



GStreamer conference 2026

Making GStreamer Feel Native in Zig

I Just Wanted to Build a Transcoder

- GStreamer was the obvious choice
- Zig seemed ideal for the surrounding app
- Calling the C API was easy
- Writing an idiomatic Zig app wasn't
- So I started writing bindings

Why Zig?

Simple and explicit

No GC, predictable resource management

Translates C headers directly, `gst/gst.h` just works

Powerful compile-time programming



"Easy": the C API, called from Zig

Easy

- No glue code, translateC turns `gst/gst.h` into Zig
- Zero overhead

But it isn't idiomatic Zig

```
const src = c.gst_element_factory_make("videotestsrc",);  
if (src == null) return error.ElementCreationFailed;  
  
c.g_object_set(@ptrCast(src),  
    "pattern",      @as(c_int, 18),  
    "is-live",      @as(c.gboolean, 1),  
    "num-buffers",  @as(c_int, 100),  
    @as(*anyopaque, null));
```

With gstreamer-zig

What changed?

- Errors, not NULL
- Struct literal, not varargs

```
const src = try gst.Element.init("videotestsrc",...);

src.set({
    .pattern = 18,
    .is_live = true,
    .num_buffers = 100
});
```

A simple program

```
pub fn main() !void {
    gst.init(null);
    defer gst.deinit();

    const pipeline = try gst.Pipeline.initLaunch(
        "videotestsrc ! autovideosink"
    );
    defer pipeline.deinit();

    try pipeline.start();
    defer _ = pipeline.setState(.null_state);

    const bus = try pipeline.getBus();
    defer bus.deinit();

    while (bus.timedPop(gst.clock.TIME_NONE)) |msg| {
        defer msg.deinit();
        switch (msg.getType()) {
            .eos => break,
            .err => {
                msg.parseErrorAndPrint() catch {};
                break;
            },
            else => {},
        }
    }
}
```

So, what's the idea?

Preserve GStreamer

- Concepts
- Flexibility

Embrace Zig

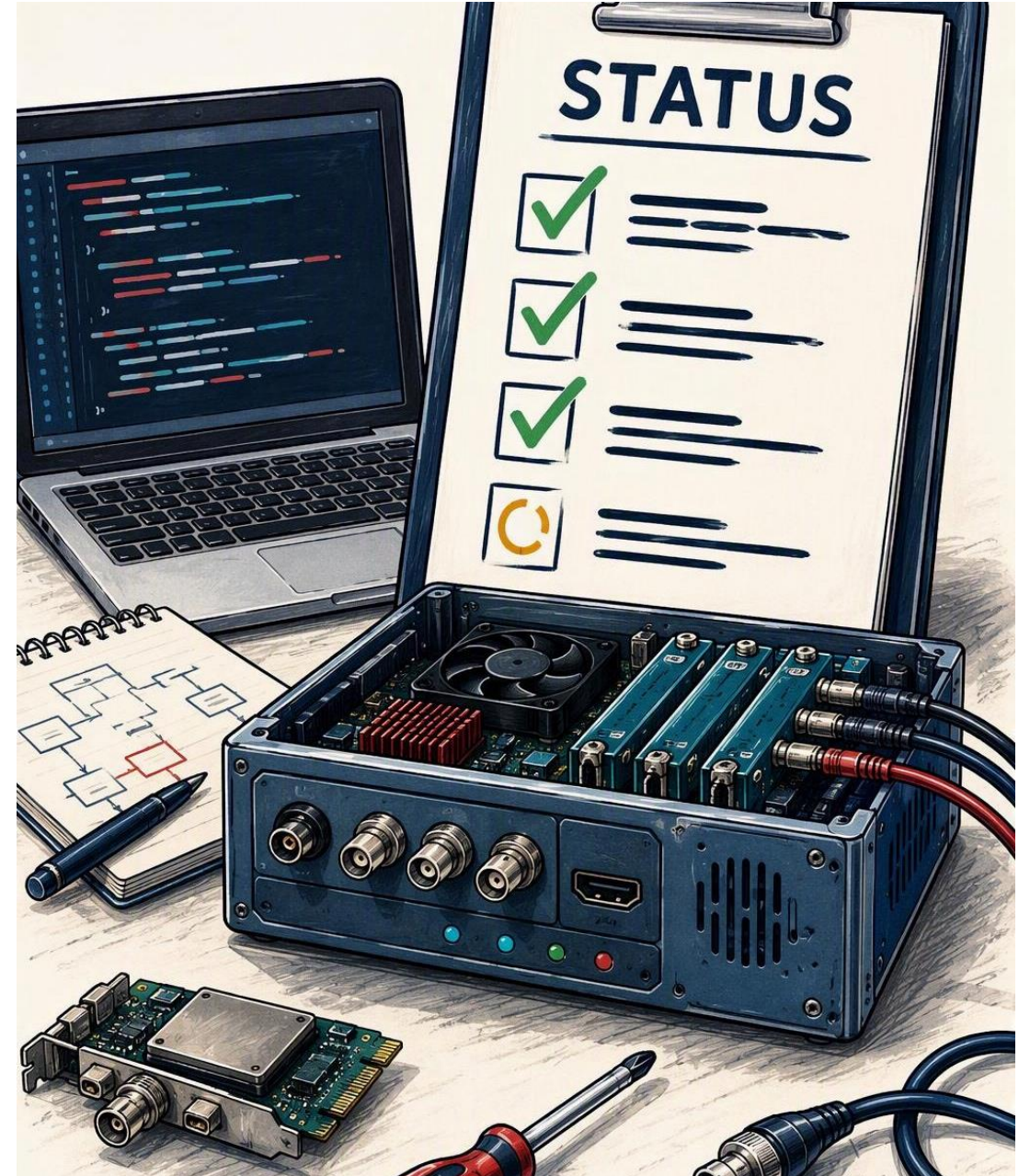
- Style
- Features

The challenges?

- Floating refs, transfer rules and double frees
- Zig is pre 1.0

Status

- Core, App (appsrc, appsink), video, pbutils, GLib MainLoop
- Examples mostly written from the Rust examples
- Only tested on macOS (using `gst_macos_main`). Not tested on Linux and Windows
- A lot hasn't been wrapped yet



Thanks



<https://codeberg.org/at39p/gstreamer-zig>

Per Andersen
pele@tv2.dk