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

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

- 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

© 2023 The Zephyr Project — Content made available under CC BY-SA 4.0.
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: docs.zephyrproject.org/latest/boards
120+ Sensors Already Integrated

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

- ARC
- Arm
- Intel
- MIPS
- Nios II Processor
- RISC-V
- SPARC
- Tensilica

- Cortex-M, Cortex-R & Cortex-A
- x86 & x86_64
- 32 & 64 bit
- Xtensa

[docs.zephyrproject.org/latest/hardware/index.html#hardware-support](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, 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

```
&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

© 2023 The Zephyr Project — Content made available under CC BY-SA 4.0.
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](docs.zephyrproject.org/latest/guides/portability/posix.html)
A real-time OS

Benchmark on Arm Cortex-M4F running at 120 MHz

<table>
<thead>
<tr>
<th>Operation</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thread create</td>
<td>2.5 µs</td>
</tr>
<tr>
<td>Thread start</td>
<td>3.6 µs</td>
</tr>
<tr>
<td>Thread suspend</td>
<td>3.3 µs</td>
</tr>
<tr>
<td>Thread resume</td>
<td>3.8 µs</td>
</tr>
<tr>
<td>Context switch (yield)</td>
<td>2.2 µs</td>
</tr>
<tr>
<td>Get semaphore</td>
<td>0.6 µs</td>
</tr>
<tr>
<td>Put semaphore</td>
<td>1.1 µs</td>
</tr>
</tbody>
</table>

[Read more in Zephyr's GitHub repository](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

<table>
<thead>
<tr>
<th>Object</th>
<th>Bidirectional?</th>
<th>Data structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIFO</td>
<td>✗</td>
<td>Queue</td>
</tr>
<tr>
<td>LIFO</td>
<td>✗</td>
<td>Queue</td>
</tr>
<tr>
<td>Stack</td>
<td>✗</td>
<td>Array</td>
</tr>
<tr>
<td>Message queue</td>
<td>✗</td>
<td>Ring buffer</td>
</tr>
<tr>
<td>Mailbox</td>
<td>✔</td>
<td>Queue</td>
</tr>
<tr>
<td>Pipe</td>
<td>✗</td>
<td>Ring buffer</td>
</tr>
</tbody>
</table>

Data passing objects available in Zephyr kernel
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](#13718))
- Zync synchronization primitives ([#48340](#48340))
- USB-C improvements ([#38371](#38371))
- USB device/host enhancements ([#42066](#42066))
- Replace CivetWeb HTTP server ([#46758](#46758))
- Better support for multi-core AMP SoCs ([#51833](#51833))

**Note:** Zephyr is an open-source project, and its roadmap is subject to change.
Safety & Security
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](https://github.com/zephyrproject-rtos/zephyr/releases/tag/zephyr-v2.7.0)
Long Term Support (LTS - 1.14)

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
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
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.
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
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
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
Zephyr Security Summary

- **Documented secure coding practices**
- **Weekly Coverity scans**
- **MISRA scans**
- **Vulnerability response criteria publicly documented**
- **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

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

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

⇒ Provide the evidences that open source developers can map to compliance and meet all requirements
Ecosystem & Governance
Zephyr Project: Platinum Members

- Analog Devices
- antmicro
- Baumer
- Google
- Intel
- Meta
- Nordic Semiconductor
- NXP
- Oticon
- Qualcomm Innovation Center
- T-Mobile
Zephyr Project: Silver Members
Vibrant Ecosystem

Development Tools

Governing Board

Technical Steering Committee

Contributors

Applications & Middlewares

Training & Consulting

Firmwares & Libraries
Ecosystem // Apps & Middlewares

Remote Management

- HERALD
- Golioth
- AVSystem
- Memfault
- STERNUM
- blueswireless

Robotics

- ROS
Zephyr in the RTOS landscape
Average Number of Unique Contributors per Month

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