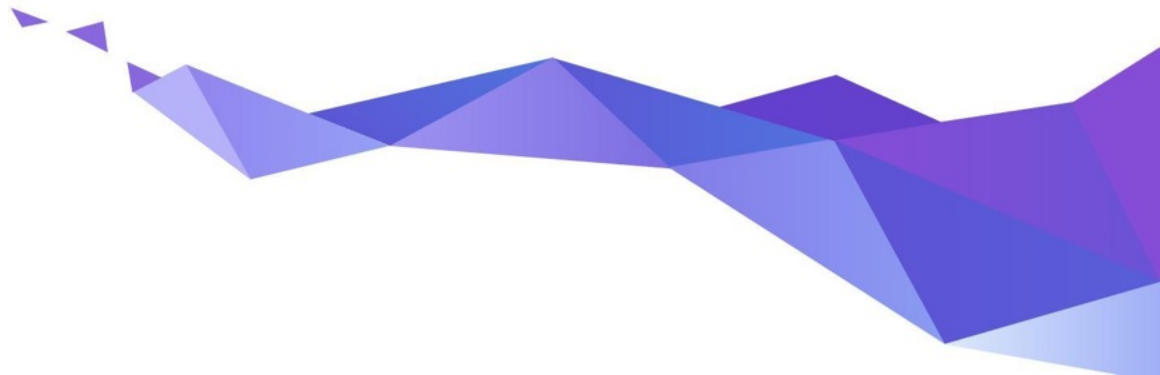


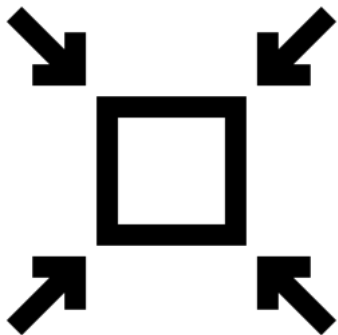
Zephyr Project Overview

A proven RTOS ecosystem, by developers, for developers



Use cases for a real-time OS



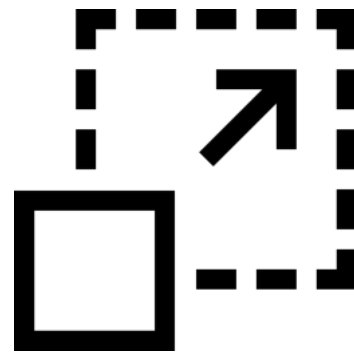


SMALL

< 8KB Flash

< 5KB RAM

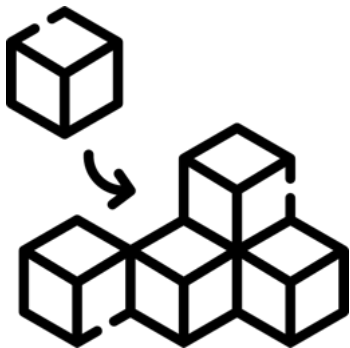
yet



SCALABLE

from small sensor nodes

... to complex multi-core systems



FLEXIBLE

Heavily customizable

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

yet



SECURE

Built with safety & security in mind

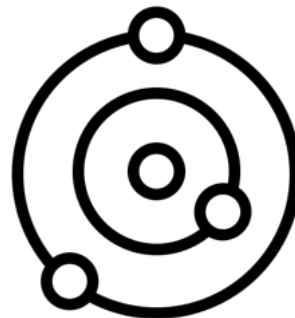
Certification-ready
Long-term Support



OPEN-SOURCE

Permissively licensed (Apache 2.0)

Vendor-neutral governance



ECOSYSTEM

Vibrant community

Supported by major silicon vendors

Features overview



- Comprehensive, **lightweight**, kernel & supporting services
 - Fits where Linux is too big
- Inherently **portable & secure**
- **Highly connected**
 - Bluetooth 5.0 & BLE
 - Wi-Fi, Ethernet, CANbus, ...
 - IoT protocols: CoAP, LwM2M, MQTT, OpenThread, ...
 - USB & USB-C
- **Developer-friendly**
 - Logging, tracing, debugging, built-in shell, Windows/Linux/macOS support, ...



Products Running Zephyr Today



Prolove



Ruuvi Tag



PHYTEC Distancer



Keeb.io BDN9



Hati-ACE



Oticon More



Adhoc Smart Waste



GNARBOX 2.0 SSD



Anicare Reindeer Tracker



Safety Pod



BLiXT solid state circuit breaker



Moto Watch 100



Lildog & Lilcat pet tracker



Rigado IoT Gateway



Livestock Tracker



Laird Connectivity sensors & gateways



BeST pump monitoring



Vestas Wind Turbines



zephyrproject.org/products-running-zephyr

450+ 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

120+ Sensors Already Integrated



adt7420
adx1345
adx1362
adx1372
ak8975
amg88xx
ams_as5600
ams_iAQcore
apds9960
bma280
bmc150_magn
bme280
bme680
bmg160
bmi160
bmi270
bmm150
bmp388
bq274xx
ccs811

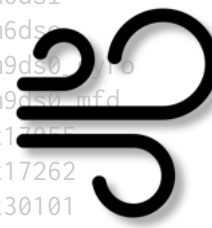
dht
dps310
ds18b20
ens
esp8266
fdc3v3
fxos8700
fxos9700
grove
grow_r502a
hmc58831
hp206c
ht221
i2c50c
i2c605
i2c670
i2c720
icp1125
iis2dh
iis2dlpc



iis2iclx
iis2mdc
iis3dhhc
ina219
ina230
isl2935
ism330dxx
ite_tach_it8xxx2
ite_vcmp_it8xxx2
lis2dh
lis2ds12
lis2dw12
lis2n
lis3n
lm75
lm77
lps22
lps22hh
lps25hb
lsm303dlhc_magn



lsm6ds0
lsm6dsl
lsm6dsx
lsm9ds0
lsm9ds0_mfd
max17055
max17262
max30101
max31875
max44009
max6675
mchp_tach_xec
mcp9804
mcp9808
mcp9810
mcp9812
mcp9814
mcp9816
mcp9818
mcp9820
mcp9822
mcp9824
mcp9826
mcp9828
mcp9830
mcp9832
mcp9834
mcp9836
mcp9838
mcp9840
mcp9842
mcp9844
mcp9846
mcp9848
mcp9850
mcp9852
mcp9854
mcp9856
mcp9858
mcp9860
mcp9862
mcp9864
mcp9866
mcp9868
mcp9870
mcp9872
mcp9874
mcp9876
mcp9878
mcp9880
mcp9882
mcp9884
mcp9886
mcp9888
mcp9890
mcp9892
mcp9894
mcp9896
mcp9898
mcp9900
mcp9902
mcp9904
mcp9906
mcp9908
mcp9910
mcp9912
mcp9914
mcp9916
mcp9918
mcp9920
mcp9922
mcp9924
mcp9926
mcp9928
mcp9930
mcp9932
mcp9934
mcp9936
mcp9938
mcp9940
mcp9942
mcp9944
mcp9946
mcp9948
mcp9950
mcp9952
mcp9954
mcp9956
mcp9958
mcp9960
mcp9962
mcp9964
mcp9966
mcp9968
mcp9970
mcp9972
mcp9974
mcp9976
mcp9978
mcp9980
mcp9982
mcp9984
mcp9986
mcp9988
mcp9990
mcp9992
mcp9994
mcp9996
mcp9998
mcp10000



nrf5
nuvoton_adc_cmp_npcx
nuvoton_tach_npcx
nxp_kin
opt3000
pcnt_encoder3
pms7003
qdec_mcp9804
qdec_nrfx
qdec_sam
qdec_stm32
rpi_pico_temp
sbs_gauge
sgp40
sht3xd
sht4x
shtcx
si7006
si7055
si7060



si7210
sm3511t
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

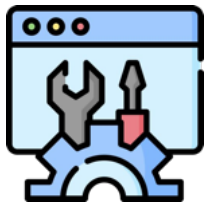


Xtensa



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

Vibrant Ecosystem



Development Tools



Governing Board

Technical Steering Committee

Contributors



Applications & Middlewares

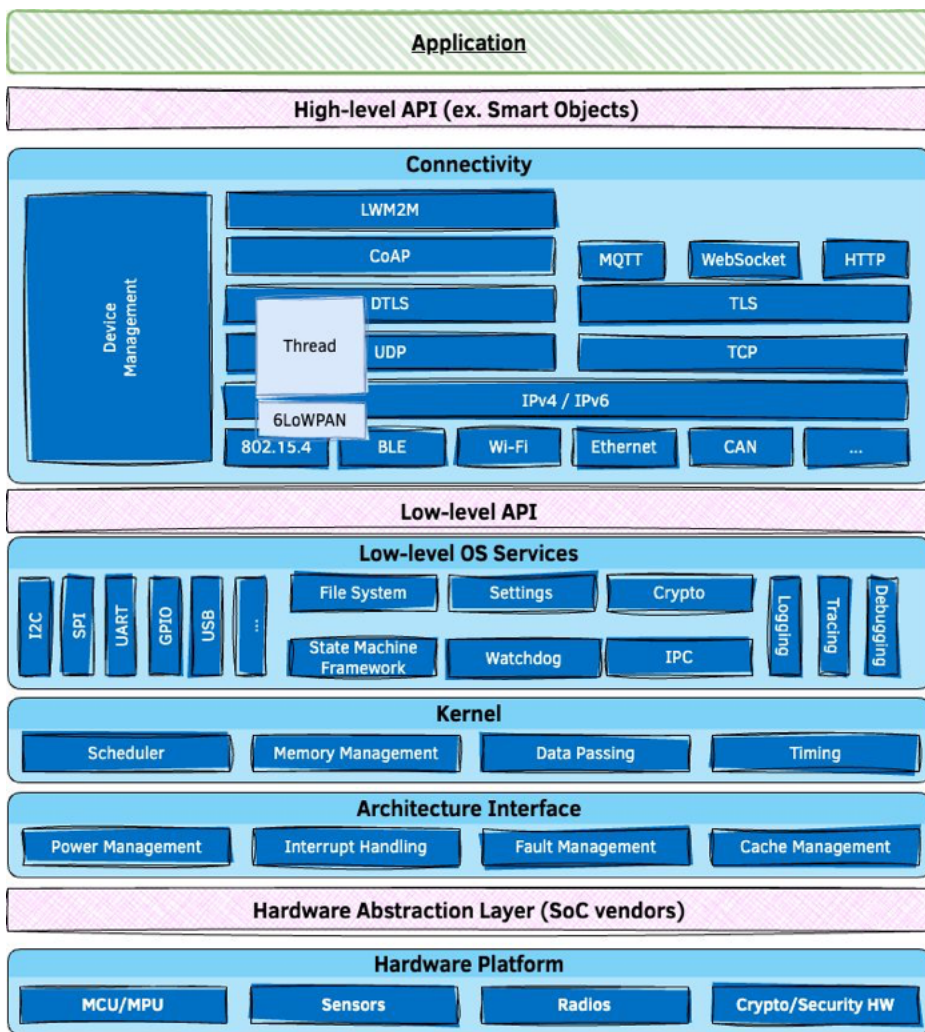


Training & Consulting

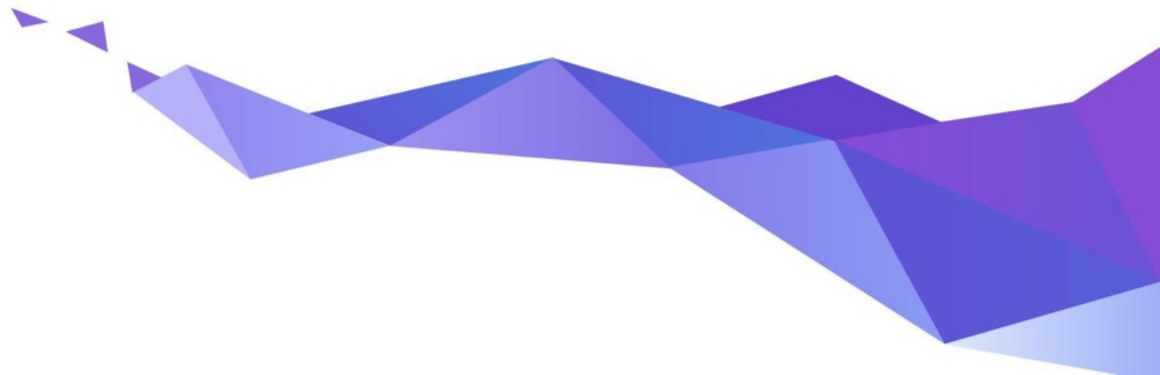


Firmwares & Libraries

Architecture



Diving into Zephyr's features



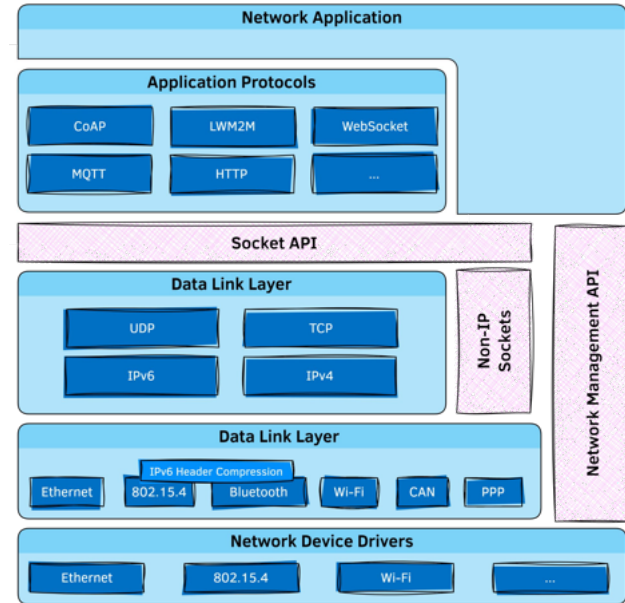
IoT Connectivity Options

- 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, SNTTP
- Multiple network interfaces support
- Time Sensitive Networking support
- **BSD Sockets**-based API
- Supports IP offloading
- **Compliance and security** tested



Bluetooth Host and Mesh

- **Bluetooth 5.3 compliant**
- Highly configurable
- Portable to all architectures supported by Zephyr
- Low Energy & experimental Bluetooth Classic
- IPSP/6LoWPAN for IPv6 connectivity over Bluetooth LE
- Multiple HCI transports

Bluetooth Low Energy Controller

- **Bluetooth 5.3 compliant** and qualified (5.1)
- Support for multiple BLE radio hardware architectures
 - Nordic nRF5x on Arm Cortex-M
 - VEGAboard on RISC-V
- Proprietary radios (downstream only)
- Unlimited role and connection count
- Concurrent multi-protocol support ready
- Multiple advertiser and scanner instances

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

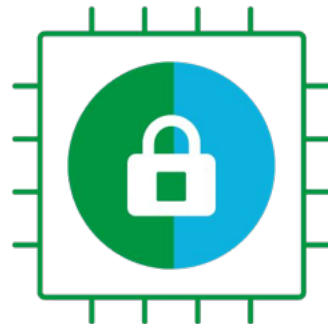
Secure boot / Device Management



- 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



A real-time OS



Benchmark on Arm Cortex-M4F running at 120 MHz

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

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



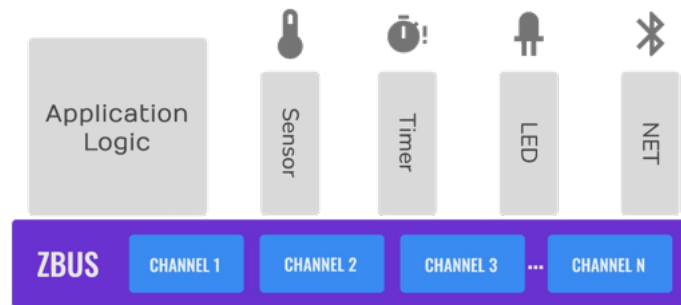
Inter-Process Communication



- **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

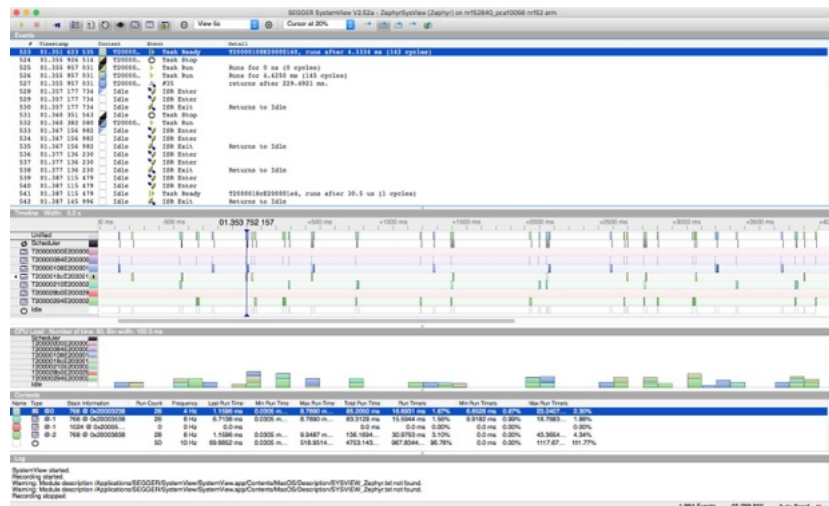


A typical zbus application architecture

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.)



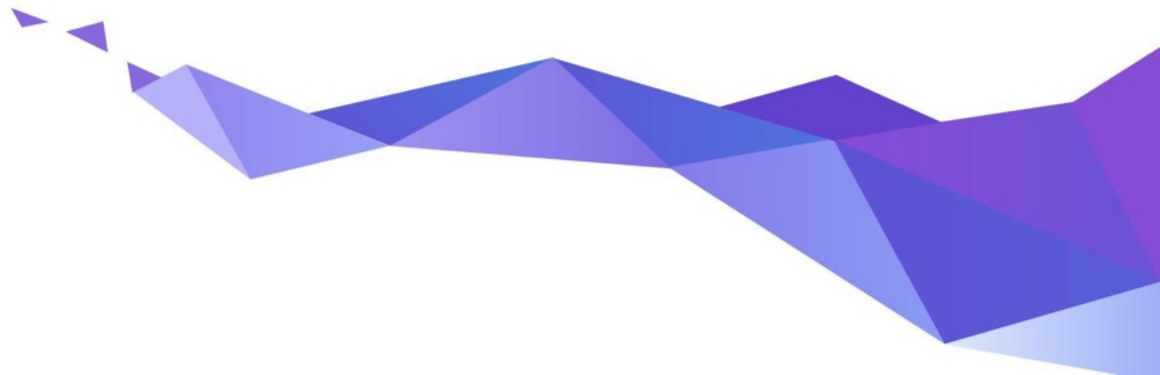
Roadmap - 3.4 release and beyond



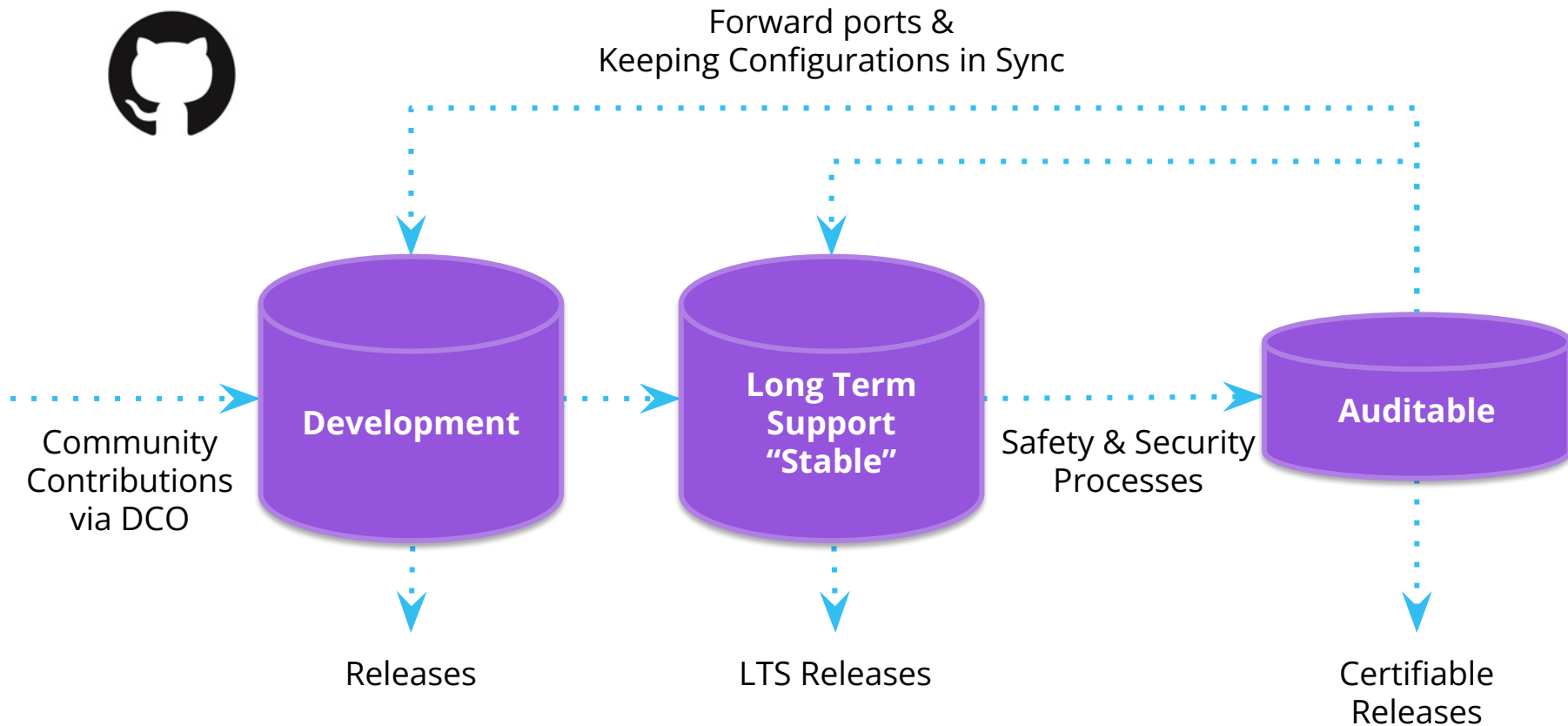
- New Sensor API ([#13718](#))
- Zync synchronization primitives ([#48340](#))
- USB-C improvements ([#38371](#))
- USB device/host enhancements ([#42066](#))
- Replace CivetWeb HTTP server ([#46758](#))
- Better support for multi-core AMP SoCs ([#51833](#))

Note: Zephyr is an open-source project, and its roadmap is subject to change.

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 a collage of four GitHub release pages for the Zephyr project, illustrating the Long Term Support (LTS) for version 1.14. The pages are arranged in a slightly overlapping manner, showing the progression of updates and security fixes.

- Zephyr 1.14.0:** Released on Apr 16 - 5128 commits to master since this release. Major enhancements include support for 160 different board configurations, a reworked and reimplemented timing subsystem, and the addition of a new CPU affinity API.
- Zephyr 1.14.1:** Released 26 days ago - 5126 commits to master since this release. This is an LTS maintenance release with fixes, including a Bluetooth qualification listing and a security vulnerability fix (CVE-2019-8506) related to the Bluetooth BR/EDR specification.
- Zephyr 1.14.2 (Maintenance Release):** Released 25 days ago - 11296 commits to master since this release. This is an LTS maintenance release with fixes, addressing several security vulnerabilities (CVEs) and fixing issues.
- Zephyr v1.14.3:** Released 23 days ago. This is an LTS maintenance release with fixes, addressing security vulnerabilities (CVEs) and fixing issues.

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



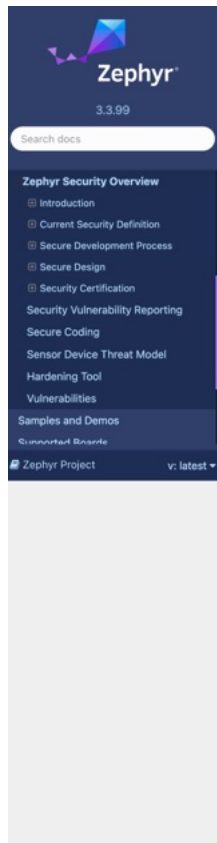
Building in Safety for LTS

- Safety Committee established in 2019
 - Community that understands safety considerations & implications.
 - Representatives from Parasoft, Intel, Synopsys, Google, NXP, Baumer, etc.
- Initial target is IEC 61508 SIL 3 / SC 3 (IEC 61508-3, 7.4.2.12, Route 3s) for a limited scope.
- Multiple safety activities in progress to establish **safety plan, coding guideline compliance, traceability, requirements, test coverage, tooling**, etc.
 - LTS 2 is starting point for Auditable code base
 - Zephyr Project Coding Guidelines based on MISRAC:2021
- Engagement with FSM and certification authority.

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



[Docs / Latest » Security » Zephyr Security Overview](#)

[Open on GitHub](#)

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

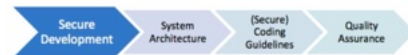
This document outlines the steps of the Zephyr Security Subcommittee towards a defined security process that helps developers build more secure software while addressing security compliance requirements. It presents the key ideas of the security process and outlines which documents need to be created. 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 is created, threats are identified, and countermeasures designed. Their correct implementation and the validity of the threat models are checked by code reviews. Finally, a process shall be defined for reporting, classifying, and mitigating security issues.
3. **Security Certification:** Defines the certifiable part of the Zephyr RTOS. This includes an evaluation target, its assets, and how these assets are protected. Certification claims shall be determined and backed with appropriate evidence.



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  
[...]
```

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**
- **Goal**: Zephyr to **fix issues within 30 days** to give vendors 60 days before publication of vulnerability
- Product makers can register to receive these alerts for free by signing up at Vulnerability Alert Registry

The screenshot shows a web page with a blue header containing the Zephyr logo and a hamburger menu icon. The main content area has a white background and is titled "Criteria for Participation" in purple. It lists four requirements, each preceded by a blue checkmark icon. Below the list, there is a "Removal" section and a link to "Security" for more information.

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 publicly listed product based on some release of Zephyr.
- ✓ Have an actively monitored security email alias.
- ✓ Accept the Zephyr Embargo Policy that is outlined below.

Removal: 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](#).

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.

AMNESIA:33
Research Report Executive Summary

150+ VENDORS AFFECTED

- Forescout Research Labs has launched **Project Memoria**, an initiative that aims at providing the community with the **largest study on the security of TCP/IP stacks**. Project Memoria's goal is to develop the understanding of common bugs behind the vulnerabilities in TCP/IP stacks, identifying the threats they pose to the extended enterprise and how to mitigate those.
- **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 for internet-connected devices, as a pivot point for lateral movement, as a persistence point on the target network or 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.

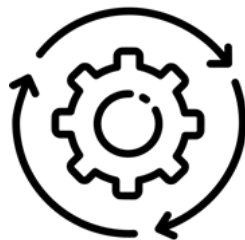
forescout.com/amnesia33 | research@forescout.com | tel: +1-866-377-8771

Zephyr Security Summary



[Documented secure coding practices](#)

Vulnerability response criteria publicly documented

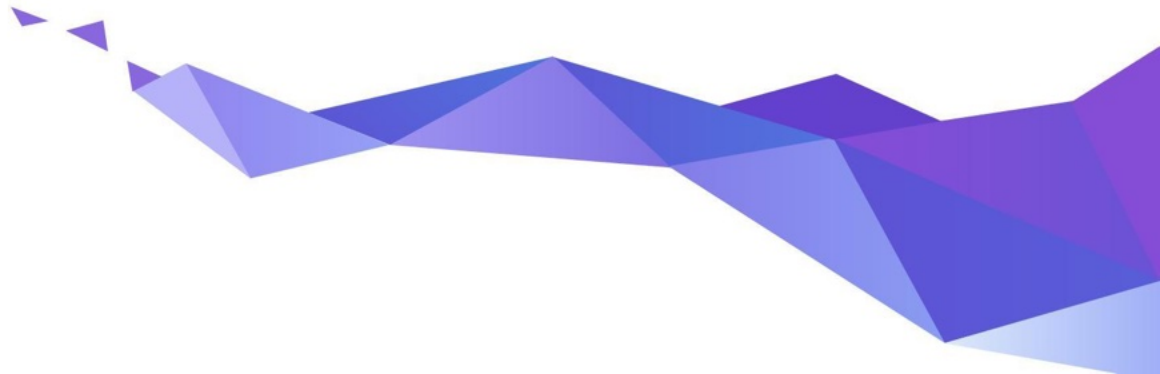


Weekly Coverity scans
MISRA scans



SBOM generation

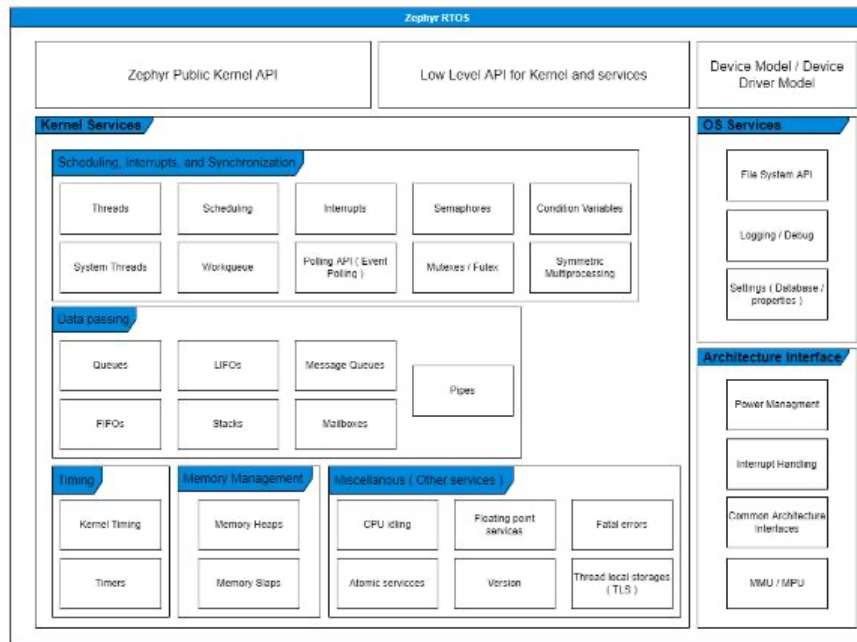
Certification



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)
- x86 and ARM is initial focus
- Scope will be **extended** to include **additional components** as determined by the safety committee



Safety Collateral Proposal



Draft (pending approval by Certification Authority)				
Phase	Assumed Collateral	Type of Doc	Owner	Sharing Model
Safety Concept	Safety Plan and Safety Assessment Plan	Plan/Process	FSM	Platinum
	Verification / Validation / Integration Test Plans	Plan/Process	Testing WG	Public
	Software Development Plan	Plan/Process	TSC	Public
	Configuration and Change Management Plans	Plan/Process	TSC	Public
	Software Architecture and Module Design Specification	Plan/Process	TSC	Public
	Coding Guideline	Plan/Process	TSC	Public
	Tools Documentation	Plan/Process	TSC	Public
	Software Requirements	Code	TSC	Public
	Software Safety Requirements Specification	Result Artifact	Safety WG	Platinum
Detailed Test Phase	Tests (Integration, Arch / Module, Validation)	Code	TSC	Public
	Code Review Report	Result Artifact	Safety WG	Platinum
	Verification / Validation / Integration Test Reports	Result Artifact	Testing WG	Platinum
	Fault Injection Test Report	Result Artifact	Testing WG	Platinum
	Tools Classification	Result Artifact	Safety WG	Platinum
	Tools Validation	Result Artifact	Safety WG	Platinum
	Traceability Report	Result Artifact	Testing WG/FSM	Platinum
	Test Coverage Report	Result Artifact	Testing WG/FSM	Platinum
	Coding Guideline Compliance Report	Result Artifact	Safety WG	Platinum
	Safety Analysis (e.g., FMEA)	Result Artifact	FSM	Platinum
	Source Code	Code	TSC	Public
	Software User Manual	Result Artifact	TSC	Platinum
	Safety Manual	Result Artifact	FSM	Platinum

Silver members have limited access, restricted use to Platinum artifacts based on participation

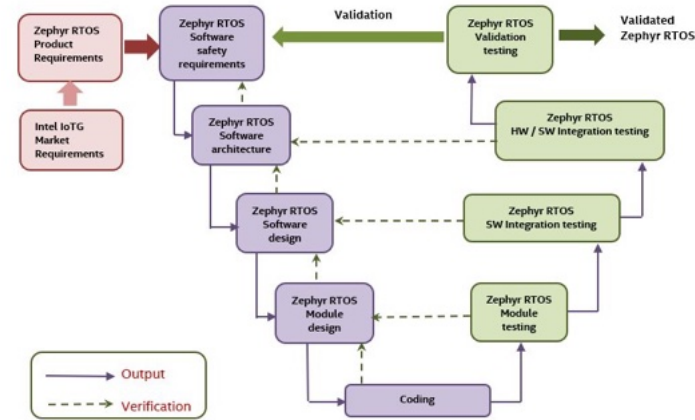
Compliant Development: V-model



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

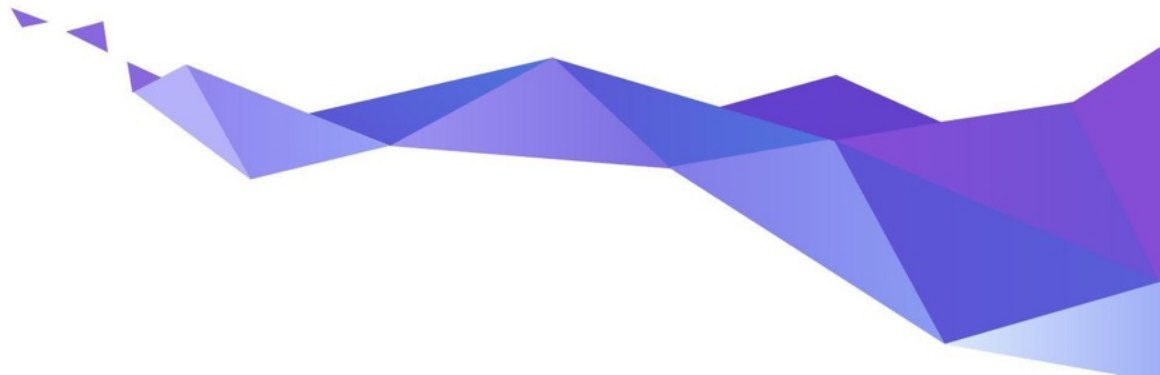
- Specification of features
- Comprehensive documentation
- Traceability from requirements to source code
- Number of committers and information known about them

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

Ecosystem & Governance



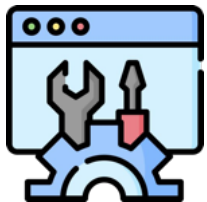
Zephyr Project: Platinum Members



Zephyr Project: Silver Members



Vibrant Ecosystem



Development Tools



Governing Board

Technical Steering Committee

Contributors



Applications & Middlewares



Training & Consulting



Firmwares & Libraries

Ecosystem // Dev Tools



Development Tools



Training & Consulting



Firmwares & Libraries



Applications & Middlewares

IDE



Compilers



Debuggers / Tracing Tools



Emulation / Simulation



Ecosystem // Training & Consulting



Training



Development Tools



Training & Consulting



Firmwares & Libraries



Applications & Middlewares

Services & Consulting



Ecosystem // Firmwares & Libraries



Development Tools



Training & Consulting

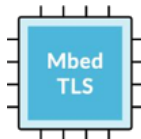


Firmwares & Libraries



Applications & Middlewares

Security



TinyML



Language runtimes



Others



Ecosystem // Apps & Middlewares



Remote Management



Robotics



Development Tools



Training & Consulting

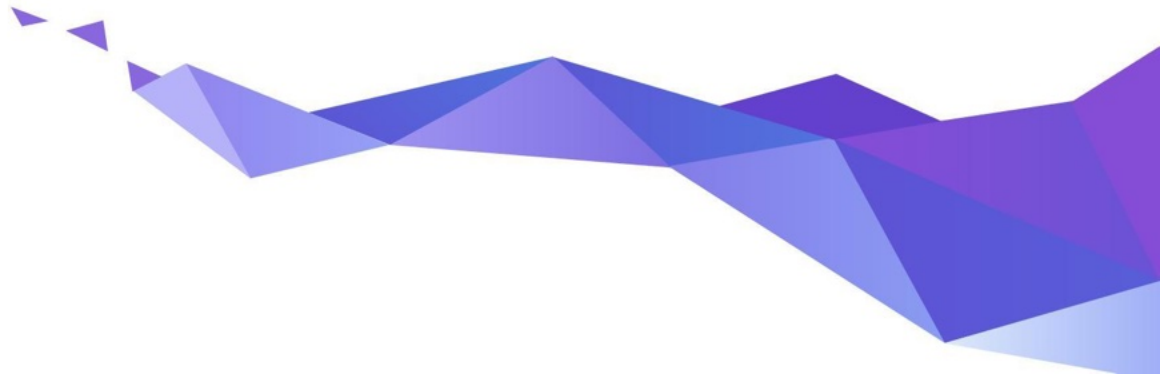


Firmwares & Libraries

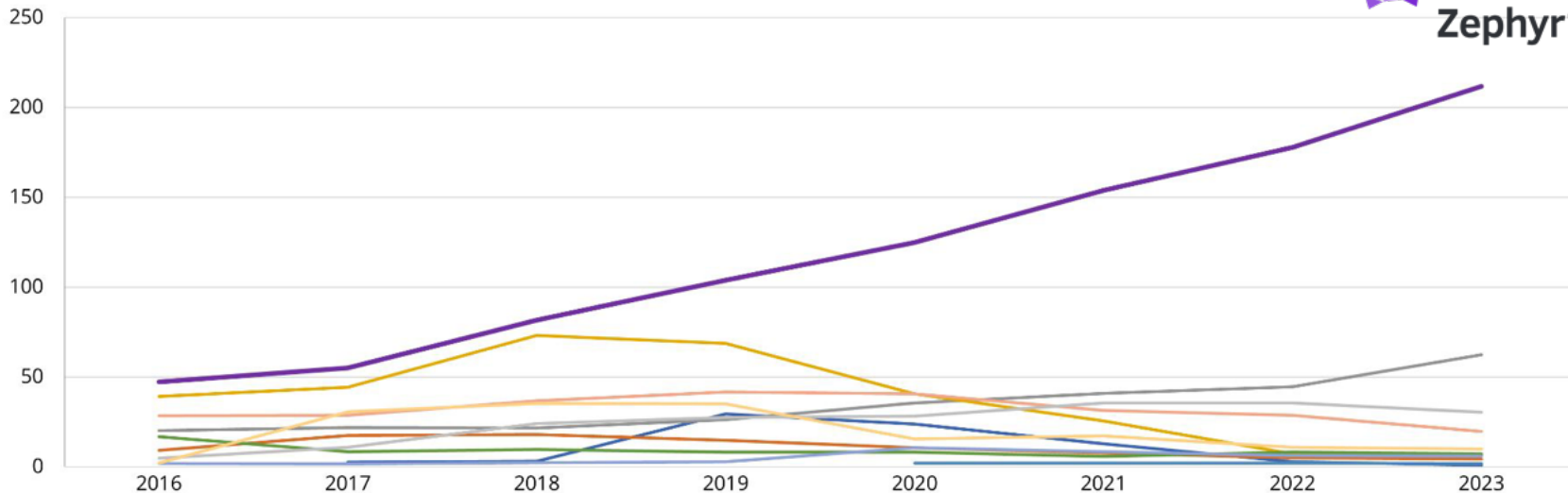


Applications & Middlewares

Zephyr in the RTOS landscape

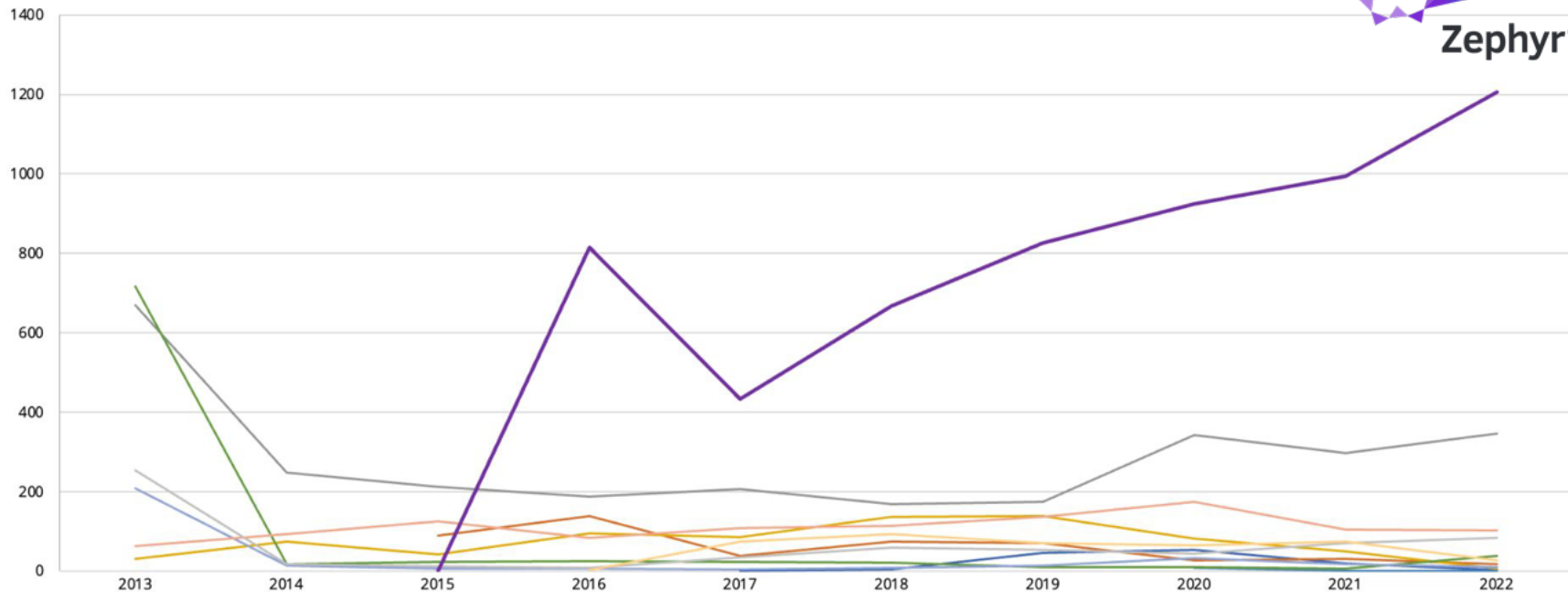


Average Number of Unique Contributors per Month



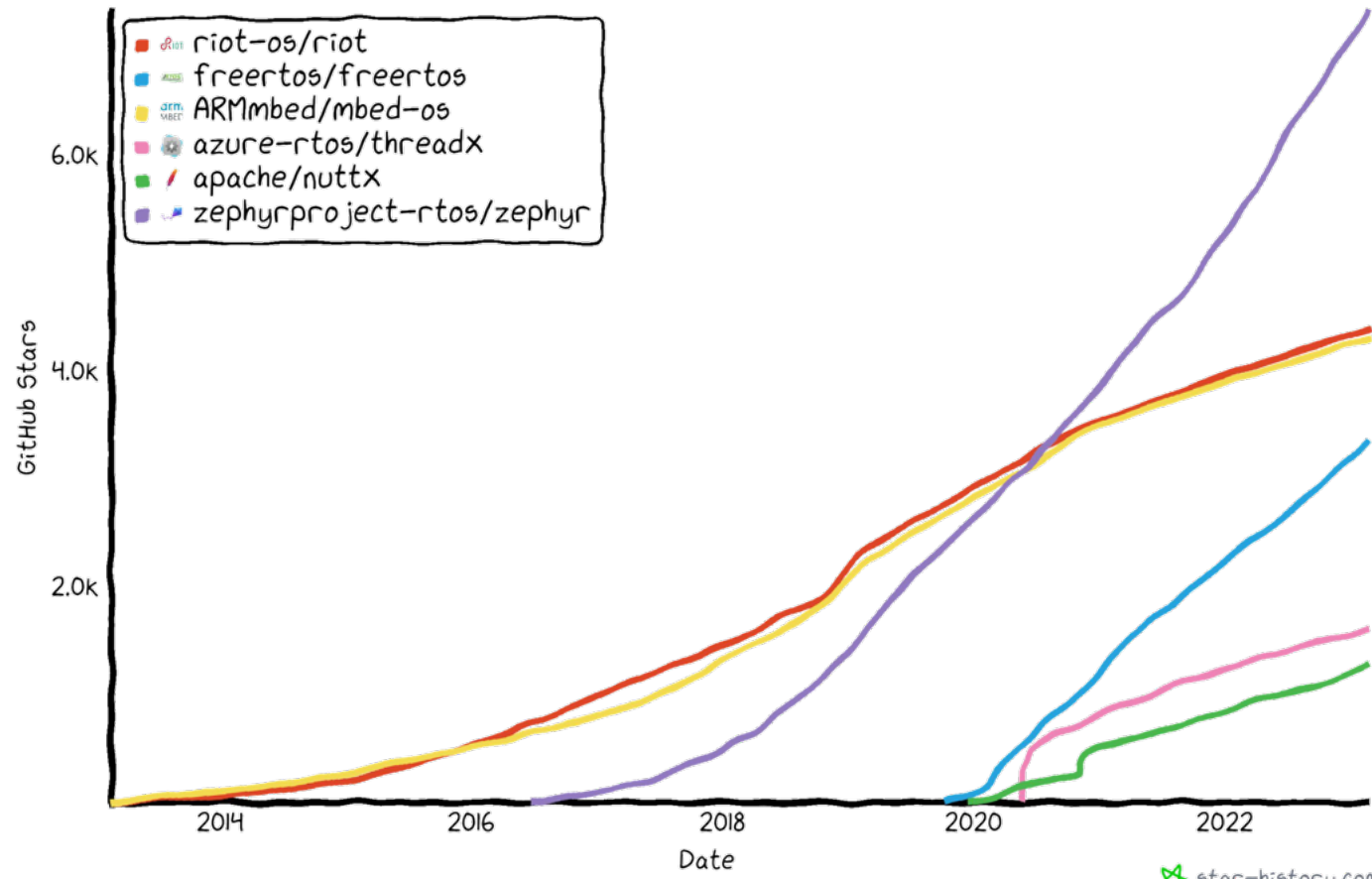
	2016	2017	2018	2019	2020	2021	2022	2023
Amazon FreeRTOS		3	3	30	24	13	3	1
Apache Mynewt	9	18	18	15	11	8	5	4
Apache NuttX	20	22	22	26	36	41	45	62
Arm Mbed OS	39	44	73	69	41	26	7	7
Azure RTOS ThreadX					2	2	2	2
Contiki-NG	17	9	10	8	8	6	8	7
FreeRTOS	2	2	2	3	11	8	6	6
RIOT OS	29	29	37	42	41	31	29	20
RT-Thread	5	11	24	28	28	36	36	30
TizenRT	2	31	35	35	16	17	11	10
Zephyr	47	55	82	104	125	154	178	212

Average Number of Commits per Month

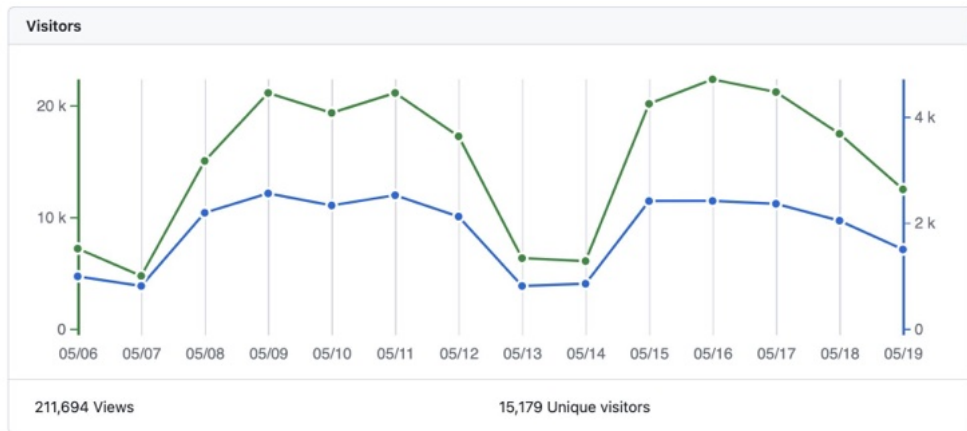


OS	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
Amazon FreeRTOS					2	4	47	53	20	2
Apache Mynewt			90	138	38	74	70	27	31	18
Apache NuttX	670	248	212	187	206	169	174	343	297	347
Arm Mbed OS	30	74	42	95	86	136	138	82	51	6
Azure RTOS ThreadX								7	1	2
Contiki-NG	717	17	23	25	23	22	9	11	7	38
FreeRTOS	209	13	6	6	4	8	13	32	17	11
RIOT OS	63	93	126	84	108	115	136	175	105	103
RT-Thread	253	18	13	9	35	60	53	43	70	84
TizenRT				2	73	93	71	64	74	27
Zephyr			0	814	434	667	825	924	995	1206

GitHub Stars History



GitHub Clones & Unique Visitors



2023-05-06 → 2023-05-19

~476 unique clones per day
~1084 unique visitors per day

Zephyr Participation Information



zephyrproject.org



github.com/zephyrproject-rtos



lists.zephyrproject.org



chat.zephyrproject.org



zephyrproject.org

