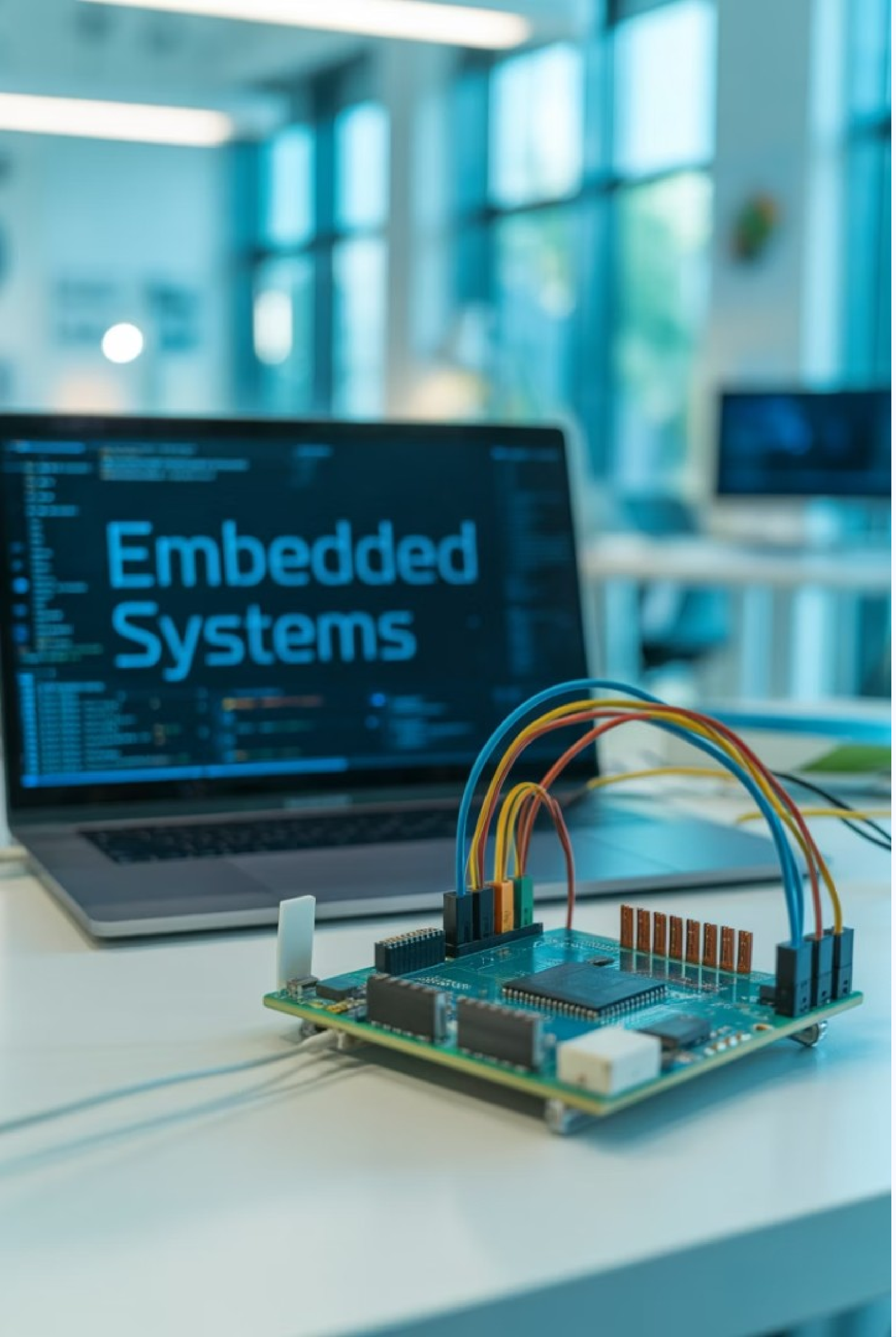


Zephyr RTOS Intro

Agenda

1 What is Zephyr RTOS?

Small, scalable open-source RTOS

2 Key Features

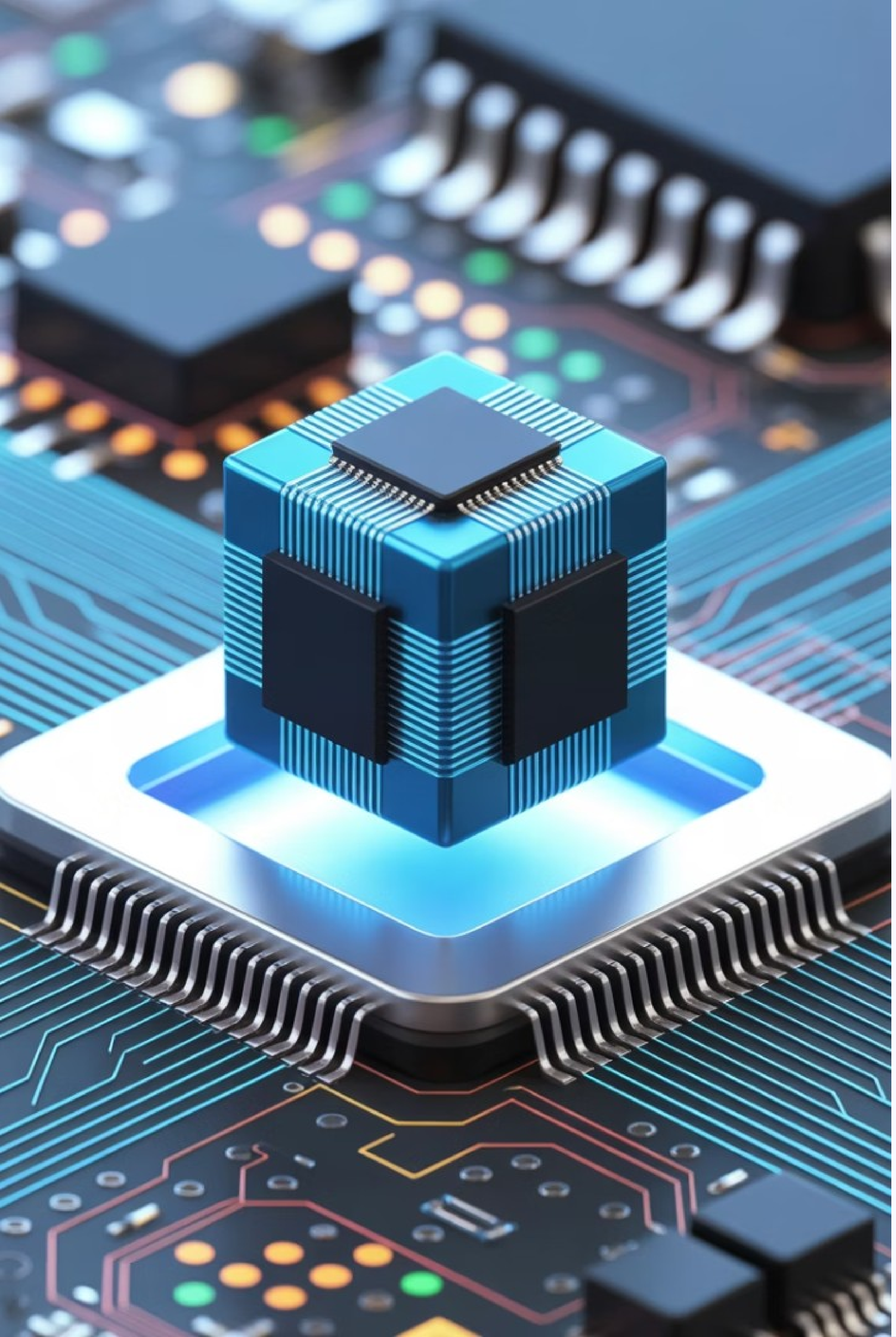
Architecture, boards, connectivity

3 Advantages

Compared to alternatives

4 Real-World Use

Products, trends, getting started



What is Zephyr RTOS?

Small & Scalable

Designed for resource-constrained embedded systems

Open Source

Vendor-neutral, community-driven

Wide Support

ARM, RISC-V, x86, Xtensa

Heritage

Originated from Wind River's Rocket OS (2015)

Key Features

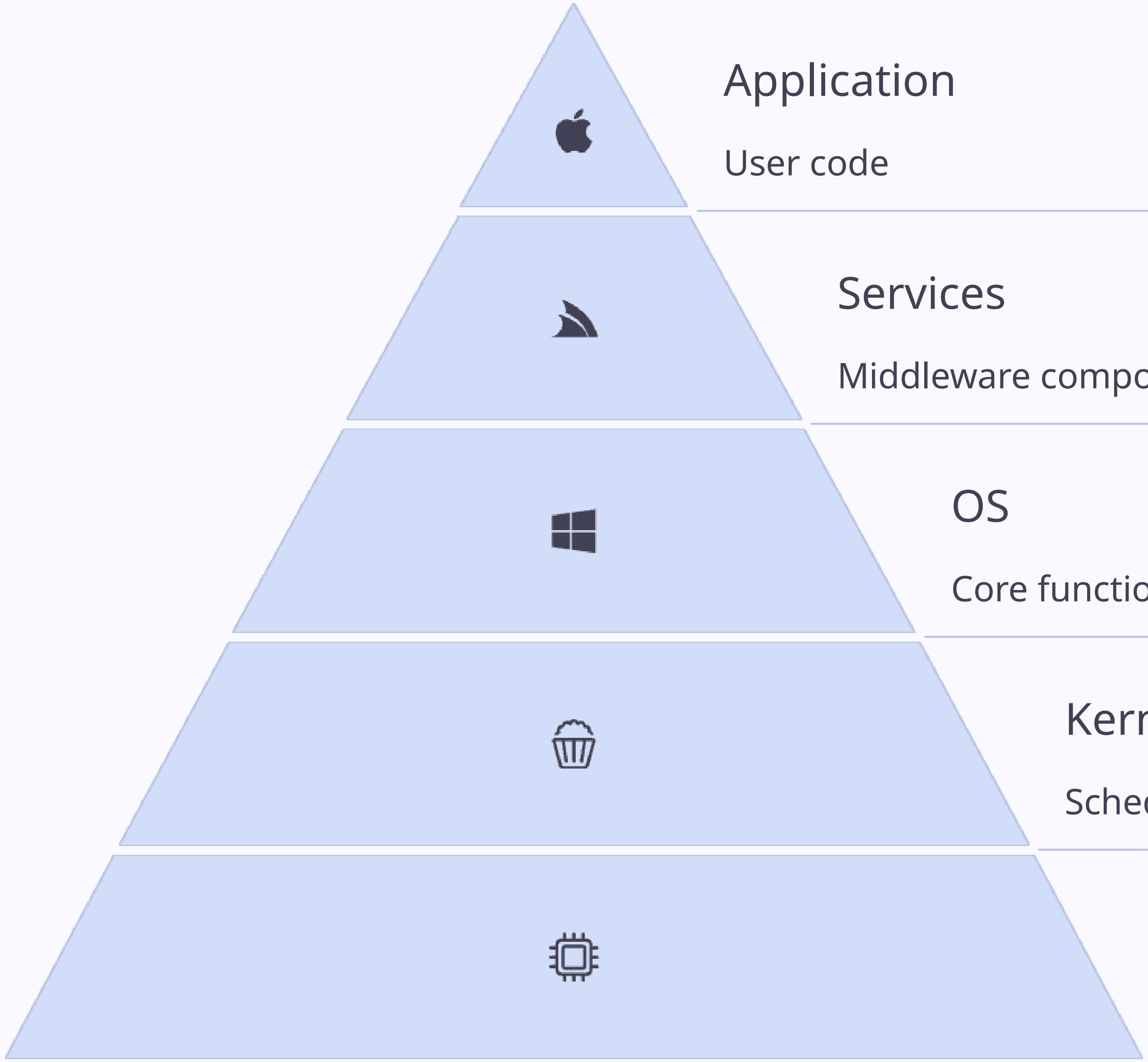
Minimal Footprint

- <8KB Flash, <5KB RAM
- Scalable to multi-core
- BLE 5.3, Wi-Fi, CAN, USB

Extensive Support

- 750+ boards, 220+ sensors
- CMake, Kconfig, Devicetree
- Full protocol support
- Power management

Architecture Overview



Application

User code

Services

Middleware components

OS

Core functionality

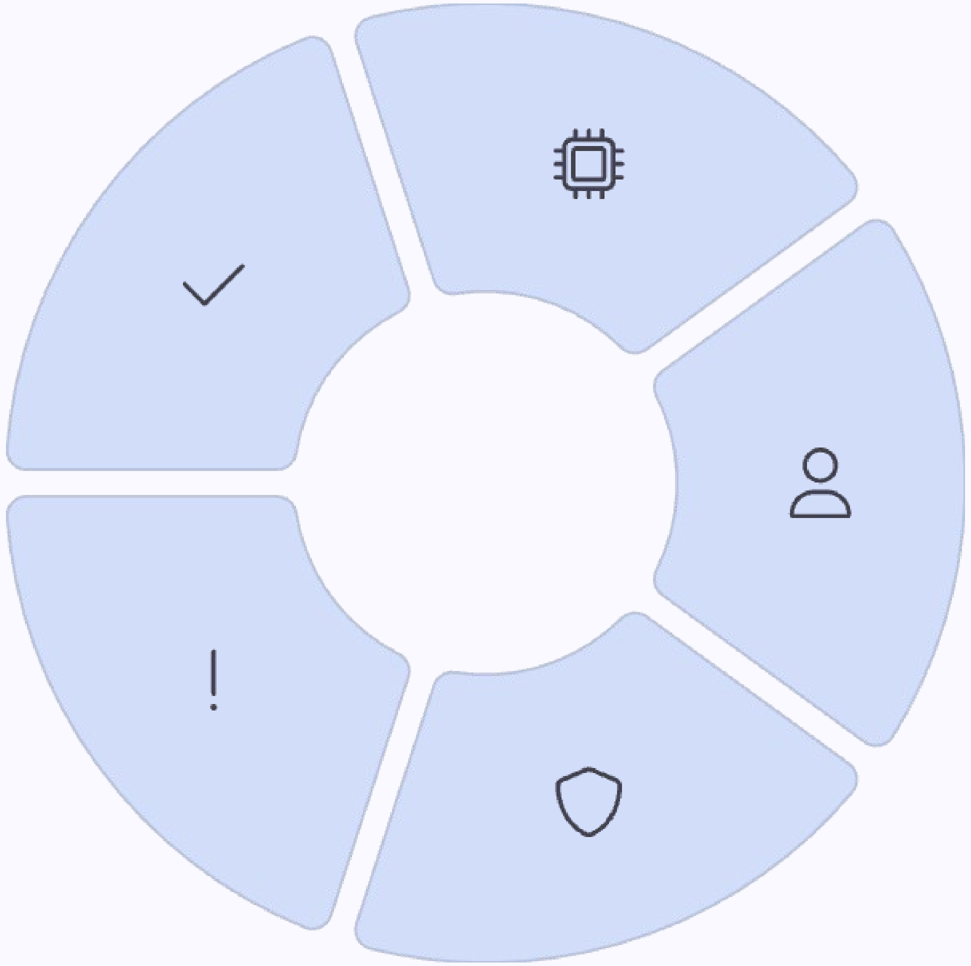
Kernel

Scheduling, memory

HAL

Hardware abstraction

Zephyr vs Other RTOSes



Open Source

Apache 2.0 vs commercial licensing

Challenges

Steeper learning curve, newer ecosystem

Hardware Support

750+ boards vs 40-100 for others

Community

1,100+ contributors, 5x growth since 2017

Security

Dedicated PSIRT and vulnerability management

Advantages over Alternatives

Feature	Zephyr RTOS	Linux	Bare Metal
Memory	KBs	MBs	Minimal
Boot Time	Milliseconds	Seconds	Immediate
Determinism	High	Low	Manual
Portability	High	Medium	None
Power	Efficient	Hungry	Variable

Real-World Products Using Zephyr

Consumer

Galaxy Ring, Moto Watch 100

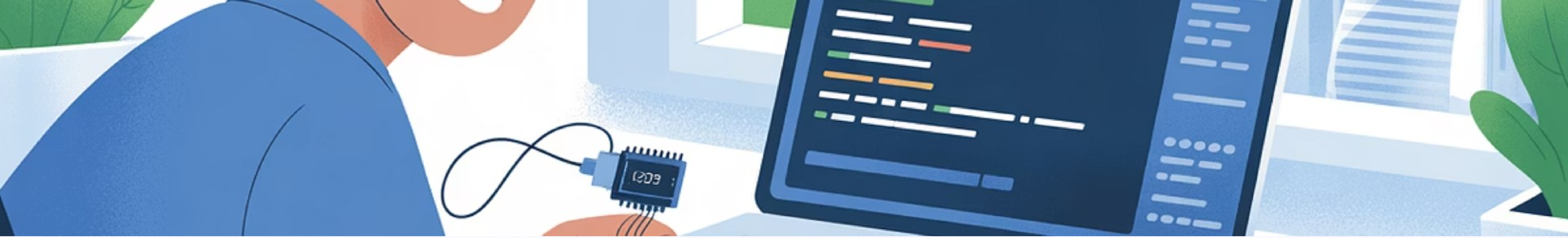
Computing

Chromebook EC, Framework Laptop

Industrial

Vestas, BLiXT, IoT devices

25+ Million Devices in the Field



Getting Started

Setup

Install deps, setup west, get SDK

Develop

Clone repo, choose sample, target board

Deploy

Build/flash/debug with west

Learn More

Docs, GitHub, Discord