

Zephyr 3.4: An RTOS of a Different Stripe

The just-released Zephyr 3.4 RTOS isn't a dominant player in the market, but its feature-rich environment and Linux Foundation roots are piquing interest globally.

With more than 100 real-time operating systems (RTOSs) available globally, according to Wikipedia, why should another like Zephyr, at the end of the alphabet, be of interest—and why does its latest version merit attention? You might also ask, “Why should I be interested in Zephyr 3.4?”

Let's start with roots. The Linux Foundation supports collaboration on open-source software, hardware, and standards. Zephyr is the Linux Foundation's RTOS, running not only on small embedded processors, but also on larger processors. It now boasts an extensive set of libraries and capabilities and runs on 450+ development boards from most major manufacturers, though that's not as many as some more established RTOSs.

Development of Zephyr has been supported by major chip producers including Texas Instruments, Intel, Nordic Semiconductor, NXP, SiFive, and Synopsys. And, with its extensive support of peripherals, connectivity, and configuration flexibility, it's becoming a strong contender as an operating system for IoT devices.

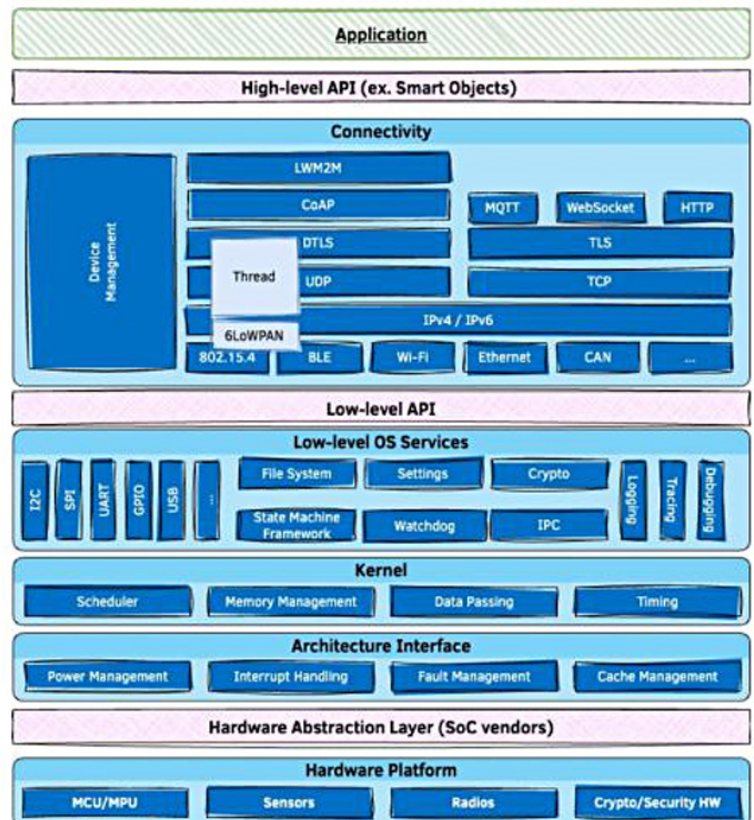
According to the Zephyr folks, the project recently achieved 80,000 commits since it was released in open source in 2015 and has a growing set of software libraries. These apply across a wide range of applications and industry sectors ranging from wearables to industrial IoT (IIoT) and machine learning. A distinguishing trait is an emphasis on broad chipset support and a growing open-source ecosystem.

While nothing is particularly revolutionary in Zephyr or its 3.4 release, there are useful, incremental improvements, especially with Bluetooth

support and the ability to interact with more current, complicated environments in the IoT world.

Zephyr vs. the Competition

While there's plenty of choice among real-time operating systems, the two biggest and best-known contenders



Zephyr Project is a small, scalable, RTOS designed specifically for small-footprint IoT devices with a modular design. It gives developers the ability to customize its capabilities and create IoT solutions that meet the needs of any device.

are FreeRTOS, now owned by Amazon, and ThreadX, now owned by Microsoft.

Despite being robust and popular, some commentators have expressed concern that the respective owners might make both less “open” in the future, which happened to some similar projects after being brought under the wing of large corporations. In this case, both now live within each company’s IoT offerings.

But concern about other RTOS options isn’t all that gives Zephyr devotees hope the younger RTOS could become a key alternative. Each of them has different strengths.

For example, Amazon’s FreeRTOS kernel features a small memory size and generally “low overhead” as well as rapid execution. ThreadX (aka Azure RTOS ThreadX) from Microsoft has some interesting functional features including thread synchronization and interthread communication, as well as preemptive scheduling and rapid interrupt response.

Then Zephyr blew onto the scene. With a name derived from Zephyrus, the Greek personification of the west wind, for some it has been a breath of fresh air because they believe its Linux Foundation roots will lead to more independence, visibility, and openness to contributors.

The difference can be seen in GitHub contributor guidelines that, in the case of Zephyr, “welcomes and encourage the community to submit patches directly to the project.” And, of course, Zephyr uses the [Apache 2.0 license](#), which enables contributors to freely use, modify, distribute, and sell their own products that include Apache 2.0 licensed software.

In comparison, ThreadX insists that contributors agree to a Contributor License Agreement (CLA) to “grant us the rights to use your contribution.” It warns that when someone submits a pull request, a CLA-bot will automatically determine whether you need to provide a CLA and decorate the PR appropriately” (e.g., label, comment). The project has also adopted the [Microsoft Open Source Code of Conduct](#).

Every Lilbit Counts

Beyond the legalese, there are fans who appreciate other aspects of Zephyr. For instance, a recent [blog](#) created by Lilbit, a provider of IoT pet trackers, explains the adoption of Zephyr RTOS “to achieve optimal performance and reliability.”

The posting goes on to describe the scalability and ease of use of Zephyr RTOS and the value provided by drivers, specifically the inter-integrated circuit (I²C) driver, which supports communication between microcontrollers and peripheral devices such as sensors and displays. This, the company says, made possible interfaces with accelerometers, external flash storage, and LED controllers, “reducing development time and increasing product compatibility.”

Credit was also given to the general-purpose input/output (GPIO) driver used to control the state of digital input/out-

put pins on microcontrollers employed by Lilbit to monitor different types of button presses. Furthermore, a Bluetooth Low Energy (BLE) driver was important for an IoT device that requires local wireless communication. By using the Zephyr BLE driver, Lilbit was able to provide seamless communication with smartphones using the Lilbit app.

New and Improved Zephyr RTOS

Among the features present in the new [Zephyr 3.4](#) are:

- Support for new boards
- New peripheral support
 - Auxiliary displays: Text communication through common API (text, backlight etc.), including drivers
 - NVMe disc support
 - Retained memory for persistent information storage
 - SMBus: Variation of I²C
 - Real-time clocks: Battery-powered clock chip available through API
- Input subsystem: Simplifying interaction with input devices and interfaces (like touchscreen)
- Additional Bluetooth support
 - 5.4: Low-power broadcast, mesh
- Zephyr IoT-oriented state-machine framework

Testing Framework

Zephyr 3.4 includes improvements to its built-in testing framework (Twister) that enable creation of more comprehensive tests than in previous versions. Developers can also use third-party testing frameworks such as pyTest, GoogleTest, and RobotFramework to write end-to-end tests running on either real or emulated hardware.

Returning to our question of “Why Zephyr or why 3.4?” The answer isn’t definitive, but clearly valuable features and scalability cover many avenues of development and implementation. Just as important is the community, which seems to be healthy and growing. For Zephyr, time will tell which way the wind blows, but the links in this article should help satisfy anyone’s curiosity.

References

RTOS Comparisons: <https://slashdot.org/software/real-time-operating-systems-rtos/>

• Zephyr: <https://slashdot.org/software/p/Zephyr-OS/>

Zephyr from Linux Foundation:

• Zephyr Project: <https://zephyrproject.org/learn-about/>

• Zephyr web site: <https://zephyrproject.org/>