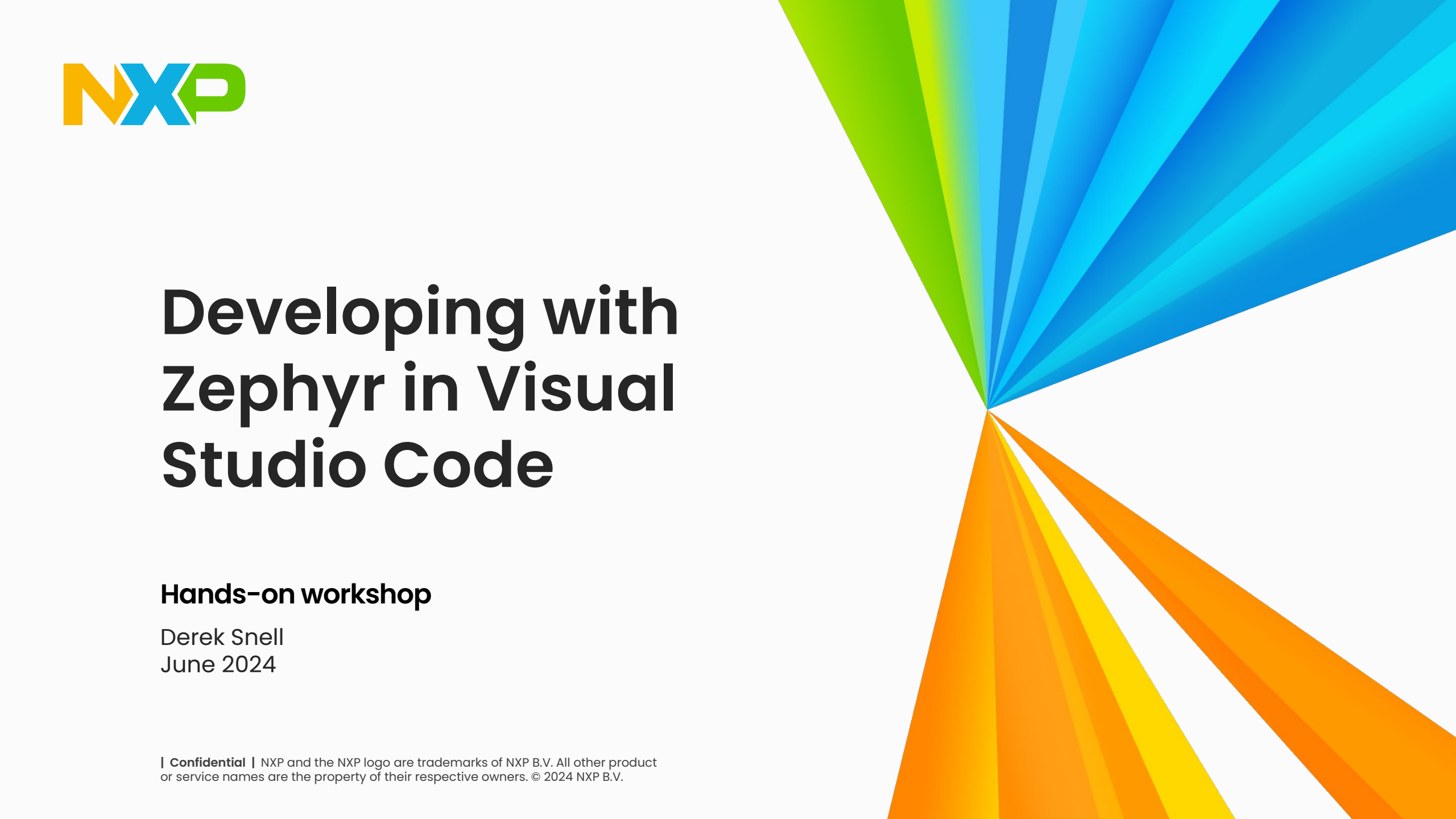


Developing with Zephyr in Visual Studio Code

Hands-on workshop

Derek Snell
June 2024

Agenda

- MCUXpresso extension for VS Code

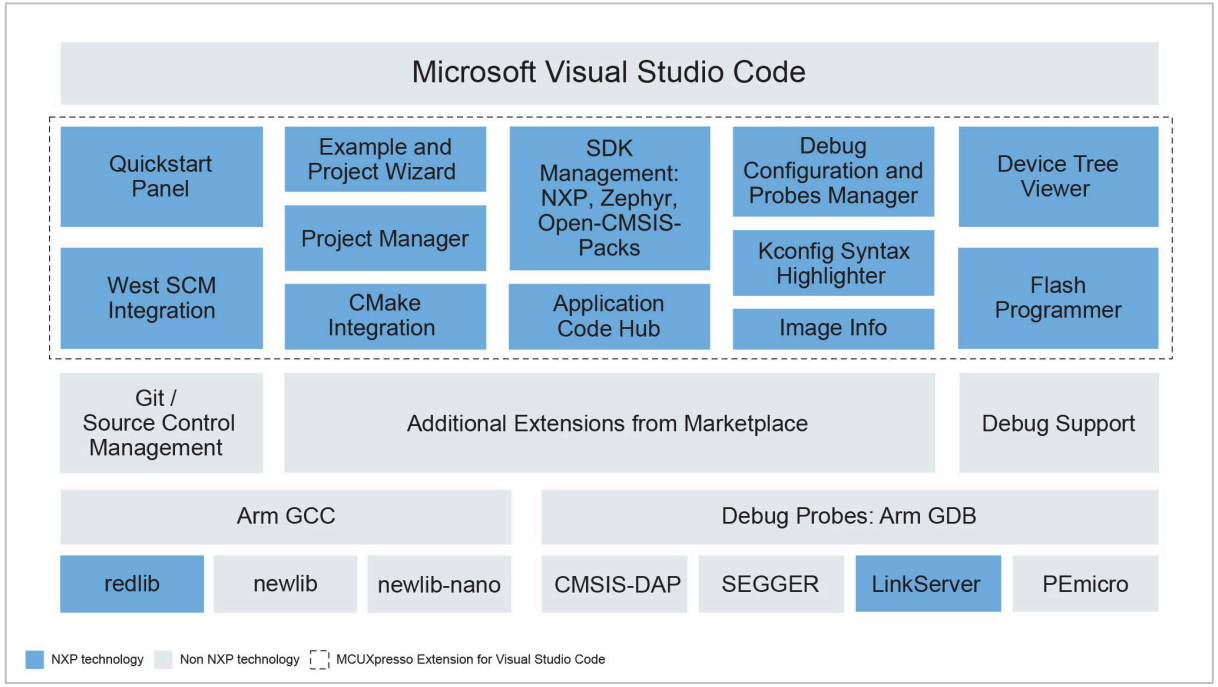
- Zephyr:
  - Portability
  - Devicetree
  - Kconfig
  - Hands-on Labs

01

MCUXpresso for VS Code

MCUXpresso for Visual Studio Code

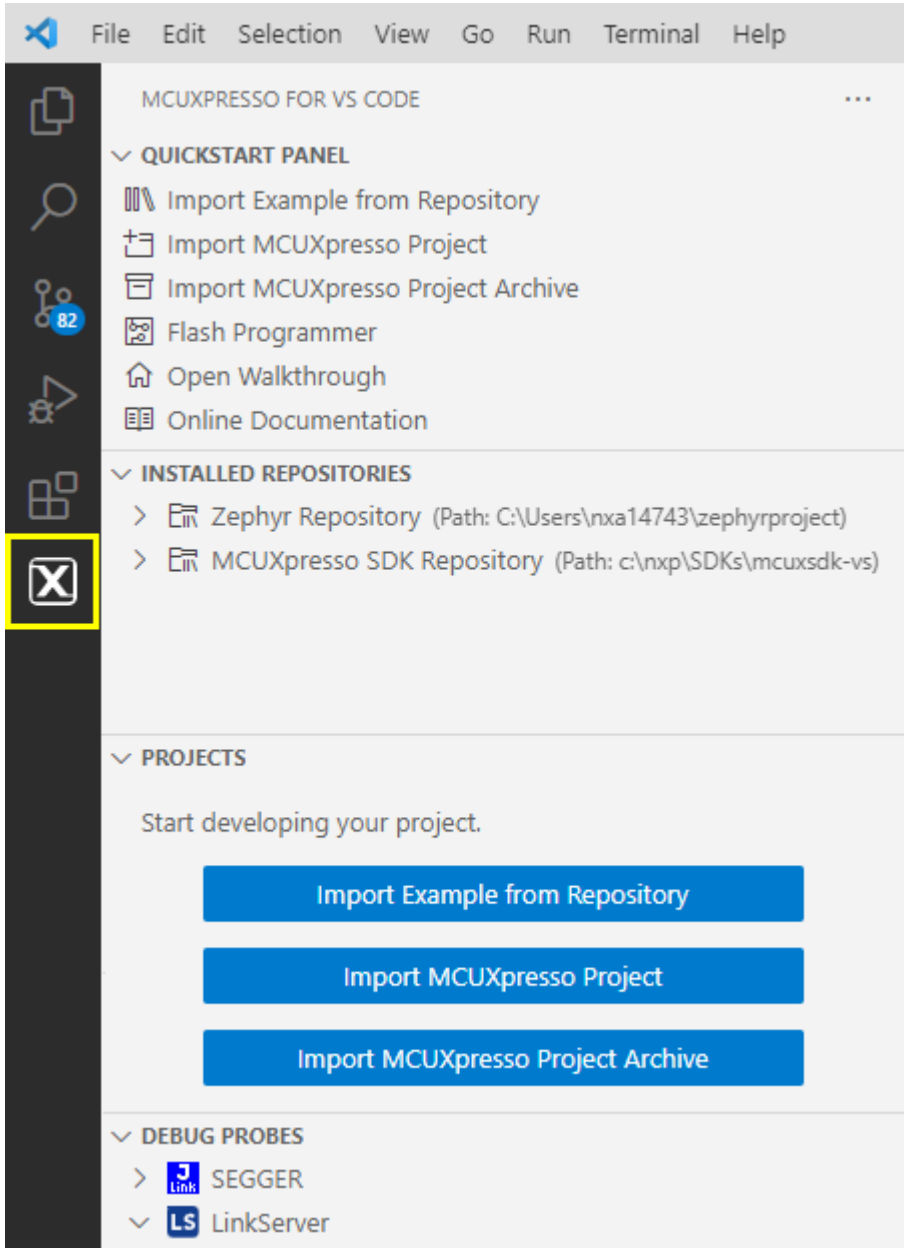
- Free to use
- Optimized for ease-of-use
- Extended for MCUXpresso SDK or Zephyr based development flows
- Based on Arm GCC compiler
- Compatible with standalone MCUXpresso Configuration tools



Introduction to Visual Studio Code

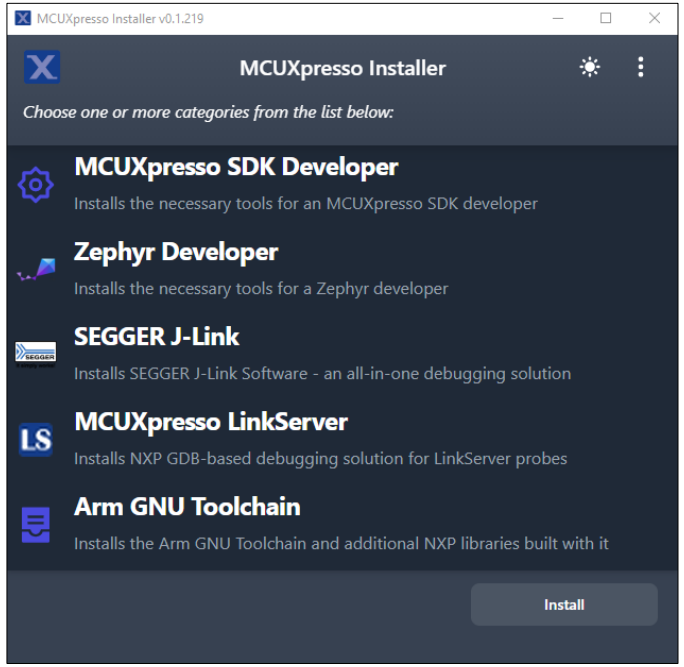
- NXP developed MCUXpresso Extension
  - Installed from VS Code Extensions Marketplace
  - Improves NXP development experience

- Featured Improvements:
  - MCUXpresso Installer
  - Quickstart Panel
  - Zephyr & MCUXpresso Applications Importer
  - Editing
  - Build & Debug



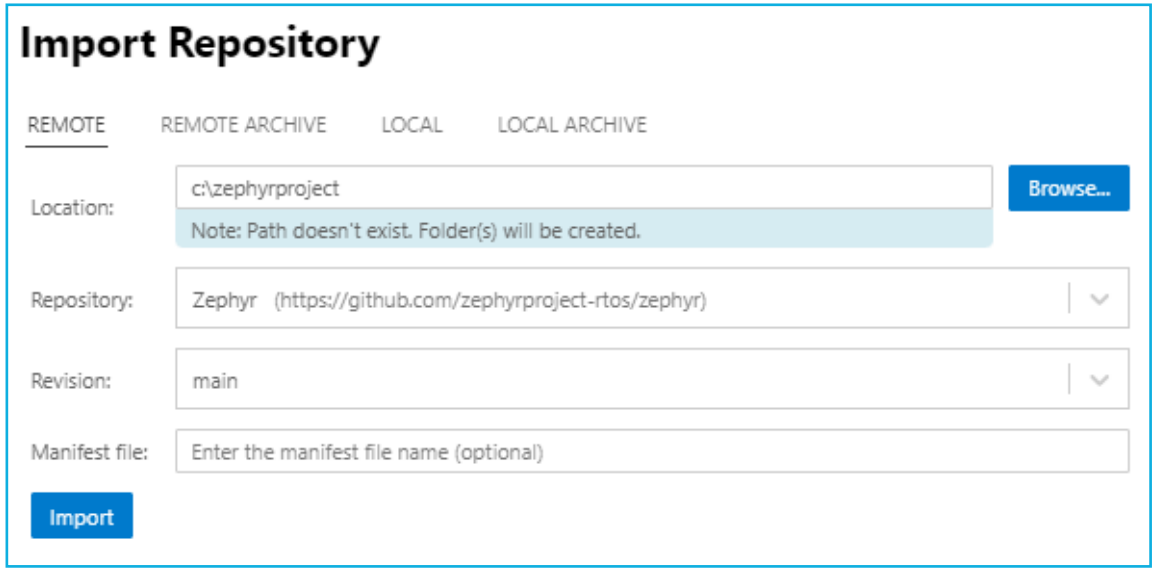
MCUXpresso for VS Code: Installation

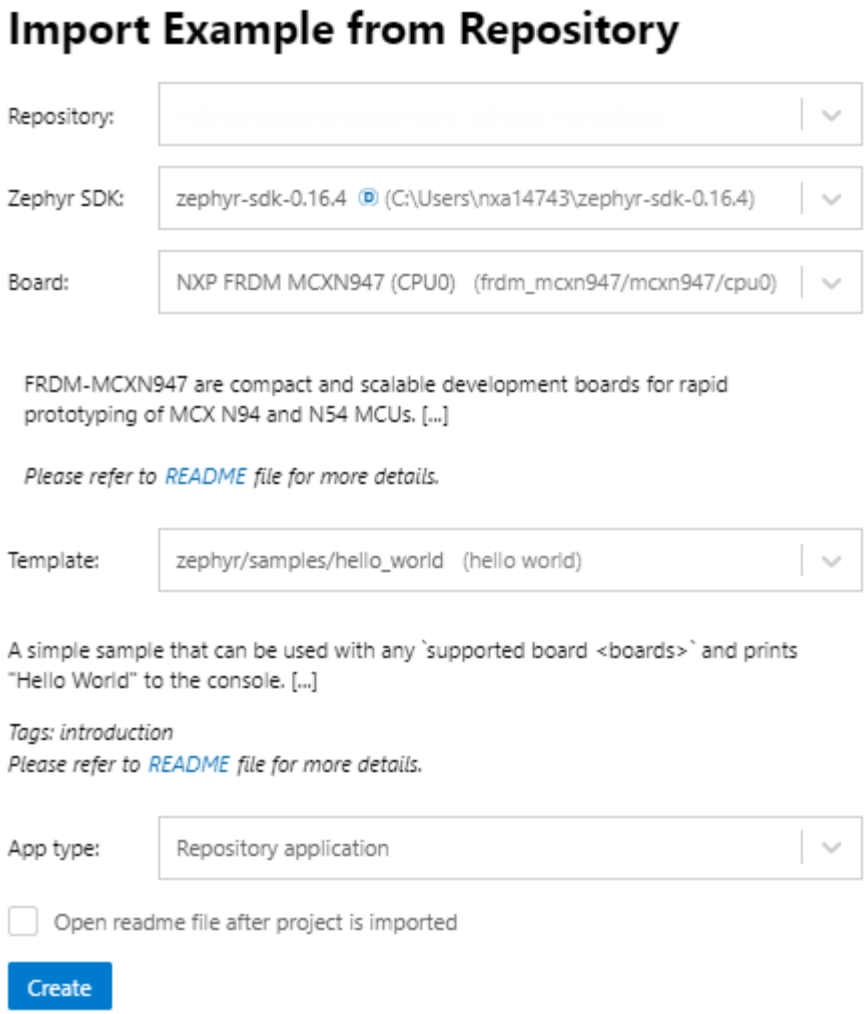
- Installer handles toolchain dependencies
  - Git
  - Python
  - West
  - Segger
  - LinkServer
  - Arm® GNU
  - Zephyr SDK
  - Cmake
  - Ninja
  - Devicetree compiler
  - other Zephyr dependencies



MCUXpresso for VS Code: Zephyr Applications

- Imports Zephyr and dependency repositories
- Imports Zephyr sample applications
  - Choose board to target
- Device Tree Viewer
- Zephyr debugger features:
  - Thread Awareness
  - Stack usage



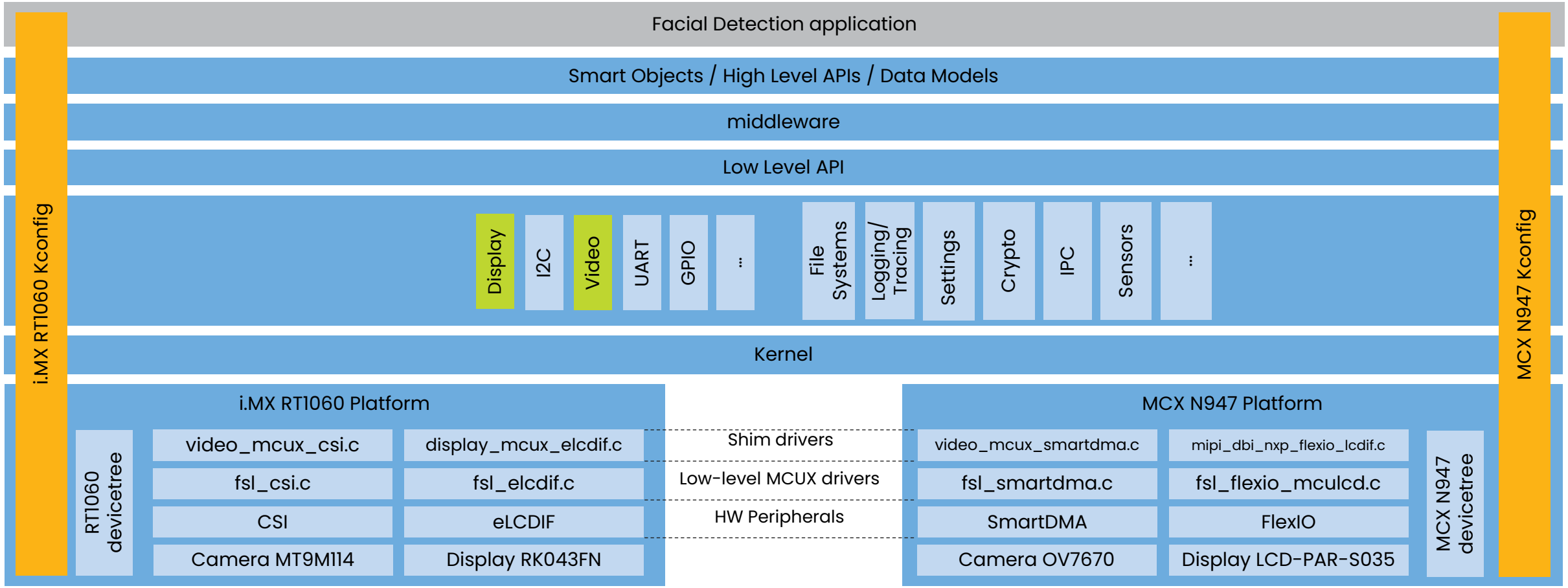


02

Zephyr Portability

Hardware Abstraction with devicetree and Kconfig

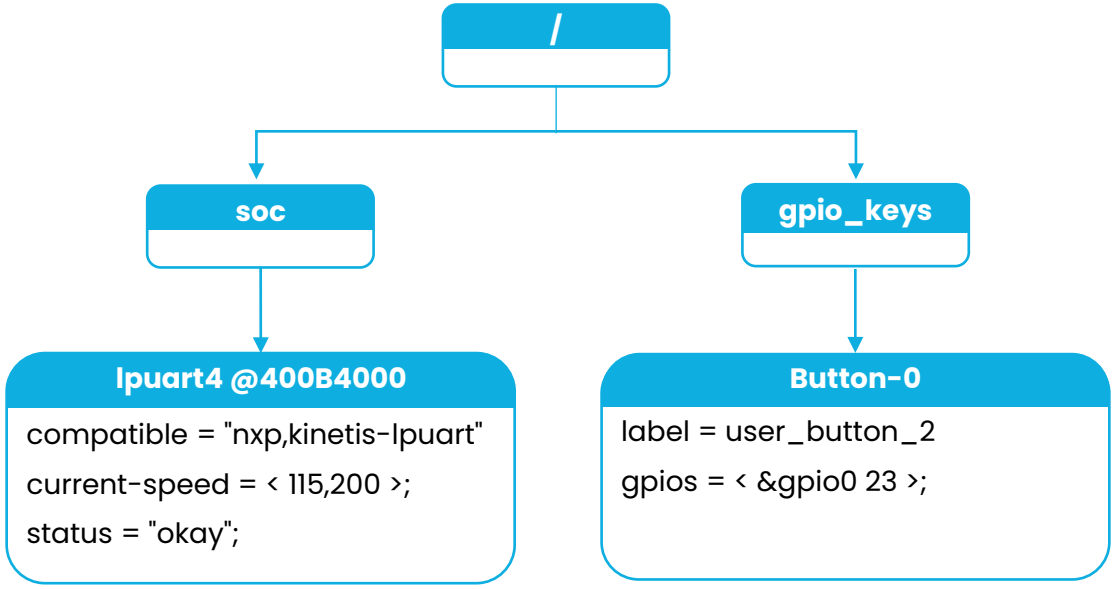
- Application on both mimxrt1060_evk and frdm_mcxn947 boards
  - Same application, same kernel and upper layers, but different HAL controlled by device tree and Kconfig settings



DeviceTree Overview – from FRDM-MCXN947 board

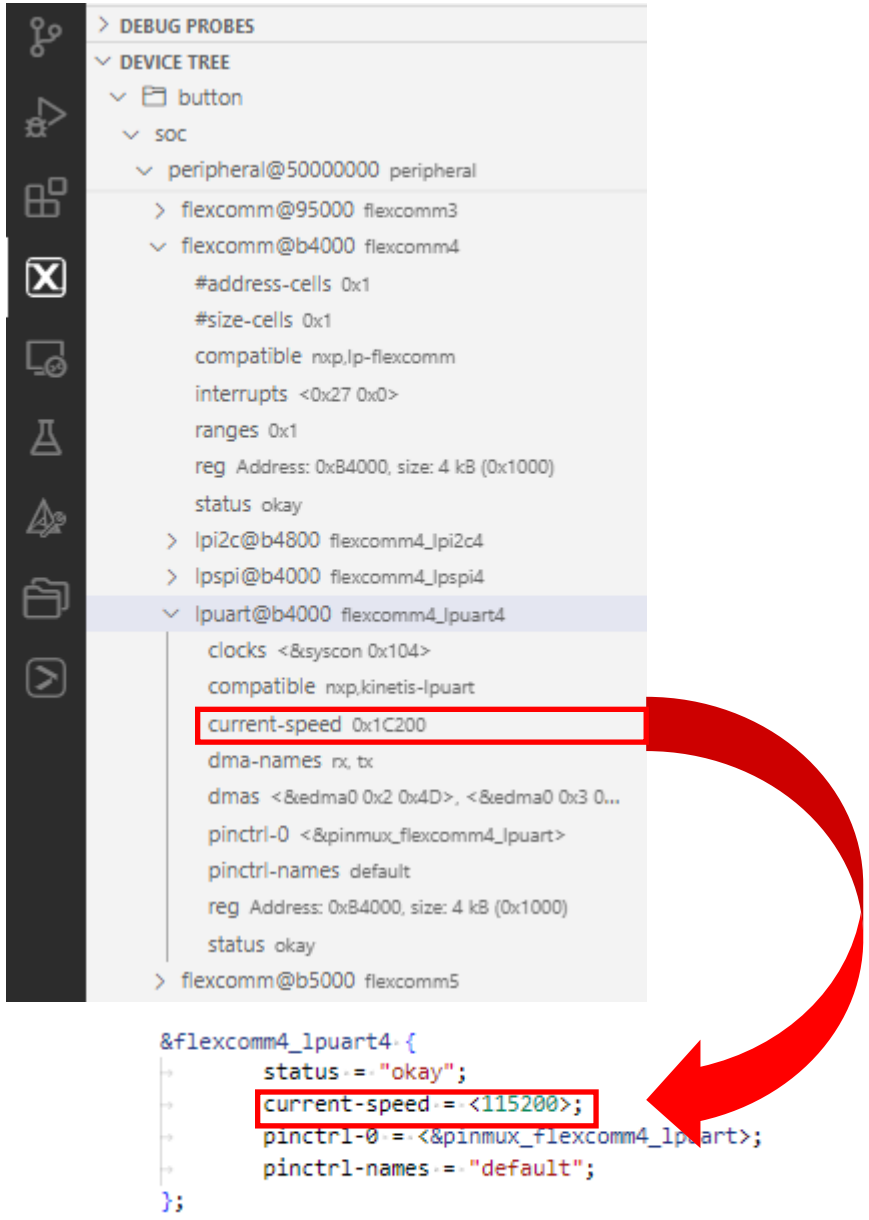
- Nodes correspond to some hardware component
  - Nodes are hierarchical, and can reflect the hardware's physical layout
- Properties will describe or configure the hardware node, such as a UART's baud rate

```
/dts-v1/;

/ {
    soc {
        flexcomm4_lpuart4: lpuart@b4000 {
                compatible = "nxp,kinetis-lpuart";
                reg = < 0xb4000 0x1000 >;
                status = "okay";
                current-speed = < 115200 >;
                pinctrl-0 = < &pinmux_flexcomm4_lpuart >;
                pinctrl-names = "default";
        };
         . . .
    };
    gpio_keys {
        compatible = "gpio-keys";
        user_button_2: button_0 {
                label = "User SW2";
                gpios = < &gpio0 23 (GPIO_ACTIVE_LOW | GPIO_PULL_UP)>;
                status = "okay";
        };
    };
};
```



MCUXpresso Device Tree Viewer

- VS Code extension includes a Device Tree Viewer
- Displays all nodes and properties included in the application build
- Clicking a node or property in the Viewer finds the location in the source file



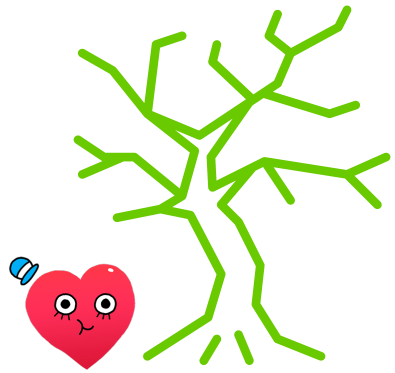
*The Viewer is showing the baud rate in hex: 0x1C200 = 115,200

Advantages of using Device Tree

- Fully describes all enabled HW in system for the application
  - Describes dependencies between nodes, HW hierarchy fully described

- Application does not need to include HW-specific files, or call HW-specific APIs

- No overhead
  - Uses preprocessor macros to bind Device Tree to drivers
  - No additional memory footprint used over the device driver model



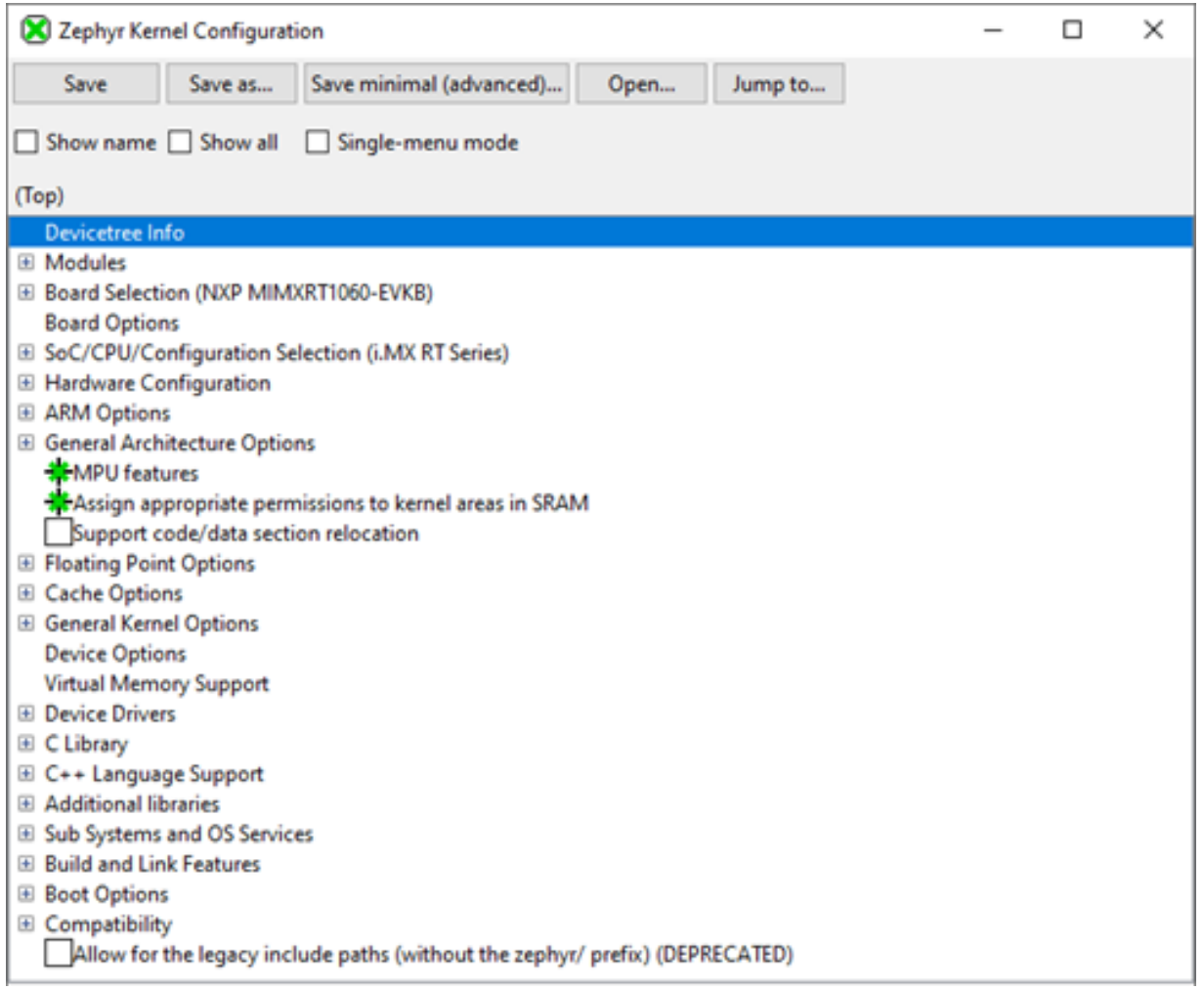
Kconfig – Kernel Configuration system

- Zephyr uses the same configuration system used by the Linux kernel
- The goal is to support configuration without having to change any source code
- For this compiler optimization example:
  - Applications are given a choice of options
  - Application selects for size like this:
    `CONFIG_SIZE_OPTIMIZATIONS=y`
- Another example:
  - App can set stack size for Main thread:
    `CONFIG_MAIN_STACK_SIZE=1024`

```
choice COMPILER_OPTIMIZATIONS
        prompt "Optimization level"

config SIZE_OPTIMIZATIONS
        bool "Optimize for size"
        help
          Compiler optimizations will be set to -Os.

config SPEED_OPTIMIZATIONS
        bool "Optimize for speed"
        help
          Compiler optimizations will be set to -O2.

config DEBUG_OPTIMIZATIONS
        bool "Optimize debugging experience"
        help
          Compiler optimizations will be set to -Og.

config NO_OPTIMIZATIONS
        bool "Optimize nothing"
        help
          Compiler optimizations will be set to -O0.

endchoice
```

Interactive Config tools

- Guiconfig is optional tool for viewing and changing Kconfig settings:



Guiconfig helps browse and explore the Kconfig settings

When to configure using Devicetree or Kconfig

Whether to use devicetree or Kconfig for a particular purpose:

- Use devicetree to describe hardware and boot-time configuration
  - Peripherals on a board, boot-time clock frequencies, interrupt lines, etc.
- Use Kconfig to configure software support to build into the final image
  - Where to add network support, which drivers are needed

- Kconfig for software, devicetree for hardware

- An example, a device with 10/100 Ethernet MAC
  - devicetree:
    - Describe the presence of the Ethernet MAC
    - Compatible drivers
    - Hardware configuration, such as IEEE1588 support and interrupts
  - Kconfig:
    - Determines which software features should be built, such as selecting the Networking stack and supported protocols

Summary of Zephyr build steps

VS Code calls these build tools when a project is built

- Scripts merge device tree for build
  - Uses sources from board and SOC, plus overlay files

- Scripts merge kernel config (Kconfig) options
  - From sources: board-specific config, application config, and CMake entries

- Generates build files using Cmake

- Compiles source code

- Links all libraries, and then final application

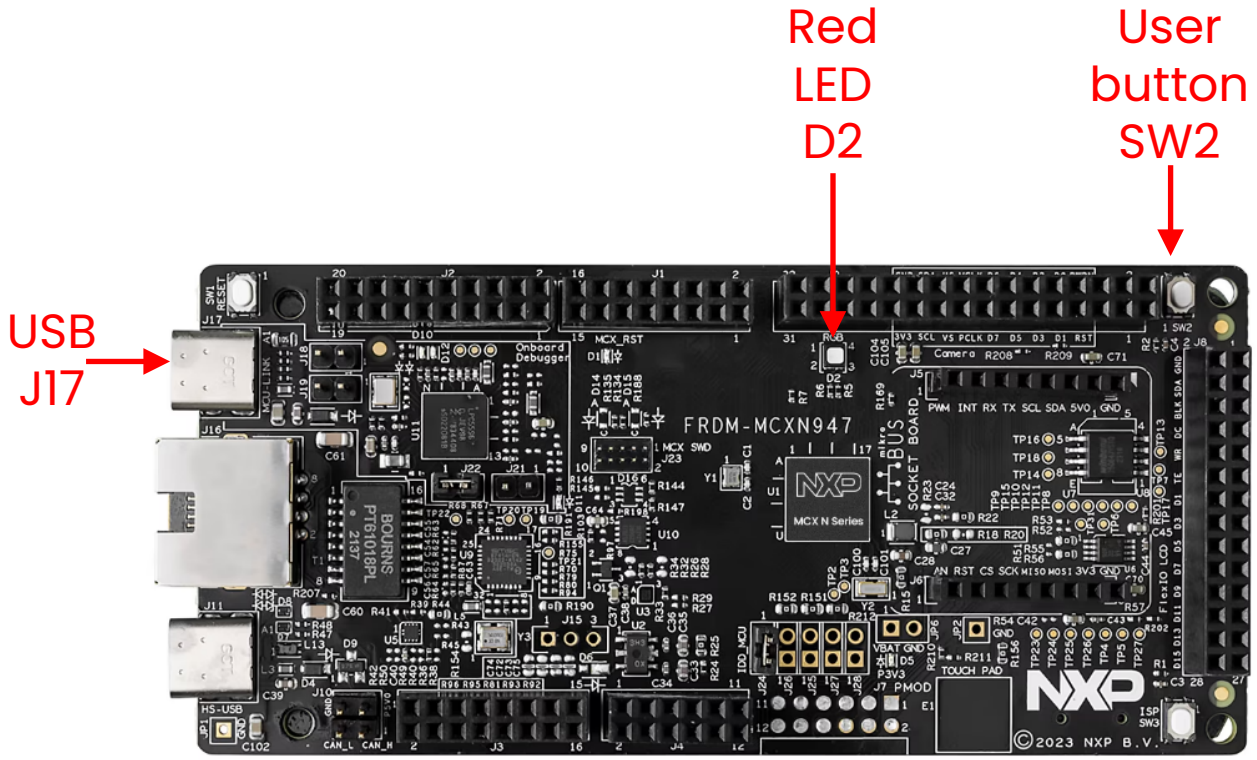
03

Hands-On Labs

Lab Pre-Requisites

- Hardware
  - FRDM-MCXN947
  - USB Type-C cable
  - On-board MCU-Link debugger

- Software and Tools:
  - Zephyr post v3.6.0 release (main branch)
  - VS Code with MCUXpresso extension
  - See *AMF-ENT-T4788 Hands-On Pre-requisite Developing with Zephyr in Visual Studio Code.pdf*



Zephyr Hands-on Lab Guides

1. Hello_World
   - Import sample, build, debug

2. Kconfig
   - Explore and modify Kconfigs, guiconfig tool

3. Debugging with VS Code
   - Thread Awareness, Stack Usage, thread call stack

4. Devicetree
   - MCUXpresso Device Tree Viewer, board overlay files, modifying hardware configuration

Zephyr On-Demand Webinar Series
www.nxp.com/zephyr Training

- Series of customer webinars introducing Zephyr and highlighting NXP-supported platforms
- Includes demonstrations (but not hands-on training or lab tutorials)

- **NXP and Zephyr OS: Unlocking Innovation with an Open Source RTOS**
  - Maureen Helm (NXP)
  - Kate Stewart (Zephyr Project)
- **Application Portability with Zephyr OS and NXP**
  - Derek Snell, Senior Systems Engineer (NXP)
- **Accelerate Development with Zephyr OS Features and Modules**
  - Derek Snell, Senior Systems Engineer (NXP)
- **NXP and SEGGER: Debug, Visualize and Analyze Zephyr OS Applications with Ease**
  - Derek Snell, Senior Systems Engineer (NXP)
  - Axel Wolf (SEGGER)
- **Should I Care About Zephyr OS? – Real Experiences of an RTOS Expert**
  - Brendon Slade, MCU Ecosystem Director (NXP)
  - Eli Hughes, Pro Support Engineer
- **OTA Device Management with Golioth Cloud Platform Using Zephyr RTOS and NXP i.MX RT**
  - Mike Szczys, Developer Relations Engineer (Golioth)
  - Derek Snell, Senior Systems Engineer (NXP)
- **Designing for Low Power Using Zephyr® Project: A Smartwatch Example**
  - Derek Snell, Senior Systems Engineer (NXP)

Additional Resources - Zephyr

- Zephyr:
  - NXP Zephyr page
  - Zephyr Getting Started web page
  - Zephyr Community and Support
  - GitHub
  - Zephyr's Board Porting Guide
  - Samples and Demos
  - Supported boards
  - Beyond the Getting Started Guide
  - Zephyr Documentation

- Kconfig:
  - Zephyr Kconfig documentation
  - guiconfig and menuconfig interactive Kconfig tools
  - Sources of Kconfig symbols, and how they are merged

- Devicetree:
  - Zephyr Devicetree documentation
  - Zephyr Devicetree HOWTOs
  - Devicetree specifications

Additional Resources – MCUXpresso and VS Code

- MCUXpresso for VS Code
  - Wiki pages
  - Community Support Forum
- MCUXpresso Ecosystem
- VS Code
  - Download
  - Debugging
  - Embedded Tooling RTOS View

nxp.com