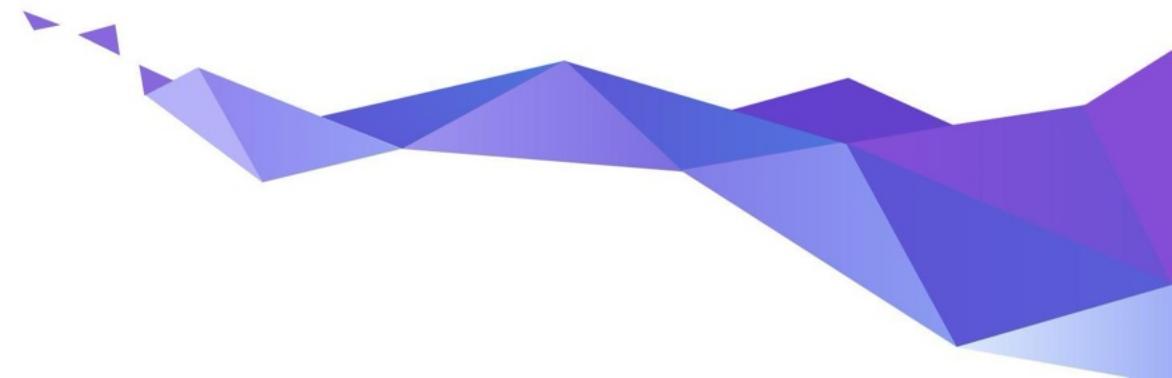


Zephyr Project Overview

A proven RTOS ecosystem, by developers, for developers



Use cases for a real-time OS



Industrial IoT



Asset Tracking



Wearables



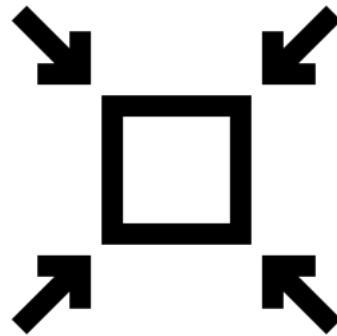
Automotive



Healthcare



Worker Safety

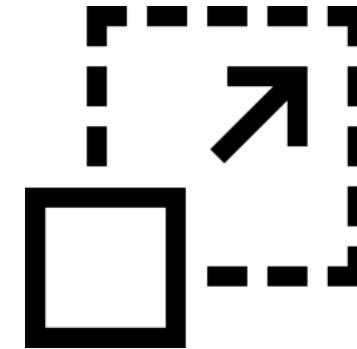


SMALL

< 8KB Flash

< 5KB RAM

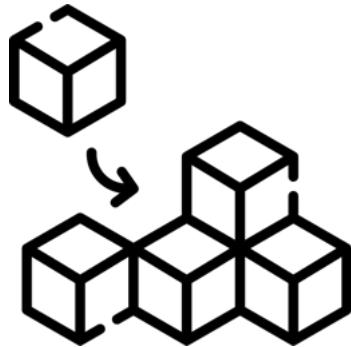
yet



SCALABLE

from small sensor nodes

... to complex multi-core systems



FLEXIBLE

yet

SECURE

Heavily customizable

Built with safety & security in mind

Out-of-the-box support for
600+ boards and 100s of sensors

Certification-ready
Long-term Support

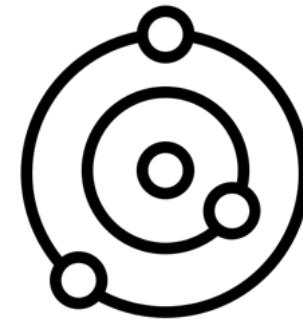




OPEN-SOURCE

Permissively licensed (Apache 2.0)

Vendor-neutral governance



ECOSYSTEM

Vibrant community

Supported by major silicon vendors

Features overview

- **Lightweight kernel & supporting drivers and services**
- **Portable, secure, power-efficient**
- **Highly connected**
 - Bluetooth 5.0 & BLE
 - Wi-Fi, Ethernet, CANbus, ...
 - IoT protocols: CoAP, LwM2M, MQTT, OpenThread, ...
 - USB & USB-C
- **Complete developer environment**
 - Toolchain and HAL management
 - Logging, tracing, debugging,
 - Emulation/Simulation
 - Testing framework





Products Running Zephyr Today



Oticon More
Hearing Aid



Lildog & Lilcat
Pet Tracker



Livestock Tracker



Moto Watch 100



Samsung Galaxy
Ring



Proglove



Adhoc Smart Waste



Google
Chromebook



Framework laptop



Keeb.io BDN9



Hati-ACE



Safety Pod



BLiXT solid state
circuit breaker



Aethero Deimos
Satellite



PHYTEC Distancer



Laird Connectivity
sensors & gateways



BeST pump
monitoring



Vestas Wind
Turbines



zephyrproject.org/products-running-zephyr



Products Running Zephyr Today



Discreet rechargeable hearing aid that gives you access to all relevant sounds

Oticon More supports the brain in making sense of sound and it is easy to operate with a double push button for volume and programme control. It features Bluetooth wireless technology for seamless connectivity with your favourite devices.

oticon
life-changing **technology**

Bluetooth LE

Low Power



zephyrproject.org/portfolio/oticon-more



Products Running Zephyr Today



Sustainable energy solutions

Vestas is the energy industry's global partner on sustainable energy solutions. We design, manufacture, install, and service onshore and offshore wind turbines across the globe, and with more than 164 GW of wind turbines in 87 countries, we have installed more wind power than anyone else. Through our industry-leading smart data capabilities and unparalleled more than 144 GW of wind turbines under service, we use data to interpret, forecast, and exploit wind resources and deliver best-in-class wind power solutions. Together with our customers, Vestas' more than 28,000 employees are bringing the world sustainable energy solutions to power a bright future.

Vestas®

CANbus

Industrial Control



zephyrproject.org/portfolio/vestas-wind-turbines

Products Running Zephyr Today



Thin, light, high-performance 13.5" notebook

A thin, light, high-performance 13.5" notebook that is also easy to repair, upgrade, and customize. The embedded controller firmware is a fork of the Zephyr version of chromium-ec, and is fully open source.

 framework

Embedded Controller

USB / USB-C

Power Mgmt



zephyrproject.org/portfolio/framework-laptop-13-diy-edition-amd-ryzen-7040-series



Products Running Zephyr Today



Professional grade, digital tape measure

The T1 Tomahawk, the world's first, professional grade, digital tape measure enables tradespeople, across industries, to collect measurements faster and more accurately than ever before. A live view, OLED display, shows measurements of the tape measure, digitally, in both english and metric units. With a click of a button, measurements are saved to a side mounted e-paper display as well as sent over Bluetooth to connected devices.

REEKÖN

Low Power

Sensing



zephyrproject.org/portfolio/reekon-t1-tomahawk



Products Running Zephyr Today



Turns your wired sensors into IP67-rated battery-operated wireless nodes, providing robust and secure messaging

Ezurio's Sentrius™ BT610 I/O Sensor with Bluetooth 5 turns your wired sensors into IP67-rated battery-operated wireless nodes, providing robust and secure messaging. Leveraging our BL654 module, it provides full Bluetooth 5 capabilities, opening up industrial and equipment monitoring applications.

ezurio

Bluetooth

Cellular

Connectivity Management

App Framework



zephyrproject.org/portfolio/sentrius

750+ supported boards... and growing



Arduino Portenta
H7



ESP32



Sipeed HiFive1



nRF9160 DK



STM32F746G Disco



M5StickC PLUS



TDK RoboKit 1



BBC micro:bit v2



Blue Wireless Swan



Arduino Nano 33
BLE



Intel UP Squared



Dragino LSN50
LoRa Sensor Node



Microchip SAM E54
Xplained Pro
Evaluation Kit



Raspberry Pi Pico



Altera MAX10



NXP i.MX8MP EVK



Adafruit Feather
M0 LoRa



u-blox EVK-NINA-B3



docs.zephyrproject.org/latest/boards

220+ Sensors Already Integrated

adt7420
 adxl345
 adxl362
 adxl372
 ak8975
 amg88xx
 ams_as5600
 ams_iAQcore
 apds9960
 bma280
 bmc150_magn
 bme280
 bme680
 bgm160
 bmi160
 bmi270
 bmm150
 bmp388
 bq274xx
 ccs811

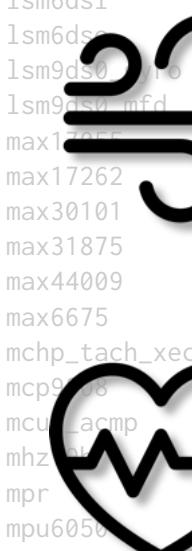
dht
 dps310
 ds1820
 ens1
 esp32⁶mp0
 fdo100
 fxo270
 fxo270
 grove
 grow_r502a
 hmc58831
 hp206c
 ht221
 i3g4250d
 i2l42605
 i2l42670
 i2l42685
 icp0125
 iis2dh
 iis2dlpc



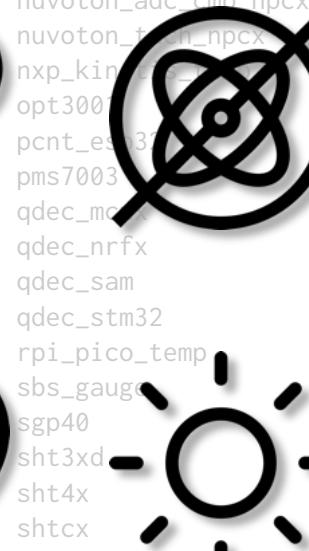
iis2iclx
 iis2mdc
 iis3dhhc
 ina219
 ina23
 isl2935
 ism330d1
 ite_tad_it8xxx2
 ite_vcmp_it8xxx2
 lis2dh
 lis2ds12
 lis2dw12
 lis2m
 lis3m
 lm75
 lm77
 lps22
 lps22hn
 lps25hb
 lsm303dlhc_magn



lsm6ds0
 lsm6ds1
 lsm6ds2
 lsm9ds0
 lsm9dsu_mfd
 max1705
 max17262
 max30101
 max31875
 max44009
 max6675
 mchp_tach_xec
 mcp958
 mcu_acmp
 mhz
 mpr
 mpu6050
 mpu9250
 ms5607
 ms5837



nrf5
 nuvoton_adc_cmp_npcx
 nuvoton_tcm_hpcx
 nxp_kin²
 opt300
 pcnt_es³3
 pms7003
 qdec_—
 qdec_nrfx
 qdec_sam
 qdec_stm32
 rpi_pico_temp
 sbs_gauge
 sgp40
 sh3t3xd
 sh4x
 shtcx
 si7006
 si7055
 si7060



si7210
 sm351lt
 stm32_temp
 stm32_vbat
 stmemsc
 stts751
 sx9500
 th02
 ti_hdc
 ti_hdc20xx
 tmp007
 tmp108
 tmp112
 tmp116
 vcnl4040
 vl53l0x
 wsen_hids
 wsen_itds



github.com/zephyrproject-rtos/zephyr/tree/main/drivers/sensor



Supported Hardware Architectures



Cortex-M, Cortex-R
& Cortex-A

x86 & x86_64



32 & 64 bit



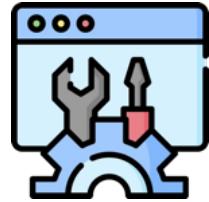
Xtensa



docs.zephyrproject.org/latest/hardware/index.html#hardware-support



Vibrant Ecosystem



Development Tools



Zephyr®



Governing Board



Technical Steering Committee



Applications & Middlewares

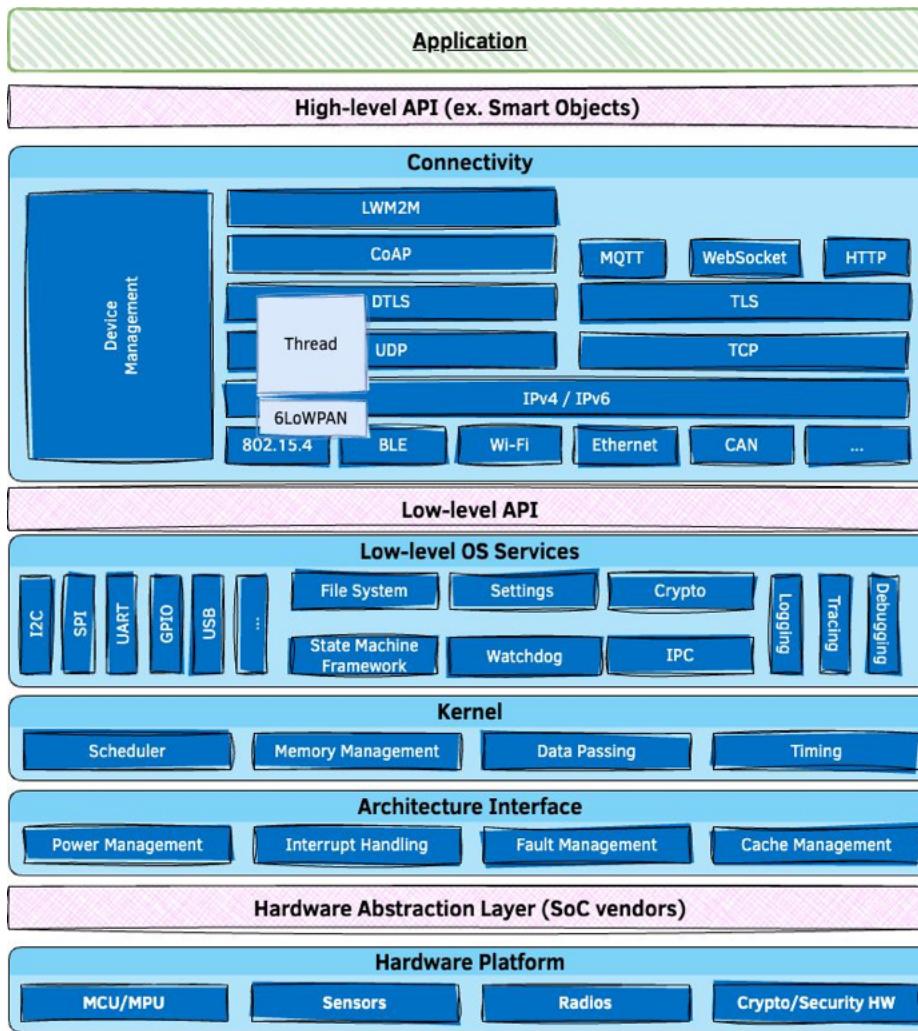


Training & Consulting

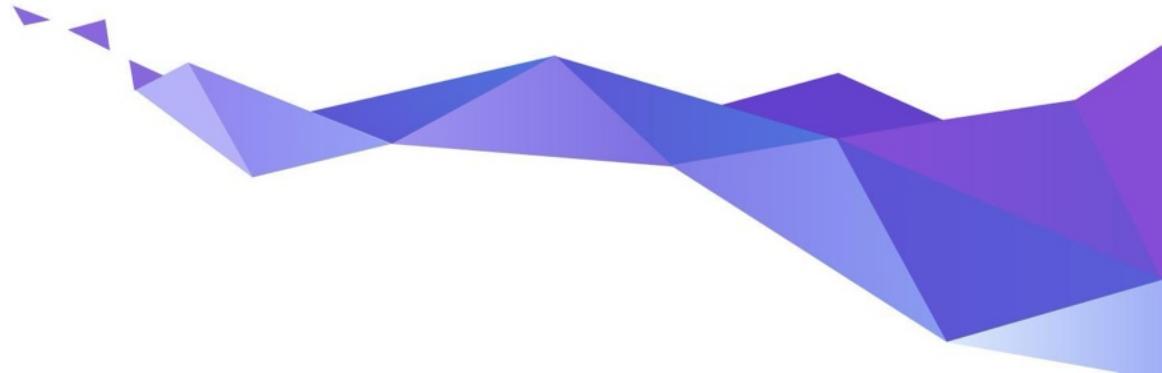


Firmwares & Libraries

Architecture



Diving into Zephyr's features



Devicetree

Describe & configure the available hardware on the target system

Decouple the application from the hardware

- + **Kconfig** for all things configuration



docs.zephyrproject.org/latest/build/dts

```
&i2c1 {
    pinctrl-0 = <&i2c1_scl_pb8 &i2c1_sda_pb9>;
    pinctrl-names = "default";
    clock-frequency = <I2C_BITRATE_FAST>;
    status = "okay";

    lsm6dsl@6a {
        compatible = "st,lsm6dsl";
        reg = <0x06a >;
    };

    hts221@5f {
        compatible = "st,hts221";
        reg = <0x5f >;
    };

    // ...
};
```

.dts file example

West meta-tool

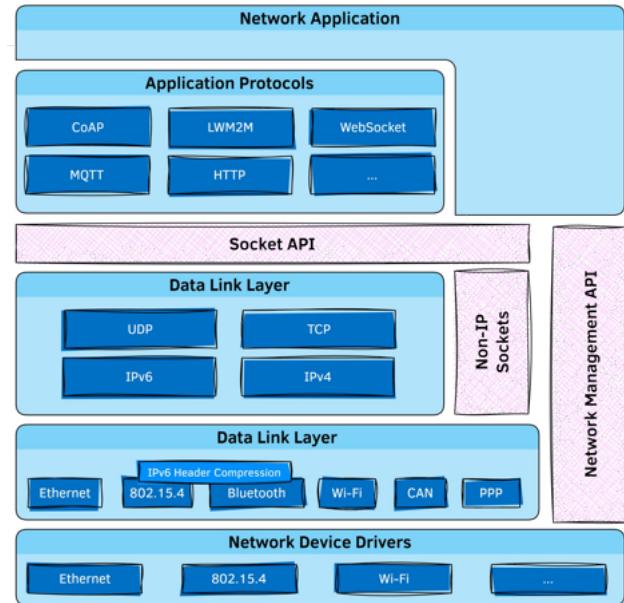
- **Module Management**
 - Simplifies Versioning and integration of various modules/libraries in the build system
- **Build**
- Extensible **command-line interface**
 - e.g. custom commands for specific board
 - Static code analysis, RAM/ROM reports

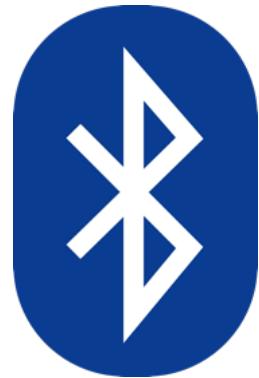
Connectivity Options

- Wide variety of **communication protocols**
 - Ethernet, 802.15.4, Thread, LoRa, Bluetooth, CAN bus, ...
- **Core network protocols** like IPv6, IPv4, UDP, TCP, ICMPv4, and ICMPv6.
- **Security** (ex. TLS, DTLS, ...)
- **Cloud integration** using MQTT, CoAP and HTTP protocols
- **Over-the-air updates**
- **Device management** using OMA LwM2M 1.1 protocol

Native IP Stack

- Built from scratch, on top of Zephyr native kernel concepts
- Dual mode **IPv4/IPv6 stack**
 - DHCP v4, IPv4 autoconf, IPv6 SLAAC, DNS, SNTP
- Multiple network interfaces support
- Time Sensitive Networking support
- **BSD Sockets**-based API
- Supports IP offloading
- **Compliance and security** tested





Bluetooth 5.3 compliant •
LE Controller • Host • Mesh •
Bluetooth-SIG qualifiable

USB 2.0 • USB-C •
Device & Host • WebUSB

Power Management

- **Goal:** reduce power consumption while preserving responsiveness
- **Key concepts**
 - **Tickless kernel**
 - System PM: idle thread, interruptions only for registered events
 - Device PM: device drivers can react to PM state changes
- Handled by the kernel / Customizable by the user



Zephyr USB Device Stack

- **USB 2.0 & USB-C** support
- Supports multiple MCU families (STM32, Kinetis, nRF, SAM,...)
- Supports most common devices classes: CDC, Mass Storage, HID, Bluetooth HCI over USB, DFU, USB Audio, etc.
- Tight integration with the RTOS
- Native execution support for emulated development on Linux
- WebUSB support

Power Management

- Goal: use as little power as possible
- Cross-platform (architecture / SoC agnostic)
- Tickless scheduler
- Handled by the kernel / Customizable by the user

Devicetree

Describe & configure the available hardware on the target system

Decouple the application from the hardware

```
&i2c1 {
    pinctrl-0 = <&i2c1_scl_pb8 &i2c1_sda_pb9>;
    pinctrl-names = "default";
    clock-frequency = <I2C_BITRATE_FAST>;
    status = "okay";

    lsm6dsl@6a {
        compatible = "st,lsm6dsl";
        reg = <0x06a >;
    };
}

hts221@5f {
    compatible = "st,hts221";
    reg = <0x5f >;
};

// ...
};
```

.dts file example



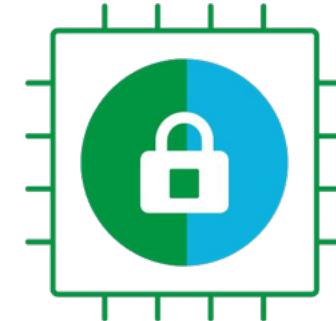
docs.zephyrproject.org/latest/build/dts

Secure boot / Device Management

- Leverage **MCUboot** as secure bootloader
- Application binary can be signed/encrypted
 - Can use hardware keys
- But also:
 - Downgrade prevention
 - Dependency checks
 - Reset and failure recovery
- Over-the-air (OTA) upgrades
 - OMA LwM2M, Eclipse hawkBit
 - Vendor offerings

Hardware security

- **Cryptography APIs**
 - Random Number Generation, ciphering, etc.
 - Supported by crypto HW, or SW implementation (TinyCrypt)
- **Trusted Firmware** integration
 - Firmware verification/encryption
 - Device attestation
 - Management of device secrets



Building on POSIX

- **Zephyr apps can run as native Linux applications**
 - Easier to debug/profile with native tools
 - Connect to real devices using TCP/IP, Bluetooth, CAN
 - Helps minimize hardware dependencies during the development phase
- **Re-use existing code & libraries by accessing Zephyr services through POSIX API**
 - Easier for non-embedded programmers
 - Implementation is optimized for constrained systems
 - Supported POSIX subsets: PSE51, PSE52, and BSD sockets



docs.zephyrproject.org/latest/guides/portability posix.html

A real-time OS

Benchmark on Arm Cortex-M4F running at 120 MHz

Operation	Time
Thread create	2.5 µs
Thread start	3.6 µs
Thread suspend	3.3 µs
Thread resume	3.8 µs
Context switch (yield)	2.2 µs
Get semaphore	0.6 µs
Put semaphore	1.1 µs



github.com/zephyrproject-rtos/zephyr/tree/main/tests/benchmarks

Graphical User Interfaces

- Drivers available for various types of displays
 - LCD
 - OLED
 - Touch panel displays
 - E-ink
- LVGL integration
- Support for video capture and output



Inter-Process Communication

- **Built-in kernel services** (see table)
- **IPC service**
 - 1-to-1 or 1-to-many communications
 - No-copy API
- **zbus** (Zephyr Message Bus)
 - 1-to-1, 1-to-many, or many-to-many channel-based communications
 - Synchronous or asynchronous

Object	Bidirectional?	Data structure
FIFO	✗	Queue
LIFO	✗	Queue
Stack	✗	Array
Message queue	✗	Ring buffer
Mailbox	✓	Queue
Pipe	✗	Ring buffer

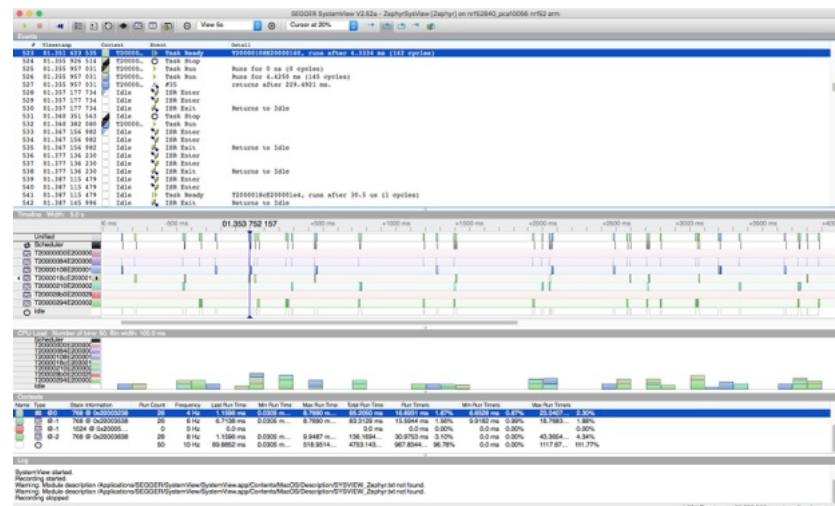
Data passing objects available in Zephyr kernel



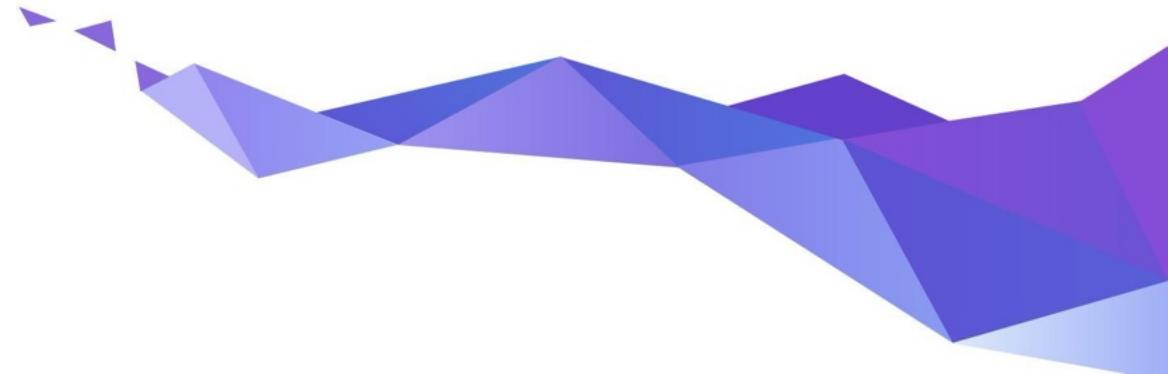
A typical zbus application architecture

Tracing & Debugging

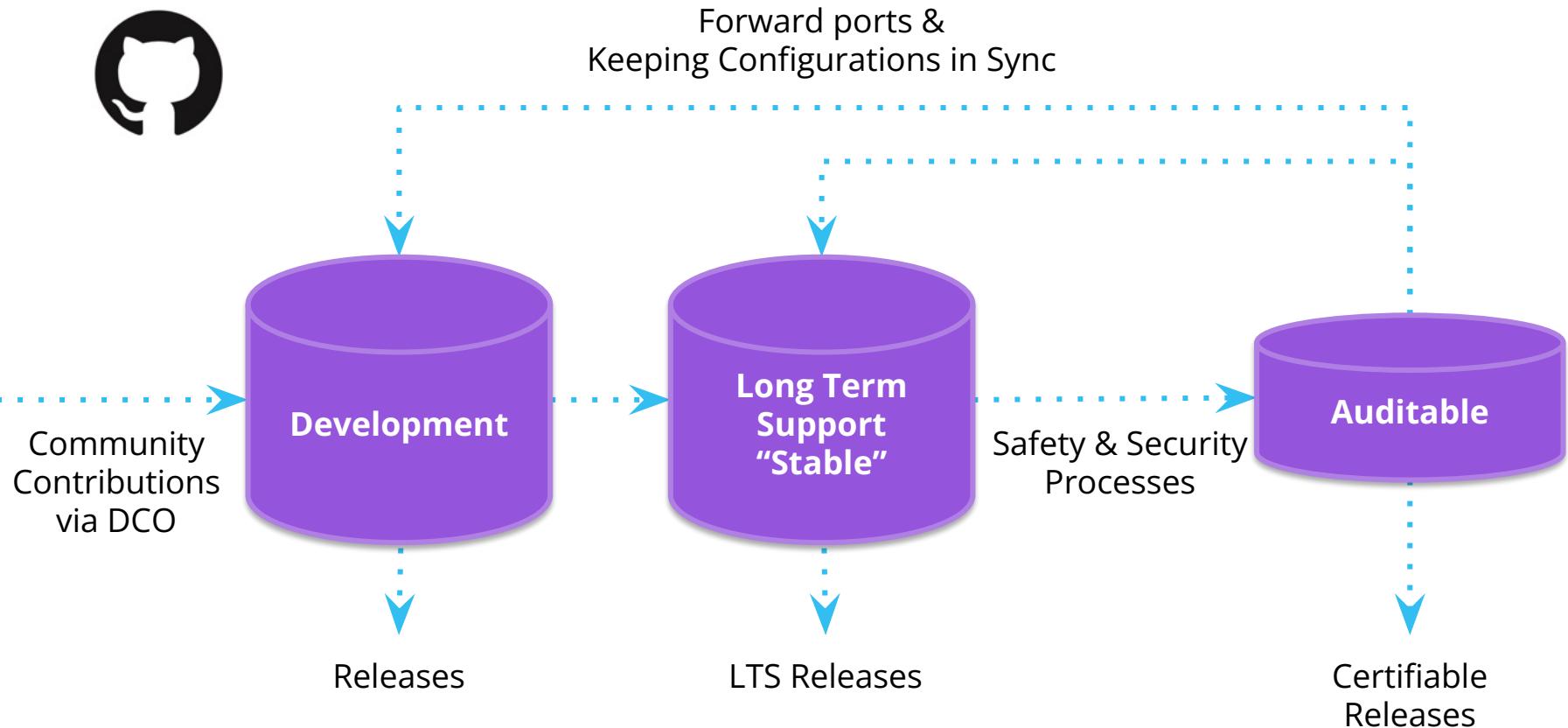
- Advanced **logging** framework
 - Multiple backends (UART, network, file system, ...)
 - Compile-time & runtime filtering
- **Tracing** framework
 - Visualize the inner-working of the kernel and its various subsystems
 - Object tracking (mutexes, timers, etc.)



Safety & Security



Code Repositories





Long Term Support (Zephyr 2.7.x)

- **Product Focused**
- Current with latest **Security Updates**
- Compatible with new hardware
 - Functional support for new hardware is regularly backported
- **Tested:** Shorten the development window and extend the Beta cycle to allow for more testing and bug fixing
- **Supported for 2+ years**
- **⚠ Doesn't include cutting-edge functionality**



github.com/zephyrproject-rtos/zephyr/releases/tag/zephyr-v2.7.0



Long Term Support (LTS - 1.14)

The image displays four GitHub release pages for the Zephyr project, illustrating the evolution of the LTS 1.14 series:

- Zephyr 1.14.0**: Released on April 16, 2018, with 5126 commits. Major enhancements include support for over 160 board configurations across 8 architectures, timing subsystem improvements, symmetric multi-processing, and support for the x86_64 architecture.
- Zephyr 1.14.1**: Released 26 days later with 5126 commits. This is an LTS maintenance release with fixes, addressing security vulnerabilities (CVEs) such as CVE-2020-10019, CVE-2020-10221, and CVE-2020-10222.
- Zephyr LTS 1.14.2 (Maintenance Release)**: Released 25 days after 1.14.1 with 11296 commits. It includes security fixes for CVEs like CVE-2020-10023, CVE-2020-10224, and CVE-2020-10227.
- Zephyr v1.14.3**: Released 23 days after 1.14.2 with 369 commits. This is an LTS maintenance release with fixes for CVEs such as CVE-2020-10066, CVE-2020-10069, CVE-2020-13601, and CVE-2020-13602.

Each release page also lists "Issues Fixed" and provides links to detailed vulnerability information and GitHub issue history.

Delivered bug fixes and latest security updates for 2 years!

Auditable

- An **auditable code base** will be established from a **subset of the Zephyr OS LTS**
- Code bases will be kept in sync
- More rigorous processes (necessary for certification) will be applied to the auditable code base.
- Processes to achieve selected certification to be:
 - Determined by Safety Committee and Security Committee
 - Coordinated with Technical Steering Committee



Project Security Documentation

- Project Security Overview
- Started with documents from other projects
- Built around Secure Development, Secure Design, and Security Certification
- Ongoing process, rather than something to just be accomplished



The screenshot shows the Zephyr Project's security documentation. At the top, there's a header with the Zephyr logo and version 3.6.99. Below it is a search bar and a navigation menu with links like 'CONTENTS', 'Introduction', 'Developing with Zephyr', 'Kernel', 'OS Services', 'Build and Configuration Systems', 'Connectivity', 'Hardware Support', 'Contributing to Zephyr', 'Project and Governance', and 'Security'. Under 'Security', the 'Zephyr Security Overview' page is selected, indicated by a purple vertical bar on the left. The main content area starts with an introduction about the security process and its goals. It then details the 'Overview and Scope' and 'Main Steps' of the process.

Docs / Latest » Security » Zephyr Security Overview

[Open on GitHub](#) [Report an issue with this page](#)

This is the documentation for the latest (main) development branch of Zephyr. If you are looking for the documentation of previous releases, use the drop-down menu on the left and select the desired version.

Zephyr Security Overview

Introduction

This document outlines the steps of the Zephyr Security Subcommittee towards a defined security process that helps developers build more secure software while addressing security compliance requirements. It presents the key ideas of the security process and outlines which documents need to be created. After the process is implemented and all supporting documents are created, this document is a top-level overview and entry point.

Overview and Scope

We begin with an overview of the Zephyr development process, which mainly focuses on security functionality.

In subsequent sections, the individual parts of the process are treated in detail. As depicted in Figure 1, these main steps are:

1. **Secure Development:** Defines the system architecture and development process that ensures adherence to relevant coding principles and quality assurance procedures.
2. **Secure Design:** Defines security procedures and implement measures to enforce them. A security architecture of the system and relevant sub-modules are created. Threats are identified and countermeasures designed. Their



Software Supply Chain

- Zephyr ships an **SBOM** (Software Bill of Materials) with each release
- Downstream consumers can leverage built-in tools to, in turn, generate source & build SBOMs for their deliverables

```
[...]
FileName: ./zephyr/zephyr.elf
SPDXID: SPDXRef-File-zephyr.elf
FileChecksum: SHA1: e74cebcac51dabd799957ac51e4edcd32541103d
[...]
Relationship: SPDXRef-File-zephyr.elf GENERATED_FROM SPDXRef-File-dev-handles.c
Relationship: SPDXRef-File-zephyr.elf GENERATED_FROM SPDXRef-File-isr-tables.c
Relationship: SPDXRef-File-zephyr.elf STATIC_LINK SPDXRef-File-libapp.a
Relationship: SPDXRef-File-zephyr.elf STATIC_LINK SPDXRef-File-libzephyr.a
Relationship: SPDXRef-File-zephyr.elf STATIC_LINK SPDXRef-File-libisr-tables.a
Relationship: SPDXRef-File-zephyr.elf STATIC_LINK SPDXRef-File-libkernel.a
[...]
```

Automating SBOM Generation During Build!



1. Create a build directory with CMake file API enabled
2. Build project with “build metadata” enabled
3. Compute SBOM(s)

```
west SPDX --init -d BUILD_DIR
```

```
west build -d BUILD_DIR -- -DCONFIG_BUILD_OUTPUT_META=y
```

```
west SPDX -d BUILD_DIR
```



`zephyr.spdx` SBOM for the **Zephyr source files** actually used by your application

`app.spdx` SBOM for the source files of your **application**

`build.spdx` SBOM for **all the build objects**, inc. of course your final image



SBOM's at Scale...Automatically

708 boards

13 apps

**All BUILT,
PASSED,
GENERATED
have 3 SBOMs
available to
download &
inspect**

The screenshot shows the RENODE ZEPHYR DASHBOARD interface. On the left, there is a sidebar with sections for ARCHITECTURE (listing ARC, ARM32, ARM64, MIPS, NIOS2, RISCV32, RISCV64, SPARC, X86, X86-64, XTENSA) and BUILD DETAILS (SHOW SIMULATION, ED17006BE4, 9D46C2F8BE). Below these are links for contacting support and a section for supported boards. The main area displays a table of boards with columns for BOARD NAME, HELLO WORLD, PHILOSOPHERS, SHELL MODULE, TENSORFLOW LITE MICRO, and MICROPYTHON. Each board row shows the count of passed tests (e.g., 11 PASSED, 7 PASSED, etc.) and a series of buttons indicating build status (PASSED, GENERATED, BUILT, NOT BUILT). A download icon is also present for each board.

BOARD NAME	HELLO WORLD	PHILOSOPHERS	SHELL MODULE	TENSORFLOW LITE MICRO	MICROPYTHON
ARC (20)	11 PASSED	7 PASSED	8 PASSED	11 PASSED	6 PASSED
ARM32 (529)					
ARM64 (26)					
MIPS (2)					
NIOS2 (2)					
RISCV32 (35)					
Andes ADP-XC7K AE350	PASSED	GENERATED	PASSED		PASSED
ESP32-C3	BUILT	BUILT	BUILT		BUILT
ESP32C3 LuatOS Core	BUILT	BUILT	BUILT	BUILT	BUILT
ESP32C3 LuatOS Core USB	BUILT	BUILT	BUILT	BUILT	BUILT
GigaDevice GD32VF103C-STARTER	GENERATED	GENERATED	GENERATED	GENERATED	NOT BUILT
GigaDevice GD32VF103V-EVAL	GENERATED	GENERATED	GENERATED	GENERATED	GENERATED
ICE-V Wireless	BUILT	BUILT	BUILT	BUILT	BUILT

Source: <https://zephyr-dashboard.renode.io/>

CVE Numbering Authority

- Registered with MITRE in 2017
 - We issue our own CVEs
- **Zephyr Project Security Incident Response Team (PSIRT)**
 - Volunteers from the Security Subcommittee led by the Zephyr Security Architect.

Zephyr Project

The majority of the links on this page redirect to external websites ⓘ; these links will open a new window or tab depending on the web browser used.

Scope	Zephyr project components, and vulnerabilities that are not in another CNA's scope
Root	MITRE Corporation
Security Advisories	View Advisories
Program Role	CNA
Organization Type	Vendors and Projects
Country*	USA

OpenSSF Gold Badge

- Core Infrastructure Initiative
Best Practices Program
- Awards badges based on
“project commitment to
security”
- Mostly about project
infrastructure: is project
hosting, etc following security
practices
- Gold status since Feb, 2019



Zephyr Project

[Expand panels](#) [Show all details](#) [Hide met & N/A](#)

Projects that follow the best practices below can voluntarily self-certify and show that they've achieved an Open Source Security Foundation (OpenSSF) best practices badge. [Show details](#)

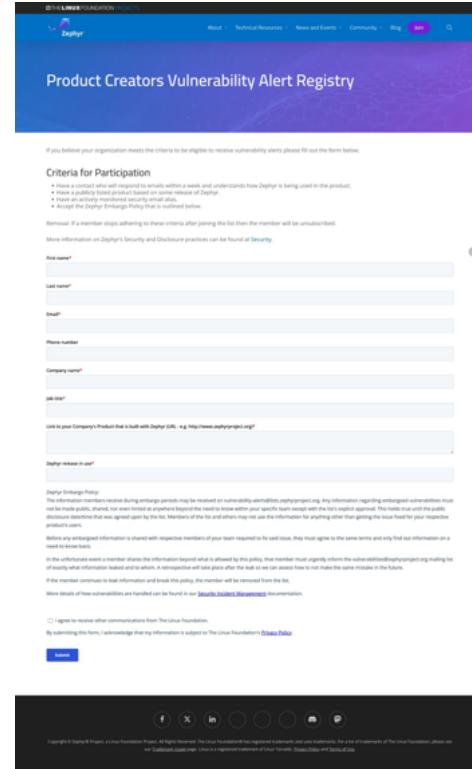
If this is your project, please show your badge status on your project page! The badge status looks like this: `openssf best practices gold`. Here is how to embed it: [Show details](#)

These are the `passing` level criteria. You can also view the `silver` or `gold` level criteria.

▼ Basics	13/13	•
▼ Change Control	9/9	•
▼ Reporting	8/8	•
▼ Quality	13/13	•
▼ Security	16/16	•
▼ Analysis	8/8	•

Vulnerability Alert Registry

- For an **embargo** to be effective, product makers need to be **notified early** so they can **remediate**
- The project aims at **fixing issues within 30 days** to give **vendors 60 days** before publication of vulnerability



The screenshot shows a web form titled "Product Creators Vulnerability Alert Registry". It includes fields for company information, contact details, and embargo preferences. A note at the bottom states that the form is subject to the Zephyr Project's Privacy Policy.

If you believe your organization meets the criteria to be eligible to receive vulnerability alerts, please fill out the form below.

Criteria for Participation

- Have a contact who will respond to emails within a week and understands how Zephyr is being used in the product.
- Have a public-facing URL for your product or service.
- Have an actively maintained security email alias.
- Accept the Zephyr Embargo Policy as outlined below.

If a member stops adhering to these criteria after joining the list then the member will be unsubscribed.

More information on Zephyr's Security and Disclosure practices can be found at [Security](#).

First Name* _____

Last Name* _____

Email* _____

Phone Number _____

Company name* _____

Job title* _____

Link to your Company's Product that is built with Zephyr (URL - e.g. <http://zephyr-project.org/>) _____

Zephyr release in use* _____

Zephyr Embargo Policy

The information received during embargo periods may be released on a confidential, pre-agreed-upon basis. Any information regarding embargo conversations must be kept confidential, shared, or even released at anyone's discretion beyond the terms within your specific issue response until the user's explicit approval. This holds true until the disclosure disclaimer that was agreed upon by the list. Members of the list and others may use the information for anything other than getting the issue fixed by your respective development teams.

Before any embargoed information is shared with respective members of your team in response to the said issue, they must agree to the same terms and only find out information on a need-to-know basis.

In the unfortunate event a member shares the information beyond what is allowed by the policy, that member must explicitly inform the subteam/distribution group/engaging mailing list if the member continues to leak information and do so. A reiteration will take place after the task so we can ensure to not make the same mistake in the future.

More details on how notifications are handled can be found in our [Zephyr Project Documentation](#) documentation.

I agree to receive other communications from The Linux Foundation.

By submitting this form, I acknowledge that my information is subject to The Linux Foundation's [Privacy Policy](#).

Submit

Copyright © Zephyr Project, a Linux Foundation Project. All Rights Reserved. The Linux Foundation has registered trademarks and uses trademarks. For a list of trademarks of the Linux Foundation, please see [Trademark List](#). Zephyr is a registered trademark of Linux Foundation. OpenZephyr™ is a trademark.



zephyrproject.org/vulnerability-registry

Zephyr PSIRT: Remediation and Response



Advisory Issued by project on 20201208:

- Zephyr current release (2.4) does not use Fnet or other stacks.
- The Zephyr LTS release 1.14 contains an implementation of the TCP stack from Fnet.

Of the vulnerabilities reported in Fnet, 2, [CVE-2020-17468](#), and [CVE-2020-17469](#), are in the IPv6 Fnet code, one, [CVE-2020-17467](#), affects Link-local Multicast Name Resolution LLMNR), and 2, [CVE-2020-24383](#), and [CVE-2020-17470](#) affect DNS functionality.

None of the affected code has been used in the Zephyr project, while 1.14 does use the Fnet TCP, it does not use the affected IPv6, DNS or LLMNR code.



zephyrproject.org/zephyr-security-update-on-amnesia33



The image shows the cover of the 'AMNESIA:33 Research Report Executive Summary'. It features a dark background with a grid of circular icons representing various technologies and research areas. At the top right, the title 'AMNESIA:33' is displayed in large, bold, white letters, with 'Research Report Executive Summary' below it in a smaller white font. On the far right, there is a small navigation bar with arrows and text.

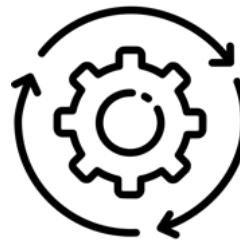
AMNESIA:33
Research Report Executive Summary

AMNESIA:33 is the first study we have published under Project Memoria. In this study, we discuss the results of the security analysis of seven open source TCP/IP stacks and report a bundle of 33 new vulnerabilities found in four of the seven analyzed stacks that are used by major IoT, OT and IT device vendors.

Four of the vulnerabilities in AMNESIA:33 are critical, with potential for remote code execution on certain devices. Exploiting these vulnerabilities could allow an attacker to take control of a device, thus using it as an entry point on a network. One of the most critical vulnerabilities found in the study is a persistent denial of service attack vector, which can be used as the final target of an attack. For enterprise organizations, this means they are at increased risk of having their network compromised or having malicious actors undermine their business continuity. For consumers, this means that their IoT devices may be used as part of large attack campaigns, such as botnets, without them being aware.

forescout.com/amnesia33/ research@forescout.com toll free 1-866-377-8771

Zephyr Security Summary



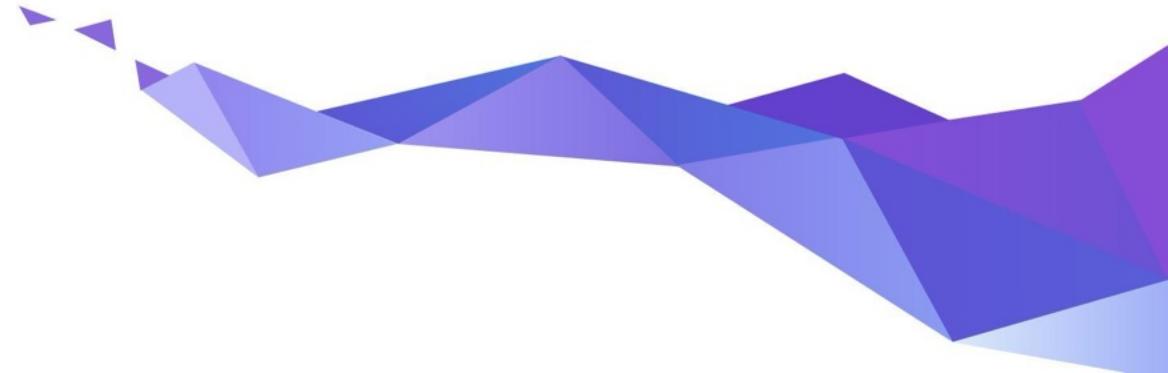
Documented secure
coding practices

Vulnerability response
criteria publicly
documented

Weekly Coverity scans
MISRA scans

SBOM generation

Certification

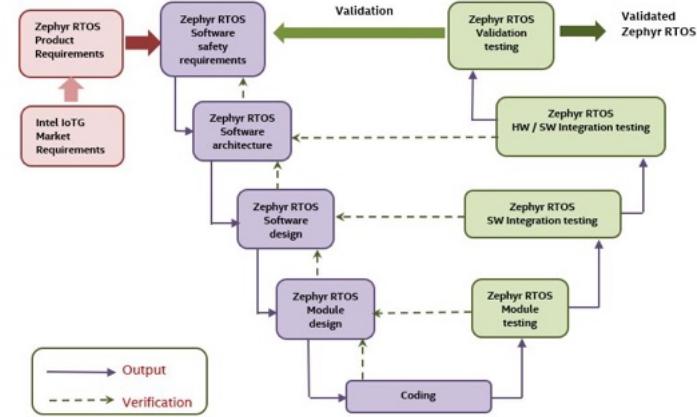


Compliant Development: V-model

It is difficult to map a stereotypical open-source development to the V-model

- Specification of features
- Comprehensive documentation
- Traceability from requirements to source code
- Number of committers and information known about them

Zephyr RTOS functional safety work products mapping to IEC 61508-3 V model



⇒ Provide the evidences that open source developers can map to compliance and meet all requirements

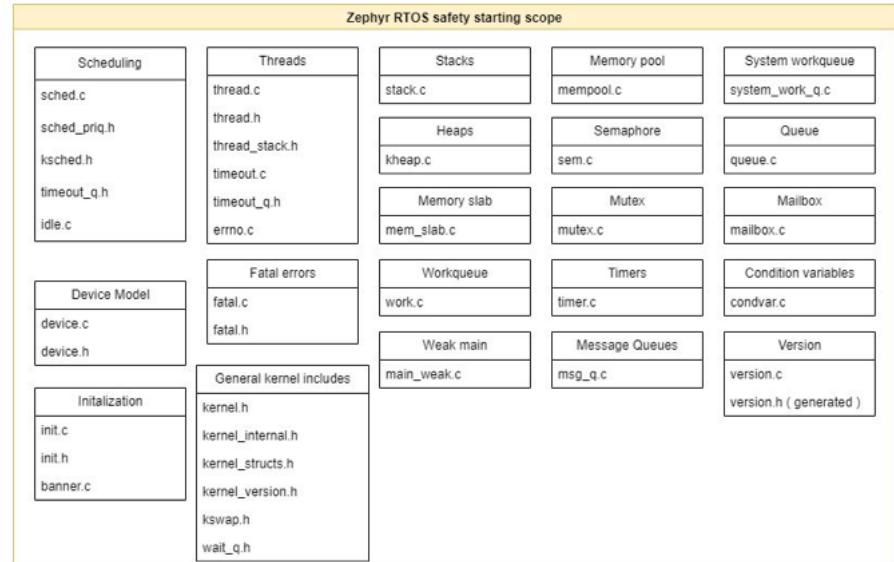
Safety Collateral Proposal

Draft (Pending Approval by Certification Authority)			
Artifacts	Type of Doc	Owner	Work in progress Visibility
Plans			
Safety Development Plan	Plan/Process	Safety Committee	Public - Project Docs
Safety Assessment Plan	Plan/Process	FSM	Safety Committee Github
Verification / Validation / Integration Test Plan	Plan/Process	Testing WG	Public - Project Docs
Software Development Plan	Plan/Process	TSC	Public - Project Docs
Configuration and Change Management Plan	Plan/Process	TSC	Public - Project Docs
Coding Guideline	Plan/Process	TSC	Public - Project Docs
Tools Documentation	Plan/Process	TSC	Public - Project Docs
Specifications			
Safety Scope Definition	Category		
Safety Software Requirement Specification (SRS) **	Spec.	Safety Committee	Safety Committee Github
Safety Software Architecture and Interface Specification (SAIS) **	Spec.	Safety Committee	Safety Committee Github
Safety Software Component Design Specification (SMDS) **	Spec.	Safety Committee	Safety Committee Github
Safety Software Component Test Specification (SMTS) **	Spec.	Safety Committee	Safety Committee Github
Safety Software Integration Test Specification (SITS) **	Spec.	Safety Committee	Safety Committee Github
Safety Software Test Specification (STS) **	Spec.	Safety Committee	Safety Committee Github
Sources			
Source Code	Category		
- Coding Guideline Compliance	Source	TSC	Public
Project Documentaton	Source	TSC	Public
- Software Requirement Specifications	Spec	TSC	Public
- Software Architecture and Interface Specification	Spec	TSC	Public
- Software Component Design Specification	Spec	TSC	Public
Project Testing	Category		
- Software Component/Unit Test Specification	Source	TSC	Public
- Software Integration Test Specification	Spec	TSC	Public
- Software Test Specification	Spec	TSC	Public
- Tests	Source	TSC	Public
Reports			
Code Review Report (pre-merge)	Category		
Code Change Test Report (post-merge)	Report	TSC	Public
Test Coverage Report	Report	Testing WG	Public
Coding Guideline Compliance Report	Report	Testing WG	Public
Traceability Report	Report	Safety WG & Security WG	Public
Tools Classification	Report	Safety WG	Public
Tools Validation	Report	Safety Committee	Public
Fault Injection Test Report	Report	Safety Committee	TBD (based on specific tools)
Safety Traceability Report (for Safety Scope) **	Report	Safety Committee	Safety Committee
Safety Test Coverage Report (for Safety Scope) **	Report	Safety Committee/FSM	Safety Committee
Safety Analysis (e.g., FMEA)	Report	Safety Committee/FSM	Safety Committee
Safety Analysis (e.g., FMEA)	Report	FSM	Safety Committee
Manuals			
Software User Manual	Category		
Safety Manual	Manual	TSC	Public
Certificates	Manual	FSM	Safety Committee
All safety certificates	Certificate	Safety Committee	N/A

- Requirement definition, Source Code & Test linkage are public; and developed in open using [strictdoc](#)
- The set of requirements (and associated traceability) that are applicable to safety scope is managed by the safety committee.
- Other project artifacts have owners designated.

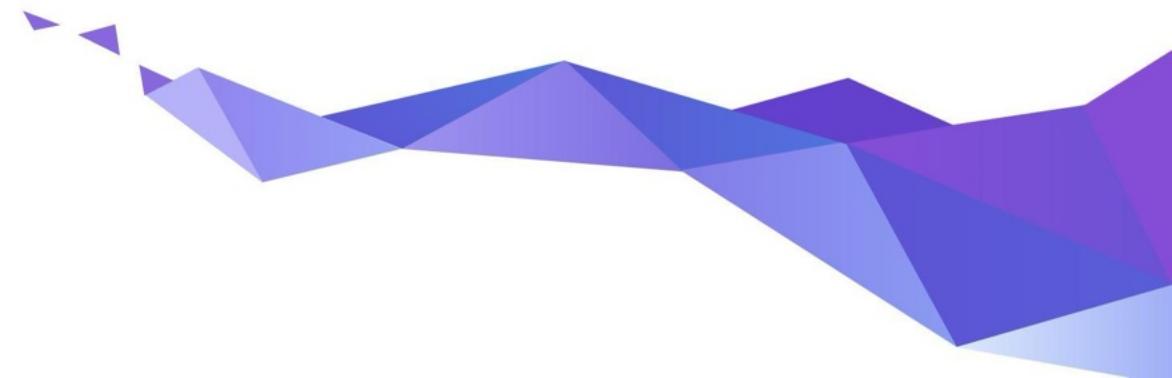
Initial certification focus

- Start with a limited scope of kernel and interfaces
- Initial target is IEC 61508 SIL 3 / SC 3 (IEC 61508-3, 7.4.2.12, Route 3s)
- Option for 26262 certification has been included in contract with certification authority should there be sufficient member interest



Scope can be **extended** to include **additional components** with associated **requirements** and **traceability** as determined by the safety committee

Ecosystem & Governance





Zephyr Project: Platinum Members





Zephyr Project: Silver Members

ac6



arm



ezurio



Qt Group

RENESAS



STERNUM



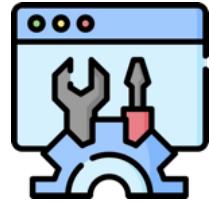
SYNOPSYS



WNDRVR



Vibrant Ecosystem



Development Tools



Zephyr®



Governing Board



Technical Steering Committee

Contributors



Applications & Middlewares



Training & Consulting



Firmwares & Libraries

Ecosystem // Dev Tools



Development Tools



Training & Consulting



Firmwares & Libraries



Applications & Middlewares

IDE



eclipse



Compilers



Debuggers / Tracing Tools



Emulation / Simulation

RENODE™

WOKWI



Ecosystem // Training & Consulting



Development Tools



Training & Consulting



Firmwares & Libraries



Applications & Middlewares

Training



NORDIC
SEMICONDUCTOR



Golioth



THE
LINUX
FOUNDATION



percepio
SENSING SOFTWARE

ac6

Services & Consulting



Baylibre



NXP



Laird
CONNECTIVITY



antmicro

WNDRVR

avsystem

Ecosystem // Firmwares & Libraries



Development Tools



Training & Consulting

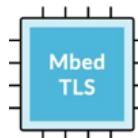


Firmwares & Libraries



Applications & Middlewares

Security



TinyML



Language runtimes



Others



Ecosystem // Apps & Middlewares



Development Tools



Training & Consulting



Firmwares & Libraries



Applications & Middlewares

Remote Management



HERALD



Golioth



AVSYSTEM



Blynk



Memfault



STERNUM



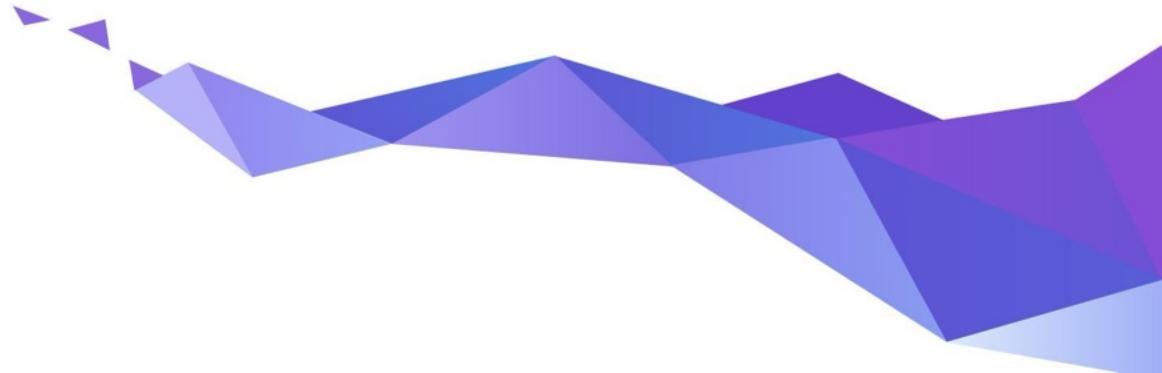
blues

Robotics

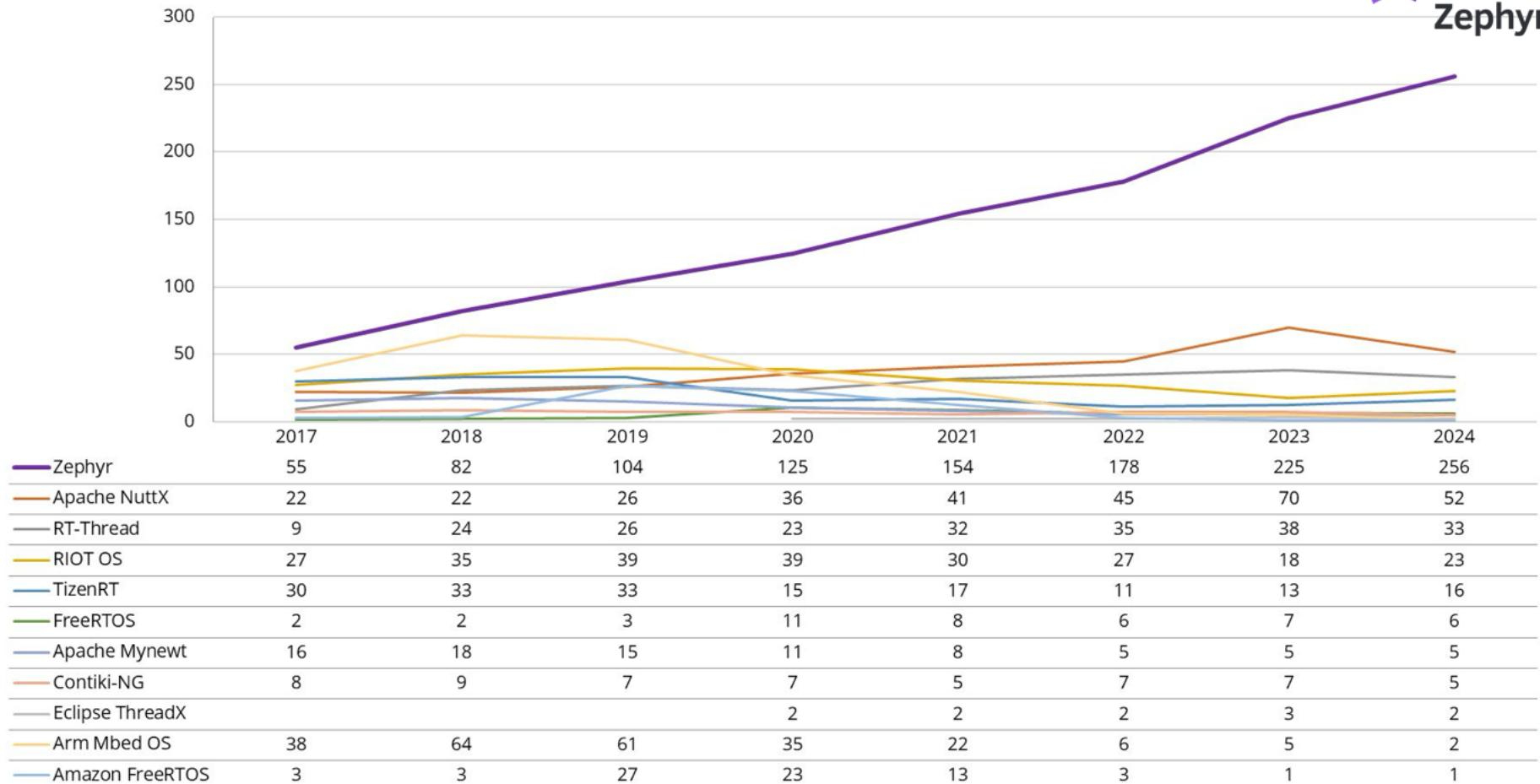


ROS

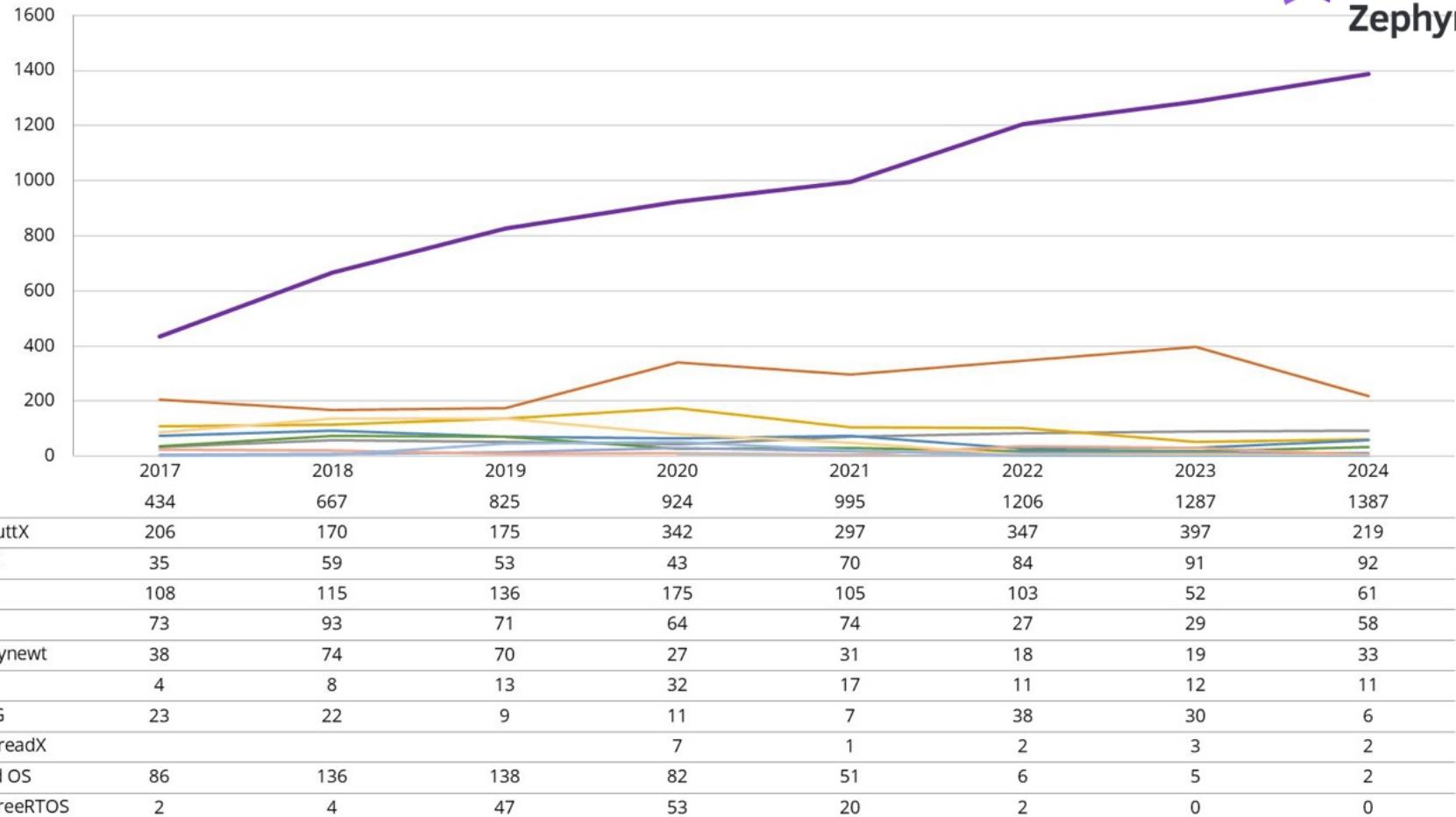
Zephyr in the RTOS landscape



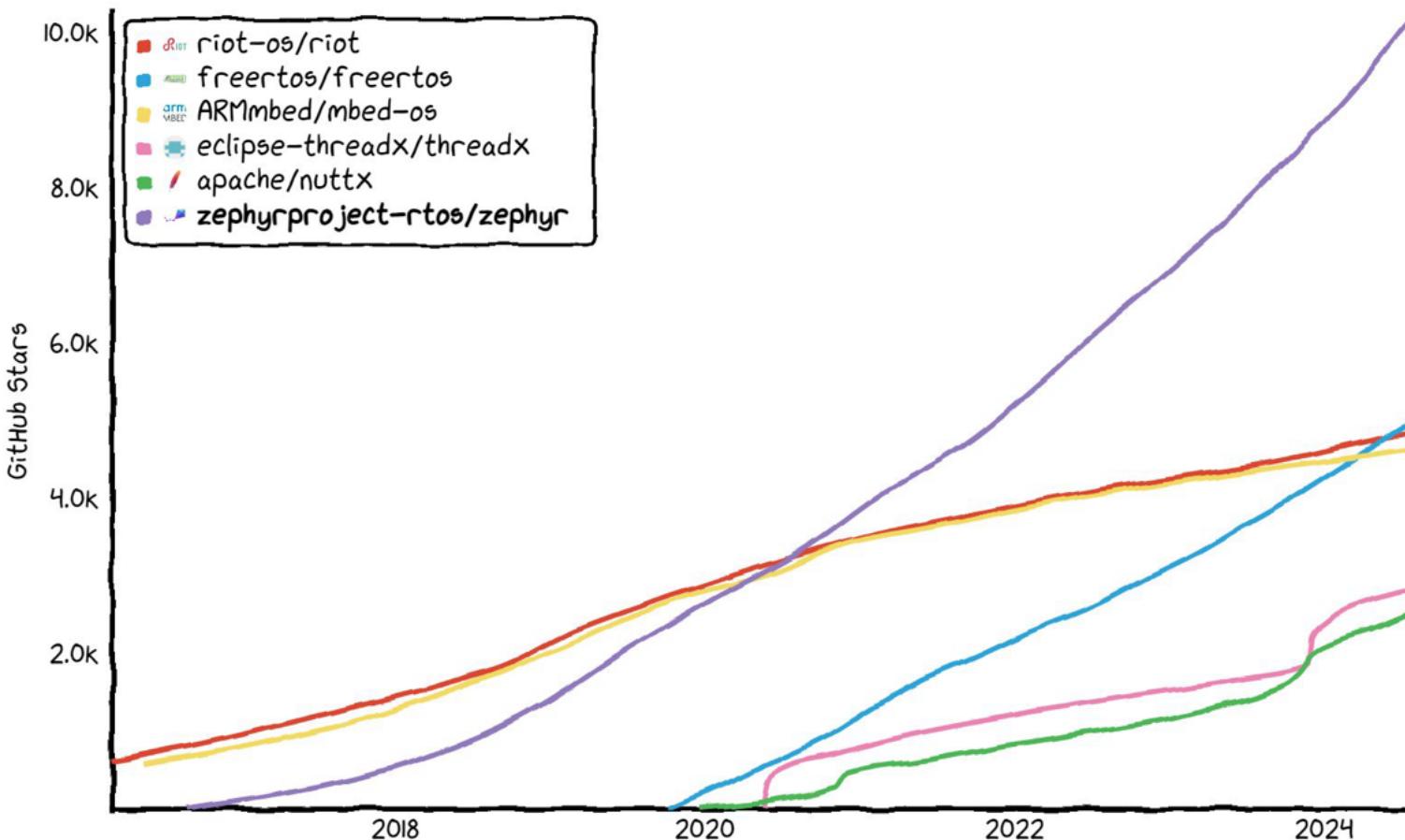
Average Number of Unique Contributors per Month



Average Number of Commits per Month



Star History



GitHub Clones & Unique Visitors



2024-07-31 → 2024-08-13

~186 unique clones per day
~1375 unique visitors per day

Getting started – Important links

- Check out the official [Getting Started Guide](#)
- Dig into the hundreds of **code samples**
- Check the catalog of 100s of available Devicetree bindings
 - No driver for your HW? Chances are a similar driver already exists and writing one is not as hard or daunting as you would think!
- Reach out to the community on **Discord**

Zephyr Participation Information



zephyrproject.org



github.com/zephyrproject-rtos



lists.zephyrproject.org



chat.zephyrproject.org



zephyrproject.org

