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 *yet* SCALABLE

< 8KB Flash

< 5KB RAM

from small sensor nodes

... to complex multi-core systems

FLEXIBLE *yet* SECURE

Heavily customizable

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

Built with safety & security in mind

Certification-ready

Long-term Support

Zephyr®

OPEN-SOURCE

Permissively licensed (Apache 2.0)

Vendor-neutral governance

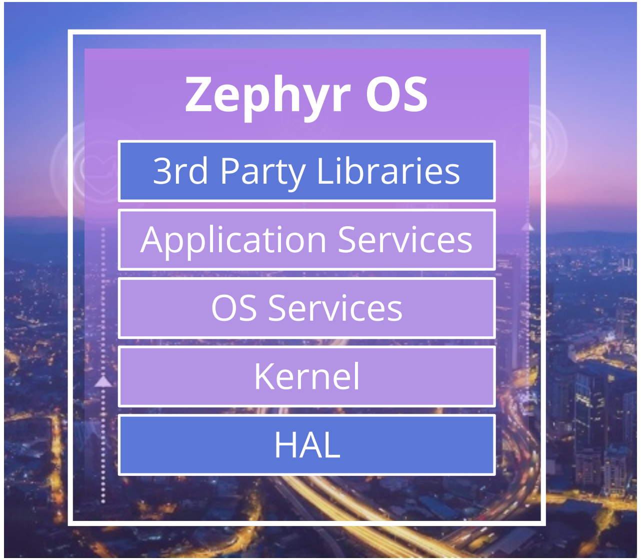
ECOSYSTEM

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

Proglove

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

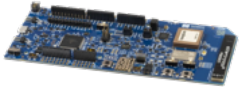
zephyrproject.org/products-running-zephyr

450+ supported boards... and growing

Arduino Portenta H7

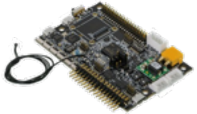
ESP32

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
adxl345
adxl362
adxl372
ak8975
amg88xx
ams_as5600
ams_iAQcore
apds9960
bma280
bmc150_magn
bme280
bme680
bmg160
bmi160
bmi270
bmm150
bmp388
bq274xx
ccs811

dht
dps310
ds18b20
ens210
esp32_temp
fdc2x1x
fxas21002
fxos8700
grove
grow_r502a
hmc5883l
hp206c
hts221
i3g4250d
icm42605
icm42670
icm42688
icp10125
iis2dh
iis2dlpc

iis2iclx
iis2mdc
iis3dhhc
ina219
ina23x
isl29035
ism330dhcx
ite_tach_it8xxx2
ite_vcmp_it8xxx2
lis2dh
lis2ds12
lis2dw12
lis2mdl
lis3mdl
lm75
lm77
lps22hb
lps22hh
lps25hb
lsm303dlhc_magn

lsm6ds0
lsm6dsl
lsm6dso
lsm9ds0_gyro
lsm9ds0_mfd
max17055
max17262
max30101
max31875
max44009
max6675
mchp_tach_xec
mcp9808
mcux_acmp
mhz19b
mpr
mpu6050
mpu9250
ms5607
ms5837

nrf5
nuvoton_adc_cmp_npcx
nuvoton_tach_npcx
nxp_kinetis_temp
opt3001
pcnt_esp32
pms7003
qdec_mcux
qdec_nrfx
qdec_sam
qdec_stm32
rpi_pico_temp
sbs_gauge
sgp40
sht3xd
sht4x
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

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

Supported Hardware Architectures

ARC®

arm

Cortex-M, Cortex-R
& Cortex-A

intel®

x86 & x86_64

MIPS

Nios® II
Processor

RISC-V®

32 & 64 bit

SPARC

tensilica

Xtensa

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

Vibrant Ecosystem

Zephyr®

- Governing Board
- Technical Steering Committee
- Contributors

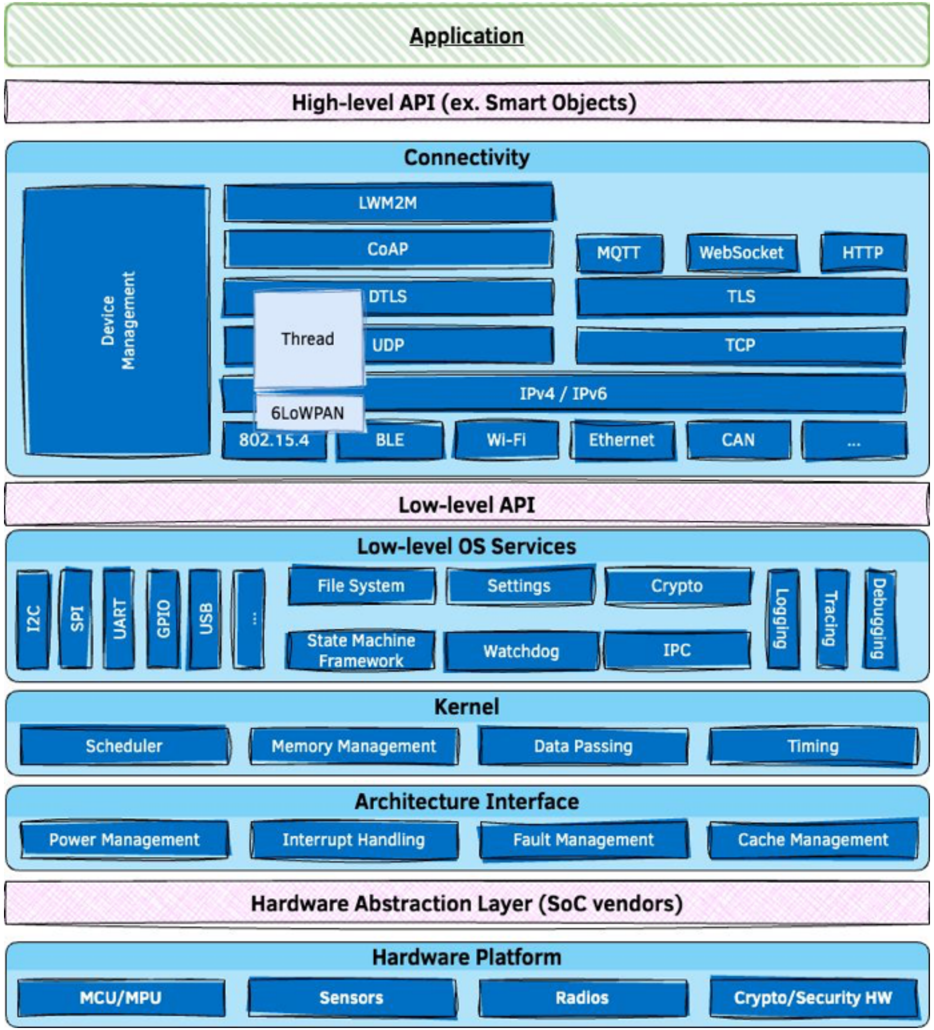
Development Tools

Applications & Middlewares

Training & Consulting

Firmwares & Libraries

Architecture



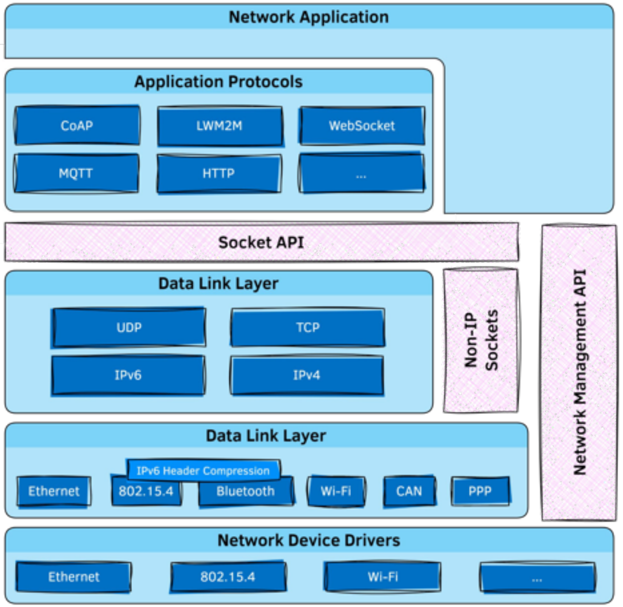
Diving into Zephyr’s features

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, SNTP
- 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

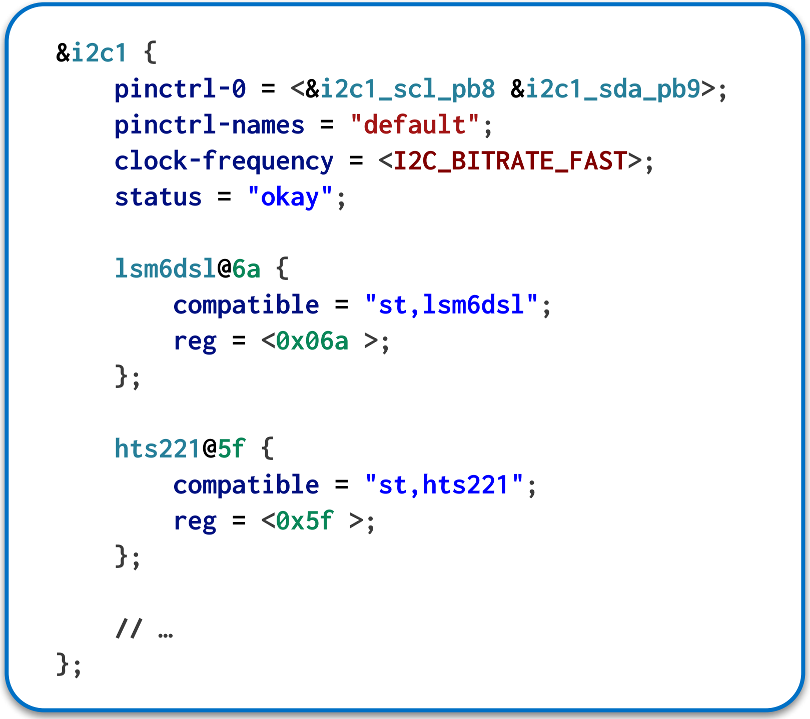
- Goal: use as little power as possible
- Cross-platform (architecture / SoC agnostic)
- Tickless scheduler
- Handled by the kernel / Customizable by the user

Devicetree

Describe & **configure** the available hardware on the target system

Decouple the application from the hardware

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

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

ℹ 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

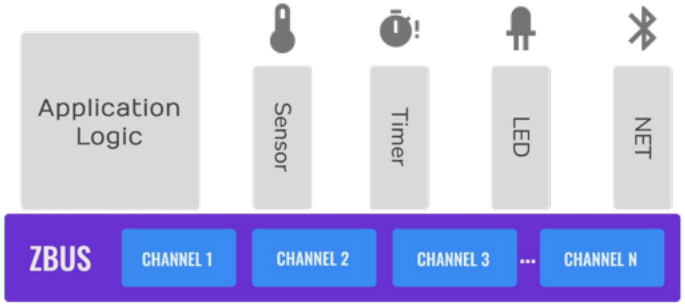
- Drivers available for various types of displays
  - LCD
  - OLED
  - Touch panel displays
  - E-ink
- LVGL integration
- Support for video capture and output

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

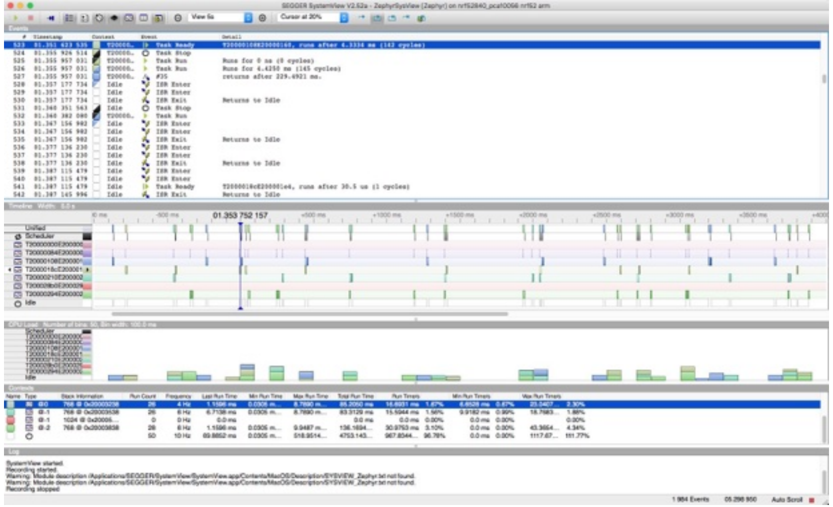
Data passing objects available in Zephyr kernel



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.)



SEGGER SystemView V2.52a - ZephyrSysView (Zephyr) on nrf52840_pca10056 nrf52 arm

```
  #  Timestamp       Context    Event       Detail
523  01.351 623 535  T20000..   Task Ready  T20000108E20000160, runs after 4.3334 ms (142 cycles)
524  01.355 926 514  T20000..   Task Stop
525  01.355 957 031  T20000..   Task Run    Runs for 0 ms (0 cycles)
526  01.355 957 031  T20000..   Task Run    Runs for 4.4250 ms (145 cycles)
527  01.355 957 031  T20000..   #35         returns after 229.4921 ms.
528  01.357 177 734  Idle       ISR Enter
529  01.357 177 734  Idle       ISR Enter
530  01.357 177 734  Idle       ISR Exit    Returns to Idle
531  01.360 351 563  Idle       Task Stop
532  01.360 382 080  T20000..   Task Run
533  01.367 156 982  Idle       ISR Enter
534  01.367 156 982  Idle       ISR Enter
535  01.367 156 982  Idle       ISR Exit    Returns to Idle
536  01.377 136 230  Idle       ISR Enter
537  01.377 136 230  Idle       ISR Enter
538  01.377 136 230  Idle       ISR Exit    Returns to Idle
539  01.387 115 479  Idle       ISR Enter
540  01.387 115 479  Idle       ISR Enter
541  01.387 115 479  Idle       Task Ready  T2000018cE20001e4, runs after 30.5 us (1 cycles)
542  01.387 145 996  Idle       ISR Exit    Returns to Idle
```

```
Name Type  Stack Information     Run Count  Frequency  Last Run Time  Min Run Time  Max Run Time  Total Run Time  Run Time/s          Min Run Time/s     Max Run Time/s
     @0    768 @ 0x20003238      26         4 Hz       1.1596 ms      0.0305 m...   8.7890 m...   85.2050 ms      16.6931 ms  1.67%   6.6528 ms  0.67%   23.0407...  2.30%
     @-1   768 @ 0x20003538      26         6 Hz       6.7138 ms      0.0305 m...   8.7890 m...   83.3129 ms      15.5044 ms  1.56%   9.9182 ms  0.99%   18.7583...  1.88%
     @-1   1024 @ 0x20005...     0          0 Hz       0.0 ms                                     0.0 ms          0.0 ms  0.00%       0.0 ms  0.00%                  0.00%
     @-2   768 @ 0x20003838      28         6 Hz       1.1596 ms      0.0305 m...   9.9487 m...   136.1694...     30.9753 ms  3.10%   0.0 ms  0.00%      43.3654...  4.34%
                                 50         10 Hz      69.8852 ms     0.0305 m...   518.9514...   4753.143...     967.8344...  96.78%  0.0 ms  0.00%      1117.67...  111.77%
```

```
SystemView started.
Recording started.
Warning: Module description /Applications/SEGGER/SystemView/SystemView.app/Contents/MacOS/Description/SYSVIEW_Zephyr.txt not found.
Warning: Module description /Applications/SEGGER/SystemView/SystemView.app/Contents/MacOS/Description/SYSVIEW_Zephyr.txt not found.
Recording stopped
```

Zephyr 3.4 (June 2023) – What's new?

- **New peripherals**

Auxiliary displays

NVMe disks & controllers

Retained memory

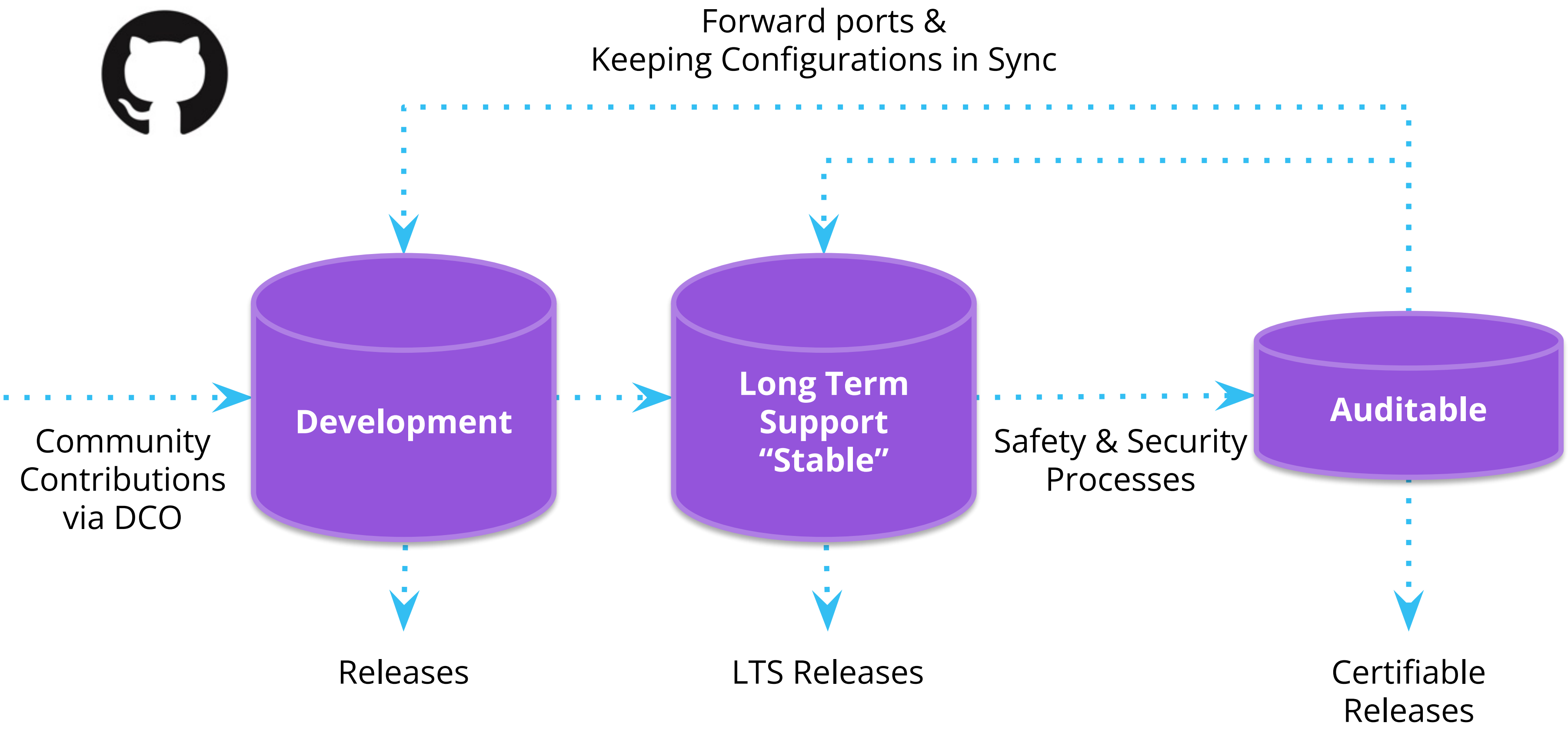
SMBus

Real-time clocks (RTC)

- **Twister improvements** (pyTest, Robot Framework, gTest)
- **Barrier API**
- **Snippets** … and more, see Release notes 3.4.

Safety & Security

Code Repositories

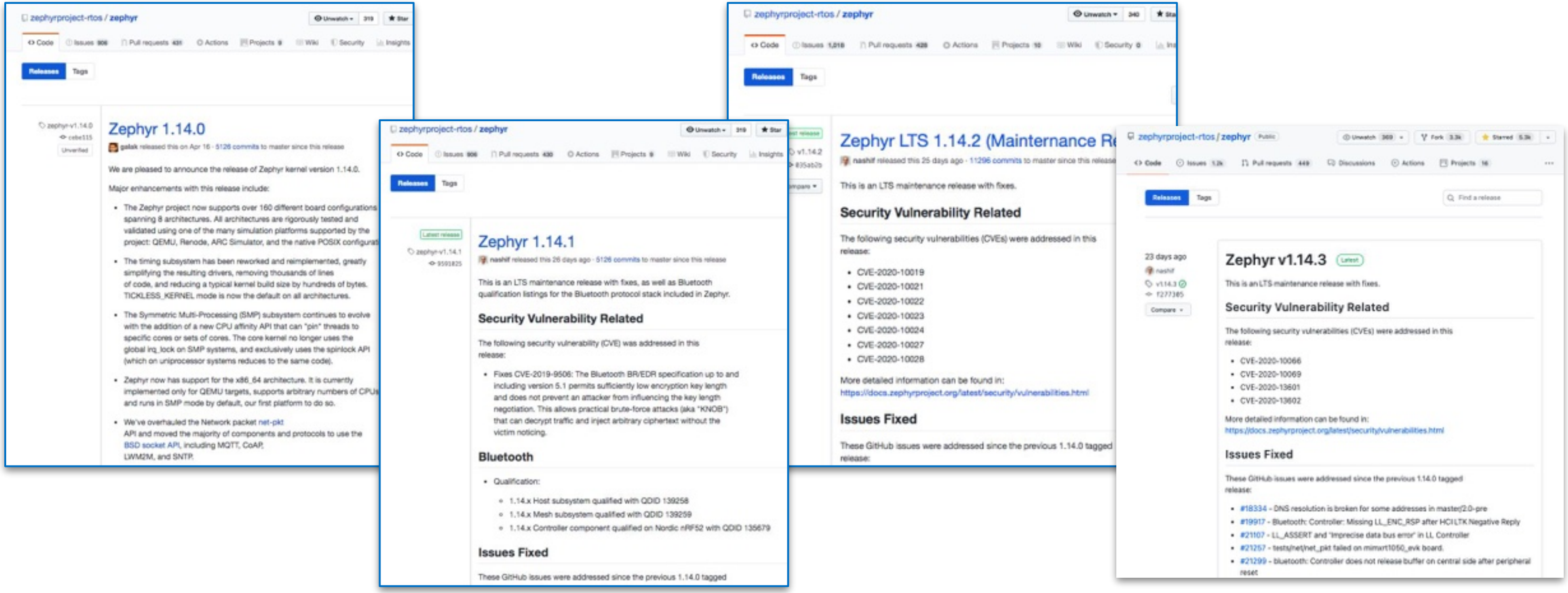


Long Term Support (Zephyr 2.7.x)

- **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**

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

Long Term Support (LTS - 1.14)



Zephyr 1.14.0

galak released this on Apr 16 · 5126 commits to master since this release

We are pleased to announce the release of Zephyr kernel version 1.14.0.

Major enhancements with this release include:

- The Zephyr project now supports over 160 different board configurations spanning 8 architectures. All architectures are rigorously tested and validated using one of the many simulation platforms supported by the project: QEMU, Renode, ARC Simulator, and the native POSIX configurat
- The timing subsystem has been reworked and reimplemented, greatly simplifying the resulting drivers, removing thousands of lines of code, and reducing a typical kernel build size by hundreds of bytes. TICKLESS_KERNEL mode is now the default on all architectures.
- The Symmetric Multi-Processing (SMP) subsystem continues to evolve with the addition of a new CPU affinity API that can "pin" threads to specific cores or sets of cores. The core kernel no longer uses the global irq_lock on SMP systems, and exclusively uses the spinlock API (which on uniprocessor systems reduces to the same code).
- Zephyr now has support for the x86_64 architecture. It is currently implemented only for QEMU targets, supports arbitrary numbers of CPUs and runs in SMP mode by default, our first platform to do so.
- We've overhauled the Network packet net-pkt API and moved the majority of components and protocols to use the BSD socket API, including MQTT, CoAP, LWM2M, and SNTP.

Zephyr 1.14.1

nashif released this 26 days ago · 5126 commits to master since this release

This is an LTS maintenance release with fixes, as well as Bluetooth qualification listings for the Bluetooth protocol stack included in Zephyr.

Security Vulnerability Related

The following security vulnerability (CVE) was addressed in this release:

- Fixes CVE-2019-9506: The Bluetooth BR/EDR specification up to and including version 5.1 permits sufficiently low encryption key length and does not prevent an attacker from influencing the key length negotiation. This allows practical brute-force attacks (aka "KNOB") that can decrypt traffic and inject arbitrary ciphertext without the victim noticing.

Bluetooth

- Qualification:
  - 1.14.x Host subsystem qualified with QDID 139258
  - 1.14.x Mesh subsystem qualified with QDID 139259
  - 1.14.x Controller component qualified on Nordic nRF52 with QDID 135679

Issues Fixed

These GitHub issues were addressed since the previous 1.14.0 tagged

Zephyr LTS 1.14.2 (Maintenance Re

nashif released this 25 days ago · 11296 commits to master since this release

This is an LTS maintenance release with fixes.

Security Vulnerability Related

The following security vulnerabilities (CVEs) were addressed in this release:

- CVE-2020-10019
- CVE-2020-10021
- CVE-2020-10022
- CVE-2020-10023
- CVE-2020-10024
- CVE-2020-10027
- CVE-2020-10028

More detailed information can be found in:
https://docs.zephyrproject.org/latest/security/vulnerabilities.html

Issues Fixed

These GitHub issues were addressed since the previous 1.14.0 tagged release:

Zephyr v1.14.3 Latest

This is an LTS maintenance release with fixes.

Security Vulnerability Related

The following security vulnerabilities (CVEs) were addressed in this release:

- CVE-2020-10066
- CVE-2020-10069
- CVE-2020-13601
- CVE-2020-13602

More detailed information can be found in:
https://docs.zephyrproject.org/latest/security/vulnerabilities.html

Issues Fixed

These GitHub issues were addressed since the previous 1.14.0 tagged release:

- #18334 - DNS resolution is broken for some addresses in master/2.0-pre
- #19917 - Bluetooth: Controller: Missing LL_ENC_RSP after HCI LTK Negative Reply
- #21107 - LL_ASSERT and "Imprecise data bus error" in LL Controller
- #21257 - tests/net/net_pkt failed on mimxrt1050_evk board.
- #21299 - bluetooth: Controller does not release buffer on central side after peripheral reset

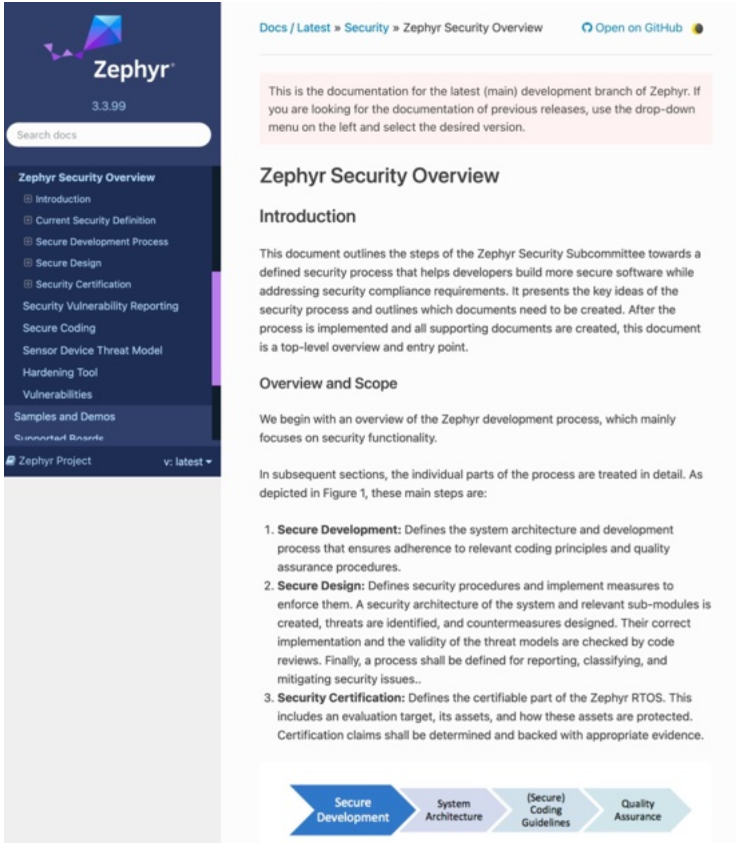
Delivered bug fixes and latest security updates for 2 years!

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



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

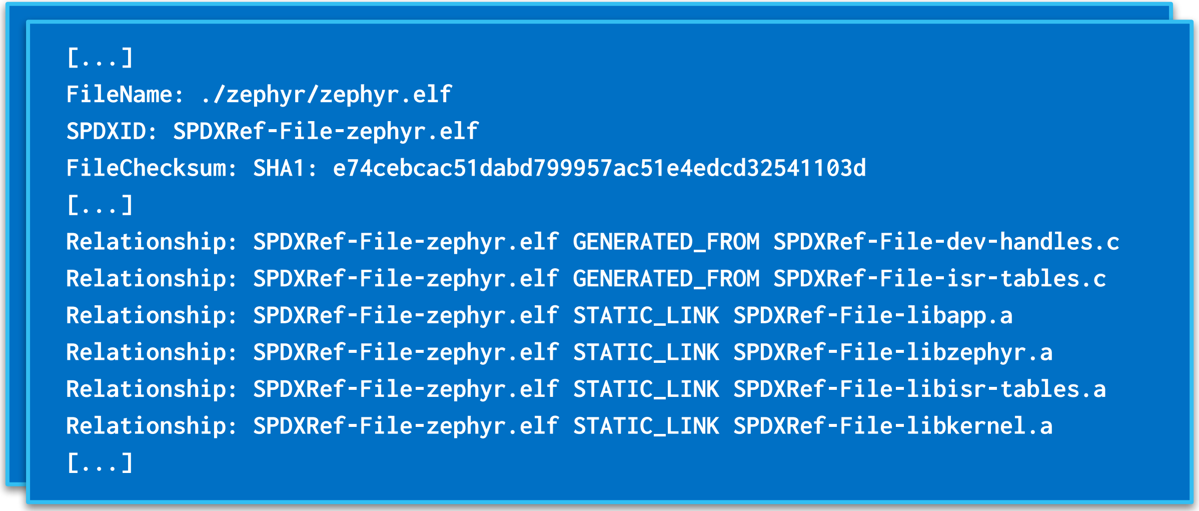
We begin with an overview of the Zephyr development process, which mainly focuses on security functionality.

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

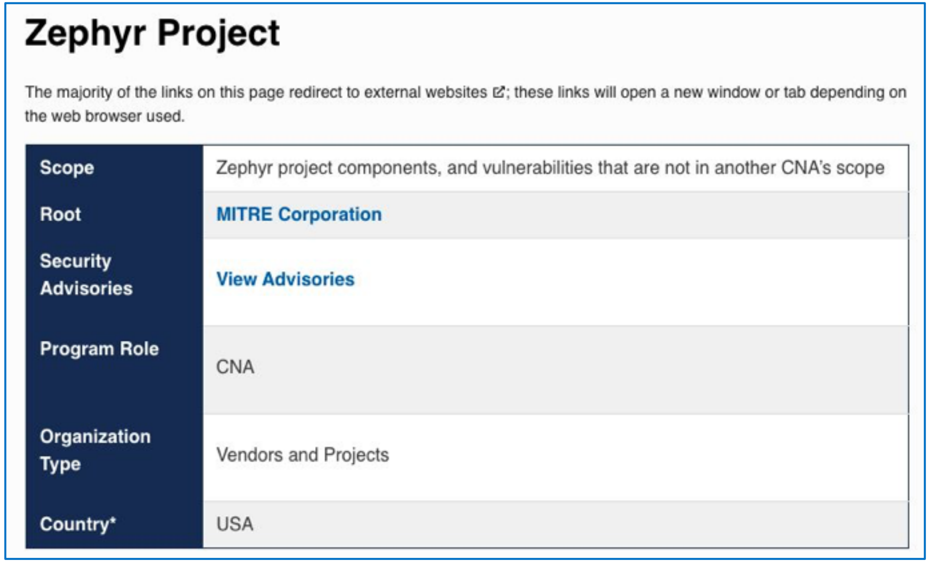
- Zephyr ships an **SBOM** (Software Bill of Materials) with each release
- Downstream consumers can leverage built-in tools to, in turn, generate source & build SBOMs for their deliverables



```
[...]
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.



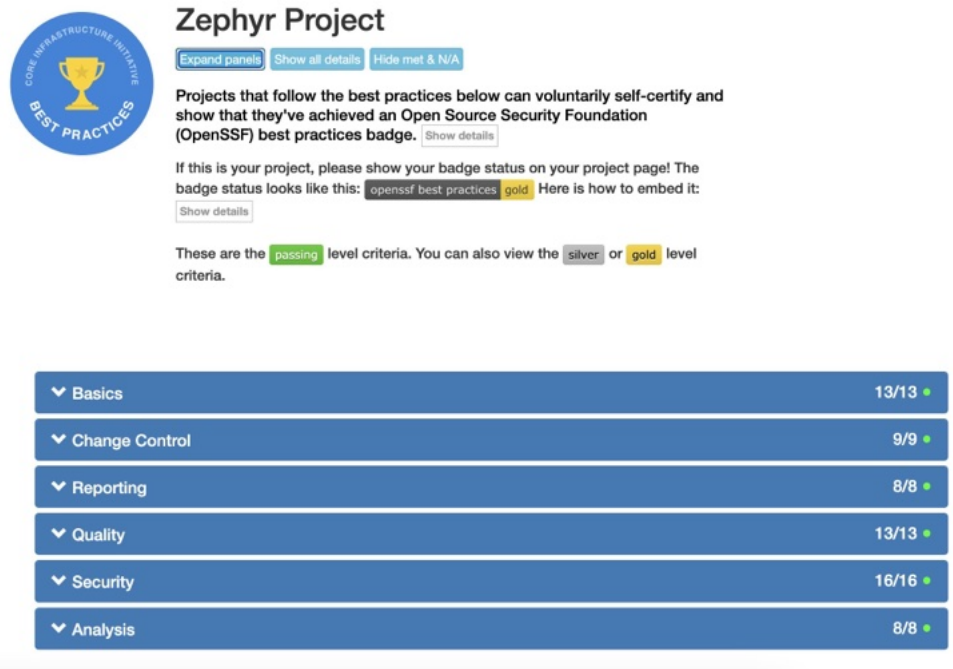
Zephyr Project

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

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.

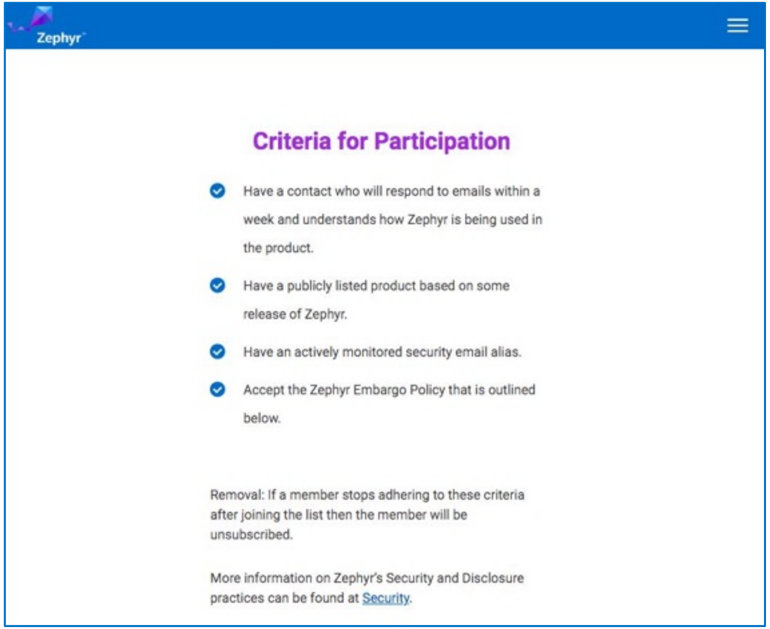
If this is your project, please show your badge status on your project page! The badge status looks like this: openssf best practices gold Here is how to embed it:

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



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.

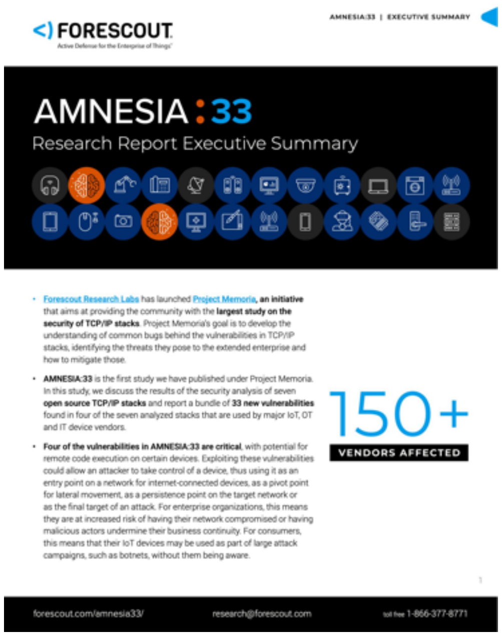
Zephyr PSIRT: Remediation and Response

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.



<)FORESCOUT
Active Defense for the Enterprise of Things

AMNESIA:33 | EXECUTIVE SUMMARY

AMNESIA:33
Research Report Executive Summary

- 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.

150+
VENDORS AFFECTED

1

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

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

Zephyr Security Summary

Documented secure coding practices

Vulnerability response criteria publicly documented

Weekly Coverity scans

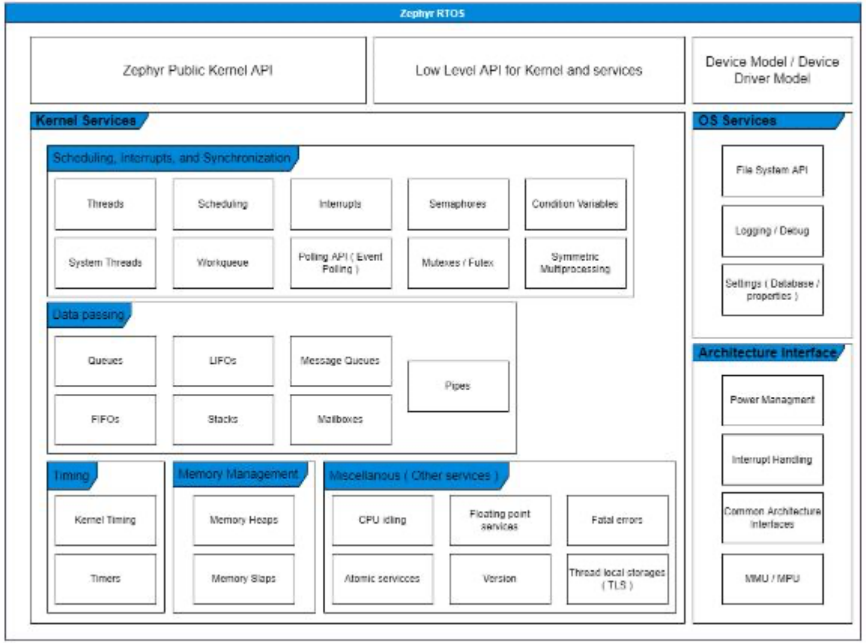
MISRA scans

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



Zephyr RTOS

Zephyr Public Kernel API

Low Level API for Kernel and services

Device Model / Device Driver Model

Kernel Services

Scheduling, Interrupts, and Synchronization

Threads

Scheduling

Interrupts

Semaphores

Condition Variables

System Threads

Workqueue

Polling API (Event Polling)

Mutexes / Futex

Symmetric Multiprocessing

Data passing

Queues

LIFOs

Message Queues

Pipes

FIFOs

Stacks

Mailboxes

Timing

Kernel Timing

Timers

Memory Management

Memory Heaps

Memory Slaps

Miscellanous (Other services)

CPU idling

Floating point services

Fatal errors

Atomic servicces

Version

Thread local storages (TLS)

OS Services

File System API

Logging / Debug

Settings (Database / properties)

Architecture Interface

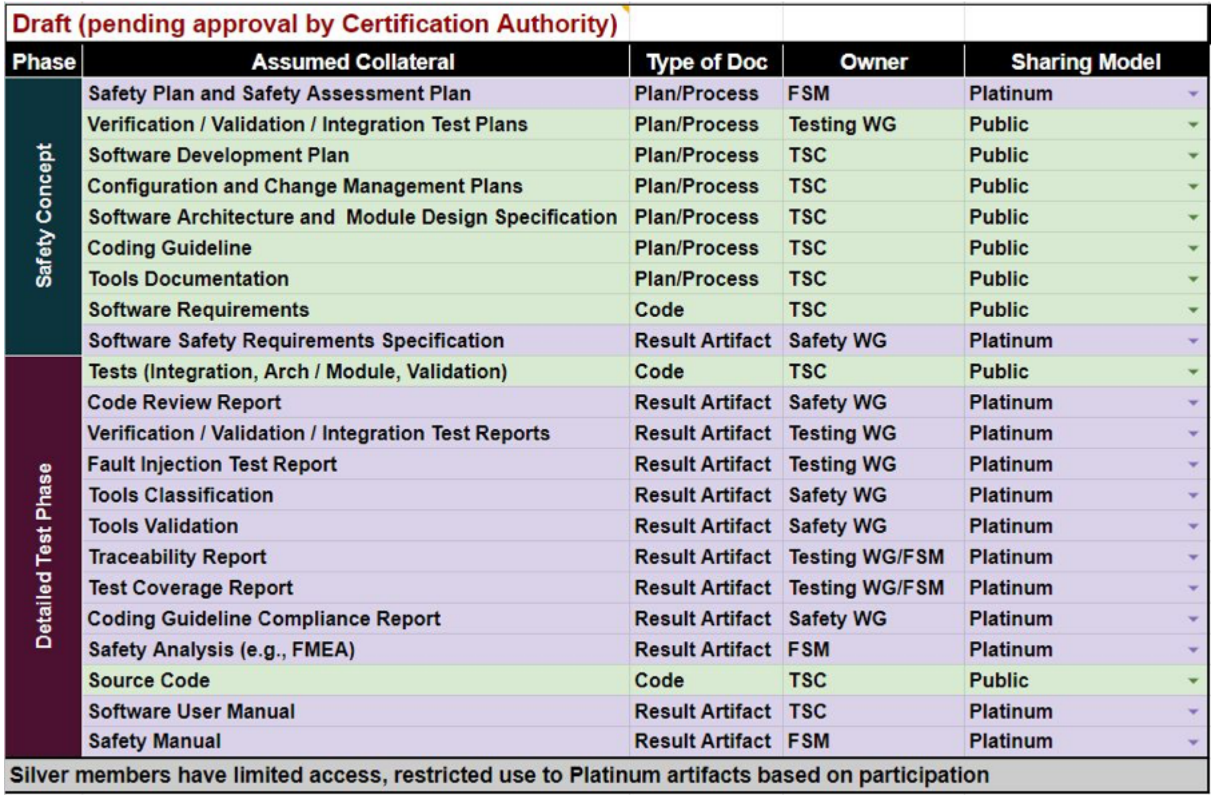
Power Managment

Interrupt Handling

Common Architecture Interfaces

MMU / MPU

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

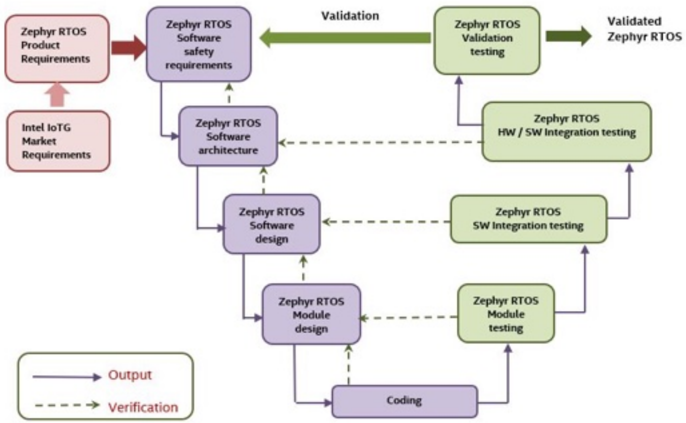
Silver members have limited access, restricted use to Platinum artifacts based on participation

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


Zephyr RTOS Product Requirements
Intel IoTG Market Requirements
Zephyr RTOS Software safety requirements
Validation
Zephyr RTOS Validation testing
Validated Zephyr RTOS
Zephyr RTOS Software architecture
Zephyr RTOS HW / SW Integration testing
Zephyr RTOS Software design
Zephyr RTOS SW Integration testing
Zephyr RTOS Module design
Zephyr RTOS Module testing
Coding
Output
Verification

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

Ecosystem & Governance

Zephyr Project: Platinum Members

Zephyr®

ANALOG DEVICES

antmicro

Baumer

Google

intel®

Meta

NORDIC® SEMICONDUCTOR

NXP

oticon life-changing **technology**

Qualcomm Innovation Center

T Mobile™

ZEISS

Zephyr Project: Silver Members

Zephyr

ARDUINO

AVSYSTEM

Baylibre

blues wireless

Golioth

infineon

irnas

Laird™ CONNECTIVITY

Linaro

Memfault

PARASOFT®

percepio SENSING SOFTWARE

SiFive

SILICON LABS

STERNUM

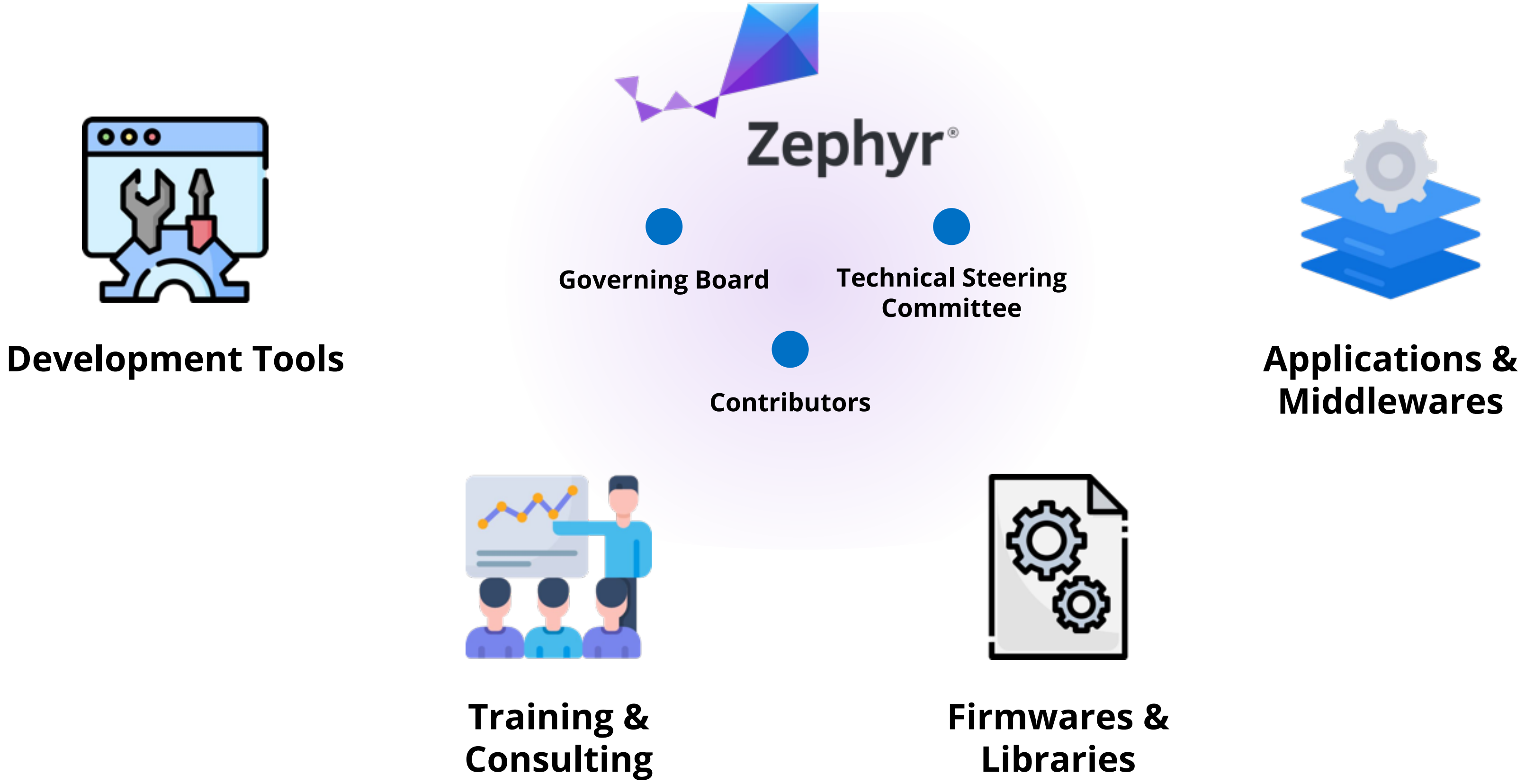
SYNOPSYS®

TEXAS INSTRUMENTS

TII Technology Innovation Institute

WNDRVR

Vibrant Ecosystem



Governing Board

Technical Steering Committee

Contributors

Development Tools

Applications & Middlewares

Training & Consulting

Firmwares & Libraries

Ecosystem // Dev Tools

Development Tools

Training & Consulting

Firmwares & Libraries

Applications & Middlewares

IDE

eclipse

IDE

Compilers

Debuggers / Tracing Tools

LAUTERBACH DEVELOPMENT TOOLS

SEGGER

Memfault

percepio SENSING SOFTWARE

Emulation / Simulation

RENODE™

WOKWi

Ecosystem // **Training & Consulting**

Zephyr

Development Tools

Training & Consulting

Firmwares & Libraries

Applications & Middlewares

Training

NORDIC SEMICONDUCTOR

Golioth

THE LINUX FOUNDATION

percepio SENSING SOFTWARE

Services & Consulting

Baylibre

NXP

Laird CONNECTIVITY

antmicro

WNDRVR

AVSYSTEM

Ecosystem // **Firmwares & Libraries**

Zephyr®

Development Tools

Training & Consulting

Firmwares & Libraries

Applications & Middlewares

Security

Mbed TLS

wolfSSL

TinyML

TensorFlow Lite

EDGE IMPULSE

Language runtimes

MicroPython

JS

R

Others

SOUND OPEN FIRMWARE

zscilib

Memfault

Ecosystem // Apps & Middlewares

Development Tools

Training & Consulting

Firmwares & Libraries

Applications & Middlewares

Remote Management

HERALD

Golioth

AVSYSTEM

Memfault

STERNUM

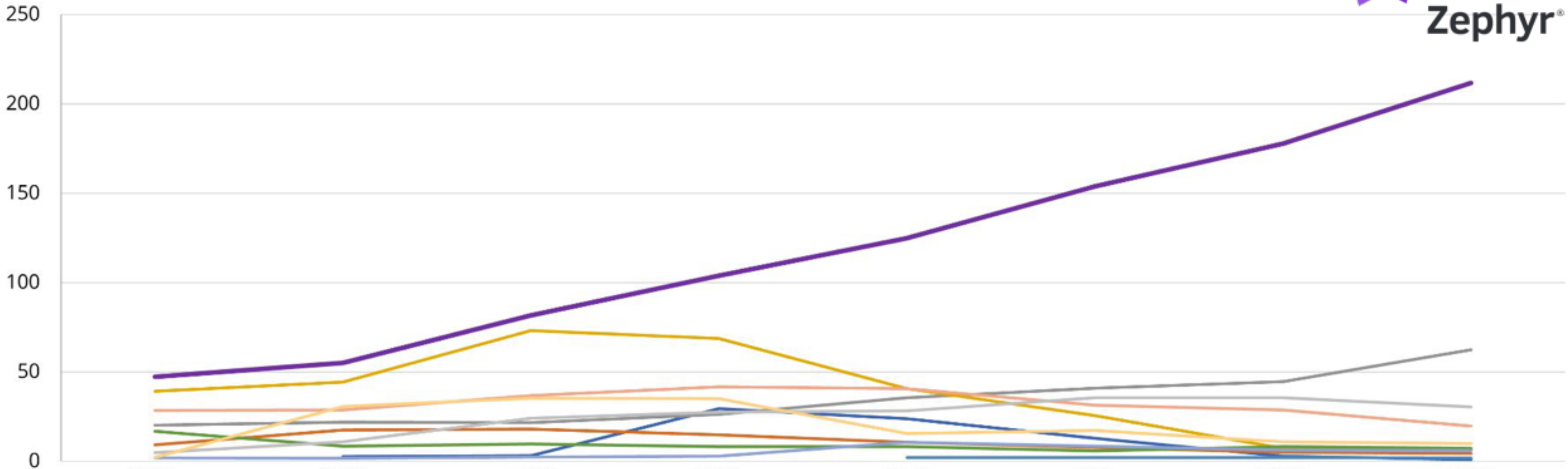
blues wireless

Robotics

ROS

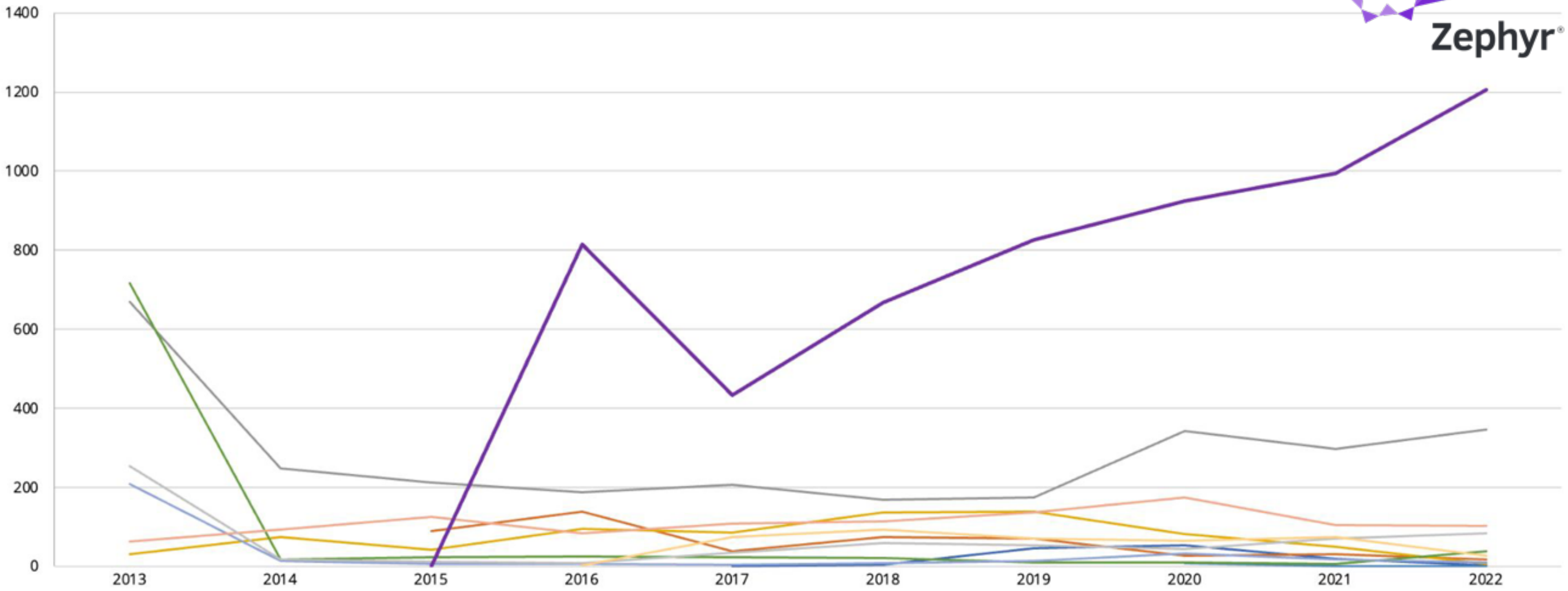
Zephyr in the RTOS landscape

Average Number of Unique Contributors per Month

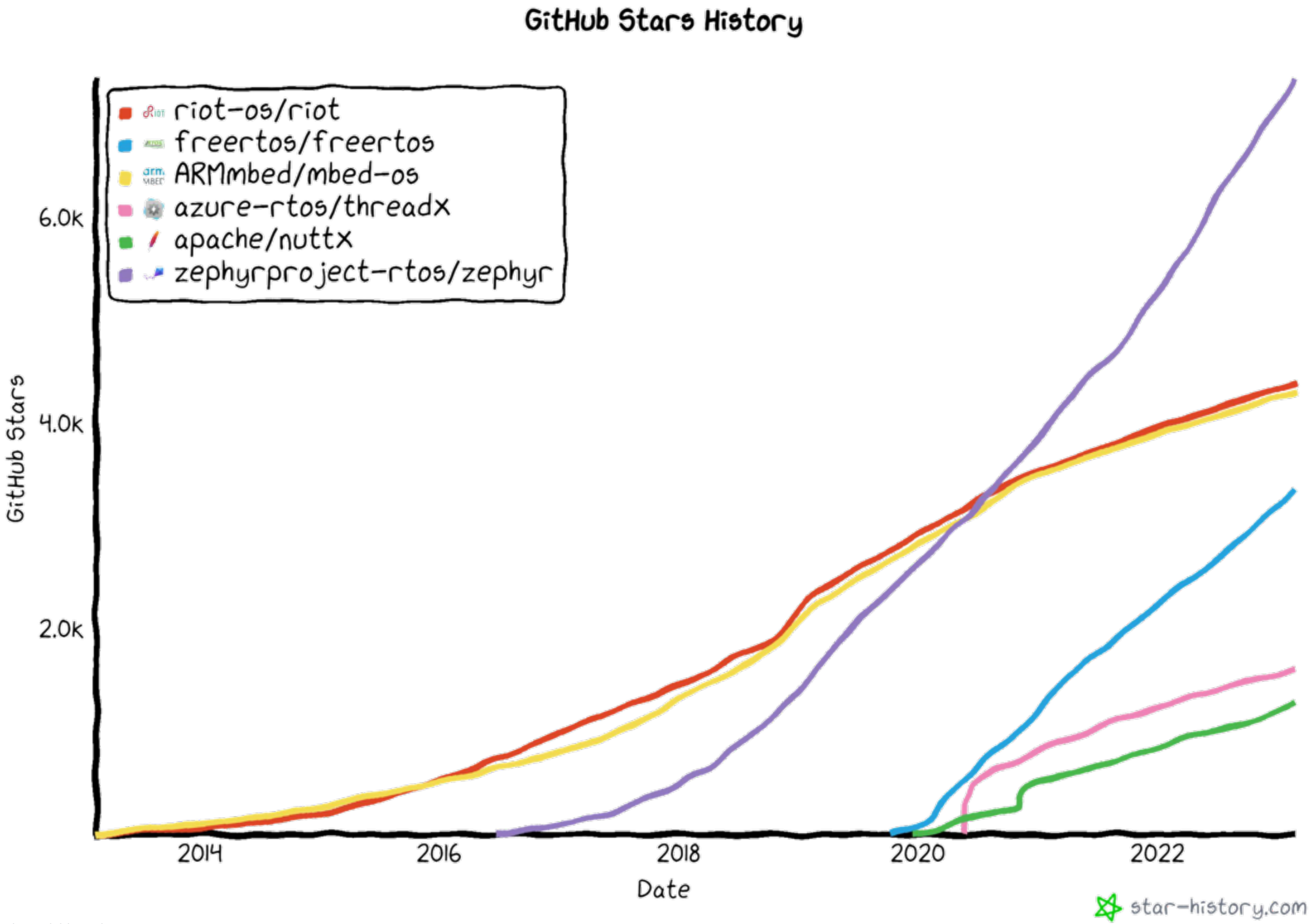


	2016	2017	2018	2019	2020	2021	2022	2023
Amazon FreeRTOS		3	3	30	24	13	3	1
Apache Mynewt	9	18	18	15	11	8	5	4
Apache NuttX	20	22	22	26	36	41	45	62
Arm Mbed OS	39	44	73	69	41	26	7	7
Azure RTOS ThreadX					2	2	2	2
Contiki-NG	17	9	10	8	8	6	8	7
FreeRTOS	2	2	2	3	11	8	6	6
RIOT OS	29	29	37	42	41	31	29	20
RT-Thread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

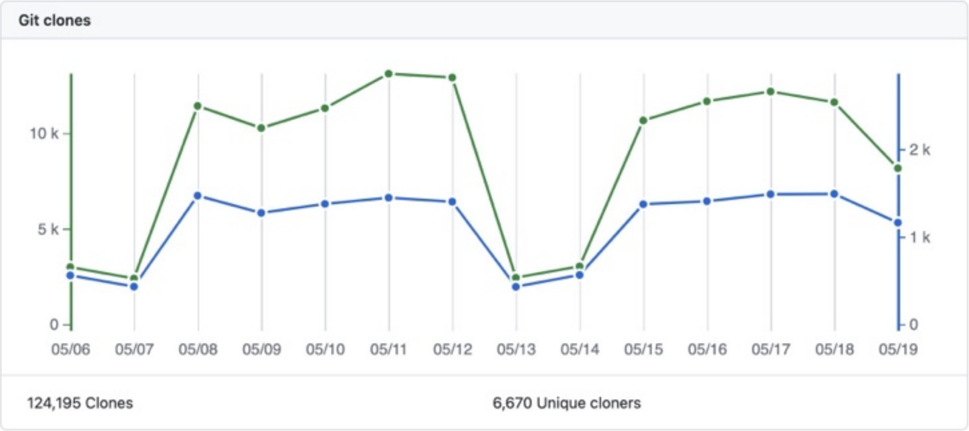
Average Number of Commits per Month



	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
riot-os/riot
freertos/freertos
ARMmbed/mbed-os
azure-rtos/threadx
apache/nuttx
zephyrproject-rtos/zephyr
GitHub Stars
6.0k
4.0k
2.0k
2014
2016
2018
2020
2022
Date
star-history.com

GitHub Clones & Unique Visitors



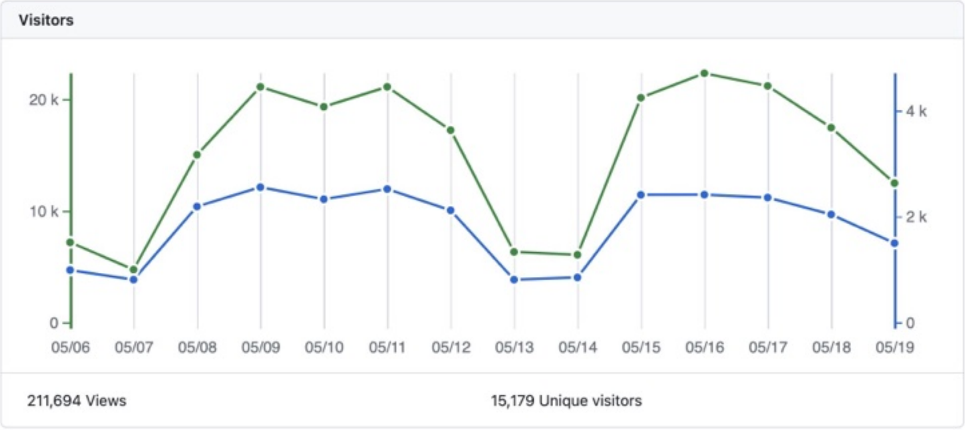
Git clones

124,195 Clones

6,670 Unique cloners

2023-05-06 → 2023-05-19

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



Visitors

211,694 Views

15,179 Unique visitors

Zephyr Participation Information

- zephyrproject.org
- github.com/zephyrproject-rtos
- lists.zephyrproject.org
- chat.zephyrproject.org

zephyrproject.org