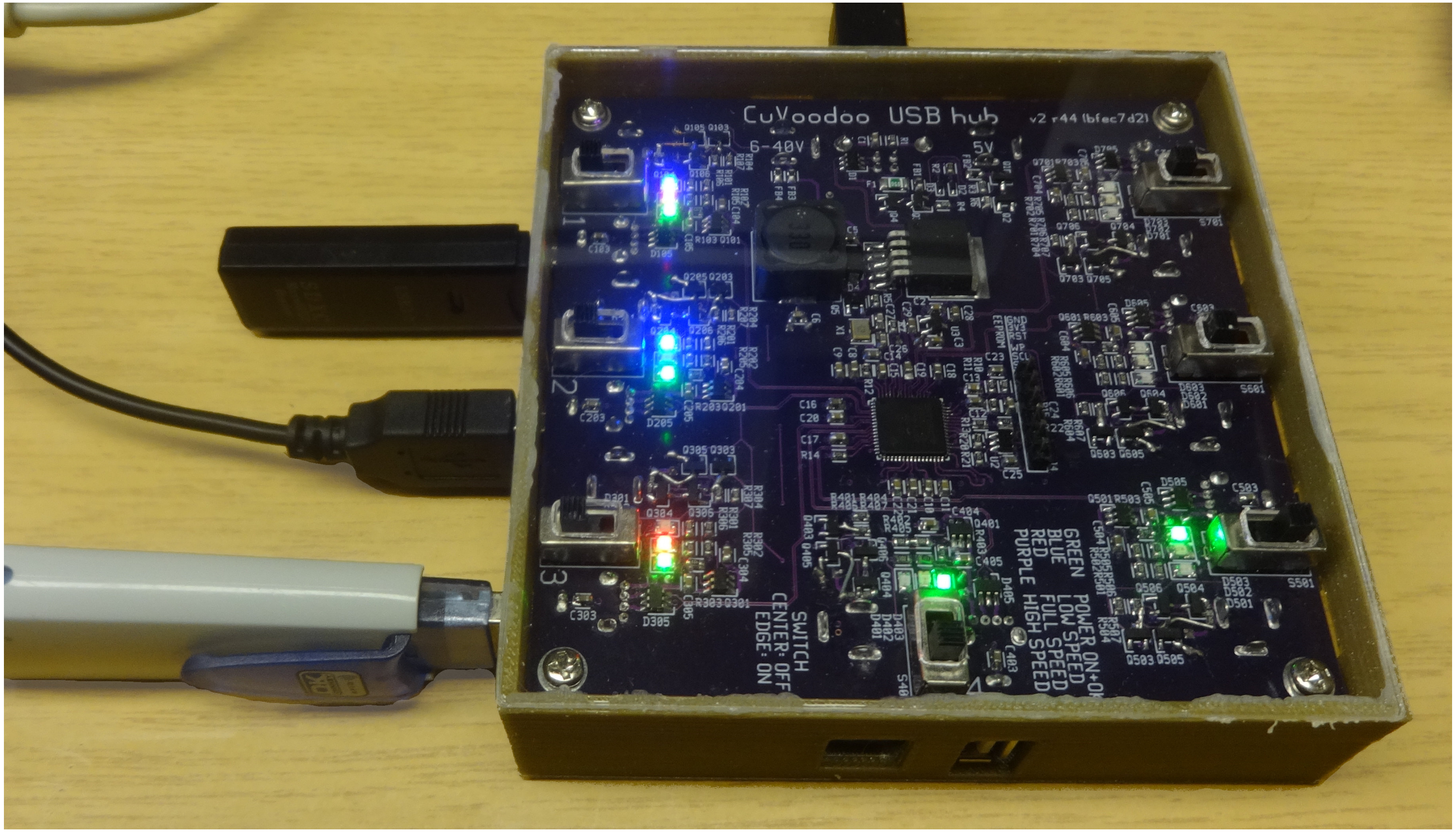
USB hub for device testing

tsaitgaist, OsmoDevCall, 2022-12-21

USB hub for device testing

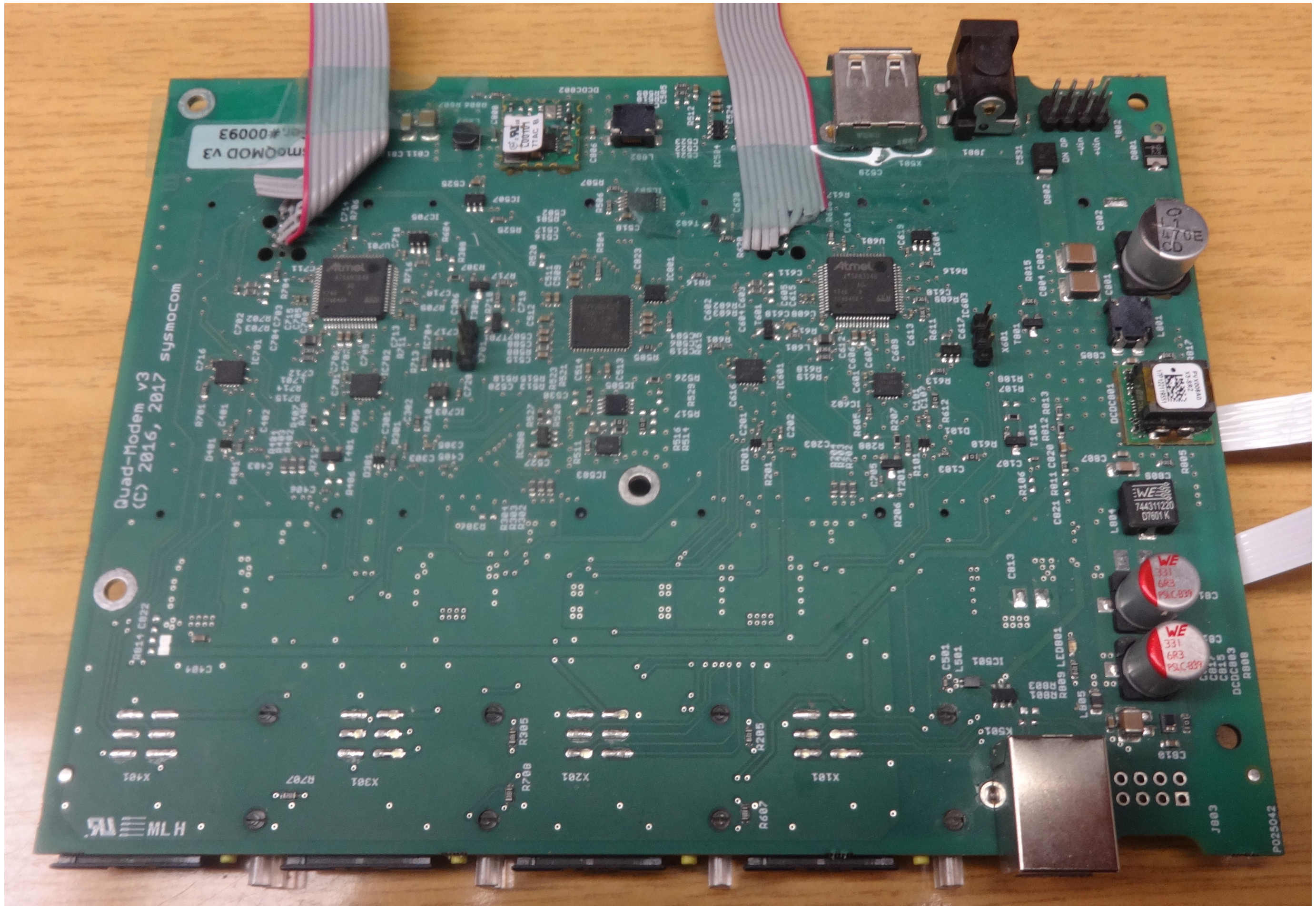


how I build a special purpose USB hub

in the beginning

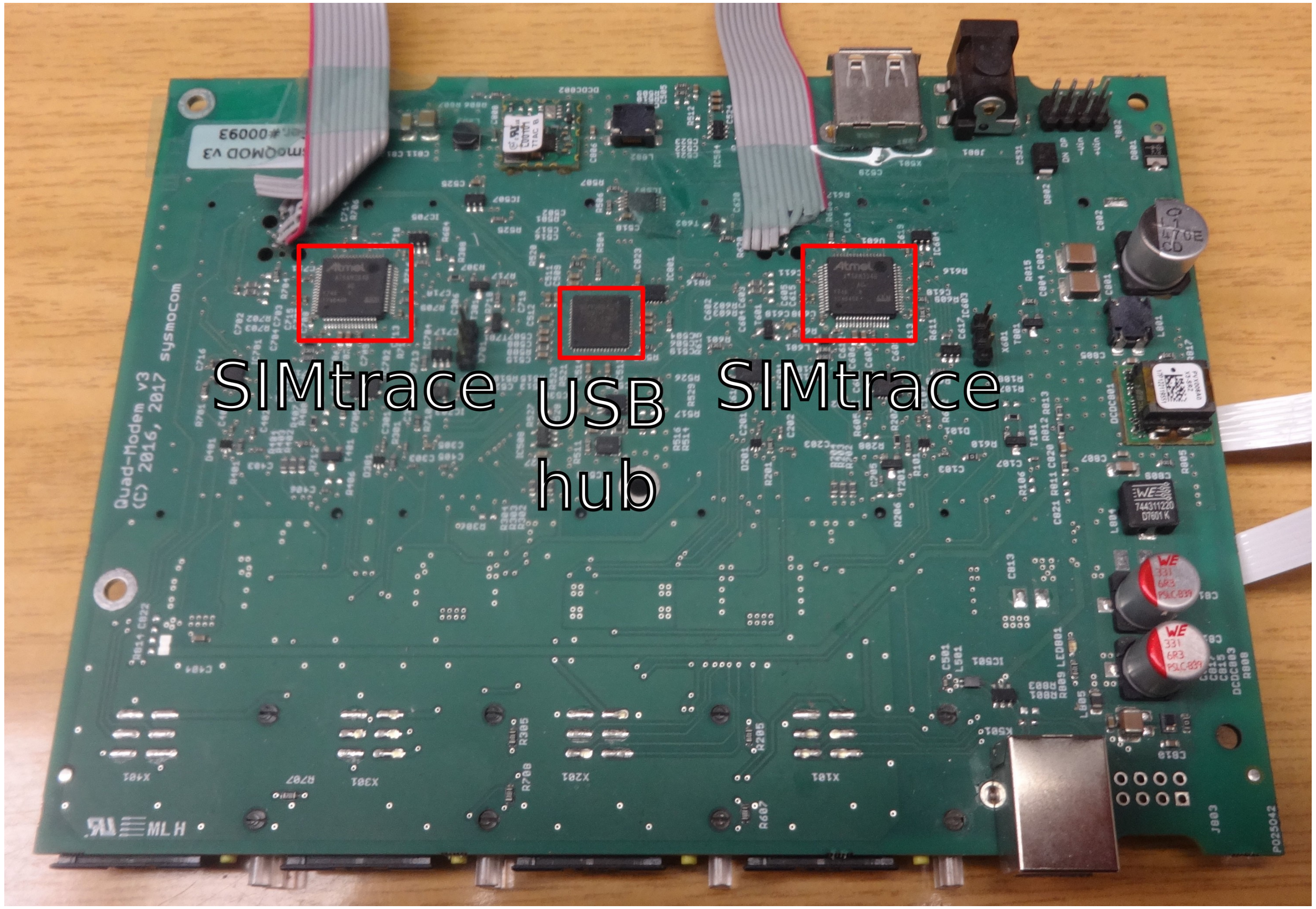
I was developing for the sysmoQMOD quad modem board

development target



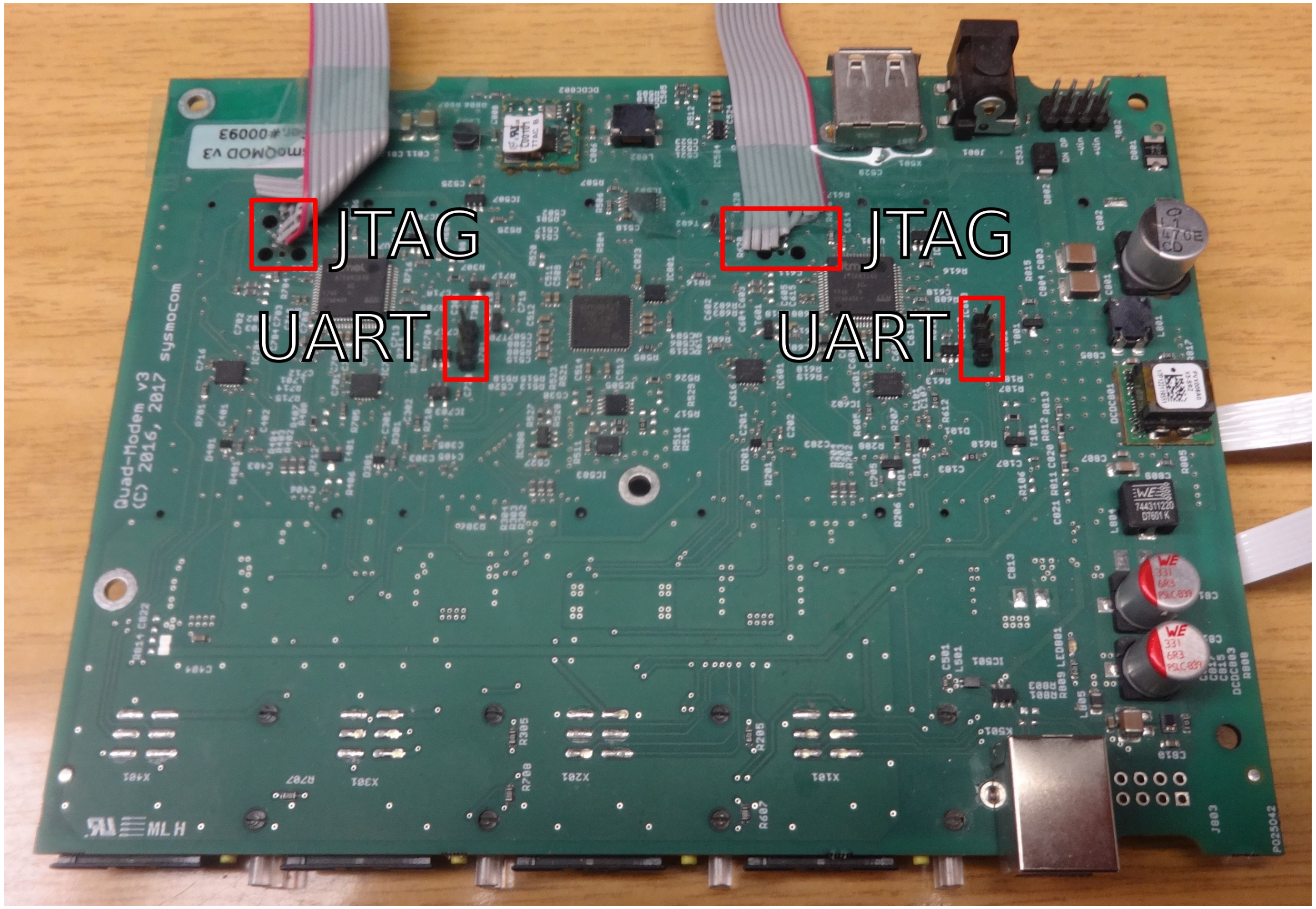
the debug interfaces are on the back

development target



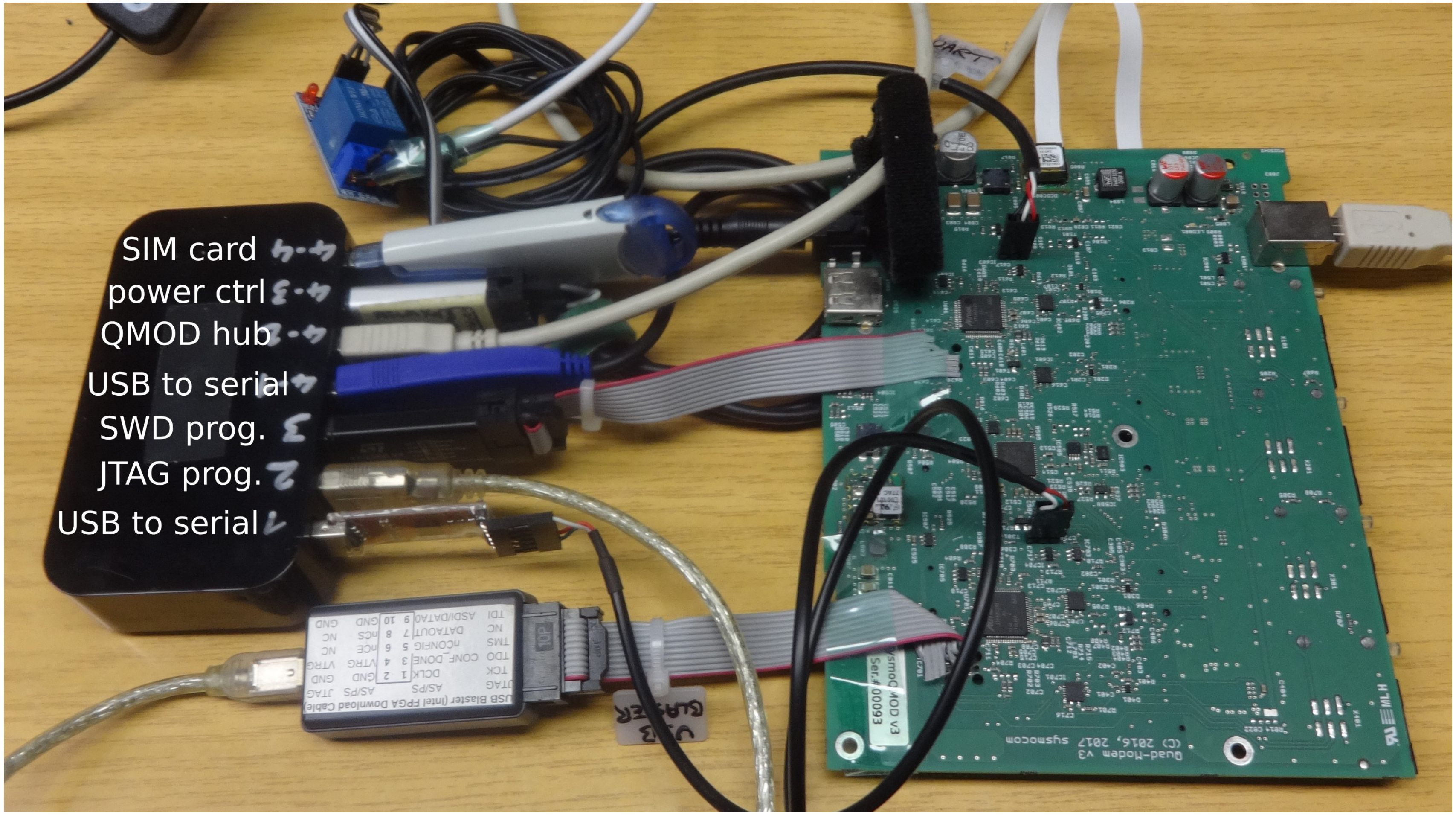
3 sub-targets

development target



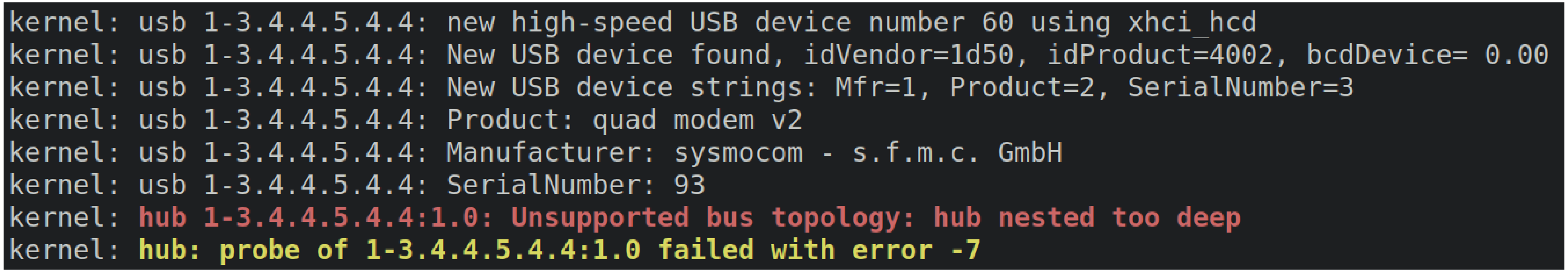
multiple debug interface

testing target



7-port hub required

USB hierarchy



```
kernel: usb 1-3.4.4.5.4.4: new high-speed USB device number 60 using xhci_hcd
kernel: usb 1-3.4.4.5.4.4: New USB device found, idVendor=1d50, idProduct=4002, bcdDevice= 0.00
kernel: usb 1-3.4.4.5.4.4: New USB device strings: Mfr=1, Product=2, SerialNumber=3
kernel: usb 1-3.4.4.5.4.4: Product: quad modem v2
kernel: usb 1-3.4.4.5.4.4: Manufacturer: sysmocom - s.f.m.c. GmbH
kernel: usb 1-3.4.4.5.4.4: SerialNumber: 93
kernel: hub 1-3.4.4.5.4.4:1.0: Unsupported bus topology: hub nested too deep
kernel: hub: probe of 1-3.4.4.5.4.4:1.0 failed with error -7
```

the USB devices dot not appear when plugging the hub in

USB hierarchy

4.1.1 Bus Topology

The USB connects USB devices with the USB host. The USB physical interconnect is a tiered star topology. A hub is at the center of each star. Each wire segment is a point-to-point connection between the host and a hub or function, or a hub connected to another hub or function. Figure 4-1 illustrates the topology of the USB.

Due to timing constraints allowed for hub and cable propagation times, the maximum number of tiers allowed is seven (including the root tier). Note that in seven tiers, five non-root hubs maximum can be supported in a communication path between the host and any device. A compound device (see Figure 4-1) occupies two tiers; therefore, it cannot be enabled if attached at tier level seven. Only functions can be enabled in tier seven.

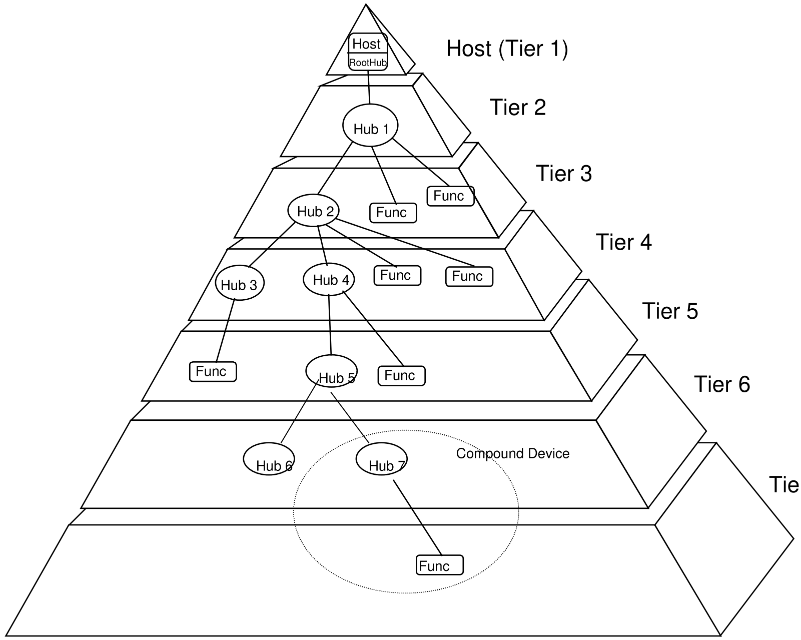
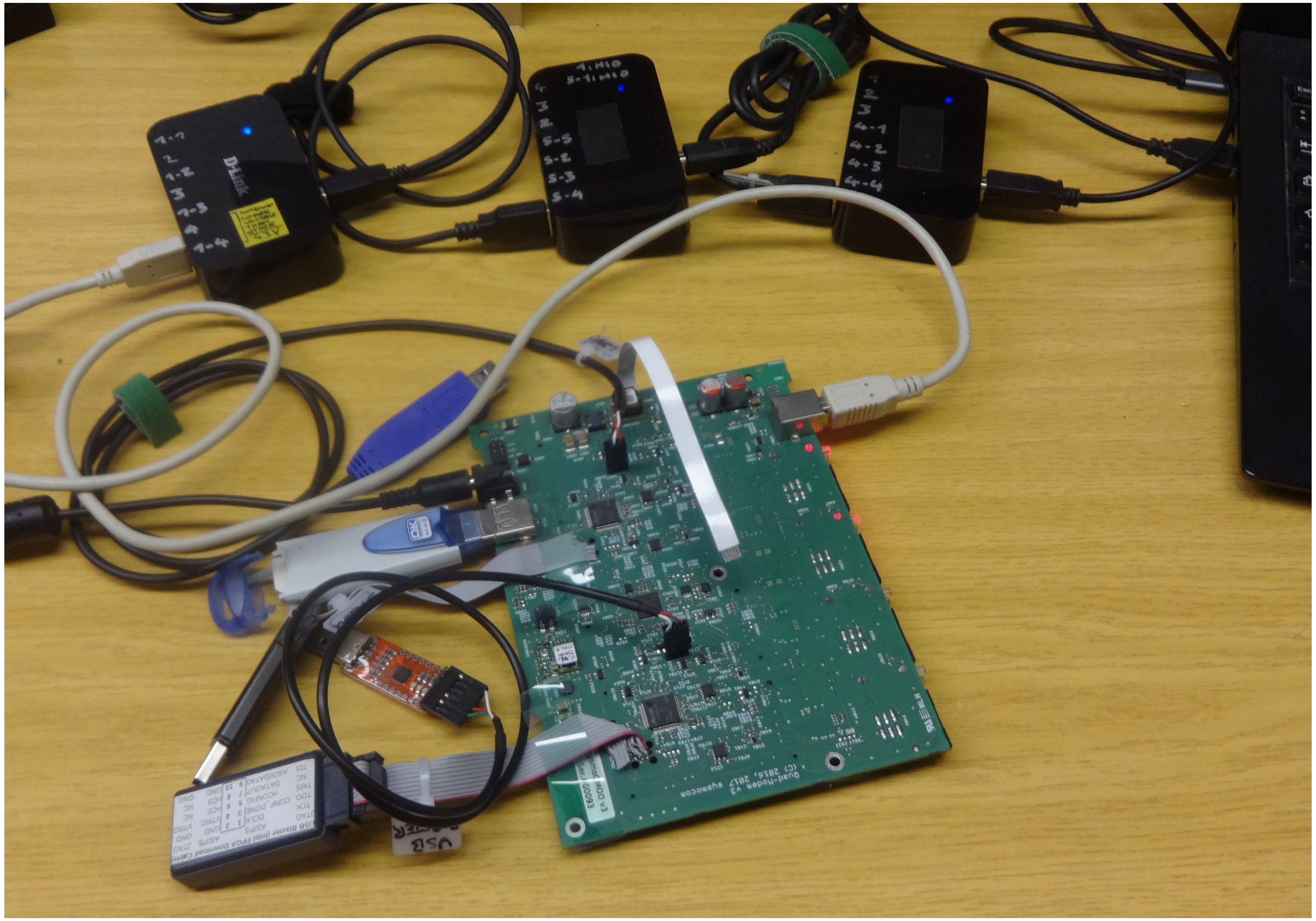


Figure 4-1. Bus Topology

USB hierarchy



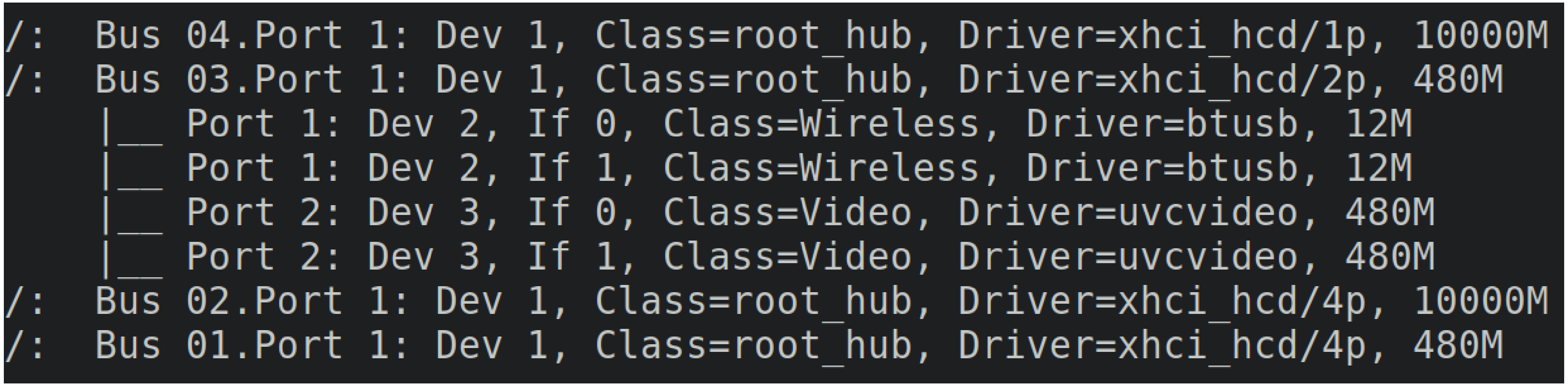
my setup has only 3 hubs

USB tree

disconnect everything and go step per step

USB tree

use `lsusb -t` to show the tree

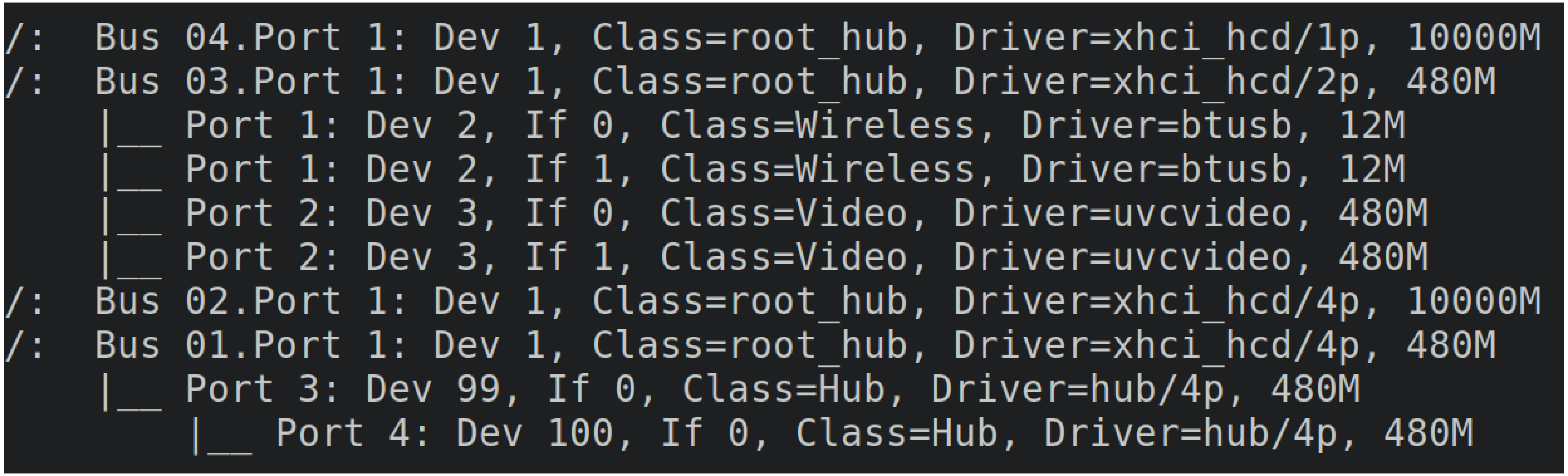


```
/:  Bus 04.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/1p, 10000M
/:  Bus 03.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/2p, 480M
    |__ Port 1: Dev 2, If 0, Class=Wireless, Driver=btusb, 12M
    |__ Port 1: Dev 2, If 1, Class=Wireless, Driver=btusb, 12M
    |__ Port 2: Dev 3, If 0, Class=Video, Driver=uvcvideo, 480M
    |__ Port 2: Dev 3, If 1, Class=Video, Driver=uvcvideo, 480M
/:  Bus 02.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/4p, 10000M
/:  Bus 01.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/4p, 480M
```

`root_hub` does not count as hub

USB tree

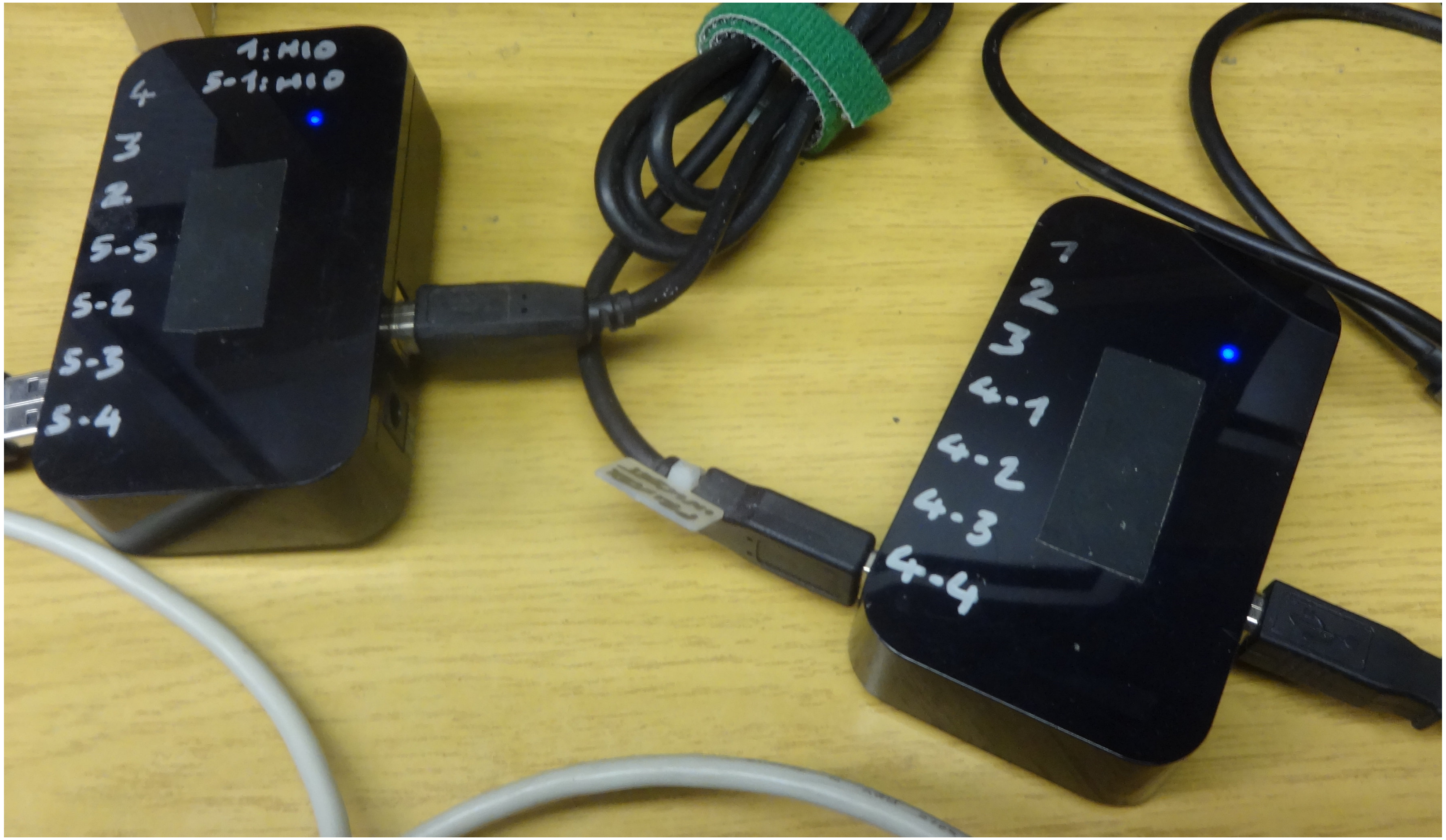
plug the first hub



```
/:  Bus 04.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/1p, 10000M
/:  Bus 03.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/2p, 480M
    |__ Port 1: Dev 2, If 0, Class=Wireless, Driver=btusb, 12M
    |__ Port 1: Dev 2, If 1, Class=Wireless, Driver=btusb, 12M
    |__ Port 2: Dev 3, If 0, Class=Video, Driver=uvcvideo, 480M
    |__ Port 2: Dev 3, If 1, Class=Video, Driver=uvcvideo, 480M
/:  Bus 02.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/4p, 10000M
/:  Bus 01.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/4p, 480M
    |__ Port 3: Dev 99, If 0, Class=Hub, Driver=hub/4p, 480M
        |__ Port 4: Dev 100, If 0, Class=Hub, Driver=hub/4p, 480M
```

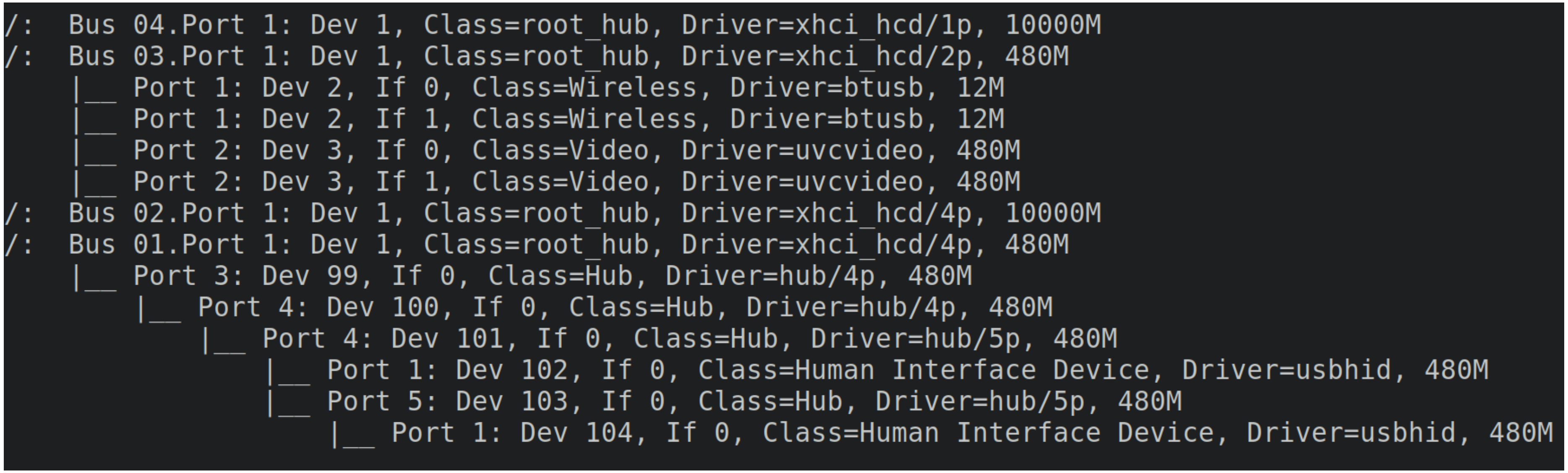
USB tree

plug the second hub



USB tree

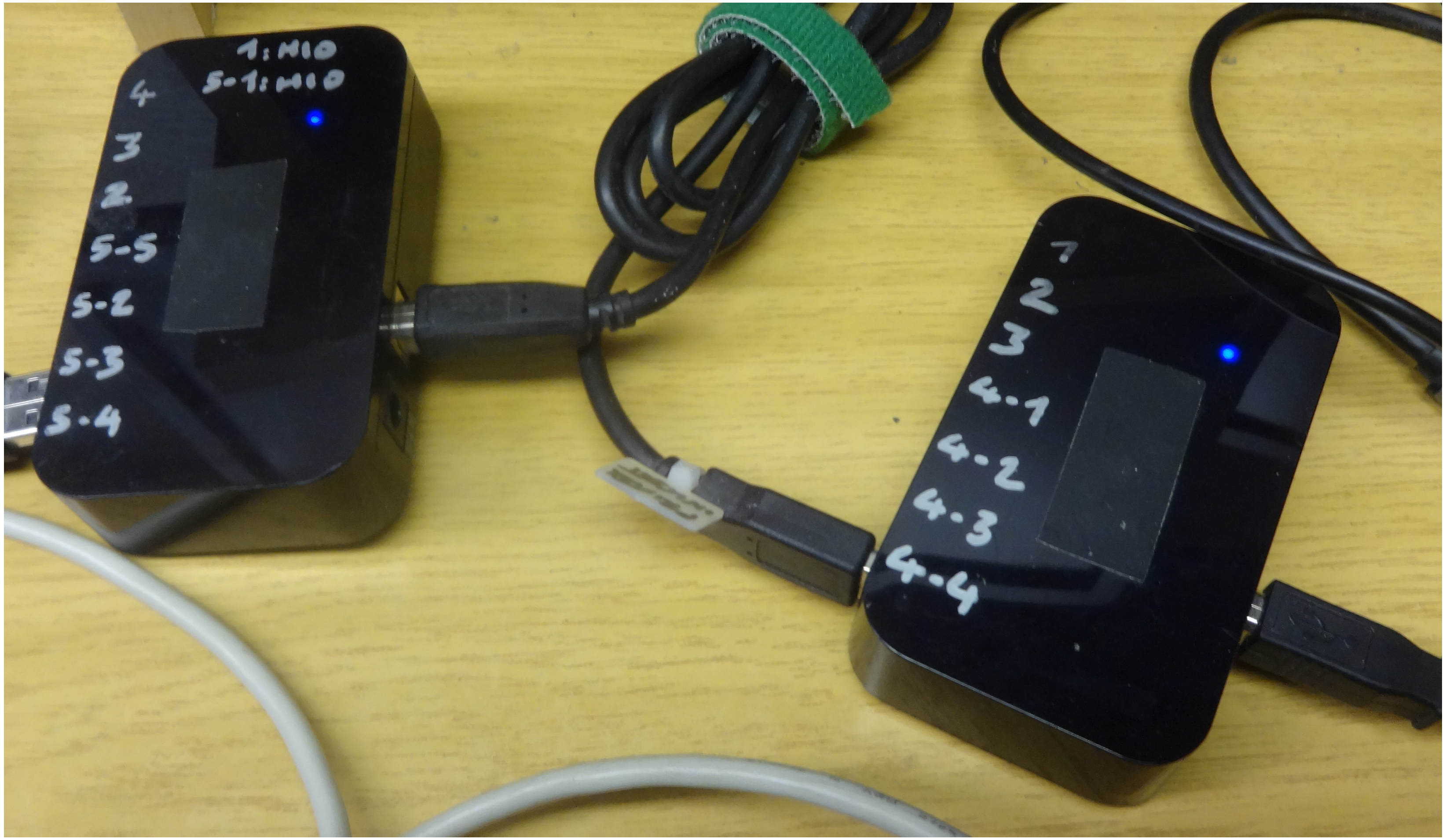
plug the second hub



```
/:  Bus 04.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/1p, 10000M
/:  Bus 03.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/2p, 480M
    |__ Port 1: Dev 2, If 0, Class=Wireless, Driver=btusb, 12M
    |__ Port 1: Dev 2, If 1, Class=Wireless, Driver=btusb, 12M
    |__ Port 2: Dev 3, If 0, Class=Video, Driver=uvcvideo, 480M
    |__ Port 2: Dev 3, If 1, Class=Video, Driver=uvcvideo, 480M
/:  Bus 02.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/4p, 10000M
/:  Bus 01.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/4p, 480M
    |__ Port 3: Dev 99, If 0, Class=Hub, Driver=hub/4p, 480M
        |__ Port 4: Dev 100, If 0, Class=Hub, Driver=hub/4p, 480M
            |__ Port 4: Dev 101, If 0, Class=Hub, Driver=hub/5p, 480M
                |__ Port 1: Dev 102, If 0, Class=Human Interface Device, Driver=usbhid, 480M
                |__ Port 5: Dev 103, If 0, Class=Hub, Driver=hub/5p, 480M
                    |__ Port 1: Dev 104, If 0, Class=Human Interface Device, Driver=usbhid, 480M
```

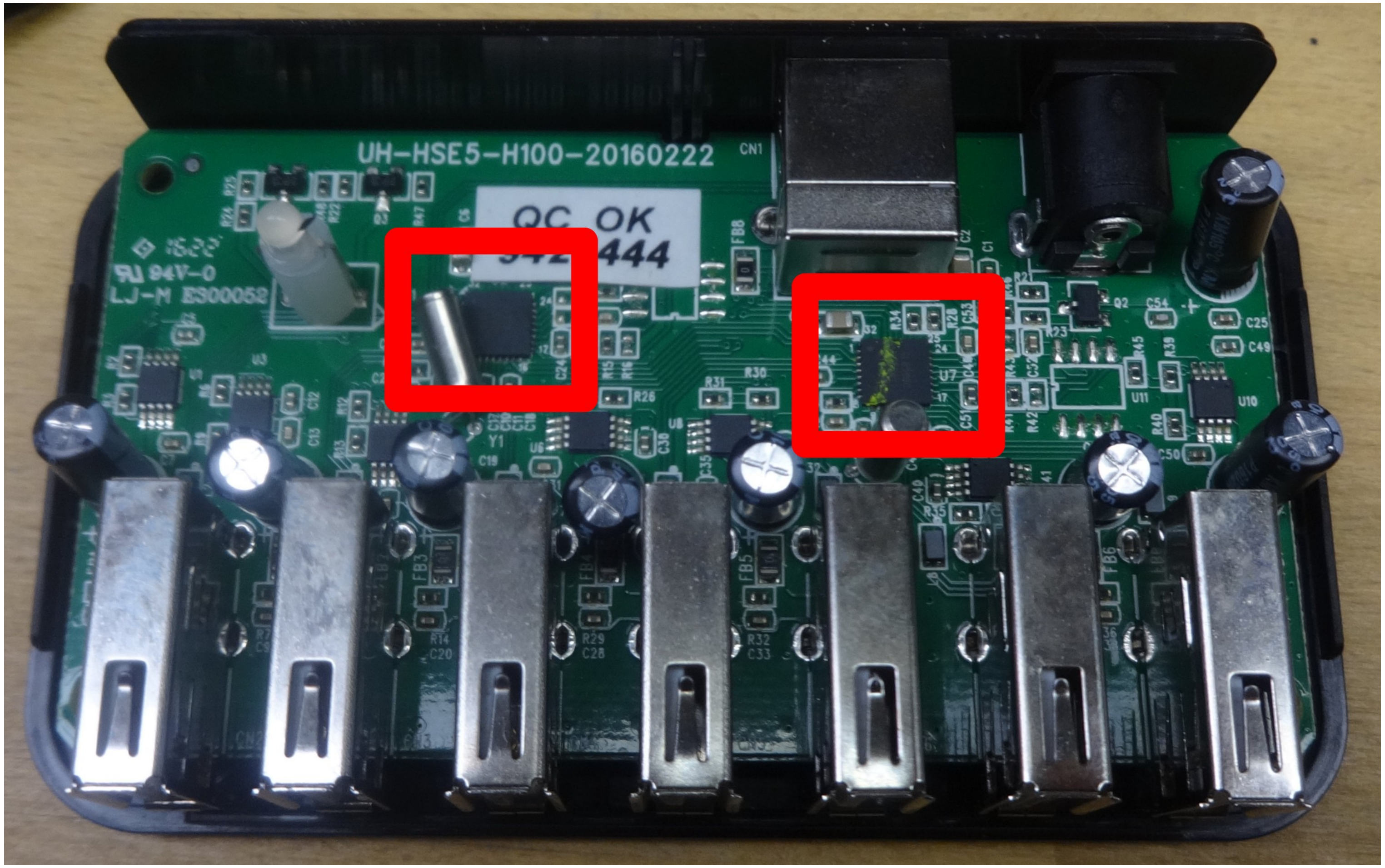
we actually have already 4 hubs

USB tree



mark the port numbers

hub internals



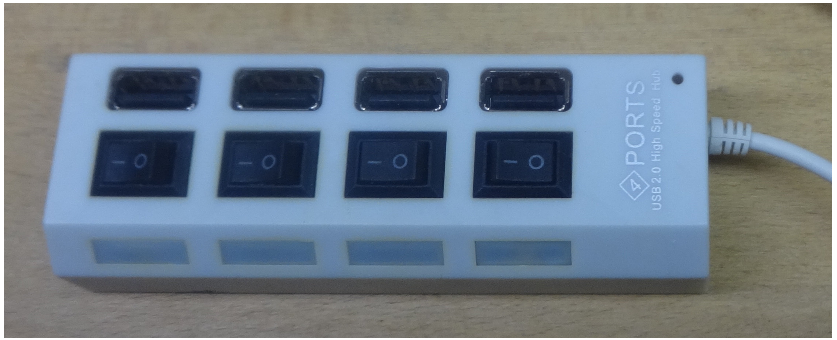
most 7-port USB hub are actually 2 4-port hubs chained

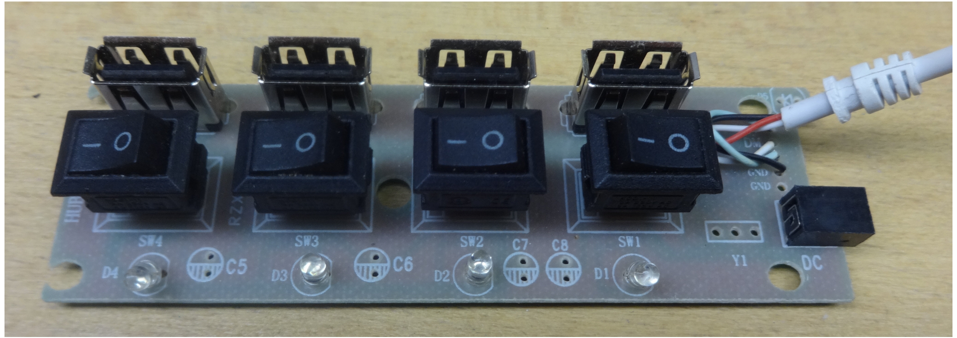
hub requirements

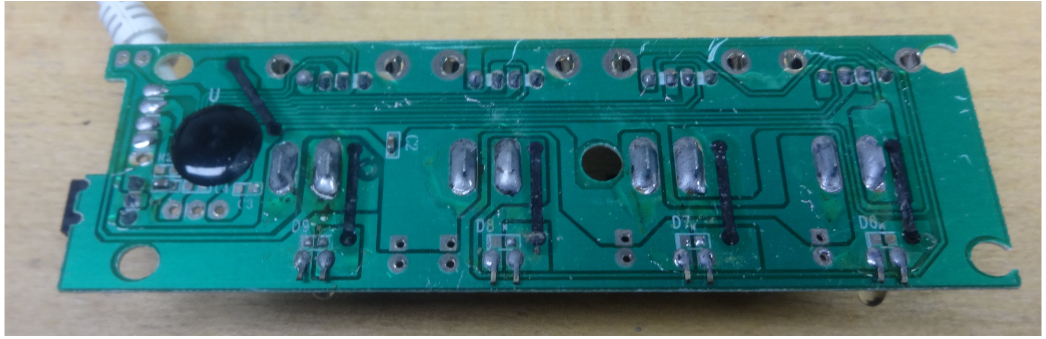
testing hub requirements:

- 7-port flat

hub internals

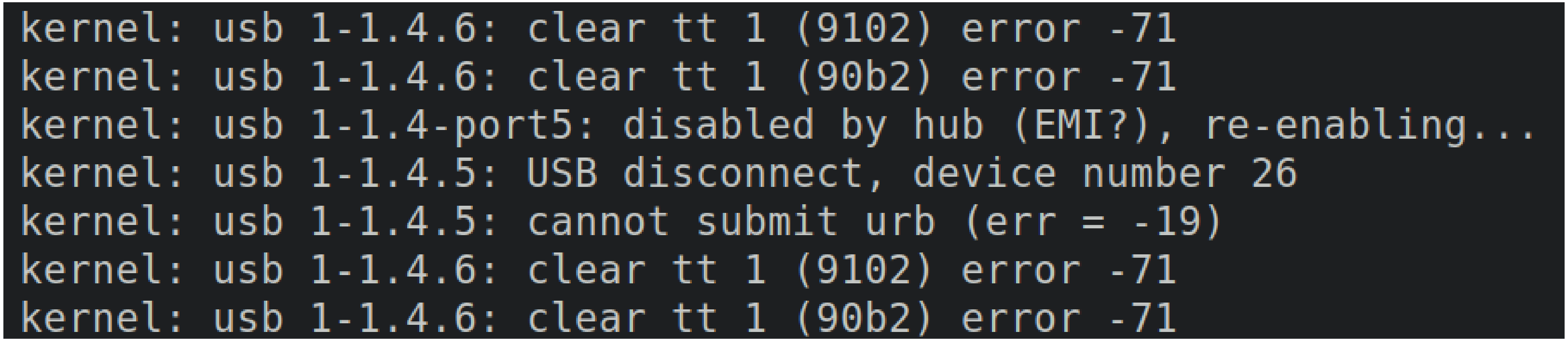






bad routing

bad routing and bad cables can lead to bad signal quality and unstable communication



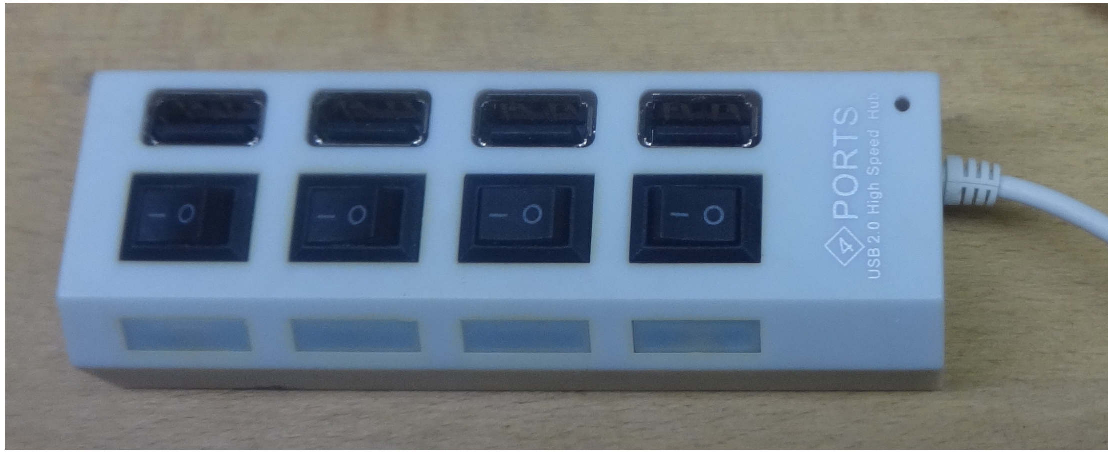
```
kernel: usb 1-1.4.6: clear tt 1 (9102) error -71
kernel: usb 1-1.4.6: clear tt 1 (90b2) error -71
kernel: usb 1-1.4-port5: disabled by hub (EMI?), re-enabling...
kernel: usb 1-1.4.5: USB disconnect, device number 26
kernel: usb 1-1.4.5: cannot submit urb (err = -19)
kernel: usb 1-1.4.6: clear tt 1 (9102) error -71
kernel: usb 1-1.4.6: clear tt 1 (90b2) error -71
```

hub requirements

testing hub requirements:

- 7-port flat
- proper routing (impedance controlled differential signals)

hub internals



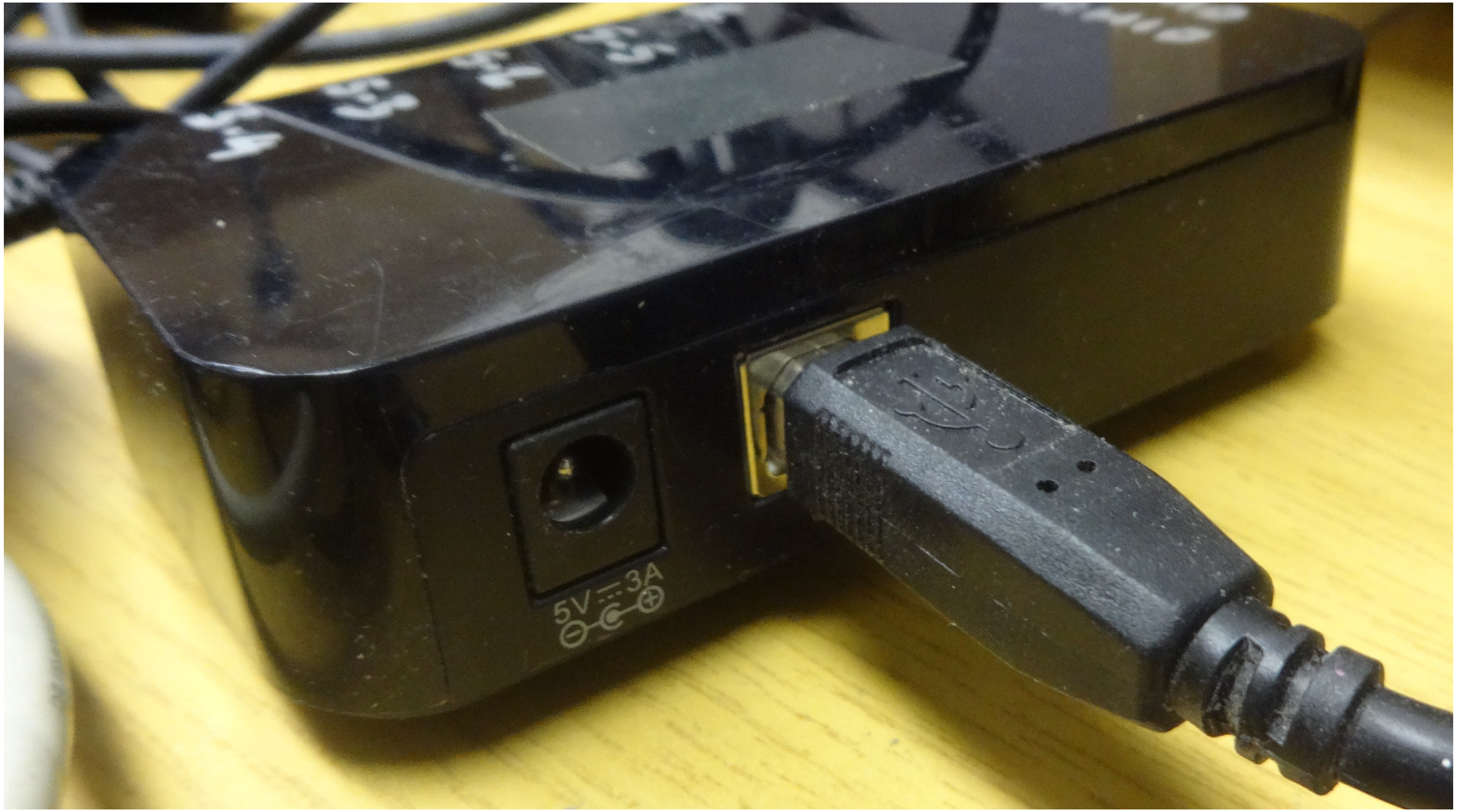
devices can be powered individually

hub requirements

testing hub requirements:

- 7-port flat
- proper routing (impedance controlled differential signals)
- switched power outputs (with indication)

power input



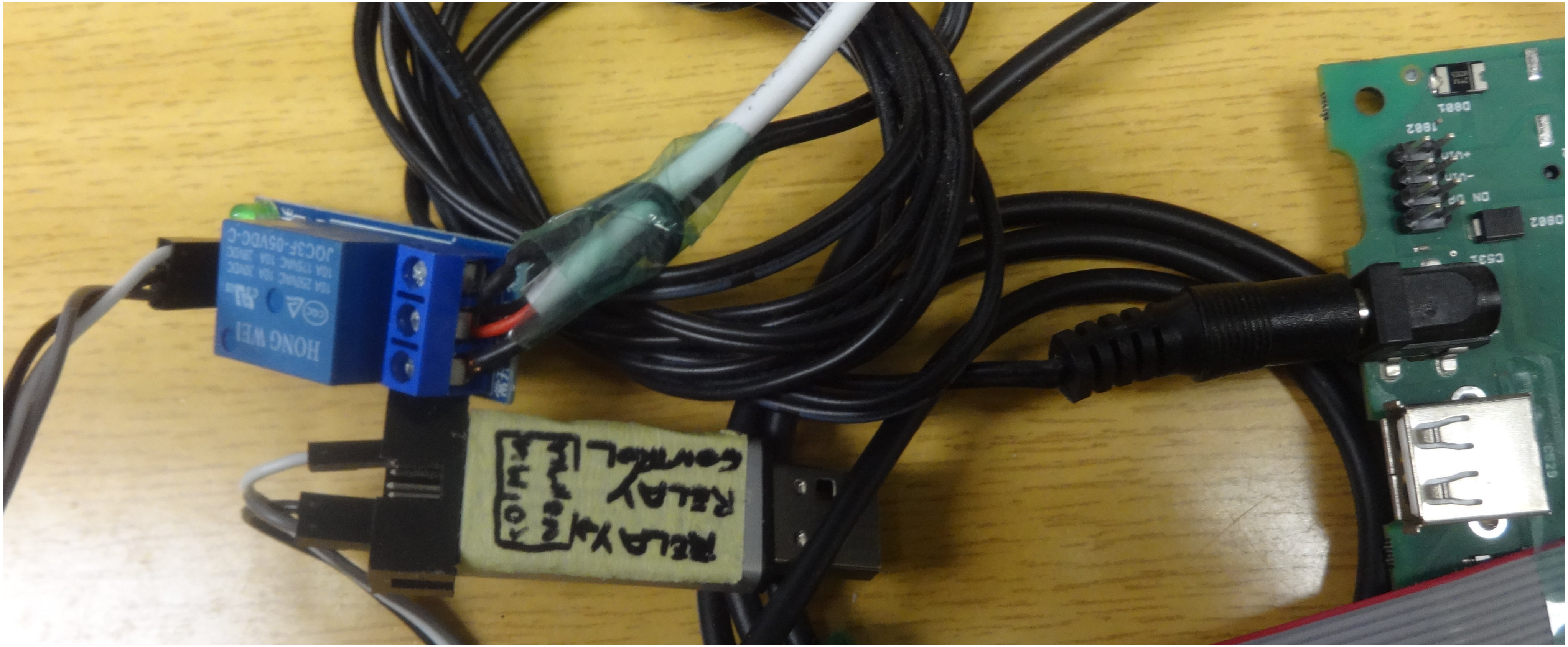
USB 2.0 should draw max. 500 mA from the upstream port. This is not enough to power 7x500 mA downstream ports. An external power supply will help.

hub requirements

testing hub requirements:

- 7-port flat
- proper routing (impedance controlled differential signals)
- switched power outputs (with indication)
- external power supply

power control



to power cycle (12V) the sysmoQMOD, I used a serial controlled relay

hub requirements

- 7-port flat
- proper routing (impedance controlled differential signals)
- switched 5V power outputs (with indication)
- external power supply
- switched high voltage (6-40V) power outputs

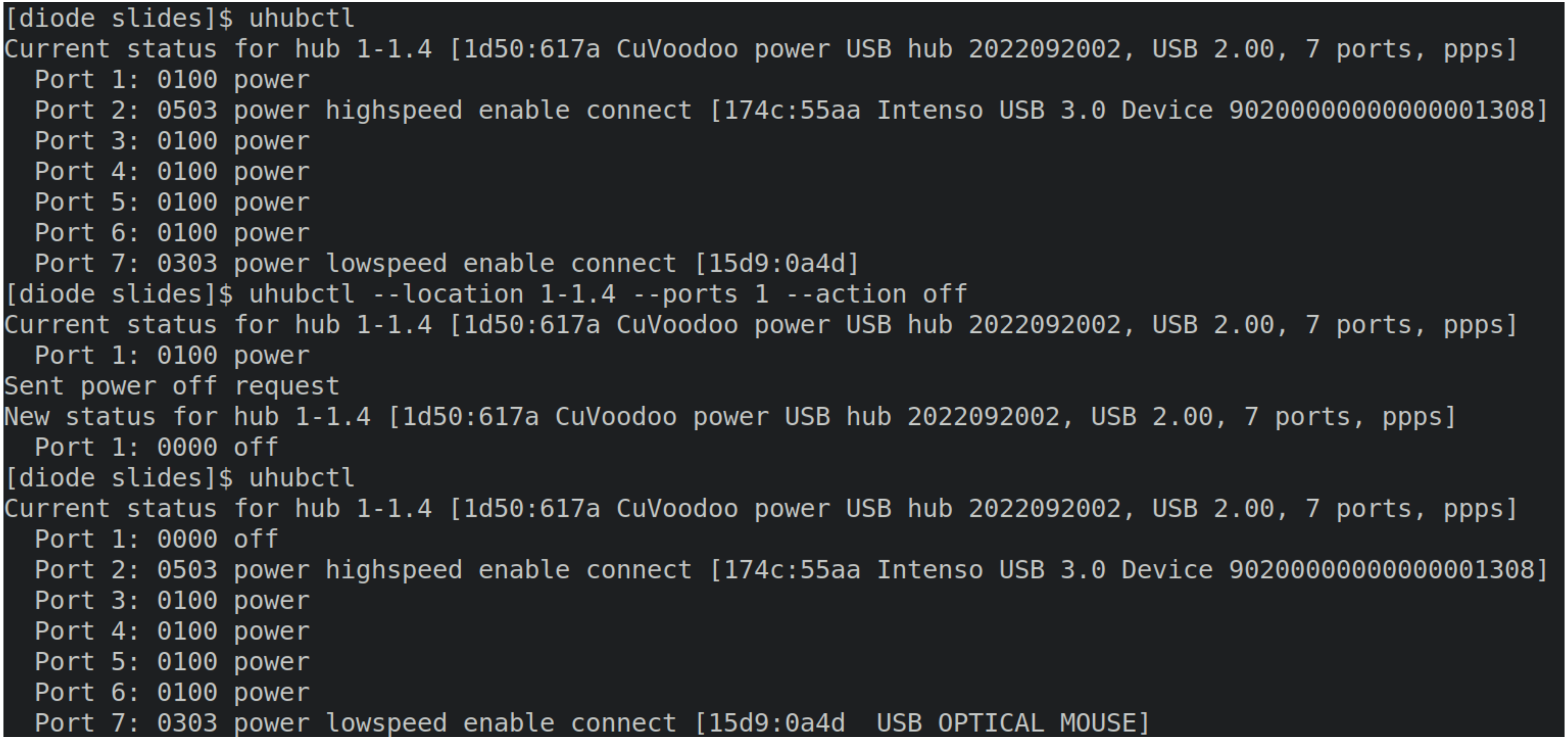
CuVoodoo USB hub

12V 2A power supplies are a lot more common than 5V 3A

hub requirements

- 7-port flat
- proper routing (impedance controlled differential signals)
- switched 5V power outputs (with indication)
- external power supply
- switched high voltage (6-40V) power outputs
- embedded 5V 3A regulator

hub power control



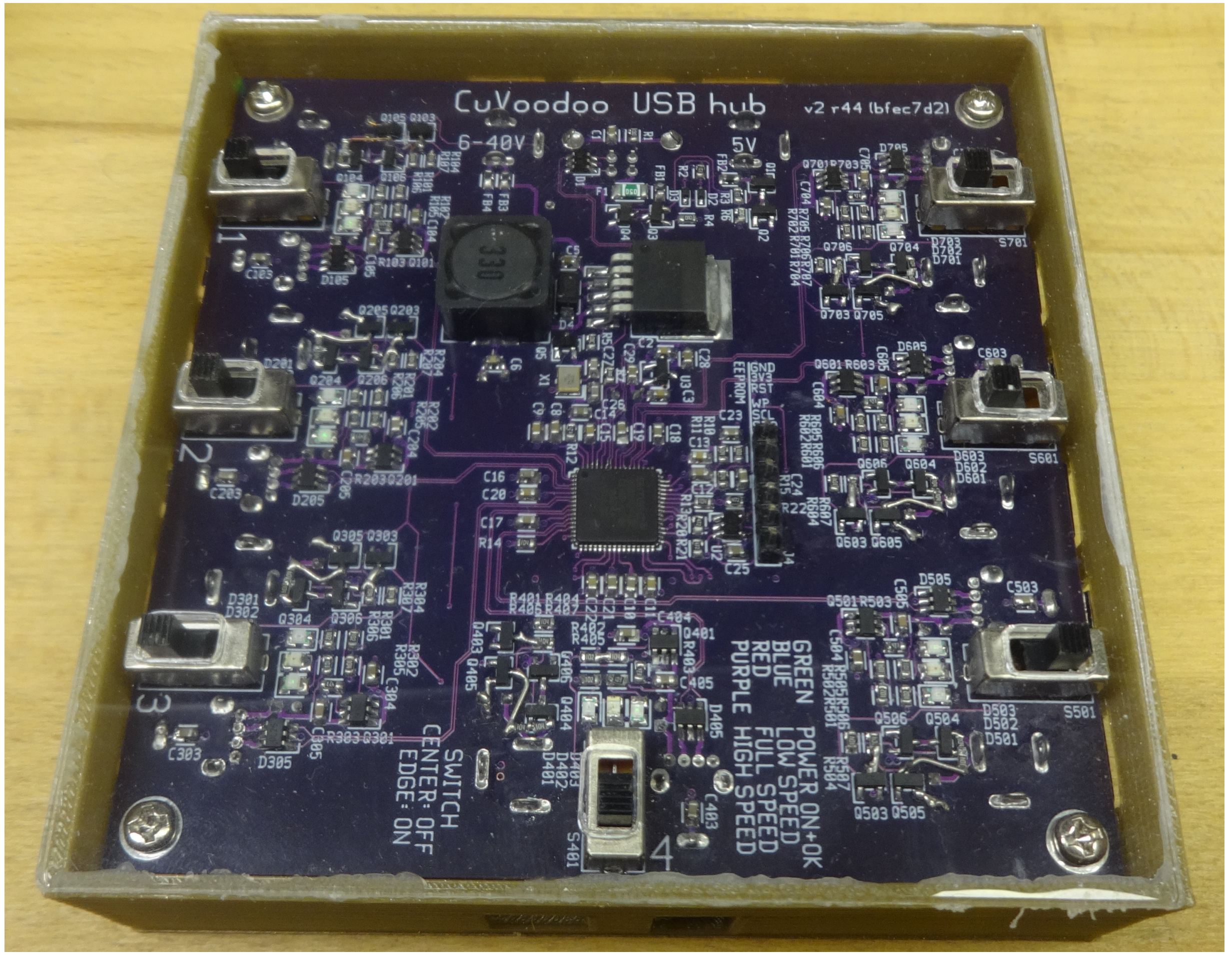
```
[diode slides]$ uhubctl
Current status for hub 1-1.4 [1d50:617a CuVoodoo power USB hub 2022092002, USB 2.00, 7 ports, ppps]
  Port 1: 0100 power
  Port 2: 0503 power highspeed enable connect [174c:55aa Intenso USB 3.0 Device 90200000000000001308]
  Port 3: 0100 power
  Port 4: 0100 power
  Port 5: 0100 power
  Port 6: 0100 power
  Port 7: 0303 power lowspeed enable connect [15d9:0a4d]
[diode slides]$ uhubctl --location 1-1.4 --ports 1 --action off
Current status for hub 1-1.4 [1d50:617a CuVoodoo power USB hub 2022092002, USB 2.00, 7 ports, ppps]
  Port 1: 0100 power
Sent power off request
New status for hub 1-1.4 [1d50:617a CuVoodoo power USB hub 2022092002, USB 2.00, 7 ports, ppps]
  Port 1: 0000 off
[diode slides]$ uhubctl
Current status for hub 1-1.4 [1d50:617a CuVoodoo power USB hub 2022092002, USB 2.00, 7 ports, ppps]
  Port 1: 0000 off
  Port 2: 0503 power highspeed enable connect [174c:55aa Intenso USB 3.0 Device 90200000000000001308]
  Port 3: 0100 power
  Port 4: 0100 power
  Port 5: 0100 power
  Port 6: 0100 power
  Port 7: 0303 power lowspeed enable connect [15d9:0a4d  USB OPTICAL MOUSE]
```

`uhubctl` allows to remotely switch power on the power

hub requirements

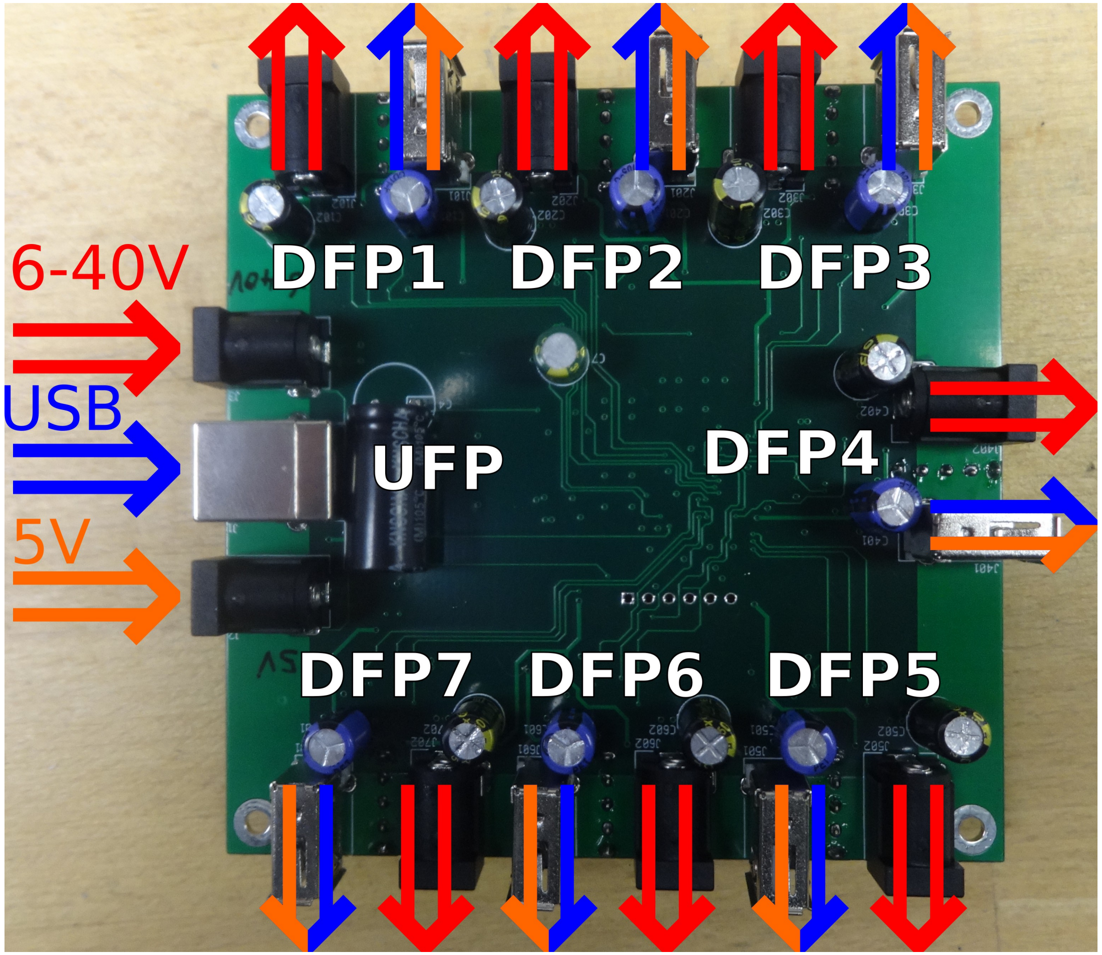
- 7-port flat
- proper routing (impedance controlled differential signals)
- switched 5V power outputs (with indication)
- external power supply
- switched high voltage (6-40V) power outputs
- embedded 5V 3A regulator
- (individual) remotely controlled power outputs (hub feature)

CuVoodoo USB hub

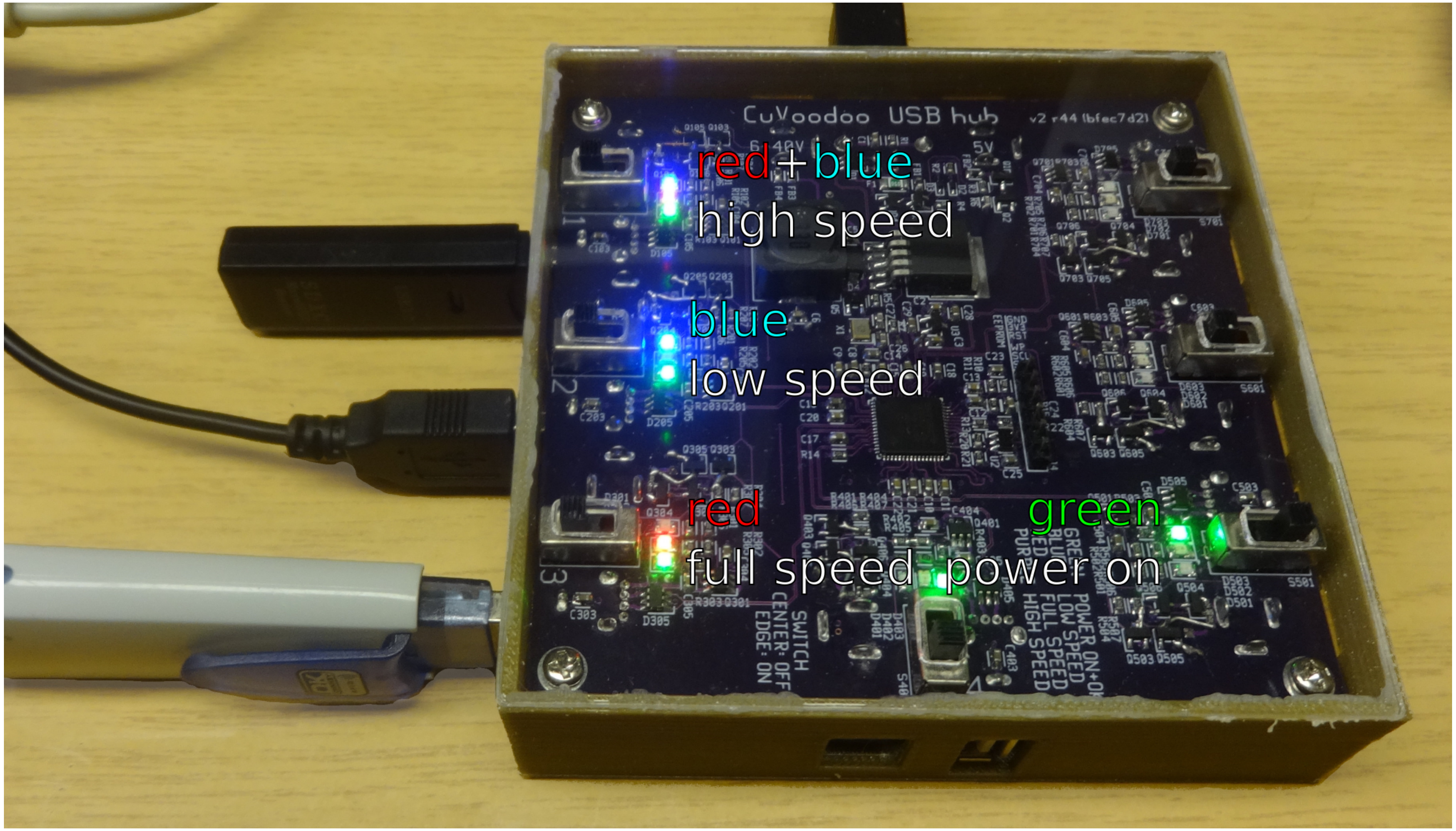


my own USB hub

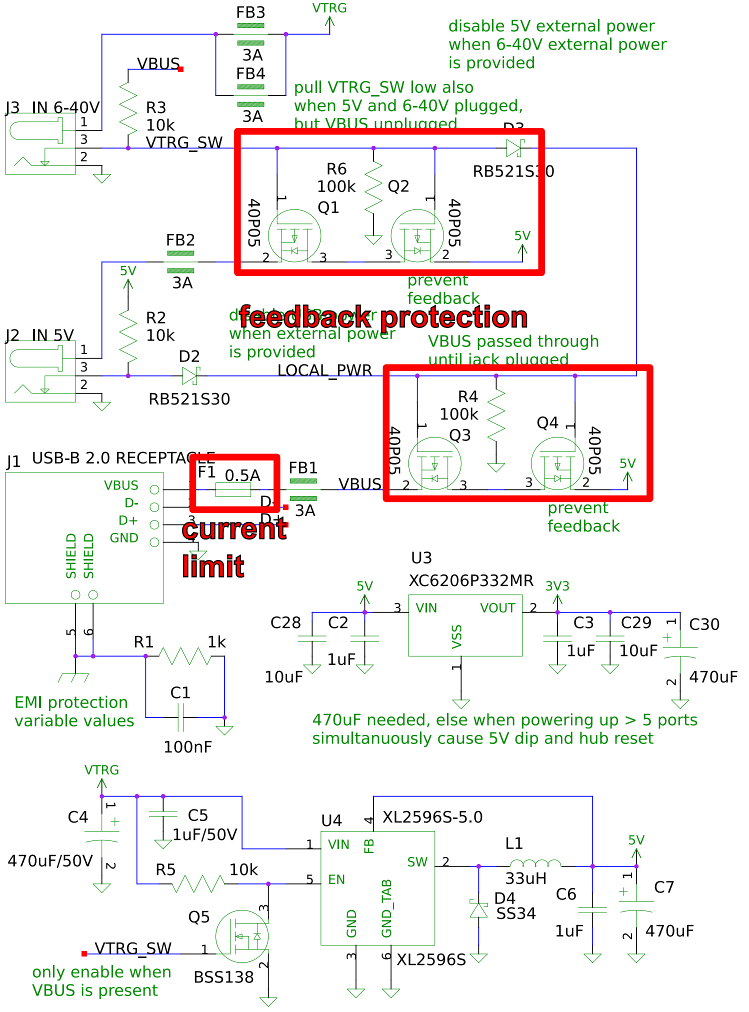
CuVoodoo USB hub



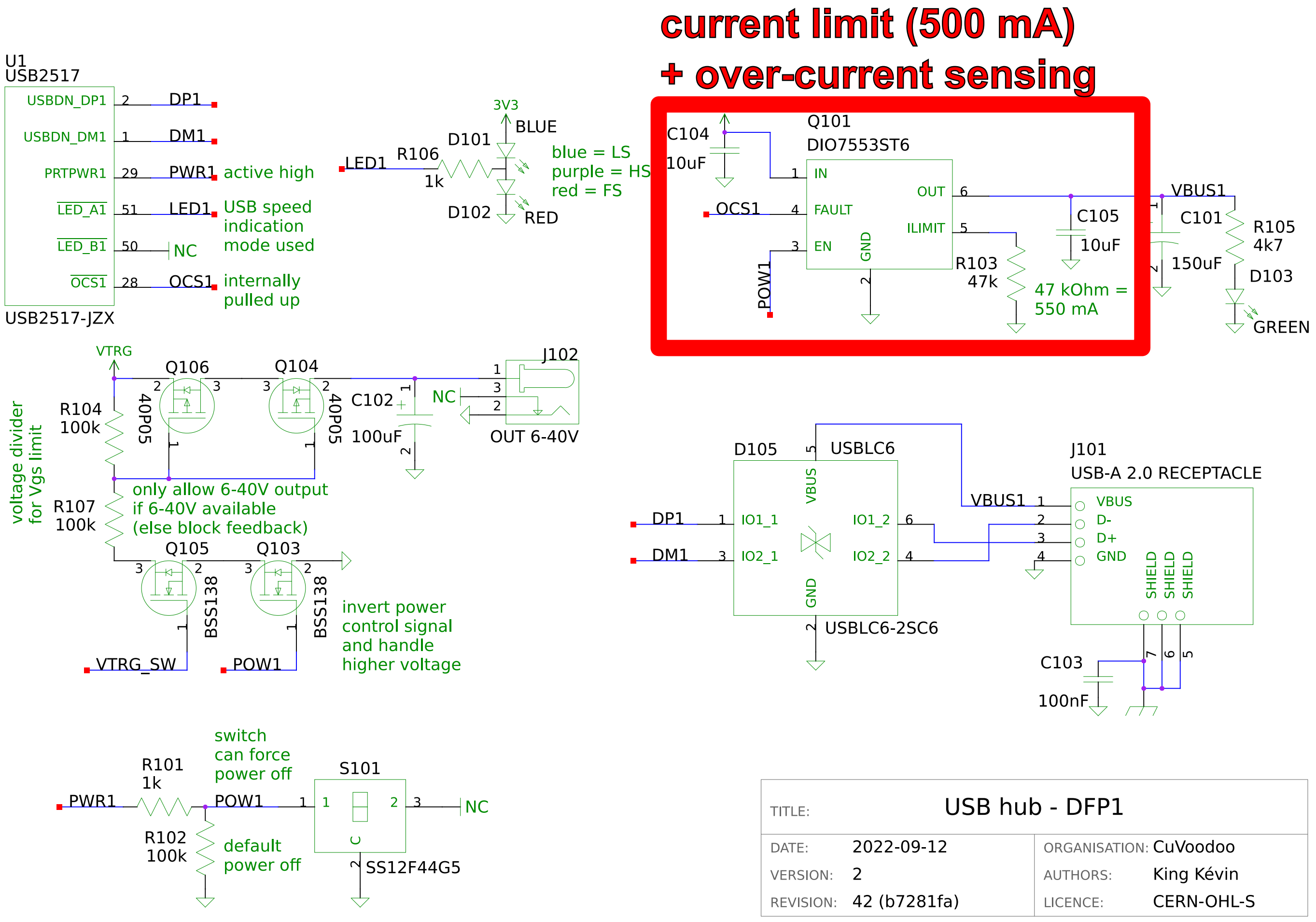
speed indication



power input protection



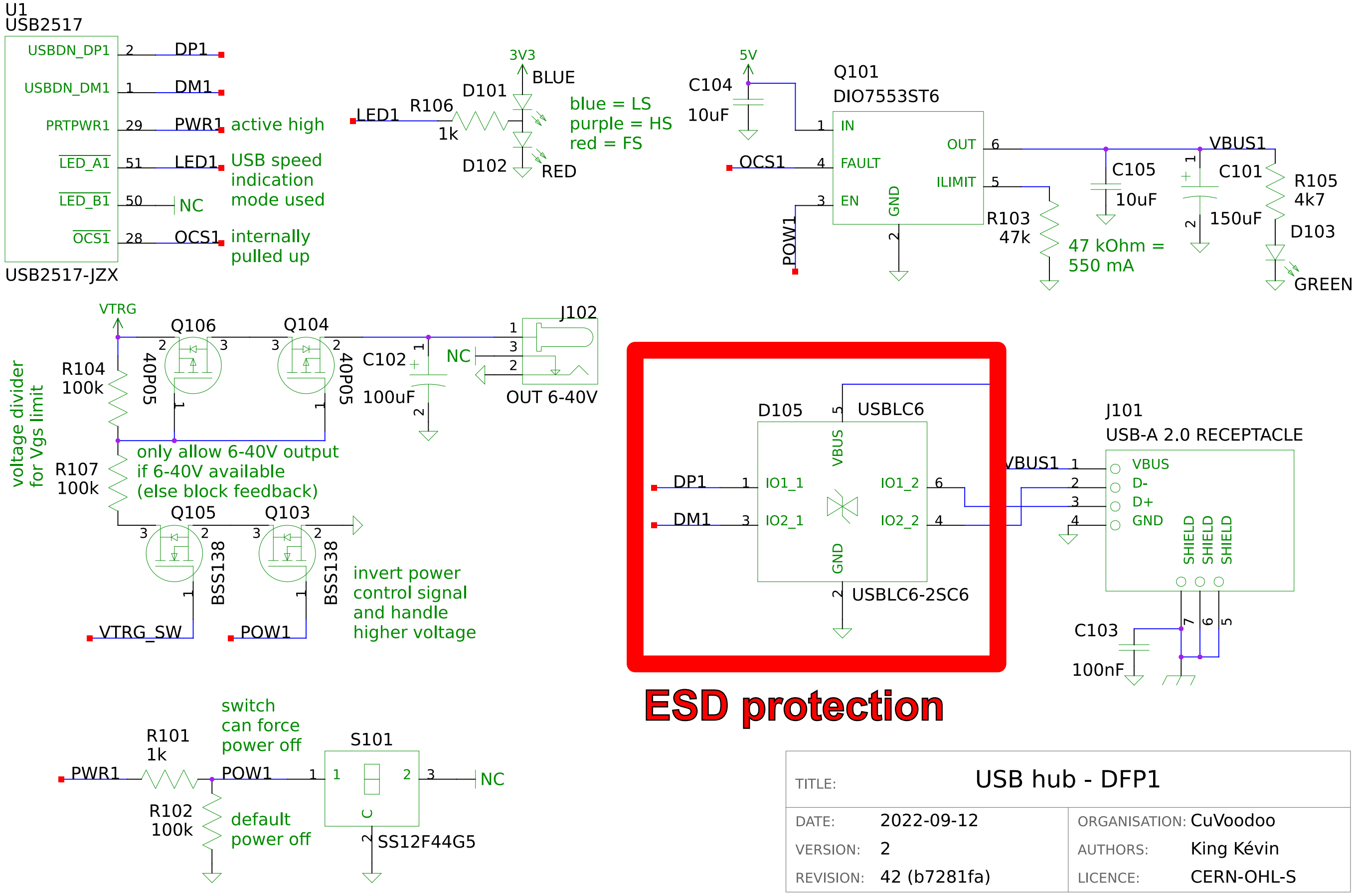
power output protection



current limit (500 mA)
+ over-current sensing

TITLE:	USB hub - DFP1		
DATE:	2022-09-12	ORGANISATION:	CuVoodoo
VERSION:	2	AUTHORS:	King Kévin
REVISION:	42 (b7281fa)	LICENCE:	CERN-OHL-S

ESD protection

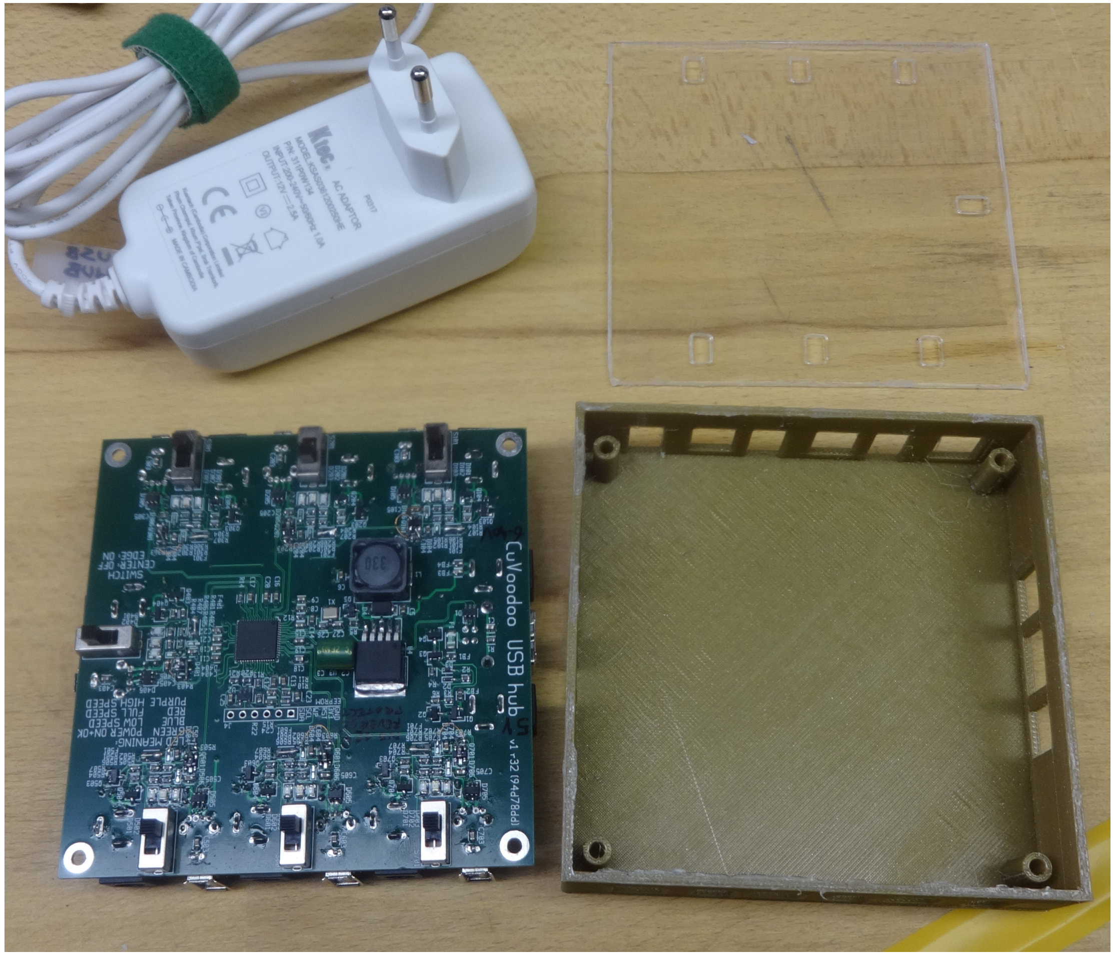


TITLE:	USB hub - DFP1		
DATE:	2022-09-12	ORGANISATION:	CuVoodoo
VERSION:	2	AUTHORS:	King Kévin
REVISION:	42 (b7281fa)	LICENCE:	CERN-OHL-S

hub features

- 7-port flat
- proper routing (impedance controlled differential signals)
- switched 5V power outputs (with indication)
- external power supply (5V, 6-40V)
- switched high voltage (6-40V) power outputs
- embedded 5V 3A regulator
- (individual) remotely controlled power outputs (hub feature)
- speed indication
- power input protection (over-current and feedback)
- (individual) power output protection (over-current and sensing)
- ESD protection (on UFP and all DFP)

CuVoodoo USB hub



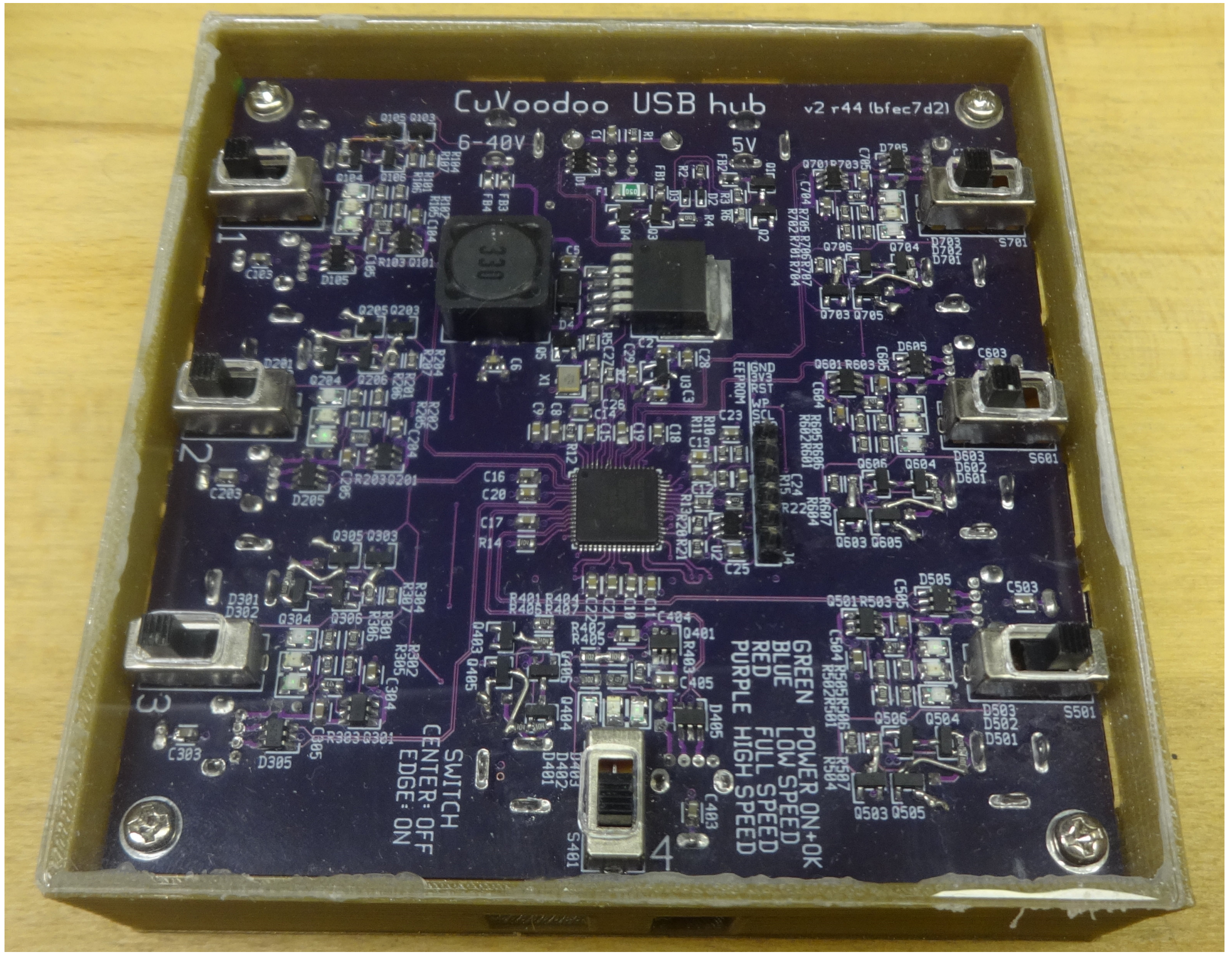
limitations

- not very compact
- USB 2.0 only (I did not need more)
- not USB-C
- over-current not reported (hub chip bug)
- not available in shops

but it’s open source, and you can build your own: https://git.cuvoodoo.info/usb_hub

(check for releases to get the fabrication files)

questions?



FAQ