# STRENGTHENING ZEPHYR'S CAMERA FRAMEWORK

By Ankit S. & Elgin P.
Silicon Signals Pvt. Ltd.

# Speakers

Elgin Perumbilly

Associate Engineer

Ankit Siddhapura

Technical Lead

# Agenda

01 An overview of the Zephyr video subsystem

02 Runtime PM support in camera driver
Design approach & Review feedback and future direction

03 Introduce Flags in Video Framework
Proposed solution, Benefits for applications

04 Q&A

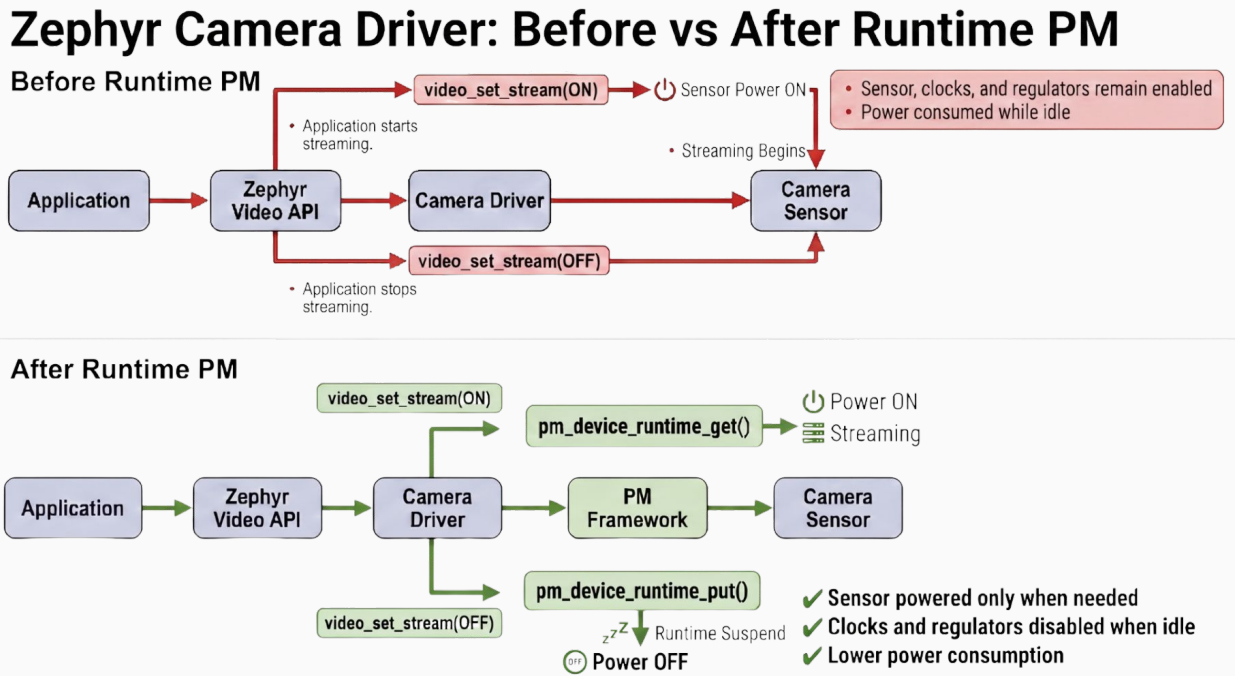
# Introduction

- Zephyr's video subsystem provides a unified camera framework, but some gaps remain for production-ready embedded vision systems.
- While working on the STM32N6 with an IMX335 sensor, we identified two areas for improvement.
- This talk covers our Runtime PM contribution for the IMX335 driver, along with an ongoing discussion around per-buffer status flags in the video framework.
- We'll share the implementation, maintainer feedback, and the design decisions that shaped these efforts.

# Zephyr’s Power Management

- Power management reduces energy consumption by turning off clocks, gating power rails, and entering low-power states when hardware is idle.
- The challenge is performing these transitions automatically and safely without disrupting software operation.
- In Zephyr, failure paths may leave resources such as clocks and power domains enabled, unlike Linux where these cases are generally handled gracefully.
- Even small runtime PM improvements can significantly improve battery life and power efficiency, especially in resource-constrained devices.

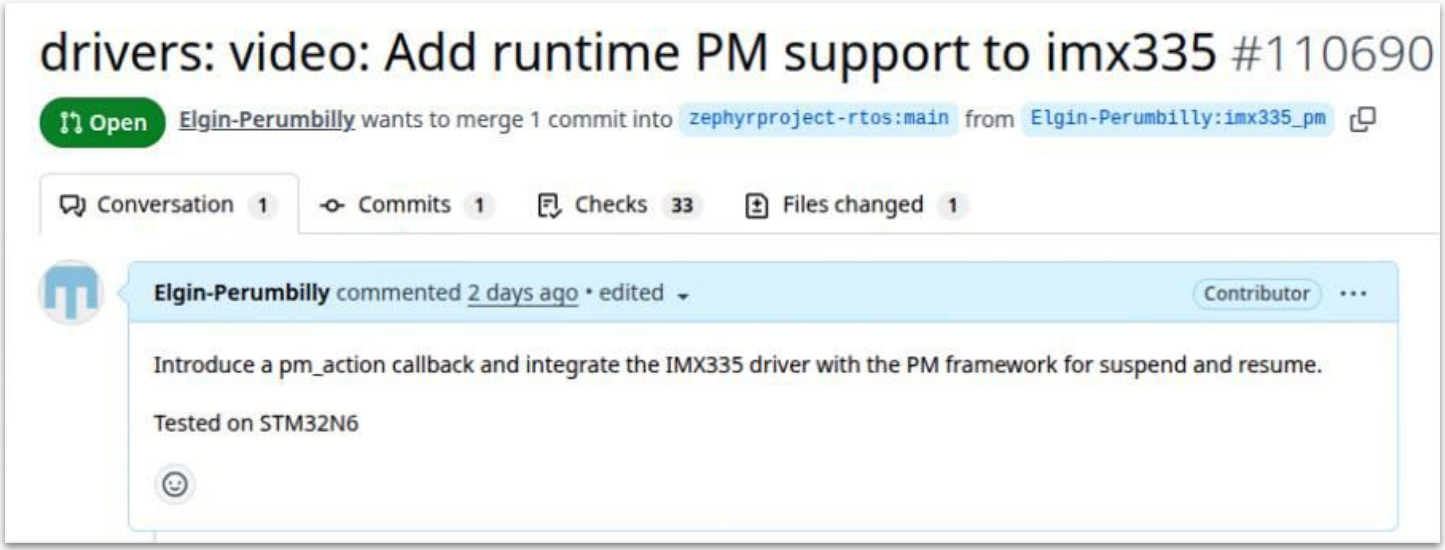
# Zephyr Camera Pipeline With PM



# Why Runtime PM Matters

- Without Runtime PM, camera sensors and related hardware resources may remain powered even when not actively capturing frames.
- Runtime PM automatically suspends devices and resumes them when they are needed again.
- This reduces power consumption, extends battery life, and improves thermal efficiency.
- It provides transparent power management and scales well for systems with multiple power-managed peripherals.

# Runtime PM Integration



# zephyr/drivers/video/imx335.c

```
video_stream_start(){
        pm_device_runtime_get();
}

....

video_stream_stop(){
        pm_device_runtime_put();
}
```

```
static int imx335_pm_action(const struct device *dev,
                            enum pm_device_action action)
{
        switch (action) {
        case PM_DEVICE_ACTION_RESUME:
        ...
        break;

        case PM_DEVICE_ACTION_SUSPEND:
        ...
        break;
}
```

# zephyr/drivers/video/imx335.c

→ Register your device with the PM framework.

```
imx335_init(){
        ...
        pm_device_runtime_enable()
}
```

```
PM_DEVICE_DT_INST_DEFINE(n, imx335_pm_action);

DEVICE_DT_INST_DEFINE(
        n,
        &imx335_init,
        PM_DEVICE_DT_INST_GET(n)
        ....
);
```

# zephyr/samples/drivers/video/capture/prj.conf

→ Enable this options in your projects prj.conf

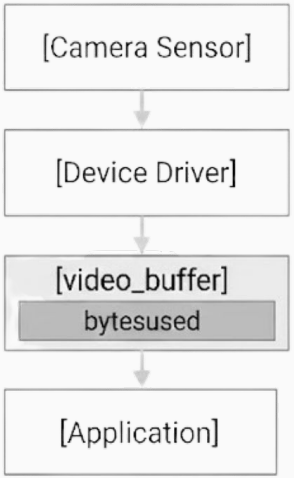
**CONFIG_PM_DEVICE=y**
**CONFIG_PM_DEVICE_RUNTIME=y**

→ Runtime PM behaviour can be verified using Zephyr Shell or need to test stream functionality just add -DCONFIG_VIDEO_SHELL=y with your build command

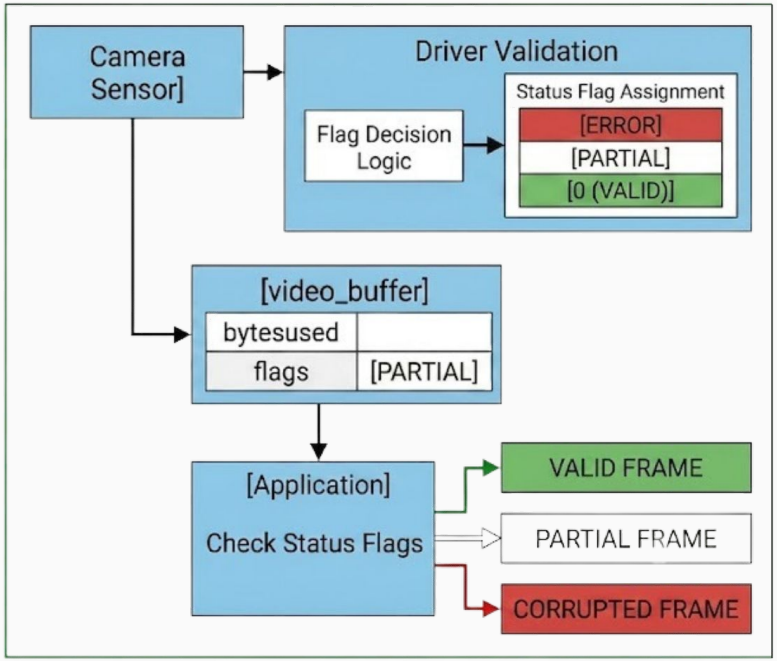
```
uart:~$ device pm_toggle camera@1a
pm_device_runtime_get(camera@1a)
```

```
uart:~$ device pm_toggle camera@1a
pm_device_runtime_put(camera@1a)
```

# Add Status Flags in Video Framework




**Note: Only 'bytesused' available.**
**No frame health information.**
**Application cannot distinguish valid, partial, or corrupted frames.**



# Continued...

```
#define VIDEO_BUF_FLAG_ERROR     BIT(6)
#define VIDEO_BUF_FLAG_PARTIAL   BIT(7)
```

```
struct video_buffer {
        ...
        uint32_t flags;
};
```

```
/* Clear stale flags */
buf->flags = 0;

/* Counter read failed */
buf->flags |= VIDEO_BUF_FLAG_ERROR;

/* Frame shorter than expected */
buf->flags |= VIDEO_BUF_FLAG_PARTIAL;
```

# Credits

josuah - josuah.demangeon

# Q&A

Any questions?
Ask away!

**Developer Survey 2026**

**10 Mins** to help shape the next
**10 Years** of Zephyr RTOS

Closes June 30

zephyrproject.org/devsurvey2026

Give us a chance to address your doubts or concerns. Let's open up the discussion with your ideas!

# Thank you!

Got more questions?
Please reach out to
info@siliconsignals.io