101 Ways to Brick Your Hardware

(With some un-bricking tips sprinkled in for good measure)

Joe FitzPatrick & Joe Grand (Kingpin)

Overview

- What's a Brick?
- Kinds of Bricks
  - 001: Bricking Firmware
  - 010: Bricking PCBs
  - 011: Bricking Connectors
  - 100: Bricking ICs
  - 101: Bricking 'WTF' scenarios
- Recap and Best Practices

TOP DEFINITION

brick

a pound or kilogram of any drug (item requires clarification from speaker as to the amount intended)

I get my dope straight off a brick. (implying said substance is pure, clean, and untampered with)

by The Butcher of Raleigh June 19, 2003

What's a Brick?

3

brick

As verb: to brick something. This is the action of rendering any small-medium size electronic device useless. This can happen whilst changing the firmware, soldering or any other process involving either hardware of software.

I bricked my mobile phone when I tried to install Linux on it.

#destroy #hack #break #nacker #kill

by jules_v March 07, 2006

What's a Brick?

Soft Brick

- Shows signs of life
- Doesn’t boot or work as intended
- May be soft-unbrickable
- Typically a software or configuration problem

Hard Brick

- Little or no sign of life
- Doesn’t even power on or flash lights
- Probably needs hardware hacking to fix it

101 Kinds of Bricks

- 001: Bricking Firmware
- 010: Bricking PCBs
- 011: Bricking Connectors
- 100: Bricking ICs
- 101: Bricking ‘WTF’ scenarios

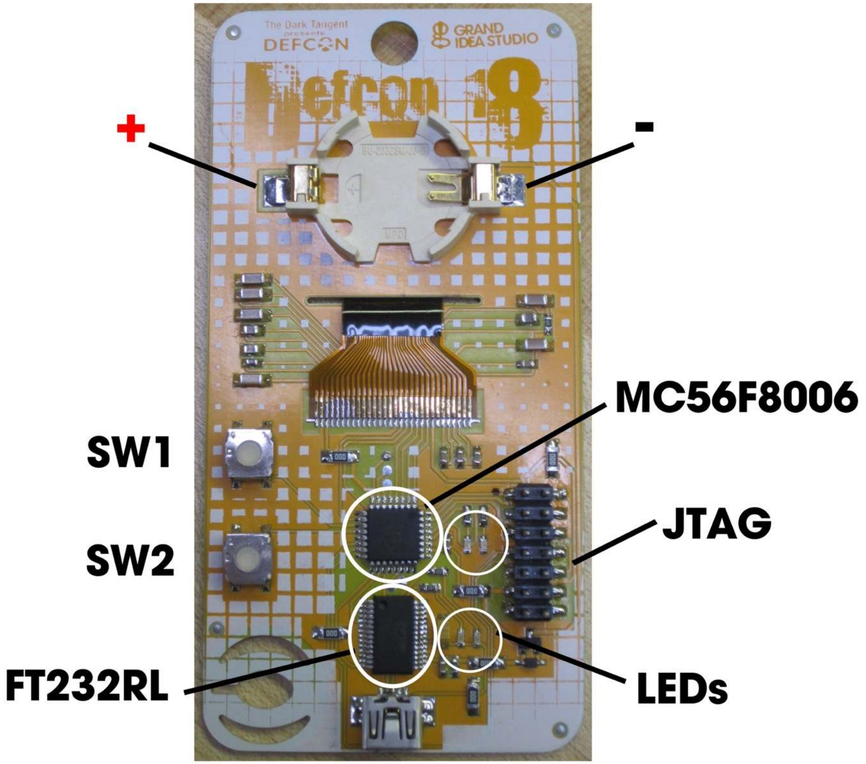
```
> xxd firmware.bin
0000000: dead dead dead dead dead dead dead dead  ................
0000010: dead dead dead dead dead dead dead dead  ................
0000020: dead dead dead dead dead dead dead dead  ................
0000030: dead dead dead dead dead dead dead dead  ................
0000040: dead dead dead dead dead dead dead dead  ................
0000050: dead dead dead dead dead dead dead dead  ................
0000060: dead dead dead dead dead dead dead dead  ................
```

001: Bricking Firmware

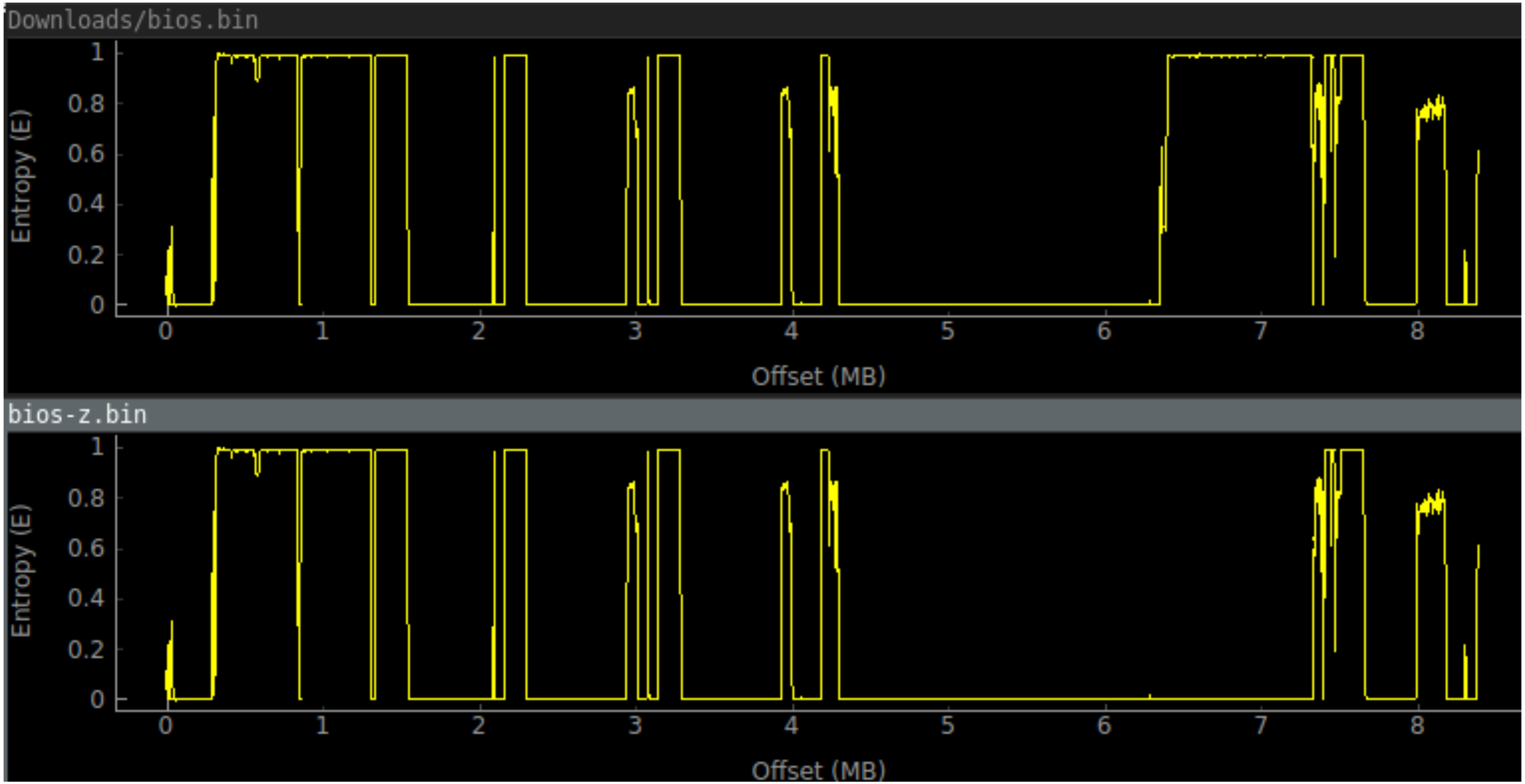
Blanking, wiping, erasing, corrupting, or otherwise invalidating your device's firmware

Flashing Bad Firmware: DEFCON 18 Bootloader

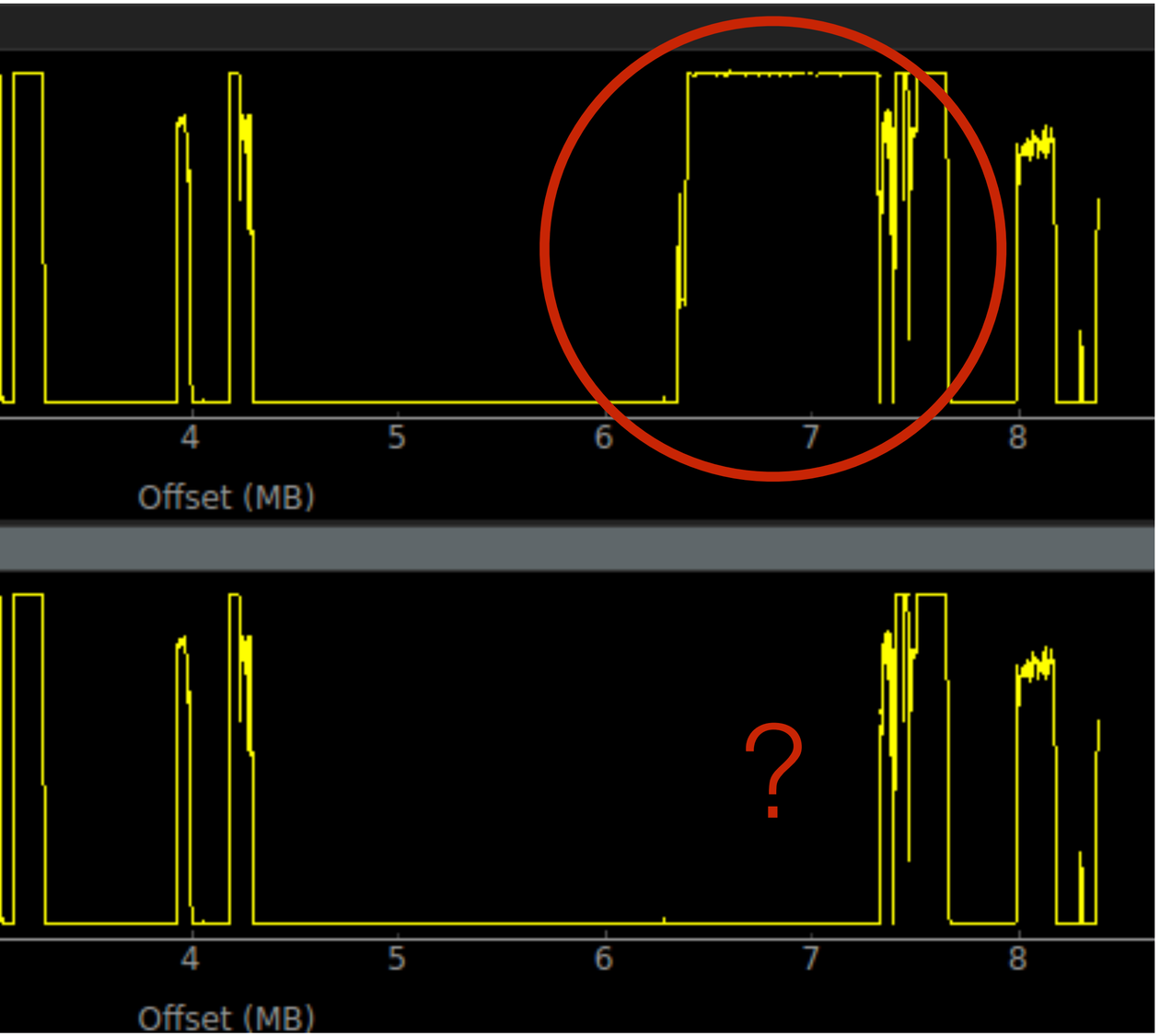
- Bootloader not in protected region
- Screw up during linking can cause bootloader to be overwritten
- Un-bricked through JTAG interface & MC56F8006 development tools



Wiping Critical Sections: Chromebook Firmware



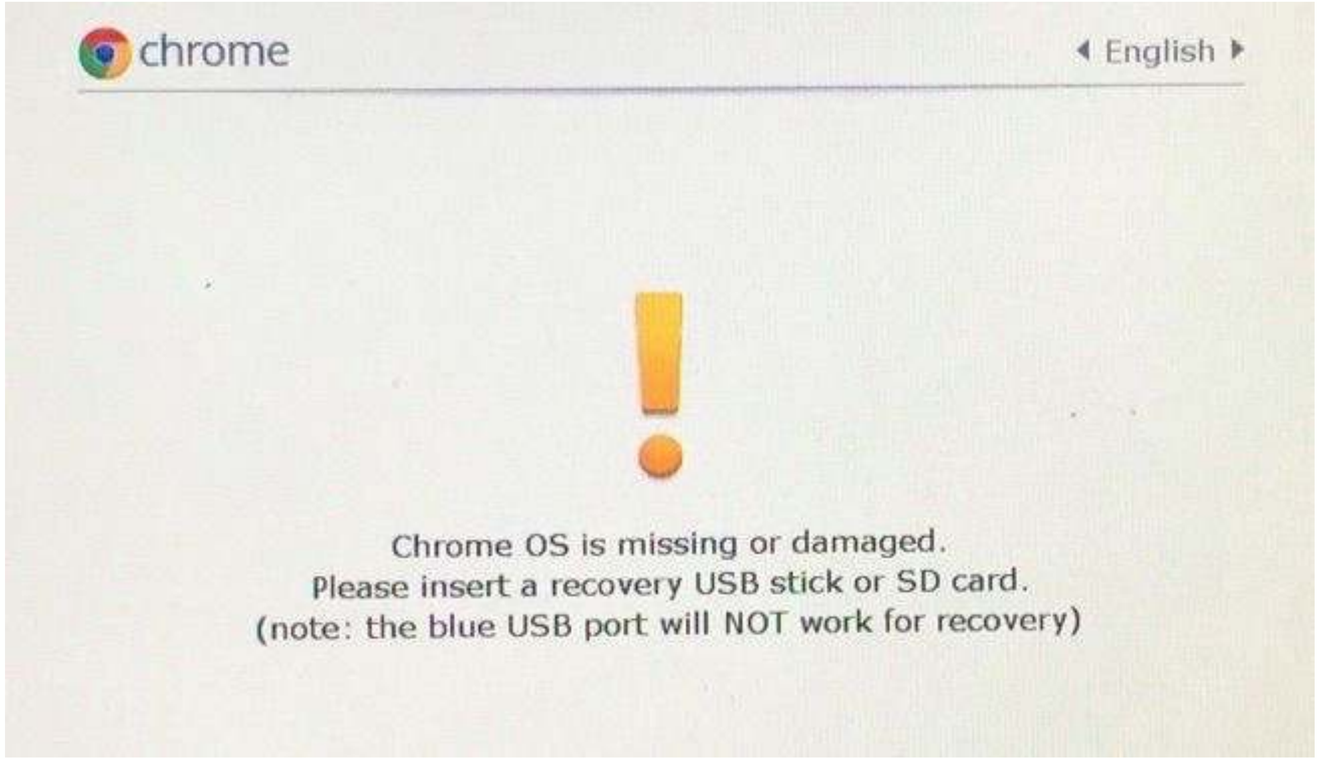
Wiping Critical Sections: Chromebook Firmware



- binwalk's histogram shows entropy in a file
- Top: Physical extraction of BIOS via SPI
- Bottom: Software dump via flashrom
- The two firmwares are different because the CPU blocks access to the ME region for software reads

Touching Signed Filesystems: Acer C720 Chromebook

- Mount R/O filesystem as R/W
- Make changes and reboot
- Kernel verifies rootfs before mounting
- Mismatch causes error



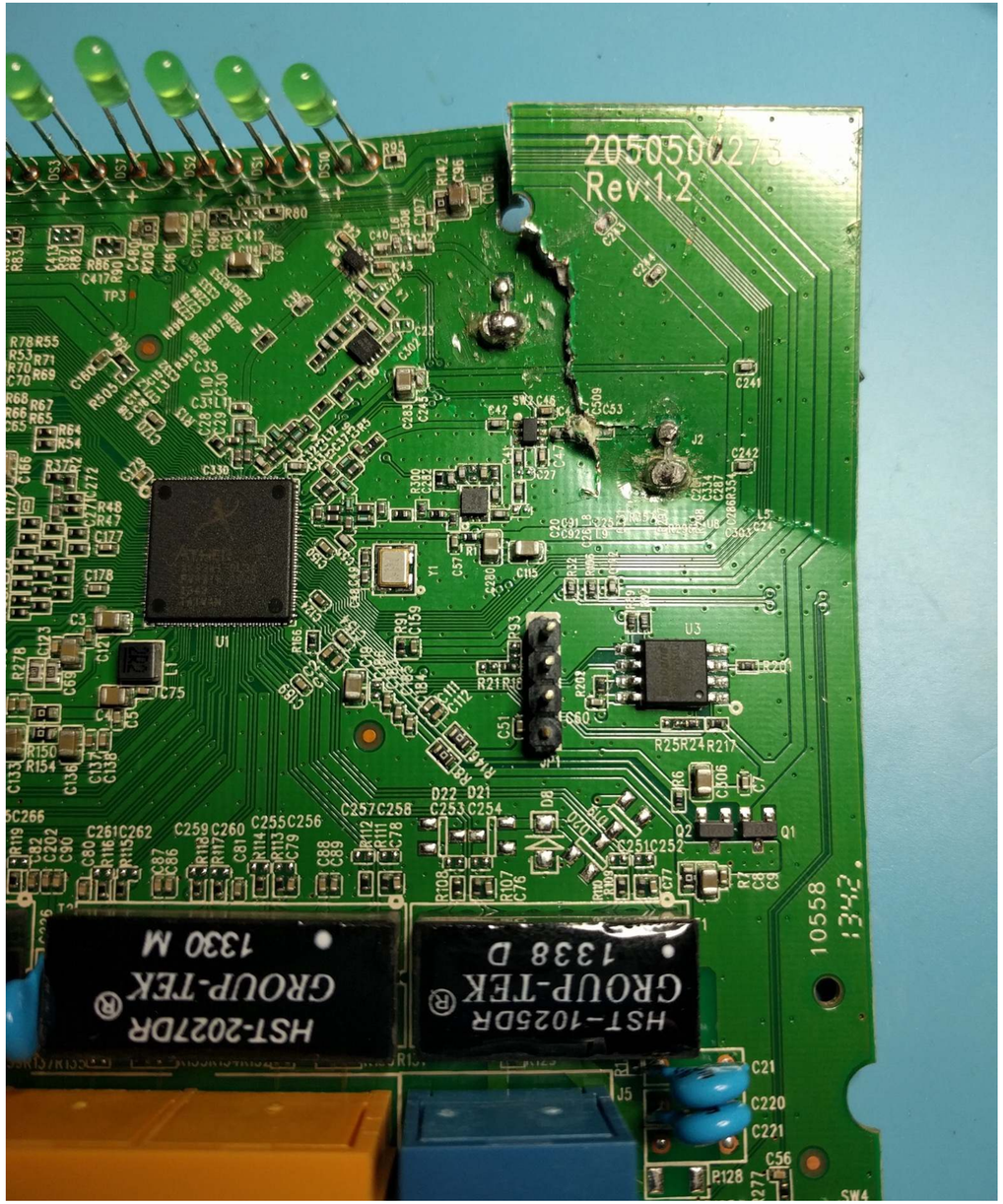
Careless Copying:
DDing the Wrong Partition

```
> sudo dd if=install.iso of=/dev/sda bs=32M
128+0 records in
128+0 records out
4294967295 bytes (4.3 GB, 4.0 GiB copied)
```

- Don't accidentally overwrite your primary media
- This is bad (except when it’s not)

Unbricking your Firmware

- Restore a known good/complete backup
- Directly read/write the storage media
- Recovery/bootloader/download mode
- On-chip program/debug interface (JTAG, ICSP, etc.)
- Swap out physical Flash device

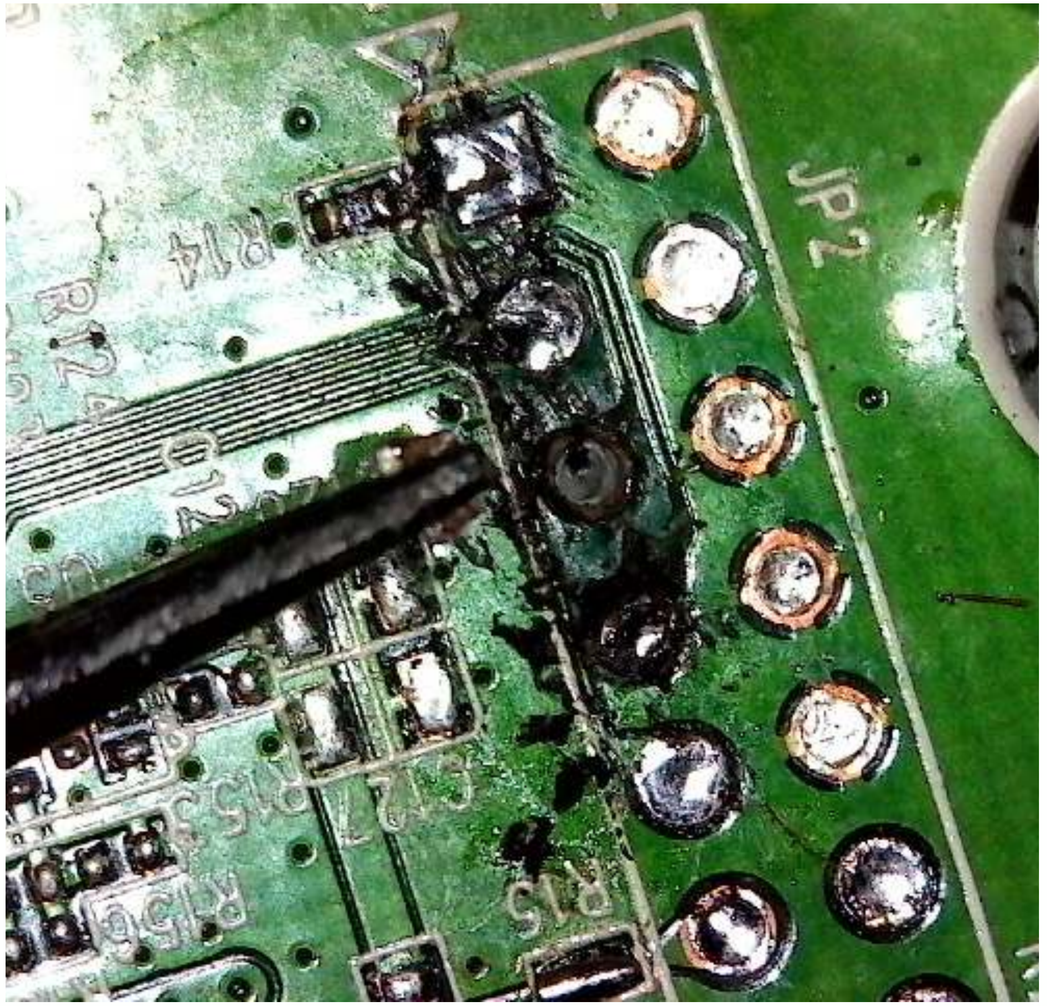


010: Bricking PCBs

Burning, melting, delaminating, shorting and scratching your PCBs and traces

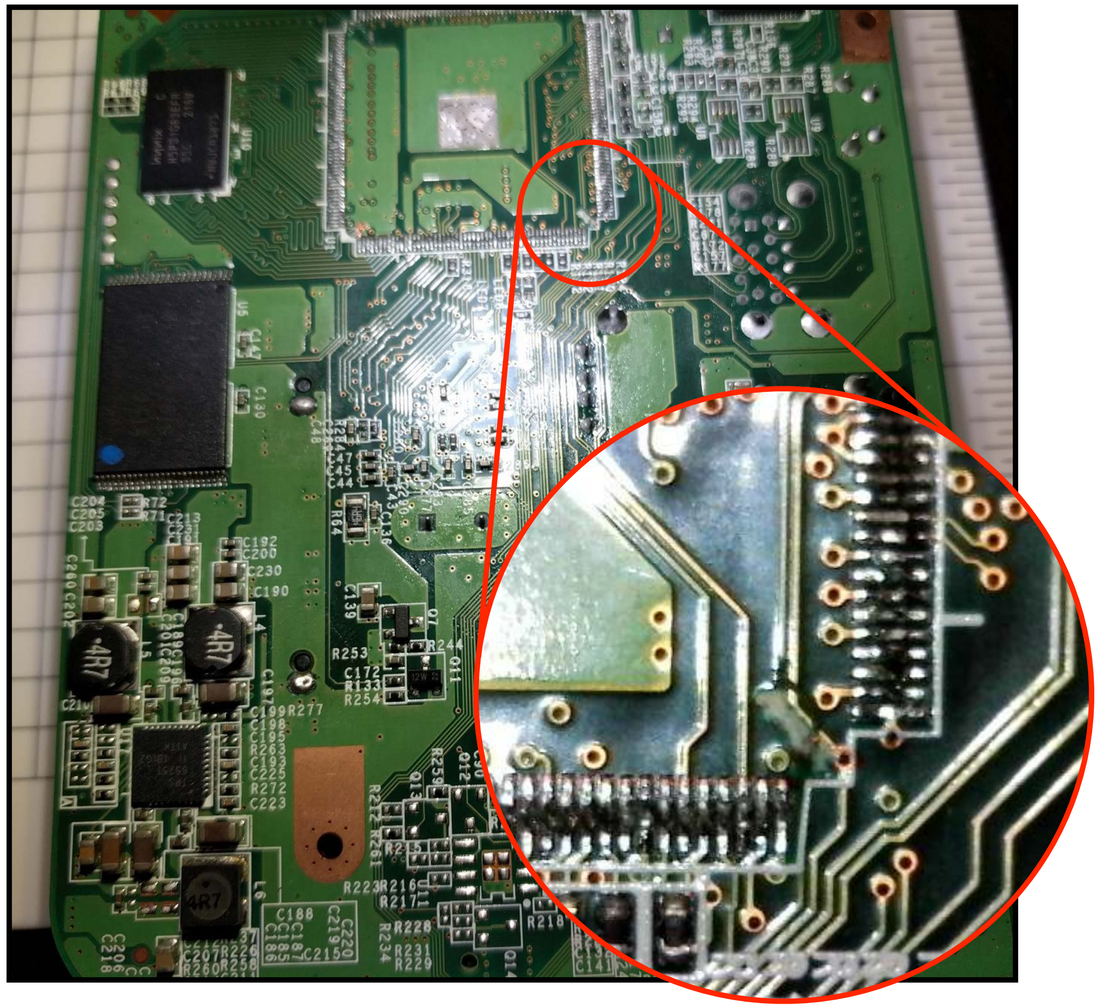
Delaminating Traces: Preparing Debug Headers

- Unpopulated JTAG header's holes were filled with solder
- Too much heat + sloppy work = completely extracted through-hole plating
- Directed heat can eventually cause copper to delaminate from substrate

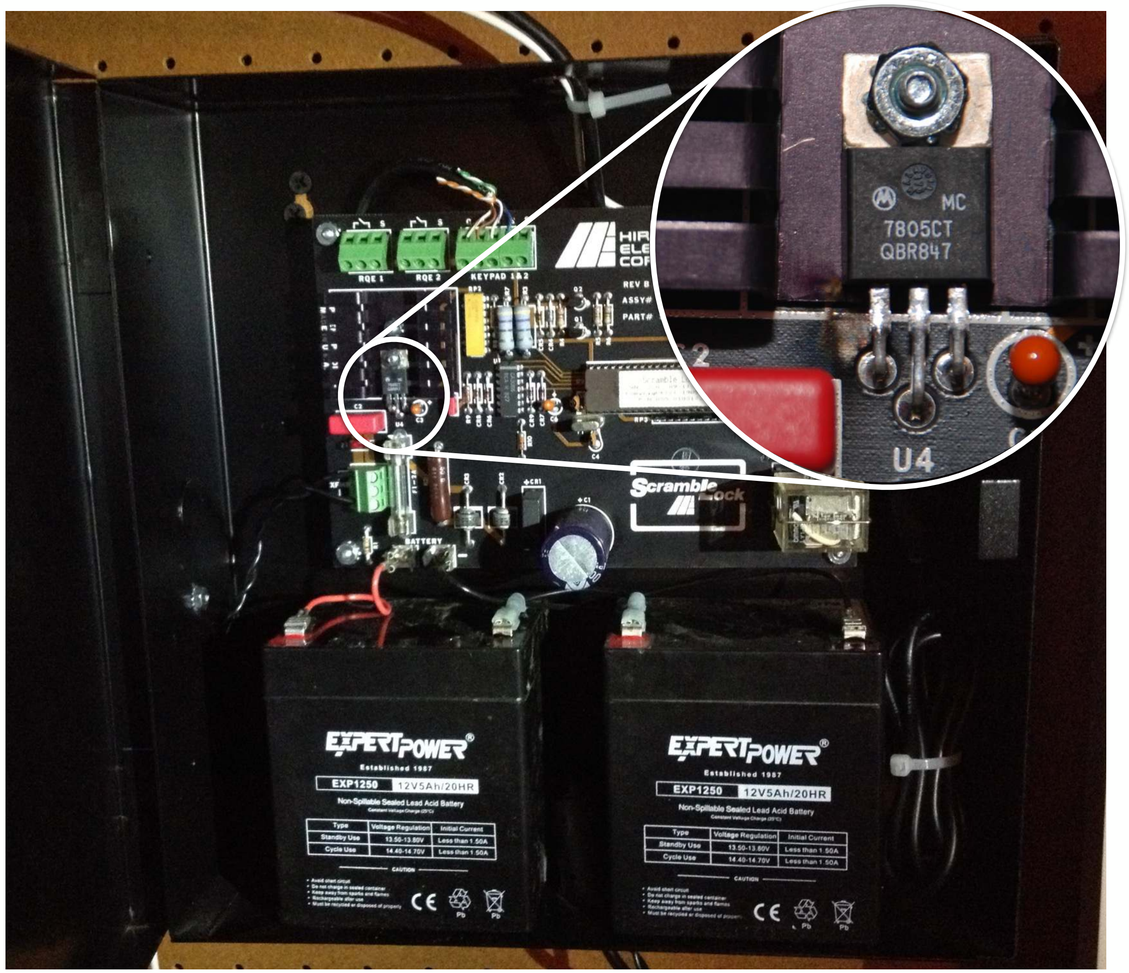


Scratching Traces:
Desoldering CPU on a Pogoplug

- Wanted to remove CPU to follow traces underneath
- Tried lifting part before solder was molten, putting too much pressure on PCB w/ sharp tool
- Damaged traces on board and broke pins on chip, but it was worth it!

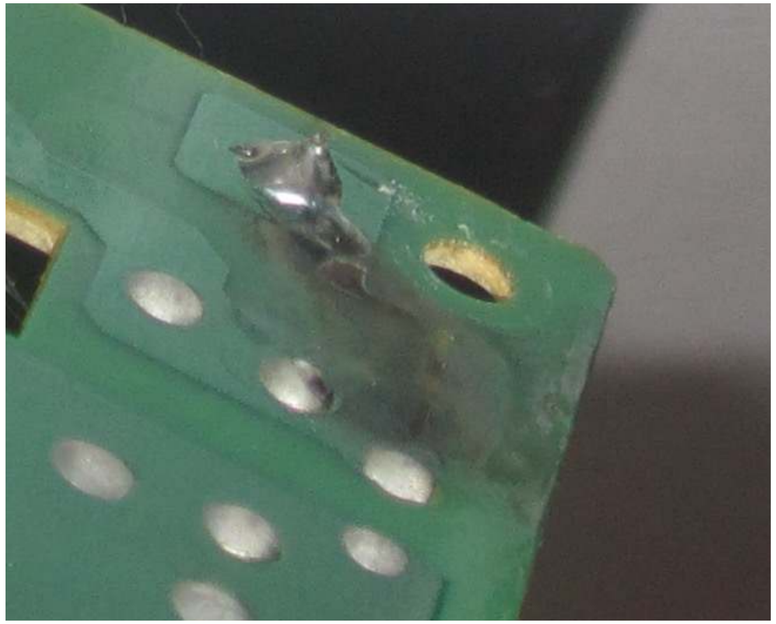


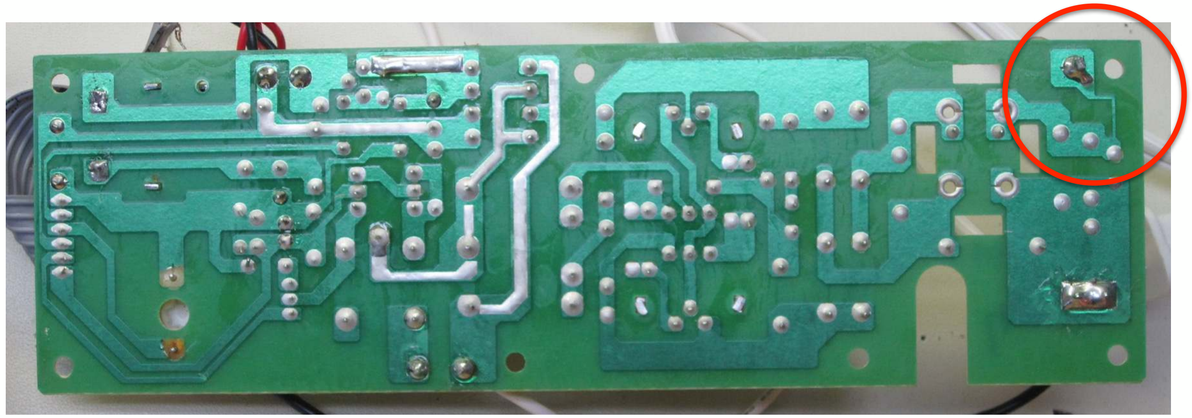
Shorting Traces/Pins:
Hirsch ScramblePad



- Using multimeter to measure input voltage to LM7805
- Probe slipped, shorting input to ground
- Spark, burned board, bruised ego

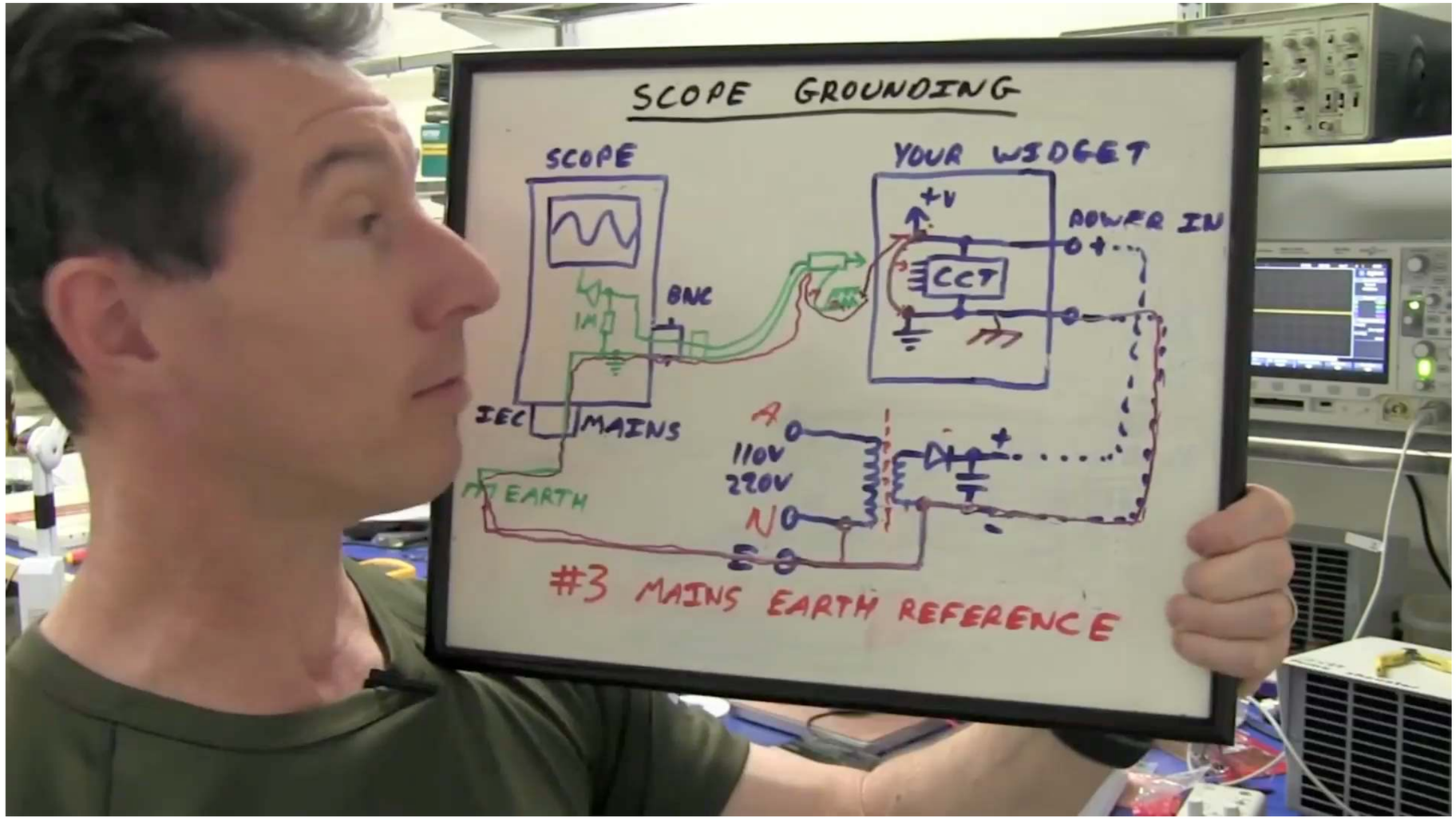
Burning Traces: FoodSaver V850





- Improper connection of oscilloscope ground
- Tried to measure an AC signal
- Blew trace that served as a low-cost fuse
- Thankfully oscilloscope not damaged!

Burning Traces:
FoodSaver V850



Unbricking your PCBs

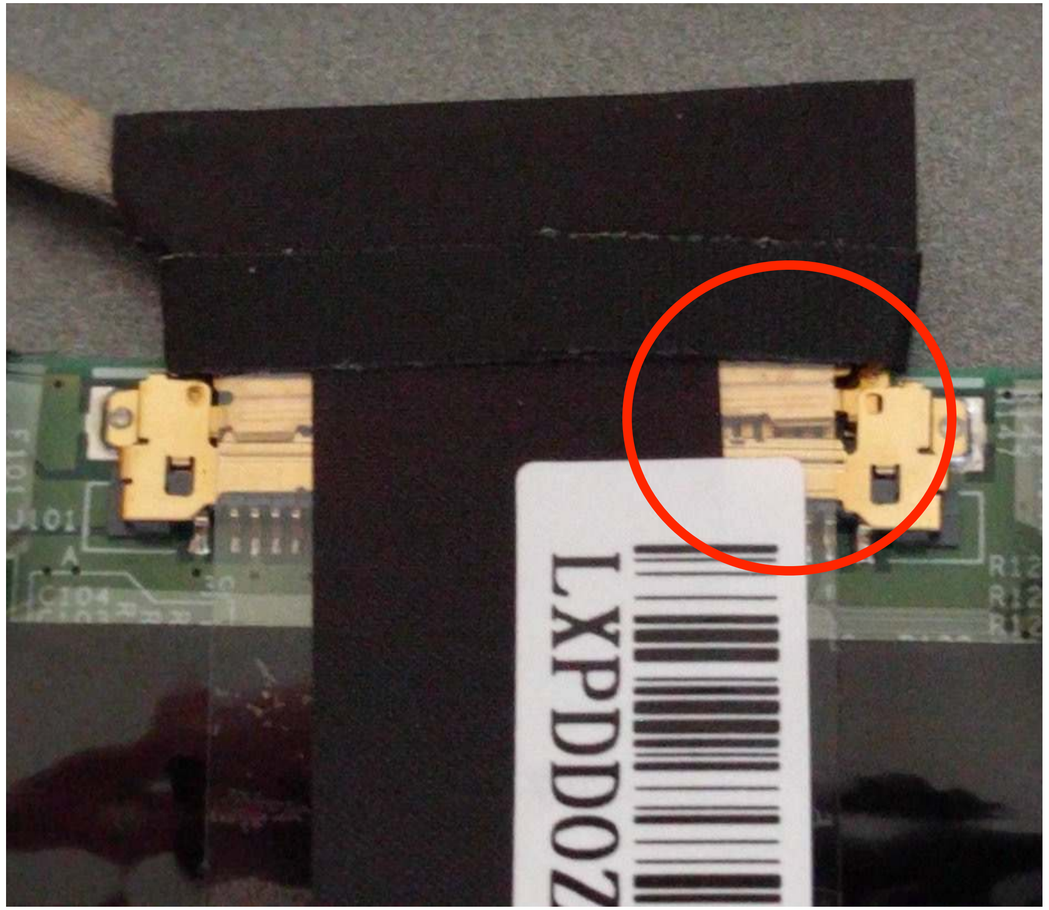
- Careful soldering to repair and/or replace
- Blue wires
- Epoxy and adhesives
- Patience

011: Bricking Connectors

Damaging power plugs, breaking solder joints, crushing internal connectors, and severing internal cabling

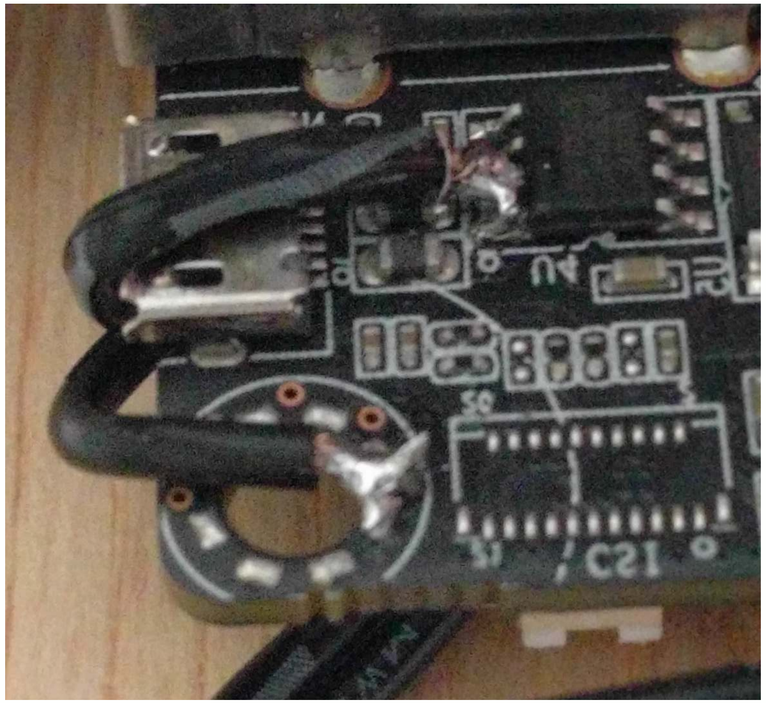
Loose Connectors:
Chromebook C720 Display

- Taut cable routing causes LCD connector to loosen over time
- 9 out of 10 ‘DOA’ C720’s were fixed by adjusting this cable and re-taping
- Sometimes normal use can brick your hardware

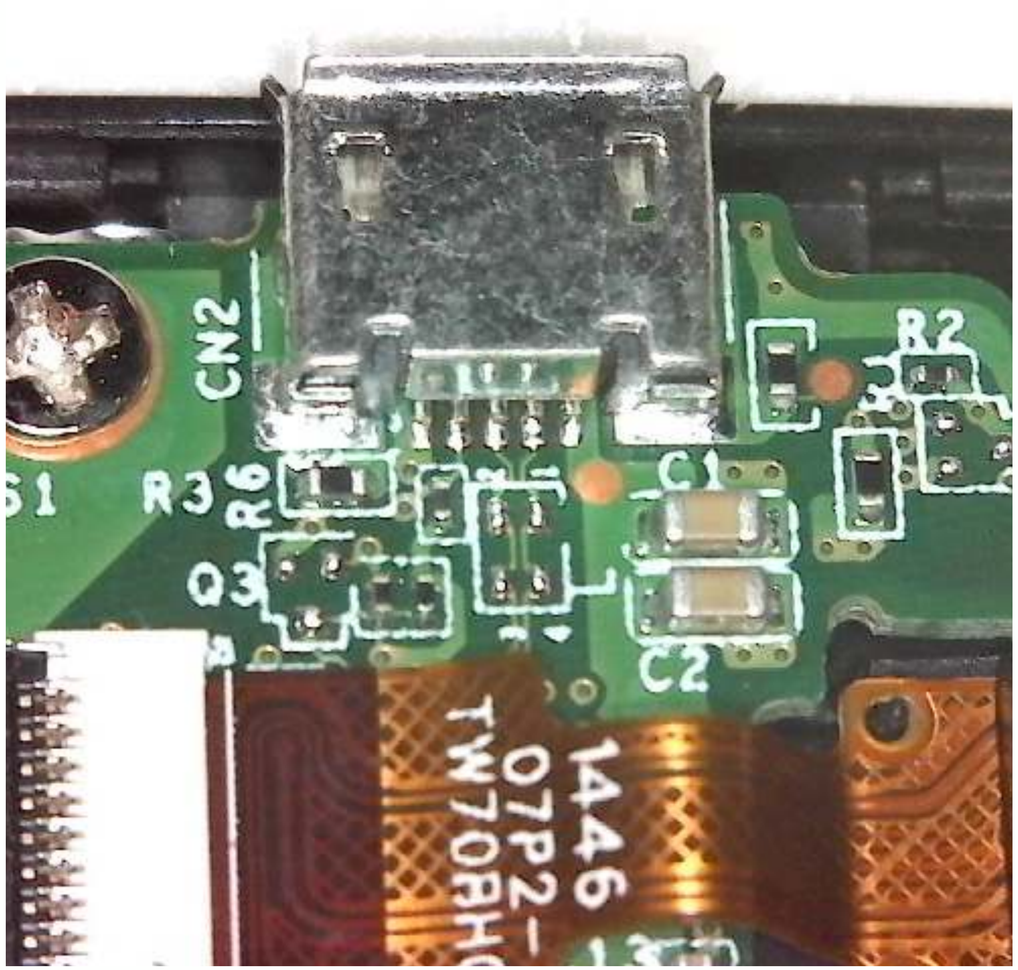


Misused Connectors:
ECS Liva Mini PC

- Micro USB connector used for power input
- Traces are not well sized for required current (3A), thermal regulation is not well controlled
- At high CPU utilization, the PCB overheats, deforms the connector, disconnects power



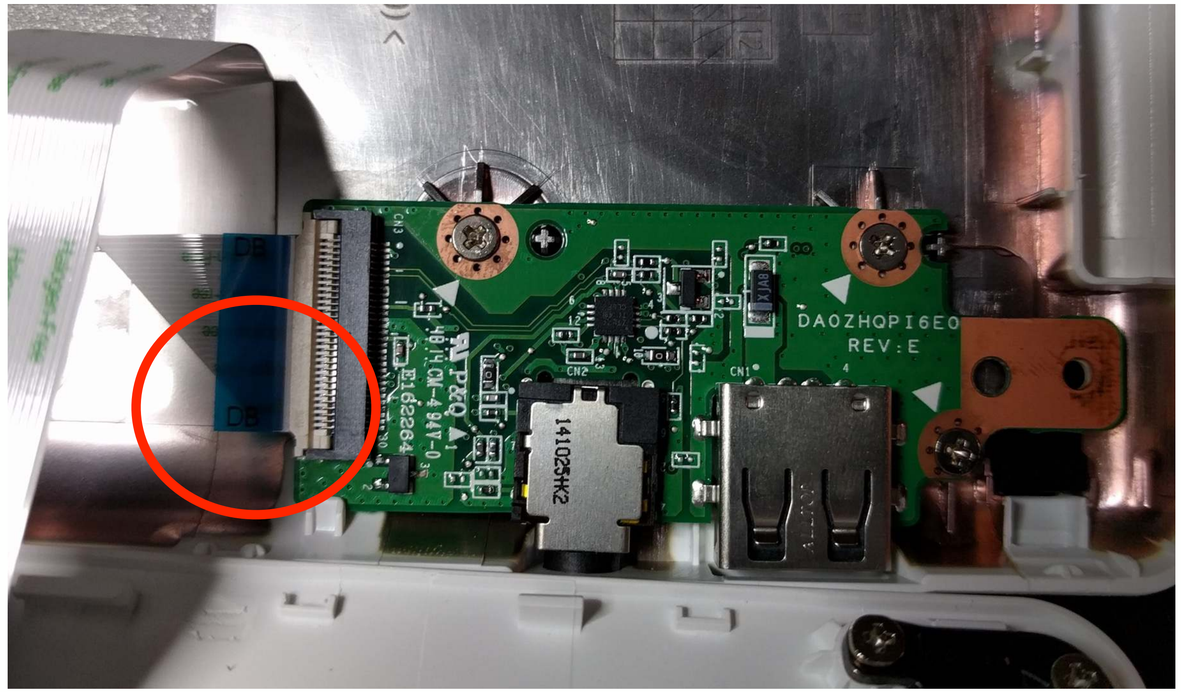
Breaking Solder Joints:
TW700 Tablets



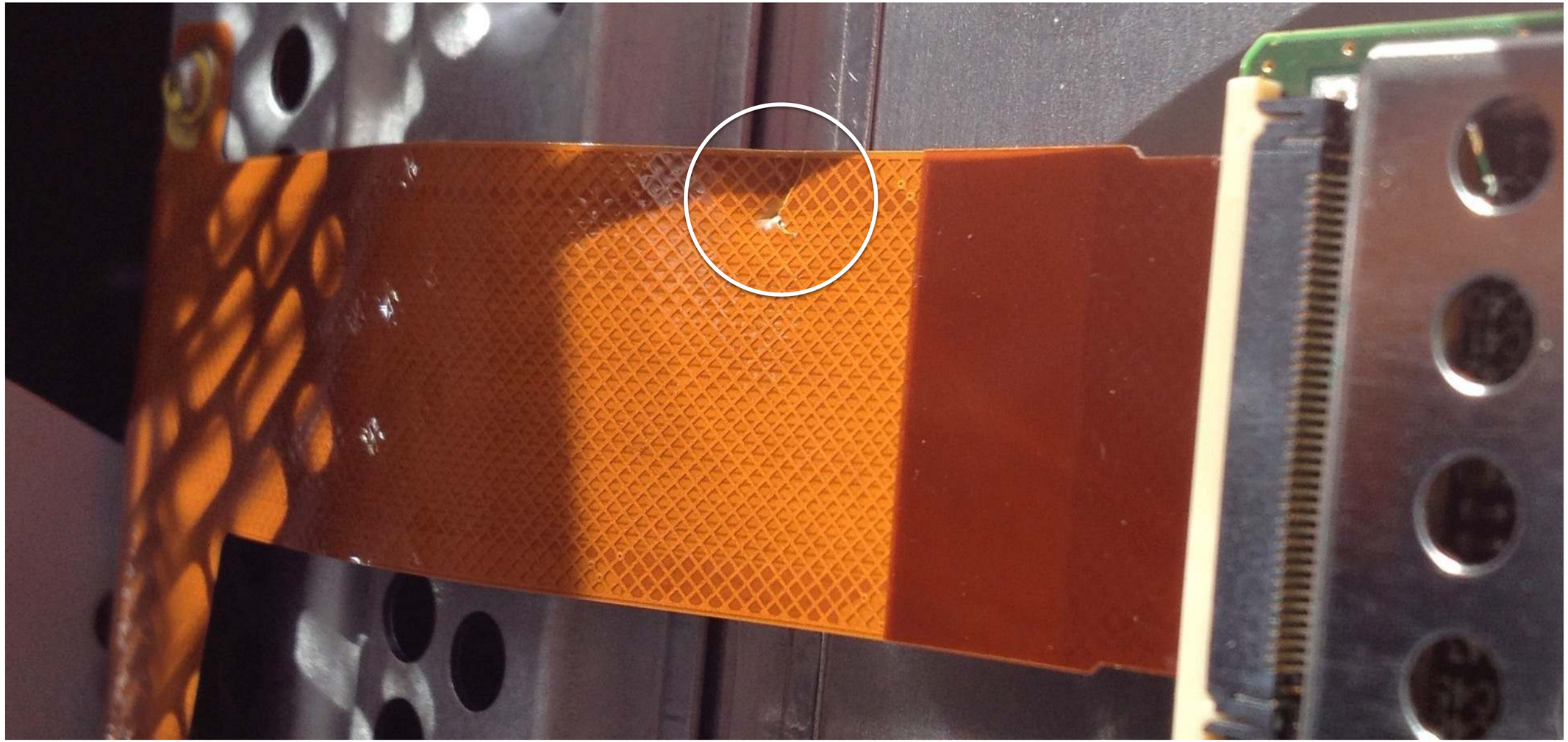
- Micro USB connector used for power/charging input
- Tablet case cutout is not snug around the connector
- Wiggling the cable moved the connector and broke solder joints
- Surface mount connectors have poor mechanical stability, solder is not designed to handle mechanical stress

Slicing Internal Cables:
Low-Cost Consumer Device

- Acer CB3 has USB & audio running over FPC (Flexible Printed Circuit)
- FPC connects between circuit boards on each side of the clamshell
- Opening the case without knowing this either disconnects cable (good) or causes cable to kink & tear (bad)



Slicing Internal Cables:
High-Cost Consumer Device



Unbricking your Connectors

- Mechanical reinforcement (e.g., tape, epoxy, not solder)
- Electrical reinforcement (e.g., upgraded wiring, more solder)
- Know how to measure & locate replacements
- Know how to read mechanical drawings
- Digi-Key is your friend

Absolute Maximum Stress Ratings (Applied conditions greater than those listed under "Absolute Maximum Stress Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these conditions or conditions greater than those defined in the operational sections of this data sheet is not implied. Exposure to absolute maximum stress rating conditions may affect device reliability.)

Temperature Under Bias . . . . . -55°C to +125°C
Storage Temperature. . . . . -65°C to +150°C
D. C. Voltage on Any Pin to Ground Potential . . . . . -0.5V to V_{DD}+0.5V
Transient Voltage (<20 ns) on Any Pin to Ground Potential . . . . . -2.0V to V_{DD}+2.0V
Package Power Dissipation Capability (T_A = 25°C). 1.0W
Surface Mount Solder Reflow Temperature . . . . . 260°C for 10 seconds
Output Short Circuit Current[1] . . . . . 50 mA

1. Output shorted for no more than one second. No more than one output shorted at a time.

Table 8: Operating Range

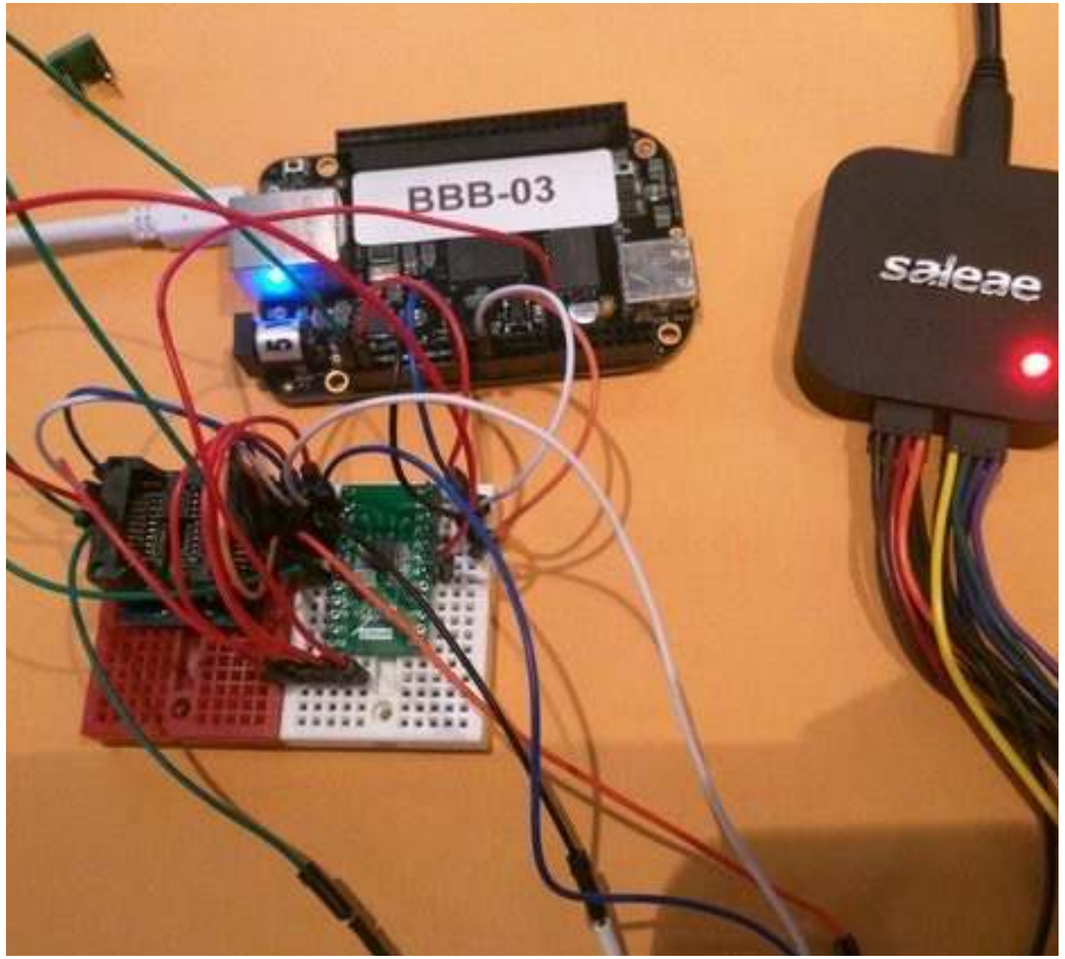
Range	Ambient Temp	V_{DD}
Commercial	0°C to +70°C	2.7-3.6V
Industrial	-40°C to +85°C	2.7-3.6V

100: Bricking ICs

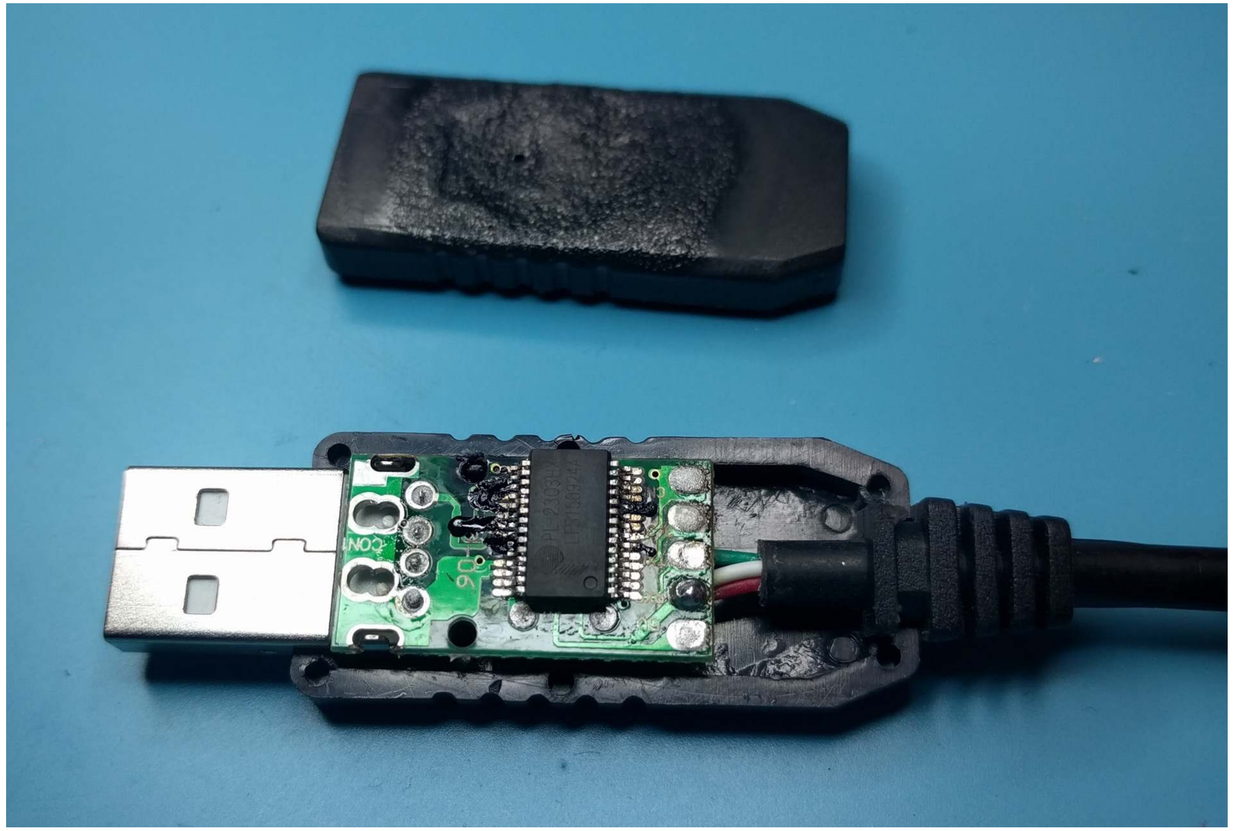
Exceeding the Absolute Maximum ratings and letting out the magic smoke

Applying Too Much Voltage: Teclast X98 1.8V SPI Flash

- Intel Bay Trail chipsets use 1.8V SPI Flash chips to store BIOS
- Many common HW tools are 3.3V or 5V
- Overvoltage could corrupt memory contents, damage chips
- Use a level shifter to bring signal voltages within allowable range



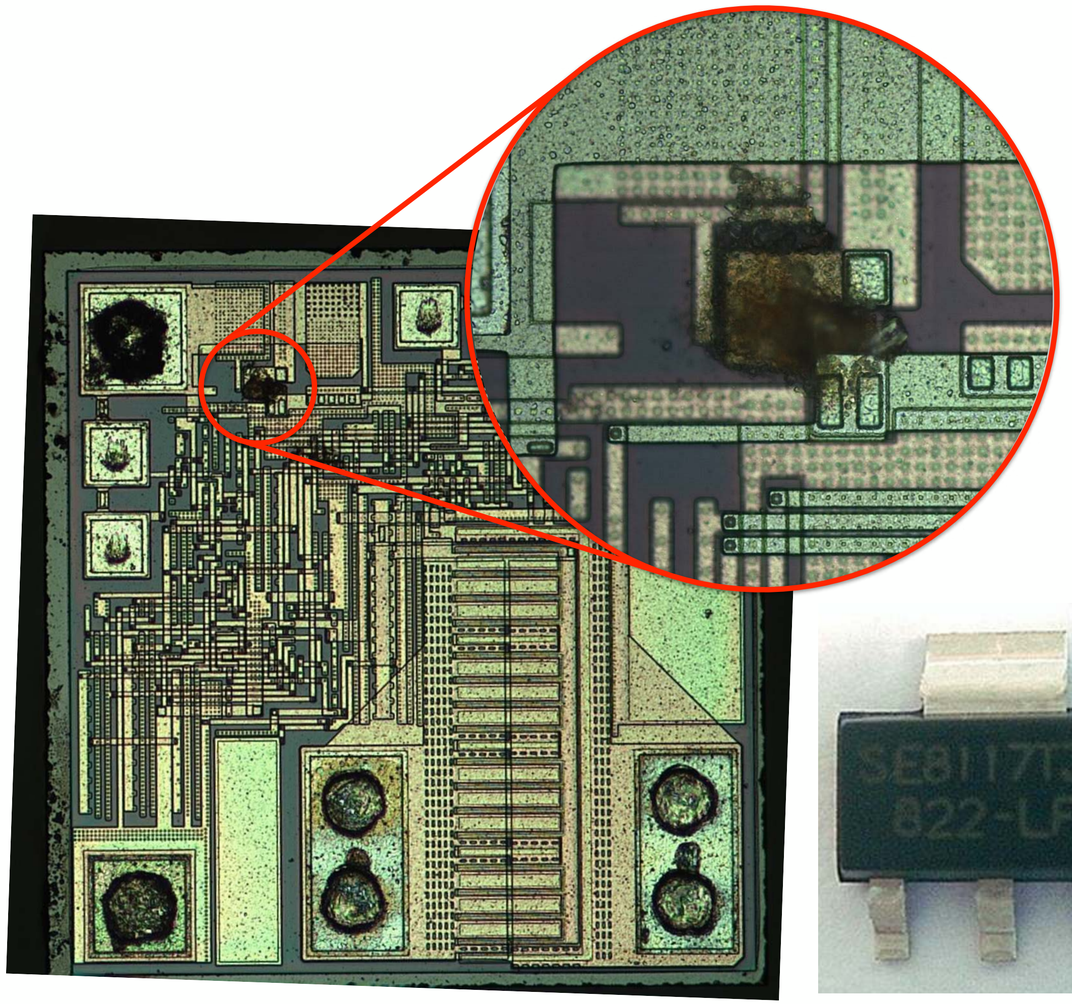
Pulling Too Much Current: Serial-to-USB Devices



- Serial-to-USB device using counterfeit Prolific PL2303
- Poor build quality caused overcurrent condition that wasn't detected by host USB port
- Case melted, PCB damaged, component fried

Pulling Too Much Current: Seaward SE8117T33 LDO Regulator

- Used in power supply circuitry of pre-production consumer device
- Die analysis reveals burned output driver caused by over current to the tab



Unbricking your ICs

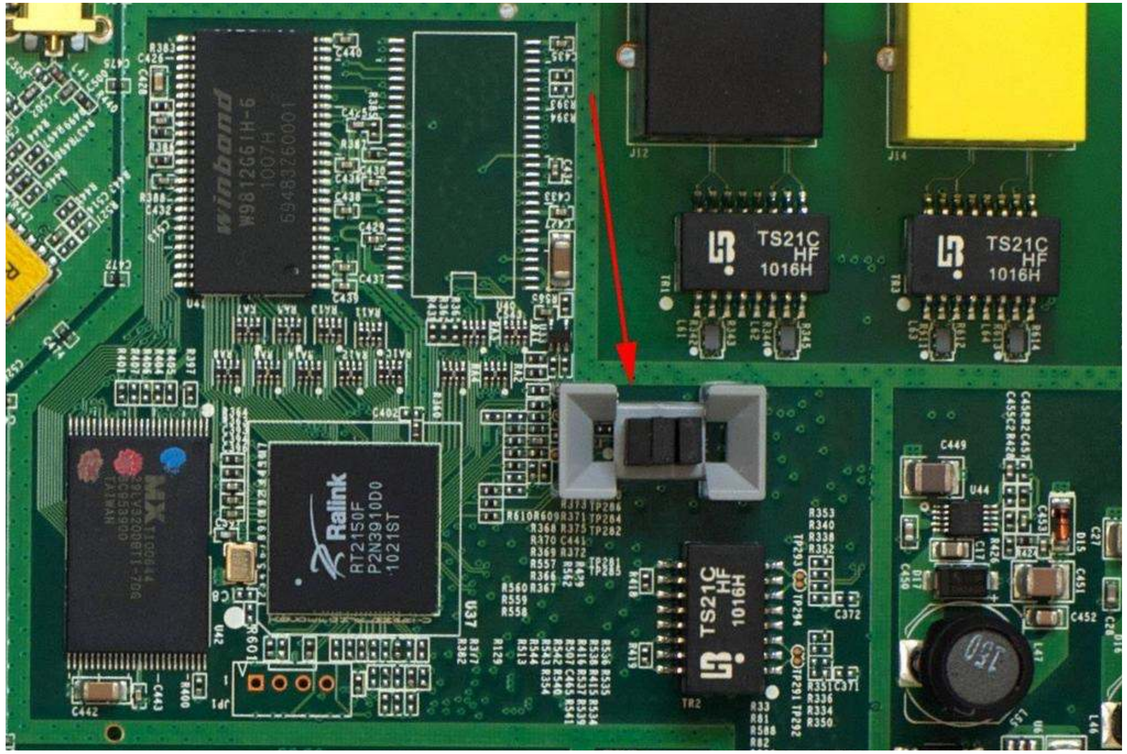
- Replace the chip
- Fix your board/connection issues first or you'll have two fried chips
- Digi-Key is still your friend

101: Bricking ‘WTF’ Scenarios

When environmental conditions and physical factors gang up against your devices

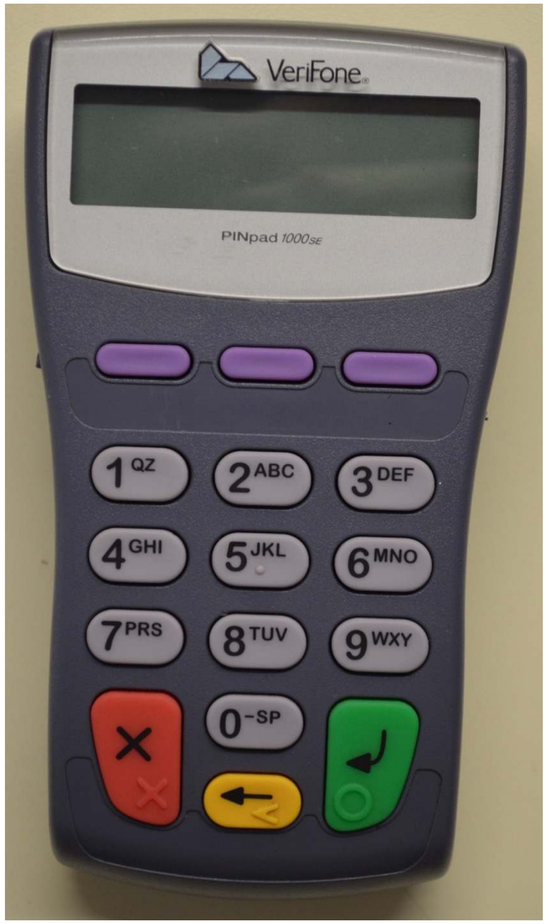
Anti-Tamper Mechanisms: AT&T Microcell

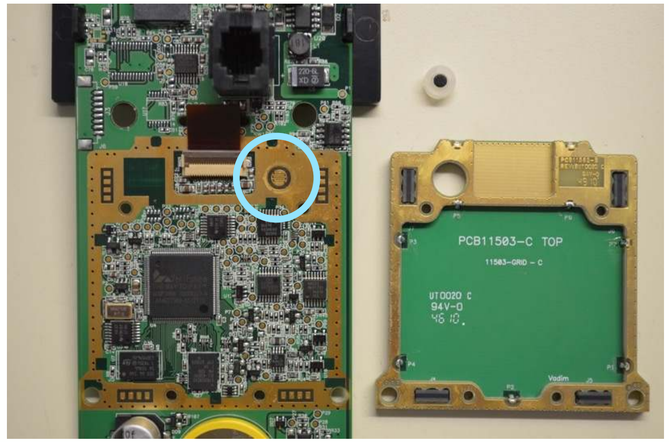
- 2x3 male headers w/ 3 jumpers each
- Jumpers are tethered to both sides of case, get pulled out when opened
- When powered up, sets tamper flag and phones home

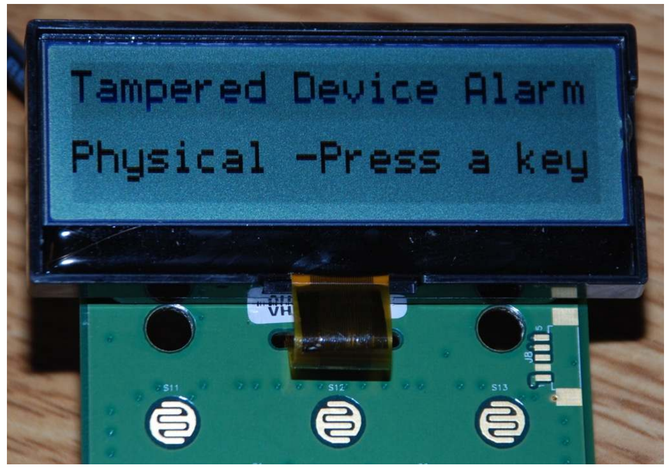


Anti-Tamper Mechanisms: VeriFone PINpad 1000SE

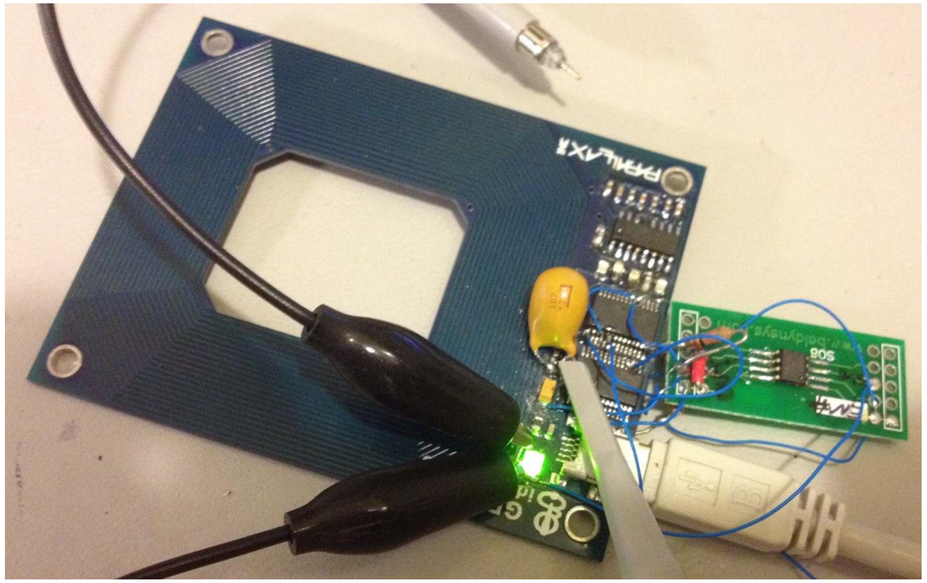
- Multiple mechanisms to detect physical intrusion (switch, active mesh PCB)
- Tamper event erases encryption keys from battery-backed RAM
- Requires special process/sequence to re-key/re-enable

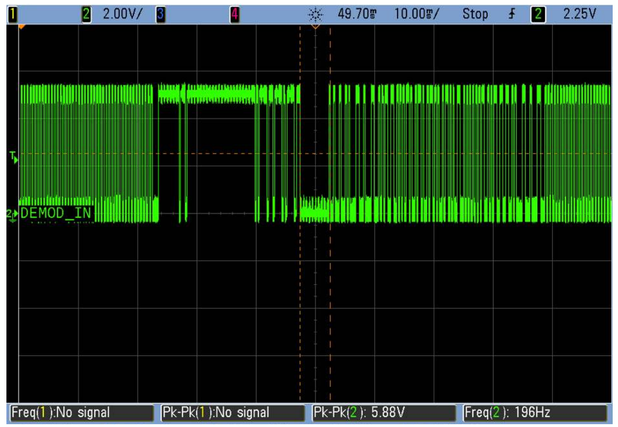






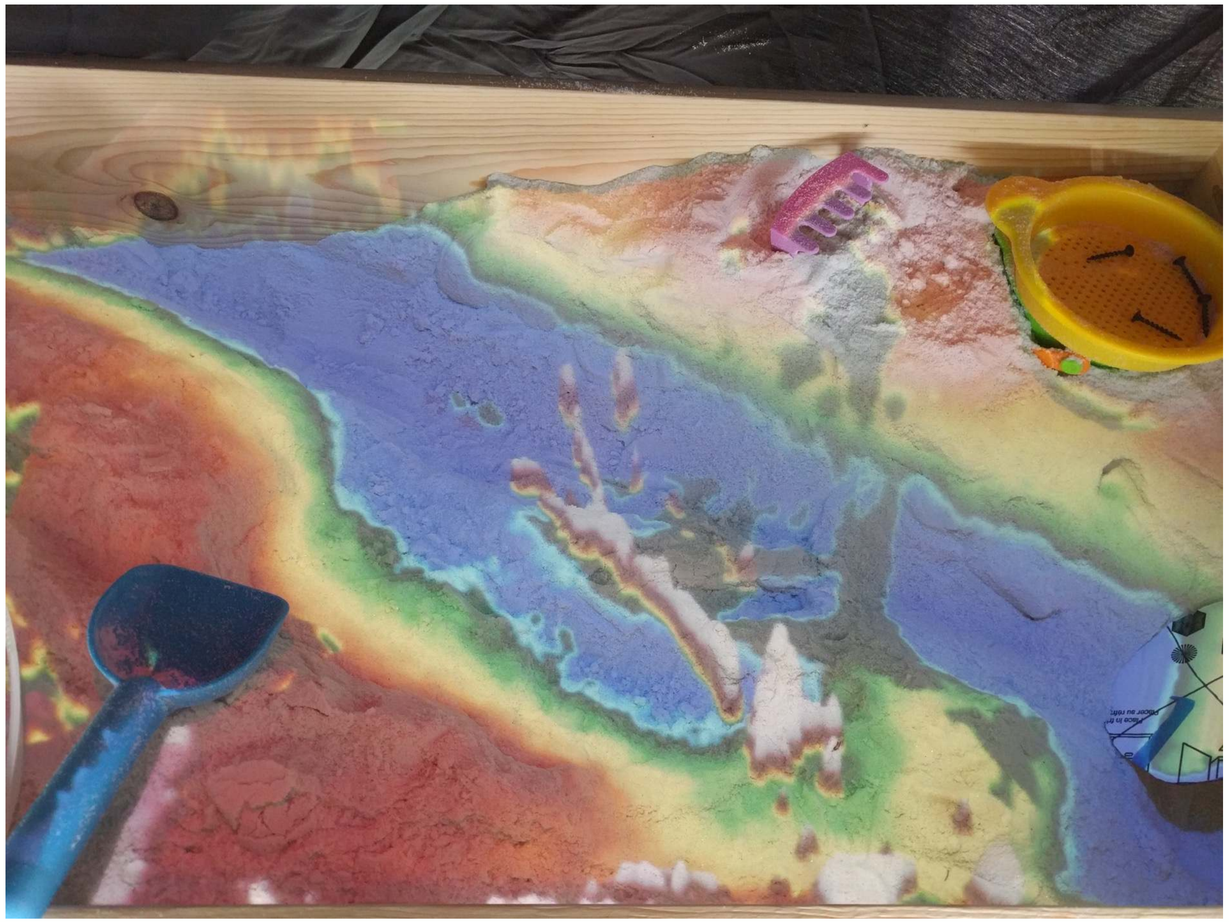
Environmental Conditions:
Parallax RFID R/W USB Module





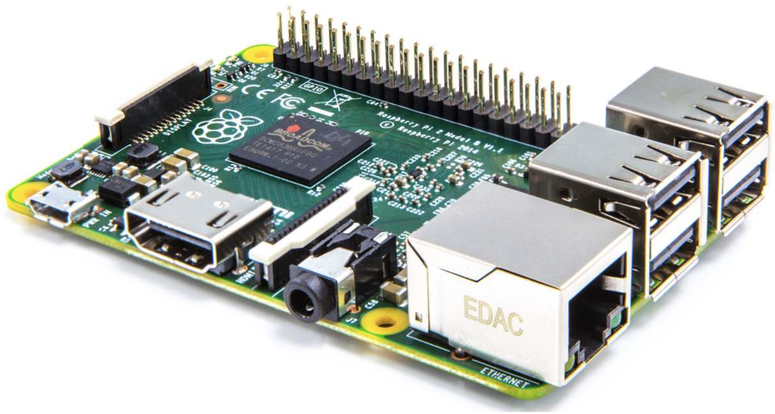
- Antenna sensitivity too high
- Received noise from environment and unclean USB power
- Demodulated noise into digital data
- Years of anguish
- Single capacitor value change solved problem

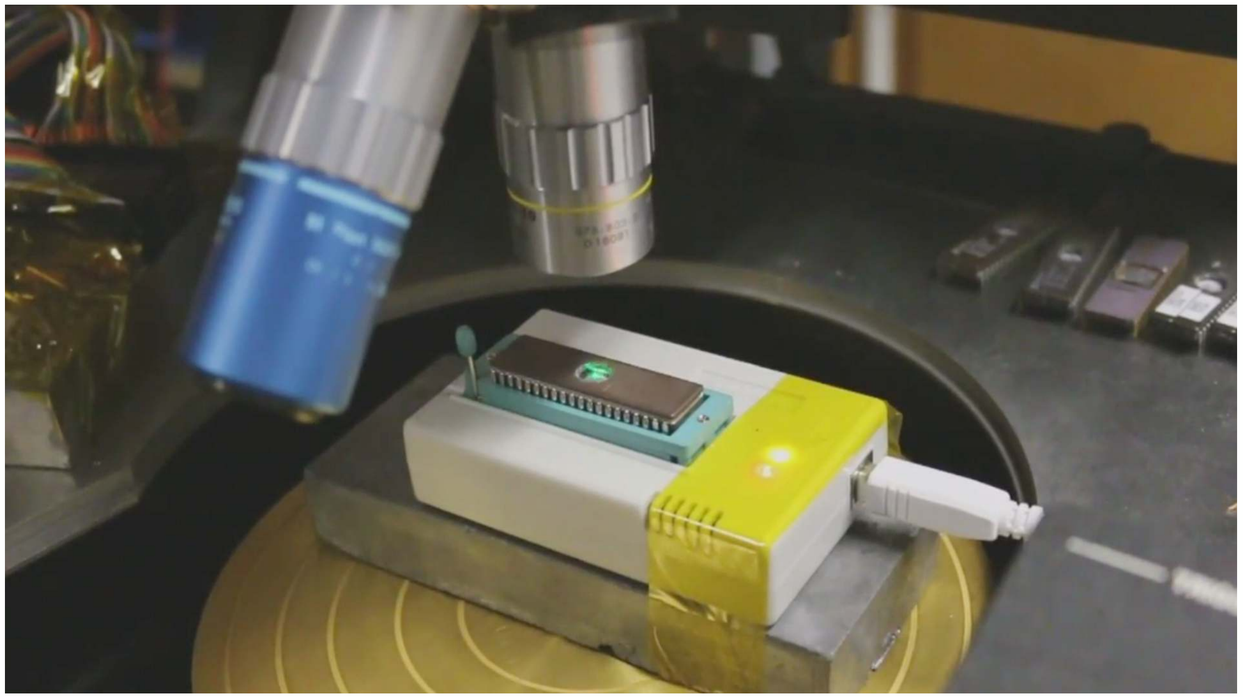
Environmental Conditions:
AR Sandbox Kinect



- Kinect uses IR light to generate a pattern
- IR light from sun interferes with pattern, so Kinect doesn’t work in daylight
- Putting a black sheet over sandbox helps block indirect light, but casts a deceiving pattern resulting in strange behavior

Environmental Conditions: Optical Glitching





- Most silicon is light sensitive and can be subject to the photoelectric effect
- Photoelectrons can intentionally or unintentionally change behavior of IC
- Not a problem when they're encapsulated in opaque package
- Raspberry Pi 2: Camera flash caused power regulator to glitch and reset
- Hirsch ScrambleLock: Camera flash caused MCU to lock up, requiring physical reset

Unbricking WTF Scenarios

- You might not know what you did!
- Get another piece of hardware and be careful this time
- Get another piece of hardware and manually ‘diff’
- Grab a bite to eat or take a nap. Maybe it’ll just work later?

The Best Ways to Brick?

- 001: Bricking Firmware
  -> Wipe your flash
- 010: Bricking PCBs
  -> Cut your traces
- 011: Bricking Connectors
  -> Smash your connectors
- 100: Bricking ICs
  -> Apply the wrong voltage
- 101: Bricking ‘WTF’ scenarios
  -> Work on anything last minute

The Best Ways to Avoid Brick?

- 001: Bricking Firmware
  -> Back up your firmware!
- 010: Bricking PCBs
  -> Plenty of workspace & protective measures
- 011: Bricking Connectors
  -> Patience and the right tools
- 100: Bricking ICs
  -> Double check pinouts and voltages (RTFM!)
- 101: Bricking ‘WTF’ scenarios
  -> Have a predictable workbench setup

The Best Ways to Unbrick?

- 001: Bricking Firmware
  -> Restore your backup
- 010: Bricking PCBs
  -> Soldering skills
- 011: Bricking Connectors
  -> Digi-Key is your friend
- 100: Bricking ICs
  -> Digi-Key is still your friend
- 101: Bricking ‘WTF’ scenarios
  -> Don’t hack what you can’t afford to lose!

Benefits of the Brick

- Sacrificial brick
- Learn from your mistakes (hopefully at someone else's expense)
- Share your mistakes so others can avoid them

Questions?

- Apparently you can make a whole presentation about bricking
- Thanks for watching!