# Zephyr® Project

## Developer Summit

Zephyr® Project
Developer Summit

# Analyze USB Traffic with Wireshark

Tomasz Moń, Nordic Semiconductor Poland Sp. z o.o.

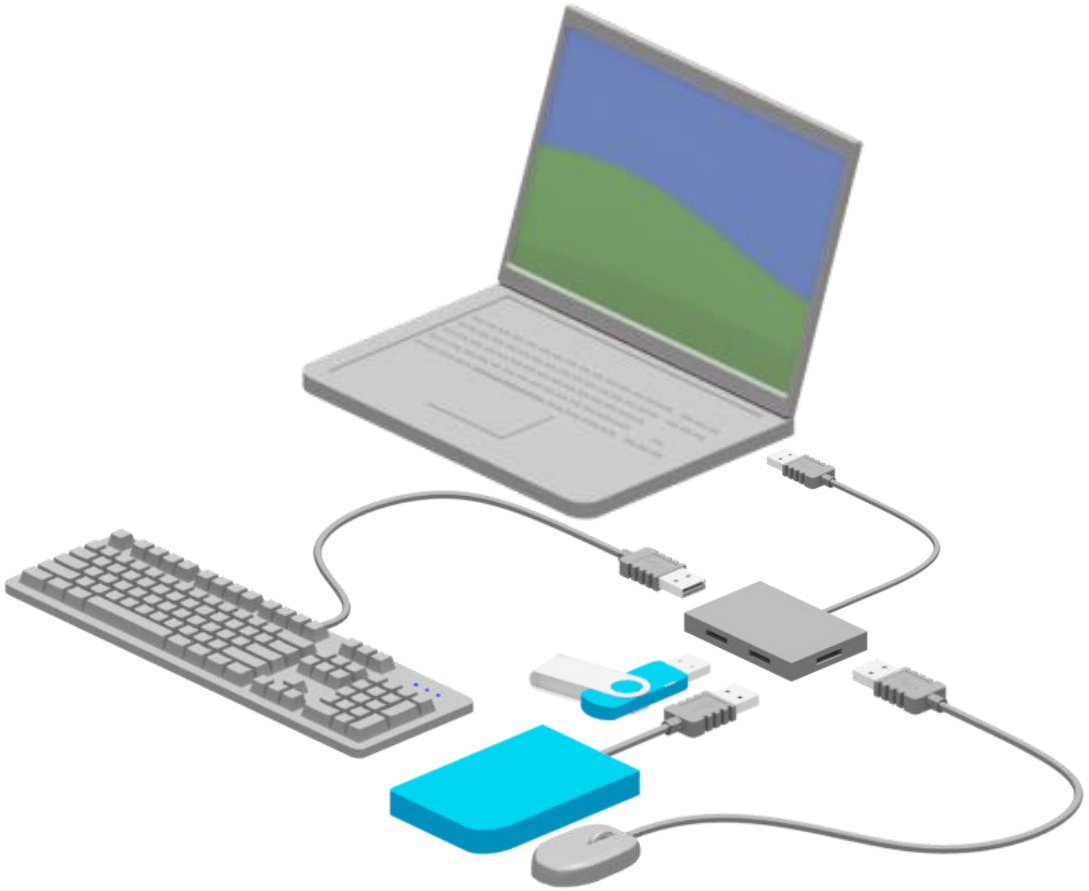
tomasz.mon@nordicsemi.no

#EMBEDDEDOSSUMMIT

# Table of contents

# Basic USB terminology

| USB | Analogous to |
|---|---|
| Host | Requester, DHCP server |
| Device | Responder |
| Port | Physical port connector |
| Hub | Switch, Hub |
| Address | Local IP address |
| Endpoint | Buffer, TCP/UDP Port |
| Class | Communication Protocol |
| Descriptor | Datasheet |
| VID | Vendor code |
| PID | Product code |



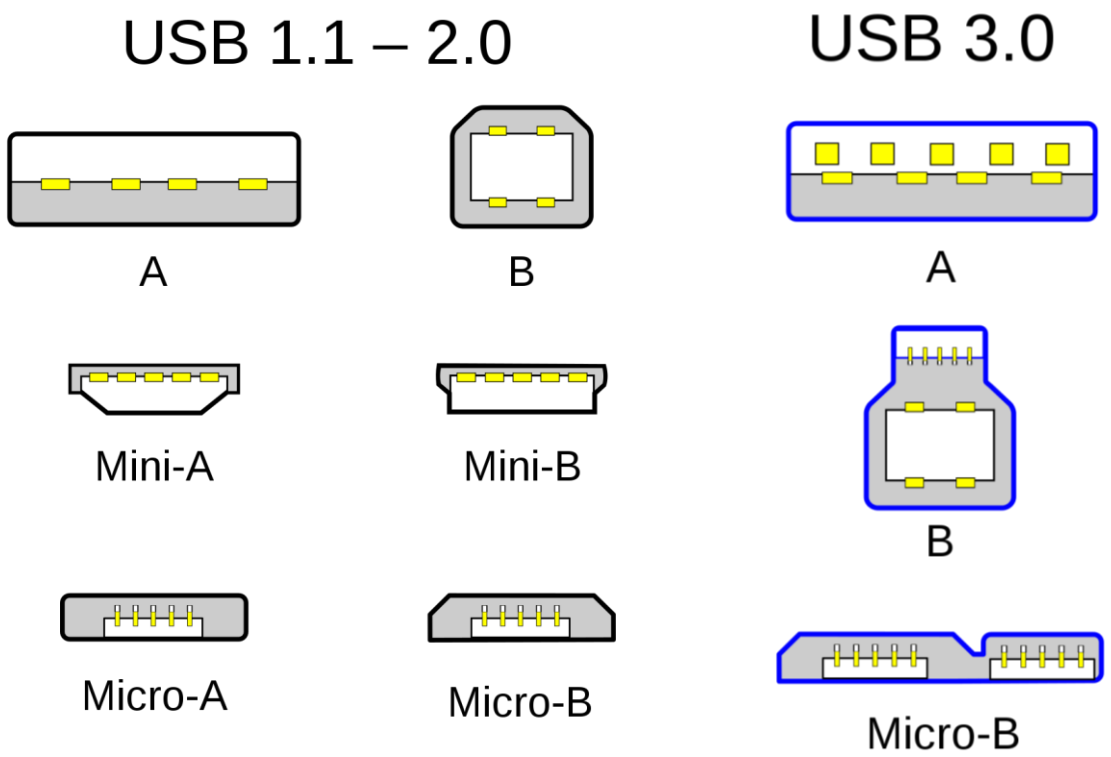
# USB connectors

| Pin | Mini/Micro Pin | Name |
|---|---|---|
| 1 | 1 | VBUS |
| 2 | 2 | D– |
| 3 | 3 | D+ |
| N/A | 4 | ID |
| 4 | 5 | GND |

**Because there is merely a single differential pair in USB 2.0, only Half-Duplex communication is possible**

| | | |
|---|---|---|
| 5 | 6 | SSTx- |
| 6 | 7 | SSTx+ |
| 7 | 8 | GND |
| 8 | 9 | SSRx- |
| 9 | 10 | SSRx+ |

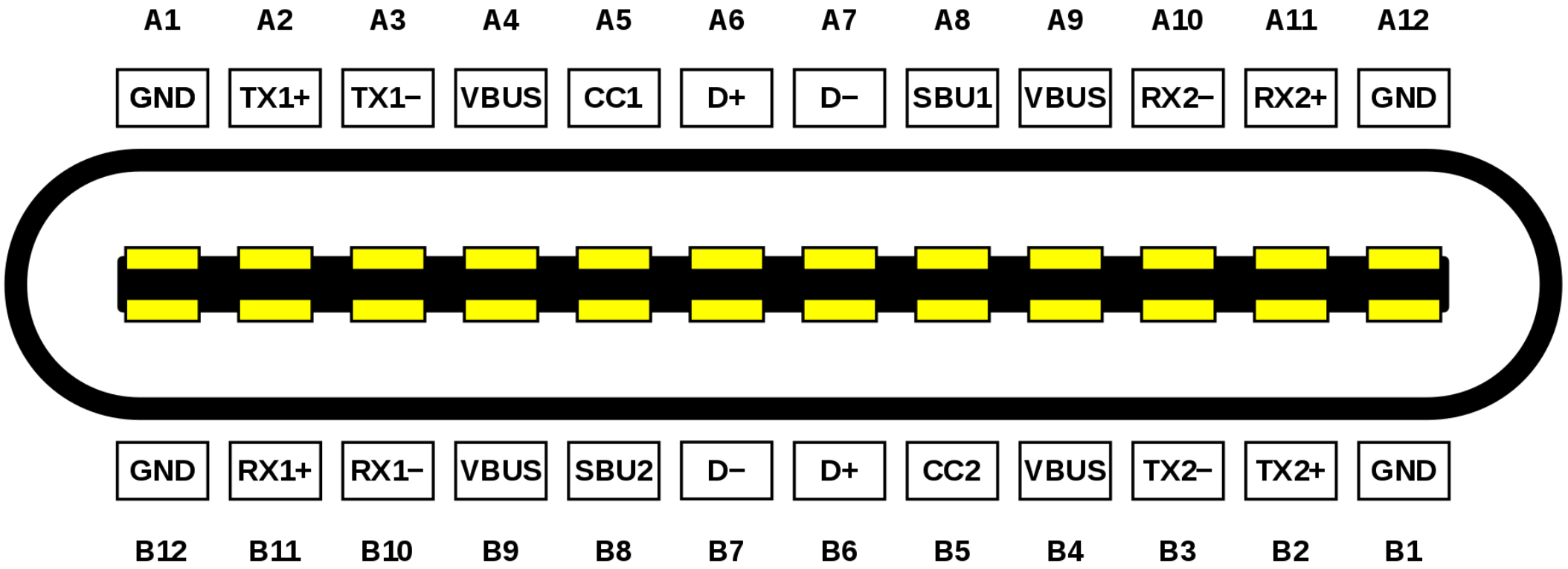
**USB 3.0 is dual-simplex**



# USB speeds

- USB 2.0 features three transmission speeds:
  - Low speed (1.5 Mbps)
  - Full speed (12 Mbps)
  - High speed (480 Mbps) (**Hi-Speed USB**)
- USB 3.x is a bit more complex:
  - SuperSpeed 3.2 Gen 1x1 (5 Gbps) (formerly USB 3.0) (**USB 5Gbps**)
  - SuperSpeed+ 3.2 Gen 2x1 (10 Gbps; 1 lane) (formerly USB 3.1) (**USB 10Gbps**)
  - SuperSpeed+ 3.2 Gen 1x2 (10 Gbps; 2 lanes) (USB-C required)
  - SuperSpeed+ 3.2 Gen 2x2 (20 Gbps; 2 lanes) (USB-C required)
- USB 4 requires USB-C and is essentially tunnelling protocol:
  - USB4 Gen 2x1 (10Gbps; 1 lane; different from USB 3.2 Gen 2x2)
  - USB4 Gen 2x2 (**USB 20Gbps**)
  - USB4 Gen 3x1
  - USB4 Gen 3x2 (**USB 40Gbps**)
  - USB4 Gen 4 (Symmetric = **USB 80Gbps**; Asymmetric)

# USB Type C connector



# Why USB 2.0 is still relevant?

- USB 3.x and USB 4.0 are not replacing USB 2.0
  Backwards compatibility is achieved by dual bus
  The upper layers are pretty much the same
- Every USB 3.x hub contains both USB 2.0 and USB 3.x hub inside
- USB 3.x and USB-C connectors contain dedicated USB 2.0 D+/D-
  All USB 2.0 rules apply on D+/D- signals
- There's a lot of devices that are fine with USB 2.0 speeds:
  - Keyboard
  - Mouse
  - Controllers
  - Zephyr applications ☺

# USB Transfer Types

USB generalizes all possible transfers into 4 types:

- Control
  Used for handling commands, e.g. GET DESCRIPTOR
  Class and vendor commands possible, e.g. volume adjustment
- Interrupt
  Periodic, guaranteed latency, retry on errors, e.g.: keyboard, mouse
- Isochronous
  Periodic, guaranteed bandwidth, no retry or delivery guarantee, e.g.: audio
- Bulk
  Transfer large data, retry on errors, e.g.: mass storage

# USB Classes

## the communication protocols

USB specific classes, e.g.:

- Hub
- Human Interface Device (HID)

Protocol wrappers, e.g.:

- Mass Storage Class (MSC)
- Communications Device Class (CDC)
- Printer

Vendor specific:

- Quite a few ☺

# USB traffic capture

USB traffic can be captured in software:

- On Linux using usbmon module
- On Windows using USBPcap
- On Mac OS using XHC interface

There are Open Source hardware USB 2.0 sniffers available:

- OpenVizsla
- LambdaConcept USB2Sniffer

Sigrok can decode Low and Full speed signaling (capture with logic analyzer)

To my best knowledge, there are no Open Source USB 3.x or USB4 hardware sniffers

Zephyr® Project
Developer Summit

# sudo modprobe usbmon

The Wireshark Network Analyzer

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

Welcome to Wireshark

## Capture

...using this filter: Enter a capture filter ...

- vmnet1
- vmnet8
- enp0s13f0u1u4u4
- any
- Loopback: lo
- docker0
- wlp0s20f3
- bluetooth-monitor
- usbmon4
- usbmon3
- usbmon2
- usbmon1
- usbmon0
- nflog
- nfqueue
- dbus-system
- dbus-session
- Cisco remote capture: ciscodump
- DisplayPort AUX channel monitor capture: dpauxmon
- OpenVizsla Full Speed USB capture: ov-full
- OpenVizsla High Speed USB capture: ov-high

## Learn

User's Guide · Wiki · Questions and Answers · Mailing Lists · SharkFest · Wireshark

You are running Wireshark 4.1.0 (v4.1.0rc0-2406-g7decbda52297).

Ready to load or capture

tmon@tmon-laptop:~

File Edit View Terminal Tabs Help

```
[tmon@tmon-laptop ~]$ lsusb
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 004: ID 0a5c:5843 Broadcom Corp. 58200
Bus 003 Device 003: ID 0c45:6738 Microdia Integrated_Webcam_HD
Bus 003 Device 007: ID 413c:b06e Dell Computer Corp. Dell dock
Bus 003 Device 011: ID 413c:b06f Dell Computer Corp. Dell dock
Bus 003 Device 015: ID 0bda:1100 Realtek Semiconductor Corp. HID Device
Bus 003 Device 014: ID 045e:082c Microsoft Corp. Microsoft Ergonomic Keyboard
Bus 003 Device 012: ID 046d:c52b Logitech, Inc. Unifying Receiver
Bus 003 Device 010: ID 0bda:5409 Realtek Semiconductor Corp. USB2.1 Hub
Bus 003 Device 017: ID 2fe3:0008 NordicSemiconductor Zephyr USBD MSC
Bus 003 Device 016: ID 1366:1050 SEGGER J-Link
Bus 003 Device 013: ID 2109:2817 VIA Labs, Inc. USB2.0 Hub
Bus 003 Device 009: ID 2109:2817 VIA Labs, Inc. USB2.0 Hub
Bus 003 Device 008: ID 08bb:2902 Texas Instruments PCM2902 Audio Codec
Bus 003 Device 006: ID 0bda:5413 Realtek Semiconductor Corp. Dell dock
Bus 003 Device 005: ID 0bda:5487 Realtek Semiconductor Corp. Dell dock
Bus 003 Device 002: ID 1d5c:5801 Fresco Logic USB2.0 Hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 005: ID 0bda:8153 Realtek Semiconductor Corp. RTL8153 Gigabit Ethernet Adapter
Bus 002 Device 007: ID 0bda:0409 Realtek Semiconductor Corp. USB3.2 Hub
Bus 002 Device 008: ID 2109:0817 VIA Labs, Inc. USB3.0 Hub
Bus 002 Device 006: ID 2109:0817 VIA Labs, Inc. USB3.0 Hub
Bus 002 Device 004: ID 0bda:0413 Realtek Semiconductor Corp. Dell dock
Bus 002 Device 003: ID 0bda:0487 Realtek Semiconductor Corp. Dell dock
Bus 002 Device 002: ID 8087:0b40 Intel Corp. USB3.0 Hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
[tmon@tmon-laptop ~]$
```

# Wireshark extcap

The Wireshark Network Analyzer

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

Welcome to Wireshark

Capture

...using this filter: Enter a capture filter ...

- USBPcap1
- USBPcap2
- OpenVizsla Full Speed USB capture
- OpenVizsla High Speed USB capture
- OpenVizsla Low Speed USB capture

Learn

User's Guide · Wiki · Questions and Answers · Maili

You are running Wireshark 4.1.0 (v4.1.0rc0-1645-g02238edf92

About Wireshark

Wireshark | Authors | Folders | Plugins | Keyboard Shortcuts | Acknowledgments | License

extcap

| Name | Location | Typical Files |
|---|---|---|
| Global Extcap path | C:\Program Files\Wireshark\extcap | external capture (extcap) plugins |
| Personal Extcap path | C:\Users\tomo\AppData\Roaming\Wireshark\extcap | external capture (extcap) plugins |

extcap

File Home Share View

« Wire... › extcap

Search extcap

This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
OSDisk (C:)
USB Drive (D:)

| Name | Date modified | Typ |
|---|---|---|
| ftdi1.dll | 10/18/2022 3:47 PM | Ap |
| getopt.dll | 10/18/2022 3:46 PM | Ap |
| libusb-1.0.dll | 10/18/2022 3:47 PM | Ap |
| openvizsla.dll | 10/19/2022 9:20 PM | Ap |
| ovextcap.exe | 10/19/2022 9:20 PM | Ap |
| USBPcapCMD.exe | 5/22/2020 11:01 AM | Ap |
| zip.dll | 10/18/2022 3:48 PM | Ap |
| zlib1.dll | 10/18/2022 3:47 PM | Ap |

8 items

8 items (Disk free space: 34.2 GB) 523 KB Computer

Ready to load or capture | No Packets | Profile: Default

# USBPcap extcap options

The Wireshark Network Analyzer

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

Welcome to Wireshark

Capture

...using this filter: Enter a capture filter ...

- USBPcap1
- USBPcap2
- OpenVizsla Full Speed USB capture
- OpenVizsla High Speed USB capture
- OpenVizsla Low Speed USB capture

Learn

User's Guide · Wiki · Questions and Answers · Mai

You are running Wireshark 4.1.0 (v4.1.0rc0-1645-g02238edf9

Wireshark · Interface Options: USBPcap2

Default

Snapshot length: 65535

Capture buffer length: 1048576

- [ ] Capture from all devices connected
- [x] Capture from newly connected devices
- [ ] Inject already connected devices descriptors into capture data

Attached USB Devices:

- [ ] [9] Generic USB Hub
  - [ ] [12] Generic USB Hub
    - [ ] [16] USB Composite Device
      - JLink CDC UART Port (COM14)
      - JLink CDC UART Port (COM13)
      - J-Link driver
    - [x] [17] USB Mass Storage Device
      - Zephyr FlashDisk USB Device

- [x] Save parameter(s) on capture start

Restore Defaults | Save | Close | Help

Ready to load or capture | No Packets | Profile: Default

# Mac OS with disabled System Integrity Protection

Zephyr® Project Developer Summit

The Wireshark Network Analyzer

Apply a display filter ... <⌘/>

Welcome to Wireshark

**Capture**

...using this filter: Enter a capture filter ... All interfaces shown

Wi-Fi: en0
Loopback: lo0
Thunderbolt Bridge: bridge0
Thunderbolt 4: en4
XHC0
XHC1
XHC20
VHC128
awdl0
llw0
utun0
utun1
utun2
Thunderbolt 1: en1
Thunderbolt 2: en2
Thunderbolt 3: en3
gif0
stf0
ap1
Cisco remote capture: ciscodump
OpenVizsla Full Speed USB capture: ov-full

desowin — -zsh — 80×10

```
desowin@Tomaszs-MacBook-Pro ~ % sudo ifconfig VHC128 up
desowin@Tomaszs-MacBook-Pro ~ % sudo ifconfig XHC0 up
desowin@Tomaszs-MacBook-Pro ~ % sudo ifconfig XHC1 up
desowin@Tomaszs-MacBook-Pro ~ % sudo ifconfig XHC20 up
desowin@Tomaszs-MacBook-Pro ~ % lsusb
Bus 001 Device 001: ID 13fe:6700 Kingston Technology Company Inc.
Bus 000 Device 001: ID 0781:558c SanDisk Corp. Extreme Portable SSD
Bus 020 Device 028: ID 2fe3:0008 NordicSemiconductor
desowin@Tomaszs-MacBook-Pro ~ %
```

**Learn**

User's Guide · Wiki · Questions and Answers · Mailing Lists · SharkFest · Wireshark Discord · Donate

You are running Wireshark 4.0.5 (v4.0.5-0-ge556162d8da3). You receive automatic updates.

Ready to load or capture | No Packets | Profile: Default

MacBook Pro

Hardware
ATA
Apple Pay
Audio
Bluetooth
Camera
Card Reader
Controller
Diagnostics
Disc Burning
Ethernet
Fibre Channel
FireWire
Graphics/Displays
Memory
NVMExpress
PCI
Parallel SCSI
Power
Printers
SAS
SATA
SPI
Storage
Thunderbolt/USB4
USB
Network
Firewall
Locations
Volumes
WWAN
Wi-Fi
Software
Accessibility
Applications
Developer
Disabled Software
Extensions
Fonts
Frameworks
Installations
Language & Regi...
Legacy Software
Logs
Managed Client
Preference Panes
Printer Software
Profiles
Raw Support
SmartCards

Thunderbolt/USB4 Device Tree

Thunderbolt Bus 0
Thunderbolt Bus 1
tmon-laptop

```
Route String:        0
Firmware Version:    47.5
Domain UUID:         CC2ACA3E-BA8A-5850-B327-A022F21ABBCF
Port:
    Status:                          Device connected
    Link Status:                     0x2
    Speed:                           Up to 20Gb/s x1
    Current Link Width:              0x2
    Receptacle:                      4
    Link Controller Firmware Version: 1.34.0
Port:
    Status:                          No device connected
    Link Status:                     0x7
    Speed:                           Up to 40Gb/s x1
    Current Link Width:              0x1
    Receptacle:                      3
    Link Controller Firmware Version: 1.34.0

  tmon-laptop:

    Vendor Name:   Intel Corp.
    Device Name:   tmon-laptop
    Device ID:     0x1
    Vendor ID:     0x8086
    Domain UUID:   C5FF8780-C0D0-45D5-FFFF-FFFFFFFFFFFF
    Services:
      Internet Protocol:
        Protocol ID:       1
        Protocol Version:  1
        Protocol Revision: 1
        Service UUID:      CA8961C6-9541-CE1C-5949-B8BD4F5A5F2E
```

Tomasz's MacBook Pro › Hardware › Thunderbolt/USB4 › Thunderbolt Bus 1

# OpenVizsla extcap options

The Wireshark Network Analyzer

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

Welcome to Wireshark

Development Build

Capture

...using this filter: Enter a capture filter ...

All interfaces shown

USBPcap1
USBPcap2
OpenVizsla Full Speed USB capture
OpenVizsla High Speed USB capture
OpenVizsla Low Speed USB capture

Wireshark · Interface Options: OpenVizsla Full Speed USB capture

Capture Debug

Filter NAKed transactions

Filter Start-of-Frame packets

Save parameter(s) on capture start

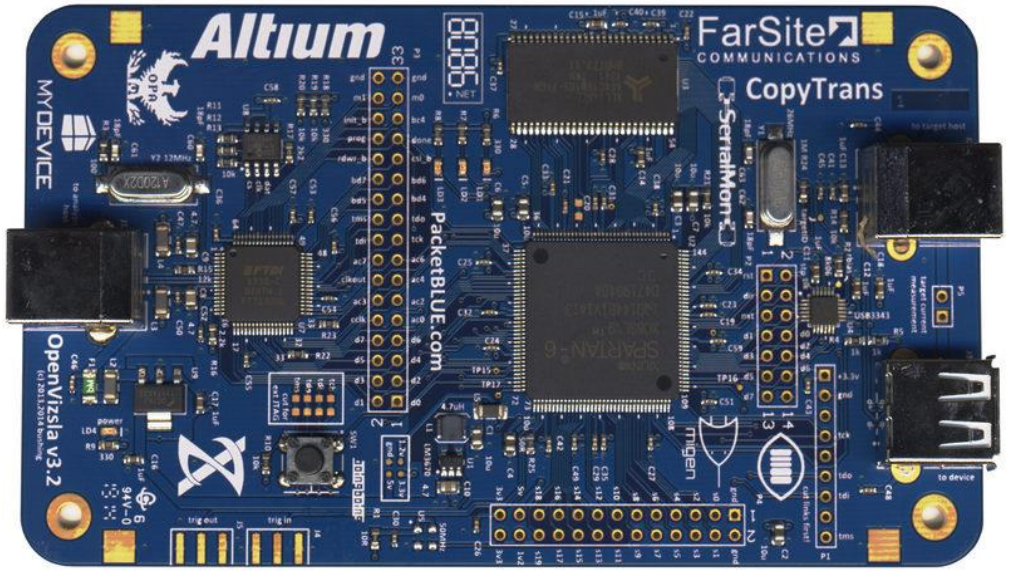
Restore Defaults Save Close Help

Learn

User's Guide · Wiki · Questions and Answers · Mailing Lists · SharkFest · Wireshark Discord · Donate

You are running Wireshark 4.1.0 (v4.1.0rc0-1645-g02238edf9297). You receive automatic updates.

Ready to load or capture | No Packets | Profile: Default




OpenVizsla PCBA photo from [sysmocom webshop](sysmocom webshop).

# USB “Packets”

Wireshark shows what the capture engine provided, e.g.:

- libpcap (usbmon) provides “USB packets with Linux header and padding”
- USBPcap provides “USB packets with USBPcap header”
- OpenVizsla provides “USB 2.0/1.1/1.0 packets”

The “USB 2.0/1.1/1.0 packets” are described in USB 2.0 Specification, Chapter 8.

Software sniffers capture USB Request Blocks submitted to the host controller driver.

The “Linux header” and “USBPcap header” contain OS specific URB information.

# What software sniffers show?

Device driver **submits URB**, HCD handles URB and **reports back** to Device driver.

All software sniffer “packets” contain OS specific metadata (URB ID, endpoint, …)

| | OUT (send to device) | | IN (receive from device) | |
|---|---|---|---|---|
| | Host→Device | Device→Host | Host→Device | Device→Host |
| Control | SETUP Data (8 bytes) + Payload (if wLength > 0) | Indicates that URB handling is done (result code is OS specific) | SETUP Data (8 bytes) | Payload (if wLength > 0)* |
| Interrupt | Payload | | | Payload* |
| Bulk | Payload | | | Payload* |
| Isochronous | Payload | | | Payload* |

* Metadata only if the URB has failed/was cancelled, e.g. device was disconnected, STLL occurred, …

# Capture time

Zephyr USBD Mass Storage sample running on nRF52840 DK

Capture simultaneously with USBPcap and OpenVizsla

# Summary

- USB 2.0 is still relevant today
  - USB 3.x backwards compatibility with USB 2.0 is achieved by dual bus
  - USB4 is essentially tunnelling protocol, operating alongside USB 2.0
- Host initiates all communication
  - IN and OUT is always from Host perspective
  - Device cannot send data unless host asks for it (driver submits “IN” URB)
- Software sniffers capture URBs
  - Every URB is captured as 2 “URB packets”
    - Driver to HCI includes data payload from host to device (if any)
    - HCI to driver includes data payload from device to host (if any)
  - URB level capture is sufficient for general use
    - Understanding USB at packet level helps make sense out of the “URB packets”