EVEN MORE

USB HACKING WITH USB-TOOLS

KATE TEMKIN • MIKAELA SZEKELY

TOORCON-21 2019

Katherine/Kate Temkin (@ktemkin):

- software lead, Great Scott Gadgets
- glitch witch & open-source-tool-builder
- educational (reverse) engineer
- lauded by the Daily Mail as a “cyber criminal”

Mikaela Szekely (@Qyriad):

- student, and yet master*
- got a bit too deep in some open-source USB stuff
- apparently better at cybercrime (not caught by the Daily Mail)

SO, WHO *ARE* YOU?

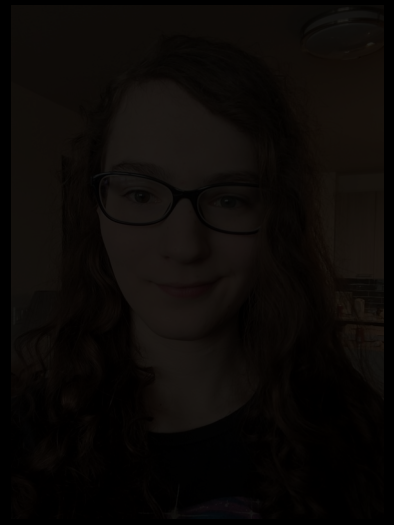
Katherine/Kate Temkin (@ktemkin):

- software lead, Great Scott Gadgets
- glitch witch & open-source-tool-builder
- educational (reverse) engineer
- lauded by the Daily Mail as a "cybercriminal"

WE DON'T HAVE TIME!

Mikaela Szekely (@Qyriad):

- student, and yet master*
- got a bit too deep in some open-source USB stuff
- apparently better at cybercrime (not caught by the Daily Mail)



← Kate

Mikaela →

SO: WHAT ARE USB-TOOLS?

USB Hacking Tools

A set of USB hacking tools from **@ktemkin**, **@Qyriad**, **@greatscottgadgets**, and co. See also **@hacking-usb** for educational materials..

https://discord.gg/HKAhHub usb@ktemkin.com

Find a repository… | Type: **All** | Language: **All** | Customize pins | **New**

ViewSB

open-source USB analyzer toolkit with support for a variety of capture hardware

Python 9 77 7 0 Updated 3 hours ago

nu-map

Forked from nccgroup/umap2

nü-map: a somewhat-more-modern (expeirmental) derivative of umap2 for modern FaceDancer

Python AGPL-3.0 39 5 1 2 Updated 14 days ago

pyfwup

Python FirmWare UploaderPy -- a DFU (and similar) utility for python

Python 0 2 0 0 Updated 19 days ago

Facedancer

modern FaceDancer core for multiple devices-- including GreatFET

Python BSD-3-Clause 32 227 8 0 Updated on Sep 21

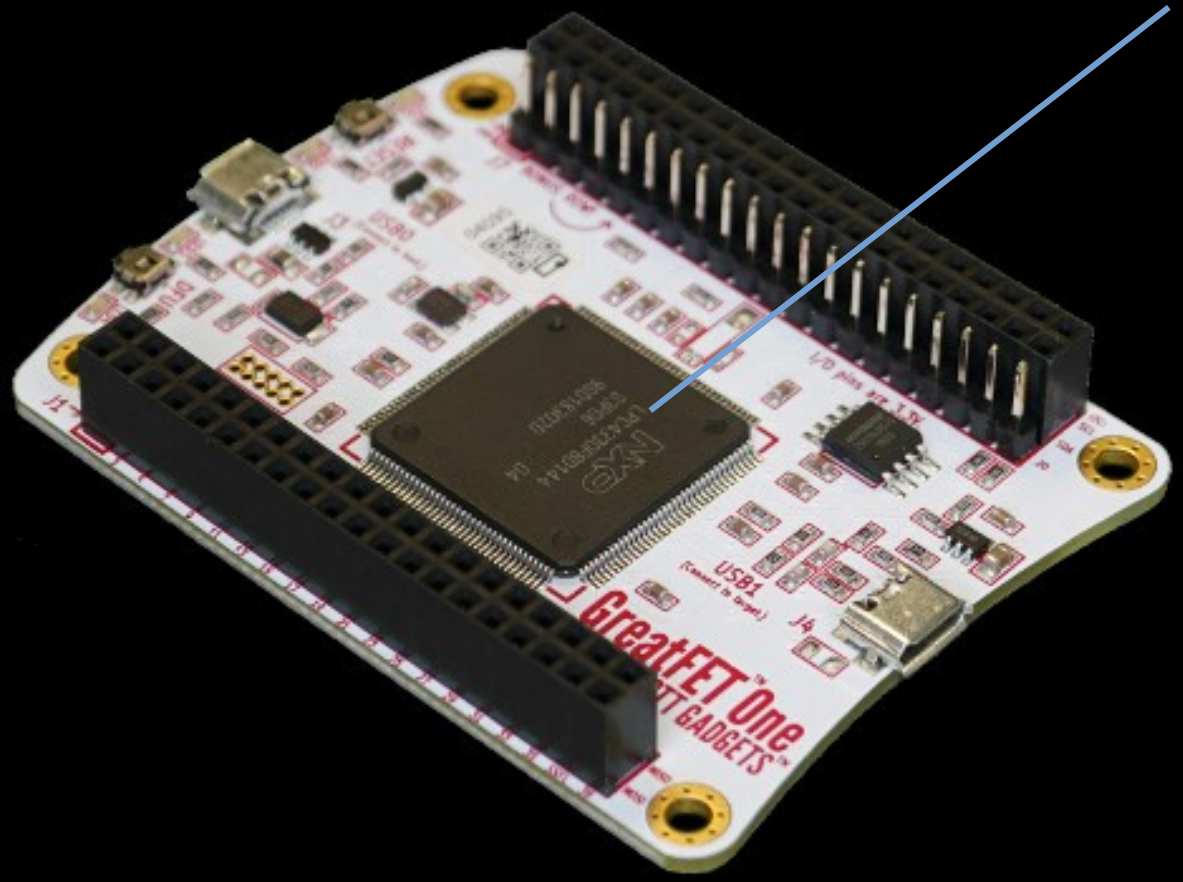
Top languages

Python C++ C HTML

People 4 >

Invite someone

https://github.com/usb-tools



What's new with FaceDancer?

- Not too much – mostly that more hardware platforms are actually available for purchase.

What's coming down the line?

- Soon enough™: new asyncio-driven model
- Mid-term: support for Linux UDC backends
- Longer term: FPGA-based extensions

Dev	EP	Len	D	Packet			
				descriptor #0			configuration descriptor
40	0	2	↩	⊞ requesting 2 bytes of string	ACK		Length 9
				descriptor #2			Descriptor Type confi
40	0	18	↩	⊞ requesting 18 bytes of string	ACK	GreatFET	Length including 32
				descriptor #2			subordinates
40	0	2	↩	⊞ requesting 2 bytes of string	ACK		Interface count 1
				descriptor #1			Configuration number 1
40	0	40	↩	⊞ requesting 40 bytes of string	ACK	Great Scott Gadgets	Description string 0
				descriptor #1			Attributes 128
40	0	2	↩	⊞ requesting 2 bytes of string	ACK		Max power Consumption 250
				descriptor #3			
40	0	66	↩	⊟ requesting 66 bytes of string	ACK	0000000000000000d1c466e631873113	interface
				descriptor #3			Length 9
40	0	8	↪	⊟ control request setup transfer	ACK	value=0303 index=0409 length=0042	Descriptor Type inter
				for IN request			Interface number 0
40	0	2	↪	· SETUP token		address=64, endpoint=0x00,	Alternate setting 0
						direction=0	Endpoints included 2
--	--	8		· 8 bytes; DATA0		80 06 03 03 09 04 42 00	Class 255
--	--	0		· ACK			Subclass 255
40	0	66	↩	⊟ 66B IN transfer	ACK	42 03 30 00 30 00 30 00...	Protocol 255
40	0		↩	⊟ IN packet	NAK		String index 0
40	0	2	↩	· IN token		address=64, endpoint=0x00,	
						direction=1	endpoint
--	--	0		· NAK			Length 7
40	0	64	↩	⊞ IN packet	ACK	42 03 30 00 30 00 30 00...	Descriptor Type endpo
40	0	2	↩	⊞ IN packet	ACK	33 00	Interface number 129
40	0	0	↪	⊞ data-less OUT transfer	ACK		
40	0	9	↩	⊞ requesting 9 bytes of	ACK	1 interface	09 02 20 00 01 01 00 80 .....
				configuration descriptor #0			
40	0	32	↩	⊞ requesting 32 bytes	ACK	1 interface	09 02 20 00 01 01 00 80 .....
				configuration descriptor #0			09 04 00 00 02 ff ff ff .....

https://github.com/usb-tools/viewsb (TUI shown

VIEWSB: OPEN-SOURCE USB ANALYSIS

BUT ISN’T THAT EXPENSIVE?

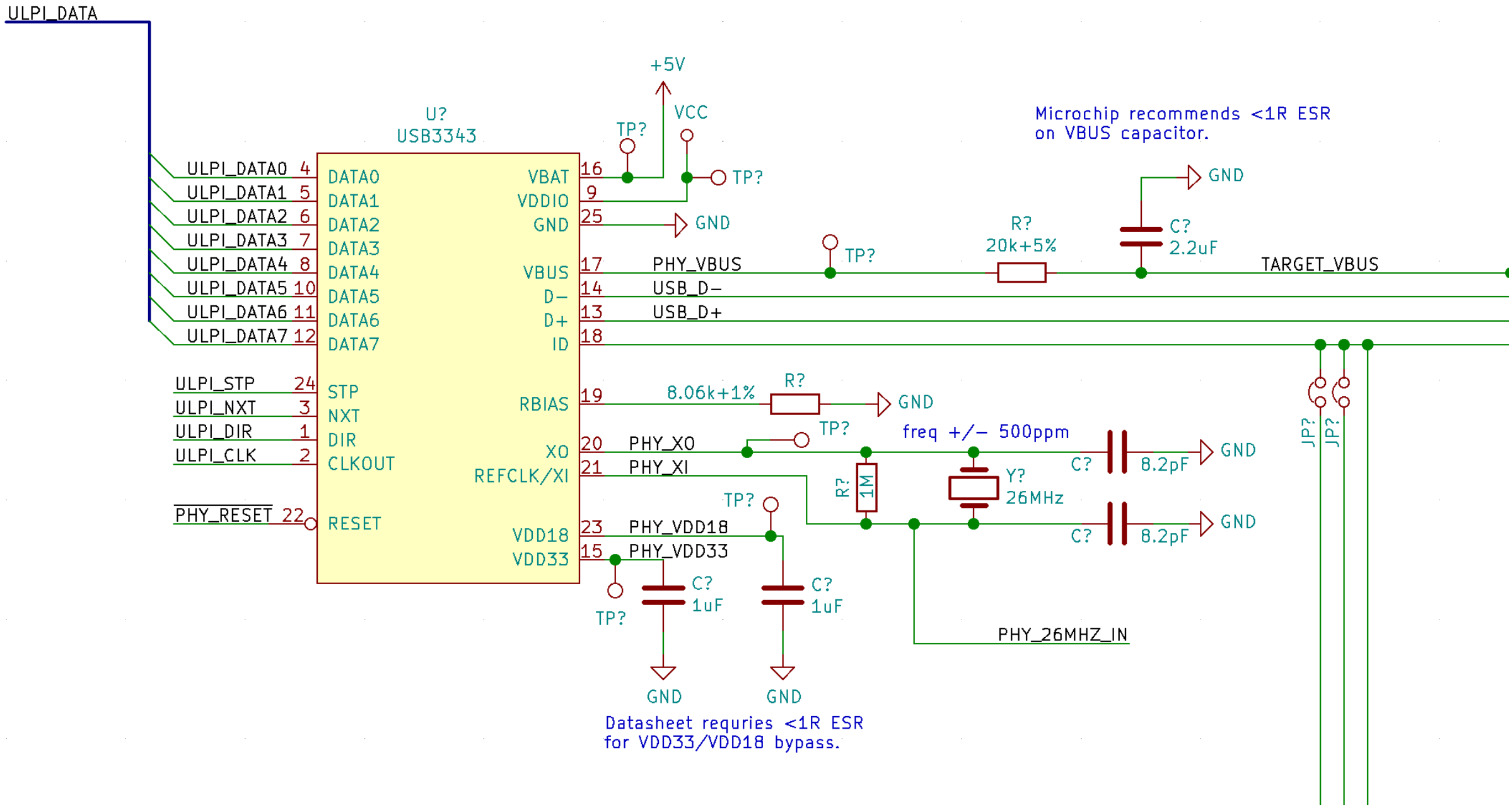
[software analysis demo]

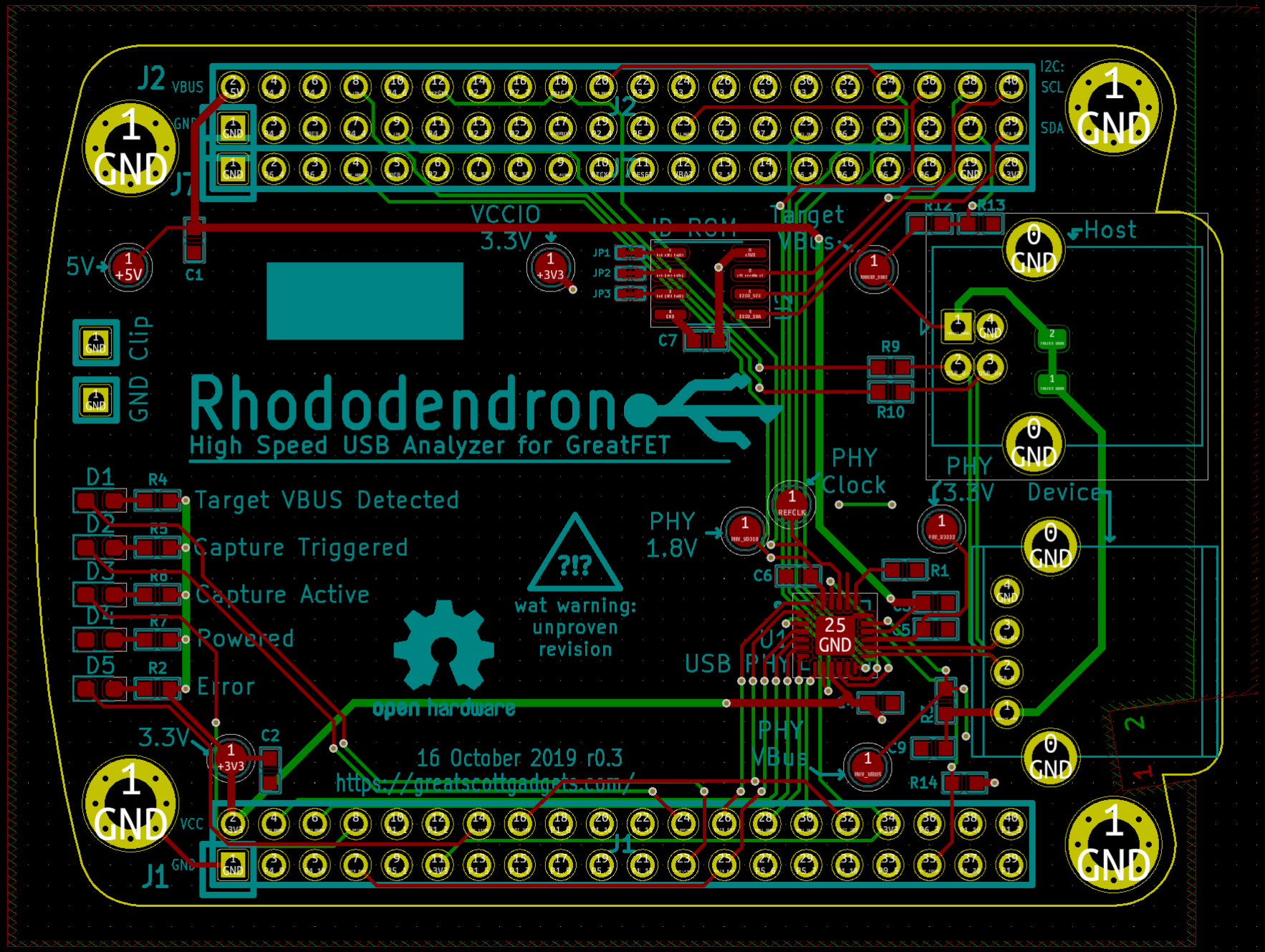
OKAY, BUT THAT'S KINDA LIMITED

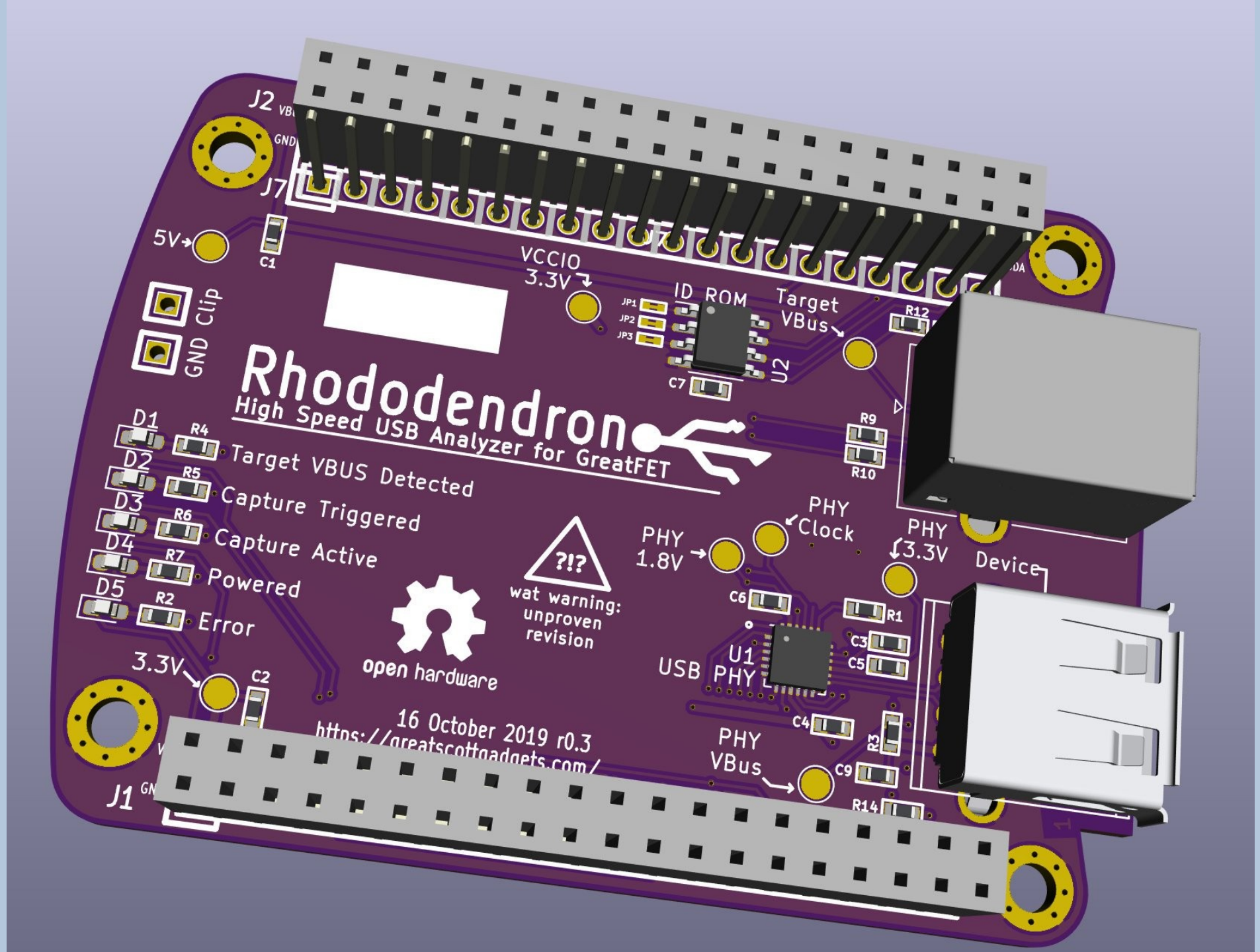
RECIPE FOR A **HIGH SPEED ANALYZER**

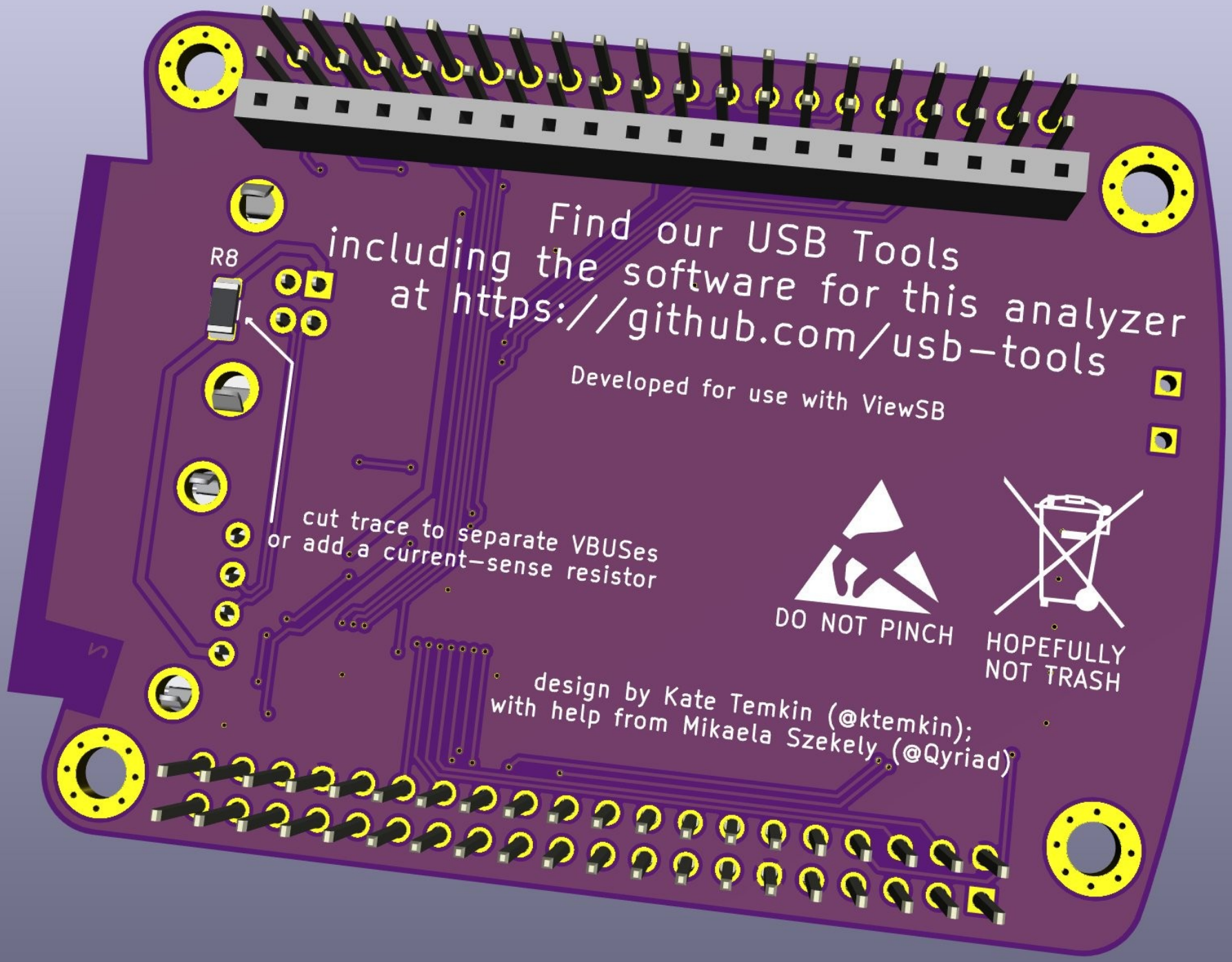
Components:

- LPC43xx; or similar, [or an ultra-cheap FPGA like the ECP5-12F]
- SDRAM for packet buffering*
- ULPI USB PHY
- SPI Flash


ULPI_DATA
U?
USB3343
ULPI_DATA0 4
ULPI_DATA1 5
ULPI_DATA2 6
ULPI_DATA3 7
ULPI_DATA4 8
ULPI_DATA5 10
ULPI_DATA6 11
ULPI_DATA7 12
ULPI_STP 24
ULPI_NXT 3
ULPI_DIR 1
ULPI_CLK 2
PHY_RESET 22
DATA0
DATA1
DATA2
DATA3
DATA4
DATA5
DATA6
DATA7
STP
NXT
DIR
CLKOUT
RESET
VBAT 16
VDDIO 9
GND 25
VBUS 17
D− 14
D+ 13
ID 18
RBIAS 19
XO 20
REFCLK/XI 21
VDD18 23
VDD33 15
+5V
VCC
TP?
GND
PHY_VBUS
USB_D−
USB_D+
Microchip recommends <1R ESR
on VBUS capacitor.
R?
20k+5%
C?
2.2uF
TARGET_VBUS
8.06k+1%
PHY_XO
PHY_XI
freq +/− 500ppm
Y?
26MHz
C?
8.2pF
R?
1M
PHY_VDD18
PHY_VDD33
C?
1uF
PHY_26MHZ_IN
JP?
Datasheet requries <1R ESR
for VDD33/VDD18 bypass.


Rhododendron
High Speed USB Analyzer for GreatFET
Target VBUS Detected
Capture Triggered
Capture Active
Powered
Error
wat warning:
unproven
revision
open hardware
16 October 2019 r0.3
https://greatscottgadgets.com/
VCCIO
3.3V
Target
VBUS
Host
Device
PHY
Clock
PHY
1.8V
PHY
3.3V
PHY
VBUS
USB PHY
GND Clip
5V
3.3V
VBUS
GND
VCC
I2C:
SCL
SDA
ID ROM
J1
J2
J7
U1
C1
C2
C6
C7
C9
R1
R2
R4
R5
R6
R7
R9
R10
R12
R13
R14
D1
D2
D3
D4
D5
JP1
JP2
JP3


Rhododendron
High Speed USB Analyzer for GreatFET
Target VBUS Detected
Capture Triggered
Capture Active
Powered
Error
wat warning:
unproven
revision
open hardware
16 October 2019 r0.3
VCCIO
3.3V
ID ROM
Target
VBus
PHY
Clock
PHY
1.8V
PHY
3.3V
Device
U1
USB PHY
PHY
VBus
5V
GND
Clip
3.3V
J1
J2
J7
U2
D1
D2
D3
D4
D5
R4
R5
R6
R7
R2
R12
R9
R10
R1
R3
R14
C1
C2
C3
C4
C5
C6
C7
C9
JP1
JP2
JP3


R8
Find our USB Tools
including the software for this analyzer
at https://github.com/usb-tools
Developed for use with ViewSB
cut trace to separate VBUSes
or add a current-sense resistor
DO NOT PINCH
HOPEFULLY
NOT TRASH
design by Kate Temkin (@ktemkin);
with help from Mikaela Szekely (@Qyriad)

[rhododendron demo]

[rhododemodron]

IN-PROGRESS TOOLS: **NUMAP**

What is it now?

- port of umap2 to the modern FaceDancer backend; which provides some fancy host-fuzzing via FaceDancer emulation
- very much a work in progress
- ~~a subtle dig at Dominic's röck döts~~

What should it be?

- a much more comprehensive tool for host and device fuzzing
- a tool with original-umap style host identification; and host driver ID'ing
- ~~functional~~

OTHER USB-TOOLS:

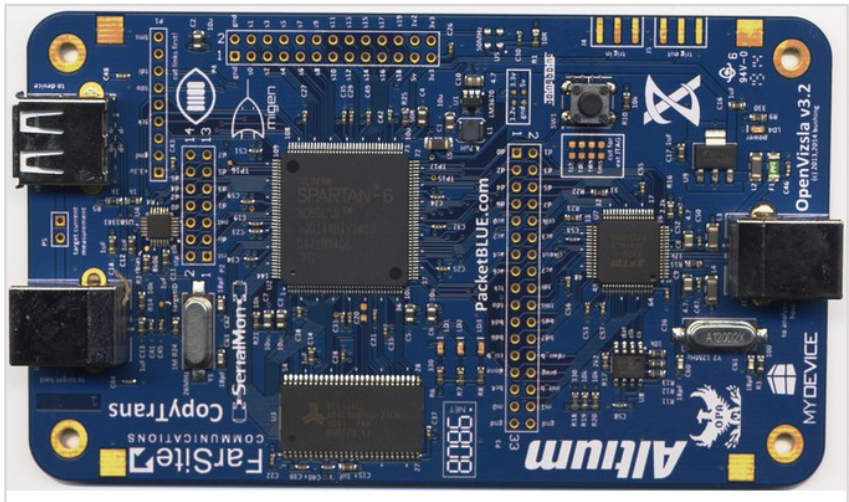
Primary tools:

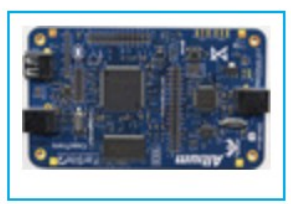
- FaceDancer (and USBProxy)
- ViewSB

Supporting tools:

- pyfwup – a tool for upgrading device firmware in pure python
- pyopenvizsla – properly-pythonic OpenVizsla support drivers
  - ...openvizsla?

IMPORTANT NOTE: No orders will be handled from June 25, 2019 until July 7, 2019. Orders placed until 10:00am German local time (CEST) on June 24th will still be handled+shipped before this period of absence.





OpenVizsla v3.2 USB Protocol Analyzer PCBA

This is fully assembled and tested OpenVizsla v3.2 USB protocol analyzer.

OpenVizsla is a bus sniffer/analyzer for USB. It allows you to passively monitor the communication between a USB host and USB periheral. It supports USB low-speed, full-speed and high-speed.

The product is shipped as a bare printed circuit board assembly, without any enclosure.

For more information about OpenVizsla, see https://openvizsla.org/

PRICE

119.00 € (inc. VAT)

1 | Add To Cart

LOOK FOR SIMILAR ITEMS

Development Boards

See also our open course materials:

https://usbc.tf
https://mini.usbc.tf

https://github.com/hacking-usb

QUESTIONS?