

The background of the slide is a deep blue with a complex network of glowing green and yellow circuit lines and nodes. Two wireframe models of a human brain, rendered in a light blue/cyan color, are positioned on either side of the central text, appearing to be part of the circuitry.

Introduction to the nRF5340 SoC and nRF Connect SDK

Garmin Developer Virtual Conference

Bjørn Kvaale / Product Marketing Engineer

October 2022

Today's hosts

Bjørn Kvaale



Product Marketing Engineer
Product Management

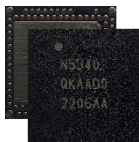
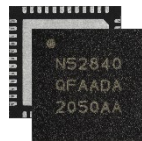
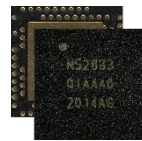
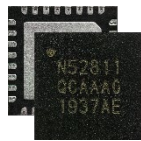
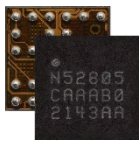


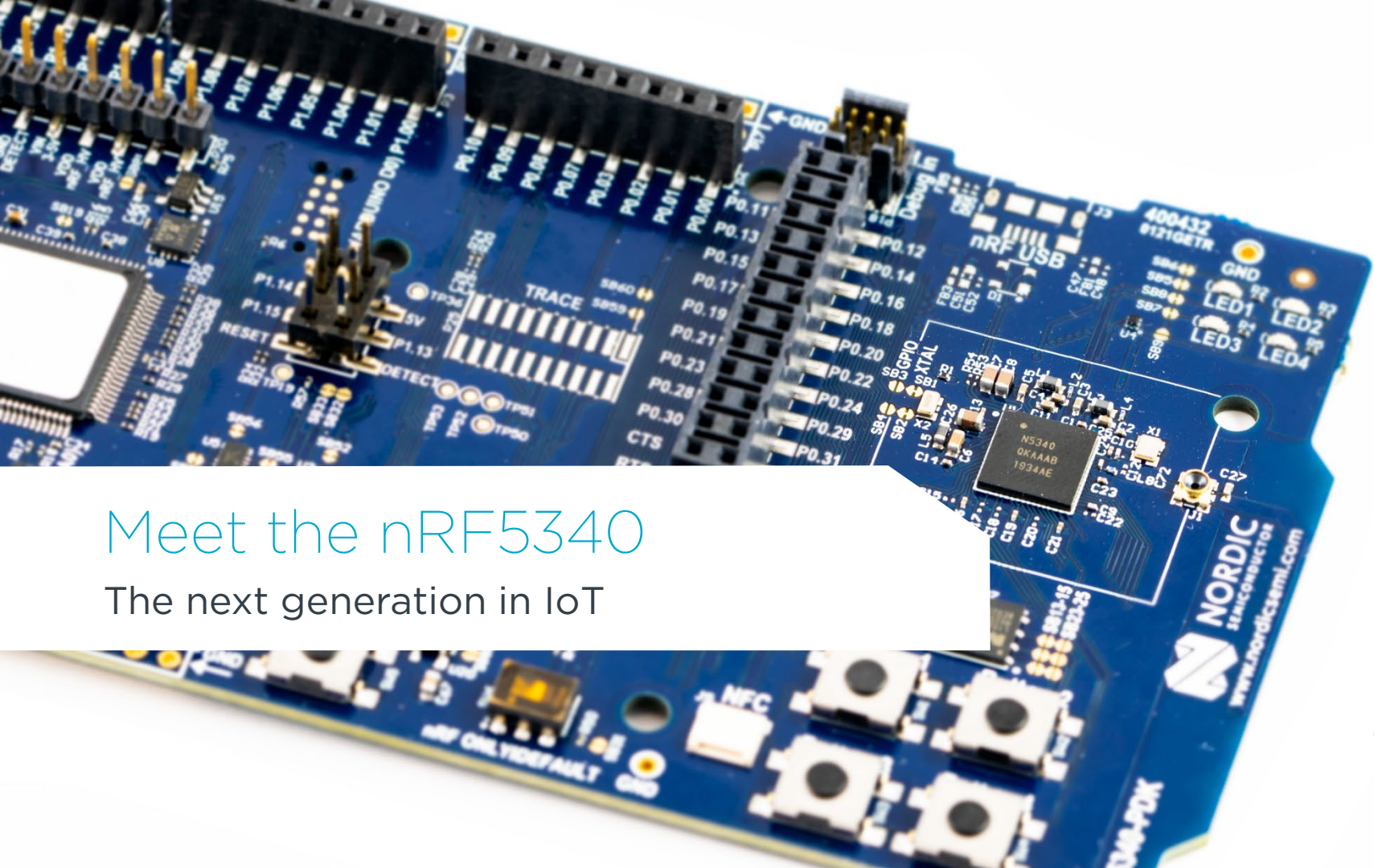
Agenda

- Intro to nRF5340 SoC
- Intro to nRF Connect SDK
 - Evolution of the nRF Connect SDK
 - RTOS Introduction
 - nRF Connect SDK Details

Nordic Low power short-range SoC

Decades of low power connectivity experience





Meet the nRF5340

The next generation in IoT

nRF5340 overview and highlights

nRF5340 SoC

Application Processor

128 MHz Arm®
Cortex®-M33

1024 KB Flash

512 KB RAM

Network Processor

64 MHz Arm®
Cortex®-M33

256 KB Flash

64 KB RAM

Multiprotocol
2.4 GHz radio

- High-performance application processor
- Fully programmable, ultra-low power network processor
- Redesigned multiprotocol radio
- Advanced security features
- 1.7-5.5 V supply range
- -40 to +105°C

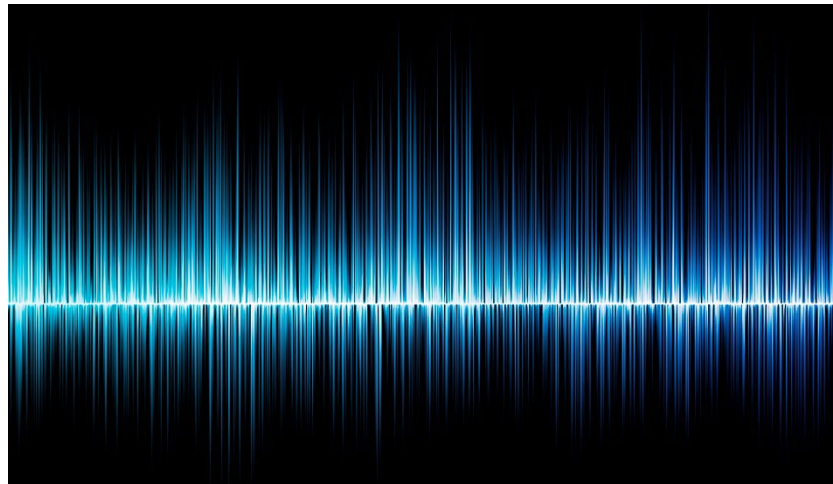
Multi-core flexibility

- High performance and high efficiency – no trade-off
- Distinct optimization
 - Performance
 - Efficiency
- Separation of concerns
 - Real-time requirements
 - Software split



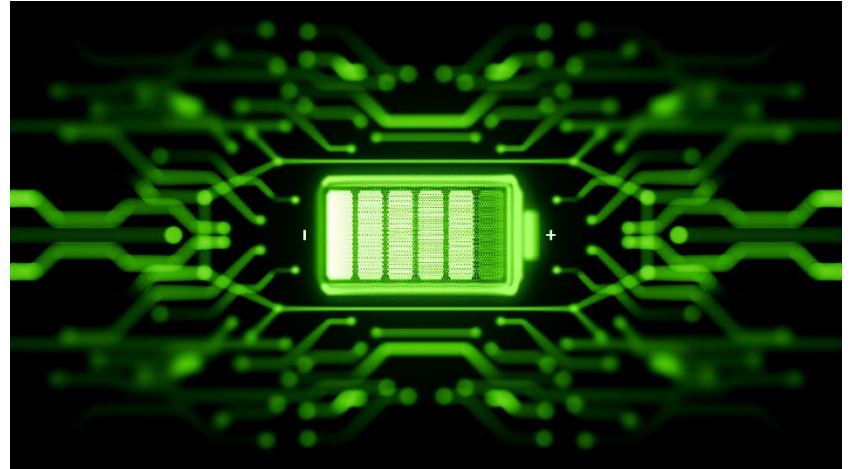
Multiprotocol radio

- Bluetooth 5.3 and beyond
 - Bluetooth LE and ANT concurrently
 - 2 Mbps, Long Range, and Advertising Extensions

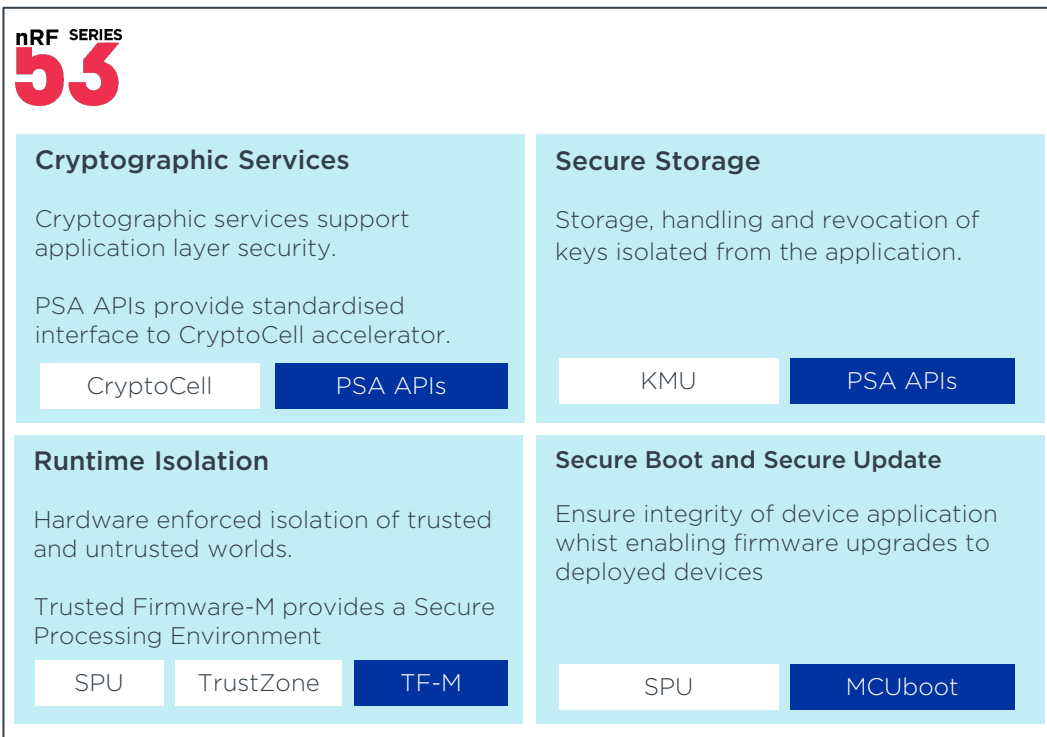


Radio improvements

- TX current reduced with **29 %** down to 3.4 mA
- RX current reduced with **41 %** down to 2.7 mA
- -98 dBm RX sensitivity
- 1 dB resolution of TX power



Nordic security features



All-in-one

- nRF52 Series feature superset
 - Bluetooth 5.3, ANT, Thread and Zigbee
 - CryptoCell, USB, QSPI, HS-SPI
 - 1.7-5.5 V and up to 105 °C
- More
 - Performance
 - Memory
 - Integration
- While minimizing power consumption



Markets and applications

What can it be used for?

Advanced wearables

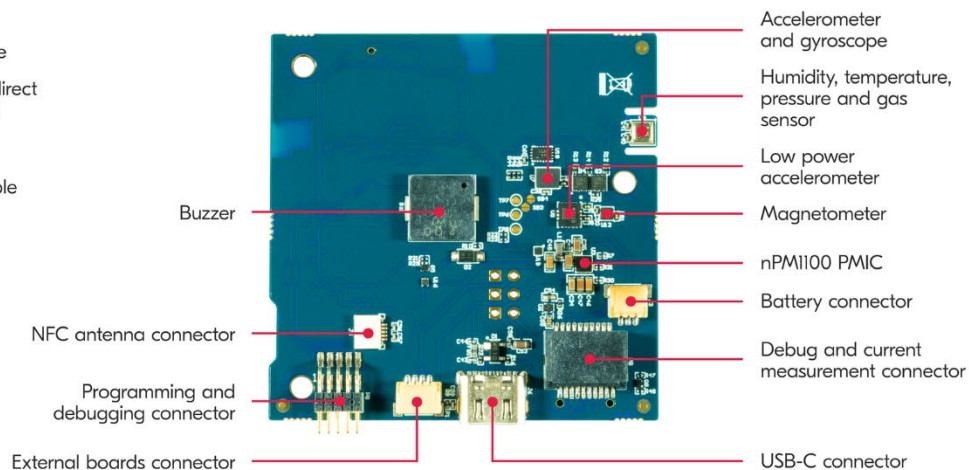
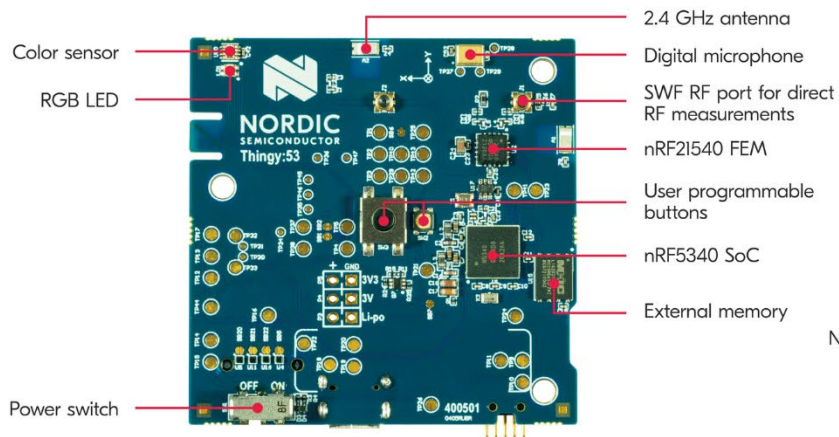


- High-performance 128 MHz processor with DSP instructions and floating-point capabilities
- 512 KB on-chip RAM for data and sensor fusion
- Memory mapped 96 MHz encrypted QSPI interface for external memory
- 32 MHz HS-SPI for displays and fast sensors
- Sensor collection in application or network core depending on bandwidth
- Compact CSP with few external components

Machine Learning

With Edge Impulse

Thingy:53 hardware overview



nRF Edge Impulse app



Embedded
machine learning



Every Thingy:53
ships pre-loaded
with firmware



Integration with
Edge Impulse
Studio



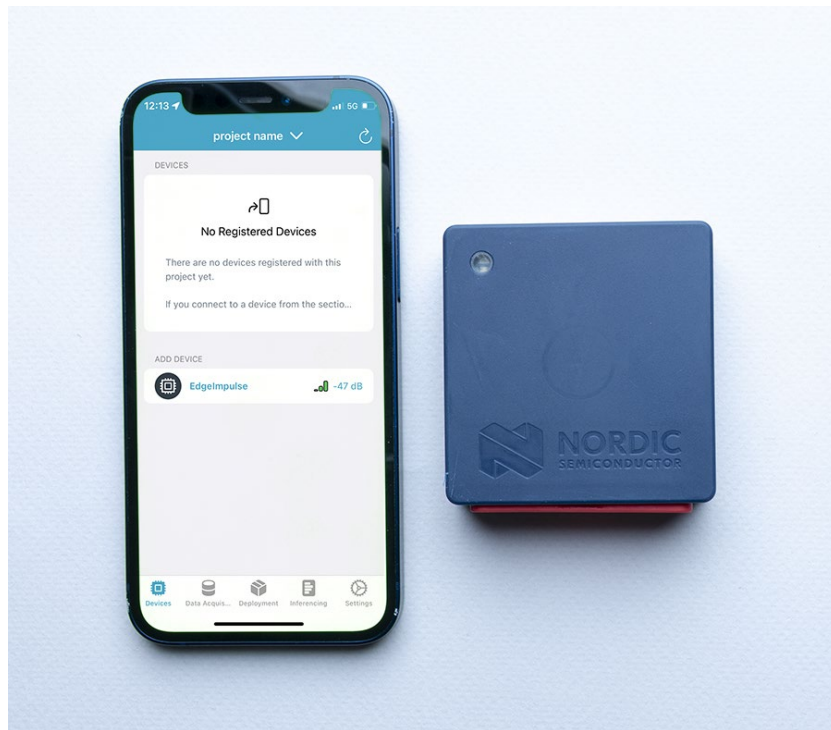
Data collection



Running ML
models

Rapid ML prototyping with Thingy:53

- Ideal for rapid embedded ML prototyping
- Integrates seamlessly with Edge Impulse Studio
- Thingy:53 for both data collection and running trained ML models
- Wake-on-event sensors
- OTA DFU for ML models through the app



Getting into the details



nRF5340 SoC

Power and clock

Power supply

LDO

Buck DC/DC

POR

BOR

Oscillators

32 MHz RC/XO

32 kHz RC/XO

Audio PLL

Debug

Debug

Application Processor

64/128 MHz
Arm®
Cortex®-M33,
DSP, FPU,
TrustZone

1024 KB Flash

512 KB RAM

8 KB I-Cache

IPC

AHB / APB / EasyDMA / DPPI

System Peripherals

3×TIMER

2×RTC

2×WDT

6×EGU

Security

Arm CryptoCell-312

SPU

KMU

Digital, analog I/F and GPIO

USB

QSPI

HS-SPI

4×UART/SPI/TWI

I2S

PDM

4×PWM

2×QDEC

NFC-A Tag

SAADC

LPCOMP

COMP

GPiOTE

Shared 48-pin crossbar

Network Processor

64 MHz Arm®
Cortex®-M33

256 KB Flash

64 KB RAM

2 KB I-Cache

IPC

AHB / APB / EasyDMA / DPPI

System Peripherals

3×TIMER

2×RTC

WDT

EGU

RNG

TEMP

ECB

AAR

CCM

Digital I/F and GPIO

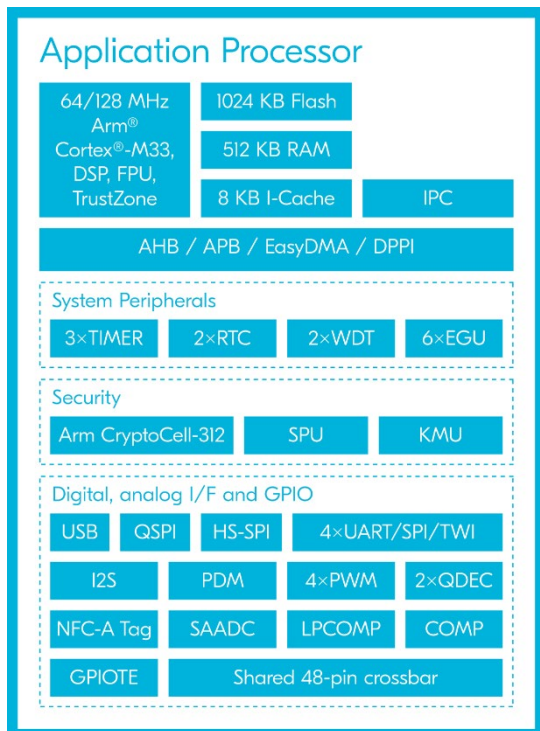
UART/SPI/TWI

GPiOTE

Shared 48-pin crossbar

Multiprotocol
2.4 GHz radio

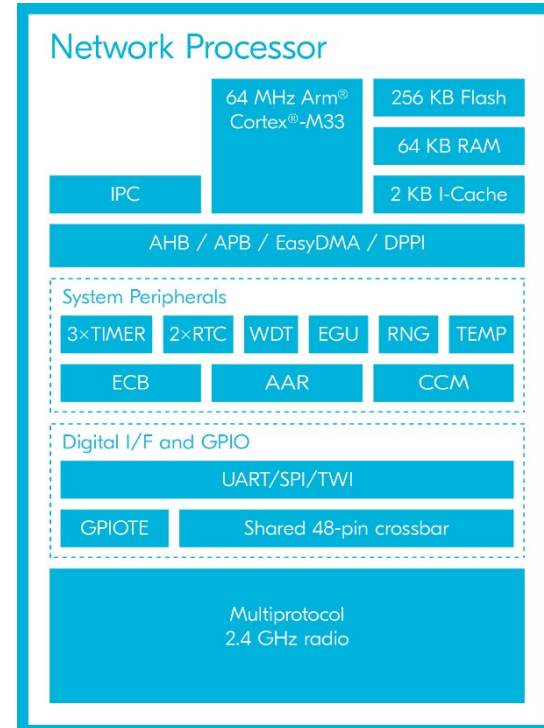
Application processor



- Arm Cortex-M33 with DSP and floating-point instructions
- 128 MHz and 64 MHz clock speeds
- Voltage-frequency scaling
- Optimized for performance
- TrustZone support
- 8 KB 2-way set associative cache
- 1024 KB flash
- 512 KB RAM

Network processor

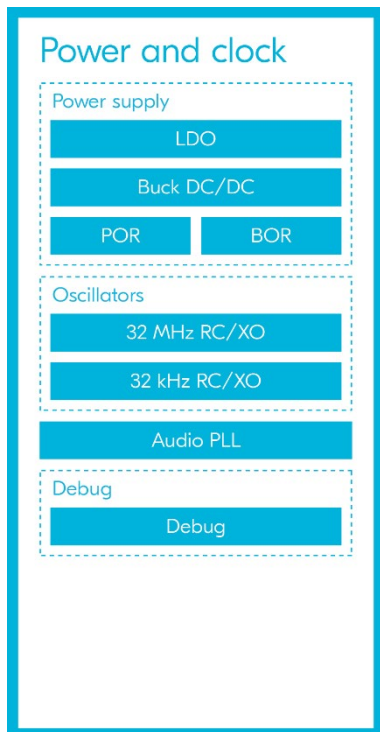
- Arm Cortex-M33
- 64 MHz clock speed
- Optimized for efficiency
- Fully programmable
- 2 KB instruction cache
- 256 KB Flash
- 64 KB RAM



Multi-core flexibility



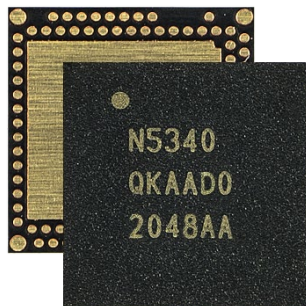
Power and clock



- Power supply
 - 1.7 to 5.5 V
 - LDO and DC/DC options
 - Direct supply from USB
- Oscillators
 - Integrated load capacitors
 - 32 MHz and optional 32 kHz crystals
- Tunable on-chip Audio PLL
 - 11.289 or 12.288 MHz
 - Low jitter, suitable for audio applications

Packages

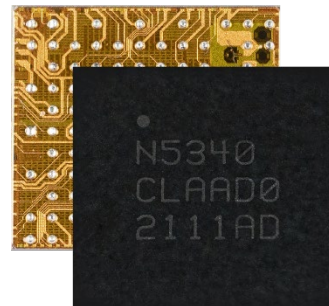
7x7 mm



aQFN94 with 48 GPIOs

In mass production

4.4x4.0 mm

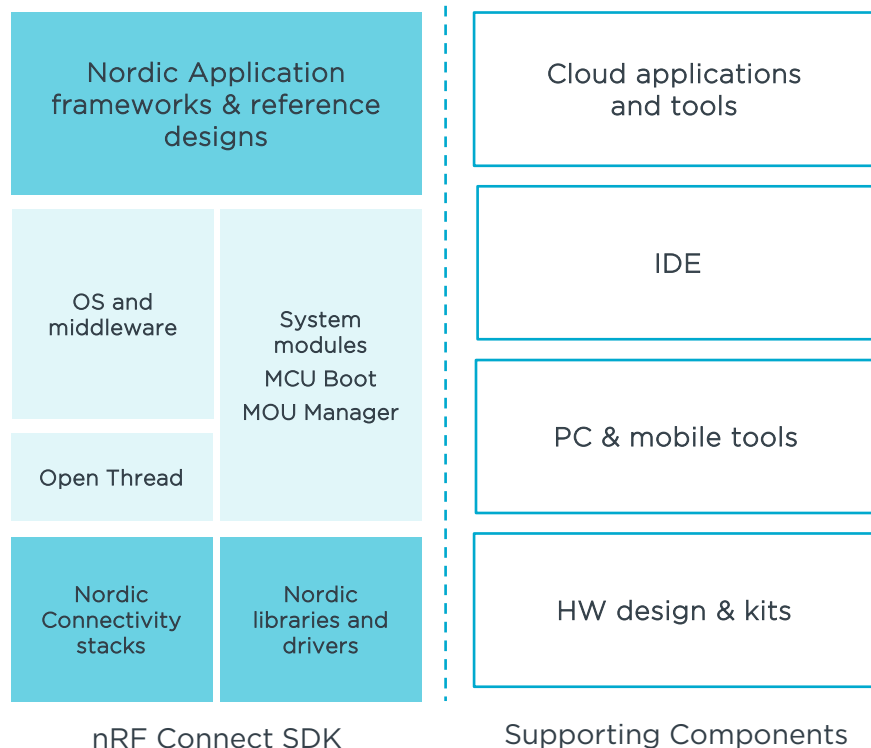


WLCSP95 with 48 GPIOs

In mass production

Software Support

nRF Connect platform



Unifying

- Support for all Nordic ICs (>nRF52 Series)
- Support for all Nordic wireless technologies (Bluetooth LE, ANT, Thread, ZigBee, etc)
- Complete solution from board to cloud

Modern

- RTOS integration for complex designs
- Open source when and where it makes sense
- Modern development and distribution model

Scalable

- For low and high complexity applications
- For single core and multi-core SoCs
- Allow customer to focus on application

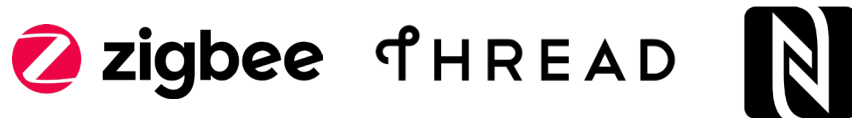
nRF Connect SDK

Introduction

nRF Connect SDK



- One code base and toolchain for nRF91, nRF70, nRF53, nRF52 and nRF21 Series
 - Optional for nRF52 Series ($\geq$ v1.3.0)
- Includes ANT, Bluetooth Low Energy, Wi-Fi, LTE-M, NB-IoT, GNSS, Bluetooth mesh, Thread, Zigbee, Matter, ESB, Gazell, NFC
- Bluetooth v5.3 qualified Host and Controller stack since v2.0.0

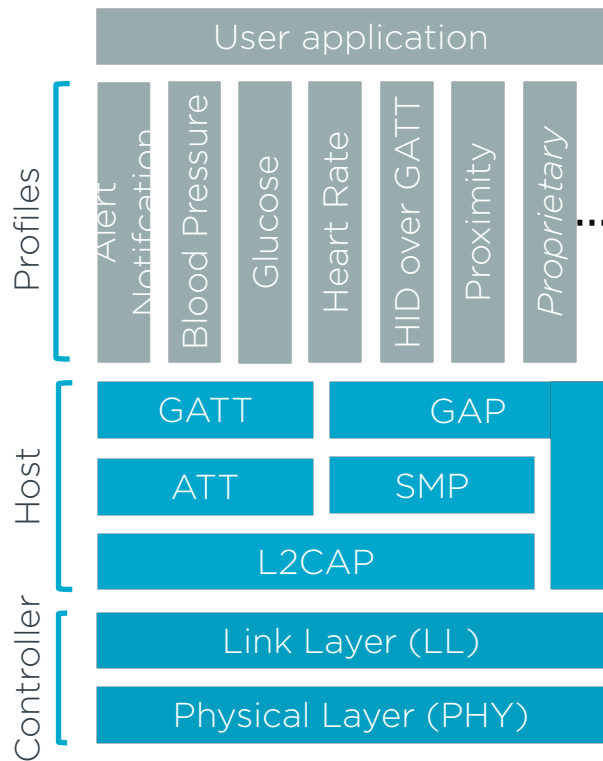


nRF Connect SDK

Evolution

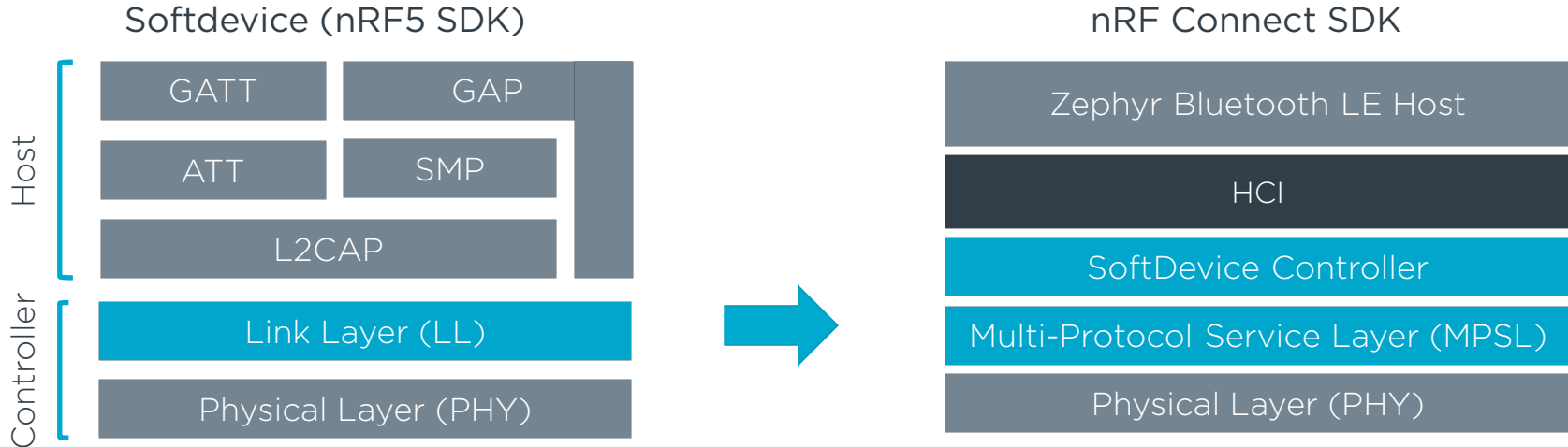
Ported SoftDevice Controller

- Market leaders in Bluetooth LE in large part to our nRF5 SDK + SoftDevice
- Link Layer (LL) enables best in class interoperability between SoC and smartphones and is difficult to get right
- Ported the SoftDevice LL to nRF Connect SDK
 - Ensures best in class smartphone interoperability in nRF Connect SDK vs nRF5 SDK + SoftDevice



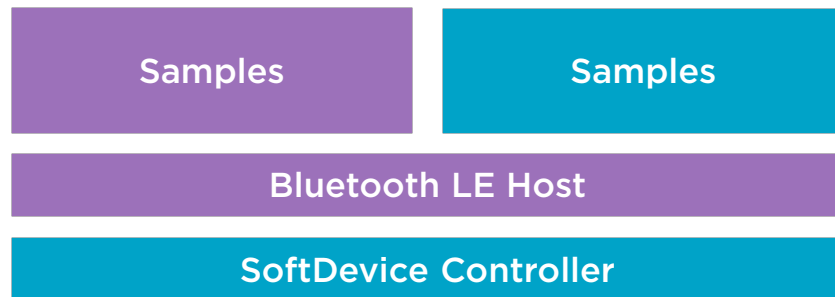
SoftDevice Controller

- The standalone SoftDevice Controller library is now the default Bluetooth LE Controller for Bluetooth samples

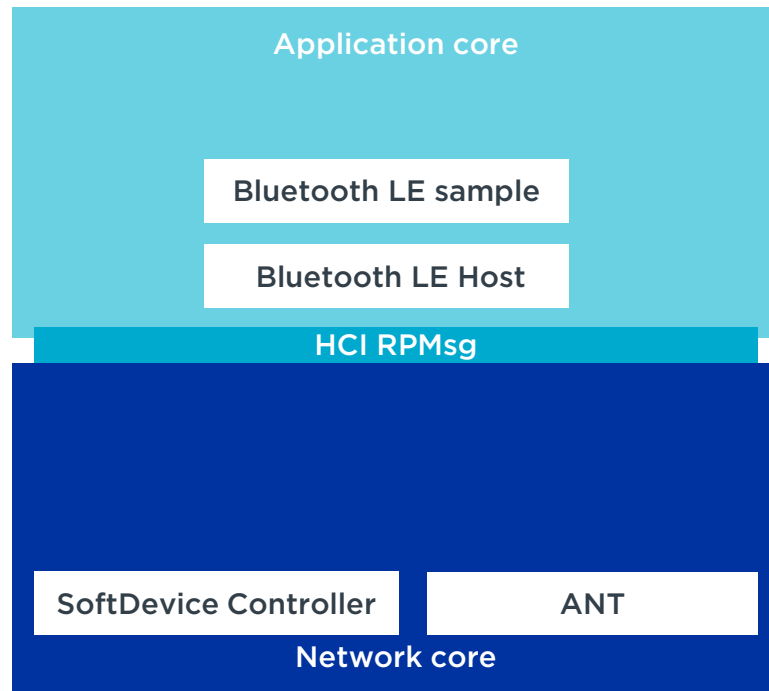


nRF Connect SDK – Bluetooth Low Energy

- Utilize our samples or samples from Zephyr
- Zephyr Bluetooth LE Host
 - Offers best integration with the Zephyr RTOS
 - Reduces complexity in the API
- SoftDevice Controller
 - Inherits the maturity and interoperability from our SoftDevices
- Nordic contribute to, test and certify the whole Bluetooth LE stack
 - QDIDs for product listings work as with SoftDevices
- nRF5340 qualified since nRF Connect SDK v1.6.0

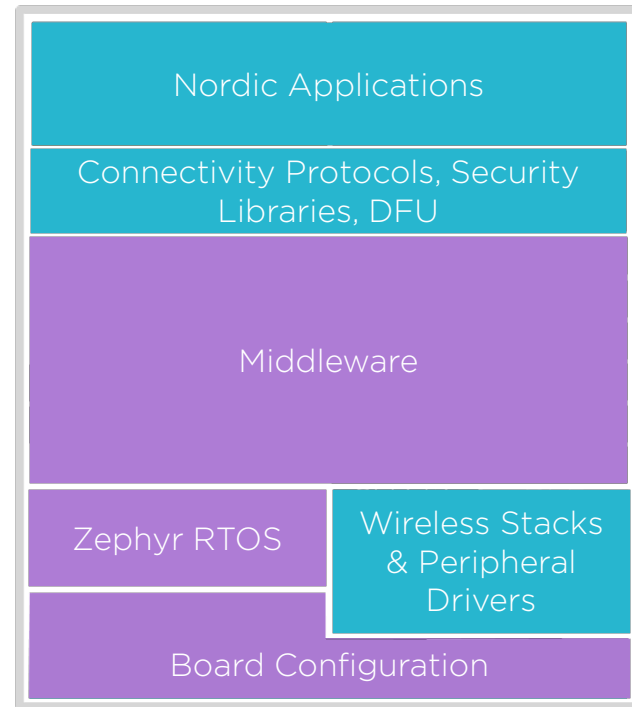


Bluetooth LE software flexibility



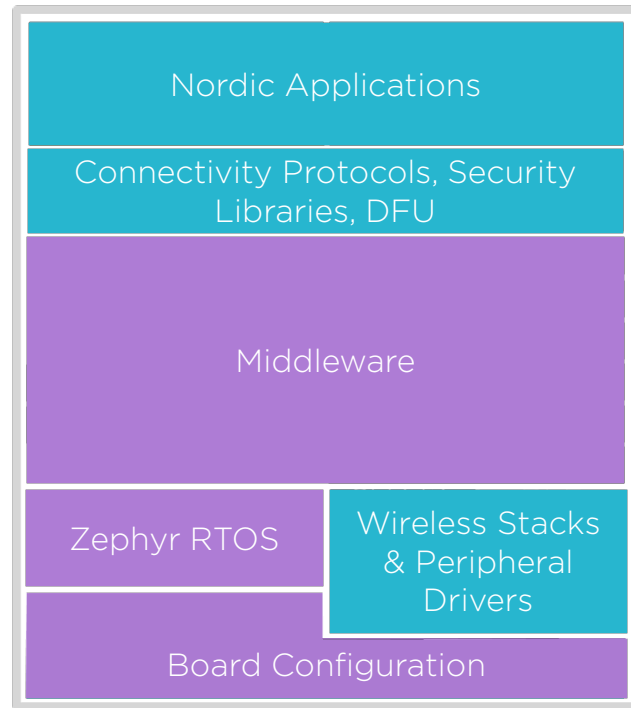
Code base

- Contains app code, connectivity protocols, wireless stacks and peripheral drivers
- Code is organized into several repositories (Nordic and Open Source (OS) code)



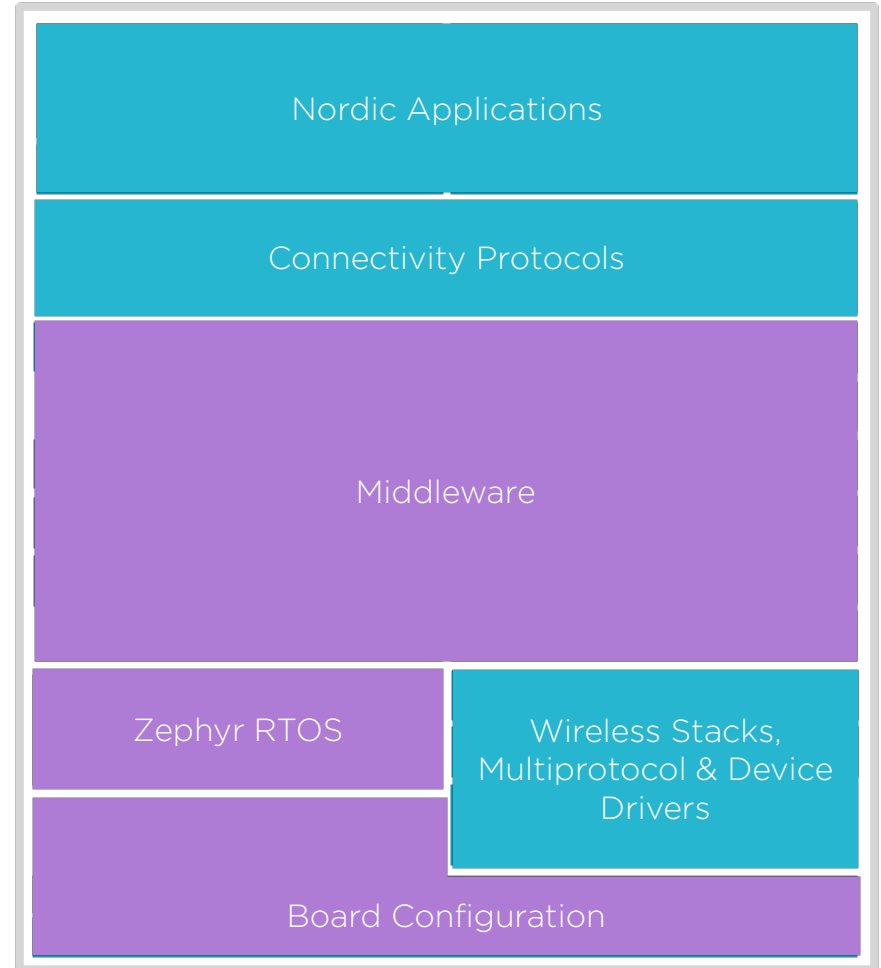
nRF Connect SDK repositories

- nRF: Application & connectivity protocols
- nrfxlib: compiled libraries where Nordic cannot distribute source code
- nrfx: peripheral drivers
- Zephyr: RTOS & board configuration (OS)
- MCUboot: Secure Bootloader (OS)
- Other repositories
 - Trusted Firmware-M, Matter, ANT etc



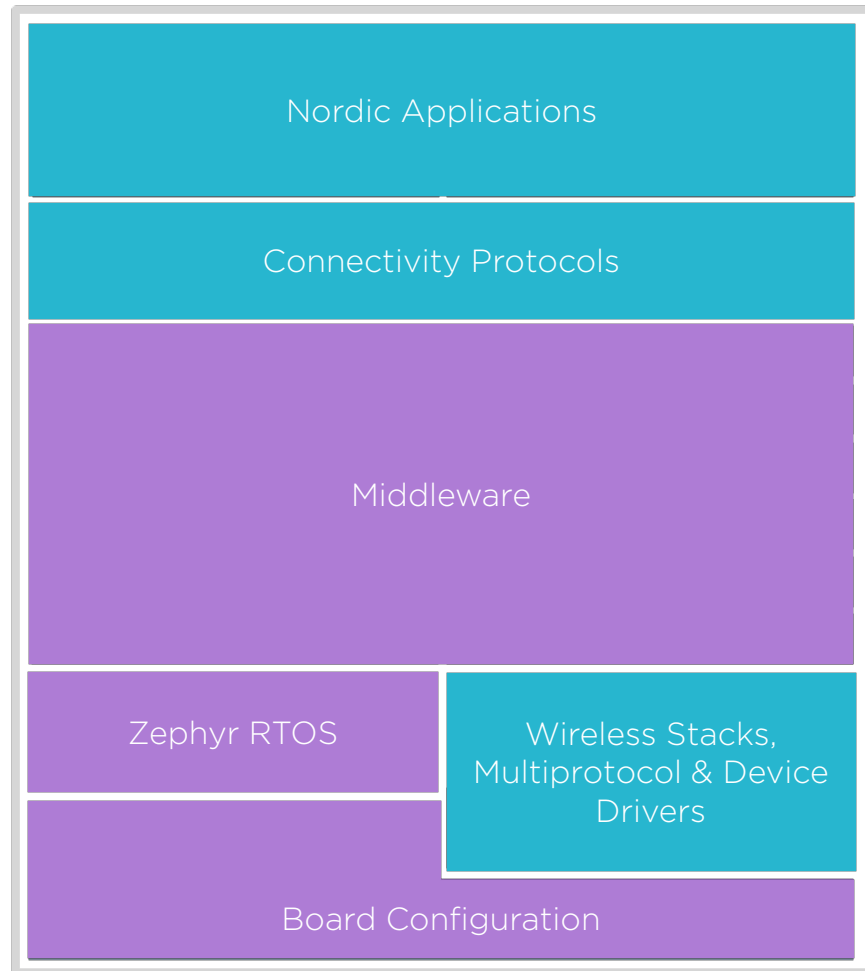
Nordic SW

- The main SDK repositories with samples, applications and connectivity stacks
- Source code exclusively written by Nordic Semiconductor
- Licensed with permissive Nordic 3-clause or 5-clause BSD license



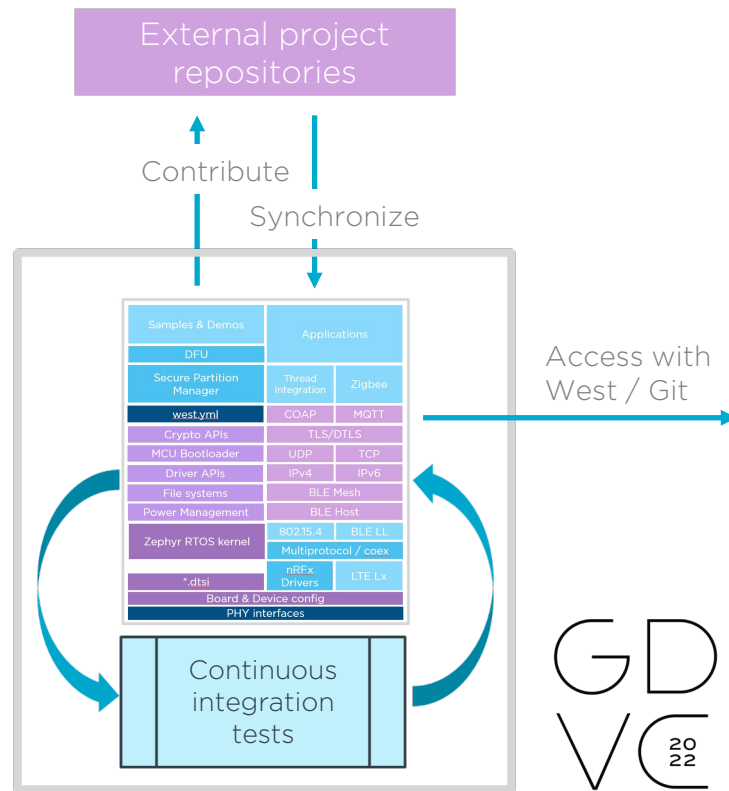
Open-source SW

- nRF Connect SDK re-distributes OS for standard platform components
- Nordic collaborates with communities of industry experts to deliver these components as part of the nRF Connect SDK
- OS repositories are licensed with permissive license (e.g. Apache 2.0)



Nordic synchronizes with external repos

- nRF Connect SDK is a single platform
- All source code is distributed by Nordic
- Includes open-source code from external projects
- Nordic contributes to, and synchronizes with external projects
- Nordic runs integration test on all source code and manages configuration
- Customers clone a tag using git and west

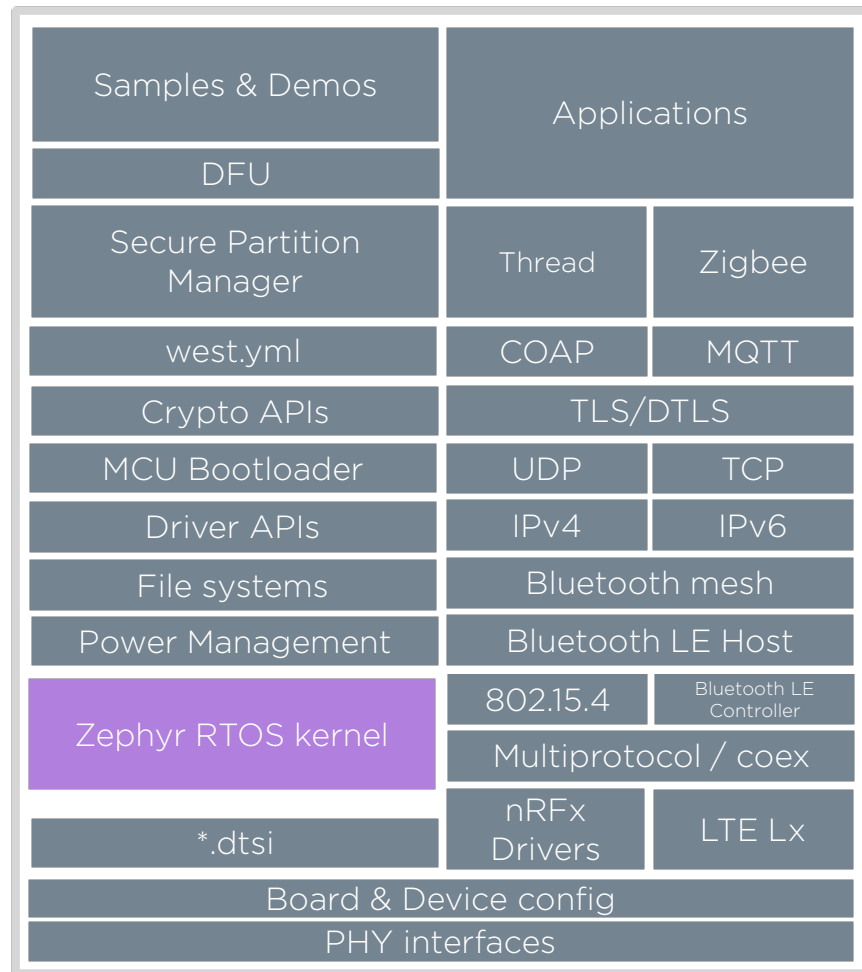


RTOS

Introduction

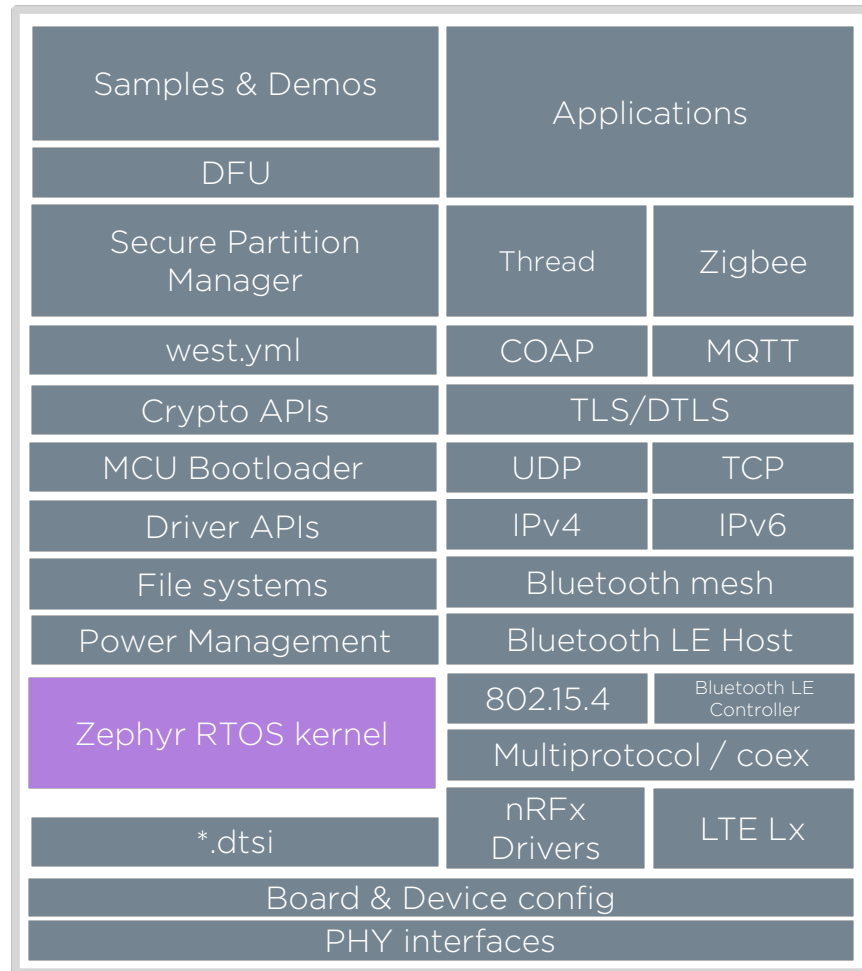
What is an RTOS?

- Real Time Operating System
 - Goal is to ensure predictable/deterministic execution pattern
 - Embedded systems often have strict timing requirements
 - Scheduler decides which task to execute at which time
 - Achievable by setting a priority for each execution thread



Why use an RTOS?

- How does an RTOS help a product developer?
 - Separation of Concern
 - More portable & re-usable applications
 - Controls application complexity in large memory devices
- Higher-level programming model
 - > faster time to market



Why the Zephyr RTOS?

- Zephyr is designed & built for low power wireless
- Zephyr is an independent technology project
 - Governed under the Linux Foundation
 - Contributions made by over 1000 industry embedded experts
- Zephyr is scalable
 - Very small configurations for memory constrained devices (min 8 kB e.g. simple LED blinking application)
 - Powerful, feature-rich, configurations for large memory, high-processing power devices (multiple MBs)
- Zephyr includes a lot of fundamental pieces of code
 - Nordic can focus on other important features



Why the Zephyr RTOS?

- Zephyr includes a lot of fundamental pieces of code
 - Nordic can focus on other important features
- Developed with a focus on security
 - Have their own Product Security Incident Response (PSIRT) team
 - PSIRT team fixes security vulnerabilities and notifies customers and public

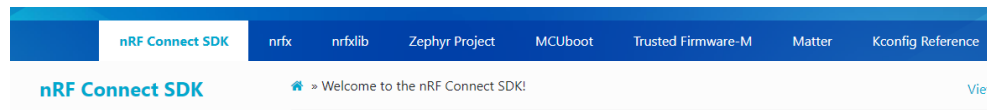


nRF Connect SDK

Details

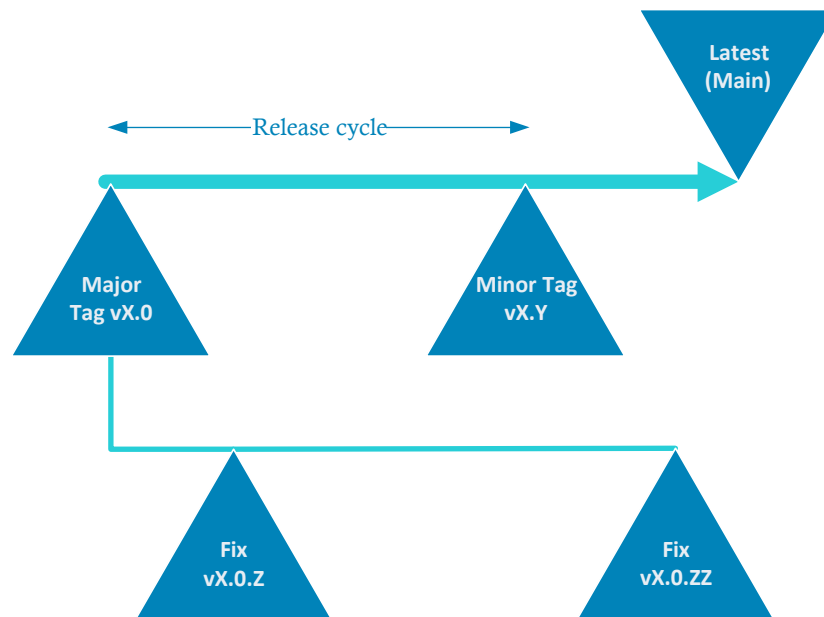
nRF Connect SDK documentation

- [Documentation link](#)
- Click on arrow in top right to choose documentation tag
 - 2.x.99 refers to main branch
 - Latest tag is 2.1.0
- Click on the tabs on the top banner to switch to nrfxlib, Zephyr, MCUboot or other documentation pages



Release cycles

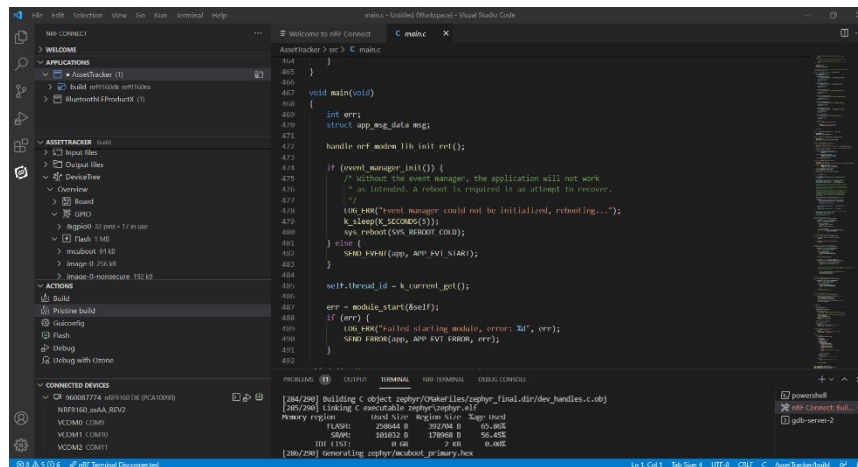
- Regular releases (e.g. quarterly)
- Publicly hosted on [GitHub](#)
- Fixes released as needed
 - Long term supported releases can have fixes applied and delivered after new releases
- Latest development version available
- Version control management with Git:
 - manage new version and fix adoption
 - tool supported merging



IDE support



- nRF Connect for Visual Studio Code
 - Built from the ground up for nRF Connect SDK
 - Highly extendable and configurable
 - CLI and GUI Interfaces
 - Cross-platform support
 - Windows, macOS, Linux
 - Create new board wizard
 - Rich set of [tutorial videos](#)

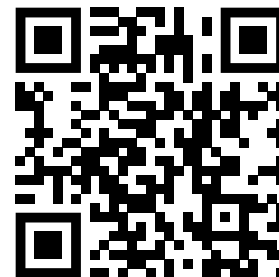


nRF Connect SDK fundamentals course

- Self-paced hands-on online course focusing on learning the essentials of nRF Connect SDK.
- Lots of hands-on exercises.
- Centralized up-to-date content.
- Protocol agnostic (cellular IoT, Bluetooth LE, Bluetooth mesh, etc..).
- Supports all our DKs and the Thingy:91.
- Ideal for new users of nRF Connect SDK or users switching from nRF5 SDK to nRF Connect SDK.
- Test your knowledge through interactive quizzes.



DevAcademy



Complete development solution

nRF5340 SoC



Multi-core flexibility

Bluetooth 5.3

Matter

Next level security

nRF Connect



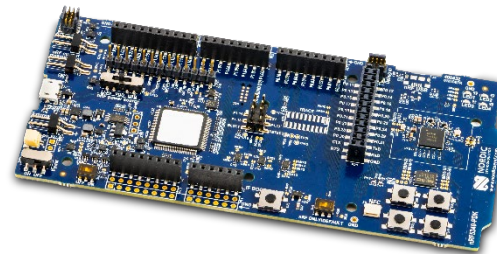
nRF Connect SDK

nRF Connect for Desktop

nRF Connect for Mobile

nRF Cloud

nRF5340 DK



nRF5340 SoC development kit

supports all features of the

nRF5340 SoC

More Information

- www.nordicsemi.com/nrf5340
- www.nordicsemi.com/thingy53
- <https://webinars.nordicsemi.com/meet-the-nrf5340-the-next-1>
- <https://academy.nordicsemi.com/>
- www.nordicsemi.com/nrfconnects_dk