

# Introducing Serberus

The Multi-headed embedded hacking tool. <https://github.com/pk-mdt/Serberus>

Patrick Kiley

Principal Red Team Consultant

# Who I Am

- Principal Consultant – Mandiant part of Google Cloud
- 20 years of information security experience
- 12 years pen testing
- 7 years focus on embedded systems
- Professionally know for bricking things

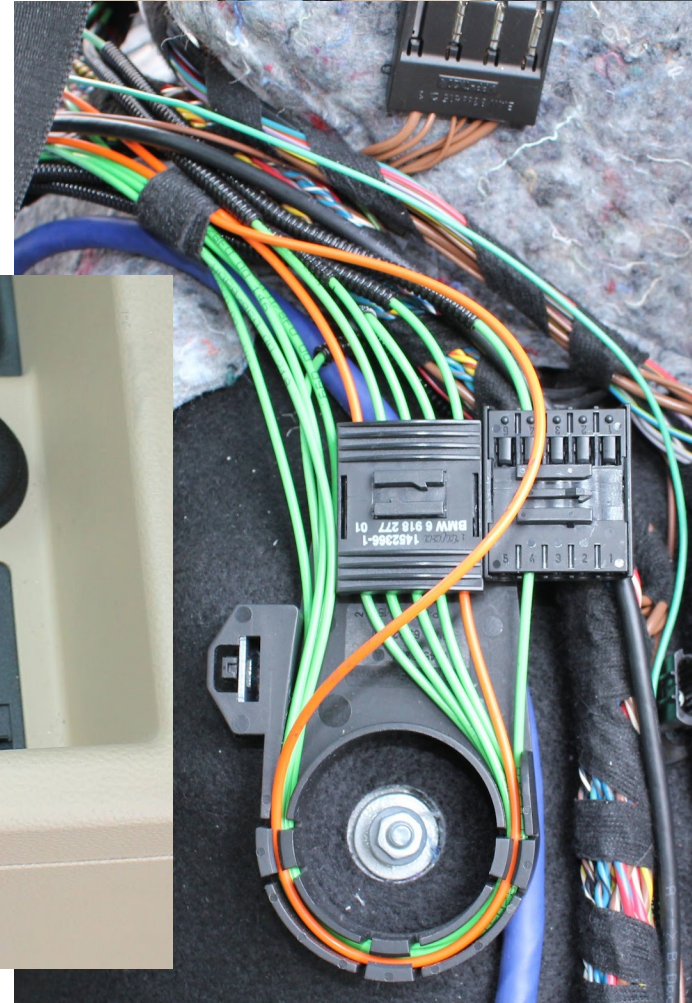




# BMW – First Brick

I Don't judge, I liked the way it drove and the turn signals even worked

Car was without an iPod  
USB adapter.  
Decided to add one  
Had to splice into the  
MOST loop (fiber optic  
media transport)  
Bricked when re-coding





# Tesla

Bricked one of these too

Dropped the battery and upgraded components to make it faster

During the reflash process I broke something and had to have it towed across state lines

Fixed it at home a couple days later





# Avionics hacking

AKA Step-by-step guide to getting on a watchlist

In truth, I spent 2.5 years from finding to release while working with DHS, FAA and industry

<https://www.cisa.gov/news-events/ics-alerts/ics-alert-19-211-01>



# The why of creation



# Why did I make the Serberus

Was messing around with serial bi-directional communication

- Had a 4 port serial to USB module with no level shifters
- Had a Tigard that had level shifters, single port
- Why not both?
  
- Name
  - Combination of Serial Bus and Cerberus, the multi-headed dog





# Why did I make the Serberus

Subhead can go here

Was messing around with serial bi-directional communication

- Had a 4 port serial to USB module with no level shifters
- Had a Tigard that had level shifters, single port



# What did I want to change?

Partially to learn as well

4 Channels

Simplify connections

MSO style ribbon connector

Rotary switch

Tx/Rx indicators

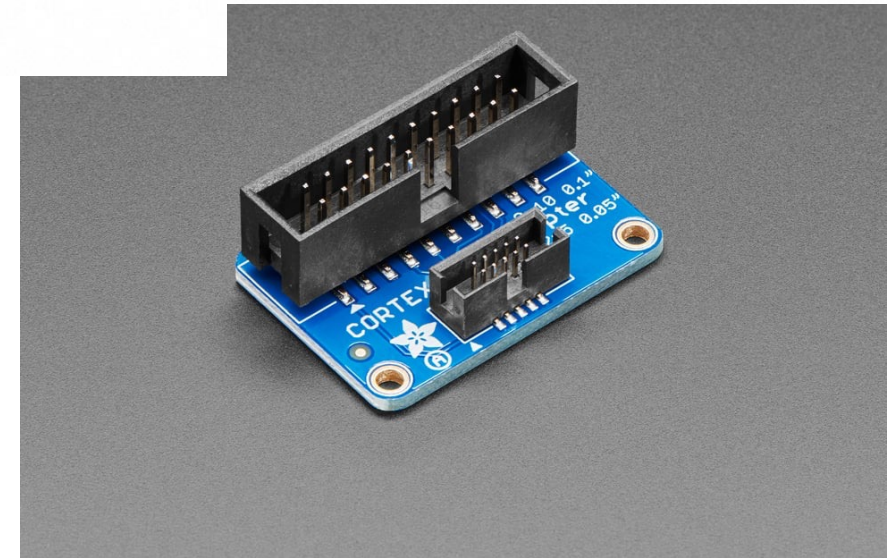
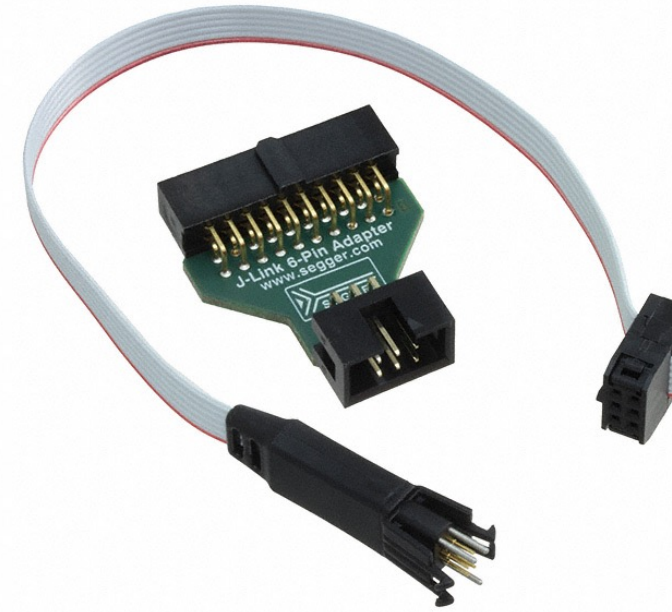
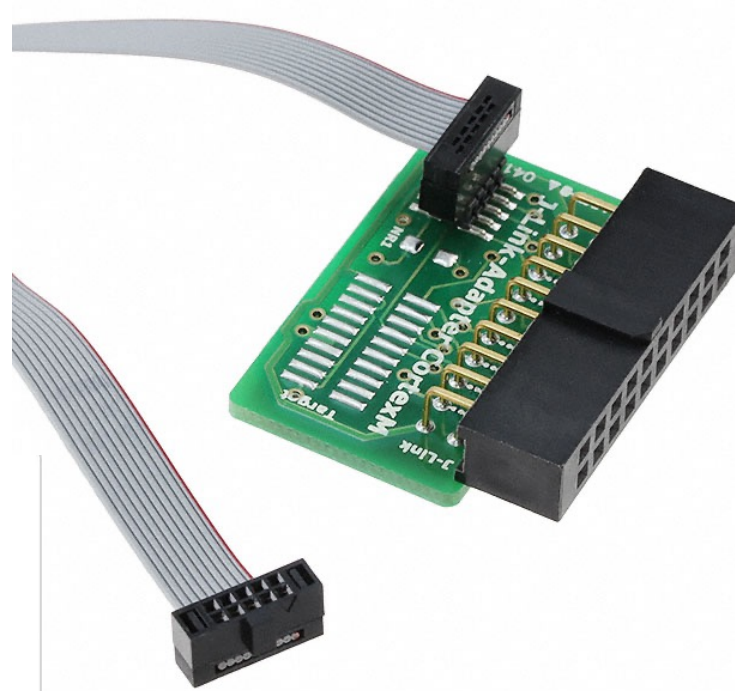
Easi(er) Logic Analyzer connector– Saleae for example



# J-Link Adapters

Why not make it easier?

20 pin adapters



# Tx/Rx Indicators

Present on the 4232 Datasheet

Could not find initially on the 4233 datasheet.

Section was present on later versions.



## 6.4 4 Channel Transmit and Receiver LED Indication Example

The following example illustrates how a 74HC595 can be used to decode the EEDATA data to indicate Tx and Rx on each of the channels. The associated LED will light when the Channel is transmitting or receiving data.

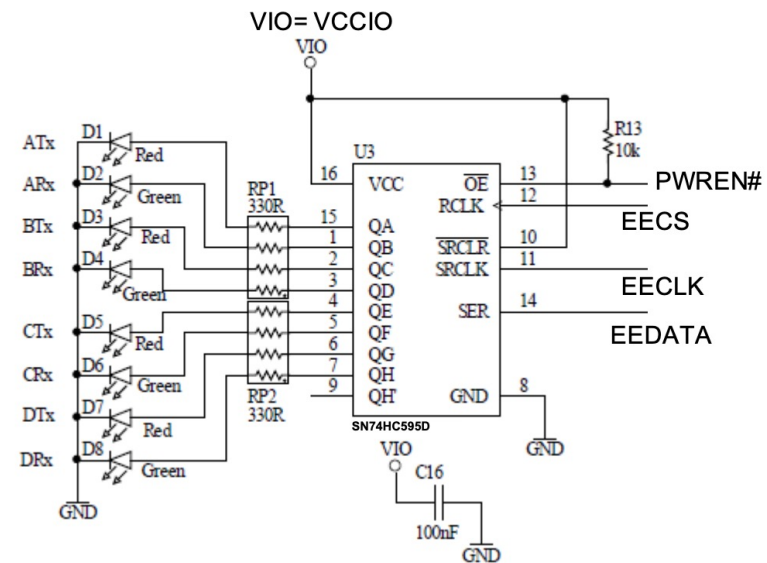


Figure 6.6 Using 74HC595 to Indicate Tx and Rx Data

In this configuration, the LEDs will flash when the EEPROM is accessed e.g. during enumeration.

Under normal operation, the EECS is held low to disable access to the EEPROM. In this special case, the EECLK (frequency = 1.56μs) will clock the EEDATA into the 74HC595 shift register (with EECS low, therefore EEPROM ignores the EEDATA). Then EECS will pulse high. The rising edge of the EECS latches the data into a storage register of the 74HC595 which drives the LEDs.

Please refer to the [74HC595 datasheet](#) for further explanation.

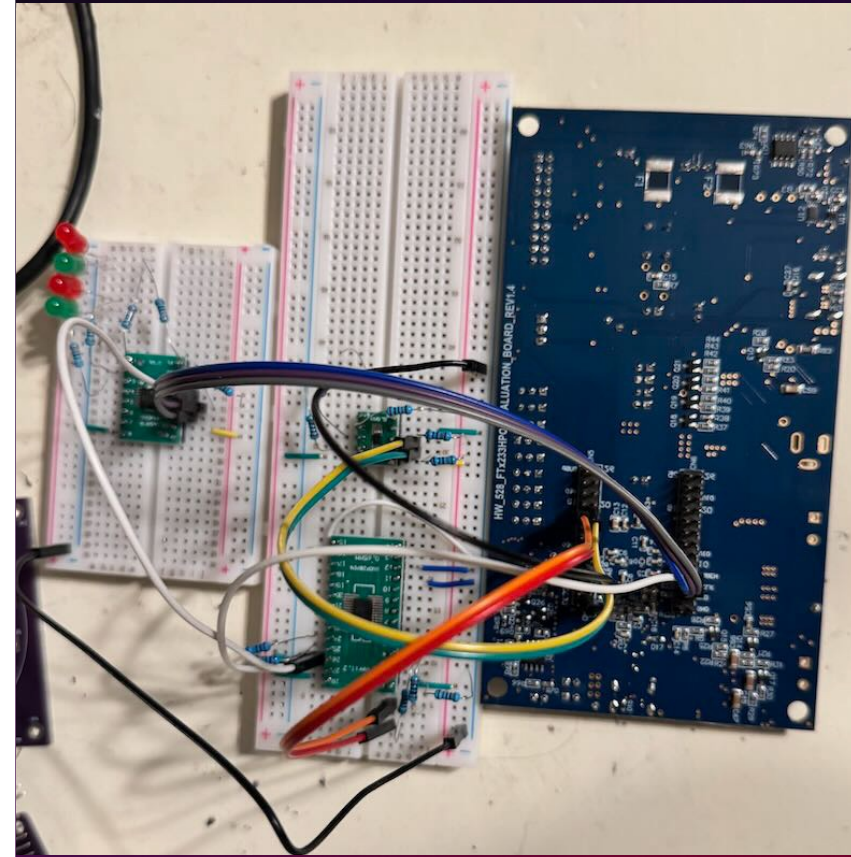


# The process of creation

How I learned to use KiCAD

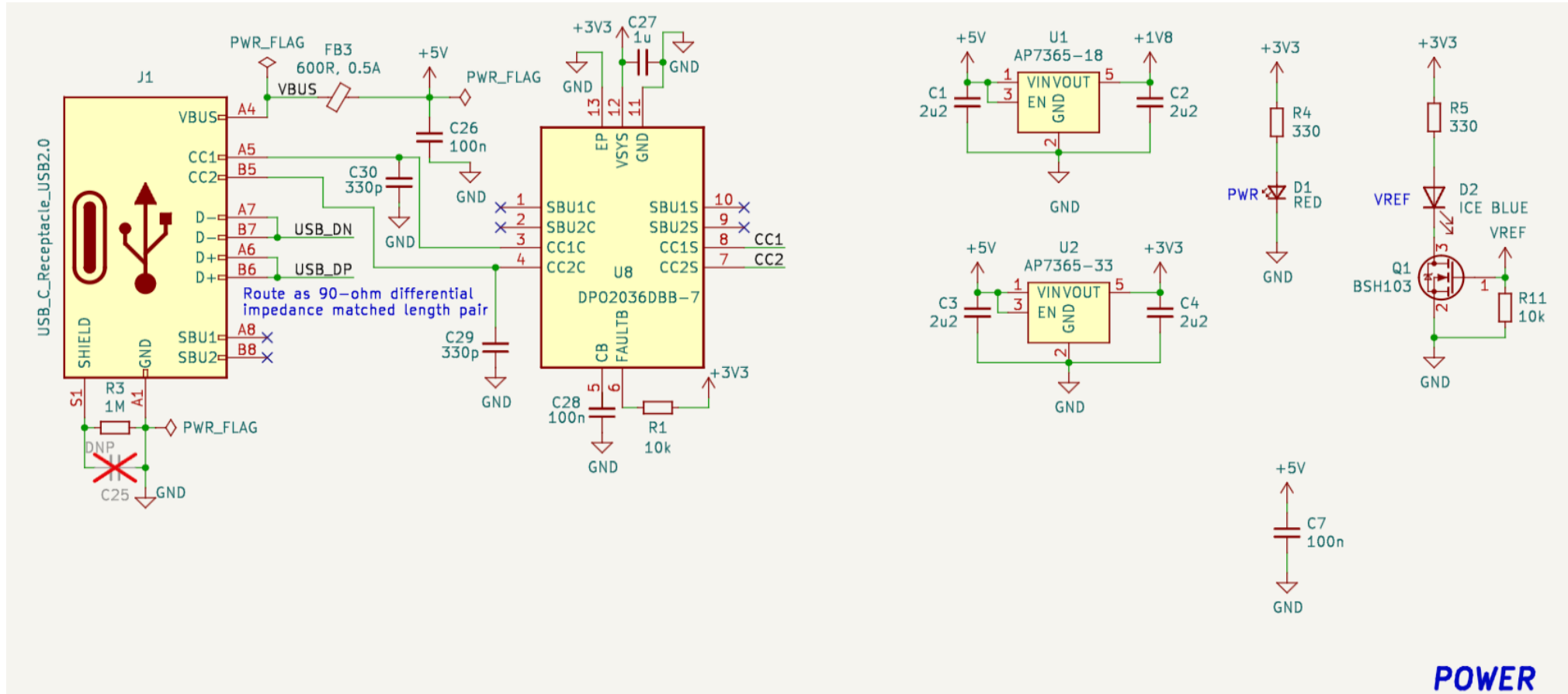
# Proof of Concept

- Tested Tx/Rx indicators for each channel
- Tested basic UART functions
- I should mention I am not an EE, even though I studied it for a minute

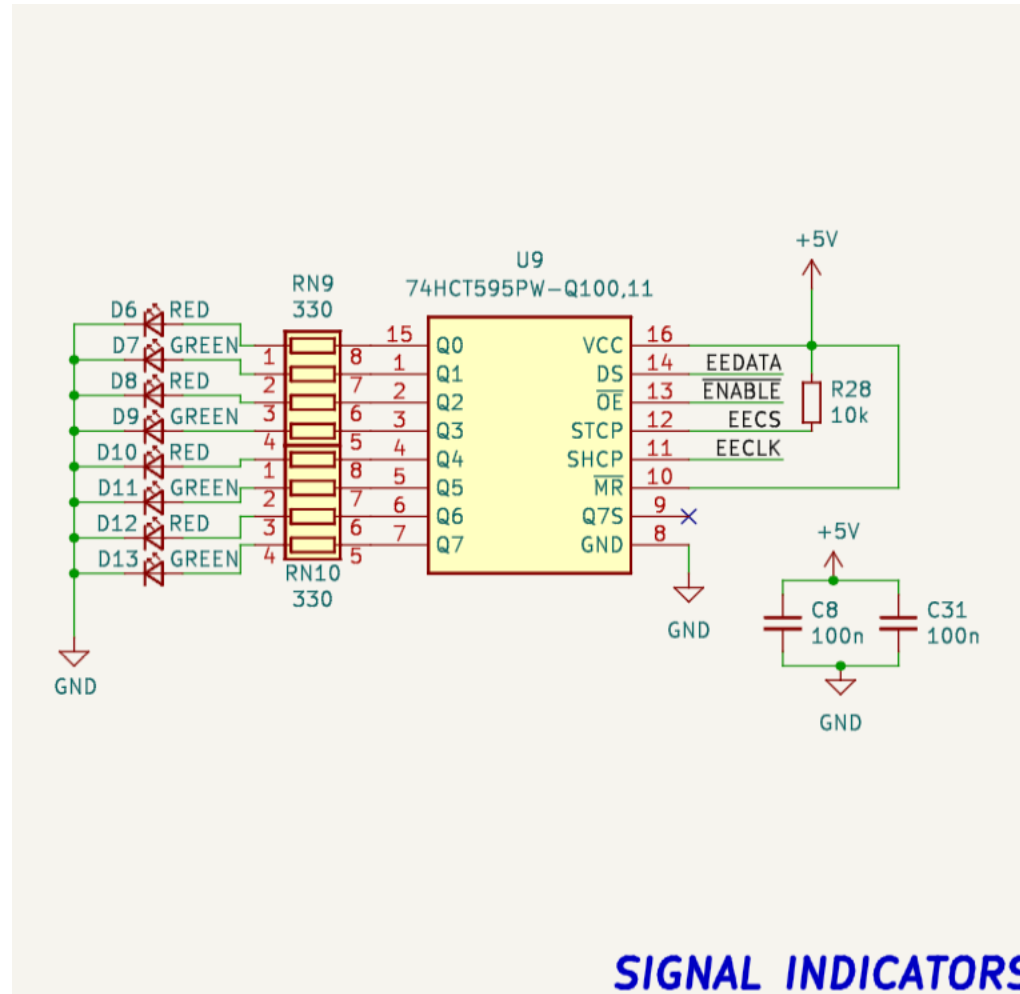




# Power

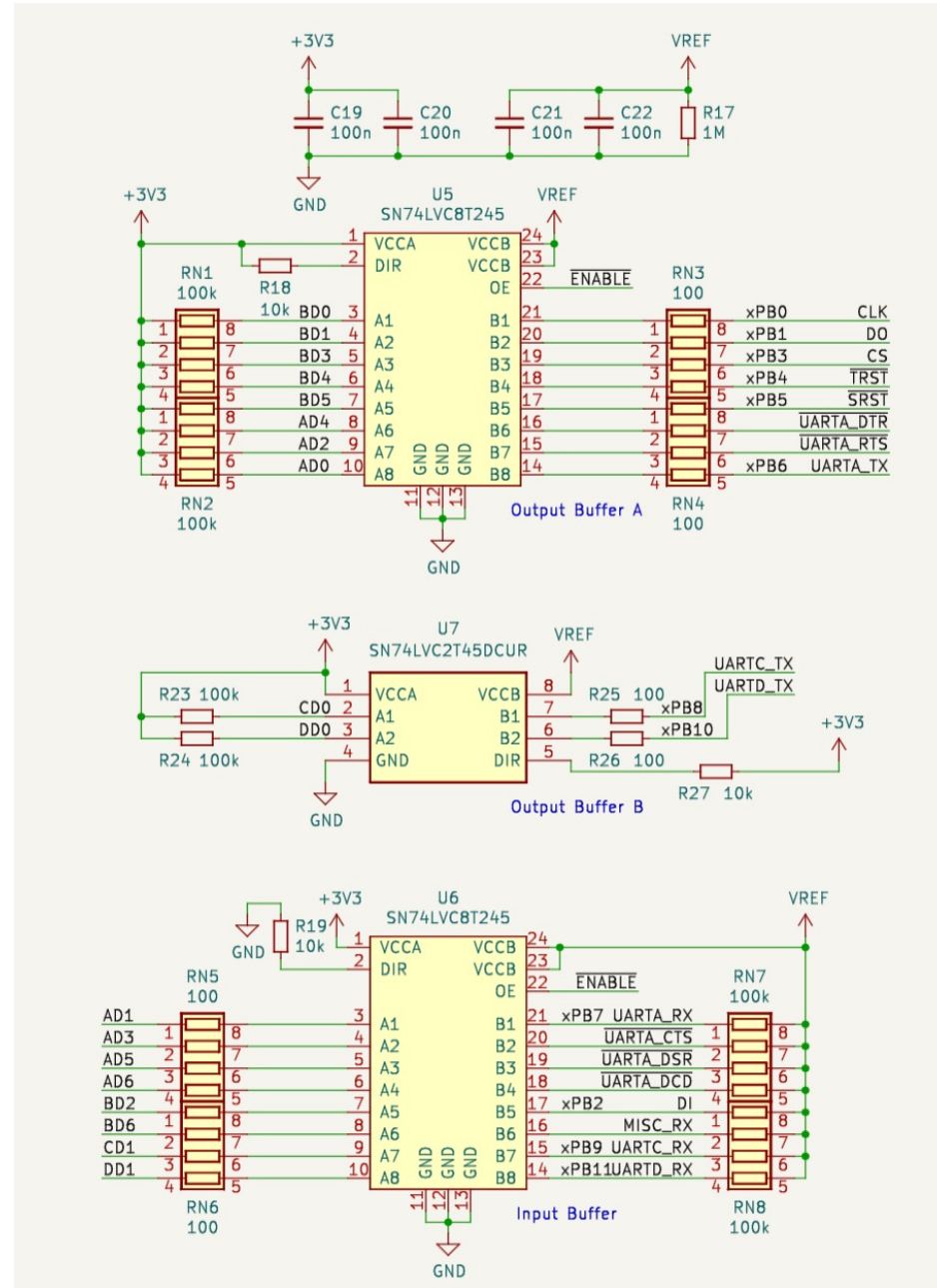


# Indicators





# Level Shifters



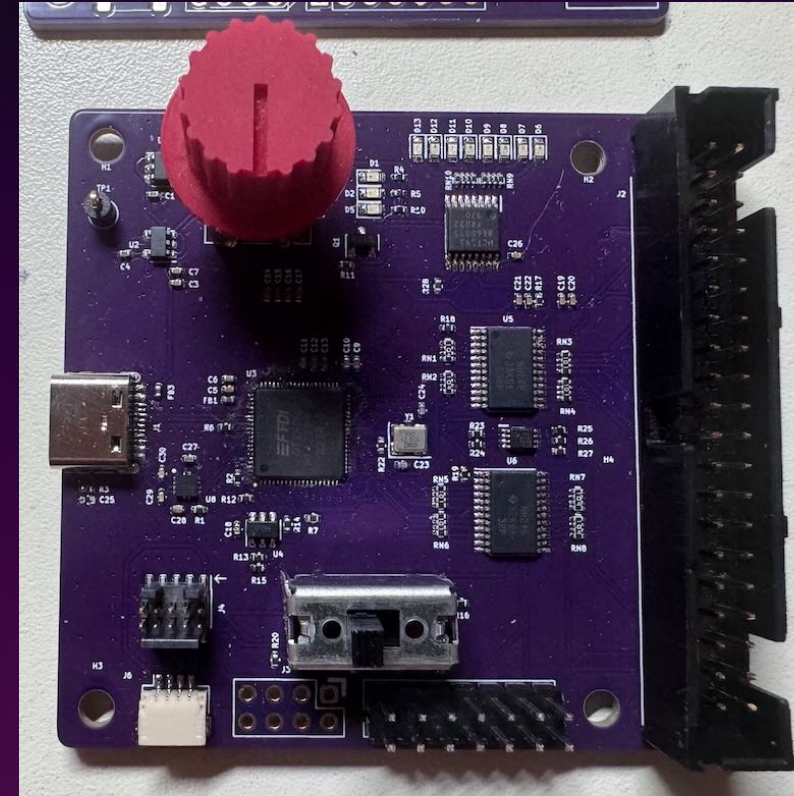
# First Attempt

Hand Assembled 2 working units

Proved serial, JTAG ports worked

Interesting to hand solder

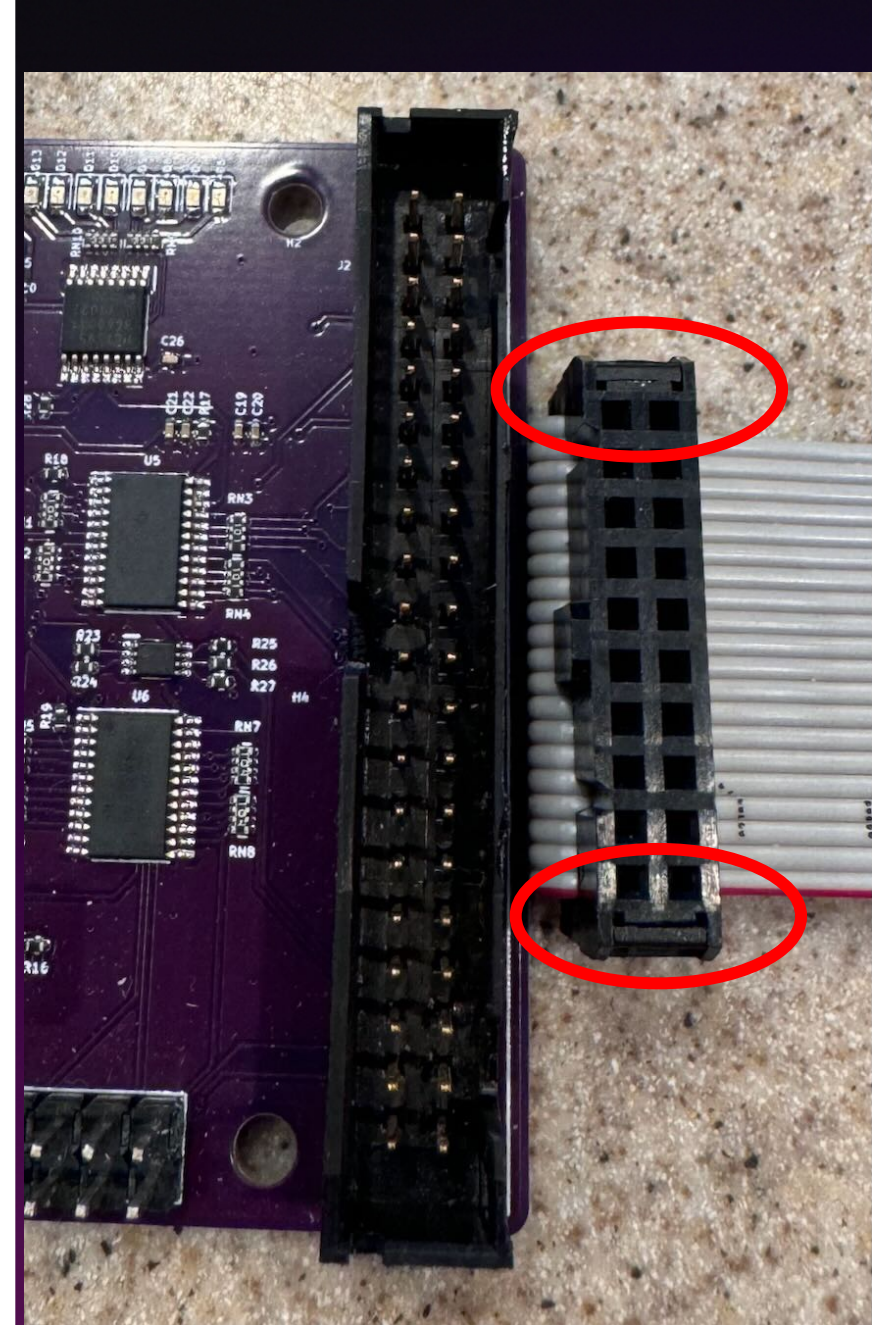
No magic smoke





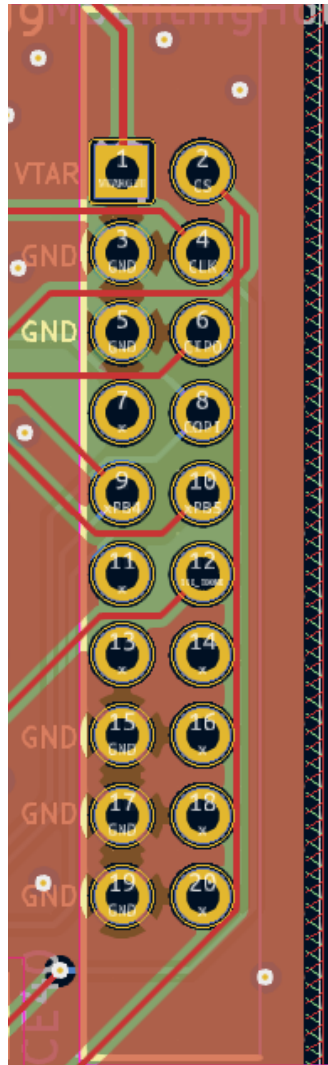
# Issue

20 pin plug incompatible



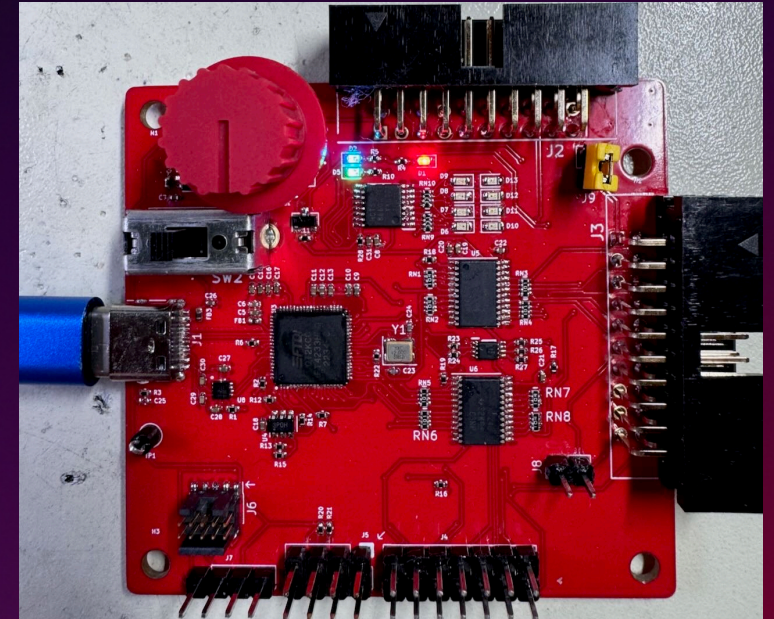
# V2

Polarity Problem



## J-Link Pinout

<b>VTref</b>	1 ● ● 2	<b>NC</b>
<b>nTRST</b>	3 ● ● 4	<b>GND</b>
<b>TDI</b>	5 ● ● 6	<b>GND</b>
<b>TMS</b>	7 ● ● 8	<b>GND</b>
<b>TCK</b>	9 ● ● 10	<b>GND</b>
<b>RTCK</b>	11 ● ● 12	<b>GND</b>
<b>TDO</b>	13 ● ● 14	<b>GND*</b>
<b>RESET</b>	15 ● ● 16	<b>GND*</b>
<b>DBGSRQ</b>	17 ● ● 18	<b>GND*</b>
<b>5V-Supply</b>	19 ● ● 20	<b>GND*</b>



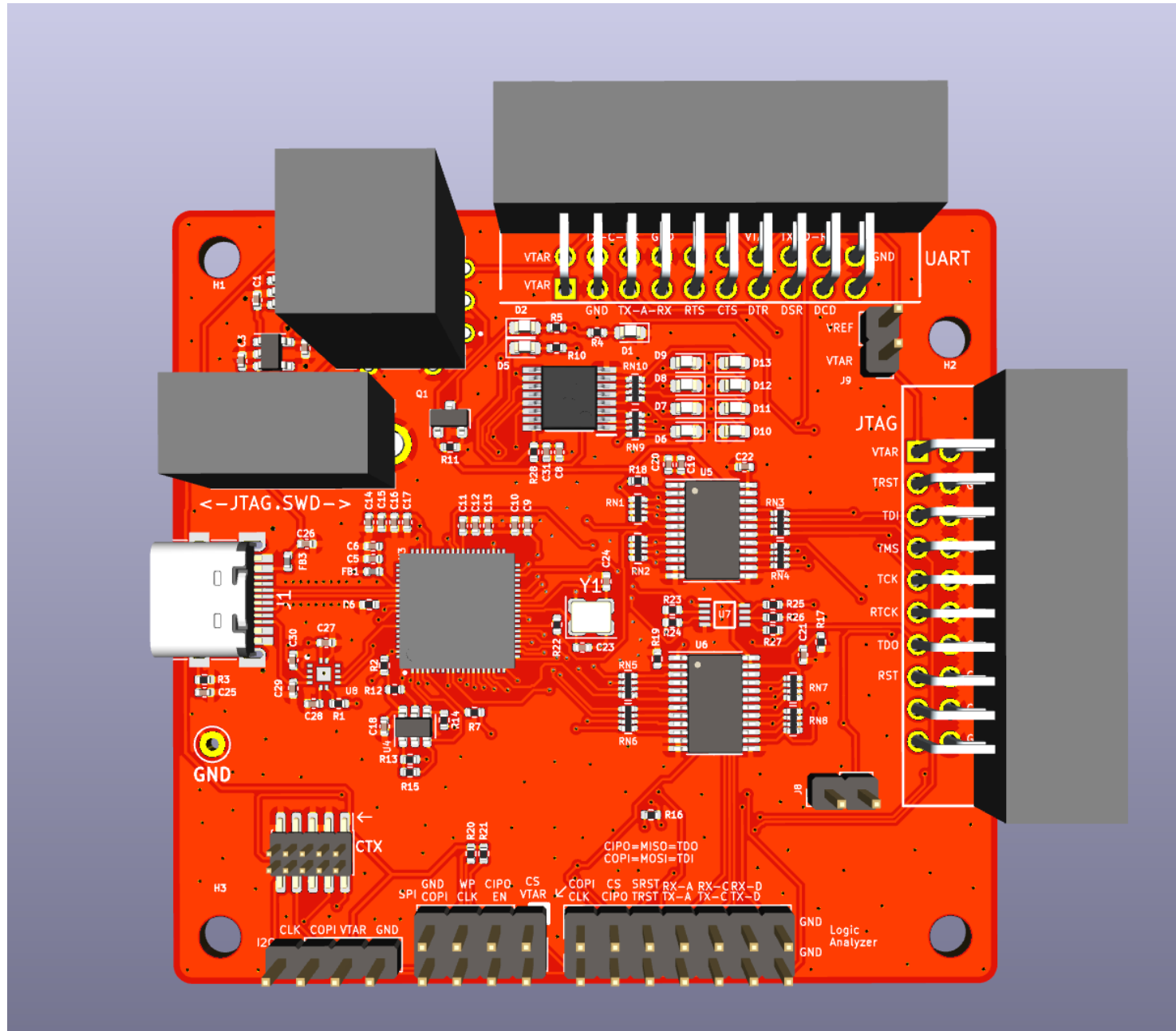






# V2 Final

I hope



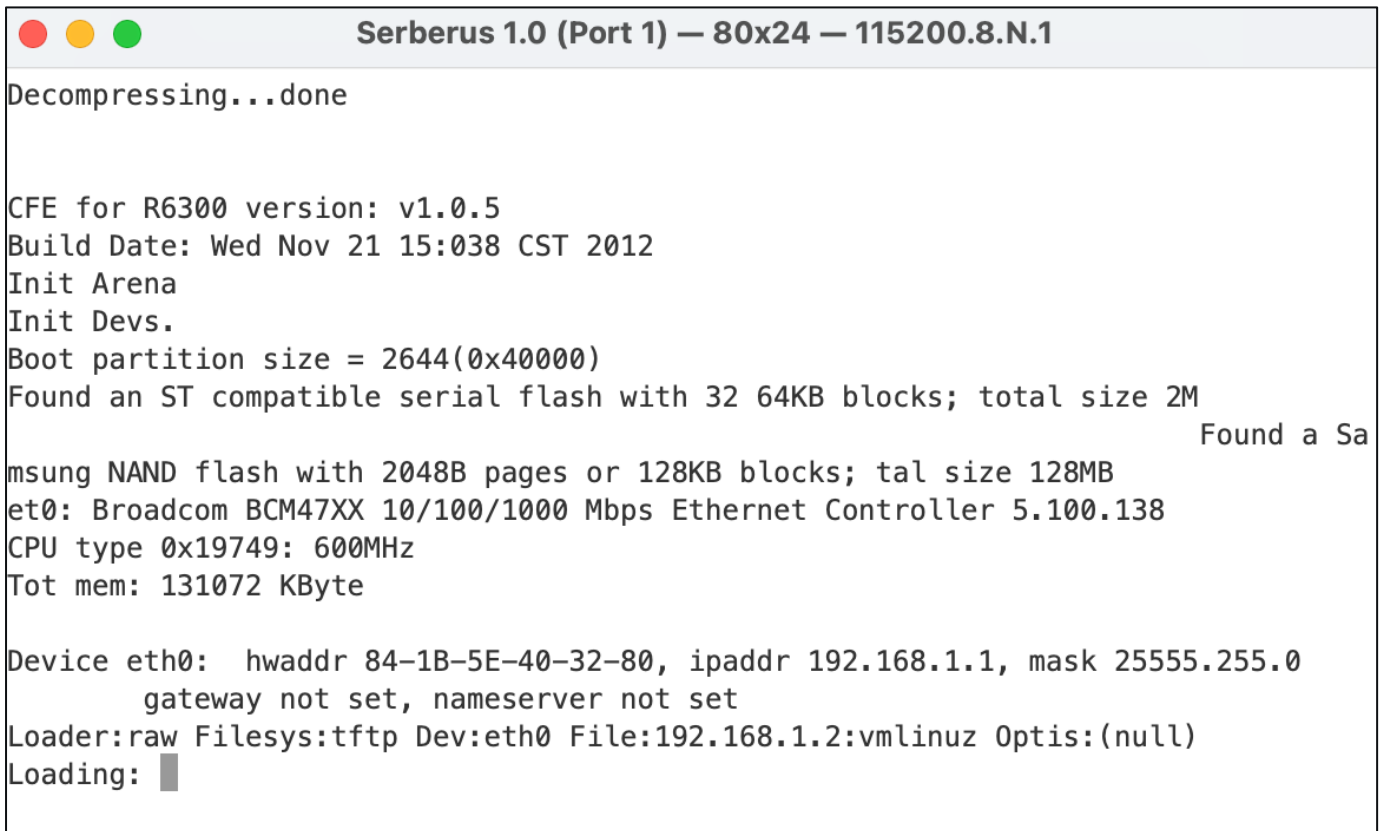
# The operators manual

Subhead can go here

# Serial Access

Simple as screen, or your favorite terminal emulation program

```
Screen /dev/TTYUSB(0,2 or 3) {baudrate}
```



```
Serberus 1.0 (Port 1) — 80x24 — 115200.8.N.1
Decompressing...done

CFE for R6300 version: v1.0.5
Build Date: Wed Nov 21 15:038 CST 2012
Init Arena
Init Devs.
Boot partition size = 2644(0x40000)
Found an ST compatible serial flash with 32 64KB blocks; total size 2M
Found a Samsung NAND flash with 2048B pages or 128KB blocks; total size 128MB
et0: Broadcom BCM47XX 10/100/1000 Mbps Ethernet Controller 5.100.138
CPU type 0x19749: 600MHz
Tot mem: 131072 KByte

Device eth0: hwaddr 84-1B-5E-40-32-80, ipaddr 192.168.1.1, mask 255.255.255.0
gateway not set, nameserver not set
Loader:raw Filesys:tftp Dev:eth0 File:192.168.1.2:vmlinuz Optis:(null)
Loading: █
```



# JTAG, SPI, Flash programming

Flashrom is a work in progress, however it should work.

Ftdi python works just fine, just need the device URI's

```
(kali@kali)-[~/pyftdi/pyftdi/bin]
└─$ sudo python ftdi_urls.py --vidpid 0403:6041
Available interfaces:
ftdi://ftdi:0x6041:SB2112/1 (Serberus 1.0)
ftdi://ftdi:0x6041:SB2112/2 (Serberus 1.0)
ftdi://ftdi:0x6041:SB2112/3 (Serberus 1.0)
ftdi://ftdi:0x6041:SB2112/4 (Serberus 1.0)
```

# Serial MitM

## Akheron Proxy

- 

<https://github.com/rapid7/akheron-proxy>

```
> list
/dev/ttyS0
/dev/ttyS1
/dev/ttyUSB0
/dev/ttyUSB1
/dev/ttyUSB2
/dev/ttyUSB3
> list -v
/dev/ttyS0
  desc: ttyS0
  hwid: PNP0501
/dev/ttyS1
  desc: ttyS1
  hwid: PNP0501
/dev/ttyUSB0
  desc: Serberus 1.0
  hwid: USB VID:PID=0403:6041 SER=SB2112 LOCATION=2-1:1.0
/dev/ttyUSB1
  desc: Serberus 1.0
  hwid: USB VID:PID=0403:6041 SER=SB2112 LOCATION=2-1:1.1
/dev/ttyUSB2
  desc: Serberus 1.0
  hwid: USB VID:PID=0403:6041 SER=SB2112 LOCATION=2-1:1.2
/dev/ttyUSB3
  desc: Serberus 1.0
  hwid: USB VID:PID=0403:6041 SER=SB2112 LOCATION=2-1:1.3
> █
```

# Serial MitM - Applications

## Akheron Proxy

With start-of-message delimiter

This is the same flow, but with a start-of-message delimiter of `0x37` set:

## Interpreting Data

```
> portset A /dev/ttyUSB1 115200
> portset B /dev/ttyUSB2 115200
> delimset start 0x37
> start
Data now PASSING between ports "/dev/ttyUSB1" <-> "/dev/ttyUSB2"...
> watch
Watching data passed between ports. Press CTRL-C to stop...
A -> B: 0x37 0x71 0x77 0x65 0x65 0x72
        0x37 0x64 0x66 0x61 0x64
        0x37 0x73
        0x37 0x68 0x68
B -> A: 0x37 0x6e 0x6d 0x62
        0x37 0x69 0x69
A -> B: 0x37 0x61 0x73 0x64 ^C
Watch mode exited.
> stop
Data now BLOCKED between ports "/dev/ttyUSB1" <-> "/dev/ttyUSB2".
>
```





# Transponder Communications

```
xpndr.txt
102c072300d0f970440000000984dad3eba05803a2fa11e42bfa2ba44f0521a4201161718191a1c465d1003
102c072300004c3a5a18c2c6fe0000000080248fa006b4c3e00c8ebbdcd47edff1d1eaf30b1b2b31e391003
10ce0001000112511003
10ce01010001a94d1003
10ce0201000164681003
102c0723000018385a321fc4fe0000000b0b847fa80c6543e8040ebbd6676f5ff1d1eaf30b1b2b366f61003
102c072300009b3c5a0681ecfe00000000482f46fa4051513e804bdcbddd4eaff1d1eaf30b1b2b3ce7b1003
102c07230080c03b5a14b4ebfe00000000609d45fa804f433e0040f0bd4924ecff1d1eaf30b1b2b3e01a1003
102c0723008090315a2b6501ff00000000e8ca46fa004b443e001010debdada947001d1eaf30b1b2b342cb1003
102c0723001c1771440000000067c51c3ed2d1ef398aeac441c555b044601d10104281961798191a1c4fdd1003
102c07230080ba315aae3901ff00000000281546fac0594d3e0038dbbda6bb20001d1eaf30b1b2b36f961003
102c0723008045285ad72b11ff0000000060a845fa80af503e002ae1bde3d70f001d1eaf30b1b2b3947c1003
102c072300003a265a685d11ff00000000c8b646fa00a0563e8030f2bdc1ace4ff1d1eaf30b1b2b332341003
102c07230000c71b5a0da71aff00000000b0c249fac0ba433e80ebdbbdf8d6ff1d1eaf30b1b2b372251003
102c0723000090195ae4351cff0000000090a549fa80274c3e00aed5bd93add4ff1d1eaf30b1b2b3c4a31003
102c072300ce1d7144000000008262073f12e1c73a2dd96841b847ad44101004004281961798191a1ca3881003
102c07230080601c5ae0f91eff00000000e88048fa00cb4b3e8081e0bda5aa08001d1eaf30b1b2b302441003
102c07230000301c5a920f1eff00000000204a49fa0098413e0082e9bd13881010001d1eaf30b1b2b3d43a1003
102c072300808d165ae4671fff0000000080fc47fa4011493e8030dcbdf21509001d1eaf30b1b2b3ae941003
102c072300801f165ad4791cff0000000030f246fa800c4d3e0026f4bd3ed31d001d1eaf30b1b2b34e761003
102c07230080200f5aa3bd20ff00000000503c47fa400a3e3e801ae7bdeba4e8ff1d1eaf30b1b2b3978d1003
102c07230000400d5a3b631fff00000000608a47fac0f6573e806abdbdaadf02001d1eaf30b1b2b33e541003
102c072300e619714400000000ac1b213ecf0fee39ba90e34077e2ba44004ae34181961798191a1c5bb01003
102c07230000ea115ac22922ff00000000c08046fa803b4f3e00f4cabd03b0e4ff1d1eaf30b1b2b3f2471003
102c072300808010105a1ffe20ff00000000180147fa004c453e8043f0bd24eff4ff1d1eaf30b1b2b3cd0d1003
102c07230000530f5ab7911eff0000000030b646fa4074413e008ed3bd2fa9e1ff1d1eaf30b1b2b3cb9c1003
102c072300001f0f5a79651cff0000000009b45fa804b4a3e0051e0bdba65edff1d1eaf30b1b2b3150a1003
102c07230080fc10105a5ff320ff0000000030b346fac010104e3e0040e4bd17073d001d1eaf30b1b2b31310101003
102c072300c3197144000000005c92253c189cf4377a6e61402c91c144d008d44181961798191a1c093b1003
102c07230000b40f5ac32c1fff00000000287947fa40a5573e0006d5bd2a29f2ff1d1eaf30b1b2b3d9491003
```

Thank you!

Questions?