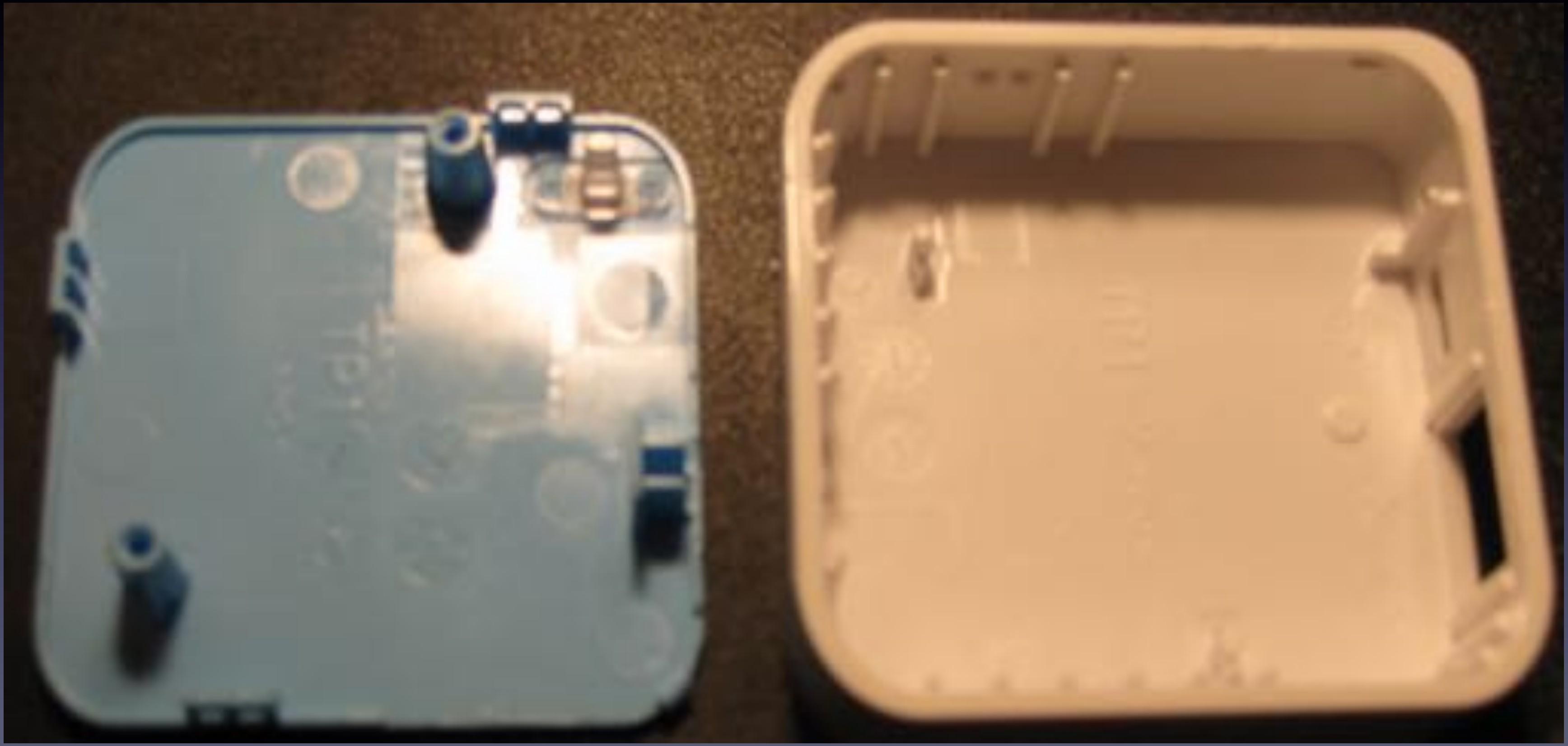
- What are your motivations for attacking the product?
- How do you think the manufacturer will react?
  - Did they hire you or did a competitor hire you?
    - Where does competitive analysis end?
- What is the public good of your research?
- Do you have a good lawyer?
  - Do not count on the DMCA security analysis exemption to protect you from your poor judgment

Vulnerability Analysis – Physical

- How hard is it to open the case?
  - Did the manufacturer use security screws
  - Did you have to use a special tool to open the case

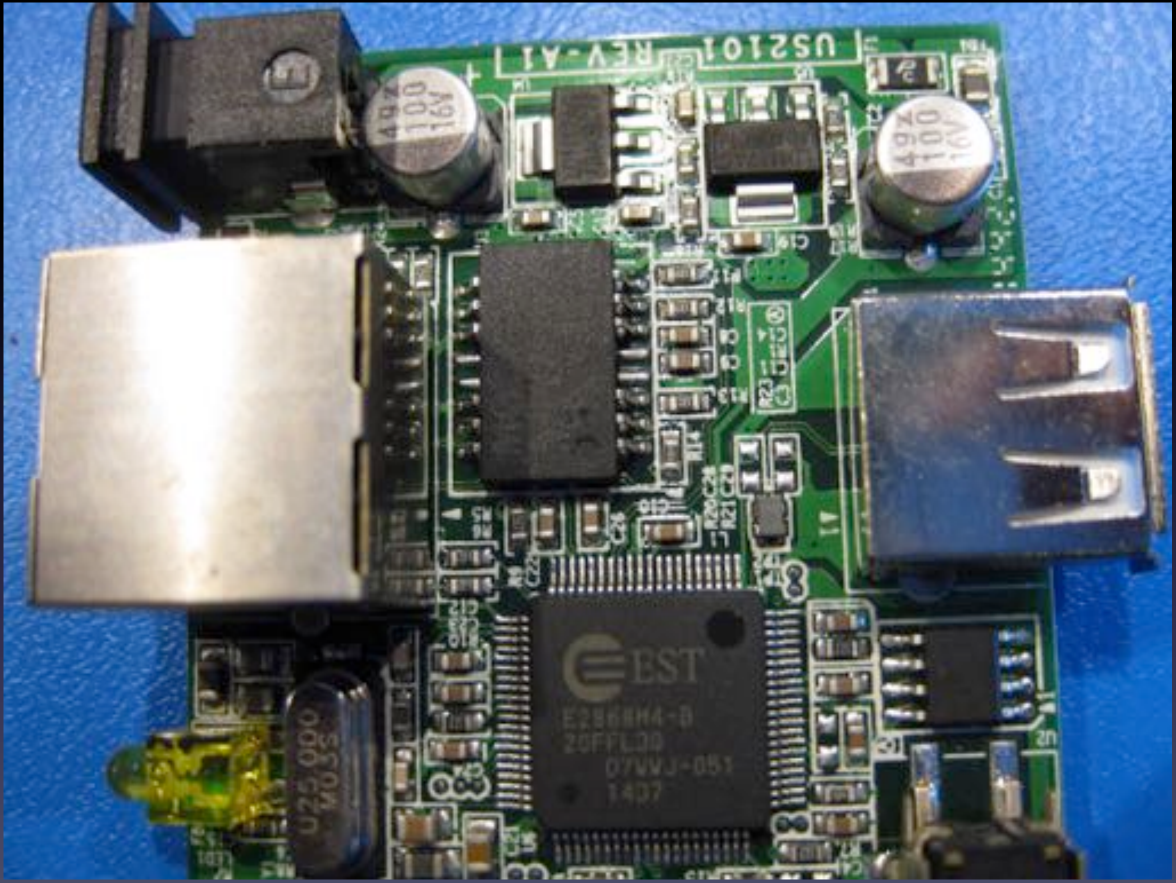
Opening the case

- Both devices used plastic latches
  - A monkey and a screwdriver pried the case open

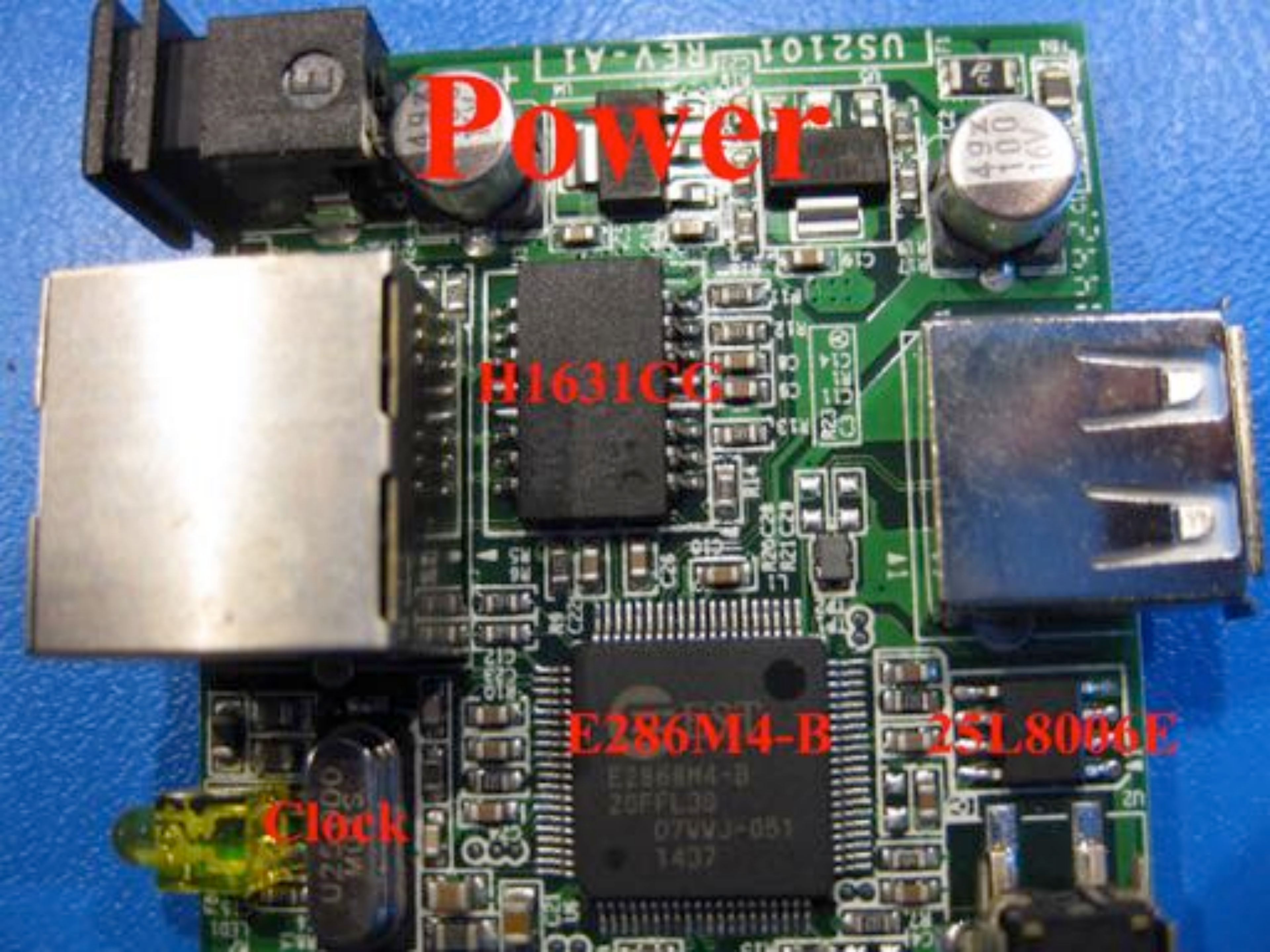


TP-Link TL-PS310U

- Inspect the board
- We found chips
- We want firmware…
- Which one has firmware?



The other side has no chips


Power
H1631CG
E286M4-B
25L8006E
Clock
US2101 REV-A1

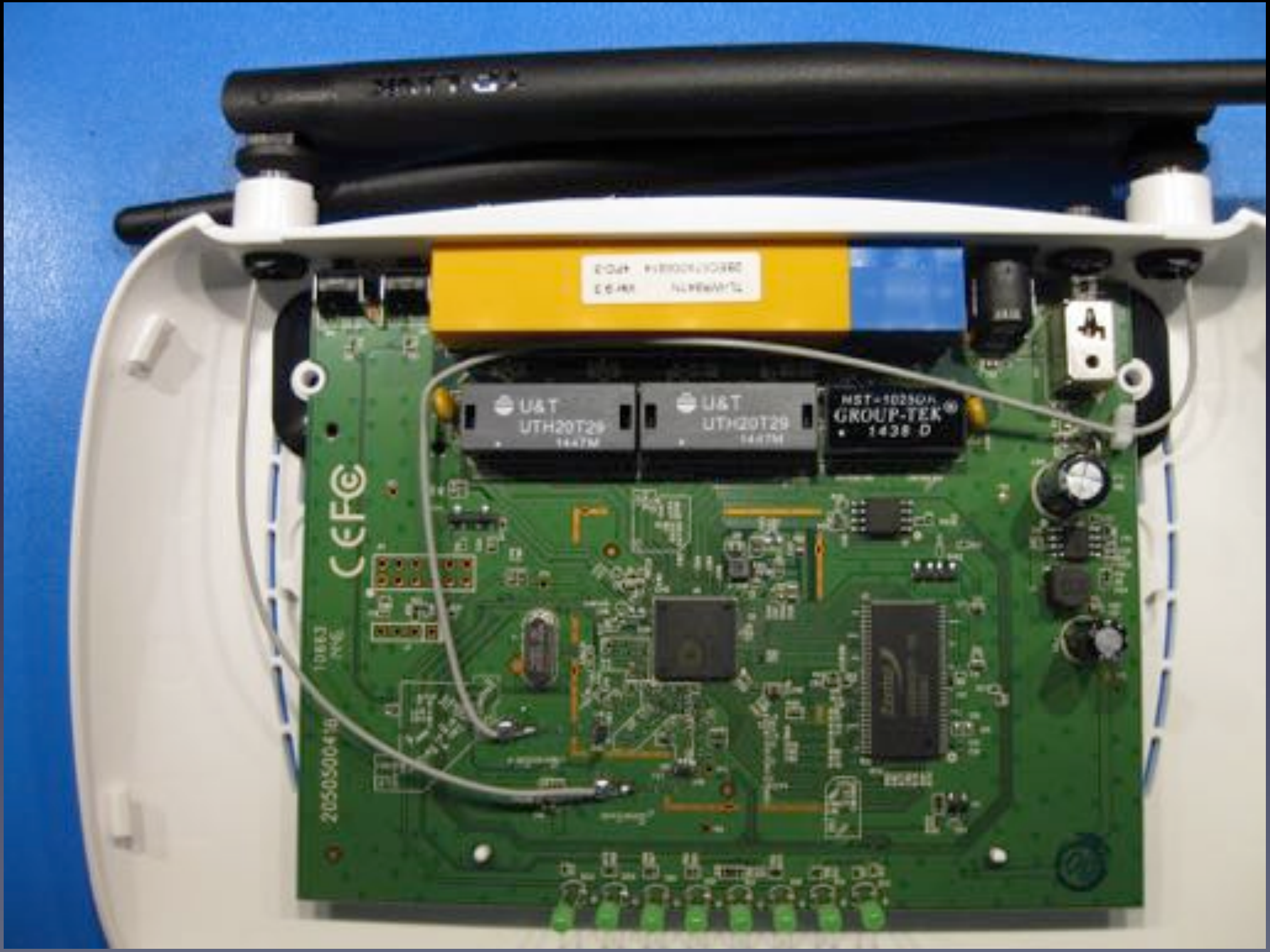
Google Is Our Friend

Designation	Purpose
MNC H1631CG	10/100 Base-T Surface Mount Magnetics
EST E2868M4-B	SoC with built in USB2 and 802.3/3u
MXIC 25L8006e	8M-BIT [x 1 / x 2] CMOS SERIAL FLASH

- **MNC H163i1CG**
  - http://www.mnc-tek.com/Private/ProductFiles/635623692418750000247995066.pdf
- EST E2868M4-B
  - http://jhongtech.com/DOWN/Ds-28xx-12-E.pdf
- MXIC 25L8006e
  - http://www.zlgmcu.com/mxic/pdf/NOR_Flash_c/MX25L1606-8006E_DS_EN.pdf

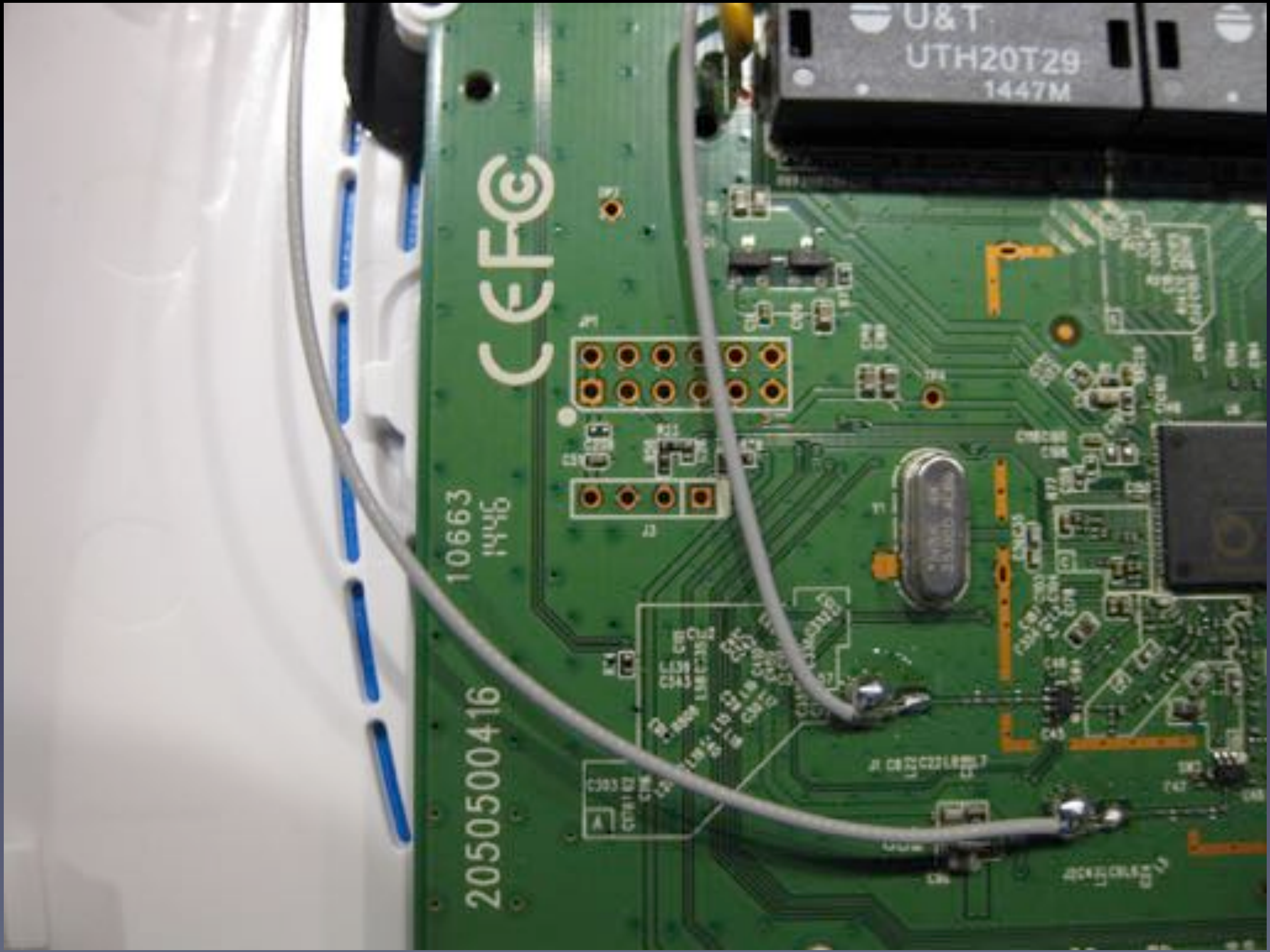
An Aside: What is missing?

- The insides of the TP-Link TL-WR841N 300Mbps Wireless N Router



An Aside: What is missing?

- The insides of the TP-Link TL-WR841N 300Mbps Wireless N Router



Easy access to serial ports

When you see rows of pins on the board like this and figure out what they are they make connecting much easier

An Aside: What is missing?

- Without a serial port it becomes harder to look at the running software on the system
  - We have a harder time doing dynamic analysis and exploring the OS from the inside
- We may loose access to data or exploits that live only in memory

An Aside: What is NOT missing?

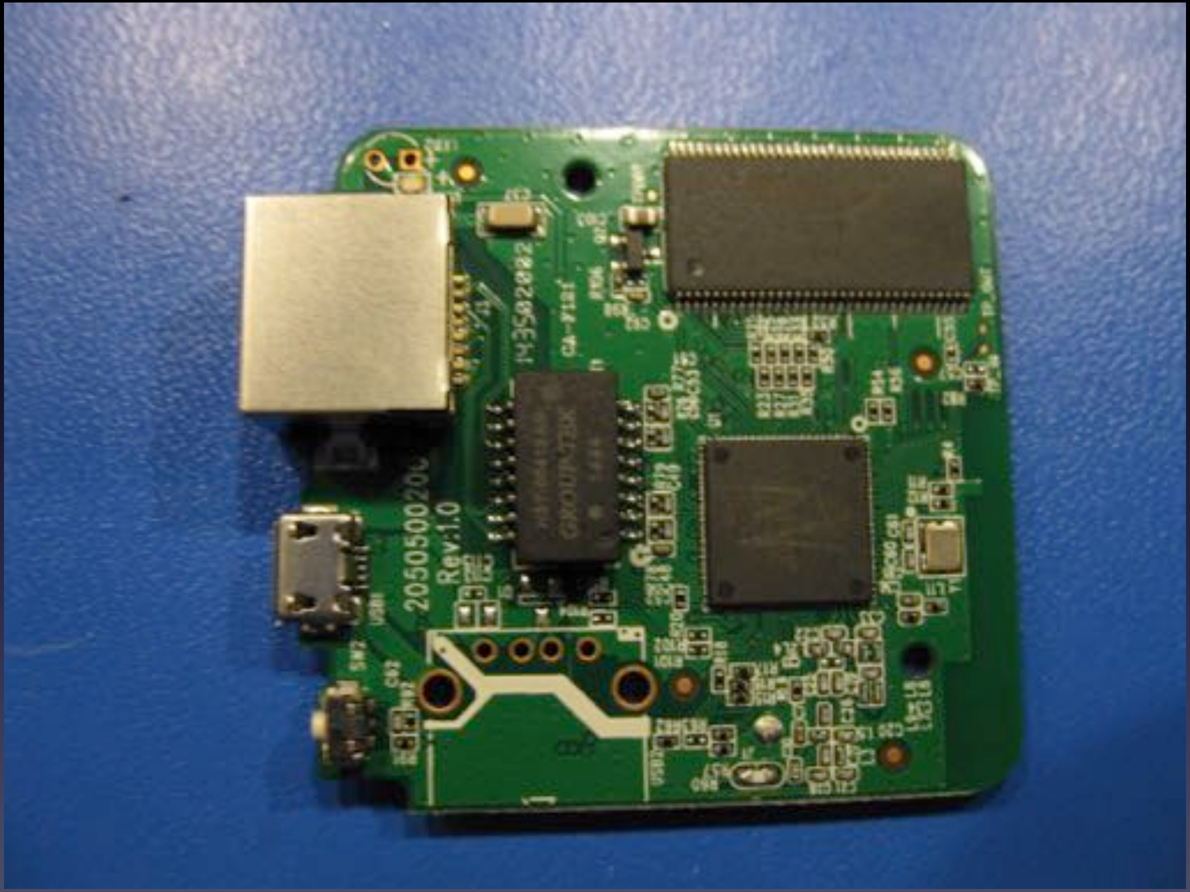
- As we have learned with years of studying rootkits

Running systems lie

- Every piece of storage is not always mounted on the system
- We want to audit the entire system not just a piece of it

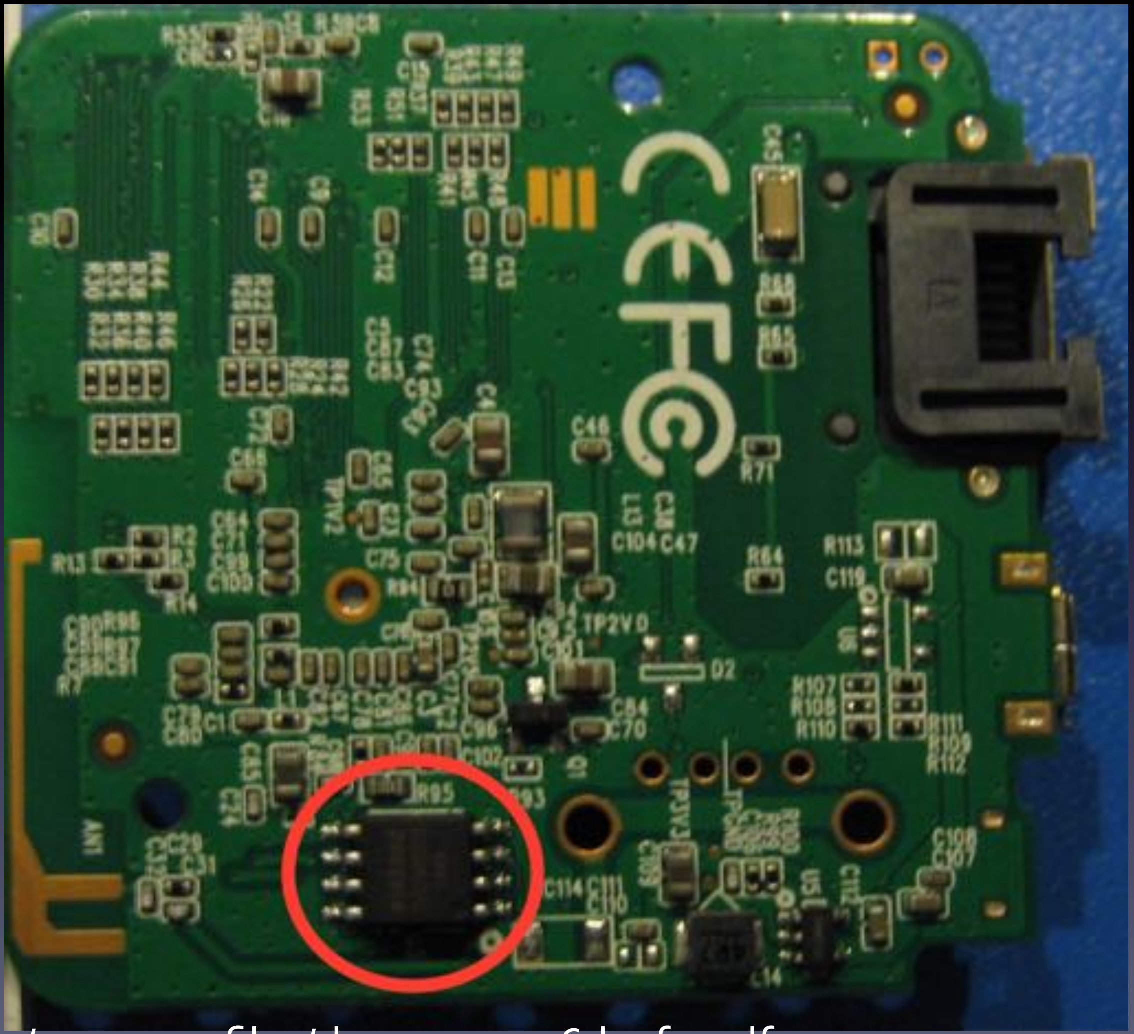
TP-Link TL-702N

- A3S56D40GTP
  - Memory module
  - SDRAM
- AR9331 SOC
  - With 802.11n Wireless
- The Ethernet magnetics
- No non-volatile storage!



TP-Link TL-702N

- It is on the other side of the board
- Winbond 25Q16DVS1G
- Just like the last one it is an 8 pin serial flash



https://www.winbond.com/resource-files/da00-w25q16dv_f2.pdf

Exploitation (Pull and Extract The Firmware)

- In all three cases we found a serial flash device
  - I'm including the the TP-Link TL-WR841N
- These chips store the firmware that is running the SoC and represent our target for extraction

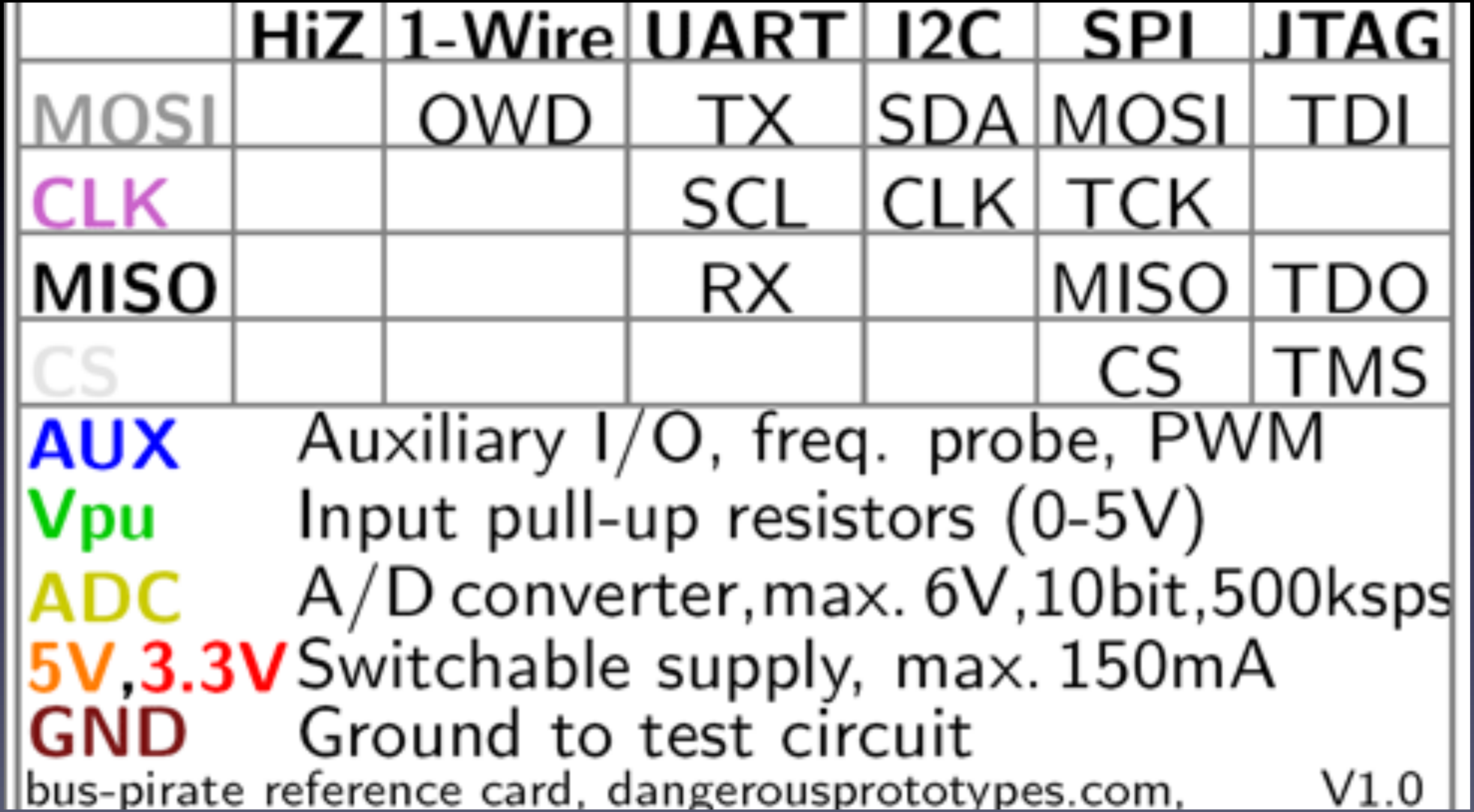
The Big Three Protocols

- Three major protocols are encountered for firmware uploads and downloads
  - –I2C
  - –JTAG
  - –SPI

Only a little protocol knowledge is needed

- While each of these protocols has extensive details we do not actually need much info to use them
- What voltage do they run on?
- What signals do they use?
- How to connect to those signals to access the chip data?

Tool – The Bus Pirate



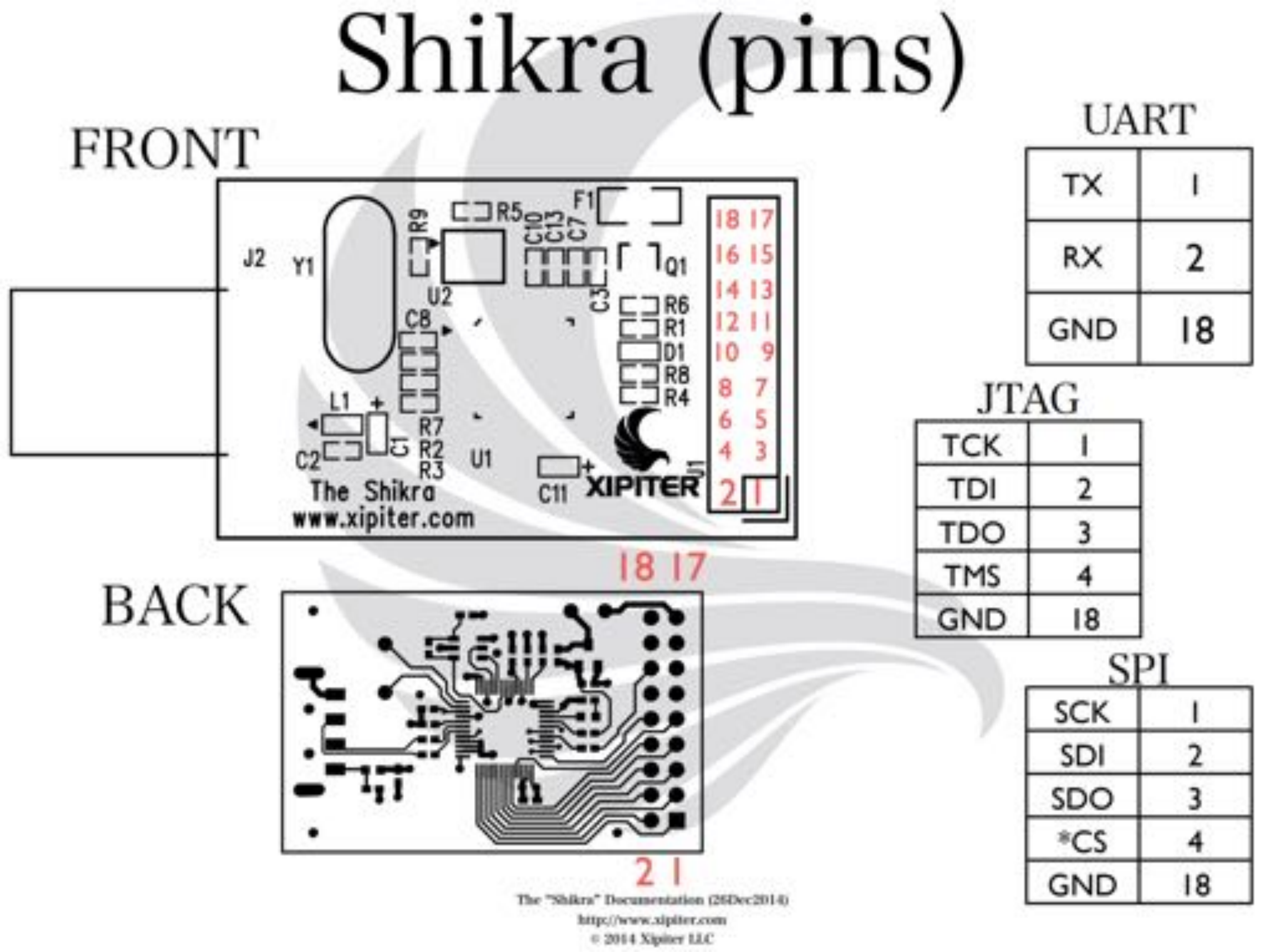
	HiZ	1-Wire	UART	I2C	SPI	JTAG
MOSI		OWD	TX	SDA	MOSI	TDI
CLK			SCL	CLK	TCK	
MISO			RX		MISO	TDO
CS					CS	TMS

AUX	Auxiliary I/O, freq. probe, PWM
Vpu	Input pull-up resistors (0-5V)
ADC	A/D converter,max. 6V,10bit,500ksps
5V,3.3V	Switchable supply, max. 150mA
GND	Ground to test circuit

bus-pirate reference card, dangerousprototypes.com, V1.0

http://dangerousprototypes.com/docs/images/o/ob/Buspirate_refcard.png

Tool – Shikra



Shikra (pins)

FRONT

BACK

UART

TX	1
RX	2
GND	18

JTAG

TCK	1
TDI	2
TDO	3
TMS	4
GND	18

SPI

SCK	1
SDI	2
SDO	3
*CS	4
GND	18

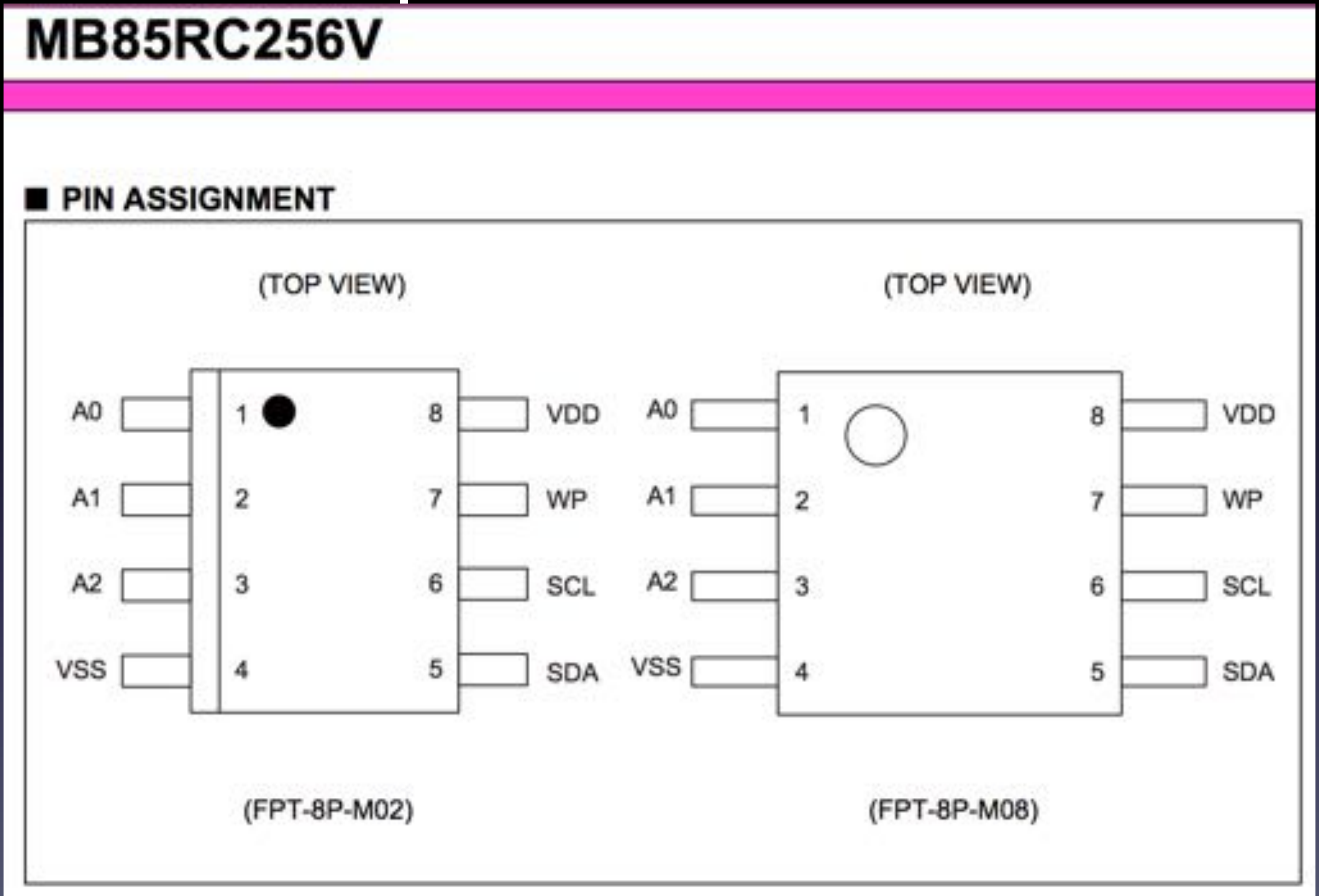
The "Shikra" Documentation (26Dec2014)
http://www.xipiter.com
© 2014 Xipiter LLC

- http://www.xipiter.com/uploads/2/4/4/8/24485815/shikra_documentation.pdf

Inter-Integrated Circuit (I2C)

- Requires 2 wires
  - Serial Data Line (SDA)
  - Serial Clock Line (SCL)
- Generally runs on 3.3V or 5V
  - The bus pirate can handle both
- Each device on the I2C bus has an address

Example I2C Non-Volatile



MB85RC256V

■ PIN ASSIGNMENT

https://cdn-shop.adafruit.com/product-files/1895/MB85RC256V-DS501-00017-3v0-E.pdf

This becomes an exercise in matching the signals

- I2C is used for low speed applications
- It is not the fastest protocol , though it does have a 3.4 Mbit/s High Speed mode
    - Bus Pirate maxes out a 1Mbit/s

Joint Test Action Group (JTAG)

- JTAG is the most complicated of the three protocols
- Up to 5 signals are used
  - TDI (Test Data In)
  - TDO (Test Data Out)
  - TCK (Test Clock)
  - TMS (Test Mode Select)
  - TRST (Test Reset)

JTAG

- The Bus Pirate supports 3 signal variants
- The Shikra supports 4 signal variants
- JTAG is more of an additional set of circuits that are built into a chip to add debugging than a communication protocol

JTAG

- Because of the added cost it is normally used when you want to support both loading firmware and debugging
- There can be a variety of devices on the JTAG network
  - They are all in a sort of pass-through mode except one in the chain

Serial Peripheral Interface (SPI)

- Four signals are used in SPI
  - SCLK - Serial Clock
  - MOSI - Master Output, Slave Input
  - MISO - Master Input, Slave Output
  - SS - Slave Select

SPI

- Both the Shikra and the Bus Pirate support SPI
- It is faster than I2C and requires less power
  - But more wires
- All the TP-Link Devices we have encountered use SPI for their firmware storage

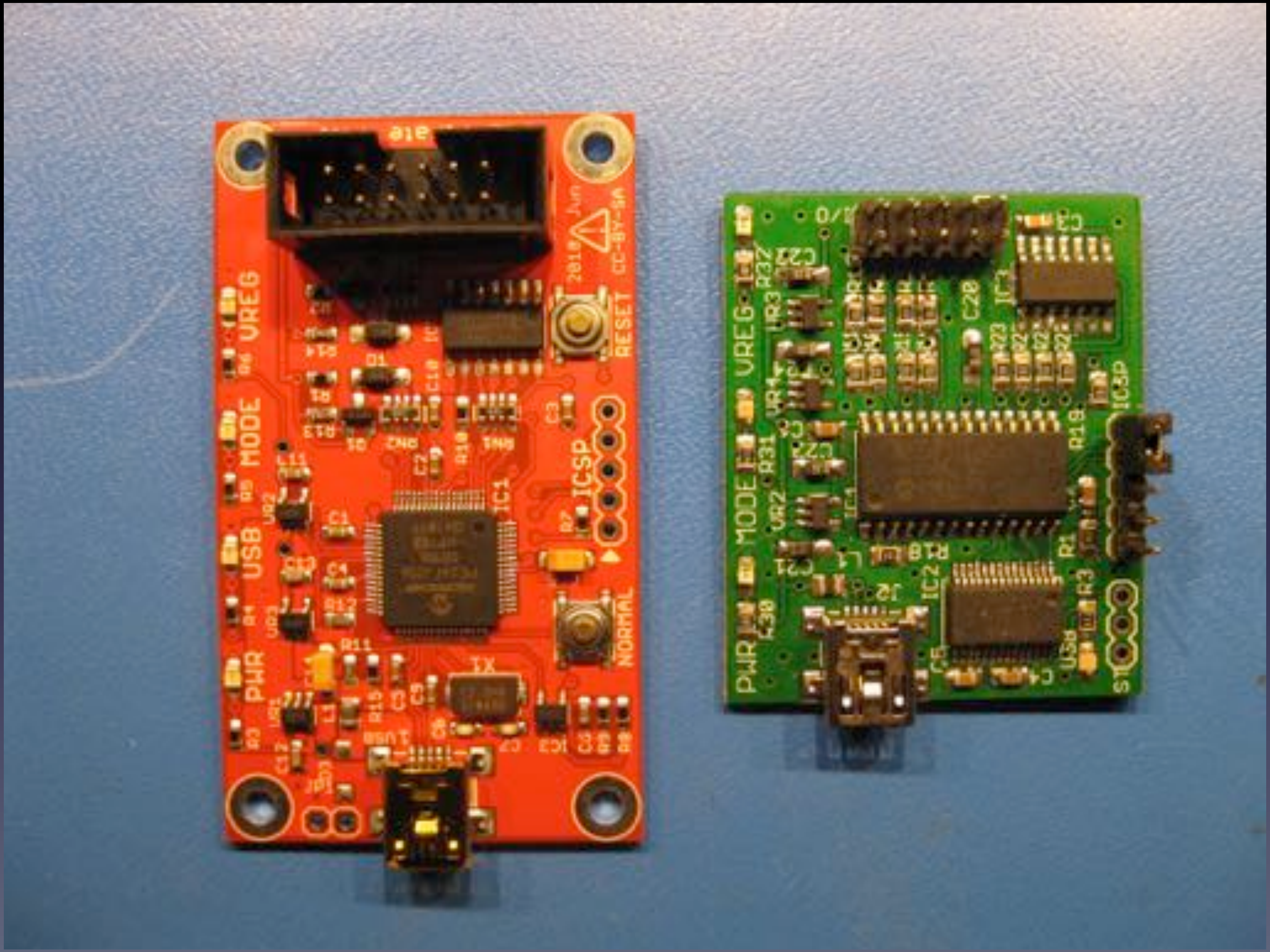
So, that is protocols

- That’s it, if you want lots of detail check out the Wikipedia entry for each protocol
- We don’t need any data besides
  - How many wires
  - Which wire is which

Hardware Adapters

- We have been discussing two hardware adapters that allow a computer to speak directly to these chips
  - The Bus Pirate
  - The Shikra

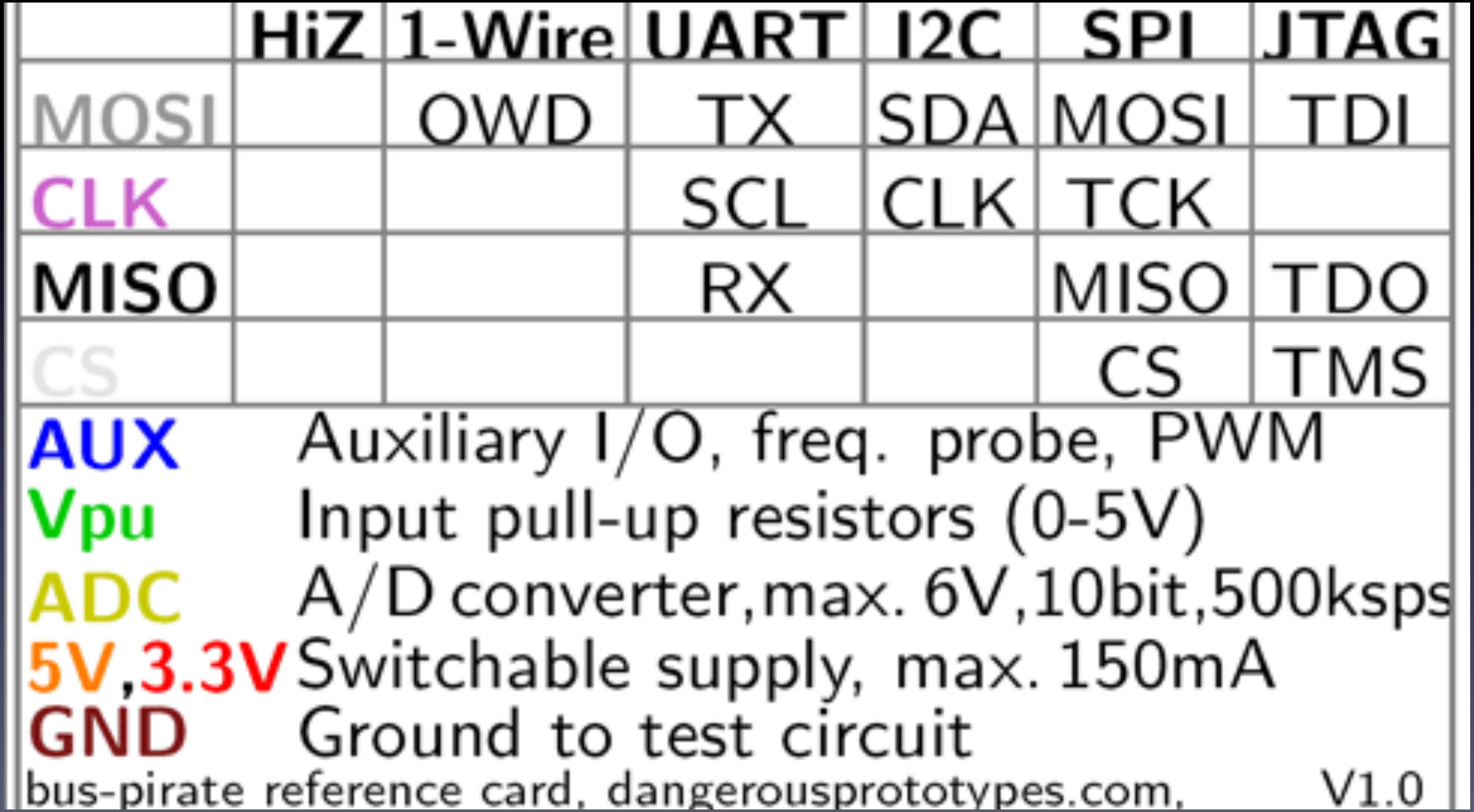
The Bus Pirate



The Bus Pirate

- Multiple generations of designs have created one of the most useful little adapters available
- Provides a Serial Port (via USB) interface that allows access to various protocols
- You can use it to send individual commands to the chip you connect it to

Tool – The Bus Pirate



	HiZ	1-Wire	UART	I2C	SPI	JTAG
MOSI		OWD	TX	SDA	MOSI	TDI
CLK			SCL	CLK	TCK	
MISO			RX		MISO	TDO
CS					CS	TMS

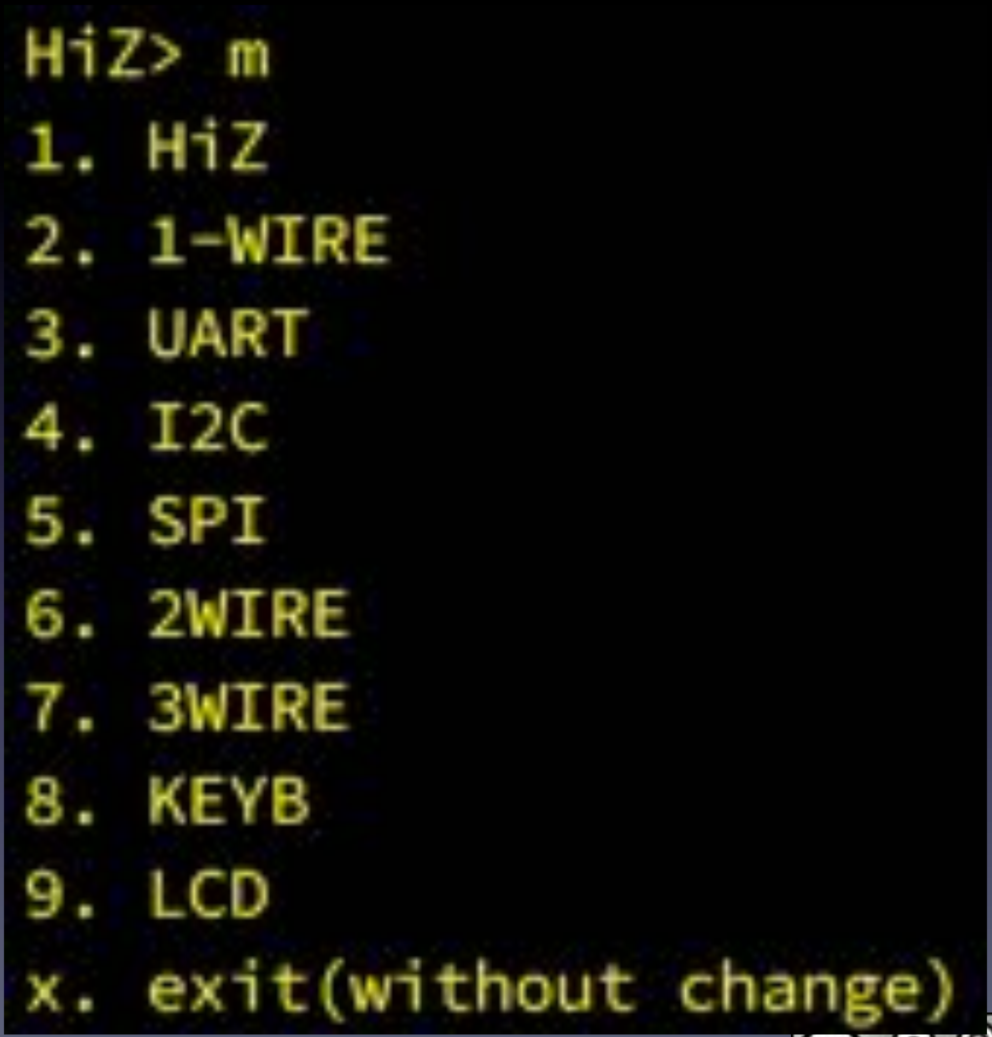
AUX	Auxiliary I/O, freq. probe, PWM
Vpu	Input pull-up resistors (0-5V)
ADC	A/D converter,max. 6V,10bit,500ksps
5V,3.3V	Switchable supply, max. 150mA
GND	Ground to test circuit

bus-pirate reference card, dangerousprototypes.com, V1.0

http://dangerousprototypes.com/docs/images/o/ob/Buspirate_refcard.png

The Bus Pirate

- Plug it in and connect to it as a serial port device
  - Say using minicom or screen
  - You interact with it as a command line device sending individual commands in the protocol selected

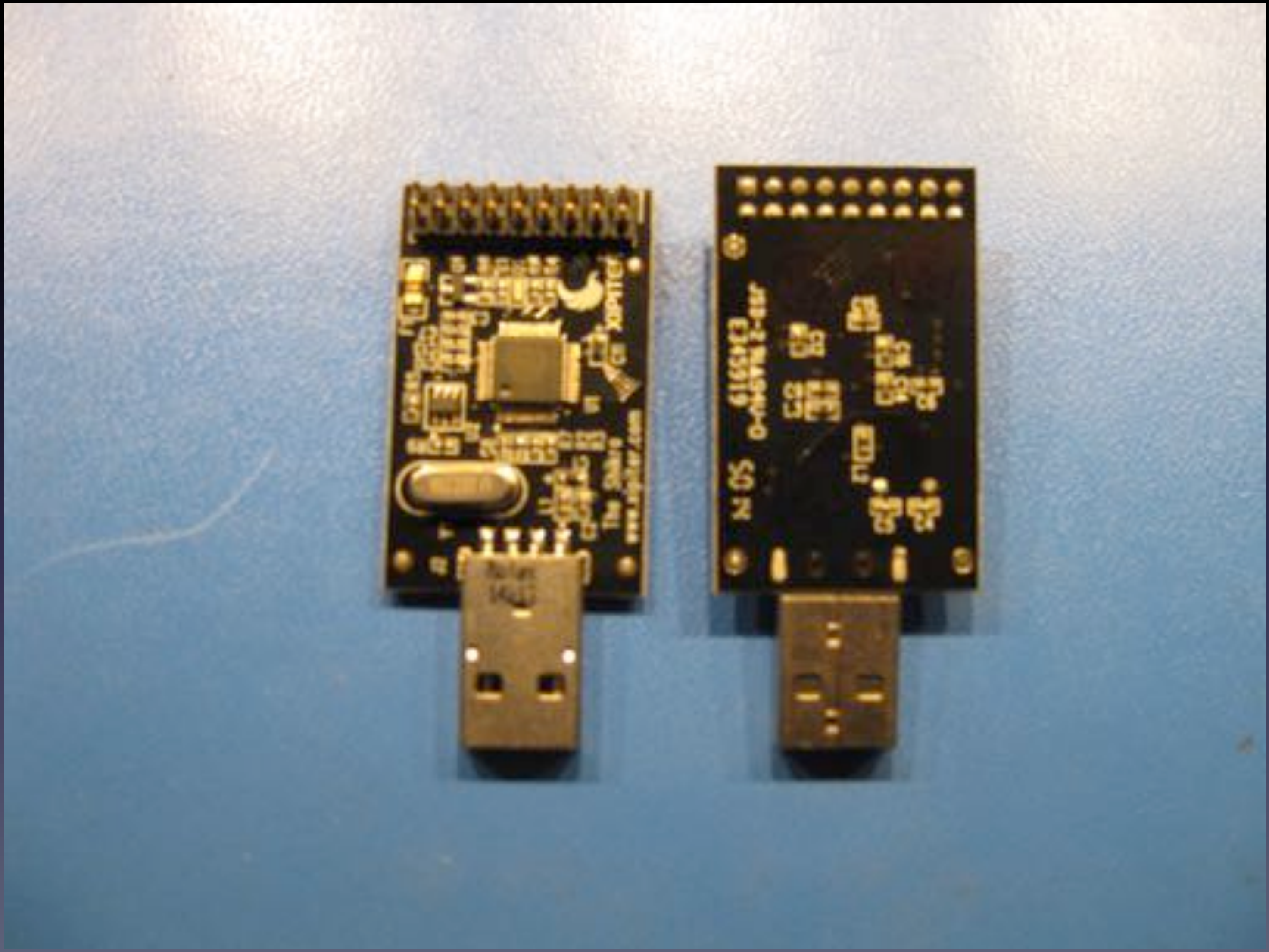


```
HiZ> m
1. HiZ
2. 1-WIRE
3. UART
4. I2C
5. SPI
6. 2WIRE
7. 3WIRE
8. KEYB
9. LCD
x. exit(without change)
```

Bus Pirate Availability

- Multiple generations of the Dangerous Prototypes design are available for about $40 USD at places like
  - Adafruit
  - Seedstudio
  - Sparkfun

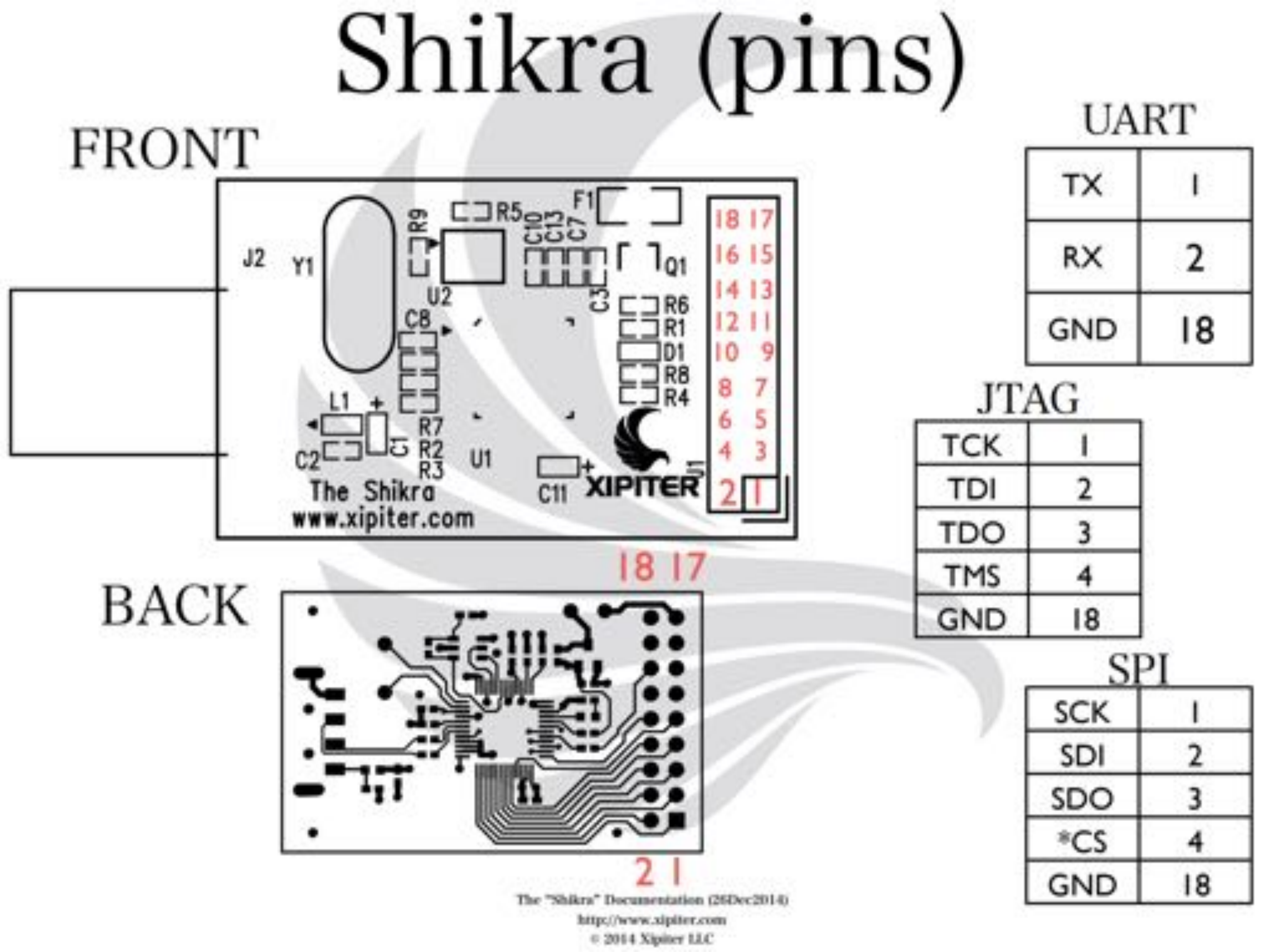
The Shikra



The Shikra

- Doesn't handle the number of protocols as the Bus Pirate
- A lot faster
- Newer chipset is very stable
- Not as well documented or interactive as the Bus Pirate
  - More of an interface than an interactive tool

The Shikra



Shikra (pins)

UART

TX	1
RX	2
GND	18

JTAG

TCK	1
TDI	2
TDO	3
TMS	4
GND	18

SPI

SCK	1
SDI	2
SDO	3
*CS	4
GND	18

The "Shikra" Documentation (26Dec2014)
http://www.xipiter.com
© 2014 Xipiter LLC

- http://www.xipiter.com/uploads/2/4/4/8/24485815/shikra_documentation.pdf

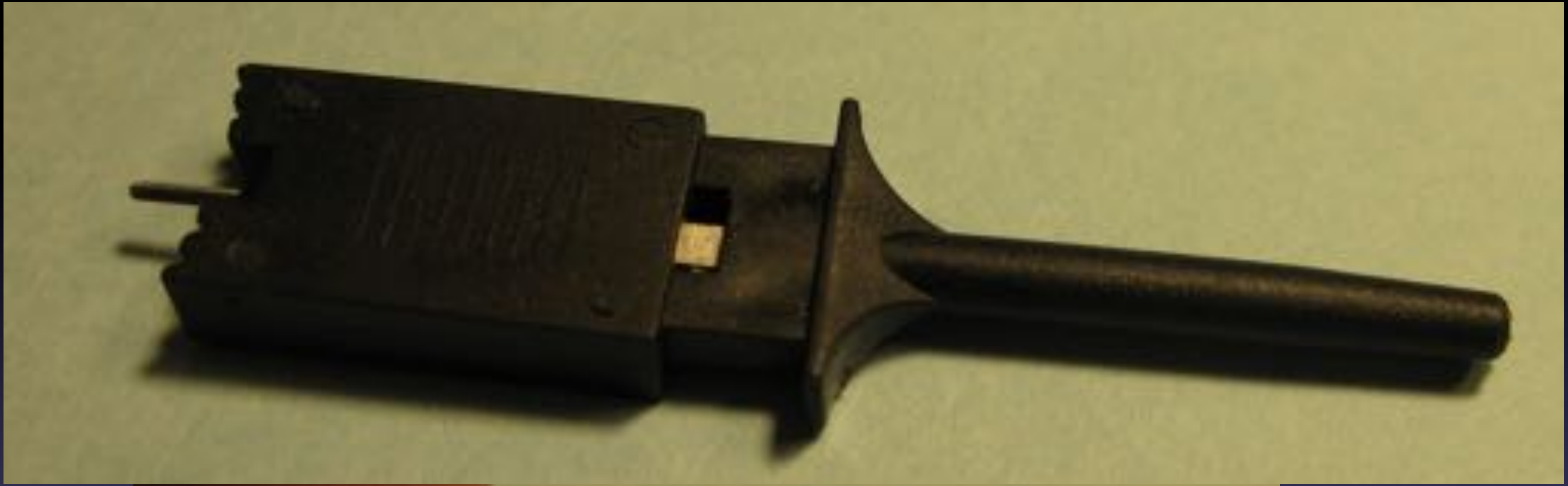
Shikra Availability

- The are available online from int3.cc for about $45USD

Connecting To The Chip

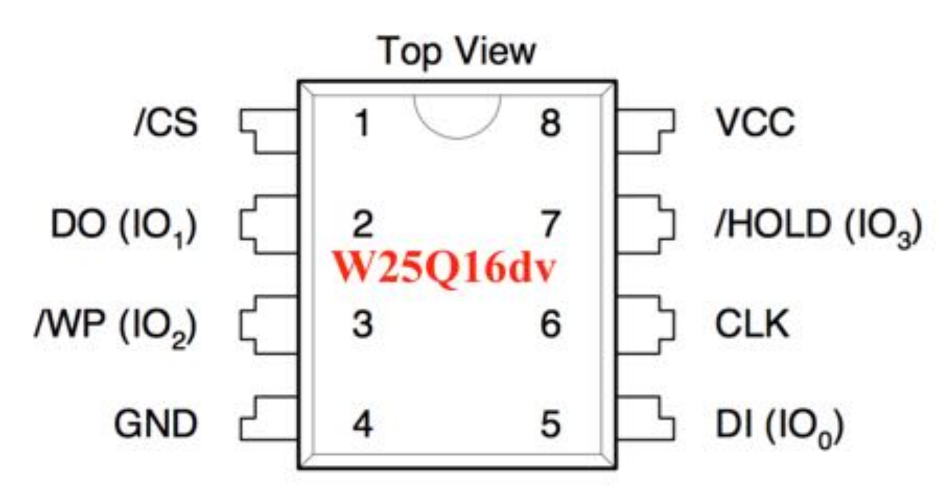
- Now that we know the chip and what protocol it speaks it is just a matter of connecting each signal from the chip to the device
- This is often the hardest (most painful) part of the entire process

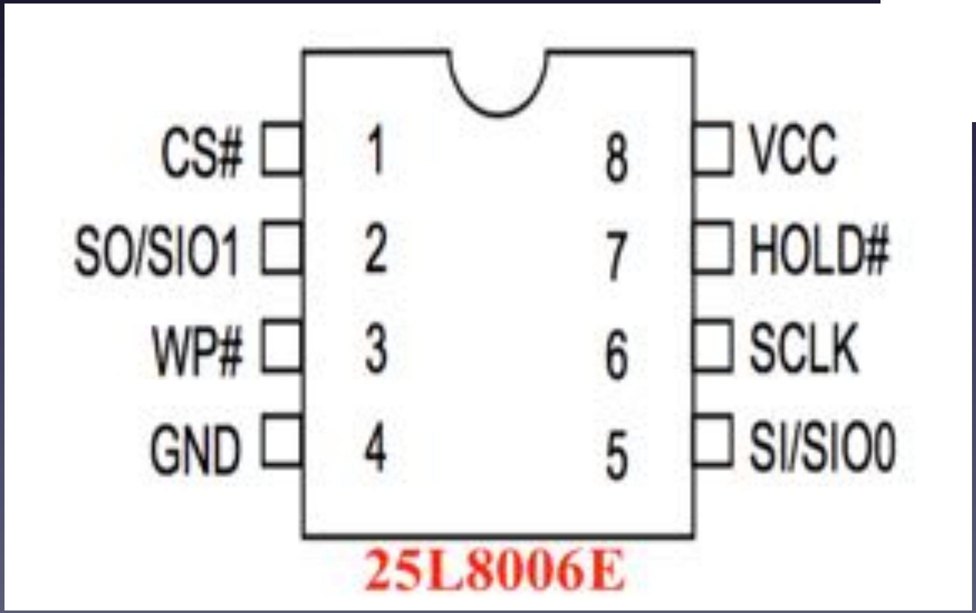
The Lead Clip



They Almost Look Alike

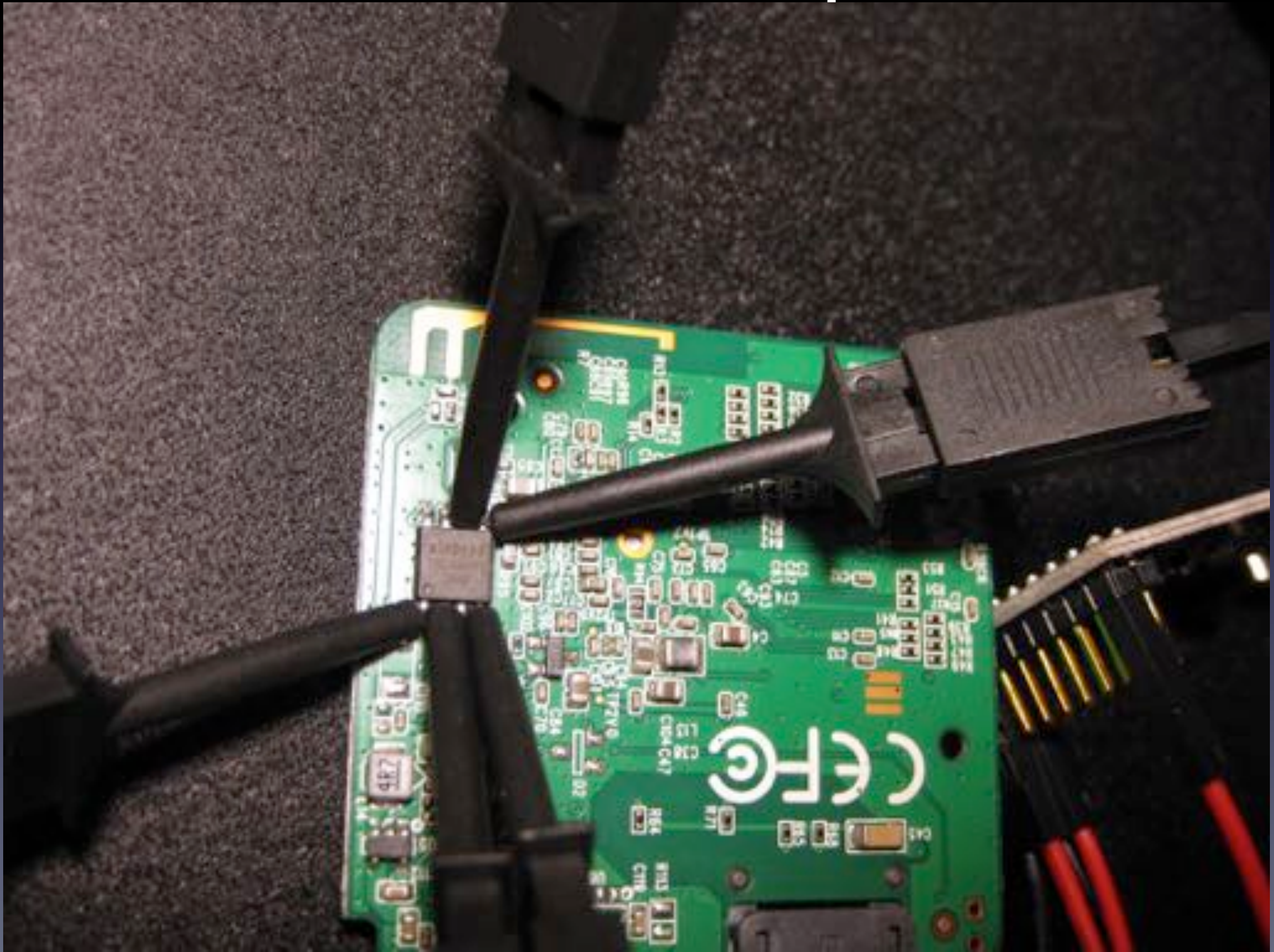
- Slightly different terminology but that is OK





Both are 8 pin SPI chips

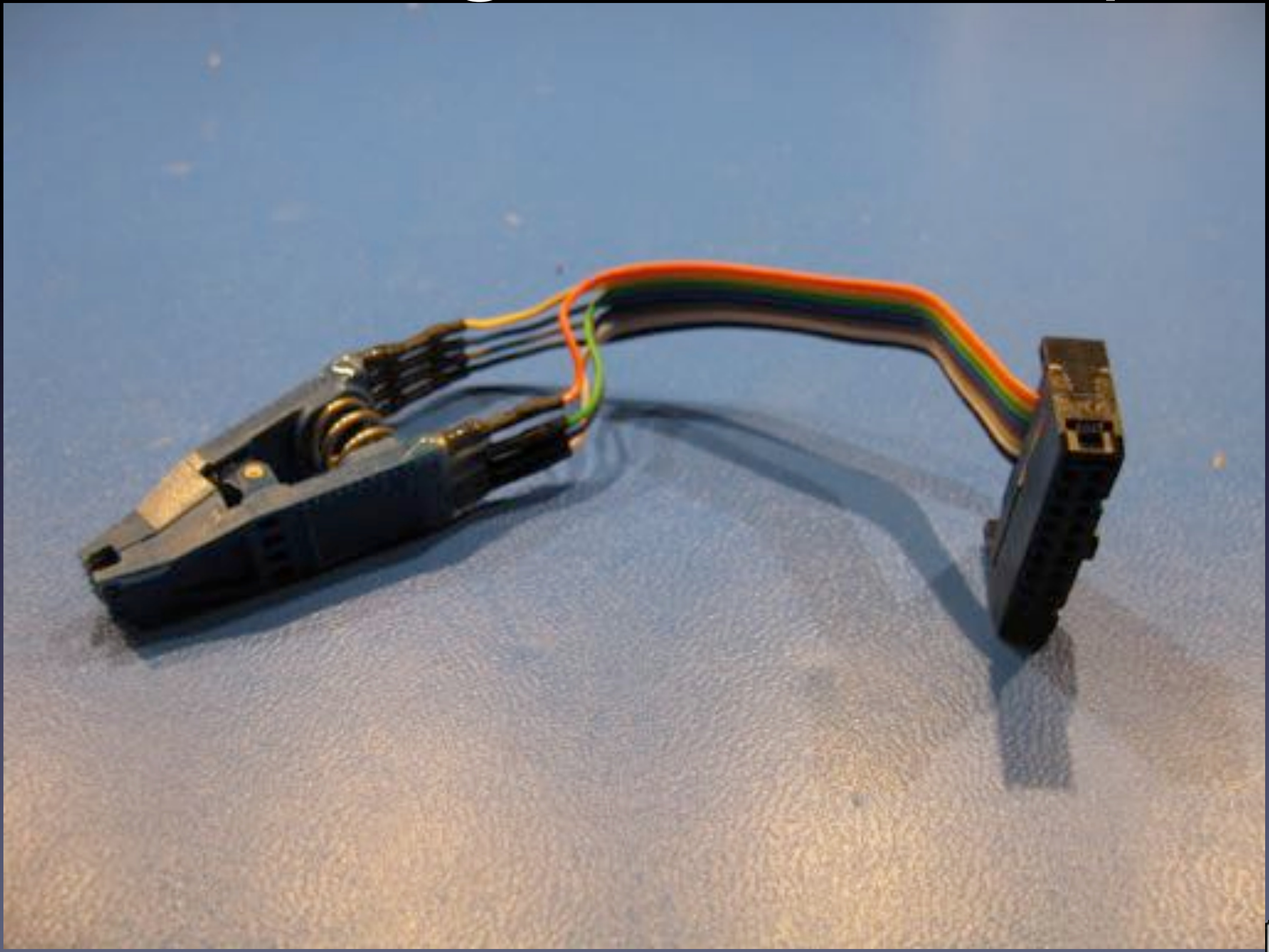
Match the Pin to the input on the adapter



Not the best solution

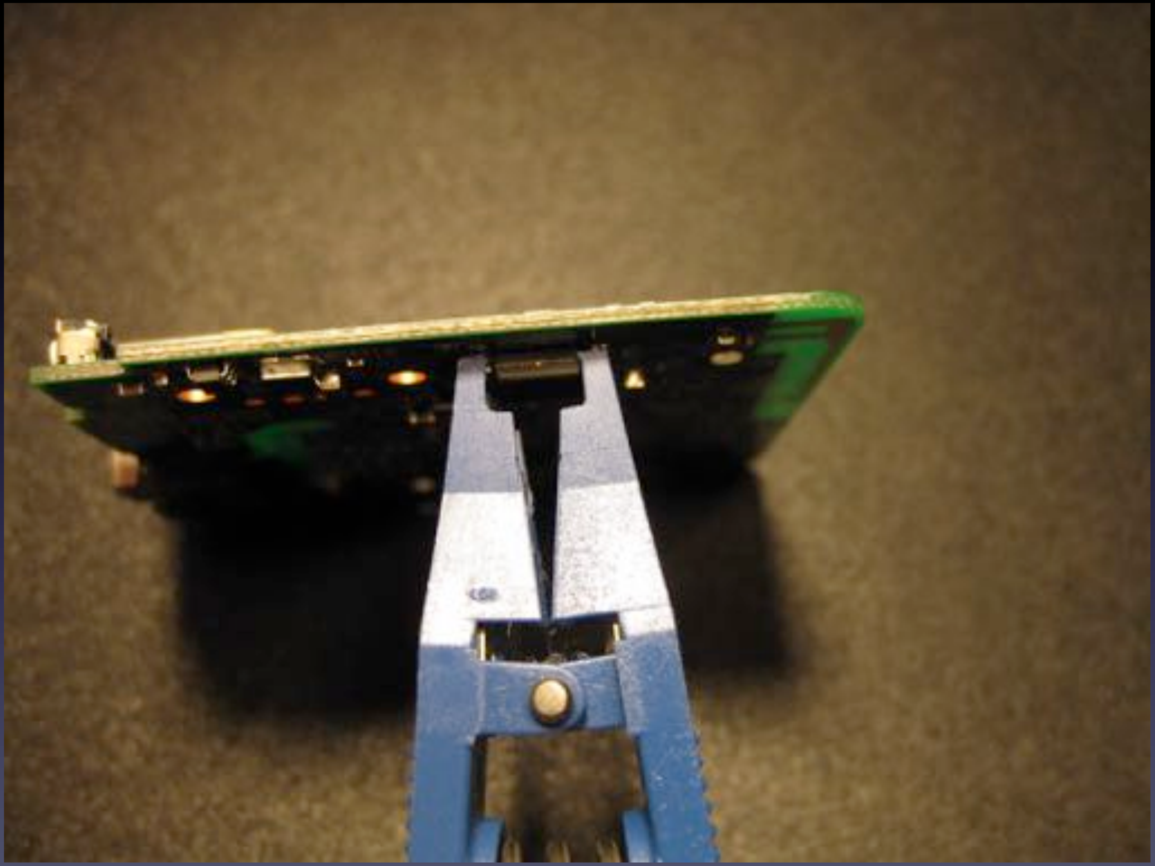
- The clips pop off all the time
- Really small clips tend to be expensive
- Clips can short if not given enough clearance
- (And these are actually physically large leads on these chips

Investing In IC Test Clips



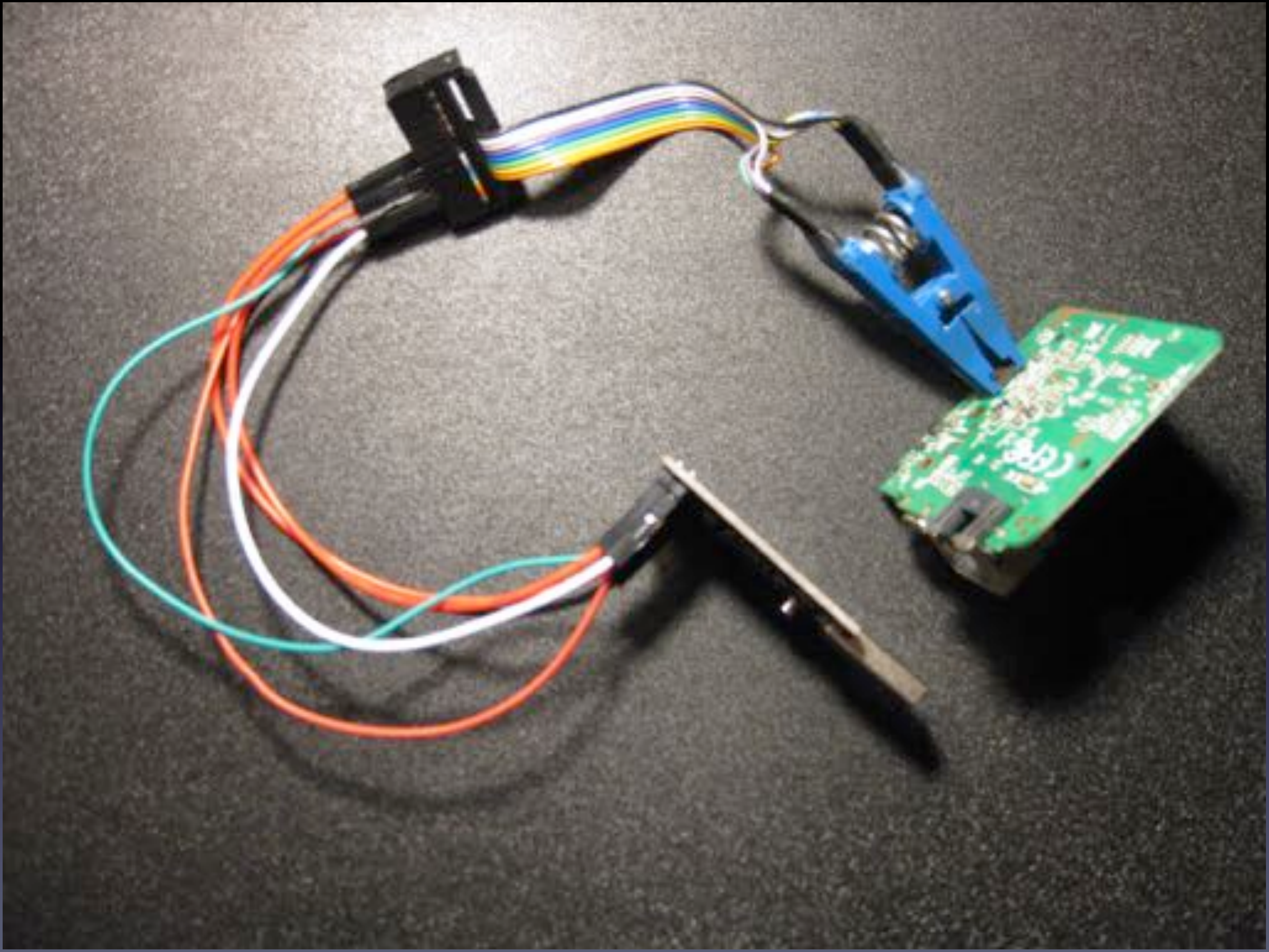
Save Time and Trouble

- These clips can run $20-$30 USD but save time
- Attach to all the chip legs at once



Where does the chip get power?

- You have a decision to make here
- Besides the signals being connected to the adapter you need power
  - The adapter and the chip must share a common ground
- Either the device must be powered up and the ground connected to the adapter
- Or the adapter must power the chip



The anticlimactic end of a SPI firmware download

- These devices on their own are good for bit banging
  - Accessing one byte of data at a time
- Since they are easily accessible from other programs it is better to use them as just adapters that speak to protocol to the chip

Enter the helpers

- For SPI use flashrom which is available to install on most Linux distros
  - https://www.flashrom.org/Flashrom
- For a wired up and connected Shikra

```
flashrom -p ft2232_spi:type=232H -r spidump.bin
```

Enter the helpers

- For JTAG on Linux use OpenOCD
- http://openocd.org/
  - More on JTAG later
  - Grab the OpenOCD configuration file for the Shikra here
    - http://www.xipiter.com/musings/using-the-shikra-to-attack-embedded-systems-getting-started
    - You may need to also specify some information about the target in the configuration file

OpenOCD for JTAG

- Once the configuration file is complete just type

```
openocd –f config-file
```

- Then connect to localhost port 4444 via telnet to connect to the openocd process
- Firmware dumps can be made via the “dump_image” command
  - Part or all of the memory can be dumped

After all this it was pretty easy

- You need to find the right chip that contains the firmware
- You need a reliable physical connection to it
- The adapter needs to work with the chip
  - Speed, voltage, etc…
- Only then will you be able to pull the firware for analysis

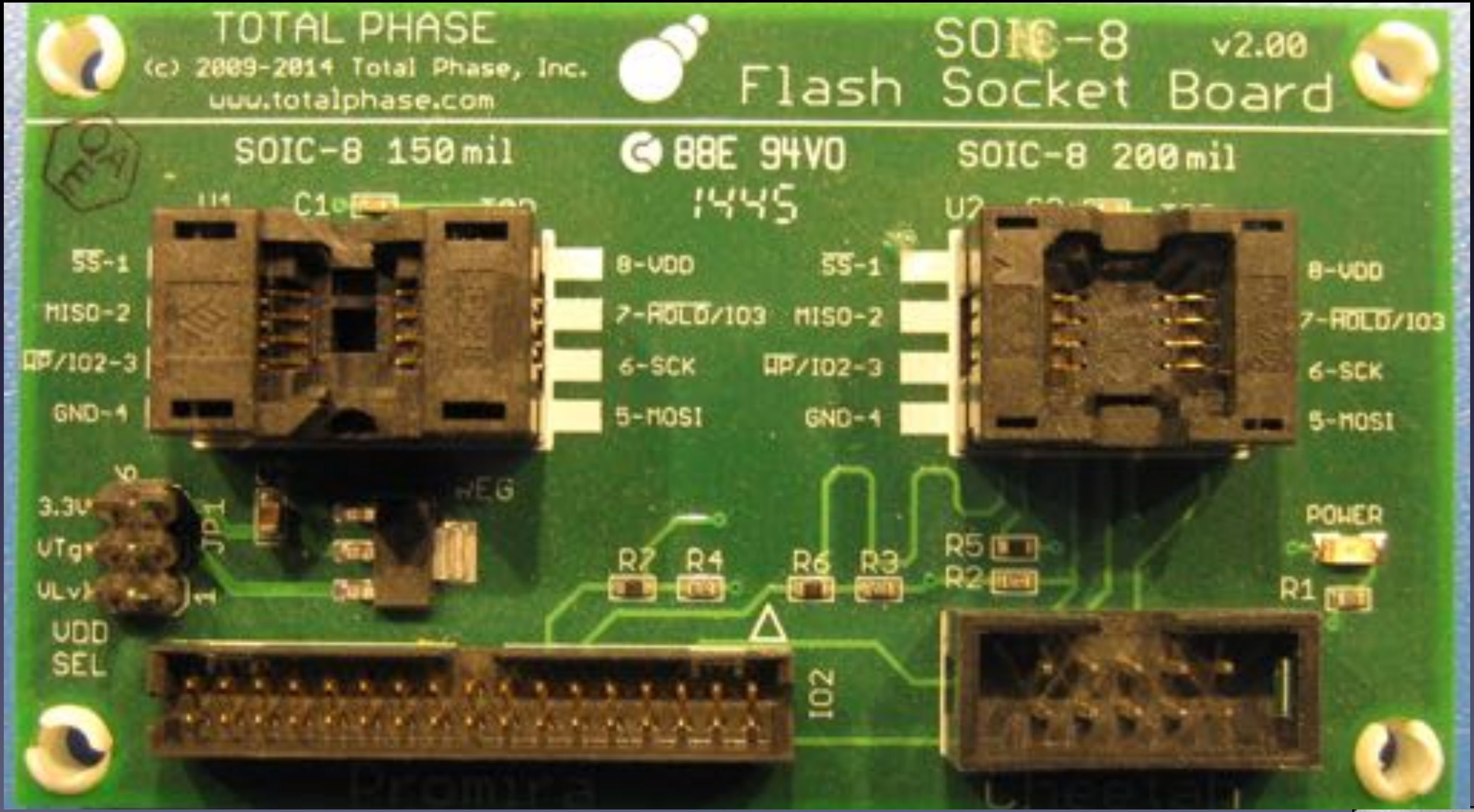
It is not always that easy

- In the example shown the memory had fairly large leads we could connect to
- Check out the size of some of the devices in the IoT Village for a challenge

Small Form Factor Chips

- In the case of chips small leads we often solder to each lead a very fine wire to make a connection
  - Remember my love of having a microscope?
- Or you de-solder the chip and put it into a specially designed holder for access

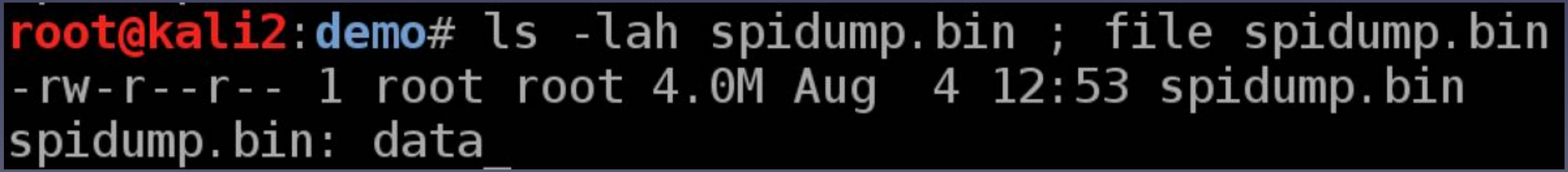
Each size of chip has it's own socket



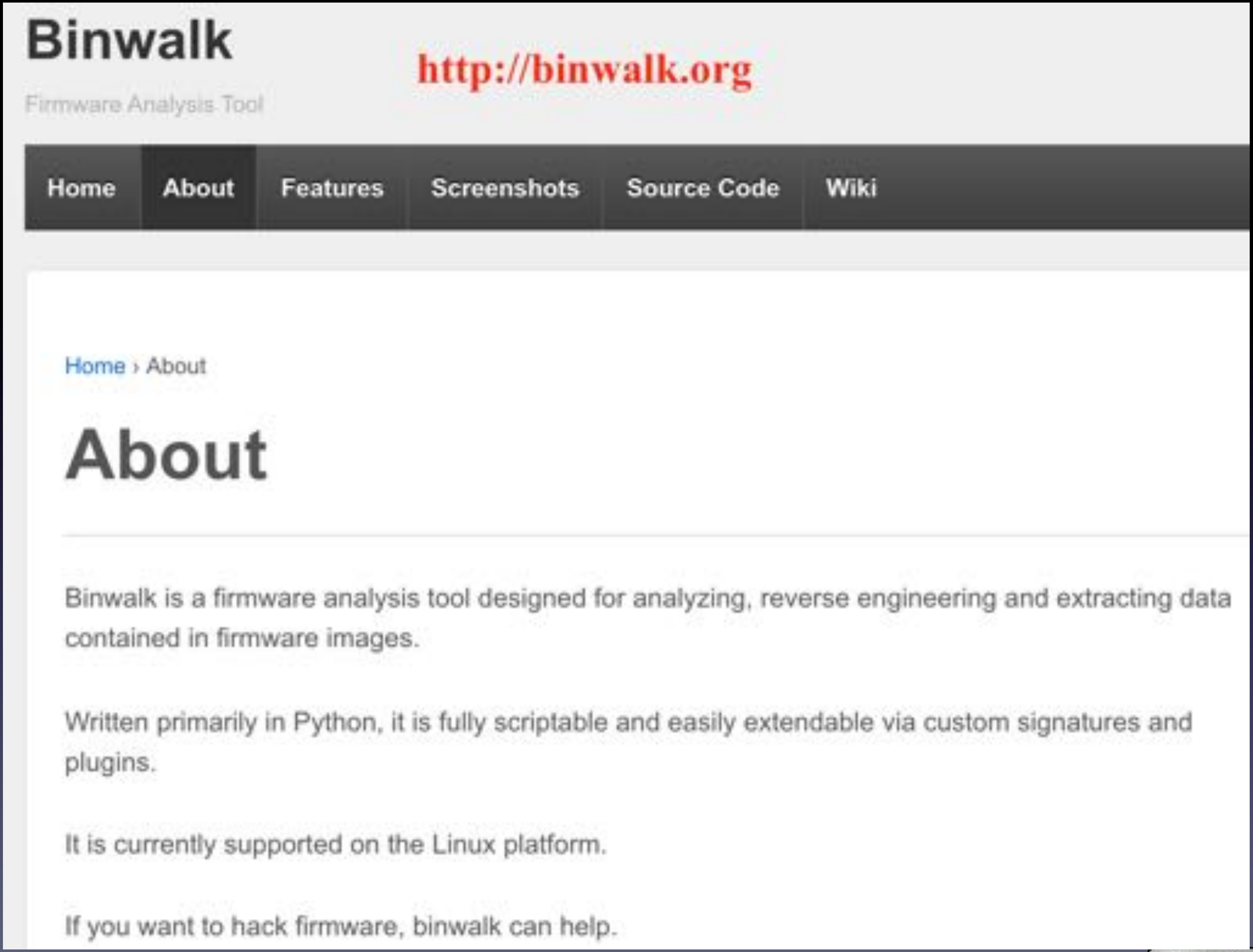
CNV-QFP32-DIP
CNV-SOP-NDIP14
CNV-SSOP20-DIP
CNV-SSOP28
CNV-SSOP28

Post Exploitation (Firmware Analysis)

- We are only part of the way there –
  - We've opened the case
  - Identified the chip with the firmware
  - Downloaded the firmware to our PC as a big binary blob



```
root@kali2:demo# ls -lah spidump.bin ; file spidump.bin
-rw-r--r-- 1 root root 4.0M Aug  4 12:53 spidump.bin
spidump.bin: data_
```



Binwalk

Firmware Analysis Tool

http://binwalk.org

About

Binwalk is a firmware analysis tool designed for analyzing, reverse engineering and extracting data contained in firmware images.

Written primarily in Python, it is fully scriptable and easily extendable via custom signatures and plugins.

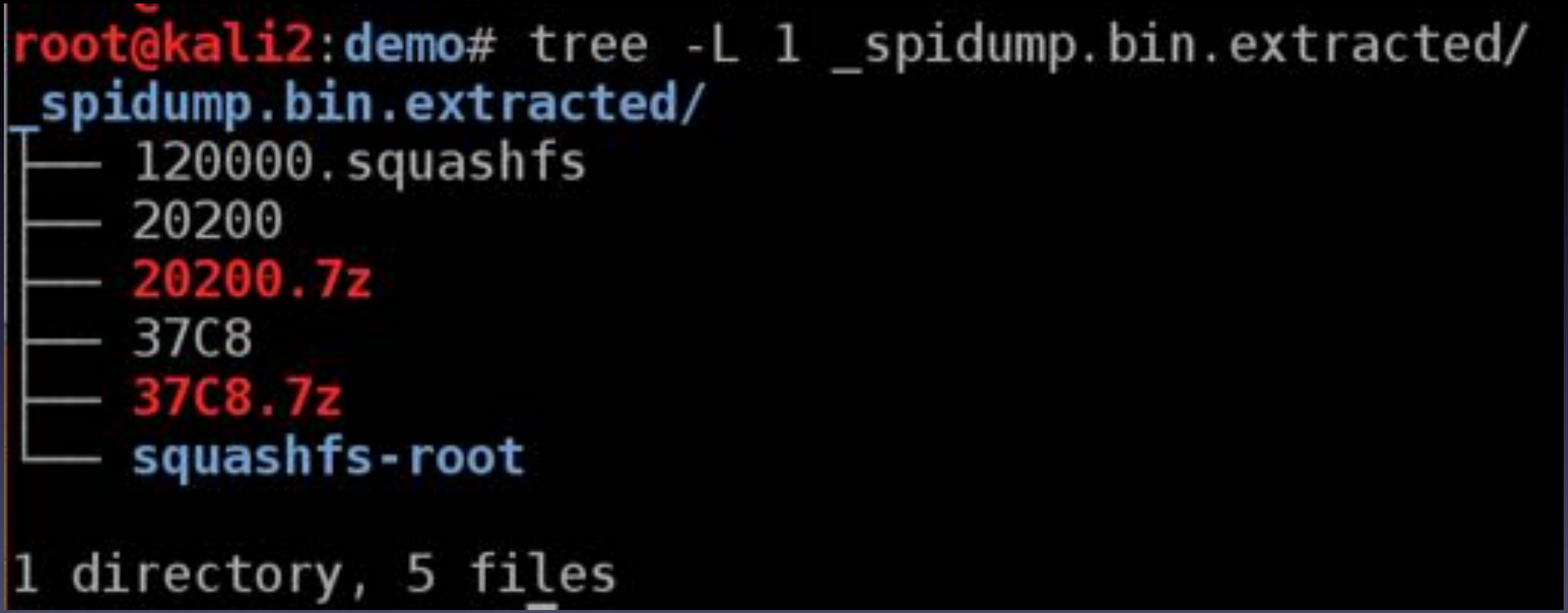
It is currently supported on the Linux platform.

If you want to hack firmware, binwalk can help.

root@kali2:demo# binwalk -e spidump.bin

```
DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
12880         0x3250          U-Boot version string, "U-Boot 1.1.4 (May  6 2013 - 13:20:35)"
14216         0x3788          uImage header, header size: 64 bytes, header CRC: 0xBA7F2047, created: Sun May  5
22:20:35 2013, image size: 34860 bytes, Data Address: 0x80010000, Entry Point: 0x80010000, data CRC:
0x263C3839, OS: Linux, CPU: MIPS, image type: Firmware Image, compression type: lzma, image name: "u-
boot image"
14280         0x37C8          LZMA compressed data, properties: 0x5D, dictionary size: 33554432 bytes,
uncompressed size: 99104 bytes
131072        0x20000         TP-Link firmware header, firmware version: 3.13.33, image version: "ver. 1.0", product
ID: 0x8410008, product version: 1, kernel load address: 0x80002000, kernel entry point: 0x801AA240, kernel
offset: 512, kernel length: 813084, rootfs offset: 1048576, rootfs length: 2883584, bootloader offset: 0,
bootloader length: 0
131584        0x20200         LZMA compressed data, properties: 0x5D, dictionary size: 33554432 bytes,
uncompressed size: 2317284 bytes
1179648       0x120000        Squashfs filesystem, little endian, version 4.0, compression:lzma, size: 2652846
bytes,  537 inodes, blocksize: 131072 bytes, created: Sun May  5 22:32:12 2013
```

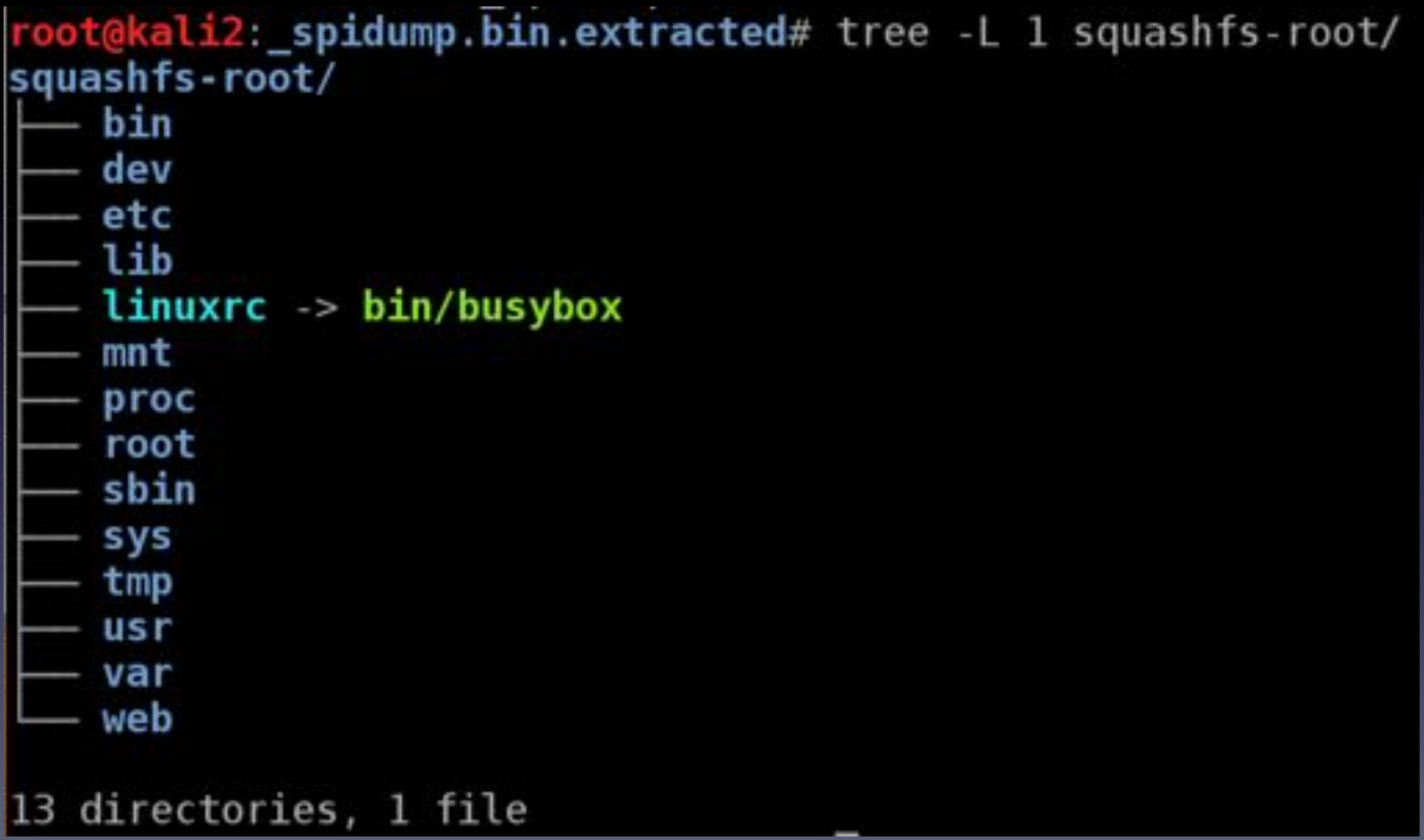
binwalk took care of the extraction



```
root@kali2:demo# tree -L 1 _spidump.bin.extracted/
_spidump.bin.extracted/
├── 120000.squashfs
├── 20200
├── 20200.7z
├── 37C8
├── 37C8.7z
└── squashfs-root

1 directory, 5 files
```

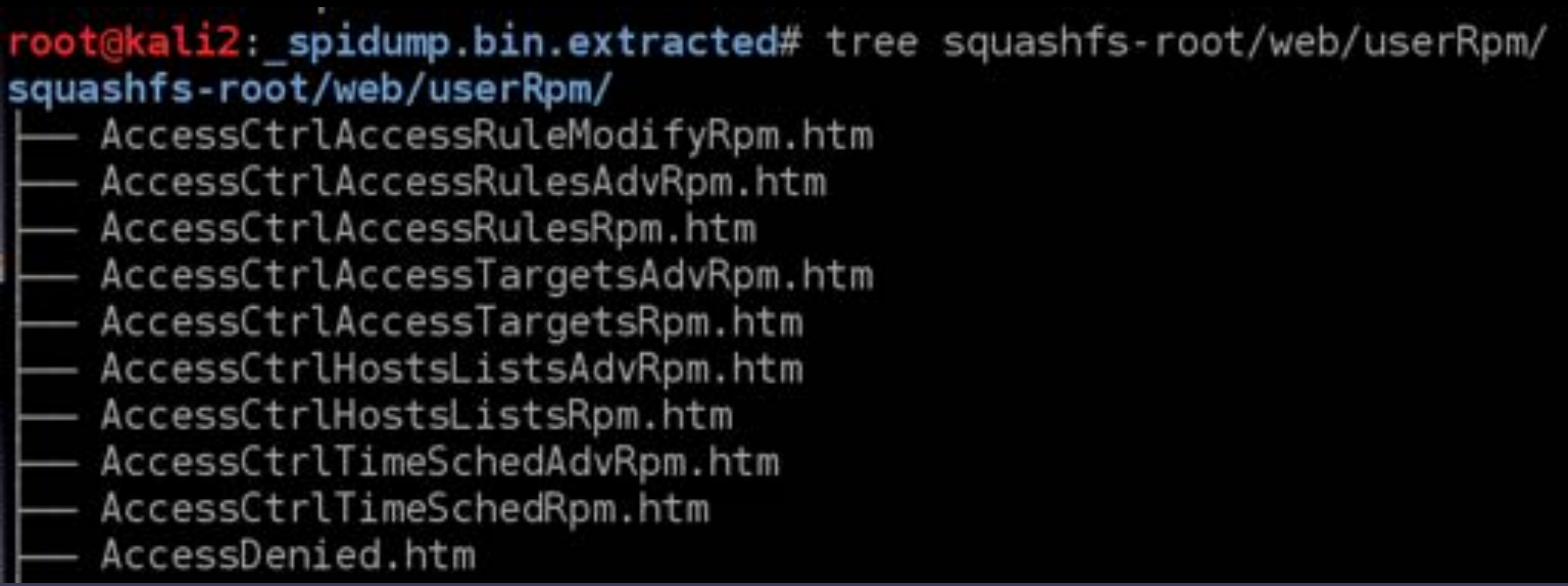
All The Extraction We Needed



```
root@kali2:_spidump.bin.extracted# tree -L 1 squashfs-root/
squashfs-root/
├── bin
├── dev
├── etc
├── lib
├── linuxrc -> bin/busybox
├── mnt
├── proc
├── root
├── sbin
├── sys
├── tmp
├── usr
├── var
└── web

13 directories, 1 file
```

Time To Begin Auditing



```
root@kali2:_spidump.bin.extracted# tree squashfs-root/web/userRpm/
squashfs-root/web/userRpm/
├── AccessCtrlAccessRuleModifyRpm.htm
├── AccessCtrlAccessRulesAdvRpm.htm
├── AccessCtrlAccessRulesRpm.htm
├── AccessCtrlAccessTargetsAdvRpm.htm
├── AccessCtrlAccessTargetsRpm.htm
├── AccessCtrlHostsListsAdvRpm.htm
├── AccessCtrlHostsListsRpm.htm
├── AccessCtrlTimeSchedAdvRpm.htm
├── AccessCtrlTimeSchedRpm.htm
├── AccessDenied.htm
```

The whole OS is now available for auditing

The whole OS is now available for auditing

- We never really powered the box up
- We never knew the root password
- We are now software auditing embedded hardware device as if we were root
- All with a $50 adapter and free open source software
  - What a time to be alive!

The GUI Comment

- At the start I pointed out that these devices had a UI
  - That led me to believe I'd find Linux or some other OS on them
- Some devices just have a binary blob that is the entire firmware image
  - Strings, IDA Pro, Binary Ninja, Vivisect, redare
    - Look at the CPU architecture and pick you analysis tool of choice

Reporting (Vendor or Client Notifications)

- Hardware can be a funny thing for those of us from the software realm
  - It can be very hard to update
    - This means a very long roll out of patches on devices that are not always easy to patch
  - Manufacturers can be very sensitive to issues in infrastructure or medical devices

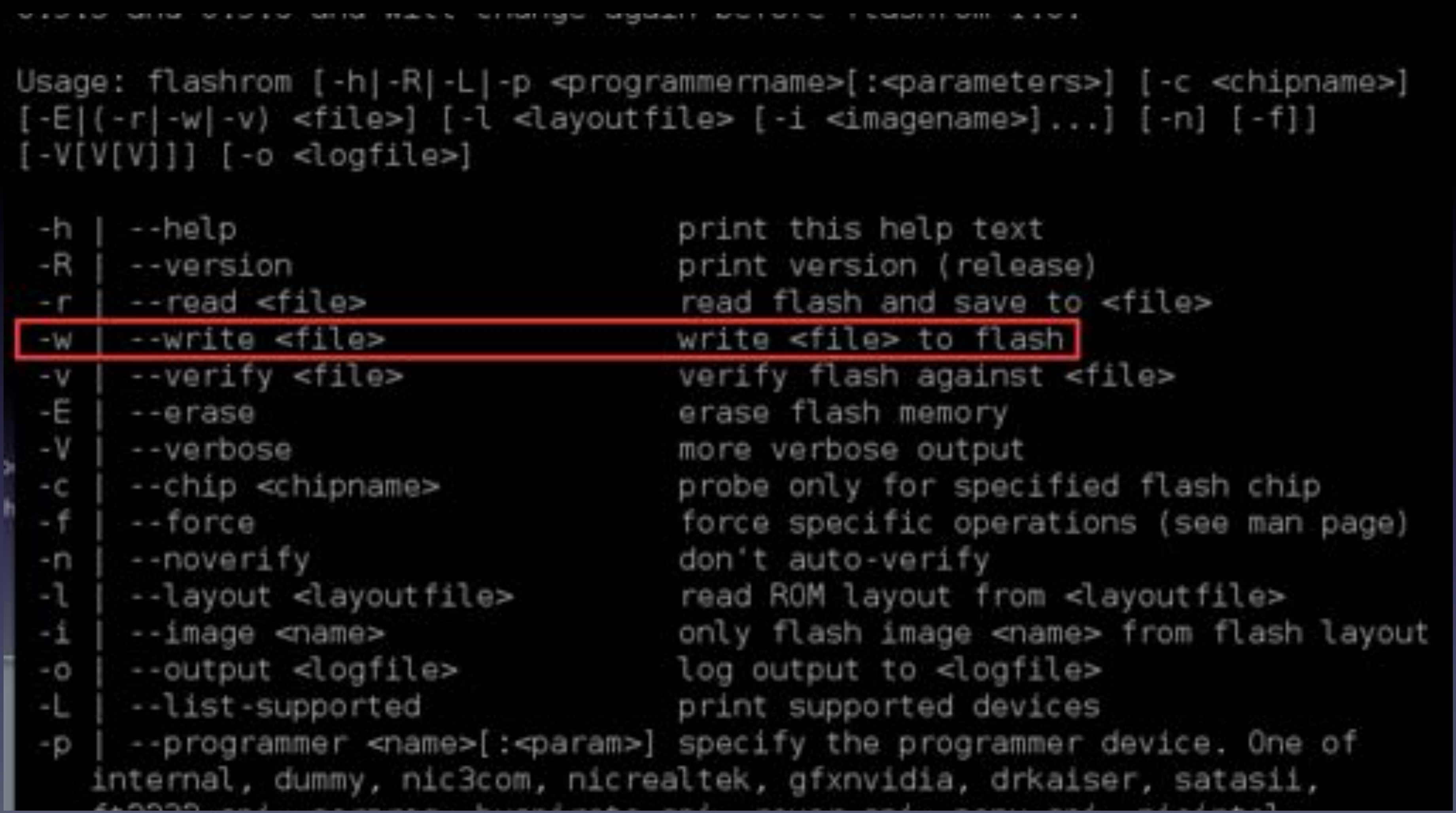
It all looks so simple

- We bought a few inexpensive devices at our local Frys
- We pried them open
- We identified the firmware storage chips
- We downloaded the software image
- We turned to into a file system we could study
- We had a great time and did not electrocute ourselves

But now what

- squashfs-tools certainly lets us make our own file systems base on the code we extracted
- Binwalk told us the exact image layout
- It is not much harder to create a new image

flashrom also writes



```
0.9.5 and 0.9.6 and will change again before flashrom 1.0.

Usage: flashrom [-h|-R|-L|-p <programmername>[:<parameters>] [-c <chipname>]
[-E|(-r|-w|-v) <file>] [-l <layoutfile> [-i <imagename>]...] [-n] [-f]]
[-V[V[V]]] [-o <logfile>]

 -h | --help                          print this help text
 -R | --version                       print version (release)
 -r | --read <file>                   read flash and save to <file>
 -w | --write <file>                  write <file> to flash
 -v | --verify <file>                 verify flash against <file>
 -E | --erase                         erase flash memory
 -V | --verbose                       more verbose output
 -c | --chip <chipname>               probe only for specified flash chip
 -f | --force                         force specific operations (see man page)
 -n | --noverify                      don't auto-verify
 -l | --layout <layoutfile>           read ROM layout from <layoutfile>
 -i | --image <name>                  only flash image <name> from flash layout
 -o | --output <logfile>              log output to <logfile>
 -L | --list-supported                print supported devices
 -p | --programmer <name>[:<param>] specify the programmer device. One of
    internal, dummy, nic3com, nicrealtek, gfxnvidia, drkaiser, satasii,
```

We are right back to the audit question again

- Seeing what has been presented it is not hard to question the validity of the software in embedded devices

Thanks

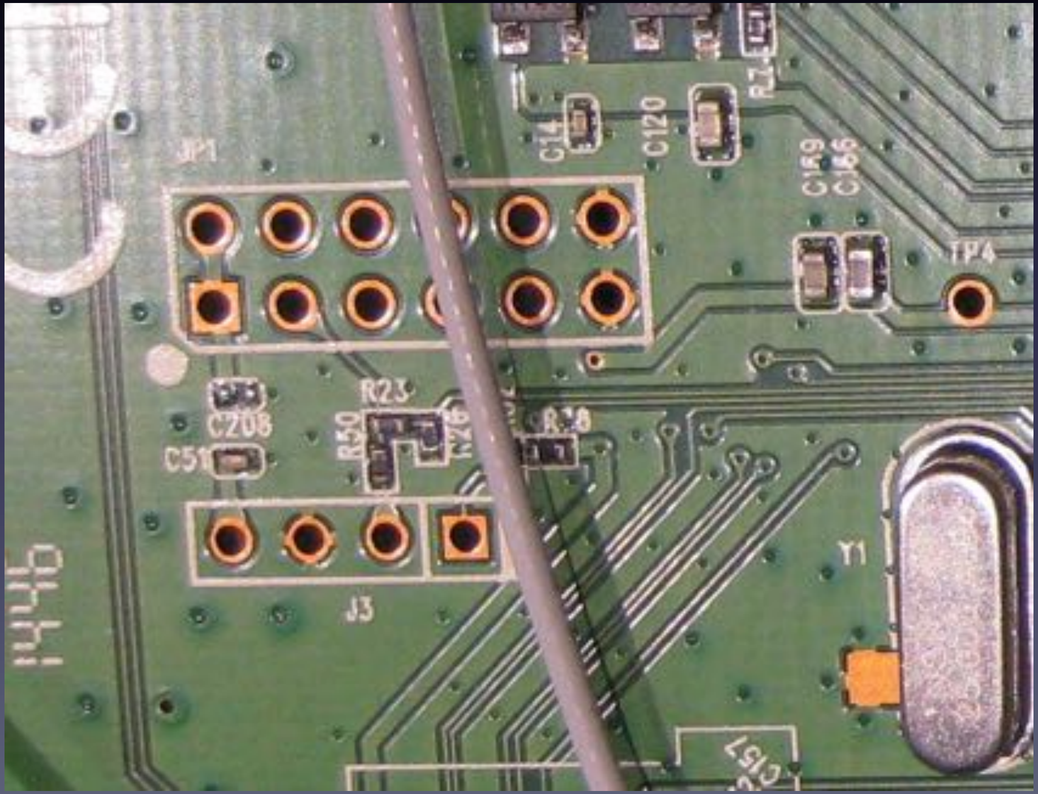
I hope this was interesting and let me know if you have any questions!

But wait there is more

- Everything presented went really well but things do not always fall so easily into place
- Where do you go next (after buying a nice trinocular microscope)

If your problem is unlabeled connectors on the board

- Remember these on the TP-Link TL-WR841N ?

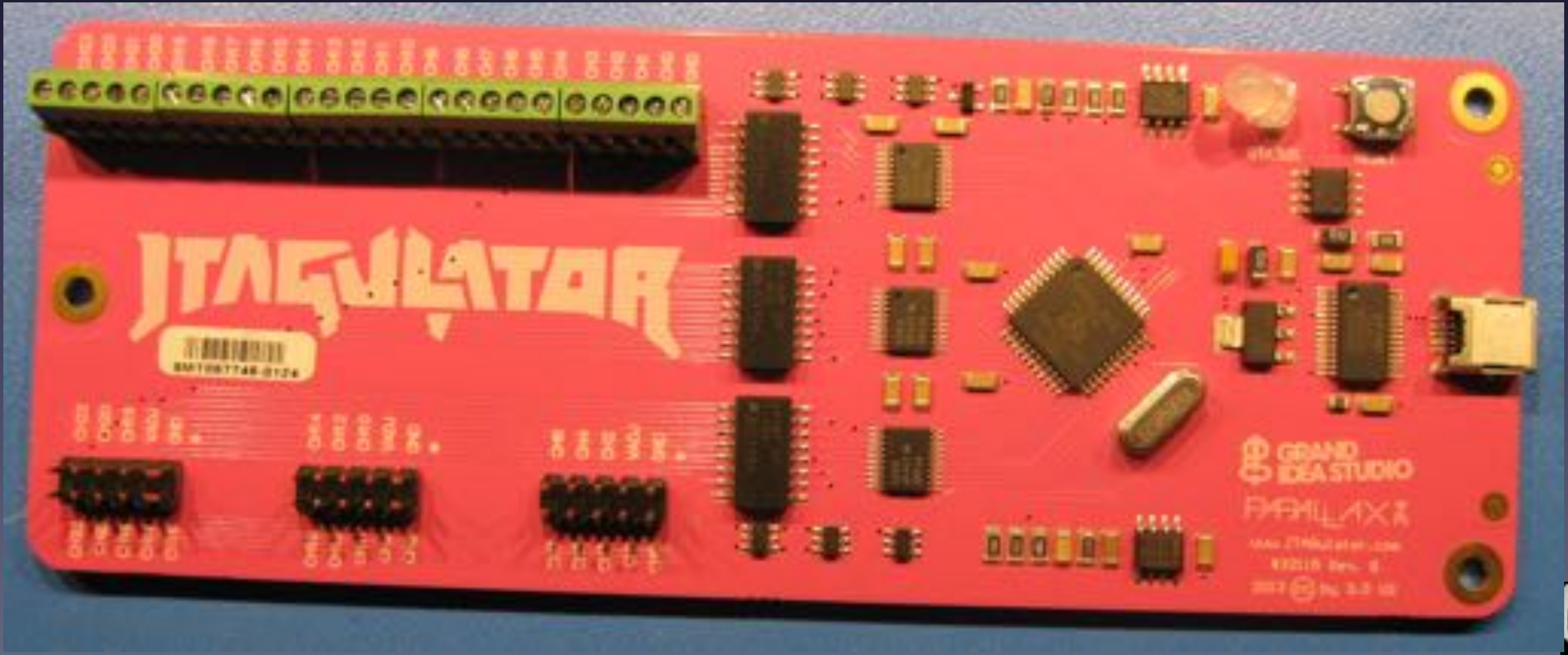


What are they?

Serial ports?
UARTS?
JTAG?

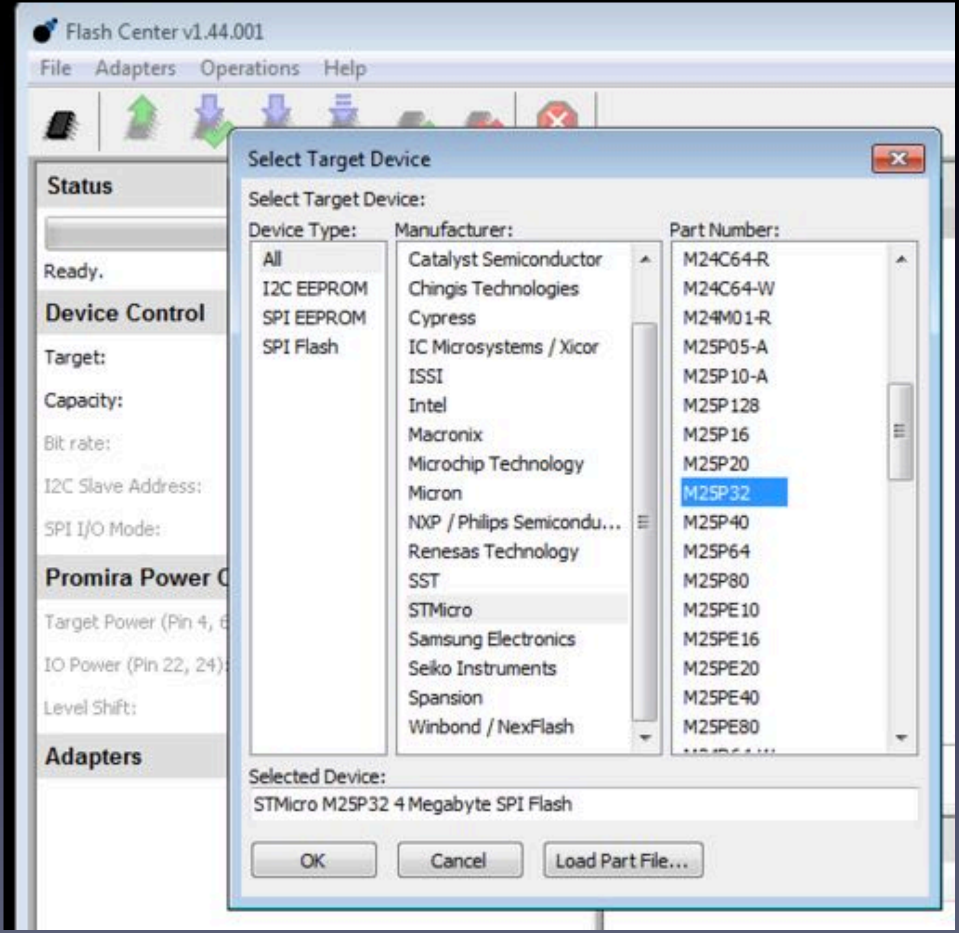
If your problem is unlabeled connectors on the board

- "JTAGulator is an open source hardware tool that assists in identifying OCD connections from test points, vias, or component pads on a target device." $174USD
- http://www.grandideastudio.com/jtagulator/



If your problem is slow reading and writing of chips

- Stepping up to a full programmer
- Generally much faster
- They are often also preconfigures with the parameters for many common types of memory



Ready.

Target:

Capacity:

Bit rate:

I2C Slave Address:

SPI I/O Mode:

Target Power (Pin 4, 6

IO Power (Pin 22, 24):

Level Shift:

Select Target Device:

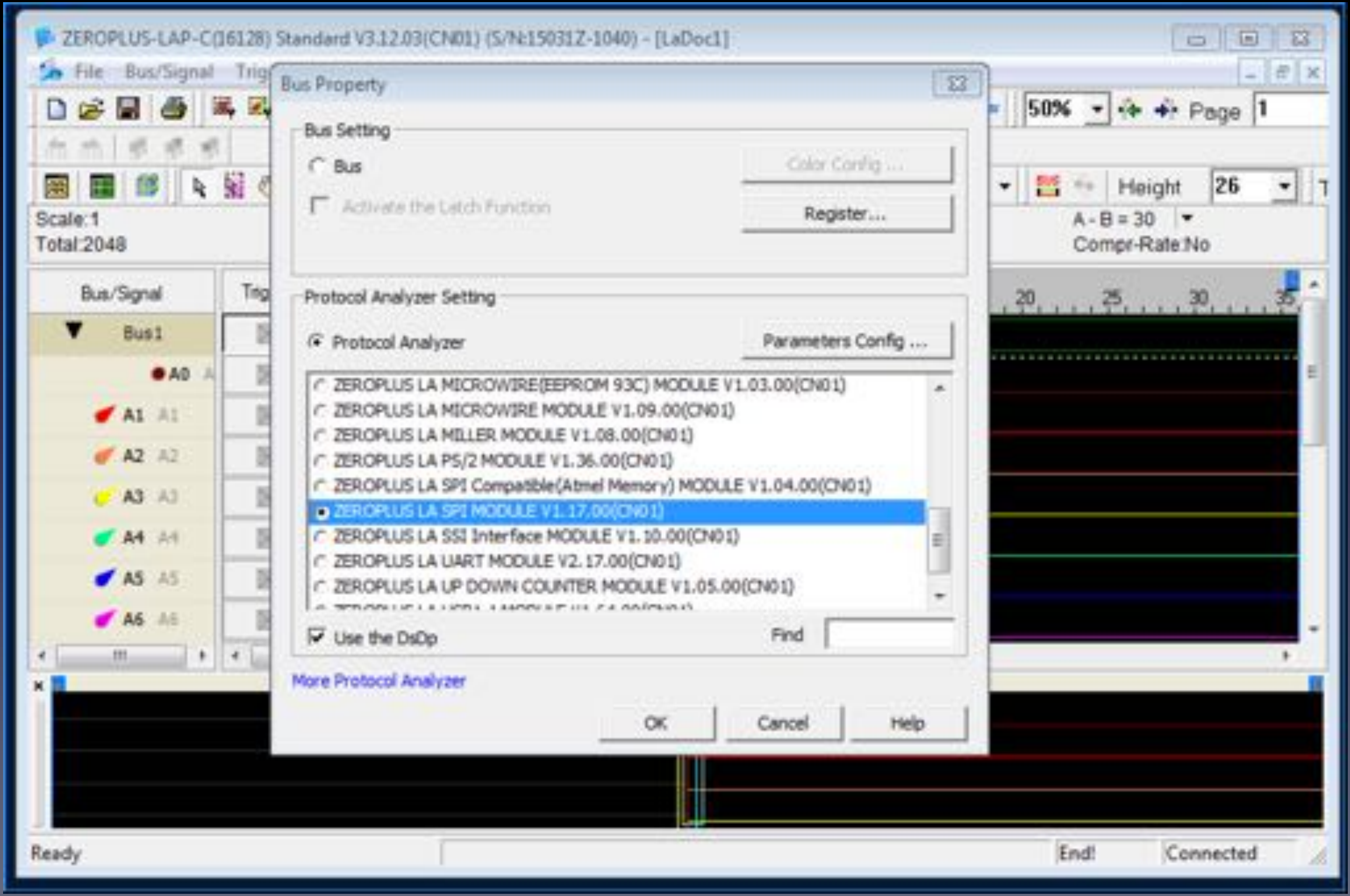
Device Type:
- All
- I2C EEPROM
- SPI EEPROM
- SPI Flash

Manufacturer:
- Catalyst Semiconductor
- Chingis Technologies
- Cypress
- IC Microsystems / Xicor
- ISSI
- Intel
- Macronix
- Microchip Technology
- Micron
- NXP / Philips Semicondu...
- Renesas Technology
- SST
- STMicro
- Samsung Electronics
- Seiko Instruments
- Spansion
- Winbond / NexFlash

Part Number:
- M24C64-R
- M24C64-W
- M24M01-R
- M25P05-A
- M25P10-A
- M25P128
- M25P16
- M25P20
- M25P32
- M25P40
- M25P64
- M25P80
- M25PE10
- M25PE16
- M25PE20
- M25PE40
- M25PE80

Selected Device: STMicro M25P32 4 Megabyte SPI Flash

If your problem is wanting to know more about the protocols



ZEROPLUS-LAP-C(16128) Standard V3.12.03(CN01) (S/N:15031Z-1040) - [LaDoc1]

Bus Property

Bus Setting

○ Bus

□ Activate the Latch Function

Protocol Analyzer Setting

● Protocol Analyzer

- ○ ZEROPLUS LA MICROWIRE(EEPROM 93C) MODULE V1.03.00(CN01)
- ○ ZEROPLUS LA MICROWIRE MODULE V1.09.00(CN01)
- ○ ZEROPLUS LA MILLER MODULE V1.08.00(CN01)
- ○ ZEROPLUS LA PS/2 MODULE V1.36.00(CN01)
- ○ ZEROPLUS LA SPI Compatible(Atmel Memory) MODULE V1.04.00(CN01)
- ● ZEROPLUS LA SPI MODULE V1.17.00(CN01)
- ○ ZEROPLUS LA SSI Interface MODULE V1.10.00(CN01)
- ○ ZEROPLUS LA UART MODULE V2.17.00(CN01)
- ○ ZEROPLUS LA UP DOWN COUNTER MODULE V1.05.00(CN01)

☑ Use the DsDp

Scale:1

Total:2048

A - B = 30

Compr-Rate:No

Height 26

50%

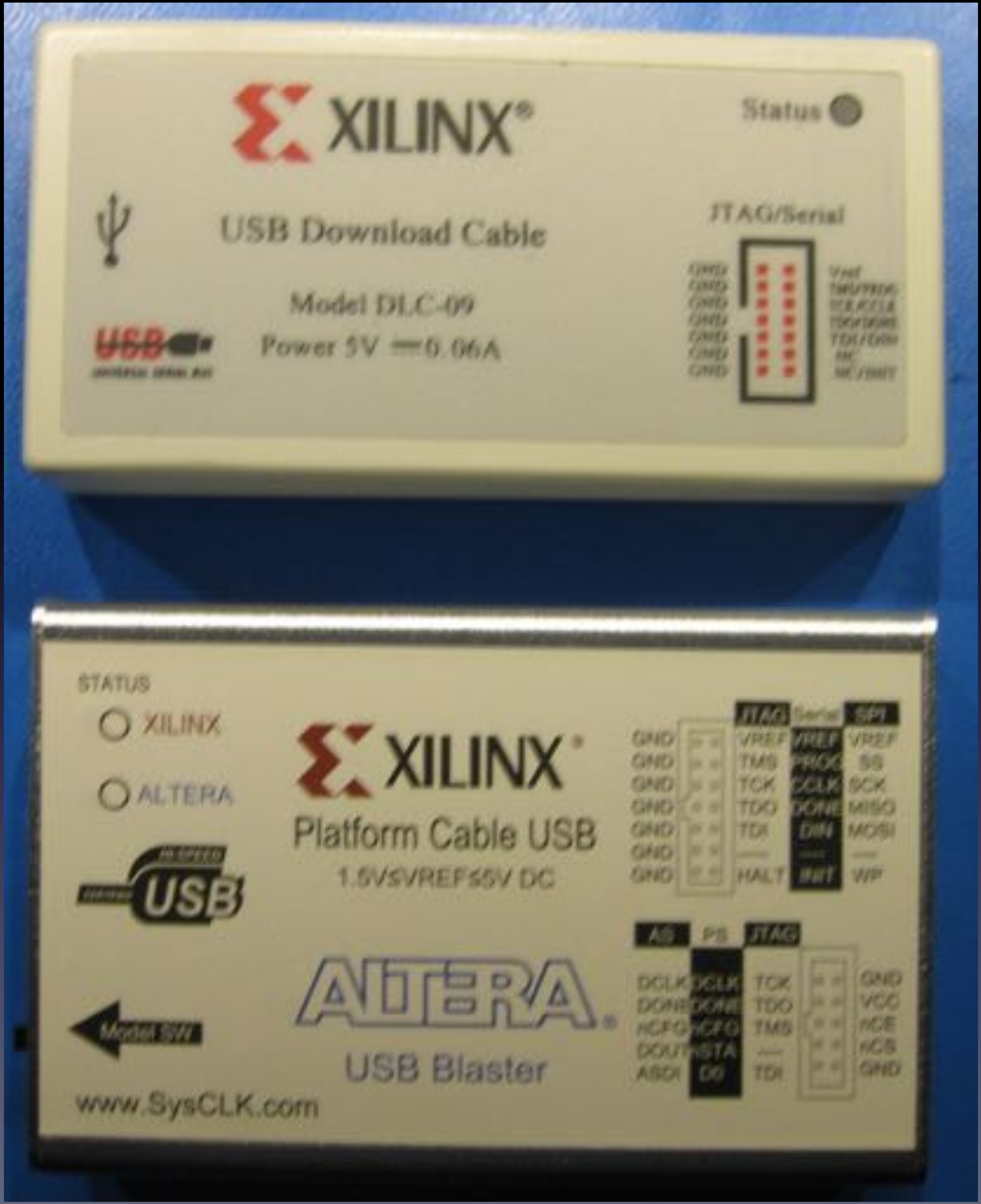
Page 1

If your problem is wanting to know more about the protocols

- Many good USB based logic analyzers have decode modules for various protocols
- Not all that useful for downloading firmware as it really decodes only one byte at a time
  - Like bit banging
- Excellent for understanding how the protocols work

If your problem is you need to debug and access firmware

- JTAG is the interface of choice
- Everybody has their own version of the JTAG adapter and they all differ slightly



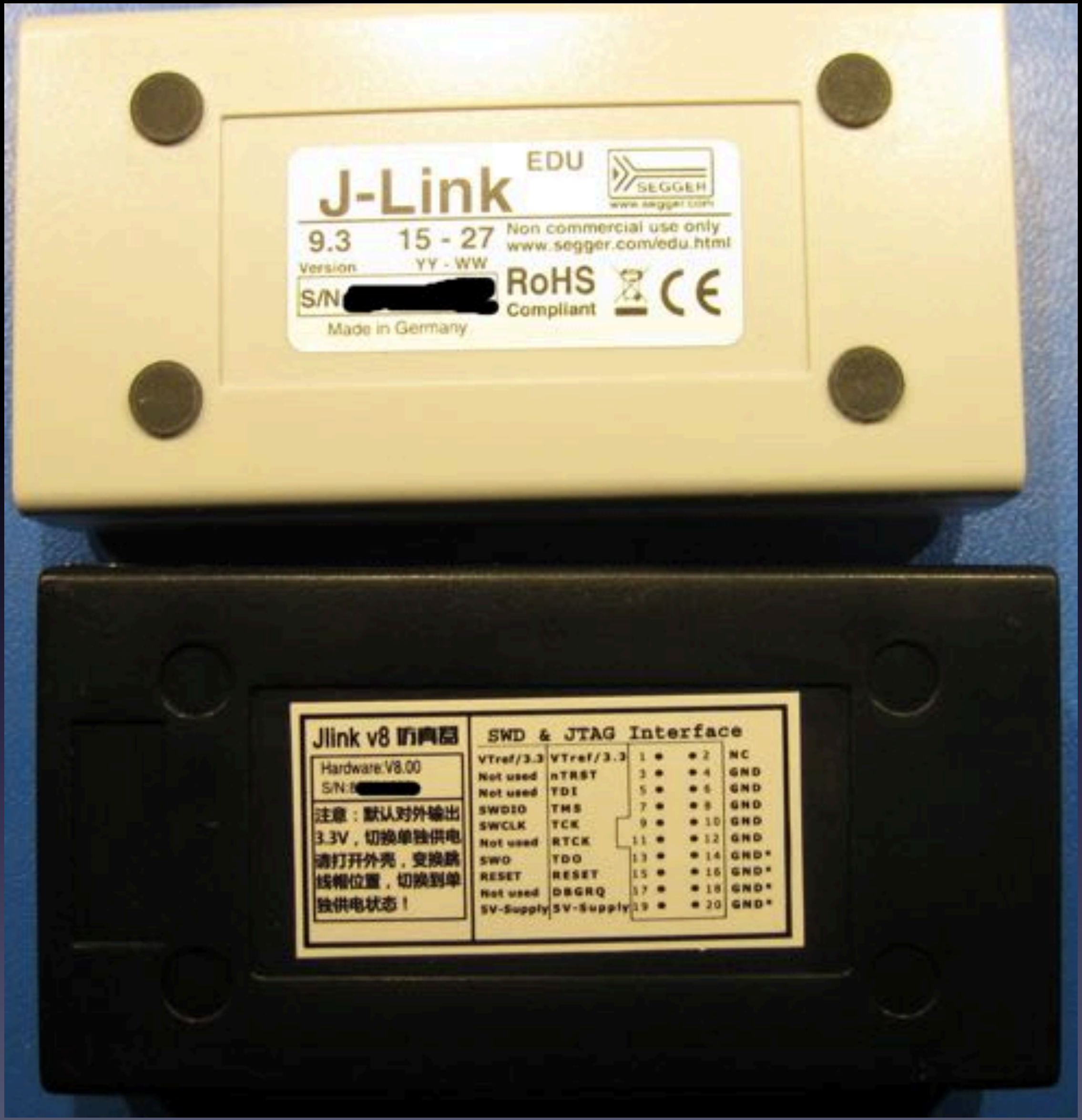
- Top: Genuine Xilinx only device
- Bottom: Chinese knockoff that works for both Xilinx and Altera FPGAs

ARM Adapters too

- Left: $69 Student version of the $399 Segger Jlink
- Right: $25 Segger Chinese knockoff

Beside the Ethical Questions

- If is pretty interesting to compare the low cost versions from China on eBay with their name brand counterparts
- One of the biggest differences I've seen is with the amount of noise on the signal lines
  - Fewer bypass caps
- If the protocols in these devices are so specific to each manufacturer what do you think the internal firmware looks like?



They run different versions of the software

- And as any good forensic examiner can tell you neither one can stand up in court (IANAL again)
    - One would violate it’s terms of use (non-commercial use)
    - One is counterfeit (but sort of works OK)

Post Defcon Updates

- Thanks for all the great questions

  (I have no business relationship with any of these, I just use the products and have bought and used them in my own research)

  - The Logic analyzer we discussed was Zeroplus LAP-C 16128 16-channel Logic Analyzer
  - The sniffers I showed were from http://www.totalphase.com/
  - The higher end programmer I showed was a http://www.dediprog.com/pd/spi-flash-solution/sf600plus

Post Defcon Updates

- Thanks for all the great questions
  - Source for the facedancer21 and a lot of other tools I used - http://hackerwarehouse.com/
    - Garrett has been doing a great job of selling pre-flashed facedancers at a great price to my past students
  - The Defcon workshop I mentioned was a shorter form of Joe's training (his site is https://securinghardware.com/course-catalog/)

Post Defcon Updates

- Thanks for all the great questions
  - The DSi Ram tracing hack I was discussing was http://scanlime.org/2009/09/dsi-ram-tracing/
  - As for dealing with encrypted firmware, please see slides 18 and 19

:>)