Tim Panton tim@pi.pe @steely-glint@chaos.social

WebRTC On Moving Devices

Part 1 - Bandwidth and Networks
kamailioworld 2023

Tim Panton tim@pi.pe @steely-glint@chaos.social

Rendezvous.team

Race car to pit realtime video

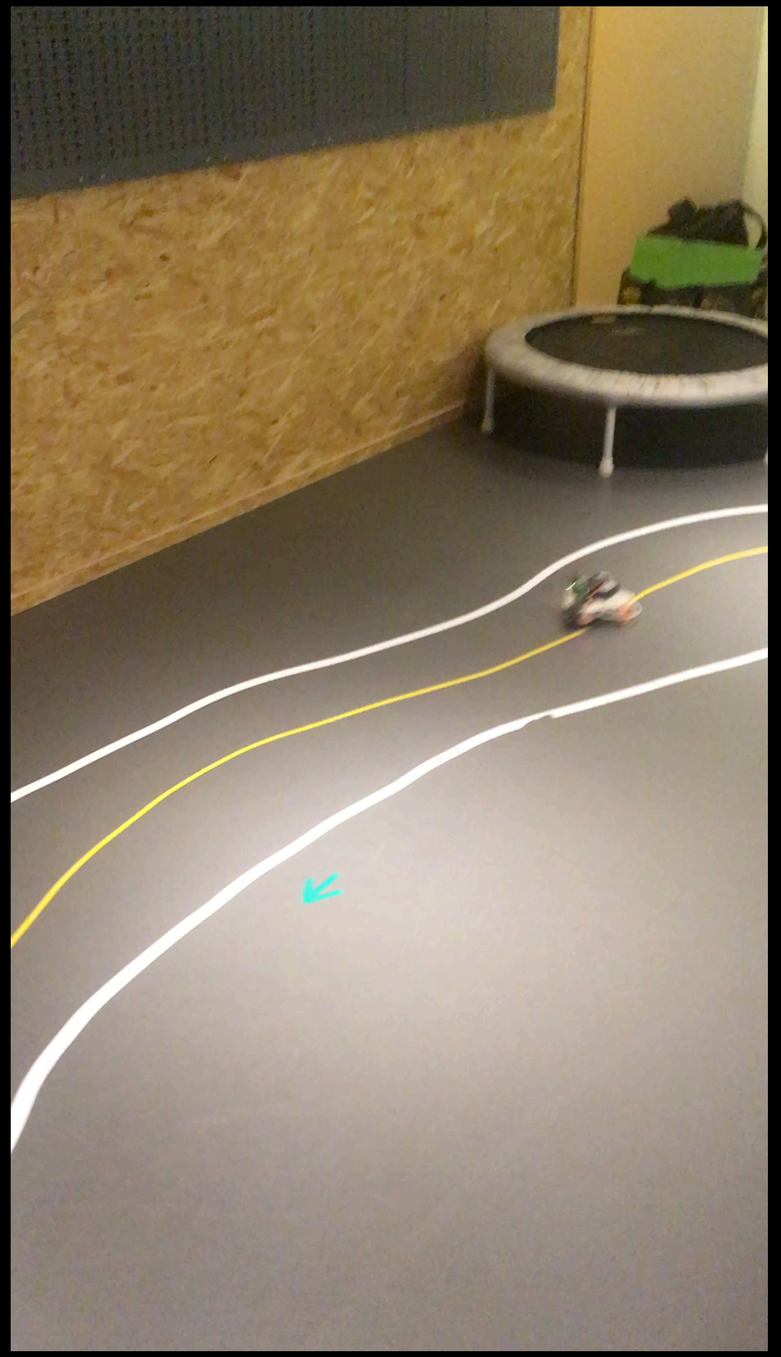
Screen recording from a prototype

Clank
RoboCars race

Teach an ai by example
By driving ‘through the lens’

(not quite how it turned out)

Drive



Engineering goals
(Common factors)

- Low latency Video
- High quality Video
- Low weight
- Low cost
- Ease of use (should not get in the way)
- Remote control

swap sds

Differences from ‘normal’ WebRTC

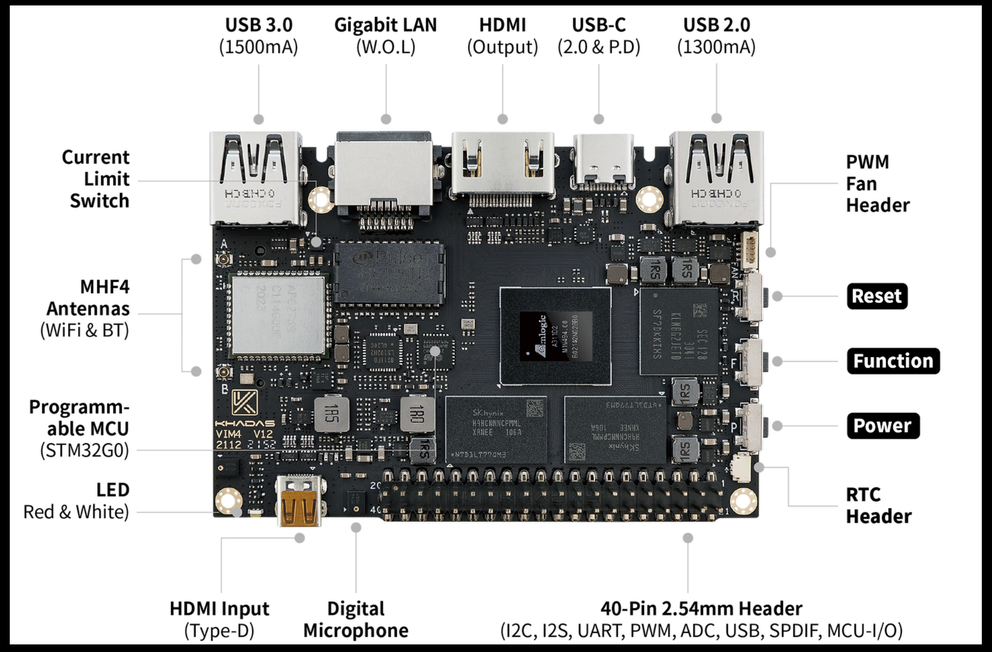
Not another video conference!

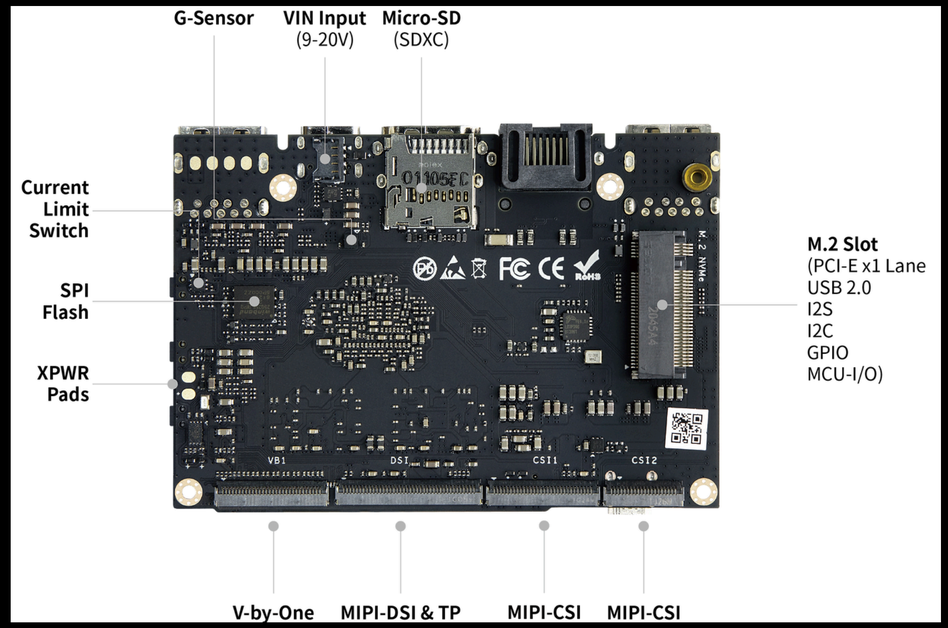
- P2P
- One way Video
- Low Network Latency
- Variable network

- Hardware encoders
- Small devices
- Not libwebrtc
- Data side channels

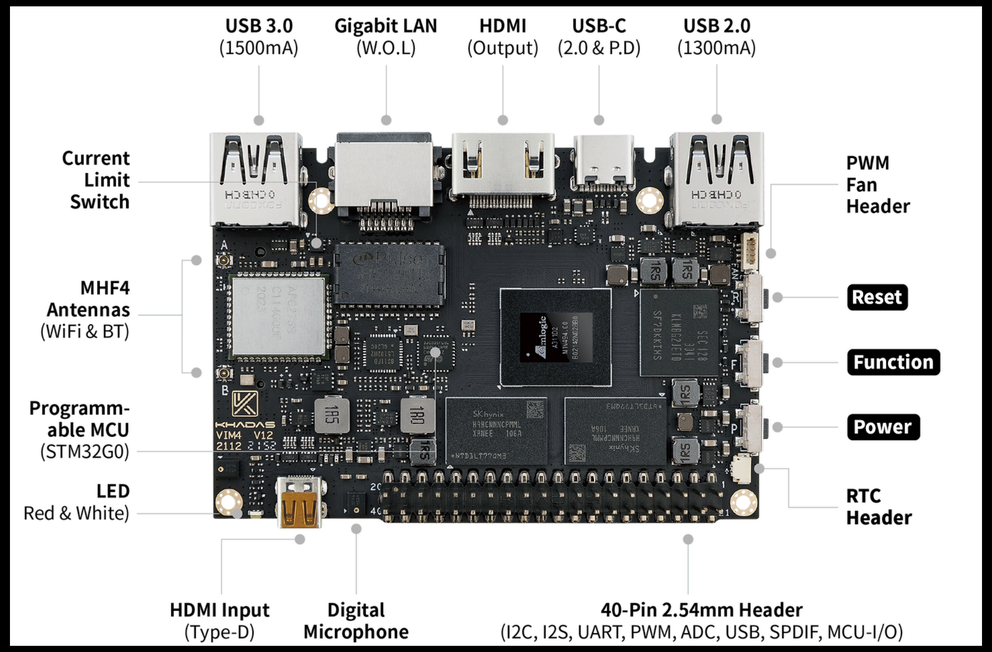
Example device -VIM4 ~200 euro

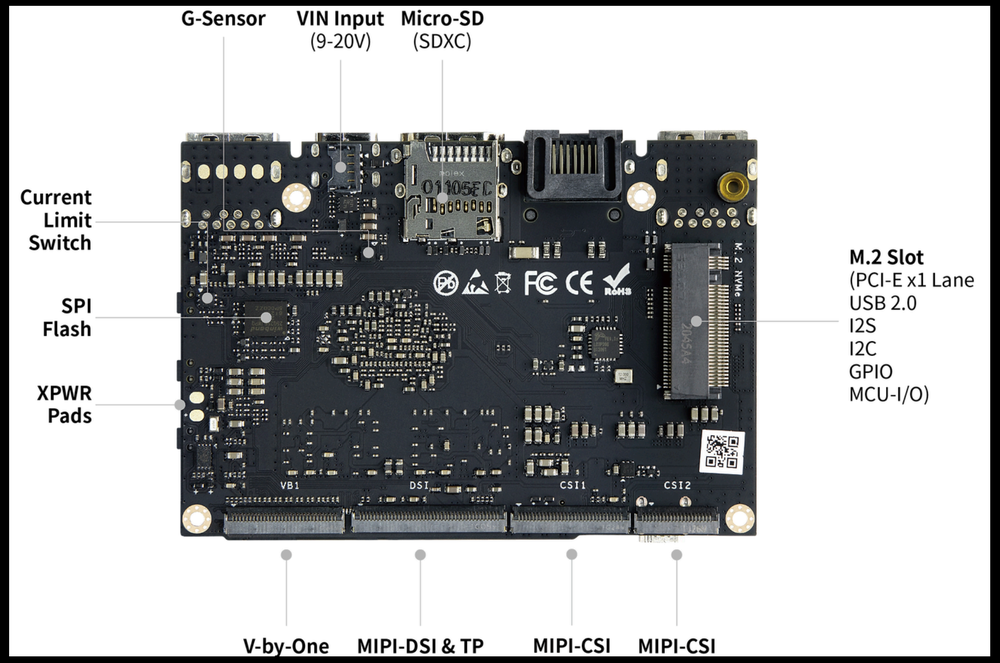
- H.264 and H.265 encoding at 4K 50fps
- 8 ARM cores
- 8Gb RAM
- 32Gb flash
- Wifi
- M2 slots for 4/5g and drive
- Ubuntu
- 185g in case+camera





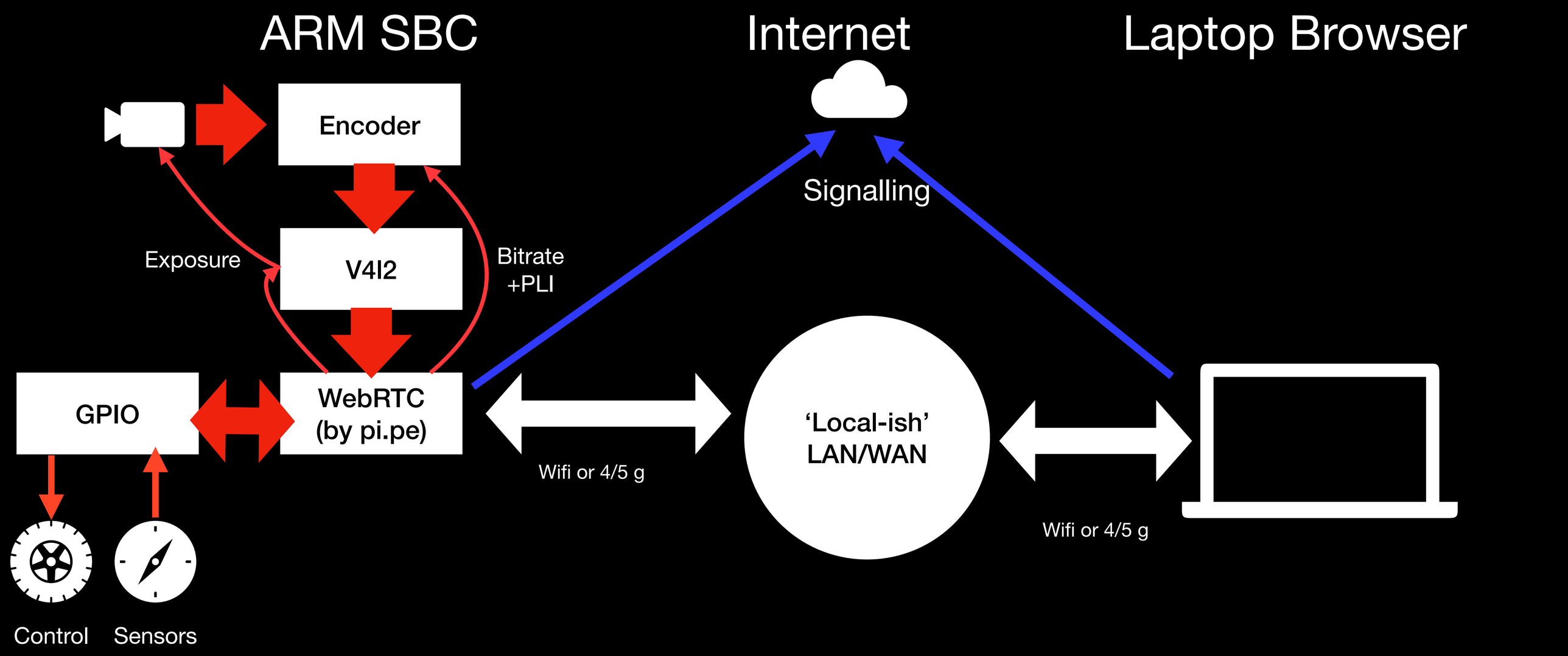
Example device -VIM4 ~200 euro





- H.264 and H.265 encoding at 4K 50fps
- 8 ARM cores
- 8Gb RAM
- 32Gb flash

- Wifi
- M2 slot for 4/5g
- Ubuntu
- 185g in case+camera


ARM SBC
Internet
Laptop Browser
Encoder
V4l2
Exposure
Bitrate
+PLI
Signalling
GPIO
WebRTC
(by pi.pe)
Wifi or 4/5 g
‘Local-ish’
LAN/WAN
Wifi or 4/5 g
Control
Sensors

WebRTC - Network interaction

How to tweak WebRTC

- BWE - estimate bandwidth -> Manage encoder params to keep within available bitrate
- NACK ->Request resend of packet(s)
- PLI -> Request sending of new Full Frame (HUGE)

Preliminary results

BWE

- Fast changing network conditions confuse BWE
  - Constrain changes to ‘acceptable’ range
  - Bitrate changes are costly on some encoders
  - Prioritise fast recovery over occasional lost frames
  - Inconsistent quality is disconcerting
  - ‘Speeding up’ to catch up is confusing to the user

NAK

- Low Latency Network makes NAK very valuable
  - If rtt < frame interval
  - At 30fps that’s 33ms
  - Wifi (and some 5g)
  - Keep a small cache of old packets to resend quickly
    - Roughly 2 frames worth of data + most recent keyframe

PLI

- Network drops (shadow or handover) cause PLI
  - De-clog the pipe
    - Clear packet cache on full frame
    - Clear the send queue
  - Only send 1 full frame at a time
    - Ignore repeated PLI whilst full frame in flight

Codec stuff

Hardware codecs are odd…

- Bit rates are quantised with sweet spots
- Some re-configs send a full frame
- Only send full frames on request
  - All the bitrate goes to movement (spacial opt)
- V4l2 creates bursty data flows - need pacer
- Always lie in the a=fmtp about the profile - claim 42e01f
  - Some decoders will ignore stuff they _can_ decode.

Network

- Look at the path (if you can)
- 4g roaming SIMs (especially) can give odd results
- Client isolation on wifi networks
- Often better results if the 2 ends are _not_ on the same network
  - E.g. 4g->4g worse than 4g-> wifi

On Device Recordings

- Use a gstreamer tee or |pipe| relay to write to file on an SD card
  - Simple but…
    - Limited capacity on SD
    - Limited bitrate - SDcards stall
    - Few keyframes
- Some GPUs support multiple encoders on the same input

On Laptop Recording

- Use media recorder API
- Suffers from same frame drops as the live version
- Plenty of space
- True record of what was _seen_ in the pit
- Preserved in crashes (Clank hits walls _often_)

Data Channel

Normal use

- Control
  - Exposure
  - Motors
- Monitoring
  - Gsensor
  - Temp
  - Range

Management

- Remote access
- Remote updates

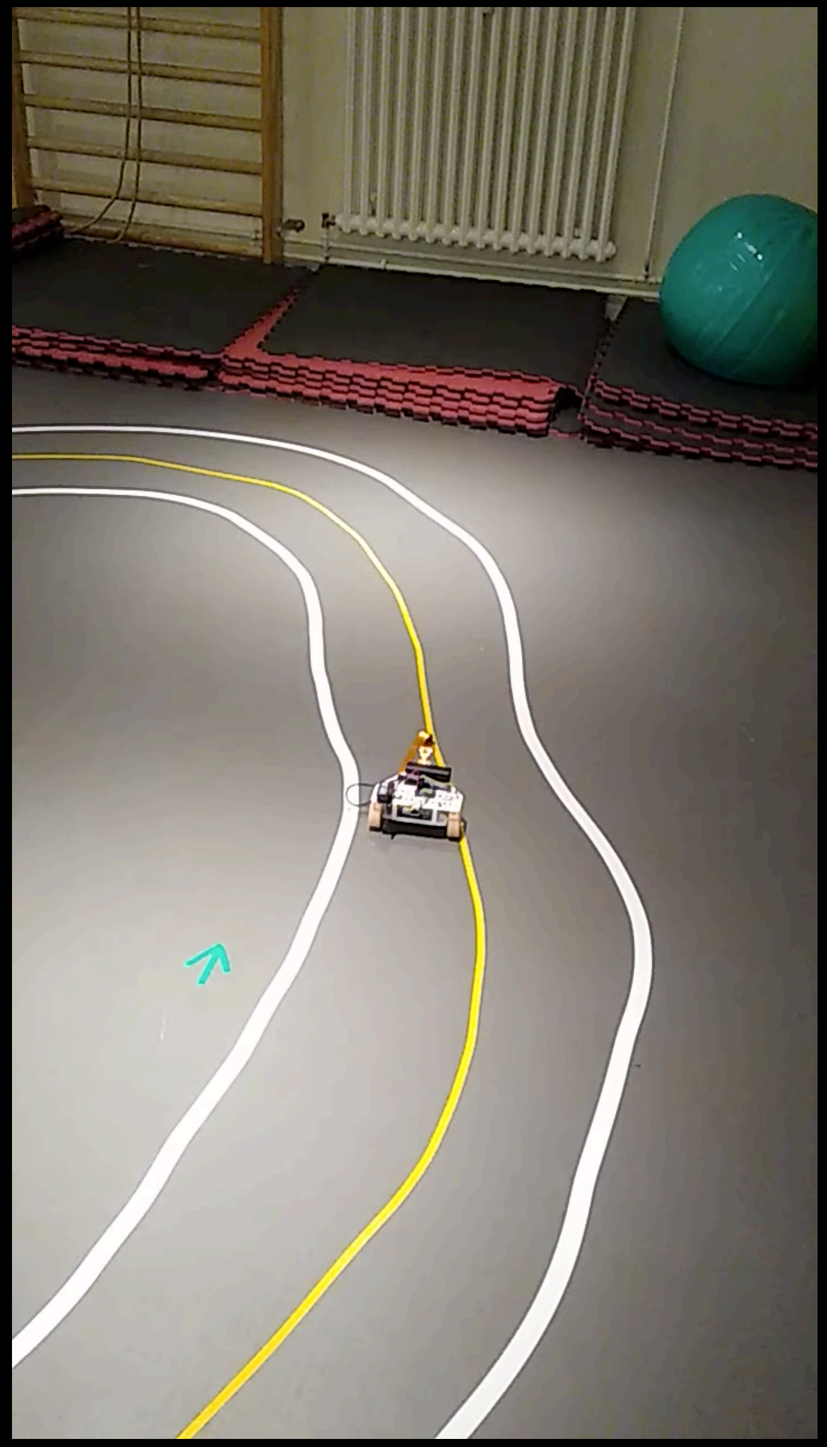
(nat traversal and security for free)

2022/23 Robocars

Training that AI didn’t go well

- Yellow line
- Ditch Tensor flow
- Explicable code
- Can lose a lot of weight 350g

- Pi zero not Coral
- Lighter battery
- Save on cloud bills



Line follower

In java with help from ChatGPT

https://github.com/steely-glint/clank2

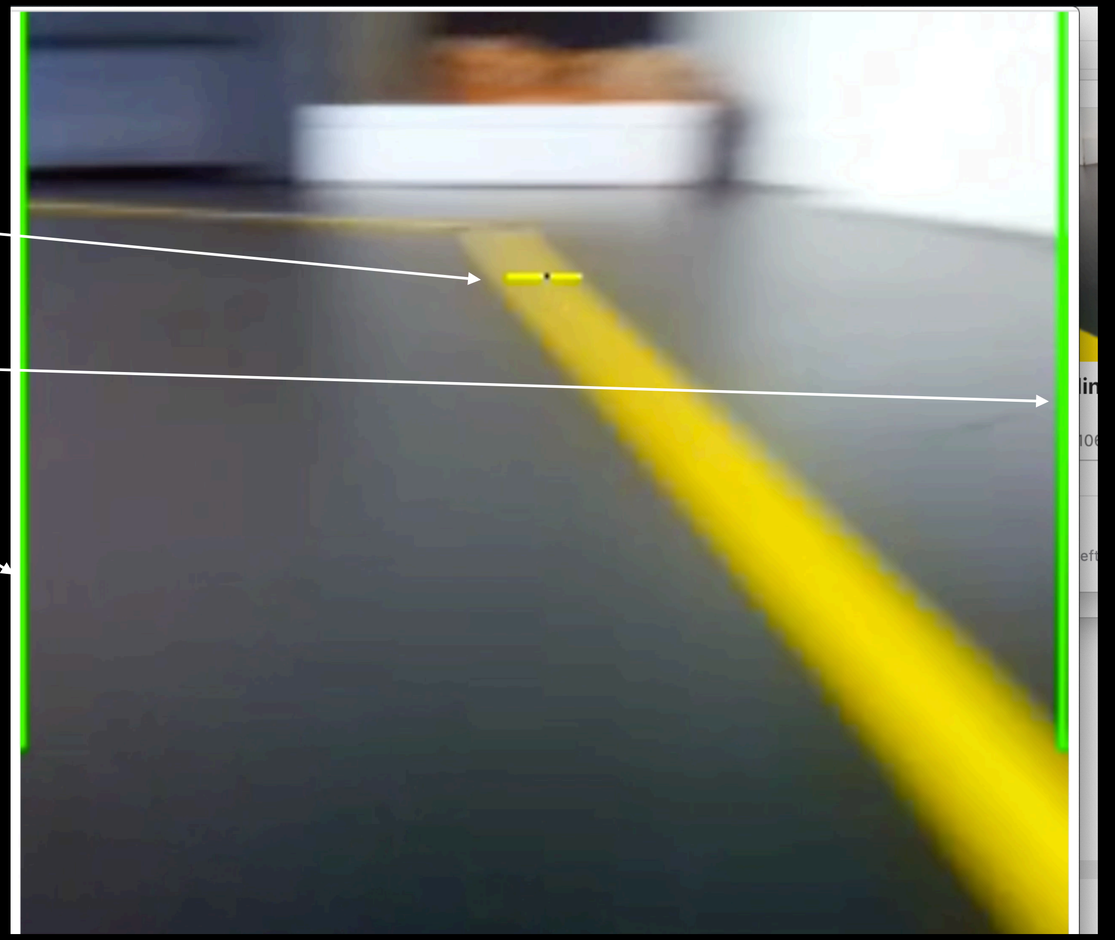
- V4l2-ctl -c saturation=80
- Read RGB from /dev/video0
- Converts a line to HSV
- Looks for largest group of yellow pixels
- Steers towards centre of yellow
- Write to /sys/class/pwmchip0/pwm[01]/duty_cycle
- 30 fps

```
private int toHSV(byte[] frame, int midlineStart) {
    int last = midlineStart + lineLen;
    int yellowStart = 0;
    int yellowLength = 0;
    int maxLen = 0;
    int center = -1;
    boolean inBlob = false;
    int pixel = 0;
    for (int i = midlineStart; i < last; i += bpp) {
        int r = Byte.toUnsignedInt(frame[i]);
        int g = Byte.toUnsignedInt(frame[i + 1]);
        int b = Byte.toUnsignedInt(frame[i + 2]);
        float[] hsv = rgb_to_hsv(r, g, b);
        boolean y = isYellow(hsv);
        if (y) {
            if (inBlob) {
                yellowLength++;
            } else {
                inBlob = true;
                yellowStart = pixel;
                yellowLength = 1;
            }
        } else {
            if (inBlob) {
                if (yellowLength > maxLen) {
                    center = yellowStart + (yellowLength / 2);
                    maxLen = yellowLength;
                }
                inBlob = false;
                yellowLength = 0;
            }
        }
        if (y) {
            if (high != null) {
                int rgb = high.getRGB();
                frame[i] = (byte) ((rgb & 0xff0000) >> 16);
                frame[i + 1] = (byte) ((rgb & 0x00ff00) >> 8);
                frame[i + 2] = (byte) (rgb & 0x0000ff);
            }
        }
        pixel++;
    }
    // cover the case that it is yellow to the right hand edge
    if (inBlob && (yellowLength > maxLen)) {
        center = yellowStart + (yellowLength / 2);
    }
```

But you still want to know what it is thinking

Patch the video on the way through

- Paint target bright yellow
- Paint sides with motor speed indicators
- Use v4l2 to turn up the hue



adrive

Only WebRTC Can Do This

In browser, low latency, P2P live video

Part 2

Cameras and Encoders

- At Commcon in 2 weeks (commcon.xyz)
  Both talks available as video/slides (after commcon)

Thanks! Questions?

I’m a bit deaf, (too many bad echo cancellers) so SHOUT!

- Contact:
  - tim@pi.pe
  - @steely_glint@chaos.social
- Consulting on opensource WebRTC and pi.pe