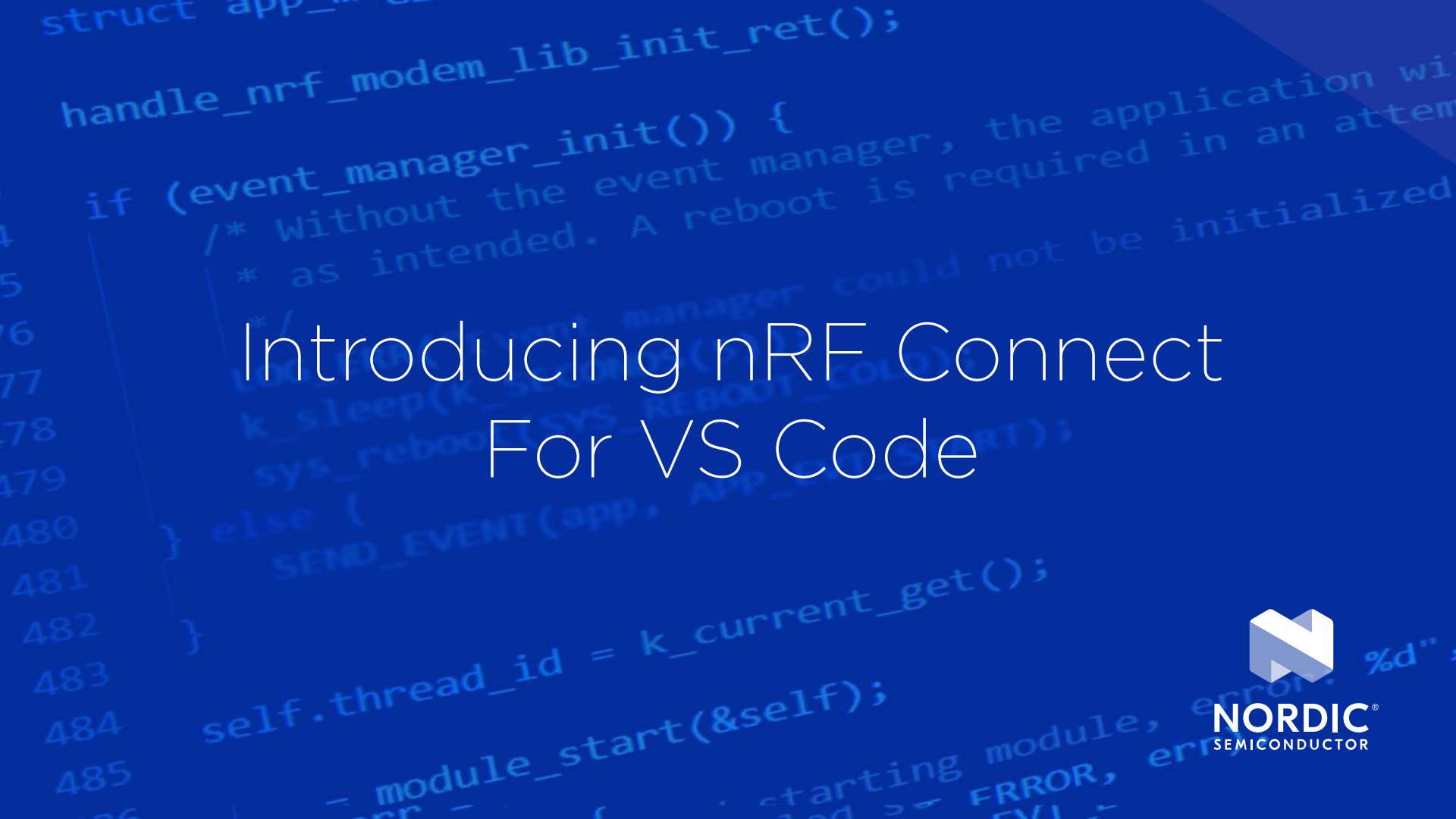
Introducing nRF Connect For VS Code



NORDIC® SEMICONDUCTOR

Today's Speaker

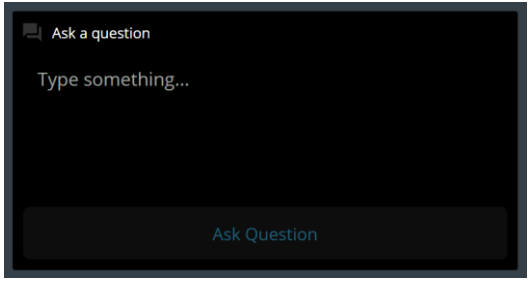
Ali Aljaani

Product Marketing Engineer

NORDIC® SEMICONDUCTOR

Practicalities

- Duration: 50-60 mins
- Questions are encouraged!
- Please type questions in the top of the right sidebar
  - All questions are anonymous
  - Try to keep them relevant to the topic
- We will answer questions towards the end
- The chat is not anonymous, and should **not** be used for questions
- If you have more questions, Go to Nordic DevZone
- A recording of the webinar will be available together with the presentation at webinars.nordicsemi.com



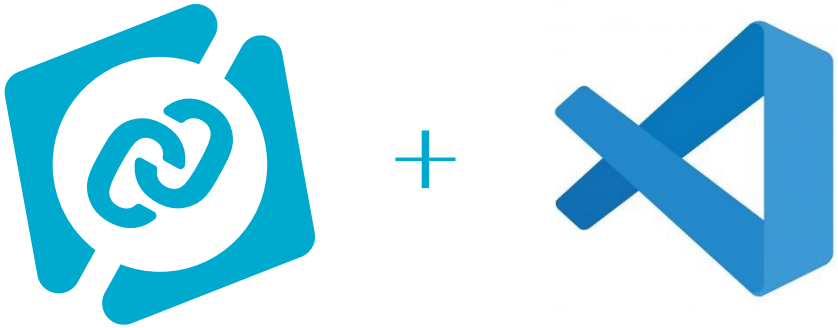
{ DevZone

Agenda

- Introduction
- Getting started with nRF Connect for VS Code
- Hands-on
  - Best practices to increase your coding productivity
  - Creating your own custom board
  - Working with multiple boards
  - Managing repositories
- Q&A

Introduction

- Built from the ground up for nRF Connect SDK
- A modern, easy to use IDE
- Highly extendable and configurable
- CLI and GUI interfaces in one place
- Create new board wizard
- Rich set of tutorial videos
- Cross-platform support
  - Windows, macOS, Linux

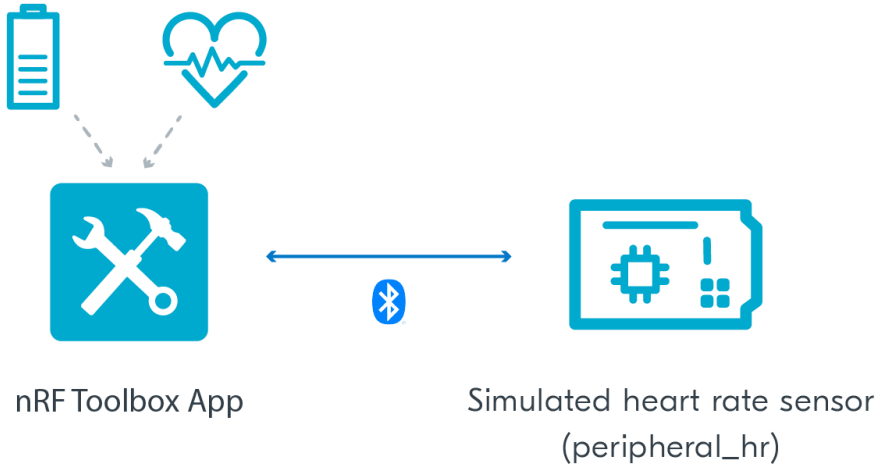


Get Started

- Watch the installation video in the tutorial series to install nRF Connect for VS Code. (If not already installed!)
- Evaluate functionality on the fly
  - Create a freestanding application based on templates
    - **Samples** showcase a single feature or library
    - **Applications** include a variety of libraries to implement a specific use case
  - Filter templates by
    - Wireless Communication Protocol
    - Device (SoC or SiP)
    - Others(crypto, drivers, etc..)
  - Try it out on a connected board

Heart Rate Peripheral Sample

- Turns your development kit into a simulated heart rate sensor
- Simulated heart rate readings and battery levels are sent over Bluetooth LE
- Any Bluetooth LE central (Smartphone, Tablet, etc..) can receive the simulated date
- Use the *nRF Toolbox* app to visualize the date received



Heart Rate Peripheral Sample

Development Kit Name	Board ID
nRF5340 DK	nrf5340dk_nrf5340_cpuappns
nRF52840 DK	nrf52840dk_nrf52840
nRF52833 DK	nrf52833dk_nrf52833
nRF52 DK	nrf52dk_nrf52832
nRF9160 DK*	nrf9160dk_nrf52840

*Switch SW10 to nRF52 to program the nRF52840 on the nRF9160 DK

Best Practices

- Organize your source code
- Utilize VS Code features
  - Customize it the way it best fits you
  - Use the power of key bindings
    - Keyboard shortcuts
    - Migrate from others like vim, emacs (@recommended:keymaps)
  - Utilize the powerful editor
  - Script repetitive tasks

Organize Your Source Code

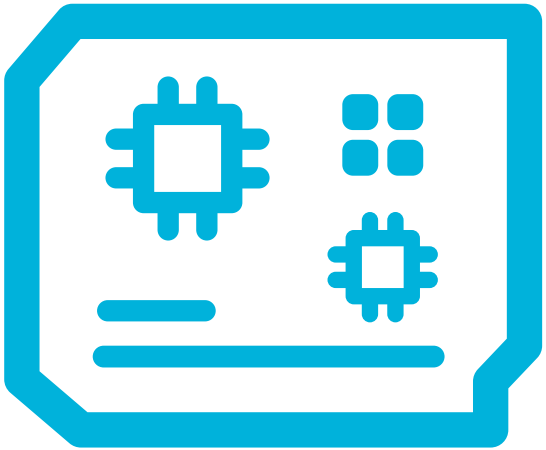
- Split and organise your source code into multiple files
- All files that your application uses must be specified in the CMakeLists.txt file
- By default, most samples include only the main application file src/main.c
- You must add all other files that you are using manually. Ex:

```
target_sources(app PRIVATE src/customfolder/customfile.c)
zephyr_include_directories(src/customfolder)
```

- You could also use wildcard to include several files in CMakeLists.txt

Creating Your Own Board

- Welcome page
  - Create a new board
- Name your board
- Select Nordic SoC/SiP used on your board
- Specify a directory to store your board's files
  - In your application directory
  - In-tree within the SDK <nRF Connect SDK>\zephyr\boards\arm



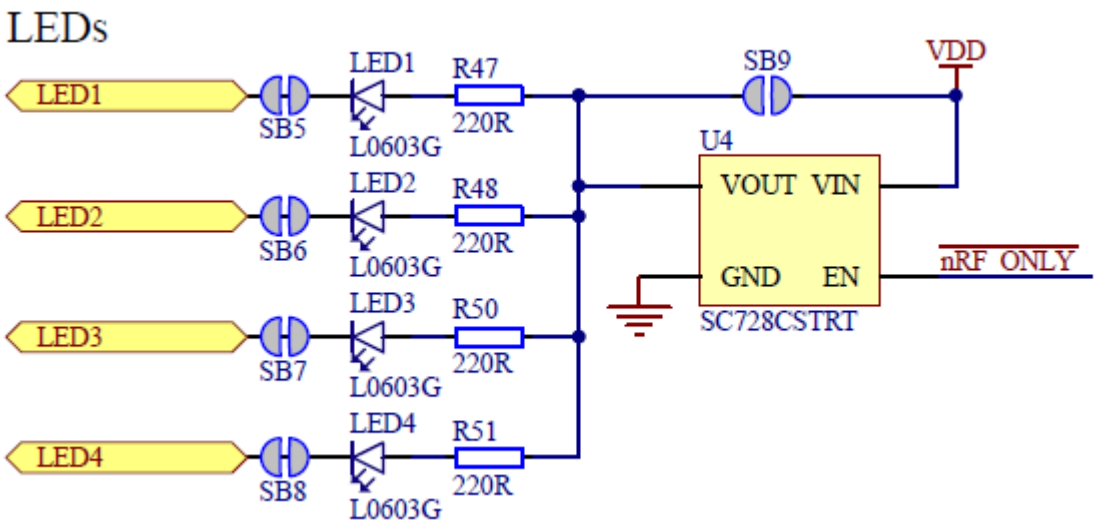
Board's Core Files

- *plank.dts*: Hardware description of your board in devicetree format.
- *Kconfig.board, Kconfig.defconfig, plank_defconfig*: Software configuration in Configuration System (Kconfig) formats.
- *board.cmake :* For flash and debug support. No action is needed.

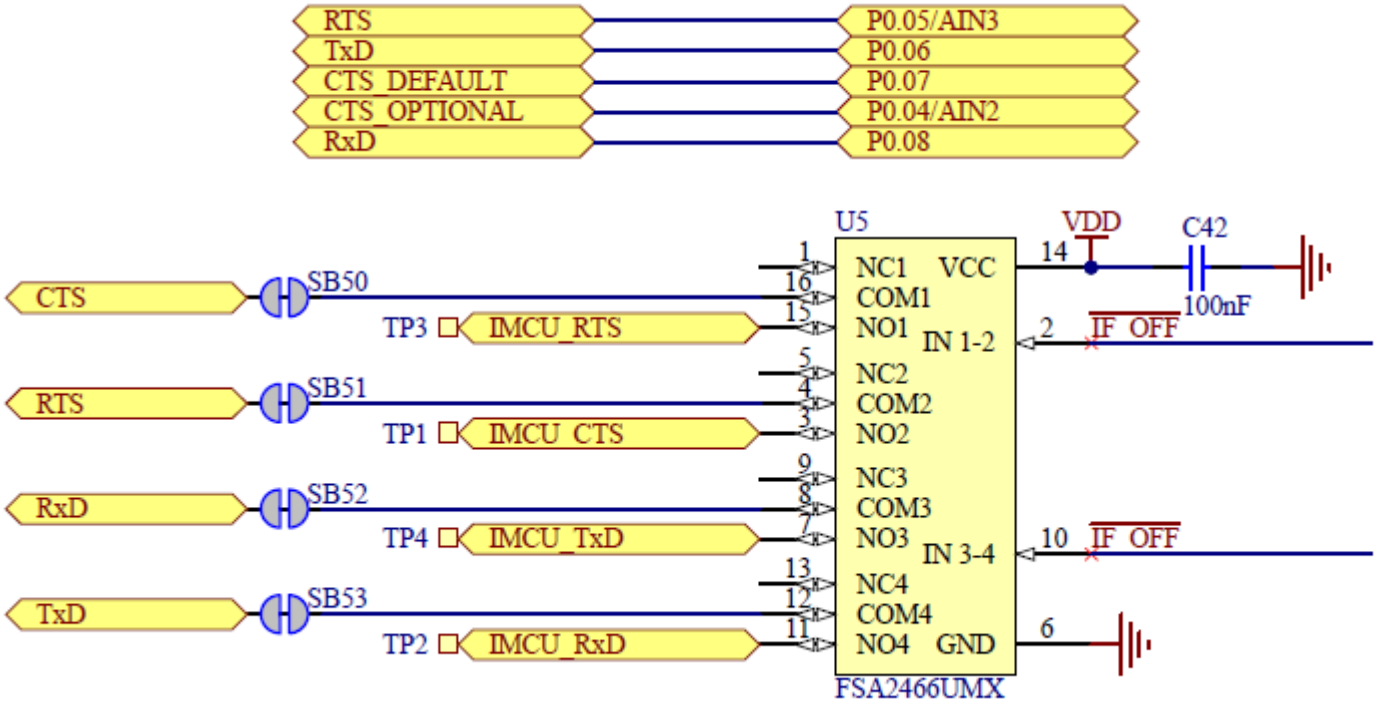
Board's Optional Files

- *CMakeLists.txt:* if you need to add additional source files to your build.
- *doc/index.rst*, *doc/plank.png*: documentation for and a picture of your board. You only need this if you're contributing your board to Zephyr.
- *plank.yaml*: a YAML file with miscellaneous metadata used by the Test Runner (Twister).

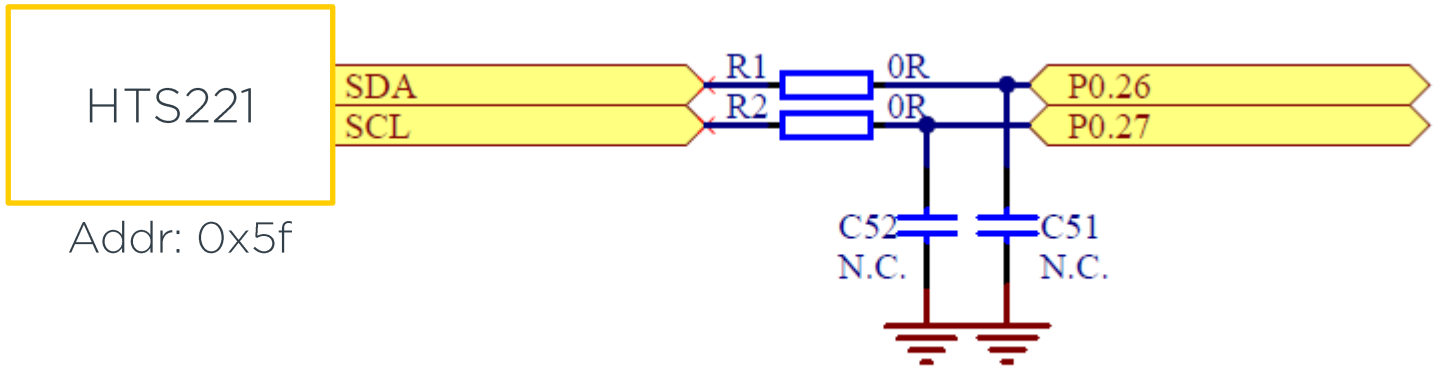
LEDs



UART over USB



Sensor



Addr: 0x5f

The HTS221 is an ultra-compact sensor from ST for relative humidity and temperature.

Writing Your Own *.dts File

- Have your schematics handy
- Start with the *compatible* property
  - This drives IntelliSense auto complete
- New nodes should be defined under / node
- Modify already defined nodes through the & operator
- Remember last assignments override old assignments

Working With Multiple Boards

- Name your boards
- Links builds to board(s)
- Create a separate Kconfig fragment per build (if sharing the same application)
- Use the power of scripting!

Version Control in VS Code

- Built-in within VS Code
- Create your own local repo
- Create remote repository
- Push to remote repository
  - Exclude build directories
- Pull from a remote repository
- And many more inside the IDE.

Resources for Creating Custom Board

- Zephyr's Board Porting Guide
- Devicetree Specification
- Devicetree Syntax and structure
- nRF52833 DK hardware files

Resource for Creating Tasks in VS Code

- Integrate VS Code with External Tools via Tasks

Q&A

Register for upcoming Nordic Tech Webinars

www.nordicsemi.com/webinars