Looping Surveillance Cameras

(like in the movies)

Who are we?

- Ordinary law-abiding citizens
- Nothing to see here, move along

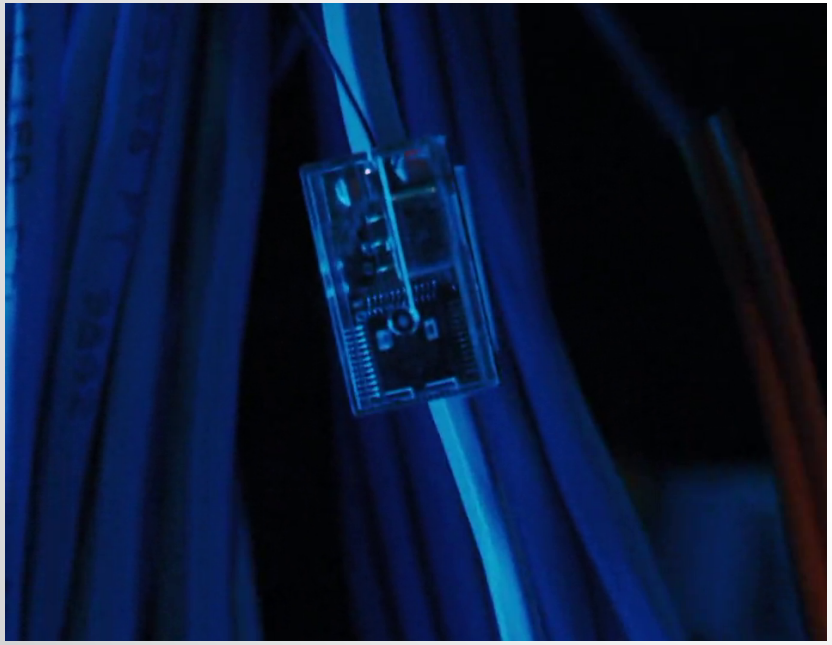
- Eric Van Albert <eric@van.al>
- Zach Banks <not-eric@van.al>

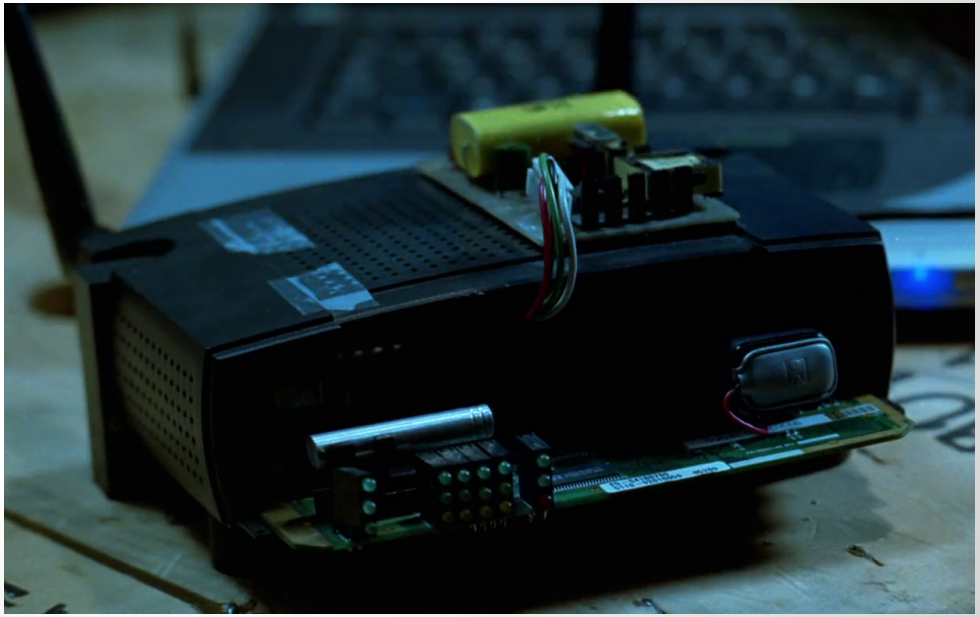
Prior Art

(what this isn't)

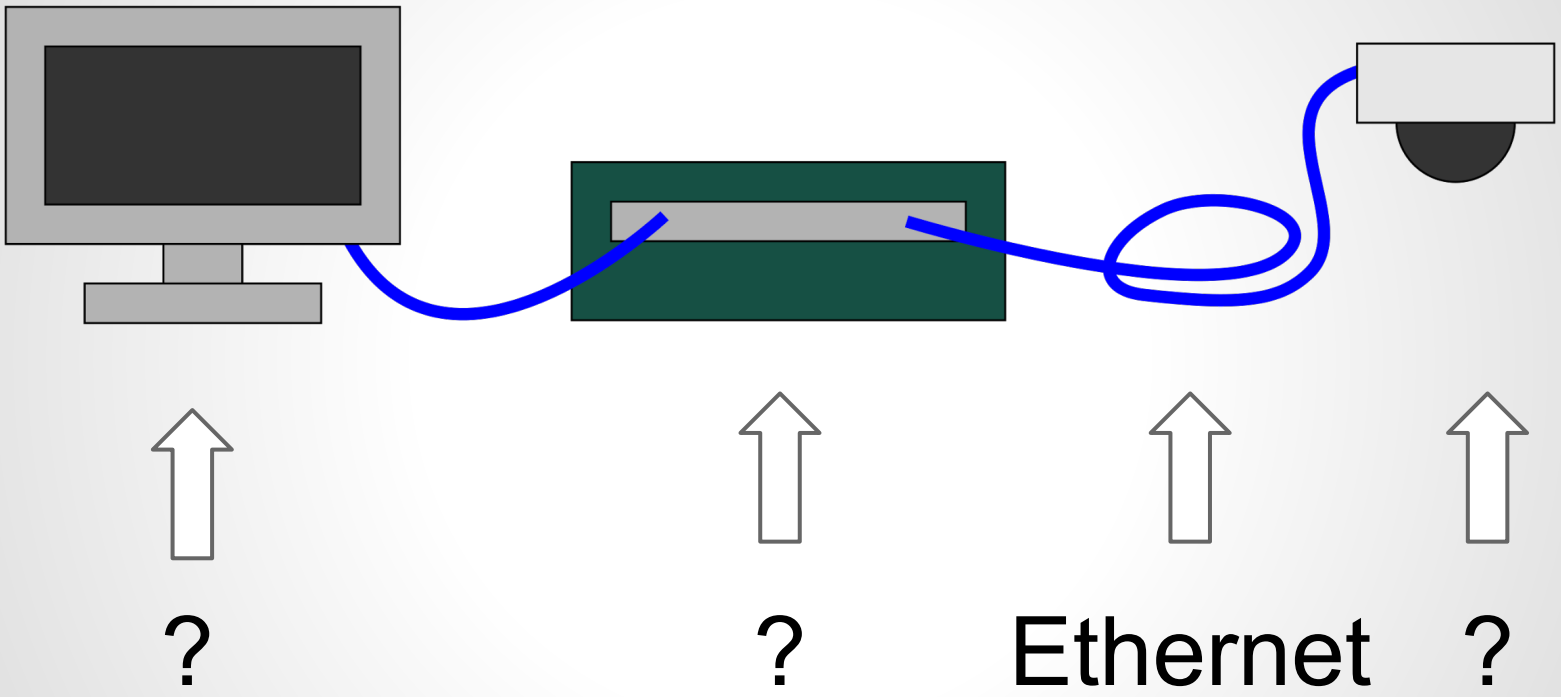
Prior Art

(what this is)

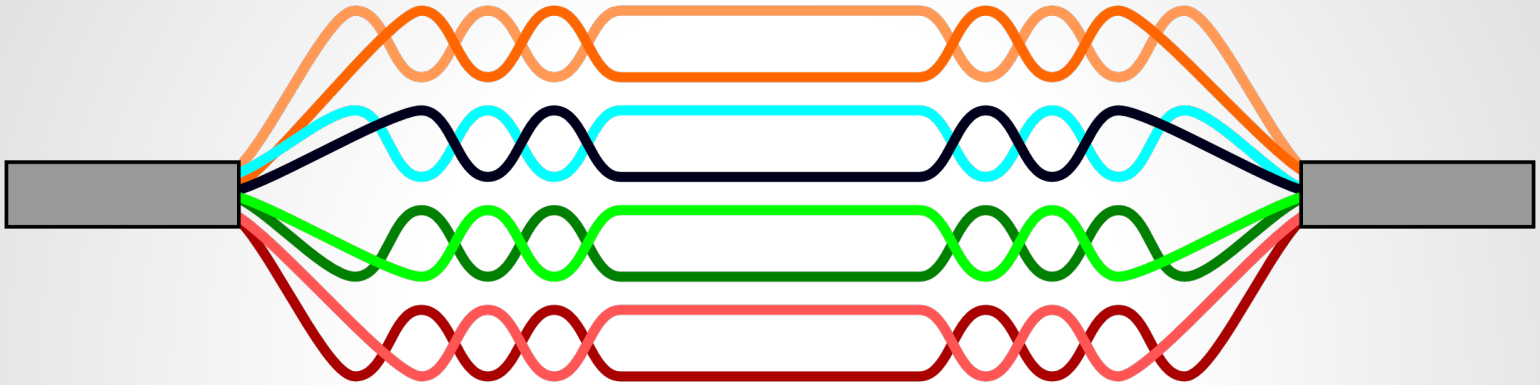




System

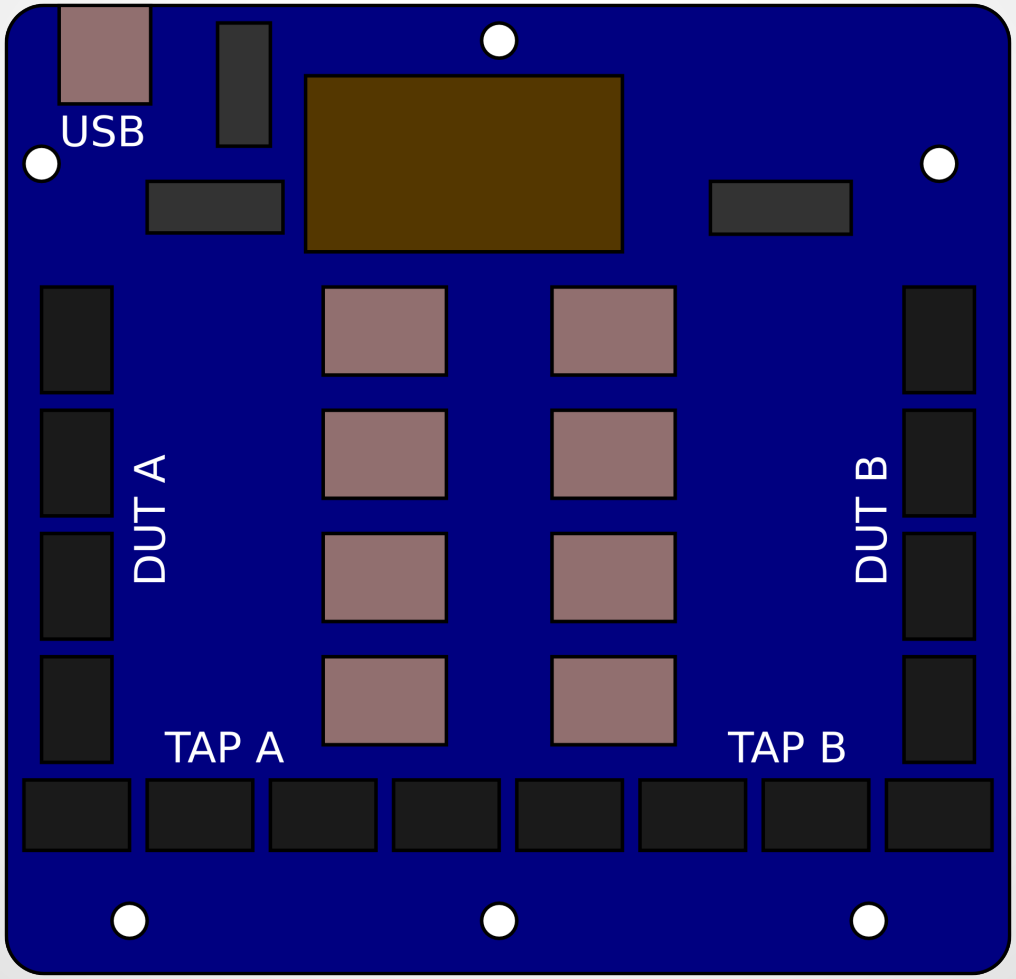


Ethernet Anatomy



- Four twisted pairs
- All may be carrying data
- Wide variety of electrical standards

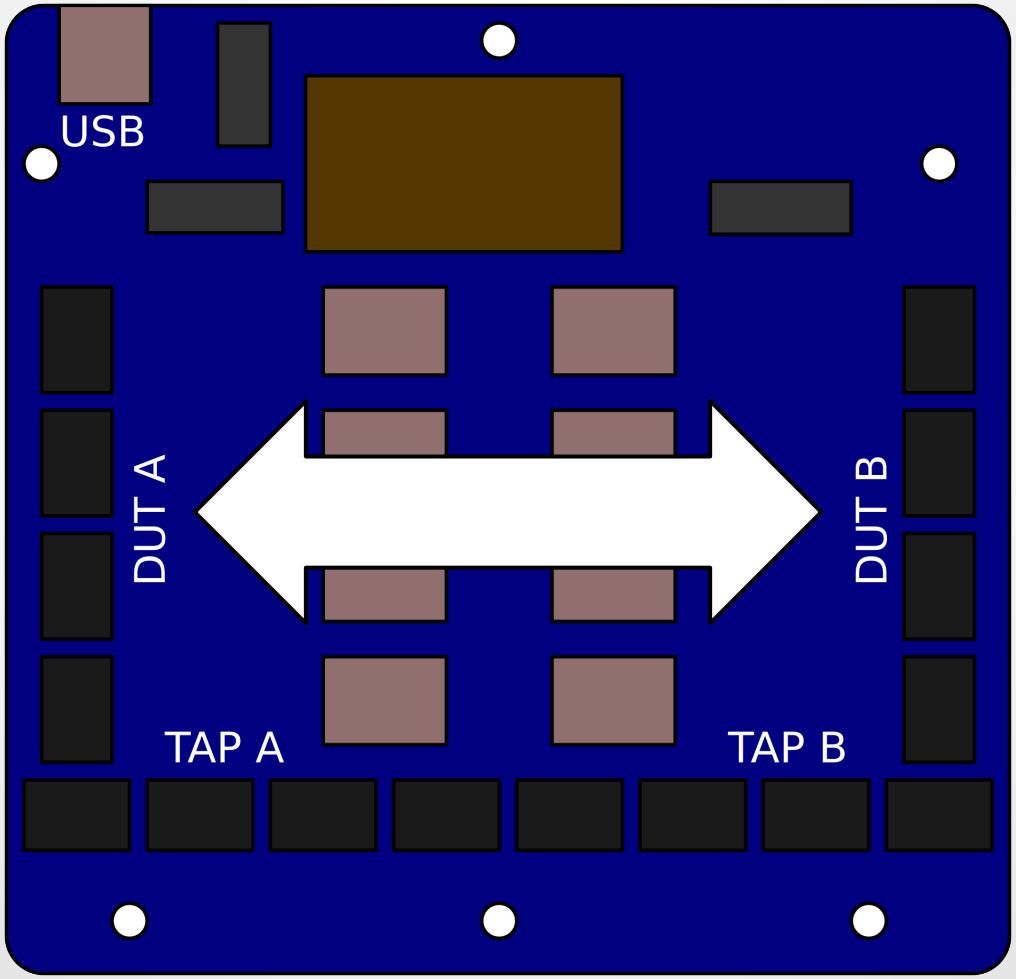
The Tap Board



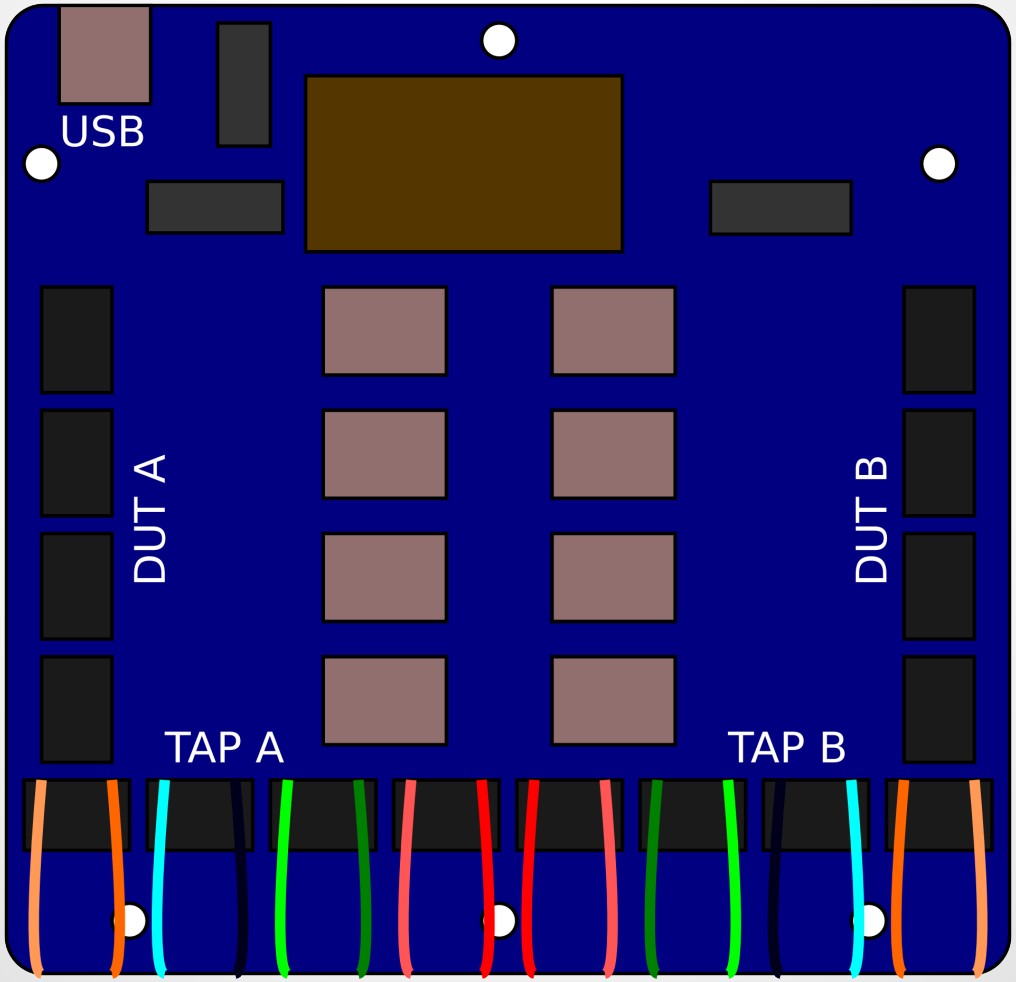
The Tap Board

- Eight DPDT latching relays
  - Rated for 1 GHz
- Punch-down connectors
- Impedance-matched traces
- Powered and controlled over USB

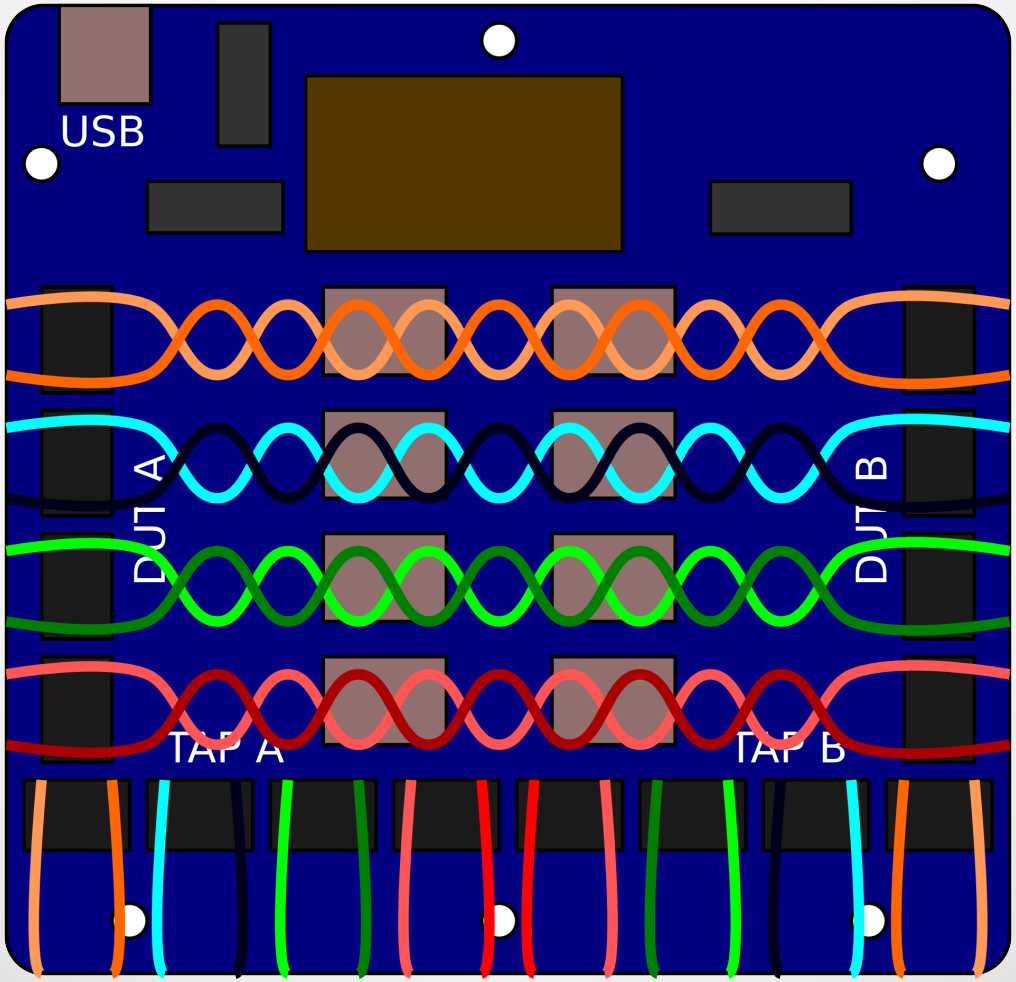
The Tap Board



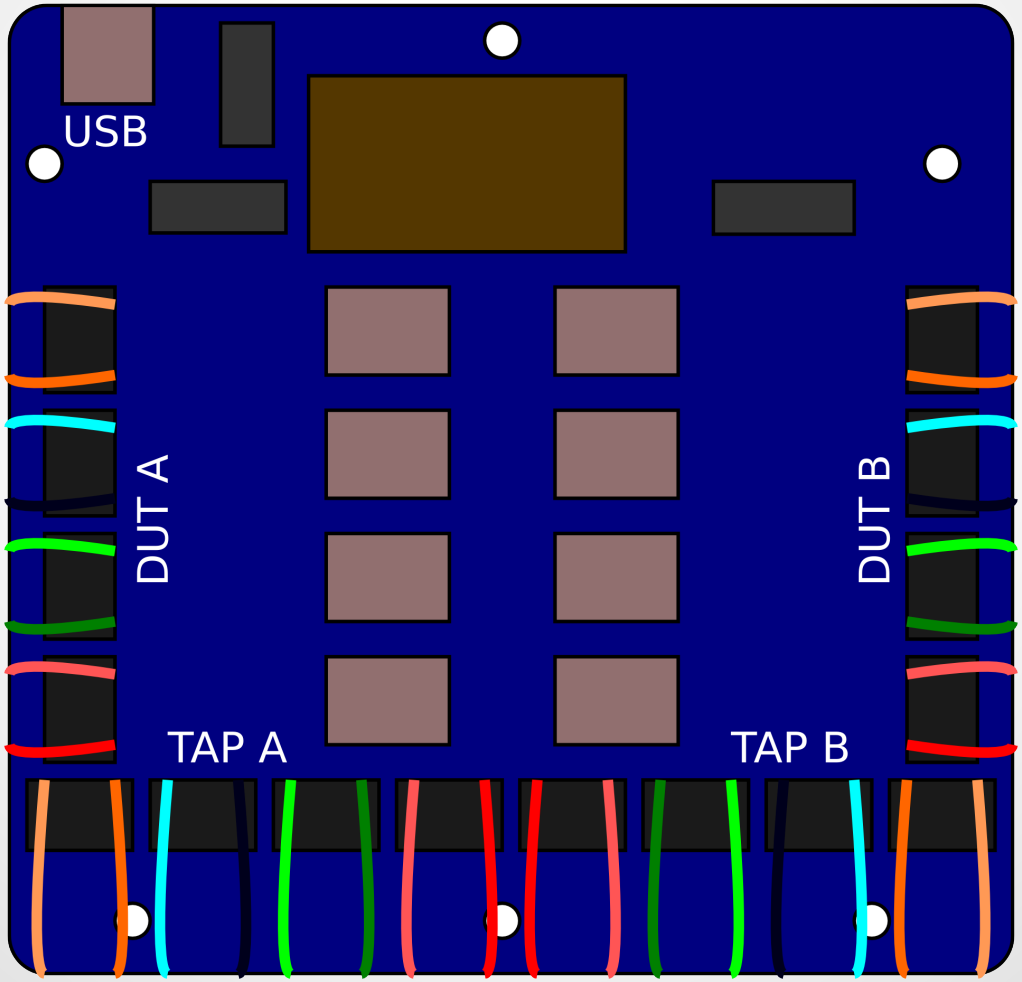
Splicing Ethernet



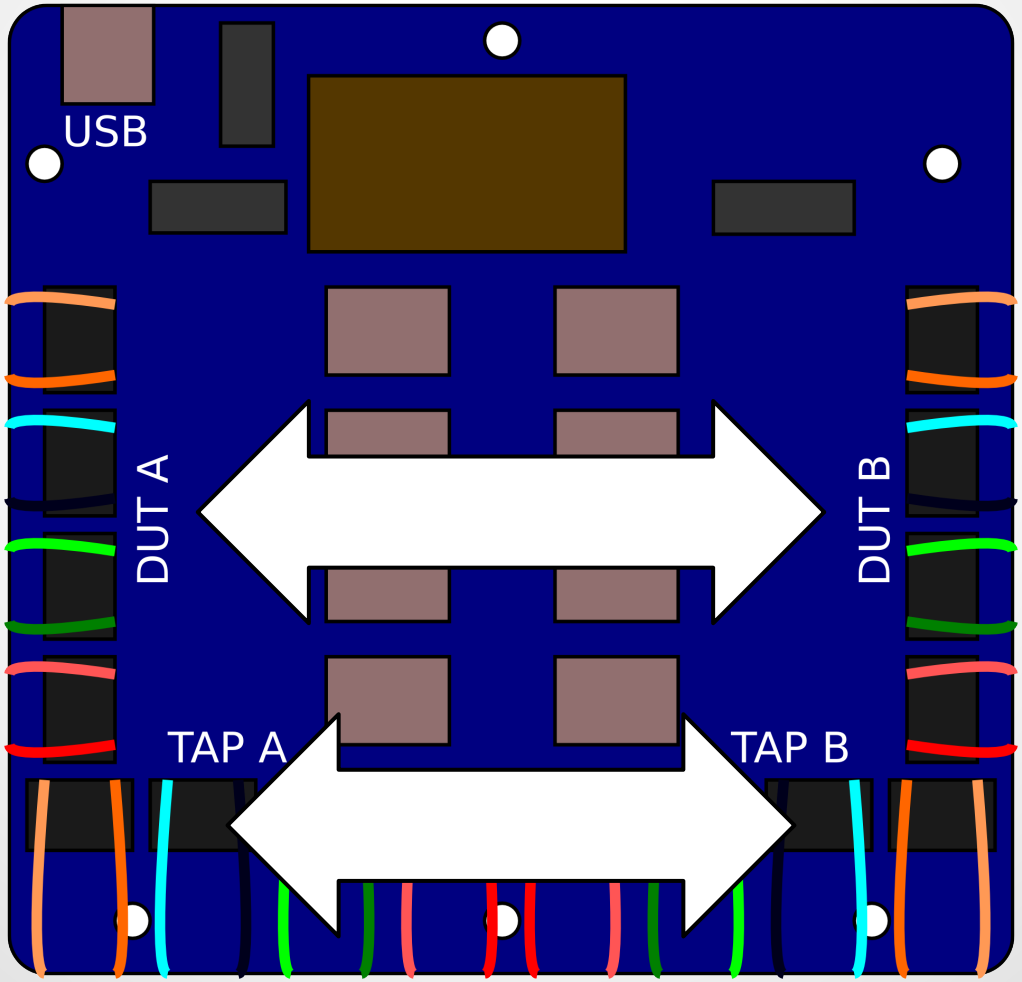
Splicing Ethernet



Splicing Ethernet



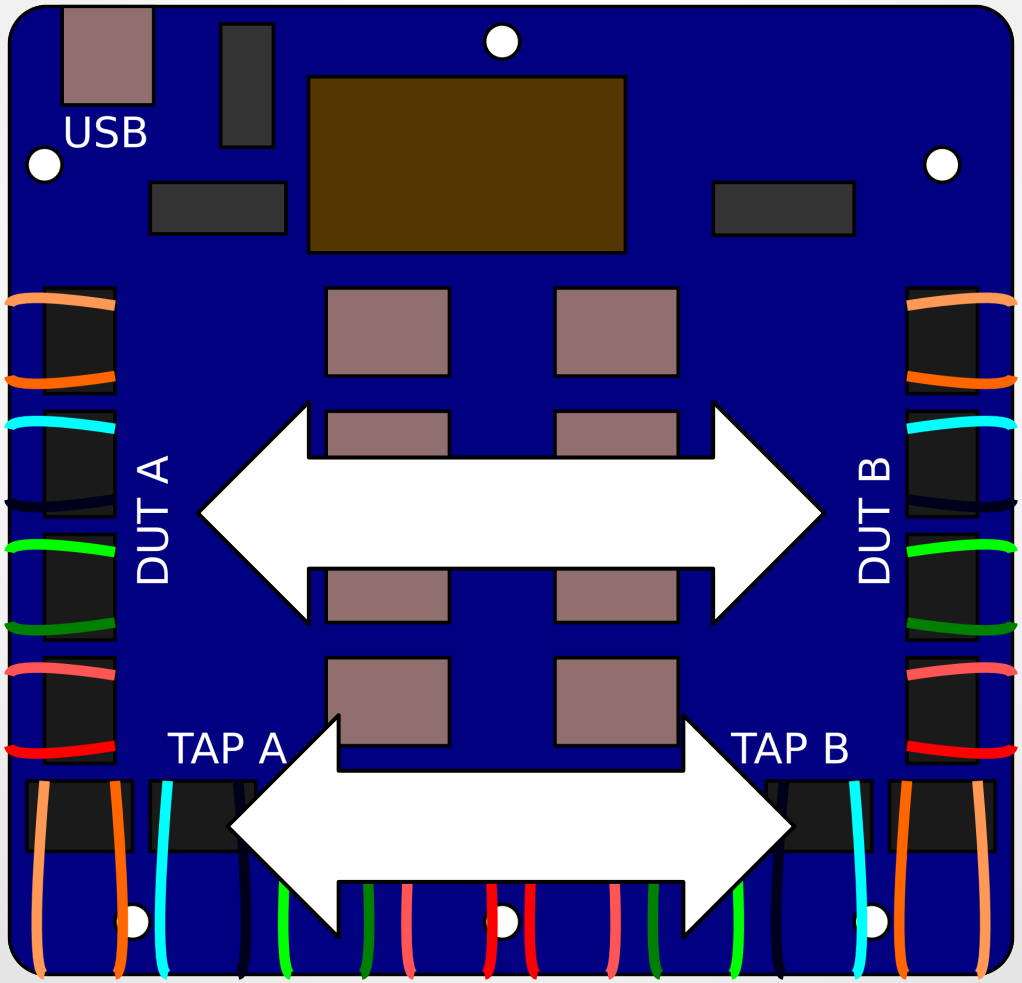
Splicing Ethernet



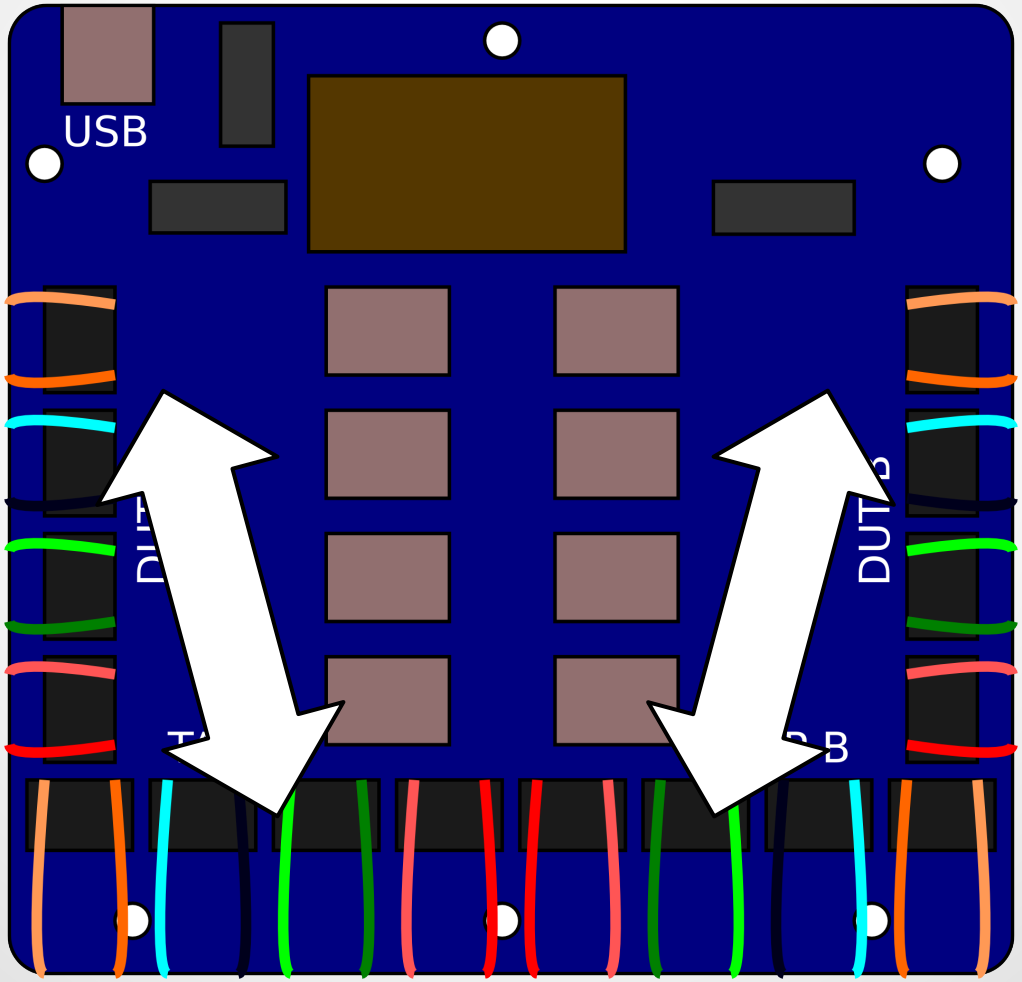
Advanced Tap Board Features

- Tamper-evident
- Fail-safe heartbeat
- Fail-safe power loss

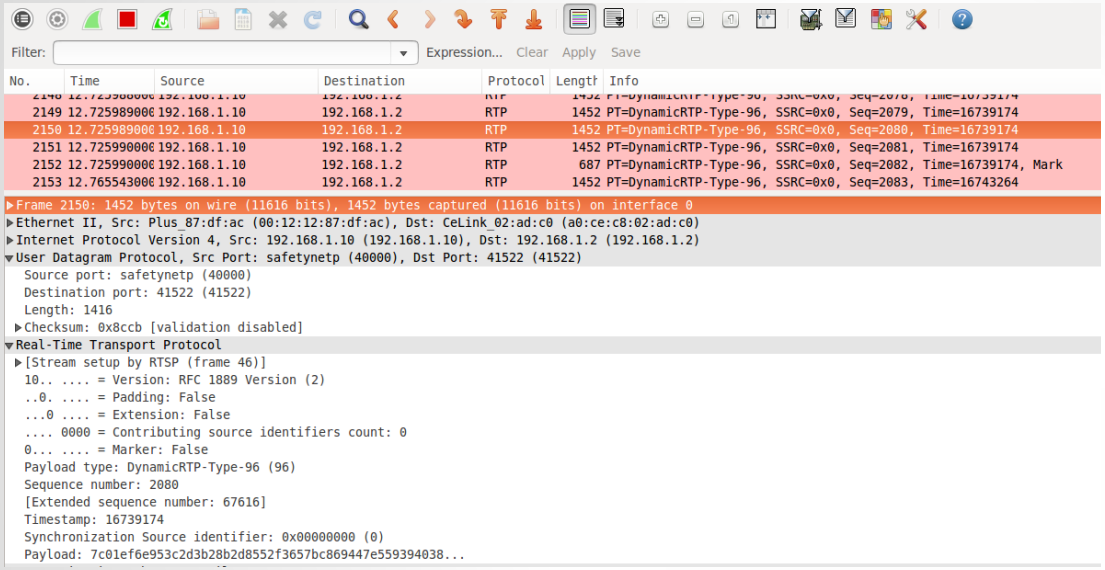
Switching to Active Tap

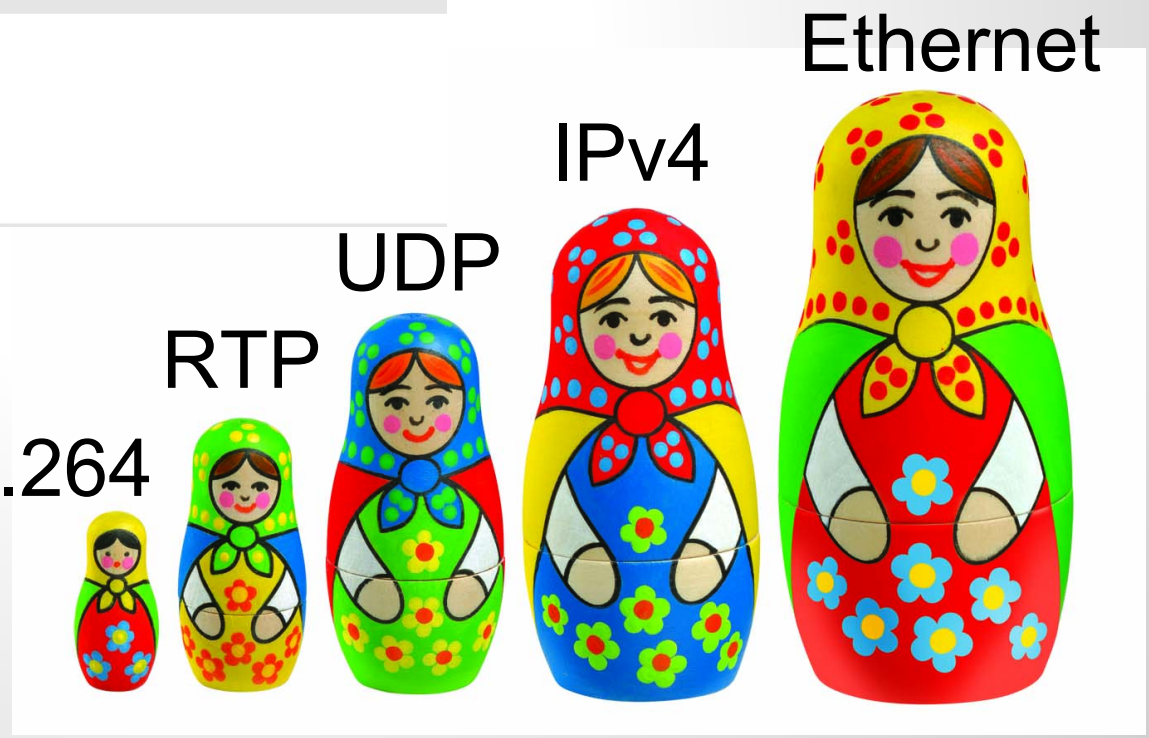


Switching to Active Tap



The Network Stack

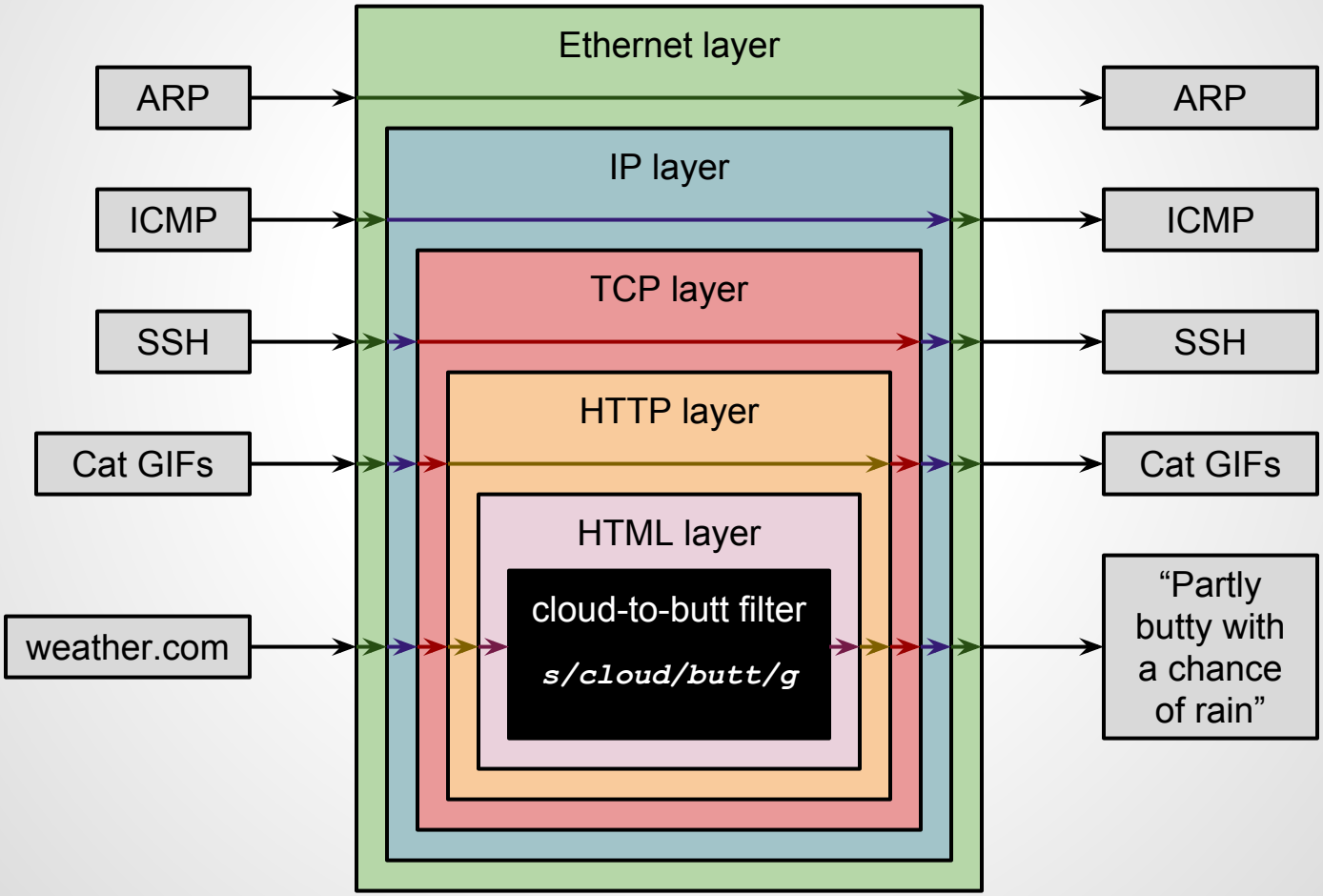




lens Overview

- MitM-centric network stack in Python
  - Ethernet, IPv4, TCP, UDP, HTTP, RT*P...
- Designed to be as transparent as possible
  - We need to be able to forge packets as necessary and have them blend in
- Allows for additional “layers” to filter data
  - Ex: turn a video stream into a loop

lens Implementation



Looping Video

- RTP: Real Time Protocol
  - RTSP: Streaming session; TCP
  - RTCP: Codec information, statistics; UDP*
  - RTP: Video data; UDP*
- `ffmpeg` solves all of your (video) problems
  - Looping, masks, transforms, and more!

Looping Video

1. Read video stream from camera over RTP
2. Create new stream using `ffmpeg`
3. Forge packets from camera of new stream
4. ???
5. Profit!

Potential Extensions

- HTTPS
  - Incredibly tricky to get right in embedded systems
- USB
- HDMI

Check it out!

- Hardware
- Firmware
- Mechanical
- Code

http://github.com/ervanalb/lens