AKARSHAN KAPOOR

Embarking on Embedded Adventures

A Beginner's Guide to Open Source and Zephyr

$ whoami

- **Recent graduate** from **IIT Mandi, India**.
- Was in **Germany** for the past **7–8 months** as an **exchange student** at **TU Braunschweig** and **TU Dresden**
- **Two-time Google Summer of Code contributor** at **OpenPrinting** under the **Linux Foundation**.
- **Speaker at Ubuntu Summit 2023 and 2024**.
- **Currently interning at Phytec**.

What is Zephyr?

An RTOS for IoT

- multiple supported architectures (ARM, RISC-V, x86...)
- Multi-threading
- Power Management

and much more!

- Open Source Bluetooth Low Energy Stack
- Networking, USB, Filesystems, Cryptography
- Shell, Logging, Sensors, Display, Audio

Ideal to build IoT products

- Well supported for a wide range of hardware
- Vendor neutral steering by Linux Foundation

Zephyr®

Basically...

- A RTOS ecosystem, by developers, for developers
- Small Footprint Kernel for resource constrained and embedded systems
- Supports a variety of different architectures like Intel x86, ARM v6/7, MIPS, RISC-V etc.

OS Landscape

General Purpose OS (Desktop/Server)

- **Ubuntu, Arch, Windows, macOS**
- **Target**: Desktops, laptops, servers
- **Resources**: GBs of RAM, powerful CPUs
- **Goals**: User experience, multitasking, rich applications
- **Boot time**: 30+ seconds acceptable
- **Real-time**: Not guaranteed (best effort)

Embedded Linux

Raspberry Pi OS, Yocto, Buildroot

- **Target**: Single-board computers, IoT gateways
- **Resources**: 100MB+ RAM, ARM Cortex-A processors, PHYTEC Reel Boards
- **Goals**: Linux compatibility, networking, file systems
- **Boot time**: 10-30 seconds
- **Real-time**: Soft real-time possible with patches

Real-Time Operating Systems (RTOS)

FrexeRTOS, **Zephyr**, ThreadX, QNX

- **Target**: Microcontrollers, embedded devices
- **Resources**: 8KB-8MB RAM, ARM Cortex-M, RISC-V
- **Goals**: Deterministic response, low power, real-time guarantees
- **Boot time**: Milliseconds to seconds
- **Real-time**: Hard real-time capabilities

Core Architecture Overview

- **Scalable RTOS**: 8KB to multi-core systems
- **Event-driven**: Responds to interrupts, timers, messages
- **Preemptive multitasking**: Higher priority tasks run first
- **Memory protection**: Optional MMU/MPU support
- **Device abstraction**: Unified API across different hardware

Key Point: Zephyr = Small footprint + Big system features

Monolithic vs Microkernel

Traditional Monolithic Kernel

- **All services in kernel space**: Drivers, filesystem, networking
- **Examples**: Linux, Windows

- **Pros**: Fast communication, simple design
- **Cons**: One crash kills everything

Zephyr's Microkernel Approach

- **Minimal kernel**: Only scheduling, IPC, memory management
- **Services as threads**: Drivers, protocols run in user space
- **Isolation**: Fault in one service doesn't crash system
- **Modularity**: Include only what you need

Bottom Line: Microkernel = More reliable, configurable embedded systems

---

System Design Basics

Scheduling

- **Preemptive**: High priority interrupts low priority
- **Time slicing**: Round-robin for equal priorities
- **Cooperative**: Tasks yield voluntarily
- **Real-time**: Deterministic response times

Threading

- **Thread states**: Ready, Running, Suspended, Terminated
- **Priority levels**: 0 (highest) to 31 (lowest)
- **Stack management**: Each thread gets own stack
- **Context switching**: Hardware-assisted when possible

Memory Management

- **Static allocation**: Compile-time memory assignment
- **Memory pools**: Dynamic allocation from predefined blocks
- **Memory protection**: Optional userspace isolation

- **Low footprint**: Minimal RAM overhead

Key Insight: Everything designed for predictable, resource-constrained systems

---

What is Device Tree?

- **Hardware description language**: Describes board layout in code
- **Platform independence**: Same driver works on different boards
- **Compile-time configuration**: No runtime hardware discovery
- **Hierarchical structure**: Represents actual hardware connections

---

Device Tree Example

```
{
    leds {
        compatible = "gpio-leds";
        led0: led_0 {
            gpios = <&gpio0 13 GPIO_ACTIVE_LOW>;
            label = "Green LED";
        };
    };
};
```

Why Device Tree Matters

- **No hardcoded pins**: Change hardware without code changes
- **Driver reuse**: Same LED driver works on any board
- **Build system integration**: Automatic configuration generation

Takeaway: Device Tree = Hardware abstraction for embedded systems

---

Z-Bus - Zephyr's Messaging System

What is Z-Bus?

- **Publish-Subscribe messaging**: Decoupled communication
- **Event-driven architecture**: Components react to events
- **Type-safe**: Compile-time message format checking
- **Efficient**: Direct function calls, no serialization overhead

---

- **Publishers**: Send messages (sensors, timers, user input)
- **Subscribers**: Receive messages (actuators, displays, loggers)
- **Channels**: Named message pathways
- **Message types**: Structured data definitions

West - Zephyr's Swiss Army Knife

What is West?

- **Meta-tool**: Manages Zephyr project and dependencies
- **Multi-repository**: Handles Zephyr + modules + your code
- **Build system**: Wraps CMake with embedded-specific features
- **Flash/Debug**: Unified interface for different programmers

---

West Configuration

- **west.yml**: Defines project structure and dependencies
- **Manifest repository**: Contains project configuration
- **Workspace**: Local development environment

Setting up the Development Environment

Types of Requirements

- Linux, macOS or Windows 10/11
- Installation requirements (CMake, Python3..)
- Python requirements (west, pyocd..)
- Zephyr SDK that provides Toolchains (gcc, gdb, newlib..)

Everything is described in the Zephyr Getting Started Guide.

Now.. let's have some fun with Zephyr!

Giveaway Question 1

In Zephyr Threads which is the correct order :

a) 5>4>3>2>1>0

b) 3>4>5>2>1>0

c) 0>1>2>3>4>5

d) 2>4>3>1>0>5

Giveaway Question 2 & 3 (Combined)

1. Which foundation hosts the Zephyr project?
2. What is Zephyr's build system?