

# Embedded Video Systems With Zephyr

Josuah Demangeon, Panoramix Labs, tinyVision.ai

# Background

A contractor working essentially with tinyVision.ai



Autoportrait: I am curious about a lot of topics, but only scratch the surface.

You are welcome and invited to dive in depth!

# Video Systems

Famous example: home cinema

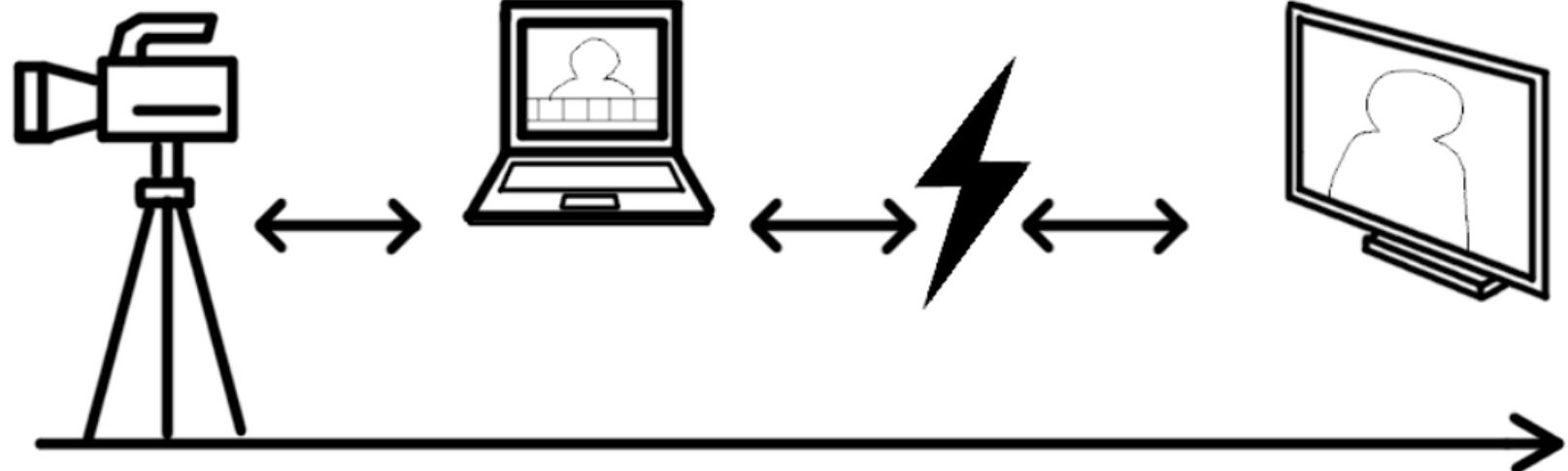


Pro Video  
Camera

Video editing  
software

Distribution  
network

Final destination:  
human eyes

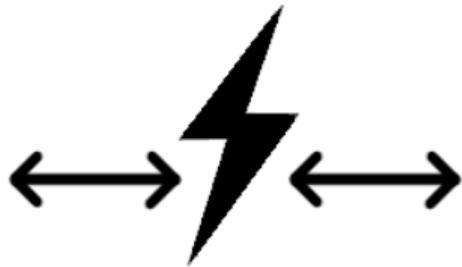


Pro Video  
Camera

Video editing  
software

Distribution  
network

Final destination:  
human eyes

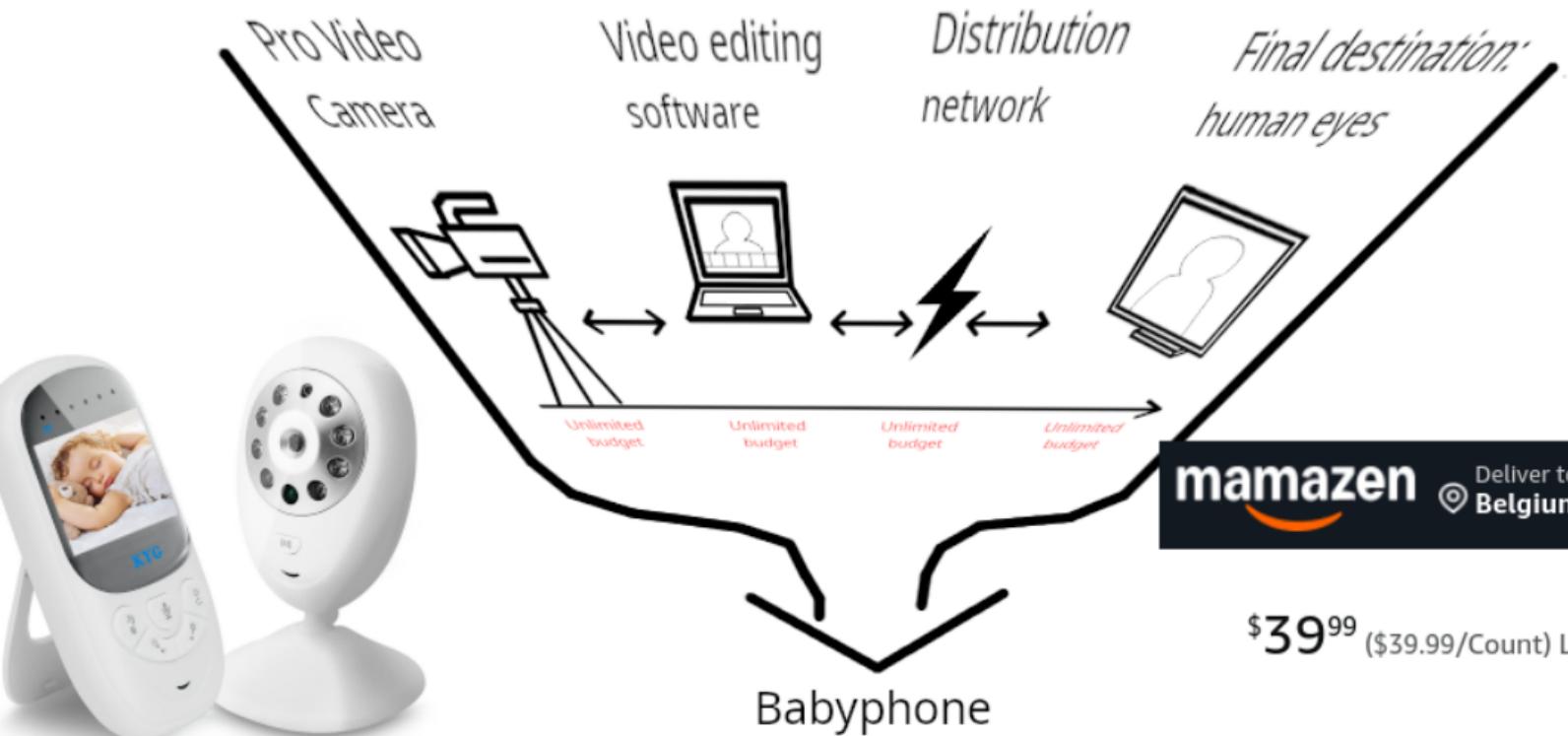


Unlimited  
budget

Unlimited  
budget

Unlimited  
budget

Unlimited  
budget



**mamazon** Deliver to Belgium

\$39<sup>99</sup> (\$39.99/Count) List: \$49.99

All ba



# Embedded Video Systems

Constraints:

- Cost budget
- Processing budget
- Time budget (low-latency, real-time)

Can only work at low-resolution...

# Embedded Video Systems

Constraints:

- Cost budget
- Processing budget
- Time budget (low-latency, real-time)

Can only work at low-resolution... <- FALSE!

Pro Video  
Camera



Video editing  
software



Distribution  
network



Final destination:  
human eyes



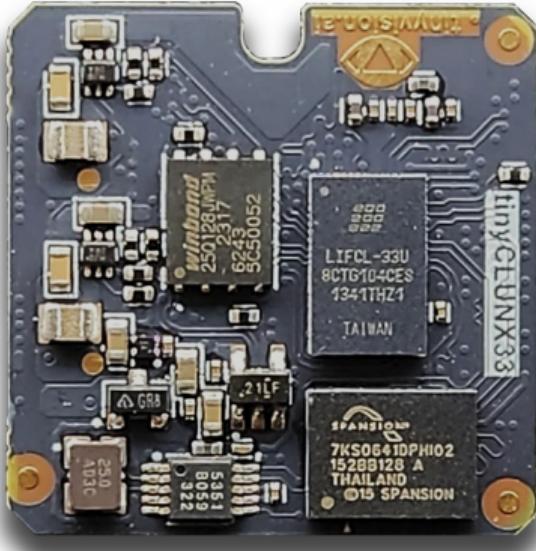
Needs an operating system too!

Embedded is not always low-end.

# Embedded Video Systems

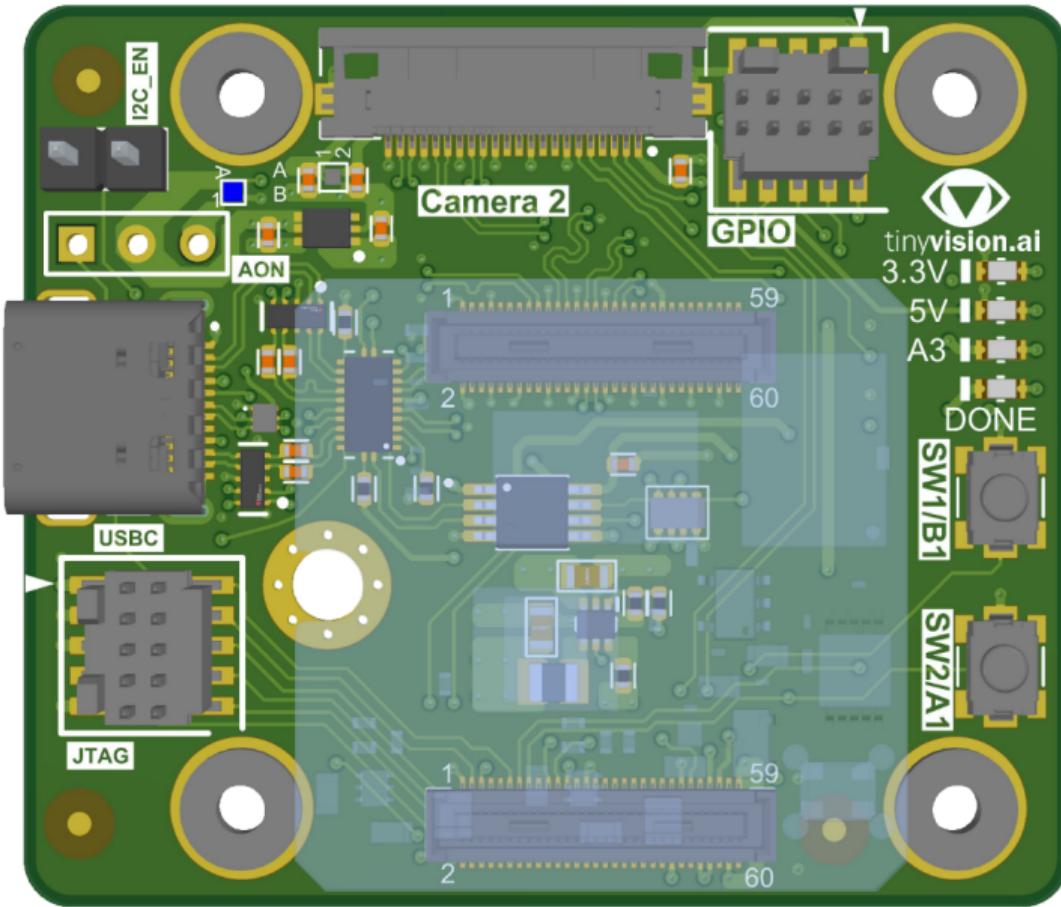
"Why not use an USB camera?"

We are now implementing the USB camera *itself*.



tinyCLUNX33: the heart of an USB 3 webcam.

3.4 Gbit/s under 200 mW





# Embedded Video Systems

"Why not just a Raspberry Pi?"

→ Power budget

→ Performance

→ Cost

→ Latency

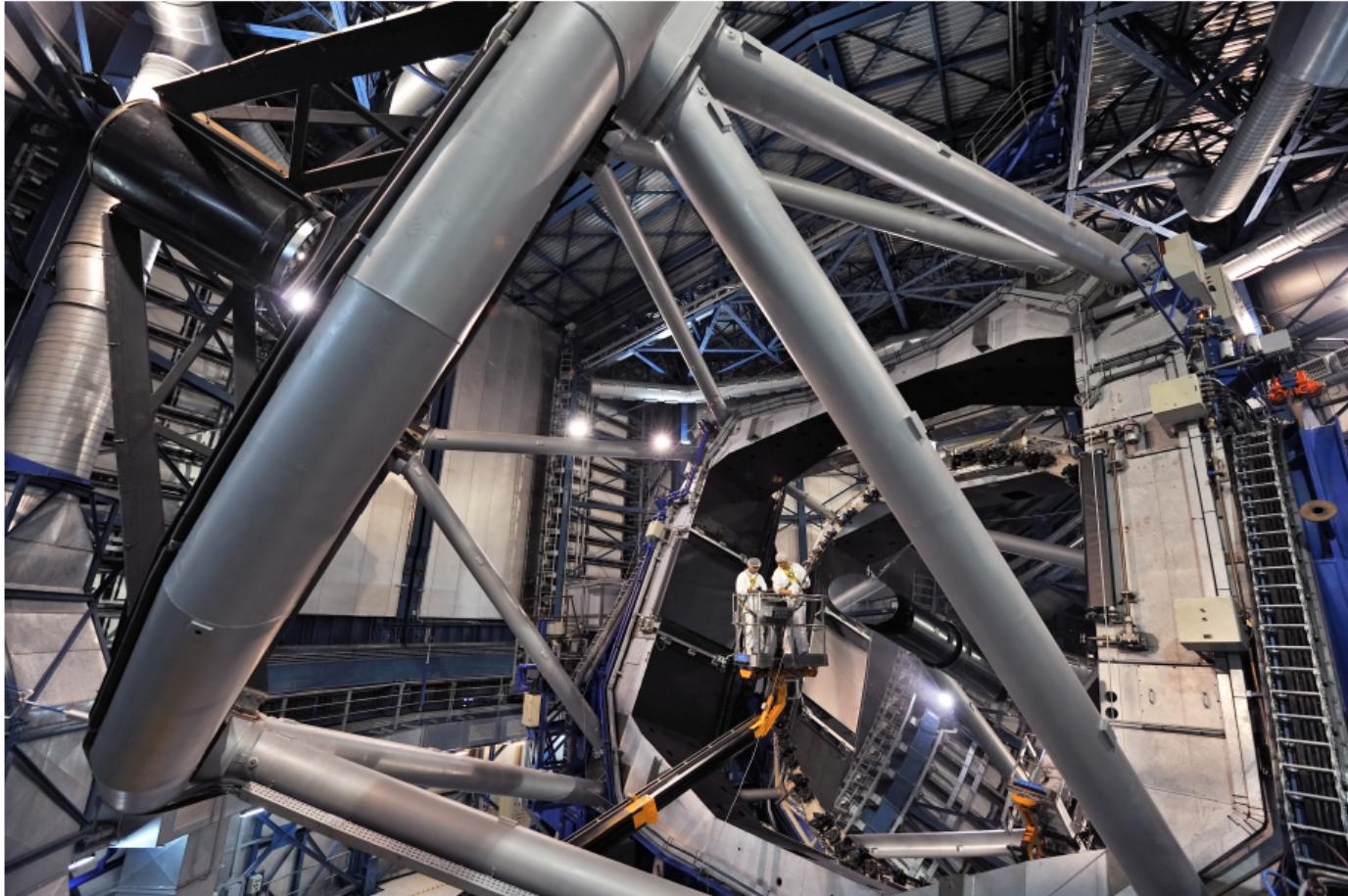




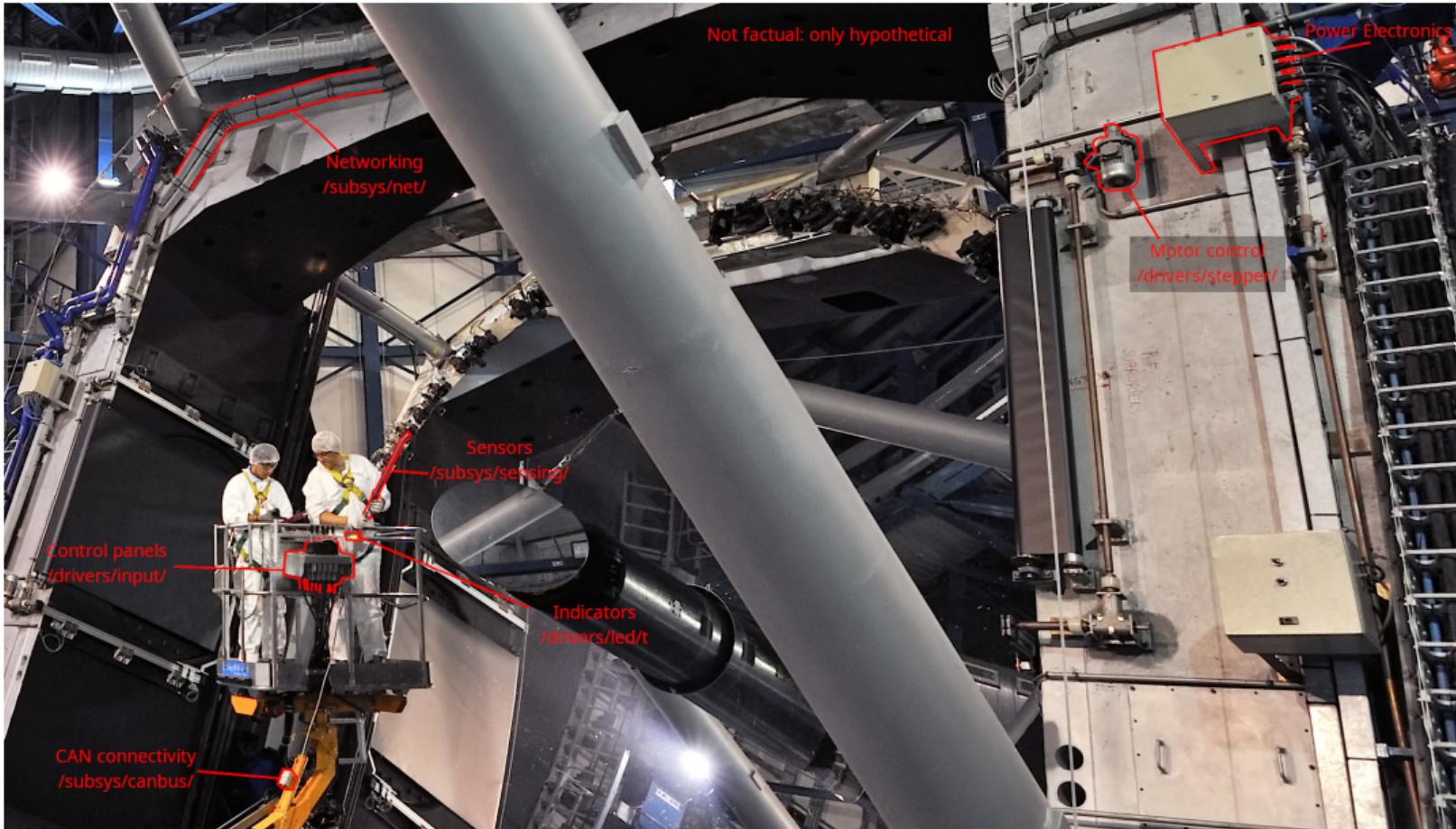
# Embedded Video Systems

Can be very large:





We can imagine a lot involved to assist the video function:



Still there on small embedded systems:

→ Motor for auto-focus ("VCM" motor

```
#include <zephyr/drivers/video-controls.h>)
```

→ I2C communication with other chips

```
(#include <zephyr/drivers/i2c.h>)
```

→ Turning on/off the chip power ([Power Management](#))

# Embedded Video Systems

But usually the smaller the better: how to shrink?

Switch from Linux OS → RTOS like Zephyr

Which board running it can do video?

# Video-capable boards running Zephyr

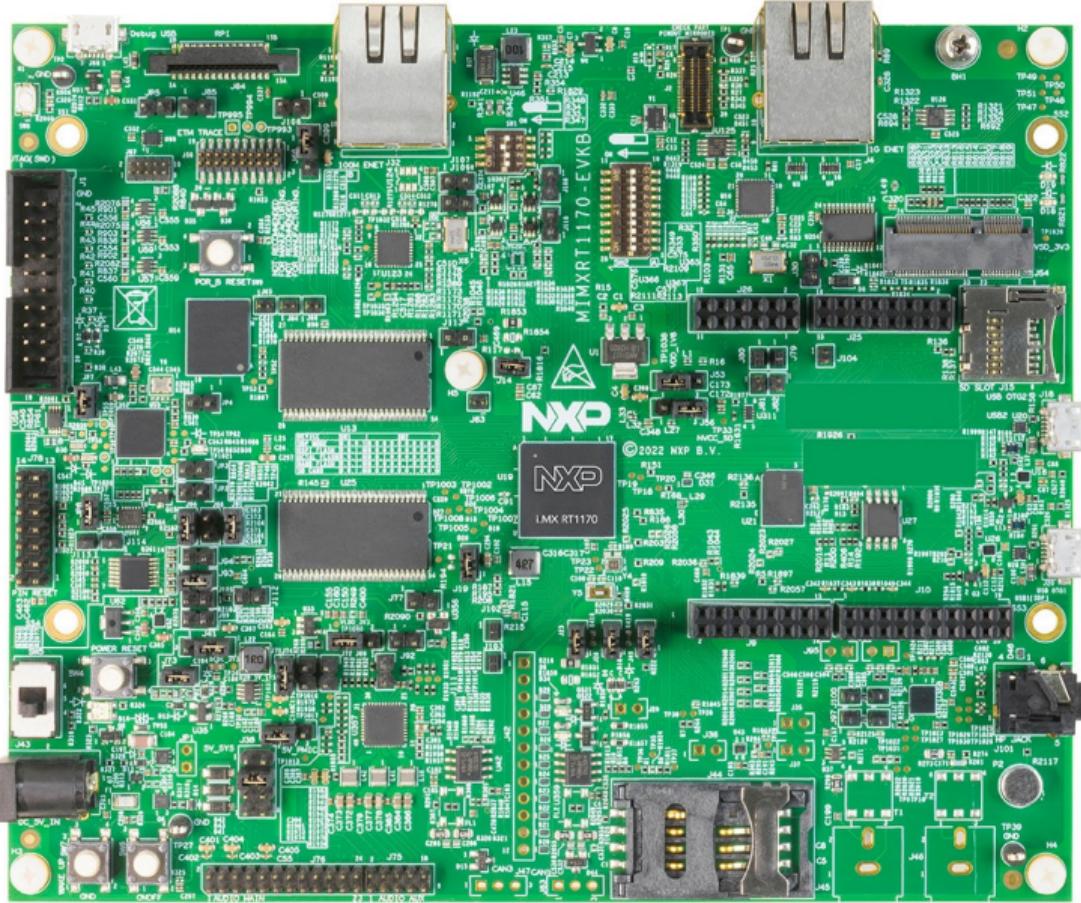
Video-capable means DVP (a.k.a. parallel port) or MIPI support.

How small can we shrink (size, price, heat, power usage)?

# i.MX RT1170

Cortex-M7 (small-medium) running at 1 GHz.

A fast CPU is good to reduce RAM usage: transmit *more often* rather than *more at once*.



MIPI camera input (nxp,mipi-csi2rx)



MIPI display output (nxp,imx-mipi-dsi) 1500 MHz, 2-lanes



USB2 (nxp,ehci)



Ethernet (nxp,enet1g)



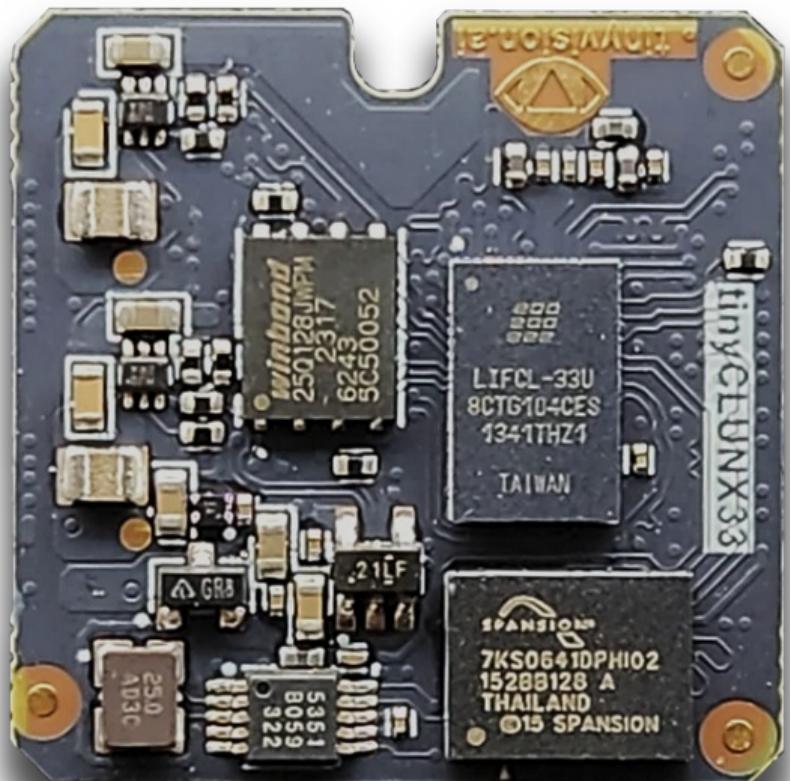
CPU cores (arm,cortex-m7 + arm,cortex-m4)



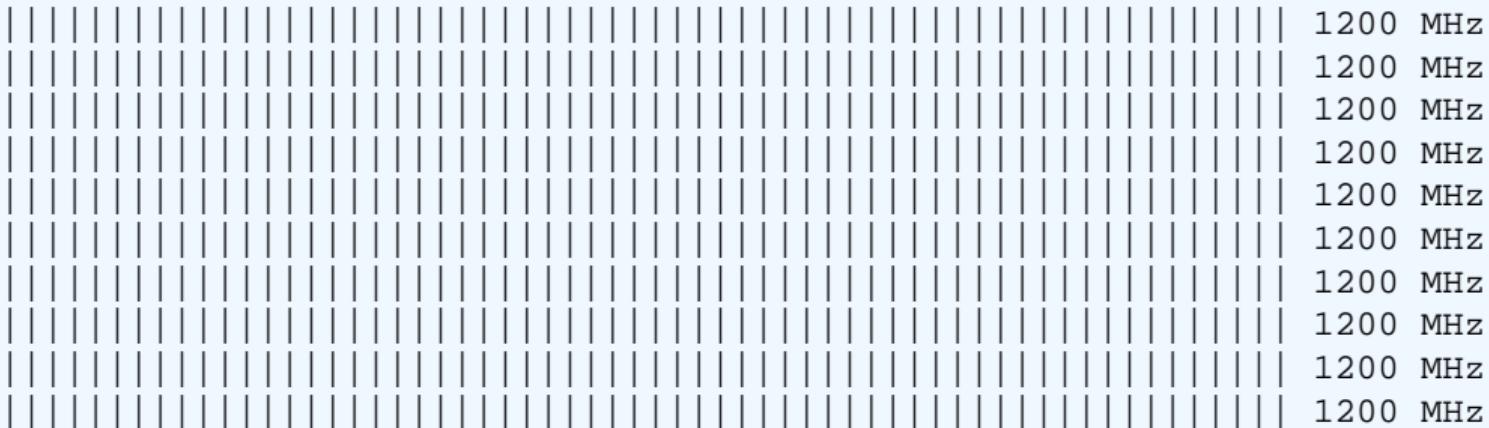
+ Video processing cores (cropping, resizing, color conversion)

# tinyVision.ai tinyCLUNX33

A system specialized for MIPI to USB3 camera systems. An FPGA: very slow CPU and needs to "build your own video cores". Not upstream yet.



MIPI (tinyvision, uvcmanager)



USB3 (lattice, usb23)



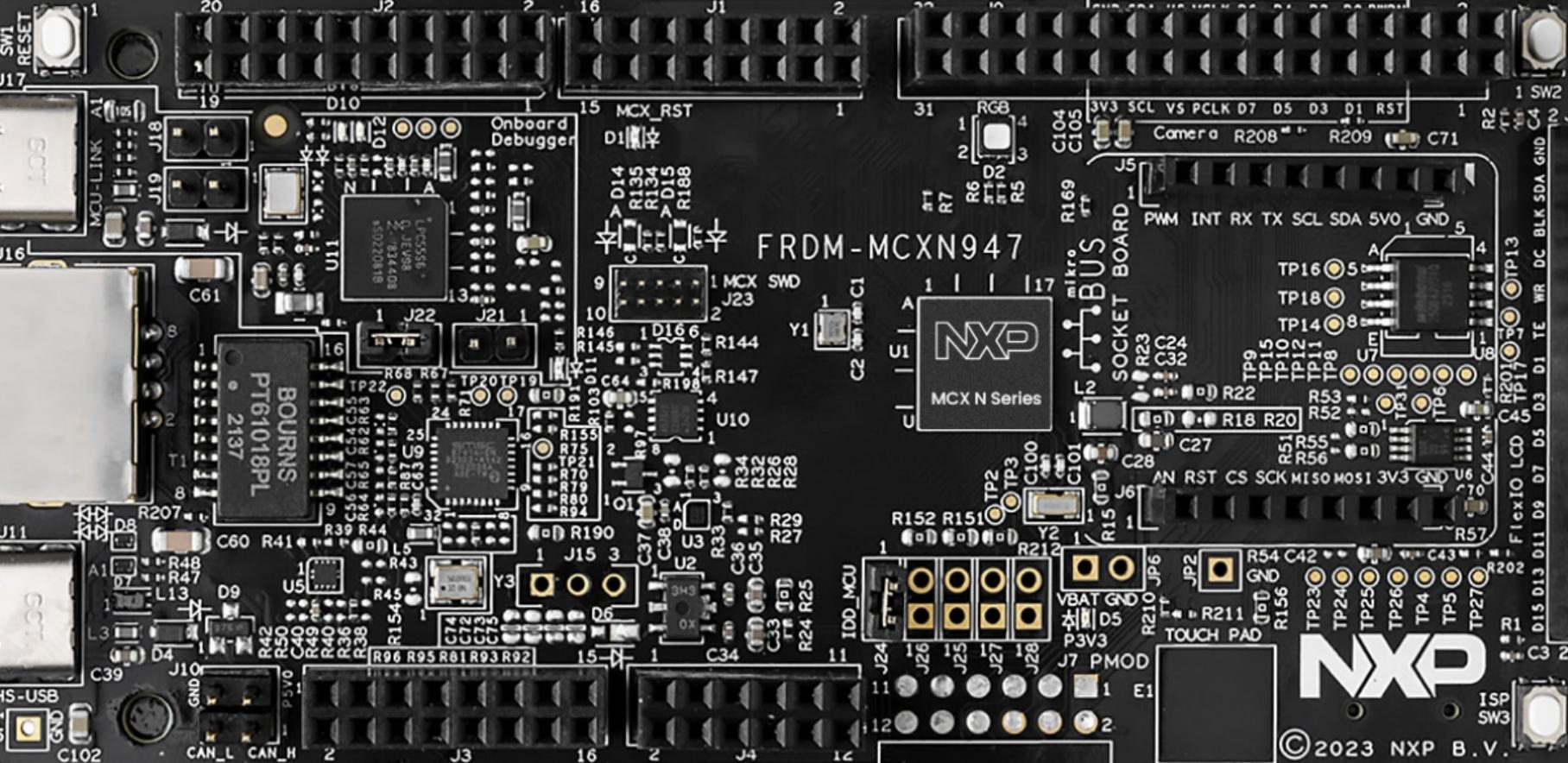
CPU core (tinyvision, vexriscv)





# FRDM-MCXN947

Dual Cortex-M33 (small) system with peripherals usually only found on larger Linux-capable devices: "do more with less"



© 2023 NXP B.V.

DVP camera input (nxp,video-smartdma)

||||||| ||||| 8 pins (16 max), 150 MHz each

USB2 (nxp,ehci)

||||||| 480 MHz

Ethernet (nxp,enet-qos)

|||| 100 MHz

CPU cores (arm,cortex-m33f)

||||||| 150 MHz

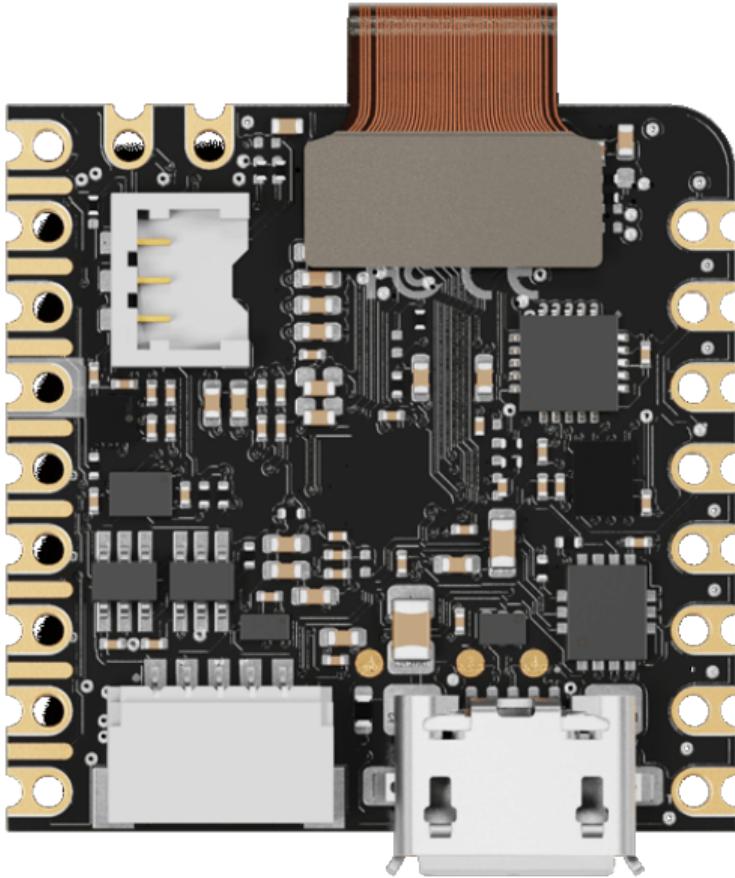
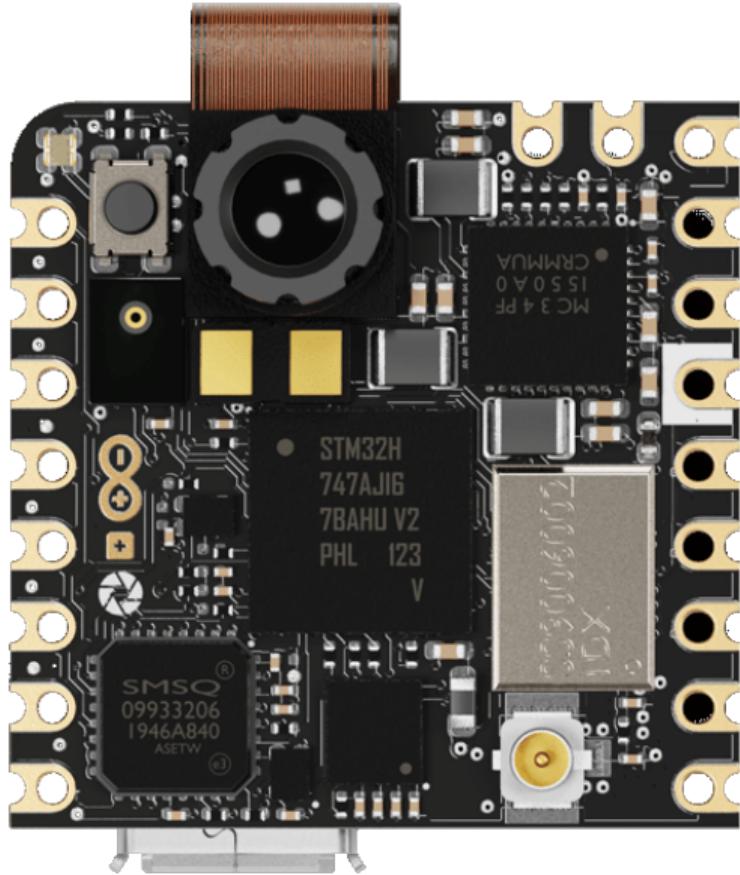
||||||| 150 MHz

+ eIQ NPU on-board for A.I. inference (release planned 2025 [1])

[1]: [eIQ application note](#)

# Arduino Nicla Vision (STM32H747)

All-in-one board with IMU, microphone, 2 MP camera built-in, fast USB.



DVP camera input (st,stm32-dcmi)  
||||| ||||| ||||| ||||| ||||| ||||| ||||| 8-pins (14 max), 80 MHz each

USB2 (st,stm32-otghs)  
||||||| ||||| ||||| ||||| 480 MHz

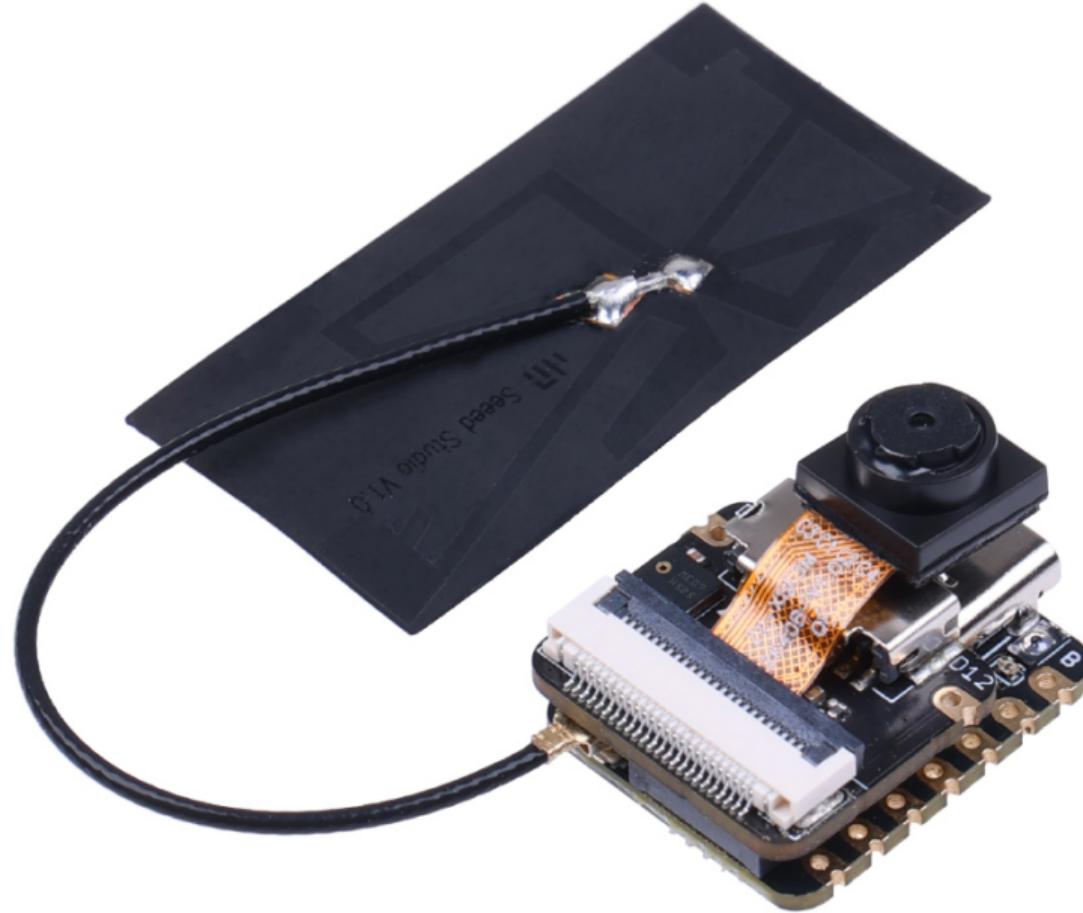
Wi-Fi (murata,1dx)  
|||| 65 Mbit/s

CPU cores (arm,cortex-m7 + arm,cortex-m4)  
||||||| ||||| ||||| ||||| ||||| 480 MHz  
||||||| ||||| 240 MHz

- + JPEG compression core
- + Video processing operations (cropping, resizing, color conversion)

# XIAO ESP32S3 Sense

Self-contained board for wireless (WiFi, Bluetooth), coming with a camera and microphone.



DVP (espressif, lsd-cam)

||||| ||||| ||||| ||||| ||||| ||||| ||||| 8 pins (16 max) 80 MHz each

Wi-Fi (espressif, esp32-wifi)

||||||| 150 Mbit/s

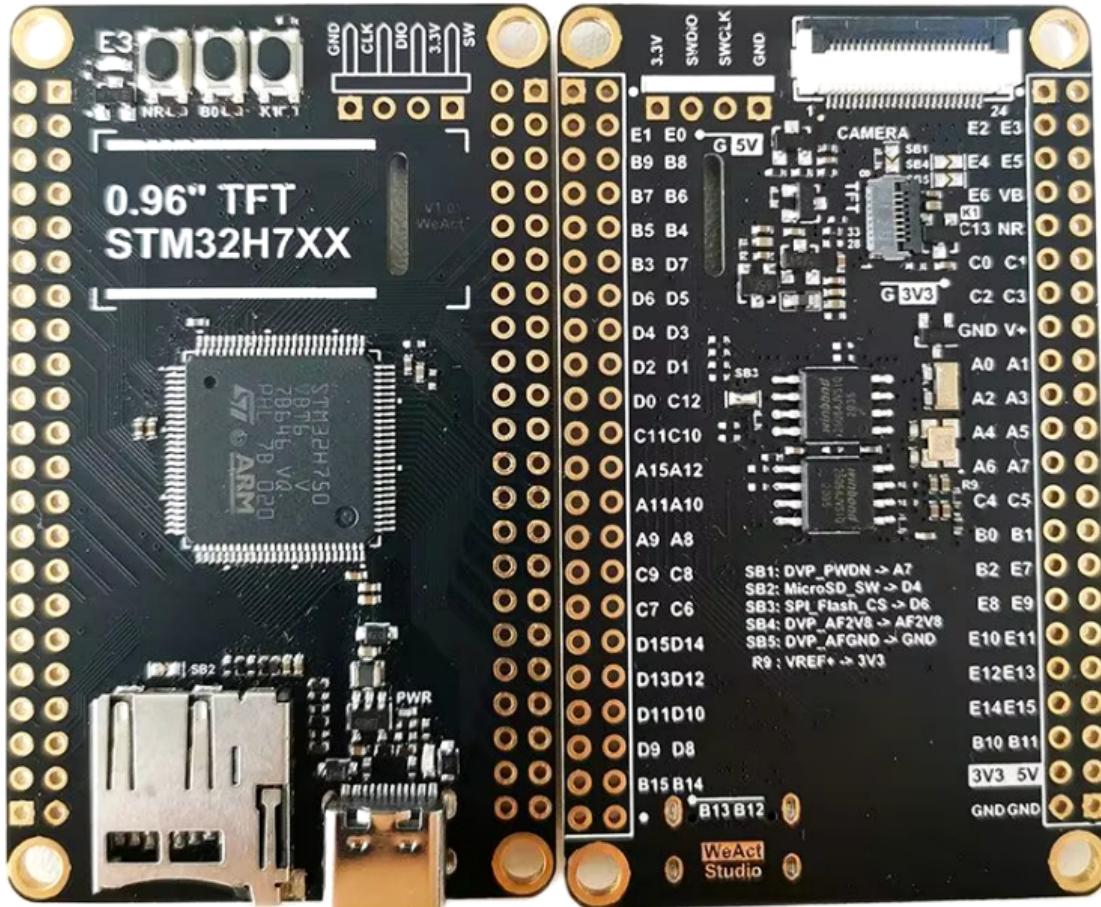
CPU core (espressif, xtensa-lx7 + espressif, xtensa-lx7)

||||||| | 240 MHz

||||||| | 240 MHz

# WeAct MiniSTM32H7xx

Minimalist approach to a video devboard, comes with a camera and a display and fast USB.



DVP camera input (st,stm32-dcmi)  
||||| ||||| ||||| ||||| ||||| ||||| ||||| 8 pins (14 max), 80 MHz each

USB2 (st,stm32-otghs / st,stm32-otghs)  
||||| ||||| ||||| ||||| ||||| ||||| 480 MHz  
| 12 MHz

Ethernet (st,stm32h7-ethernet)  
||||| 100 MHz

CPU core (arm,cortex-m7)  
||||| ||||| ||||| ||||| ||||| ||||| 480 MHz

+ JPEG compression core  
+ Video processing operations (cropping, resizing, color conversion)

# Different software ecosystem

FFmpeg → ???

Gstreamer → ???

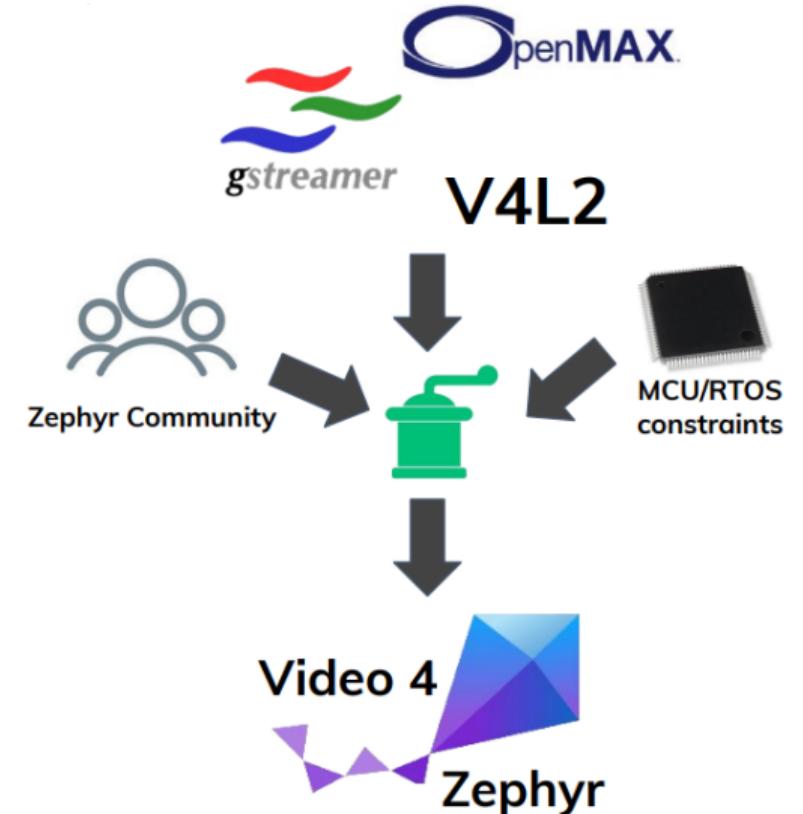
OpenCV → ???

PyTorch → ???

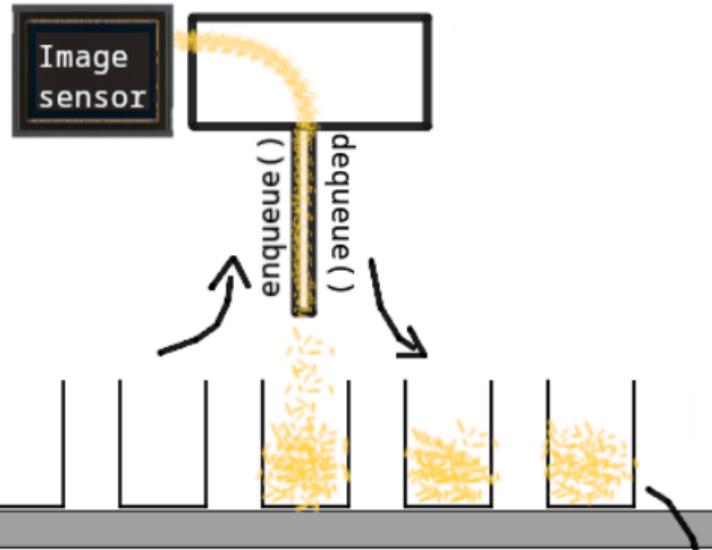
/dev/video0 → ???

These softwares assume a lot of resources. Everything to reinvent!

# Zephyr Video APIs

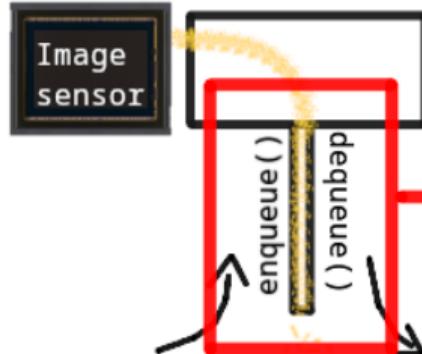


MIPI driver



Recycling buffers:  
not reallocating: reusing

MIPI driver



Data API

not implemented  
by all drivers

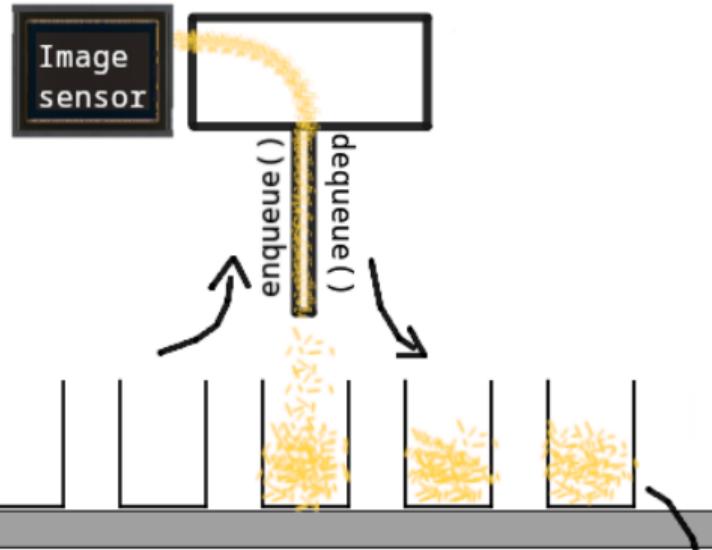


Recycling buffers:  
not reallocating: reusing

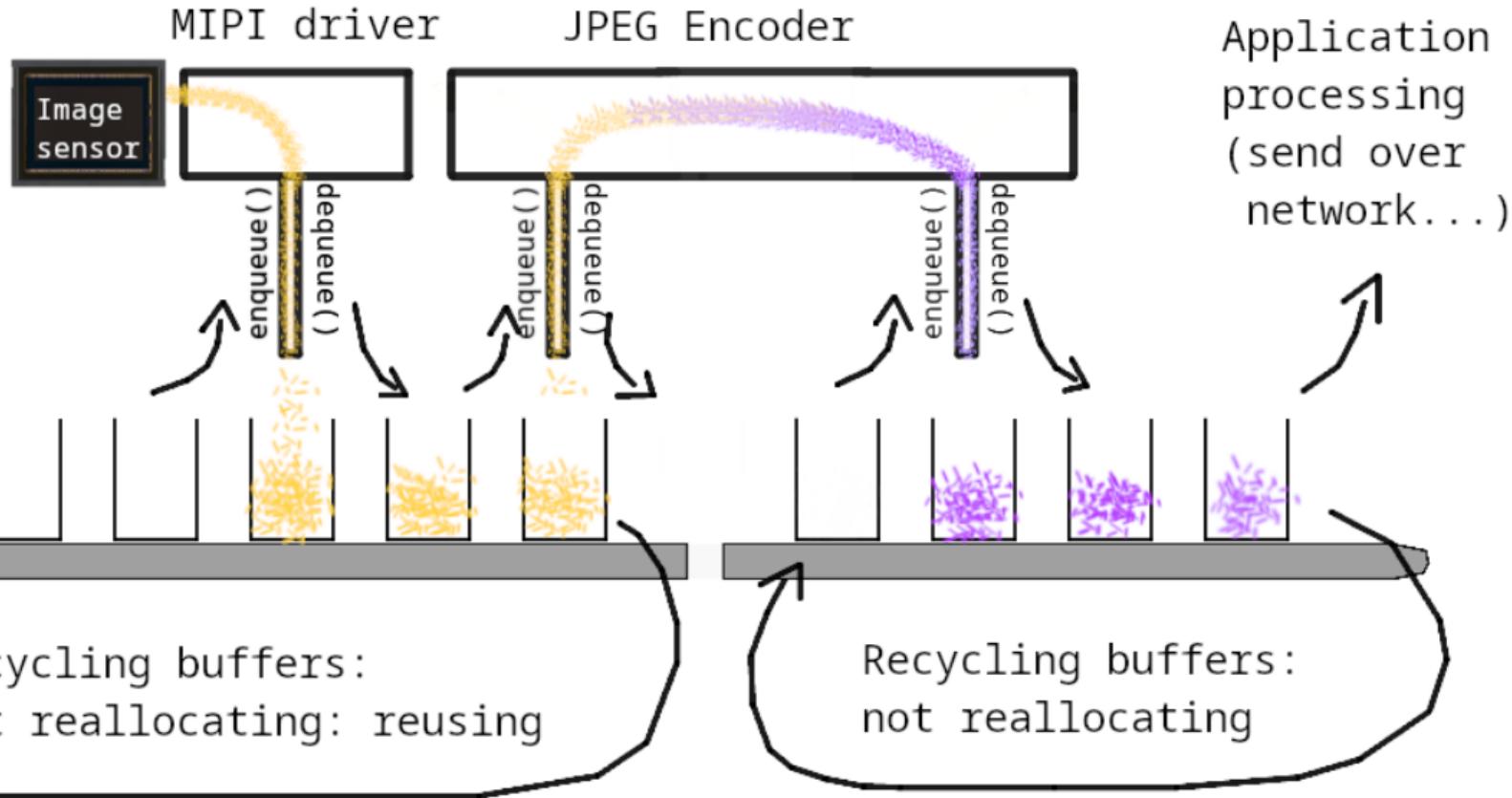
```
/* Error handling omitted */

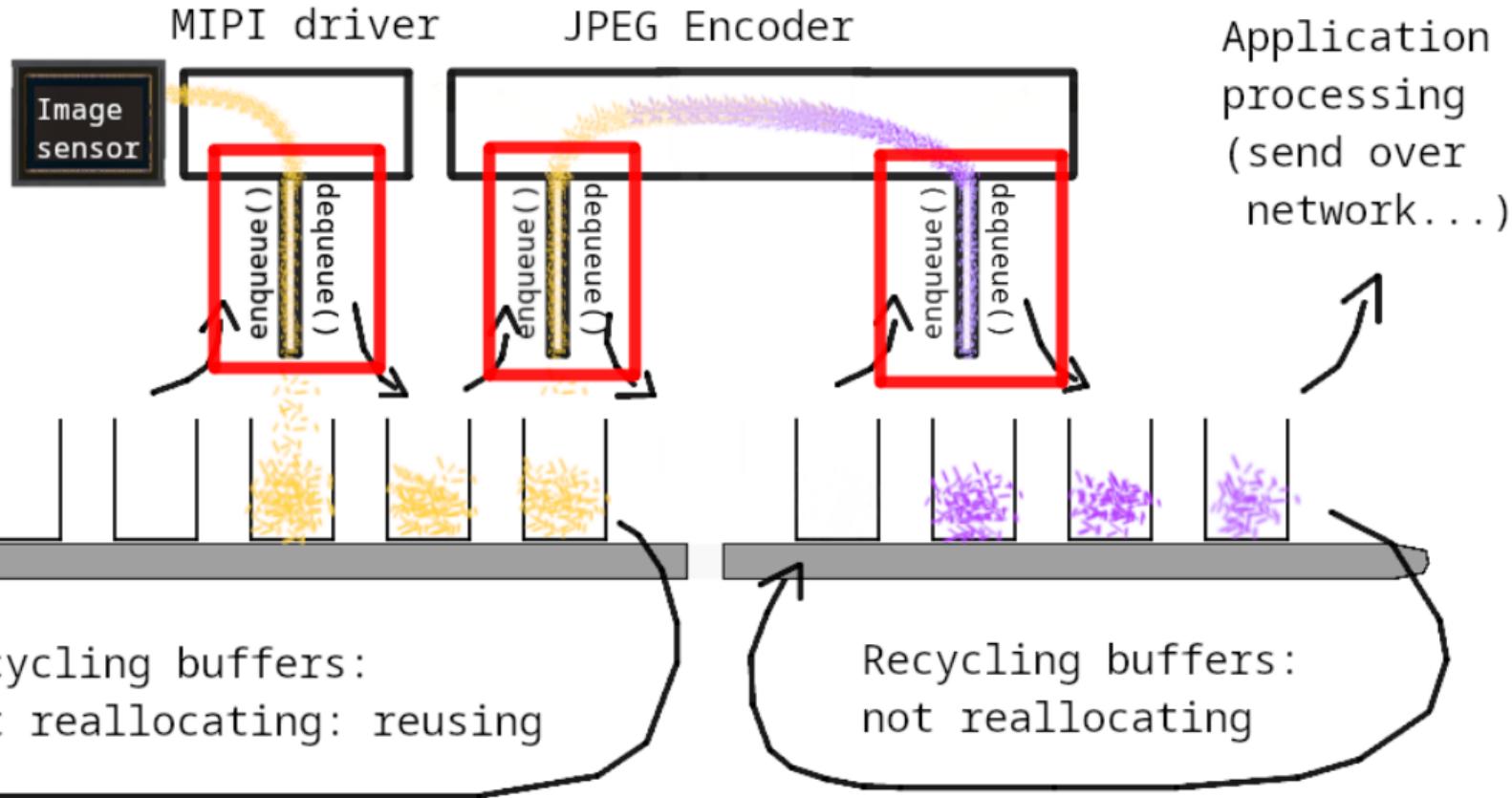
video_stream_start(mipi_dev);
vbuf = video_buffer_alloc(1280 * 720 * 3, K_FOREVER);
while (true) {
    video_enqueue(mipi_dev, VIDEO_EP_OUT, vbuf);
    video_dequeue(mipi_dev, VIDEO_EP_OUT, &vbuf, K_FOREVER);
    /* do something with vbuf->data */
}
```

MIPI driver



Recycling buffers:  
not reallocating: reusing





```
/* Only one buffer of each */
vbuf_raw = video_buffer_alloc(1280 * 720 * 3, K_FOREVER);
vbuf_jpeg = video_buffer_alloc(JPEG_BUF_SIZE, K_FOREVER);
while (true) {
    video_enqueue(mipi_dev, VIDEO_EP_OUT, vbuf_raw);
    video_dequeue(mipi_dev, VIDEO_EP_OUT, &vbuf_raw, K_FOREVER);
    video_enqueue(jpeg_dev, VIDEO_EP_IN, vbuf_raw);
    video_dequeue(jpeg_dev, VIDEO_EP_IN, &vbuf_raw, K_FOREVER);
    video_enqueue(jpeg_dev, VIDEO_EP_OUT, vbuf_jpeg);
    video_dequeue(jpeg_dev, VIDEO_EP_OUT, &vbuf_jpeg, K_FOREVER);
    /* Do something with the JPEG data */
}
```

Nice and simple, but no concurrency: a lot of time spent waiting.

```
for (int i = 0; i < 3; i++) {
    vbuf_raw = video_buffer_alloc(1280 * 720 * 3, K_FOREVER);
    video_enqueue(mipi_dev, VIDEO_EP_OUT, vbuf_raw);
}

for (int i = 0; i < 3; i++) {
    vbuf_jpeg = video_buffer_alloc(JPEG_BUF_SIZE, K_FOREVER);
    video_enqueue(jpeg_dev, VIDEO_EP_OUT, vbuf_jpeg);
}

/* Thread 1: Handle the filled buffers */
while (true) {
    video_dequeue(mipi_dev, VIDEO_EP_OUT, &vbuf_raw, K_FOREVER);
    video_enqueue(jpeg_dev, VIDEO_EP_IN, vbuf_raw);
}
```

```
/* Thread 2: Handle the emptied buffers */
while (true) {
    video_dequeue(jpeg_dev, VIDEO_EP_IN, &vbuf_raw, K_FOREVER);
    video_enqueue(mipi_dev, VIDEO_EP_OUT, vbuf_raw);
}

/* Thread 3: Handle the JPEG buffers */
while (true) {
    video_dequeue(jpeg_dev, VIDEO_EP_IN, &vbuf_jpeg, K_FOREVER);
    app_use_jpeg_buffer(vbuf_jpeg);
    video_enqueue(jpeg_dev, VIDEO_EP_OUT, &vbuf_jpeg, K_FOREVER);
}
```

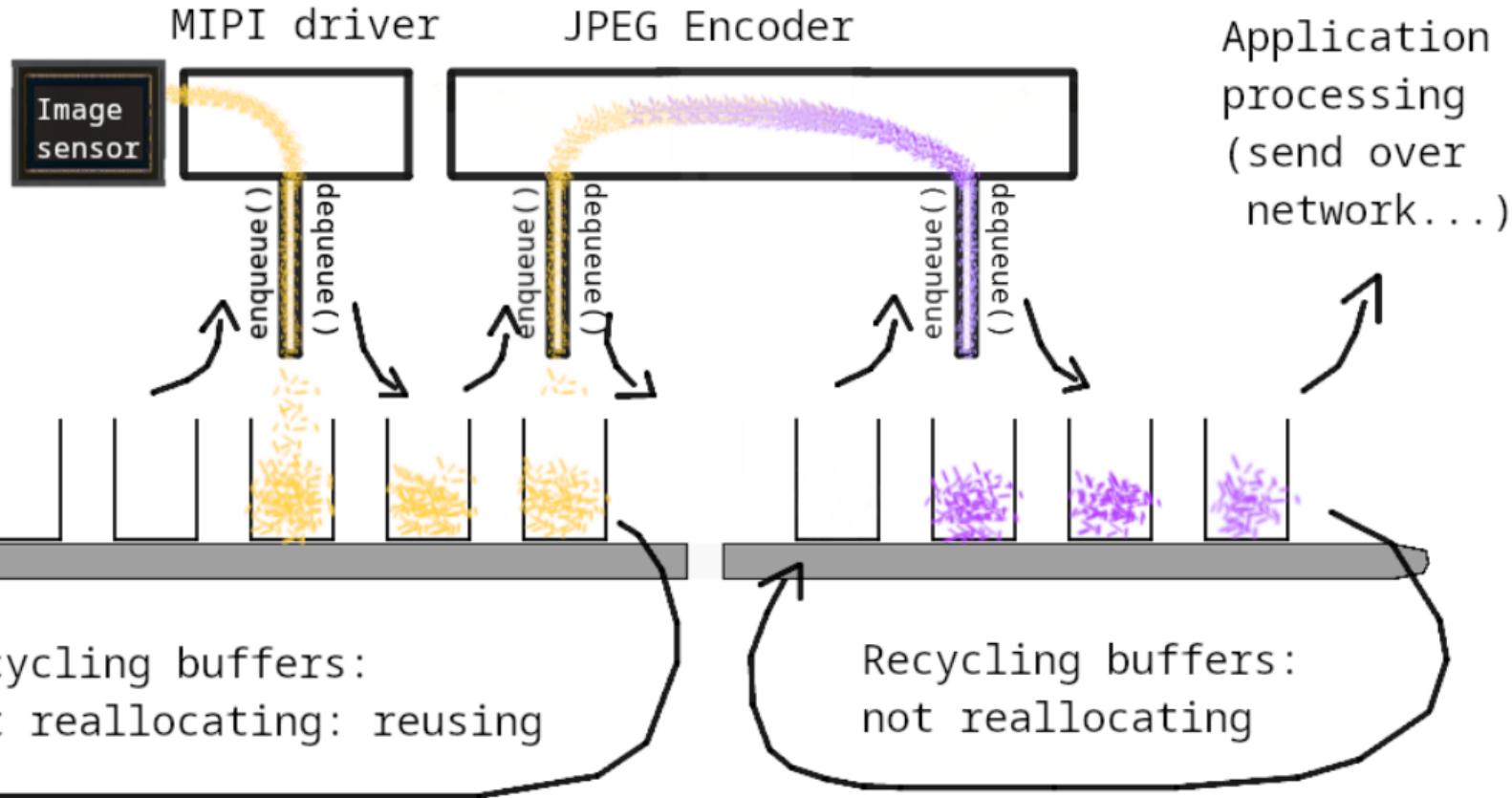
Fully parallel, but consumes a lot of threads: more memory overhead and context switching.

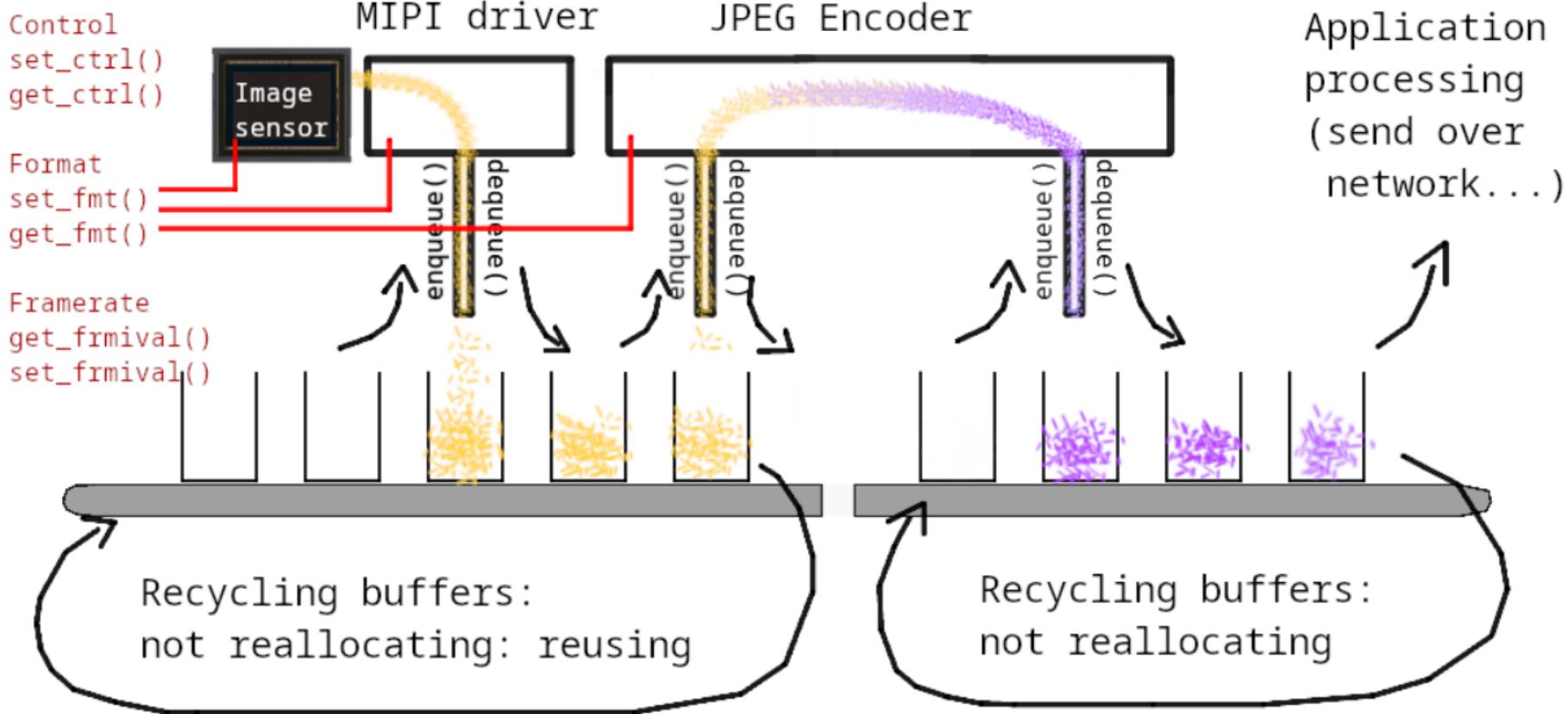
```
/* First prepare a few buffers */

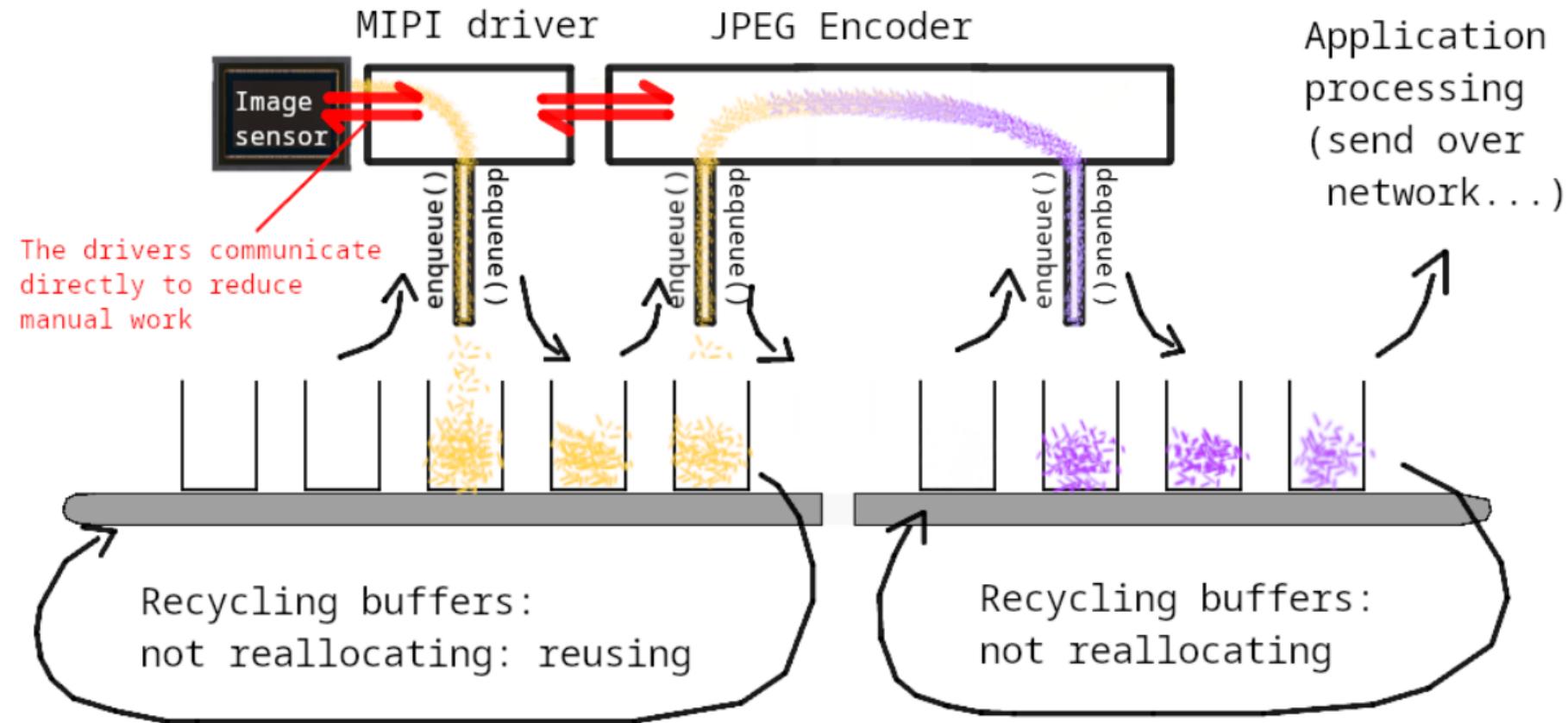
/* Subscribe to the events */
video_set_signal(mipi_dev, &signal);
video_set_signal(jpeg_dev, &signal);

/* React to the events */
while (k_poll(&events, 1, K_FOREVER) == 0) {
    if (video_dequeue(mipi_dev, VIDEO_EP_OUT, &vbuf_raw, K_NO_WAIT) == 0) {
        video_enqueue(jpeg_dev, VIDEO_EP_IN, vbuf_raw);
    }
    if (video_dequeue(jpeg_dev, VIDEO_EP_IN, &vbuf_raw, K_NO_WAIT) == 0) {
        video_enqueue(mipi_dev, VIDEO_EP_OUT, vbuf_raw);
    }
    if (video_dequeue(jpeg_dev, VIDEO_EP_OUT, &vbuf_jpeg, K_NO_WAIT) == 0) {
        app_use_jpeg_buffer();
        video_enqueue(jpeg_dev, VIDEO_EP_OUT, vbuf_jpeg);
    }
}
```

Good parallelism, single thread (more scalable). Maybe this can be automated for easier pipelines.







```
imx219: imx219@10 {
    compatible = "sony,imx219";
    port {
        imx219_ep_out: endpoint {
            remote-endpoint-label = "mipi0_ep_in";
        };
    };
};
```

```
mipi0: mipi@b1000010 {
    compatible = "tinyvision,mipi";
    port {
        mipi0_ep_in: endpoint {
            remote-endpoint-label = "imx219_ep_out";
        };
};
```

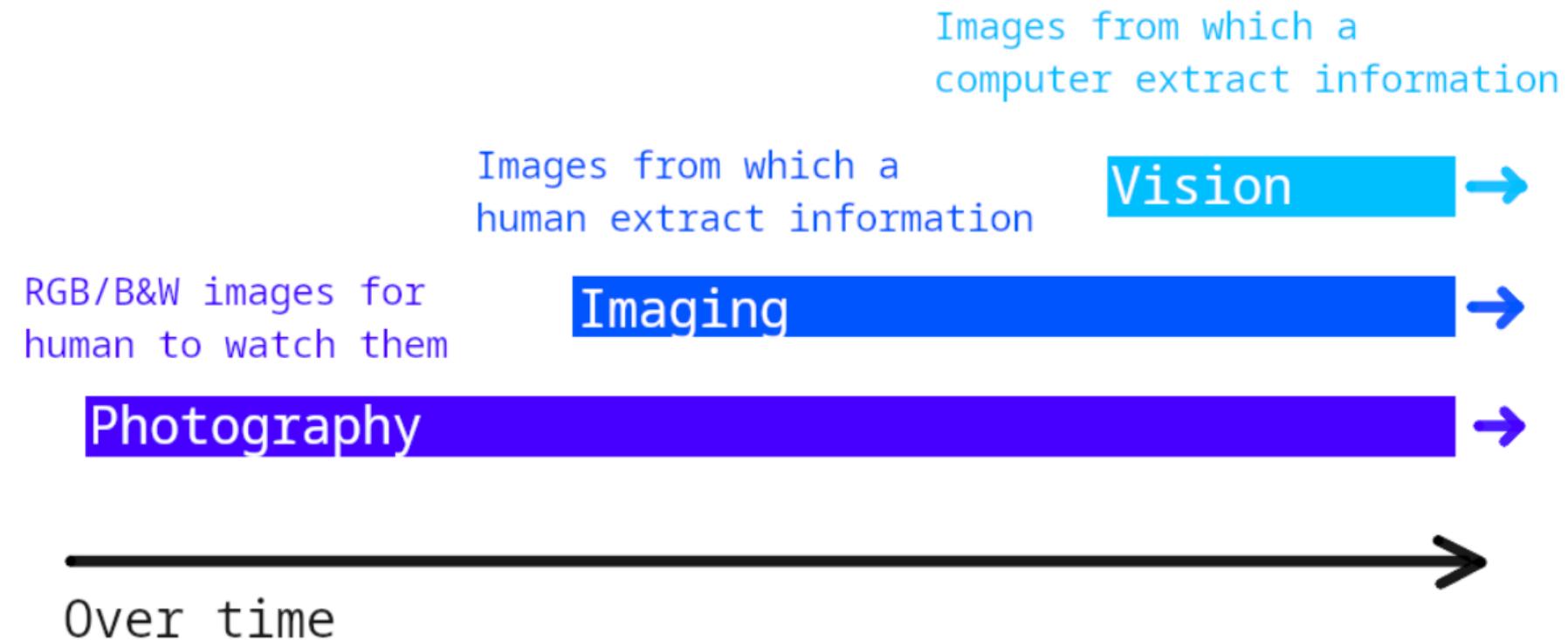
```
        mipi0_ep_out: endpoint {
            remote-endpoint-label = "jpegenc_ep_in";
        };
    };
};
```

```
jpegenc0: jpegenc@b1000010 {
    compatible = "tinyvision,jpegenc";
    port {
        jpegenc0_ep_in: endpoint {
            remote-endpoint-label = "mipi0_ep_in";
        };
        /* jpegenc0_ep_out: to the application */
    };
};
```

Next steps: automating more of the pipeline? What would make it more convenient?

# Systems doing what?

On a journey from Phontons to Video, and how that is used



Inspired from a MIPI alliance presentation (2018)

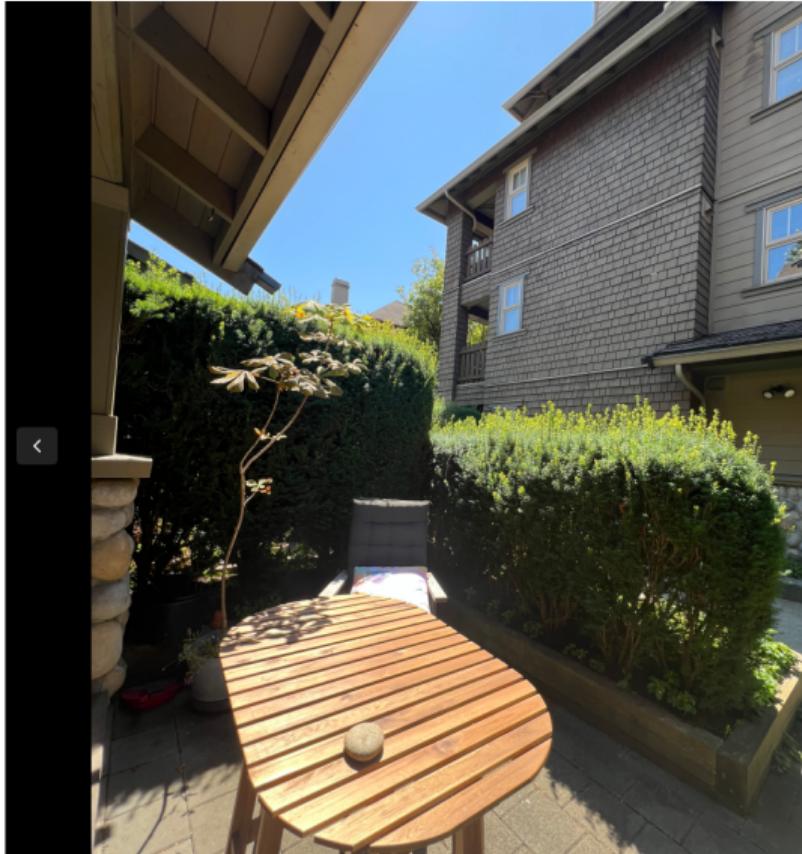
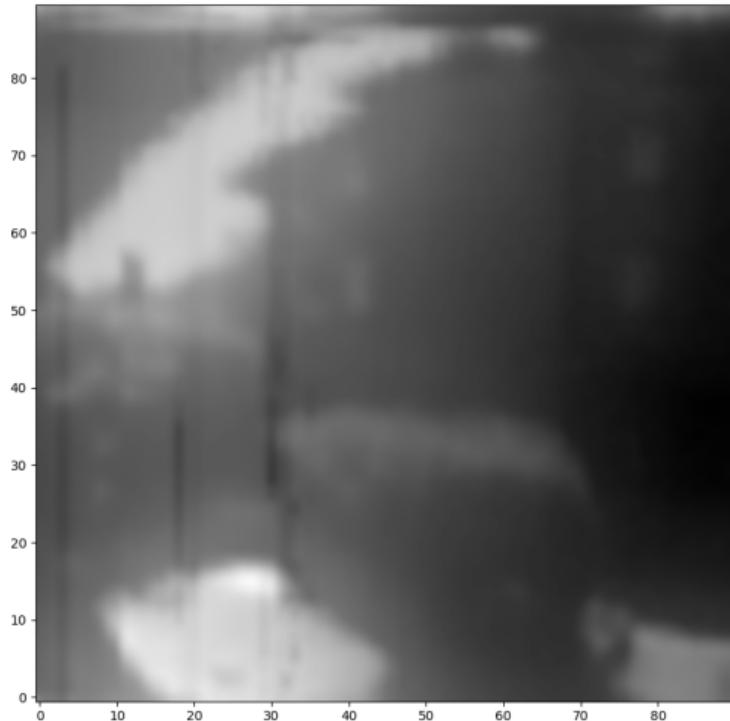


# Photodiode

Phenomenon of semiconductors producing voltage when exposed to the light.







Note: photoresistor instead of photodiode here

```
#include <zephyr/drivers/pwm.h> // if using servomotors
#include <zephyr/drivers/stepper.h> // if using stepper motors
#include <zephyr/drivers/adc.h> // measure the light intensity
```

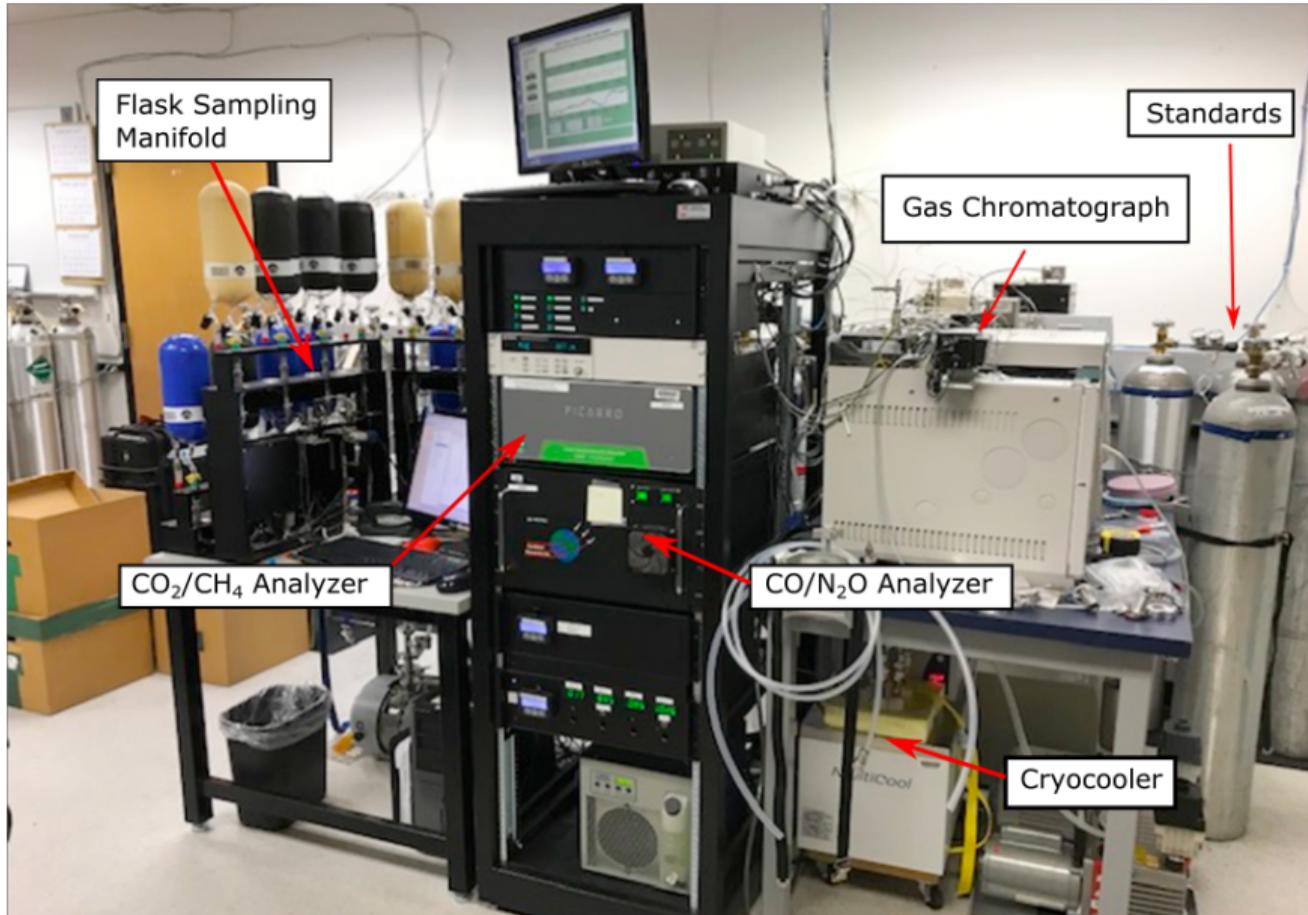
# Photons → Photonics

Much more than just video:

→ Gas detection/characterization, i.e. NDIR CO<sub>2</sub> sensors

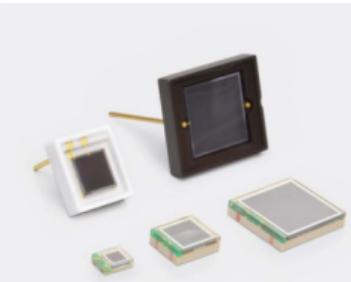
Industrial, safety, medical use-cases.

Since 1958: measuring Earth atmospheric CO<sub>2</sub> with "1-pixel image sensors"



→ Biology/medical research, i.e. DNA sequencing

# MPPC® (multi-pixel photon counter)



S13360 series

## MPPCs for precision measurement

MPPC is a type of device called SiPM (silicon photomultipliers). It is a new type of photon counting device that consists of multiple Geiger mode APD (avalanche photodiode) pixels. It is an opto-semiconductor with outstanding photon counting capability and low operating voltage and is immune to the effects of magnetic fields.

The S13360 series are MPPCs for precision measurement. The MPPCs inherit the superb low afterpulse characteristics of previous products and further provide lower crosstalk and lower dark count. They are suitable for precision measurement, such as flow cytometry, DNA sequencer, laser microscope, and fluorescence measurement, that requires low noise characteristics.

### Features

- Reduced crosstalk and dark count  
(compared to previous products)
- Outstanding photon counting capability (outstanding photon detection efficiency versus numbers of incident photons)
- Compact
- Operates at room temperature
- Low voltage ( $V_{DD} = 52 \text{ V}$  typ.) operation

### Applications

- Fluorescence measurement
- Laser microscopes
- Flow cytometry
- **DNA sequencers**
- Environmental analysis
- Various academic research

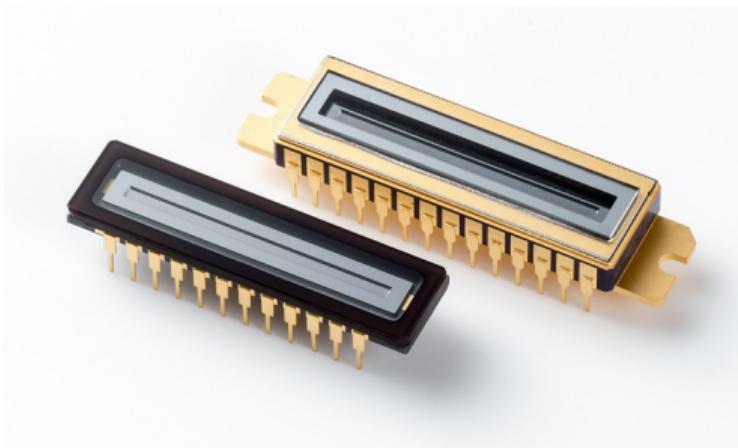
Sensing voltage: not a very Linux thing to do...

# A single line of pixels

Line sensors: a single line at a time.

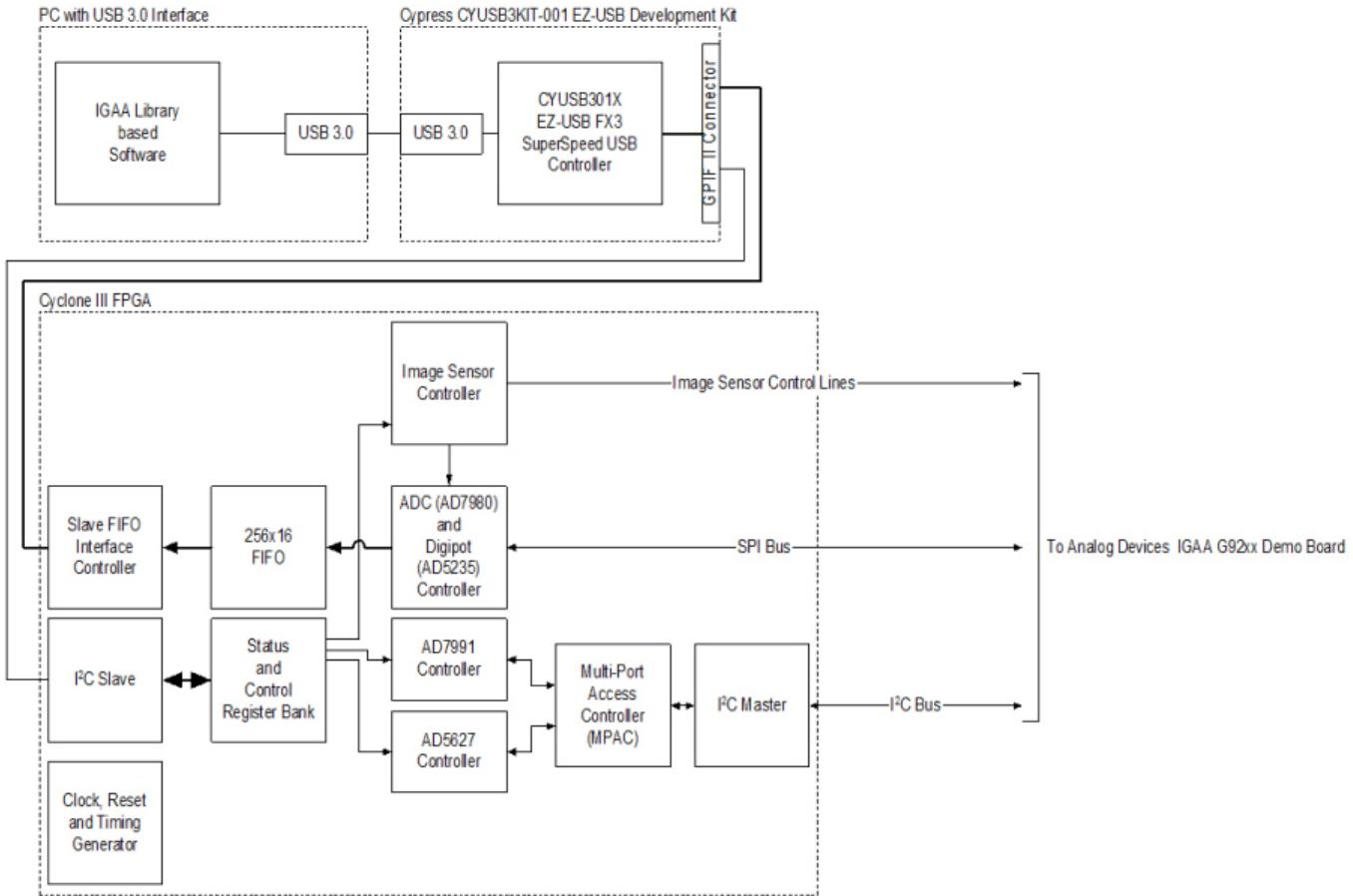
Not digital interface but analog interface: need an ADC to get digital readout.

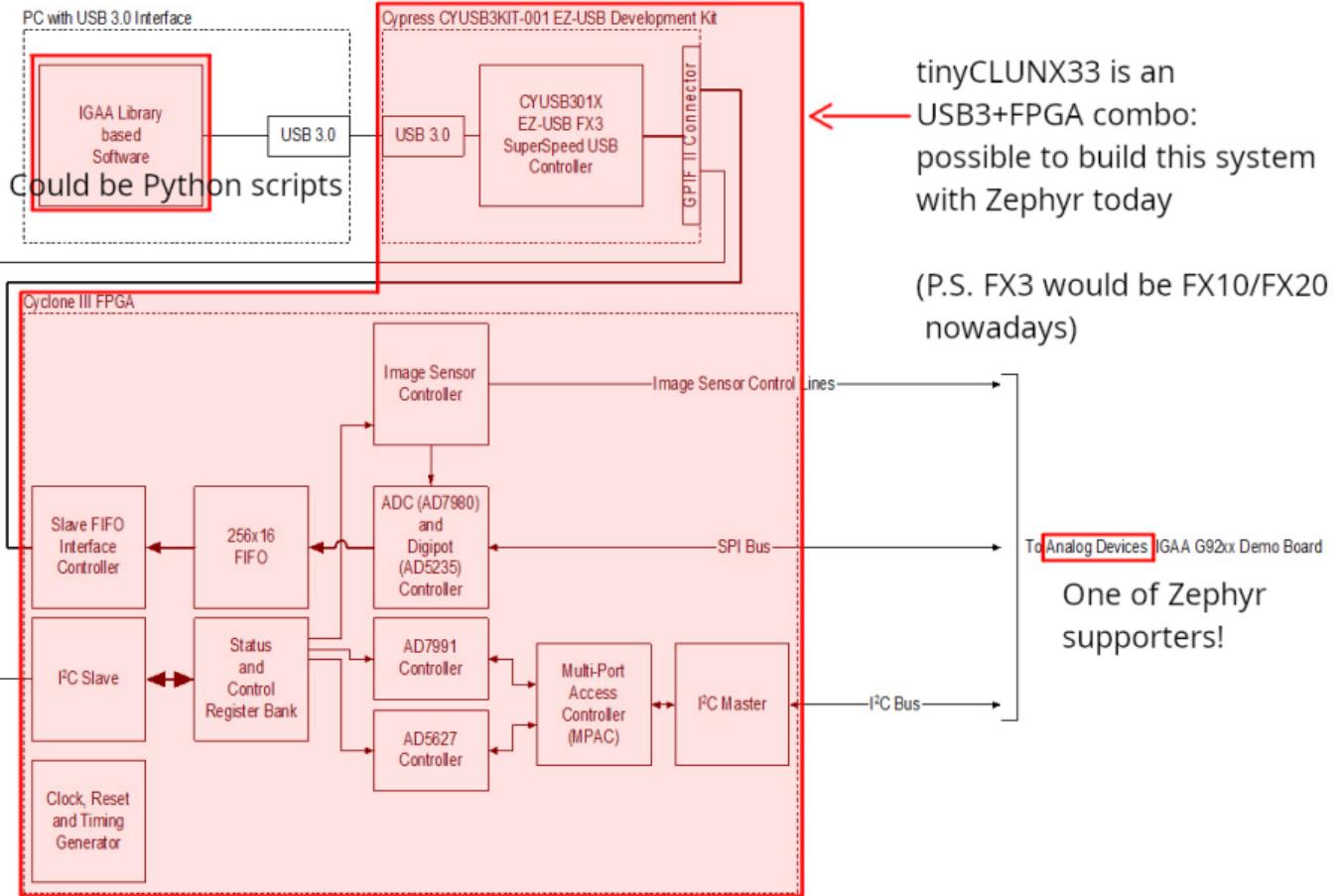
Sensing one pixel at a time, need to go fast!



Requires a fast ADC to handle the input.

Hamamatsu recommends an FPGA (customizable chip) + Analog Devices Analog-Digital-Converter (ADC).







# Multiple lines

Tools that can be used for building video systems: hardware to access the sensors implement all of that chain

- Difficulty of embedded video: accessing parallel port or MIPI
- Can use adapter chips like Himax HX6538 (not yet supported) or small FPGAs

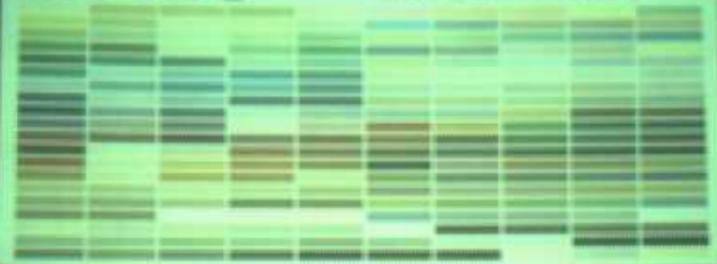
# What comes out of an image sensor

Dark (no auto-exposure) Green (no color correction)

Steps of an ISP.

## タント 200 色

(色見本と現物は異なります)



# TANT 200 COLOR PAPERS

※パッケージの写真と実物は多少異なる場合があります。  
※表示価格は消費税の適用価格です。

### 注意(ちゅうい)

保護者の方へ 念ずお読みください。

- カラーペーパーや、紙を口の中に入れたら頭を痛つたり熱気にしないでください。窒息の危険があります。
- 折り上がった作品で危険な遊びを絶対にしないでください。
- 強烈・鋭敏の色展がありますので、3才未満のお子様には絶対に与えないでください。

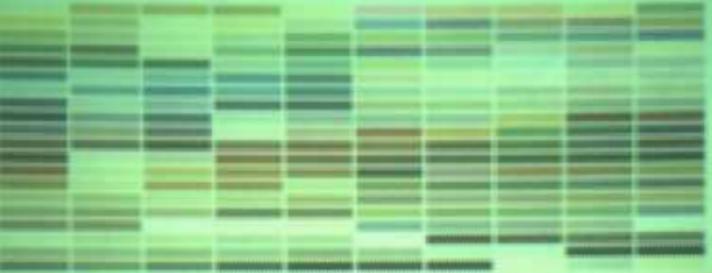
TEL 電話番号  
FAX FAX番号  
郵便番号  
郵便局名  
http://www.kintetsu-pap.jp MADE IN JAPAN



4 902637 317699

## タント 200 色

(色見本と現物は異なります)



# TANT 200 COLOR PAPERS

当パッケージの写真と製品は多少異なる場合があります。  
お手り図は実物の裏面にあります。

### 注意(ちゅうい)

保護者の方へ 必ずお読みください。

- カラーペーパーや、紙を口の中に入れたり紙を噛んだり絶対にしないでください。窒息の危険があります。
- 折り上げた商品で過度な遊びを絶対にしないでください。
- 胸元・腹巣の危険がありますので、3才未満のお子様には絶対に与えないでください。

お問い合わせ窓口

0120-0044 東洋印刷株式会社総務部企画課

電話番号 03-3668-2821

FAX番号 03-3668-2822

<http://www.tantcolor.com> WEBSITE IN JAPAN

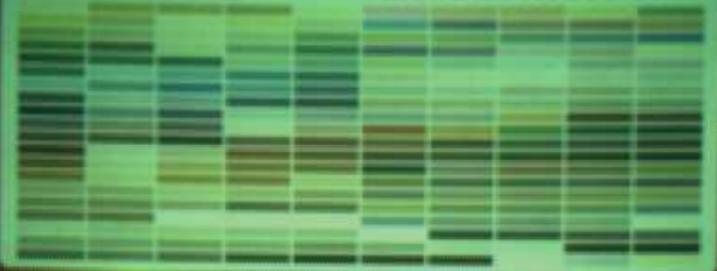
ホルコード 0588007-1200



4 902631 317699

タント 200 色

〔色見本と現物は異なります〕



TANT 200 COLOR  
PAPERS

・パッケージの写真と製品は多少異なる場合があります。  
・ヨリヤクは各色の裏面にあります。

注意(ちゅうい)

右側面の方へ、必ずお読みください。

- カラーペーパー、封筒の中に入れたり顔を覆ったり絶対にしないでください。發熱の危険があります。
- 手に上がった状態で顔を遊びを絶対にしないでください。
- 蒸熱-発熱の危険がありますので、3才未満のお子様には絶対に見えないでください。

会員登録/新規会員登録  
TEL:03-5544-0700 FAX:03-5542-1242

お問い合わせ用FAX:03-5542-7821

http://www.kodak-tant.com 無料お問い合わせ

品番コード:098007-1200



タント  
200  
色

タント 200 紙

(色見率と現物は異なります)



TANT 200 COLOR PAPERS

タッパーの可憐と製造は多少異なる場合があります。  
お取り扱い書類の裏面にあります。

手・足裏(ちゆう)

保護着の内、必ずお読みください。

- カラーペーパーを、目を口に入れたり顎を落とした  
り頭部にしないでください。腫瘻の危険があります。
- 頭上部に舌面や舌側を避けを絶対にしないでく  
ださい。
- 頭部・脳部の危険がありますので、3才未満のお子様  
に絶対に与えないでください。



品番: TANT-200  
130-0004  
TEL: 03-3888-7222

http://www.tant.jp/tant-200.html  
E-mail: tanta@007-1200.com



TANT



# TANT 200 COLOR PAPERS

牛糞的進一步的增強與發展將多少與各個省的農村生活、土地的開發和經濟的發展息息相關。

第1章

#### **REFERENCES**

- カーブーバーの箇所のところにヒゲを剃った、  
うなじがないでください。裏面の肌荒れがあります。
  - 剃り上げた両脇や腋毛を剃るためにしないでく  
ださい。
  - 脚部、足首の肌荒れがありますので、上下両面の内子孫  
のうなじに見えないようにください。

[View Details](#)

• 100 •

卷之三



167

Why an ISP is useful for robotics?

- Get always values within same range
- Poor exposure: no data at all
- De-fisheye
- Avoid artefacts to trigger a detection on the NPU or other vision algorithm

Conclusion: A lot to handle to get a reasonable image out of a sensor!

Hardware that can help accessing this image.

# What it takes...

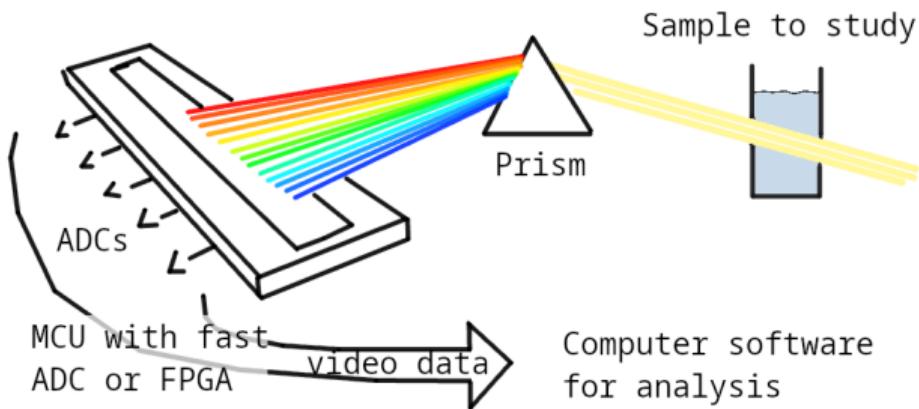
What would it take to build various devices on Zephyr

!! disclaimer: hardware is hard !!

!! disclaimer: not everything shown has drivers !!

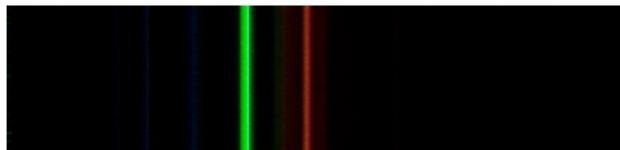
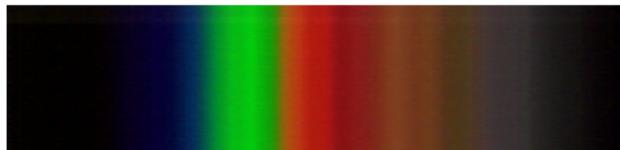
Also works without Zephyr, just putting things in perspective.

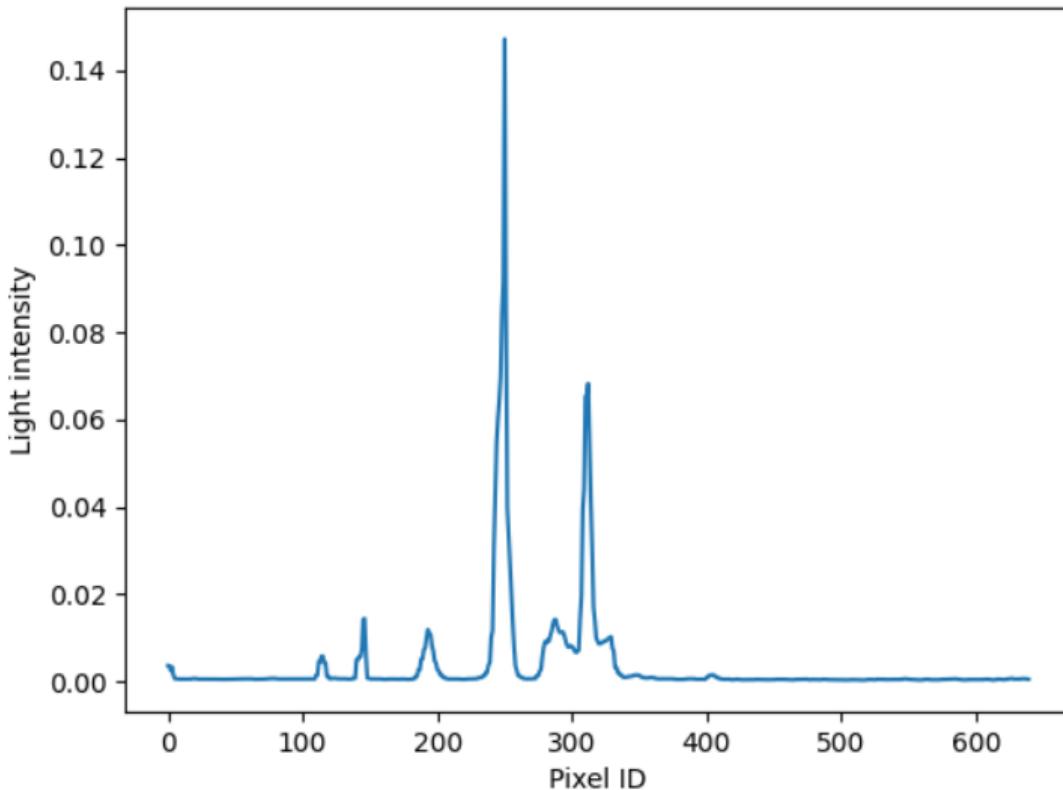
# What it takes... Spectrophotometer



Need a very fast ADC! Not many board will have one...

→ Good to have a lot of options.







# What it takes... Drones

## VMU RT1170

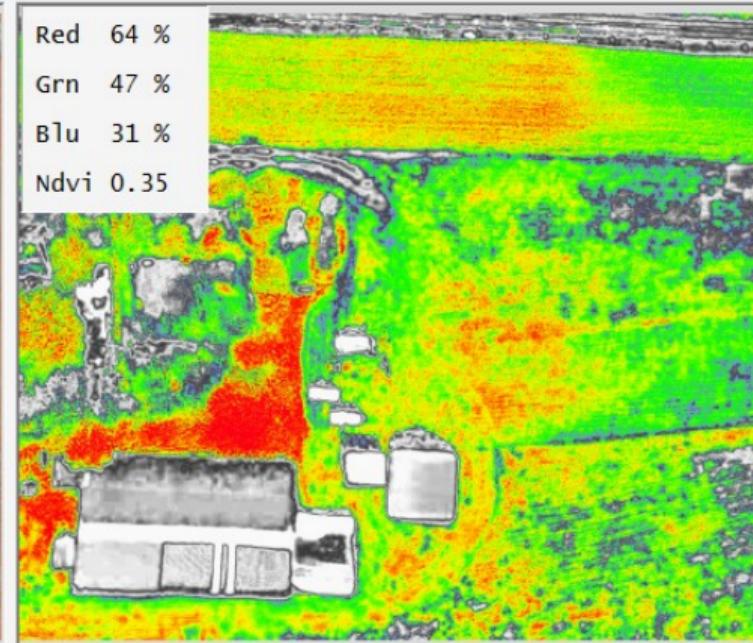
### Overview

The VMU RT1170 features an i.MX RT1176 dual core MCU with the Cortex-M7 core at 1 GHz and a Cortex-M4 at 400 MHz. The i.MX RT1176 MCU offers support over a wide temperature range and is qualified for consumer, industrial and automotive markets. The VMU RT1170 is the default VMU for CogniPilot's Cerebri, a Zephyr RTOS based Autopilot.

### Hardware

- MIMXRT1176DVMAA MCU
  - 1GHz Cortex-M7 & 400Mhz Cortex-M4
  - 2MB SRAM with 512KB of TCM for Cortex-M7 and 256KB of TCM for Cortex-M4
- Memory
  - 512 Mbit Octal Flash
  - TF socket for SD card
- Ethernet
  - 2 wire 100BASE-T1
- USB
  - USB 2.0 connector
- Power
  - Redundant dual picoflex power ports
- Debug
  - 10 pin debug and shell adapter board to 20 Pin JTAG debugger and USB-C shell
- Sensors





-0,07

0,50

50  
Max

Load image

Load false colors

NDVI  
multiple

NDVI

Histogram

-7  
Min

Double click on images to open them with the default application



# What it takes... Yeast monitoring station

Monitoring process of beer, kombucha, lactic fermentation

Video but also...

CO2 polling for building charts.

LED API for illuminating when taking a capture.

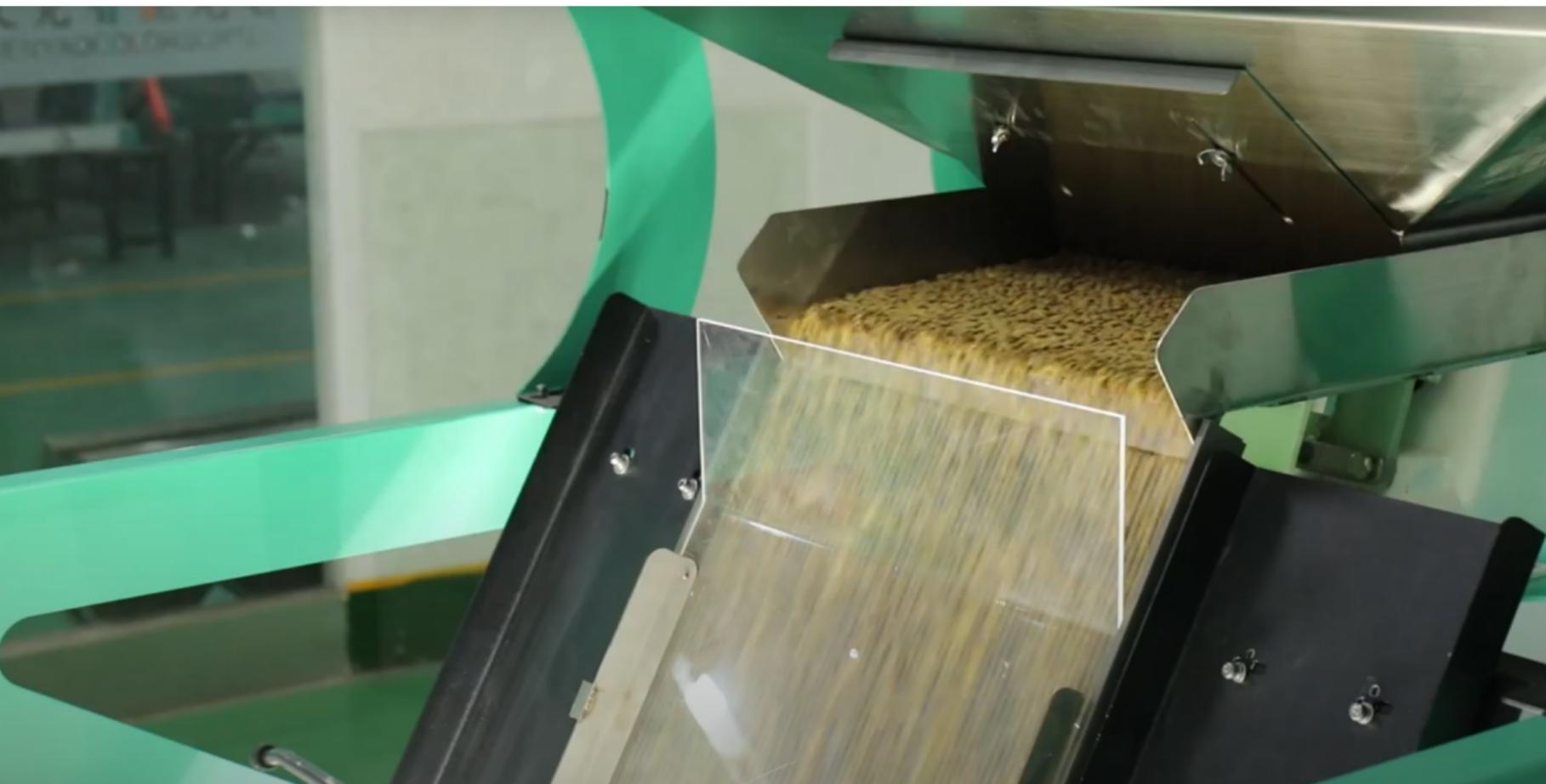
Video controls API for manual exposure tuning during "flash".

Wi-Fi to the home router.

HTTP client for sending the results.

# What it takes... Sorting machine

Fastest board you can get! Most real-time you can get!



Elimination of the rotten and unripe berries and the stems via pneumatic injectors.



Blowing air on everything not looking like a fresh grain of raisin.

16x slow motion





# What it takes... Endoscopes/Borescope

Cameras used by surgeons

Example of real endoscope camera module (CAMEMAKE):



Image sensor driver

Video transfer driver <-> Video APIs <-> UVC

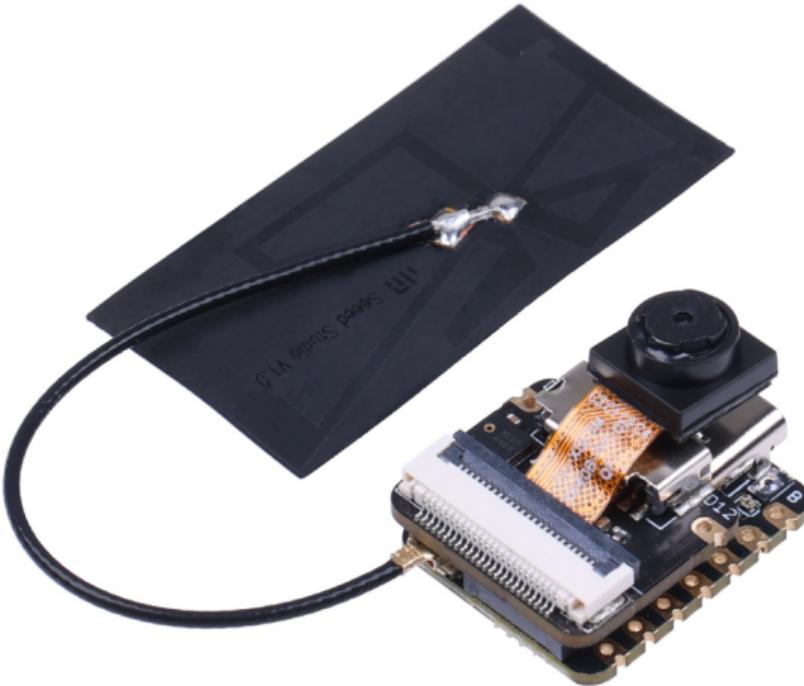
Parallel  
port



Image  
sensor

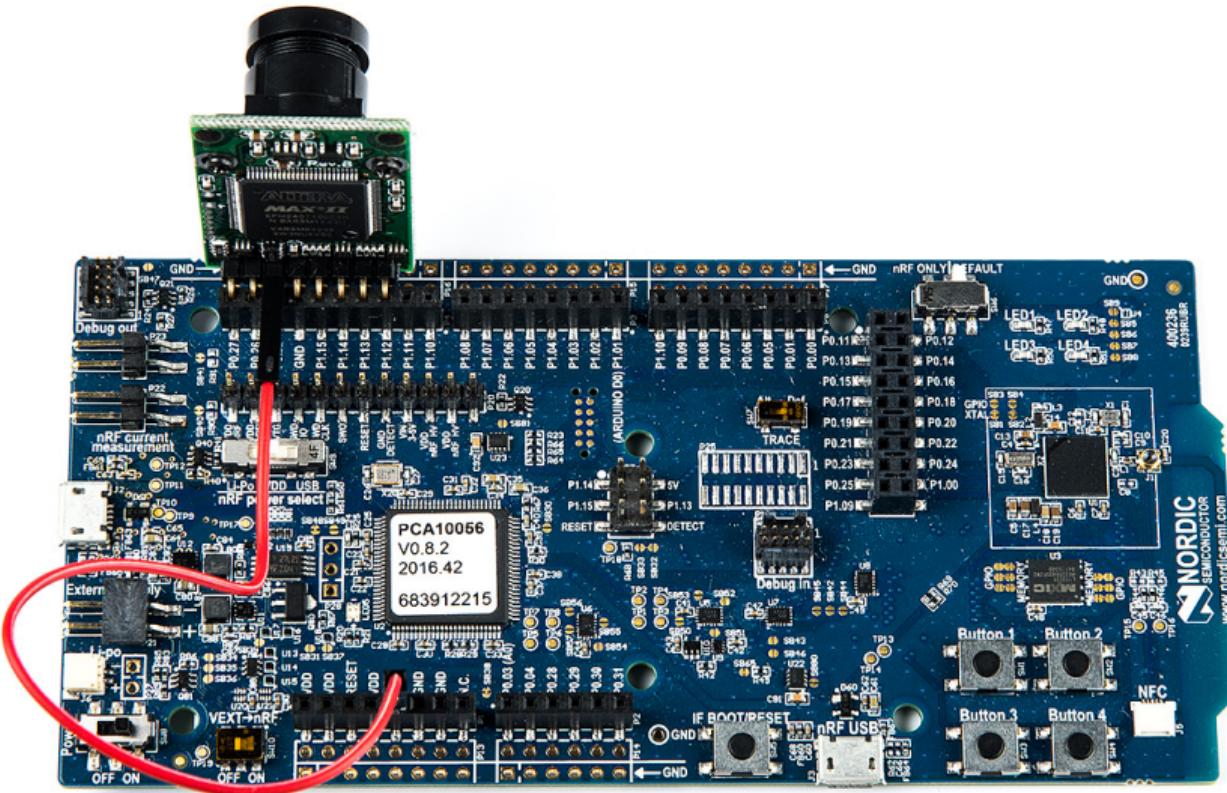


# What it takes... Wi-Fi Smartglasses



# What it takes... Bluetooth Smartglasses

Pre-Zephyr Nordic era: needs conversion.



Any MCU with  
Bluetooth

↔ SPI ↔

Himax HM0360  
(not yet supported)



640x480



# Beyond Zephyr: ecosystem around it

What UVC adds to the table:

- USB camera protocol supported on Linux, Windows, MaxOS, Android, iPad (not iOS yet), BSDs, 9front, QNX... (thanks to laptop lid cameras)
- Linux interoperability: Standardize all the video controls with Linux
- ROS2: integration of robotics (via USB cameras)
- OpenCV (via USB cameras)
- Want to support a new sensor on any ecosystem? Bring Zephyr support, and now it's everywhere