

# Introduction to Zephyr – the go-to choice for low power and secure connected wireless applications

OSADL COOL – February 2026 edition

*Tiago Monte, Developer Marketing Manager*



# Agenda

- Intro to Nordic Semiconductor
- What is Zephyr (RTOS)?
- Why did Nordic choose Zephyr?
- Nordic's Zephyr-based nRF Connect SDK
- Nordic's contributions to Zephyr
- Supporting developers beyond the SDK



# We are Nordic

Simplifying lives through all things  
connected



Fabless  
semiconductor  
company



Founded in  
Norway, 1983



~1,400  
employees



\$668 MUSD  
revenue, CY 2025  
OSEBX: NOD



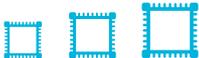
ISO 9001  
certified



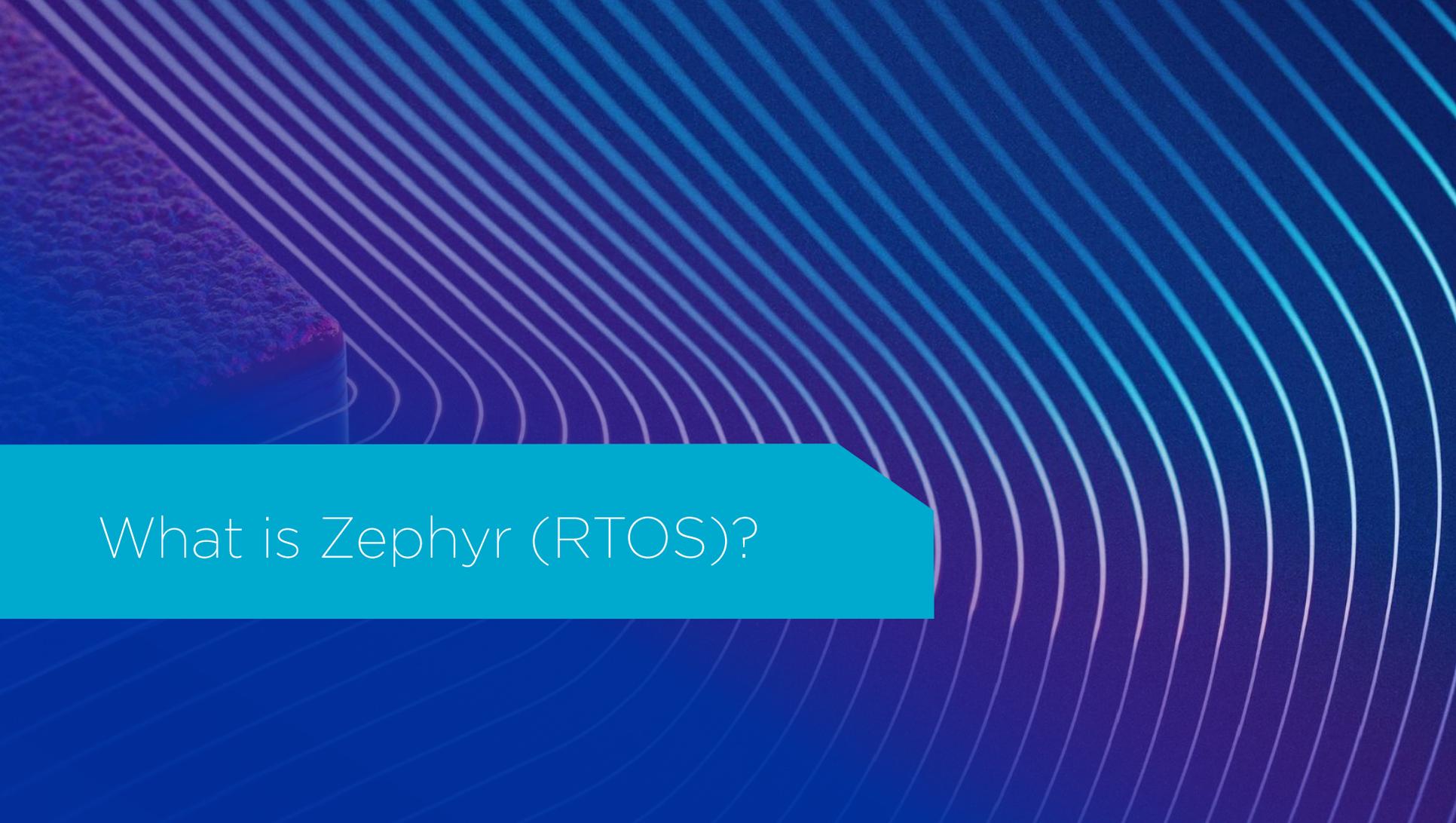
Global  
presence

# Complete solution

## Faster time-to-market

| Next-gen hardware  | Embedded software  | World-class support   | Customer device   | Cloud lifecycle services   |
|--|--|---|---|--|
|  <p>ICs, SoCs, SiPs, PMICs</p>  <p>3rd party modules</p>  <p>Embedded SW stacks</p> |  <p><b>nRF Connect SDK</b><br/>Unified software</p>  <p>Mobile Apps</p>  <p>Extensive SW/HW development tools</p> |  <p><b>DevZone</b><br/>Developer community</p>  <p><b>DevAcademy</b><br/>Online hands-on trainings</p>  <p>Extensive technology partner program</p> |  <p>Consumer</p>  <p>Healthcare</p>  <p>Industrial</p> |  <p><b>nRF CLOUD</b><br/>powered by <b>Memfault</b></p>  <p>Device management</p>  <p>Embedded observability</p>  <p>Location services</p> |



What is Zephyr (RTOS)?



# Is Zephyr “just an RTOS”?

A real-time operating system (RTOS) is an **operating system** for **real-time computing applications** that **processes data and events** that have critically defined time constraints.

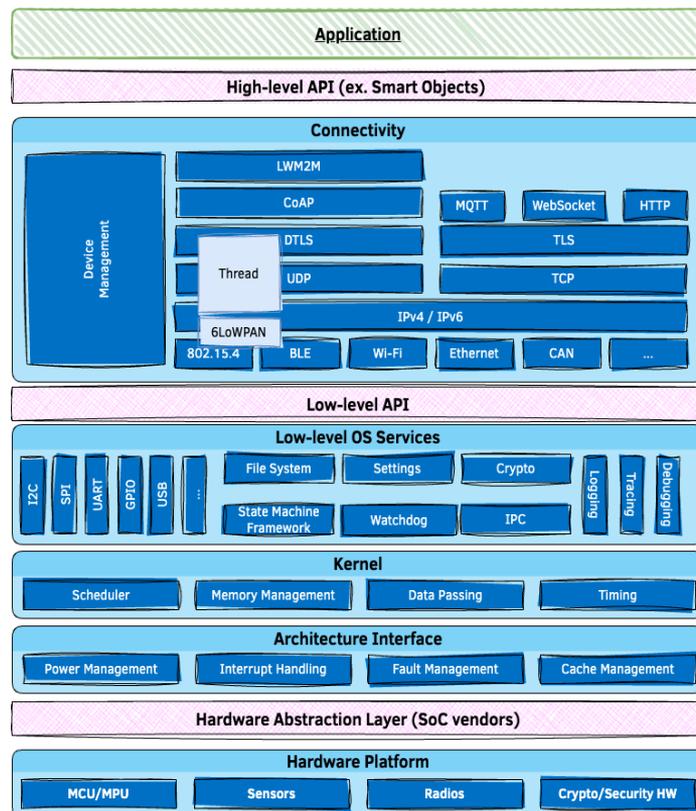
# Zephyr is more than an RTOS



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



# Zephyr is a complete solution: Overview



# Zephyr is a complete solution: Zephyr SDK



zephyrproject-rtos / sdk-ng

- ARC (32-bit and 64-bit; ARCV1, ARCV2, ARCV3)
- ARM (32-bit and 64-bit; ARMv6, ARMv7, ARMv8; A/R/M Profiles)
- Microblaze (32-bit)
- MIPS (32-bit and 64-bit)
- Nios II
- RISC-V (32-bit and 64-bit; RV32I, RV32E, RV64I)
- RX
- SPARC (32-bit and 64-bit; SPARC V8, SPARC V9)
- x86 (32-bit and 64-bit)
- Xtensa (sample\_controller and other vendor-specific variants)

- Complete means complete, including toolchain:
  - Multiple architectures supported by Zephyr
  - Comes with picolibc and all runtimes precompiled
  - Even has optional QEMU and OpenOCD packages

# Zephyr is a complete solution: Build and Config



# CMake

+

```

File Edit View Search Terminal Help
(Zephyr Kernel Configuration)
Board Selection (OEMU x86_64) --->
Board Options --->
SoC/CPU/Configuration Selection (Generic x86_64 PC) --->
Hardware Configuration --->
[ ] Debug logging at lowest level
(6) Power-of-two divisor between TSC and APIC timer
(255) Interrupt vector for irq offload
Architecture (x86_64 architecture) --->
General Architecture Options --->
General Kernel Options --->
Device Drivers --->
C Library --->
Additional libraries --->
Bluetooth --->
Console --->
+++++
[Space/Enter] Toggle/enter [ESC] Leave menu [S] Save
[O] Load [?] Symbol info [/] Jump to symbol
[F] Toggle show-help mode [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
  
```

# Kconfig

+



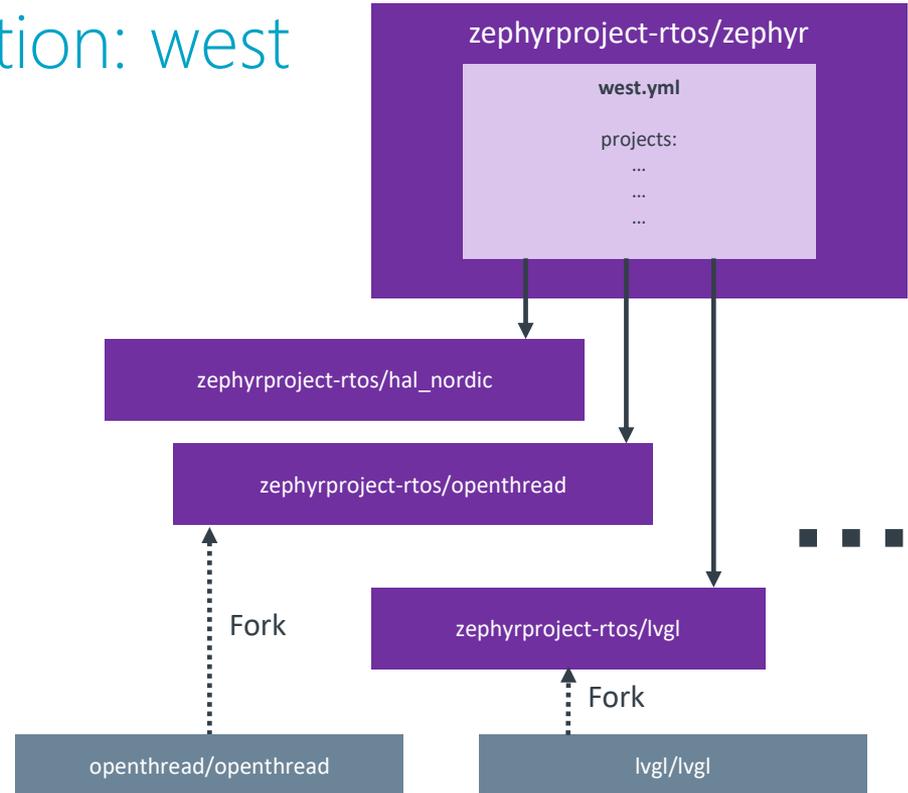
- Build and configuration systems are provided (and required) by the project
- They handle the complexity of the configuration and build processes

# Zephyr is a complete solution: west

## west 1.4.0

```
pip install west
```

- West is provided by Zephyr to:
  - Handle multiple repositories
  - Act as a command-line tool to interface with the build system



# Zephyr is a complete solution: west

## west 1.4.0

```
pip install west
```



- West does much more than repos and building – It's a CLI Swiss army knife

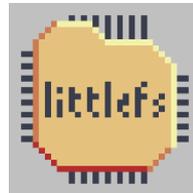
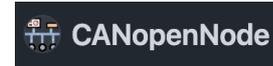
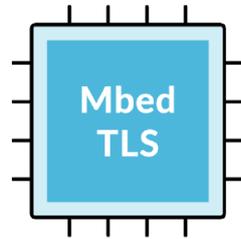
```

completion:    output shell completion scripts
boards:        display information about supported boards
shields:       display list of supported shields
build:         compile a Zephyr application
twister:       west twister wrapper
sign:          sign a Zephyr binary for bootloader chain-loading
flash:         flash and run a binary on a board
debug:         flash and interactively debug a Zephyr application
debugserver:  connect to board and launch a debug server
attach:        interactively debug a board
rtt:           open an rtt shell
zephyr-export: export Zephyr installation as a CMake config
                package
spdx:          create SPDX bill of materials
blobs:         work with binary blobs
bindesc:       work with Binary Descriptors
robot:         run RobotFramework test suites
simulate:     simulate board
sdk:           manage Zephyr SDK
packages:     manage packages for Zephyr
patch:        manage patches for Zephyr modules
gtags:        create a GNU global tags file for the current
                workspace
  
```

# Zephyr is a complete solution: connectivity



# Zephyr is a complete solution: modules





# Our software offering was facing challenges

- Nordic, like most silicon vendors, offers a Software Development Kit (SDK) for its silicon devices free of charge (nRF5 SDK)
- With more capable hardware the software complexity grew considerably, and the SDK architecture had not evolved as quickly as the hardware
- By 2016, before the introduction of nRF9160 SiP, our first LTE-M/NB-IoT device, it had become clear that the bare-metal SDK was not future-proof

- So, it was time to go shopping!

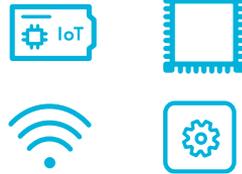


# Continuous technological evolution in IoT

2012



+



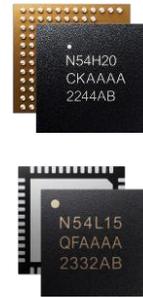
## Hardware

- 16 MHz Arm Cortex-M0
- 256 KB Flash
- 32 KB RAM
- (RAM/Flash ratio 1:8)

## Software

- Application
- Wireless stack
- HAL

2026



+



## Better Hardware

- Cortex-M33 128-320 MHz
- Multiple processors
- RISC-V coprocessors
- ~0.5 MB - 2MB NVM
- ~100 KB - 1 MB RAM
- Higher RAM/NVM ratio (1:2)

## More Software

- Application
- Wireless stacks (plural)
- Bootloader/FOTA
- Cloud connectivity
- Machine Learning
- Security

# Top desired qualities in our shopping list

## For an SDK that will power the next billion IoT devices



Modern: ability to grow alongside our portfolio and be future-proof



Scalable: from small to large devices and applications



Complete: from low-level drivers to cloud connectivity

# Long story short

- Evaluated proprietary and open source solutions
- Zephyr had just been released to the world
- Ticked all the boxes, but also introduced risk
  - New and disruptive development model (to Nordic)
  - Uncertainty about Zephyr and open source for resource-constrained wireless SoCs
  - How to combine with value-add proprietary software
- But the decision was made to move forward



“Instead of waiting to see if Zephyr and open source end up happening, let’s help make them happen!” – inspirational quote in a meeting

# Why did we pick Zephyr?

- Open source community
  - Strict standards for upstream contribution
  - High-quality and low risk due to the governance model
- Zephyr is designed & built for low-power wireless
- Vendor-neutral governance (Linux Foundation)
- More than just a kernel - “batteries included”
  - Wireless and networking stacks, crypto & security, file system, secure bootloader, shell, tracing & logging
  - Out-of-the-box support for 140+ sensors and growing



Platinum Members

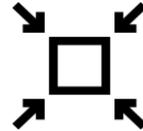


# Why did we pick Zephyr?

- Zephyr is scalable
  - Very small configurations for memory-constrained devices (min ~10 kB e.g. simple Blinky application -> [DevZone blog](#))
  - Powerful, feature-rich, configurations for large memory, high-processing power devices (multiple MBs)
- Developed with a focus on security
  - Product Security Incident Response (PSIRT) team
  - PSIRT team fixes vulnerabilities and issues notifications
- Vibrant community



Zephyr® Project



SMALL

yet



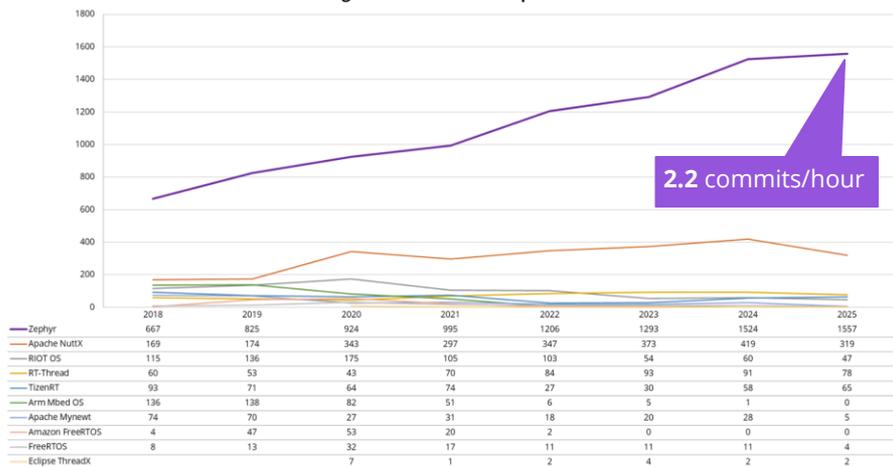
SCALABLE



# Zephyr RTOS was the right choice

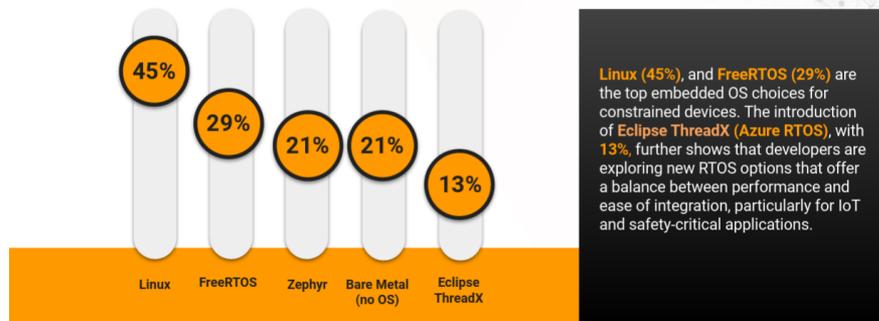
## Vibrant open-source community and fast-growing adoption

Average Number of Commits per Month



Source: Zephyr Project - Zephyr Project Overview ([link](#))

### Developer Embedded OS Preferences on Constrained Devices



Linux (45%), and FreeRTOS (29%) are the top embedded OS choices for constrained devices. The introduction of Eclipse ThreadX (Azure RTOS), with 13%, further shows that developers are exploring new RTOS options that offer a balance between performance and ease of integration, particularly for IoT and safety-critical applications.

Q: Which embedded operating system(s) do you use for your constrained devices? (select up to three)

Source: Eclipse Foundation - 2024 IoT & Embedded Developer Survey Report ([link](#))

# The most significant improvements

Architecture



From bare metal to  
RTOS based

Single code base



From multiple SDKs  
to a unified SDK

Distribution



From zip file to public  
github hosting

Development



From closed to open  
contribution model

# Combined Nordic expertise and open-source



- Proprietary features and technology
- Samples and applications
- Testing and qualification
- IDE (VS Code) integration & ease of use
- Documentation and education
- Technical support
- Kernel and OS services
- Libraries, protocols and middleware
- Build and configuration system
- Zephyr modules
- Tooling (west, twister)
- Vibrant and engaging community



nRF Connect SDK

# Maximize reusability and scalability

West

Multi-repository  
management tool

Kconfig

Source module / feature  
configuration for compile

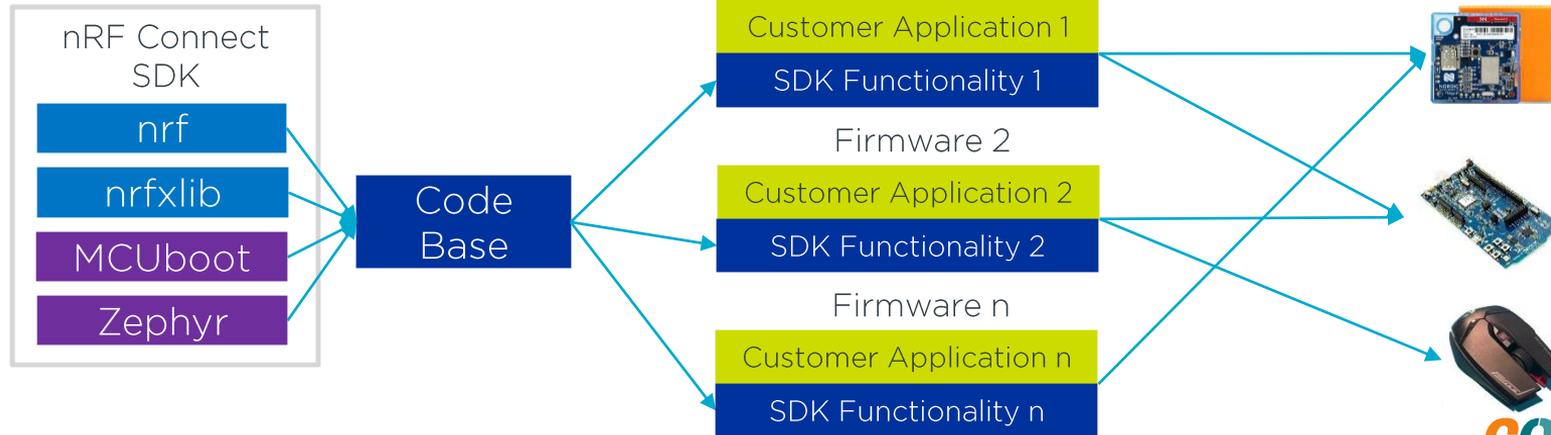
Devicetree

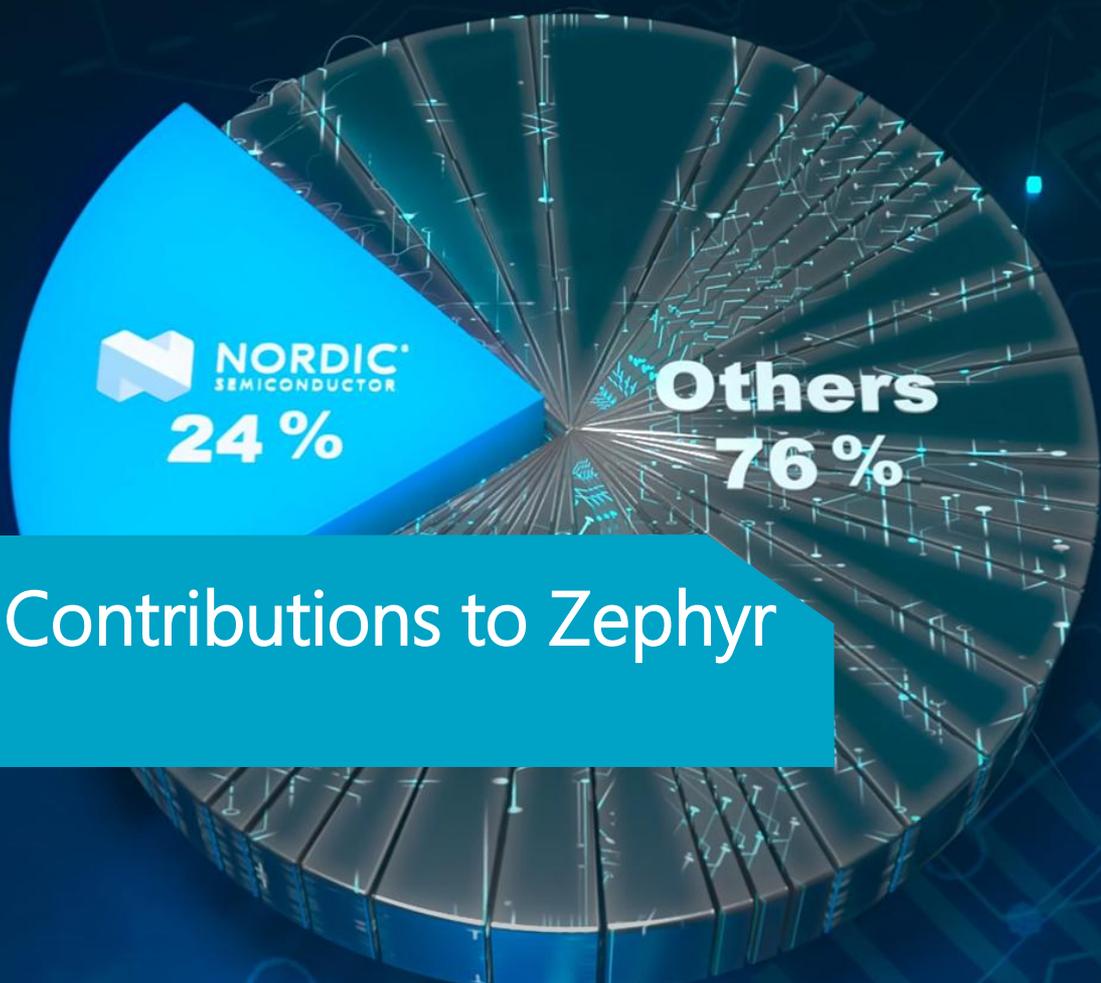
Target Board and SoC  
description

Clone / update

Configure features

Configure target





A 3D pie chart with a blue slice representing 24% and a larger grey slice representing 76%. The background features a dark blue grid with glowing blue lines and dots, suggesting a digital or network environment. The pie chart is rendered with perspective, showing its depth.

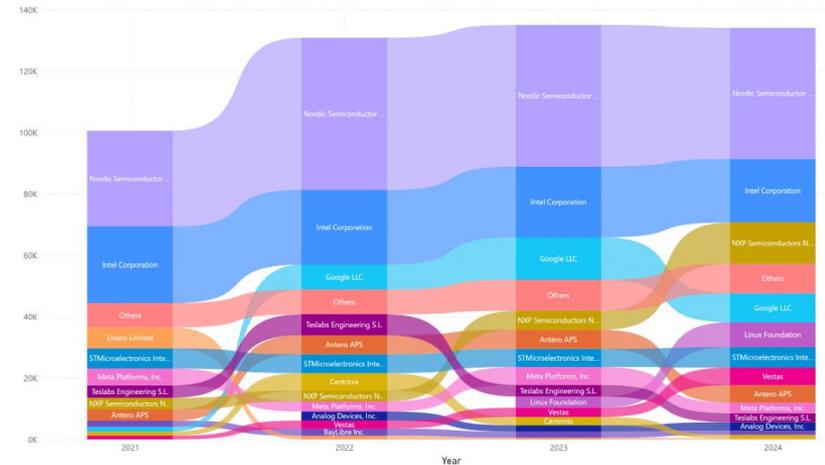
**NORDIC**  
SEMICONDUCTOR  
**24 %**

**Others**  
**76 %**

Nordic's Contributions to Zephyr

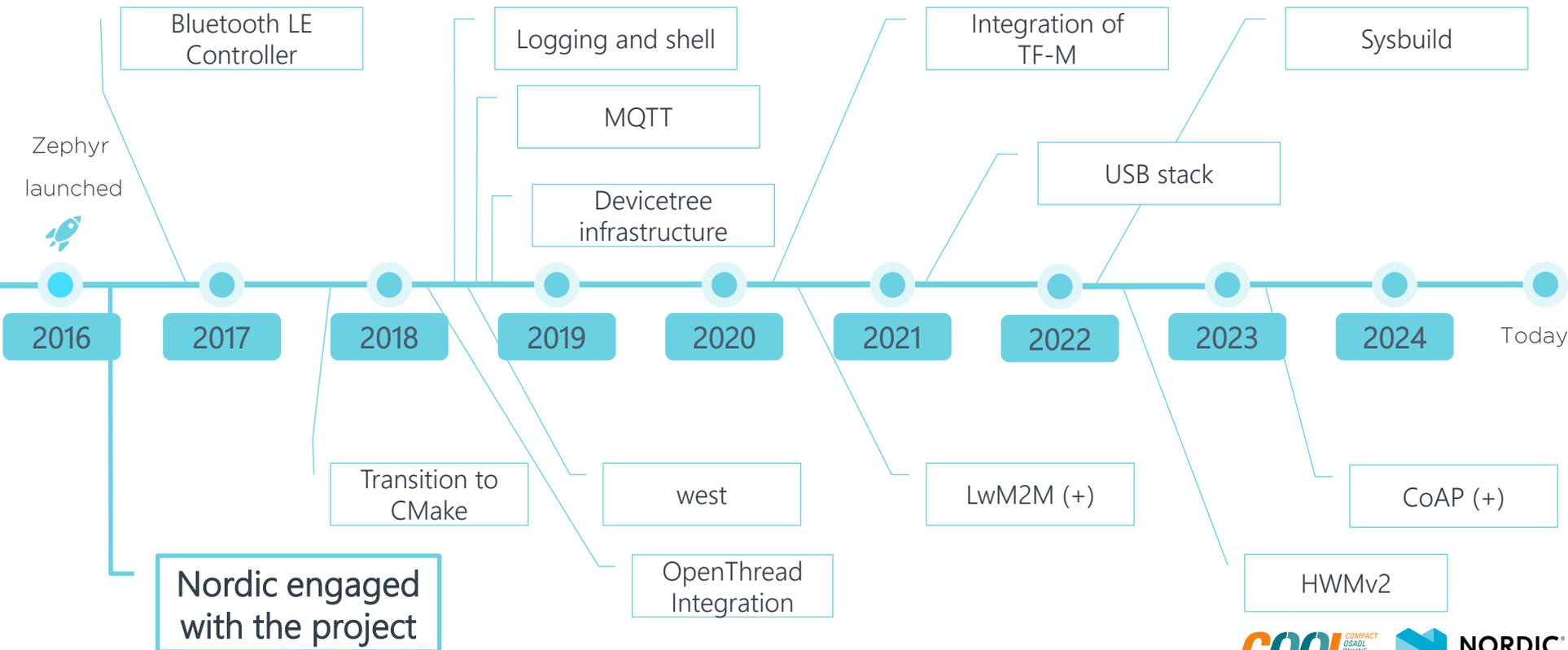
# Nordic commitment to Zephyr

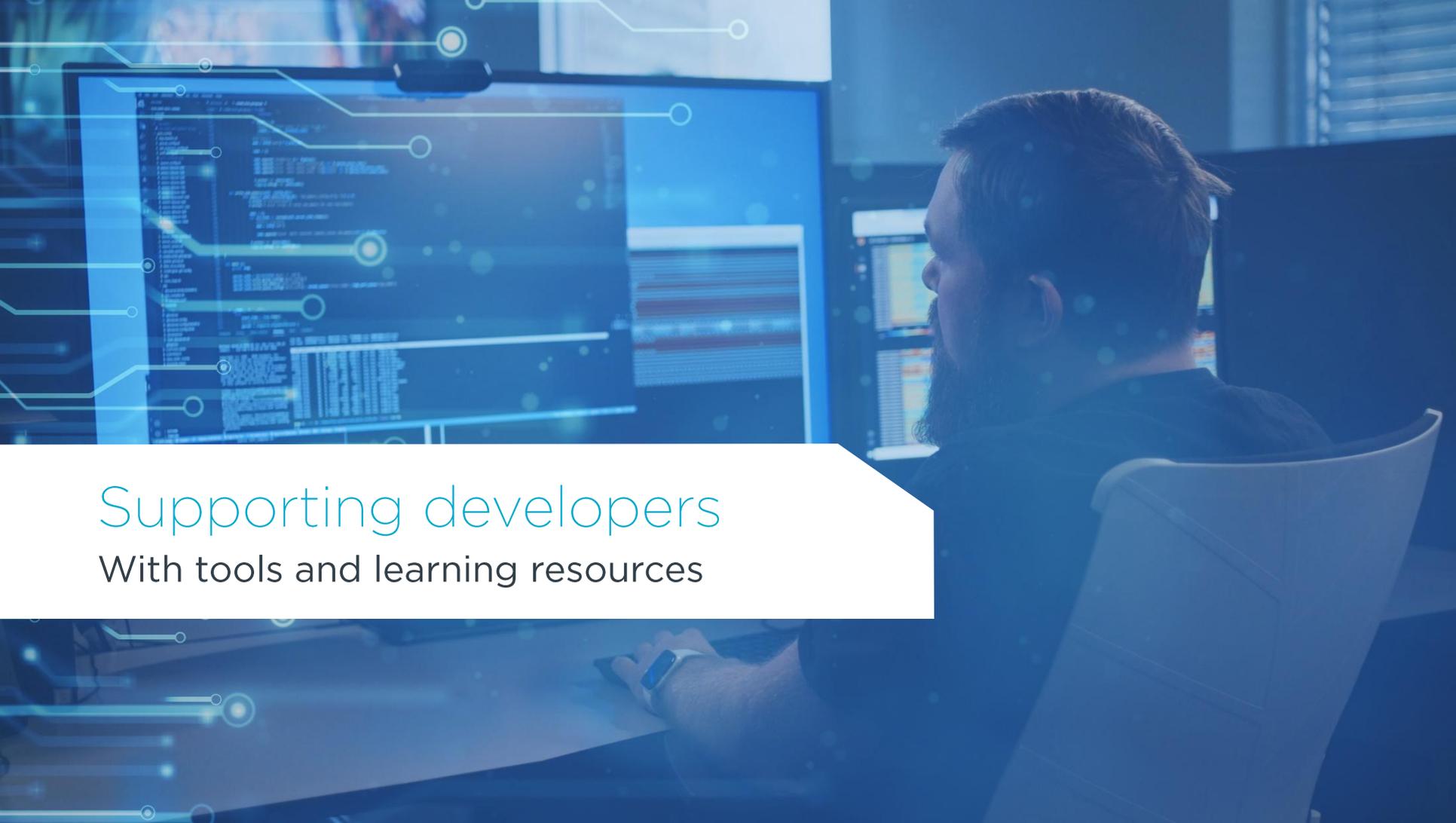
- Engaged with the project from the start in 2016
- Platinum member
- Took an active role on the governing board, even chairing it for a few years
- Chair multiple working groups
- Contribute extensively to both the technical and marketing committees
  - ~Quarter of the total contributions to the Zephyr project came from Nordic
- Host the Nordic Developer Academy, enabling thousands of developers to learn Zephyr / nRF Connect SDK



Zephyr project members' contributions 2021 - 2024  
(Source: [Zephyr Project](#))

# Nordic's major contributions to Zephyr





# Supporting developers

With tools and learning resources

# Nordic Developer Academy

- Online, self-paced, hands-on courses
- Structured learning with curriculum-based courses
- Up-to-date
- Certificates of completion
- Free of charge! All you need is a Nordic development kit



Accelerated learning



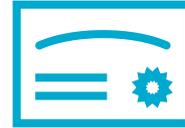
Structured courses



Self-paced learning



Up-to-date content



Certificates of completion

[academy.nordicsemi.com](https://academy.nordicsemi.com)

# Available courses on DevAcademy

NEW!

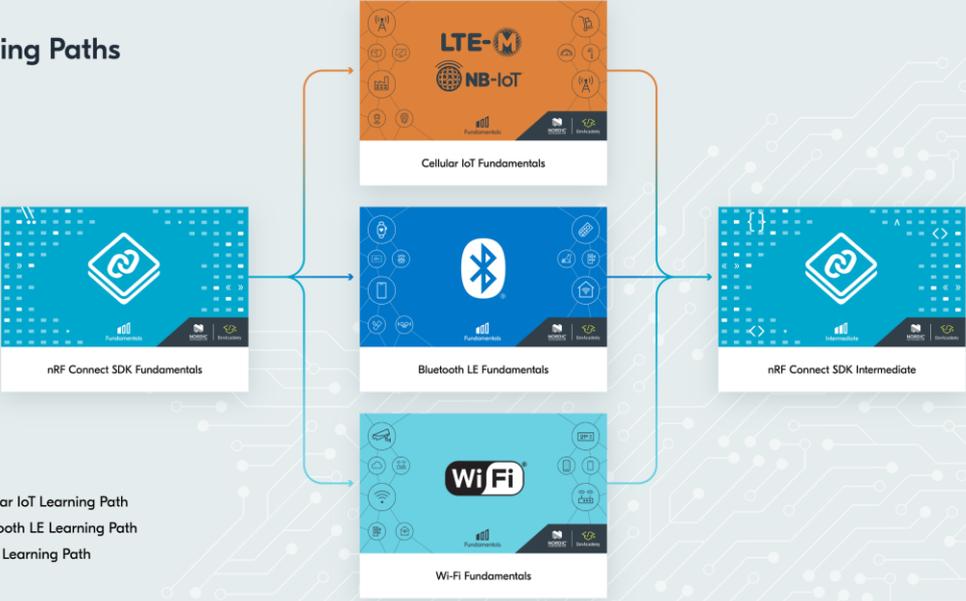
## Express Courses



nRF54L Series Express

The course cover features a blue background with a white circuit board pattern. At the top, there is a blue badge with a rocket icon and the text 'NEW'. The main title 'nRF 54L SERIES' is in white on a black background. The bottom right corner has the DevAcademy logo and 'Fundamentals' text.

## Learning Paths

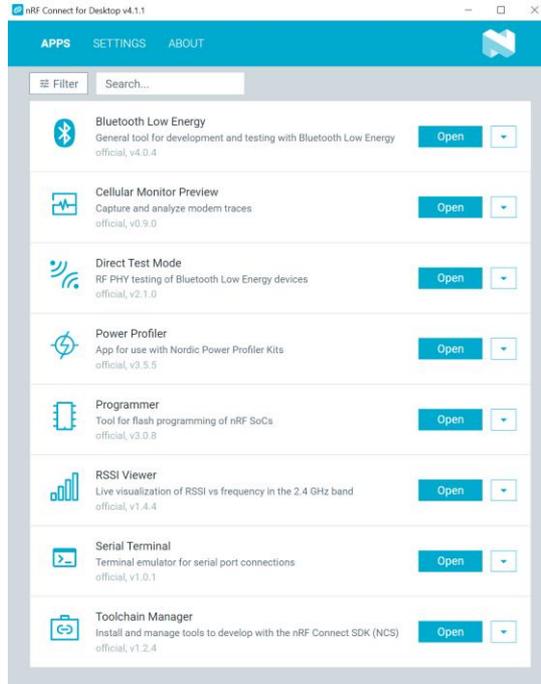


The diagram shows a flow of learning paths. It starts with 'nRF Connect SDK Fundamentals' (blue background) on the left. Three paths branch out from it: 'Cellular IoT Fundamentals' (orange background), 'Bluetooth LE Fundamentals' (blue background), and 'Wi-Fi Fundamentals' (light blue background). All three of these lead to 'nRF Connect SDK Intermediate' (blue background) on the right. A legend at the bottom left identifies the paths: an orange line for Cellular IoT Learning Path, a blue line for Bluetooth LE Learning Path, and a light blue line for Wi-Fi Learning Path.

- Cellular IoT Fundamentals
- Bluetooth LE Fundamentals
- Wi-Fi Fundamentals
- nRF Connect SDK Intermediate

Cellular IoT Learning Path  
Bluetooth LE Learning Path  
Wi-Fi Learning Path

# nRF Connect for Desktop



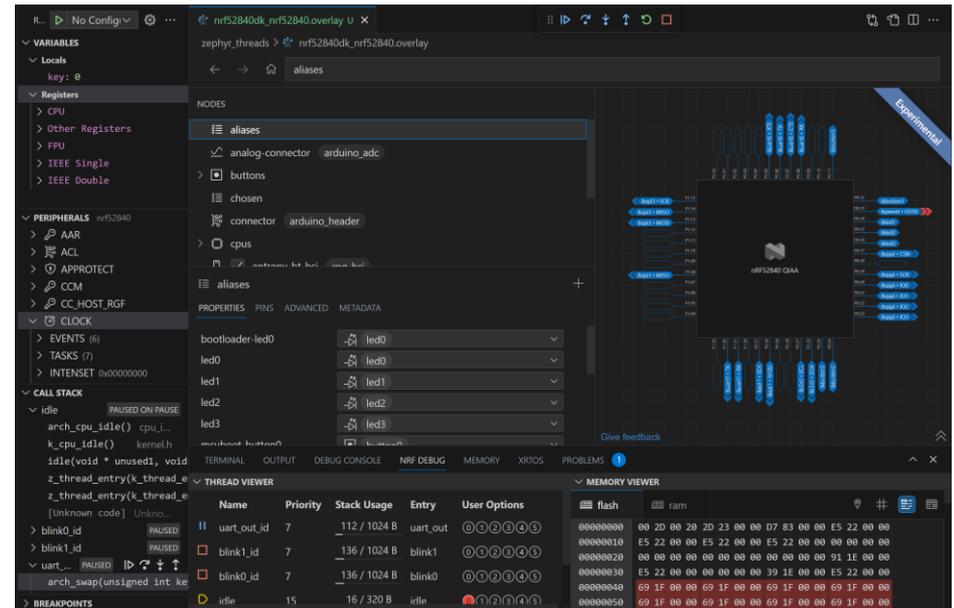
Tools aimed at SW development and general use:

- Toolchain Manager installs and manages tools related to nRF Connect SDK, including VS Code extensions
- Programmer is a great visual tool for flashing Nordic devices, updating modem FW on nRF91 Series
- Power profiler used to optimize power consumption
- Short-range tools: Bluetooth Low Energy, RSSI Viewer, Direct Test Mode
- Cellular tools: Cellular Monitor (replaces Trace Collector and LTE Link Monitor)

# nRF Connect for VS Code (IDE)



- Built from the ground up for nRF Connect SDK
- Easy to use and feature-rich
  - CLI and GUI Interfaces
  - DeviceTree Visual Editor
  - nRF Debug
  - Memory Report
  - Built-in terminal
  - nRF Kconfig
  - SDK/toolchain management
  - Rich set of [tutorial videos](#)

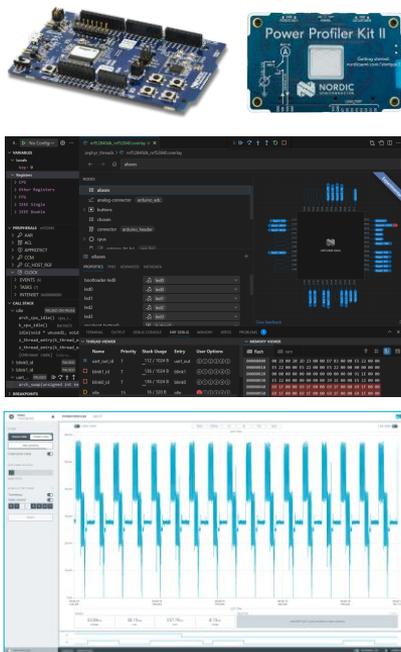


# Nordic empowers developers

## SDK



## Tools



## Support



DevAcademy

# Our sales team in Germany

Thomas Page



Regional Sales  
Director

Kevin Kotinkar



Regional Sales  
Manager

Angela Haug



Regional Sales  
Manager

Christian Kurz



Field Applications  
Engineer

Thank you