



Sensors
Converge

A Hands-On Introduction to The Zephyr Project RTOS

June 20–22, 2023 | Santa Clara, CA

#SensorsConverge

The Speaker

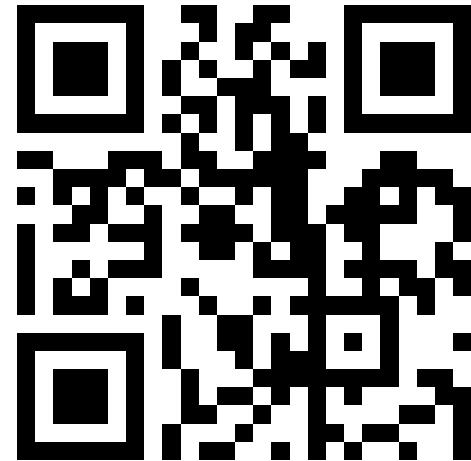
Mohammed Billoo
(www.mab-labs.com)



www.linkedin.com/in/mab-embedded

[@mabembedded](https://twitter.com/mabembedded)

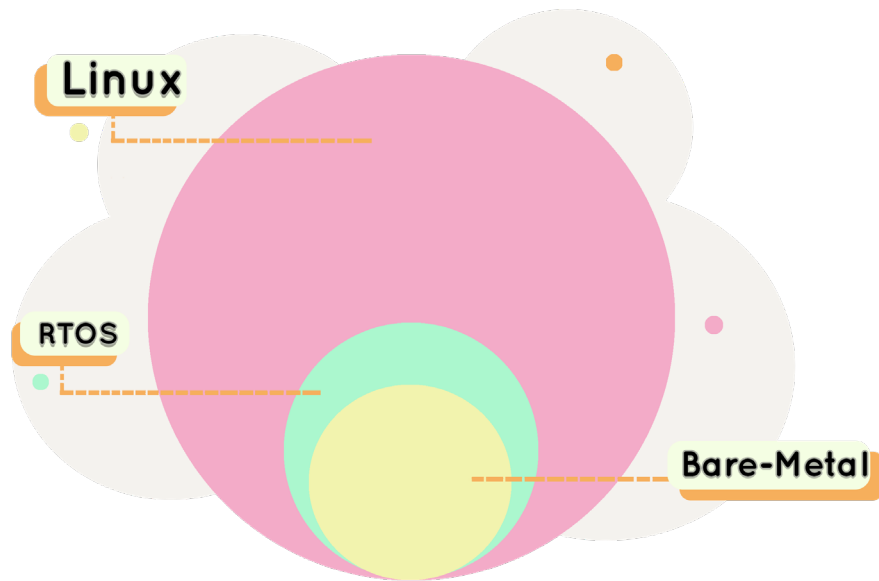
BIOS Food Newsletter



Agenda

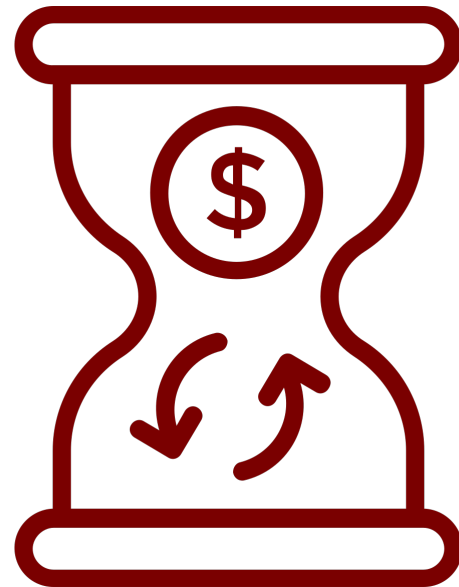
- Why?
- Getting Started
- Unique Features
- What's Next?
- Q&A

Why?



RTOS vs Bare Metal

- An RTOS Saves Development Time
- Minimal additional code size
- Serves as a HAL to MCU synchronization features
- Avoids requiring us to implement these features
 - Reinvent the wheel
 - Time-consuming
- Tested thoroughly



RTOS Benefits

- Don't reinvent the wheel
- API that implements standard features
 - Task/thread
 - Queue
 - Semaphore
 - Mutex
- Leverage MCU hardware
 - ARM
 - Assembly to enable/disable interrupts
- Leverage compiler for increased optimization



www.eringillespiecartoons.wordpress.com

Why The Zephyr Project RTOS?

- Self-contained
 - Drivers included
 - Subsystems included
 - No need to manually retrieve other repositories
- Vendor agnostic
 - Support for CPUs from many vendors
 - Support for extensive list of development boards
- Examples for all subsystems/peripherals
- VS Code Plugins



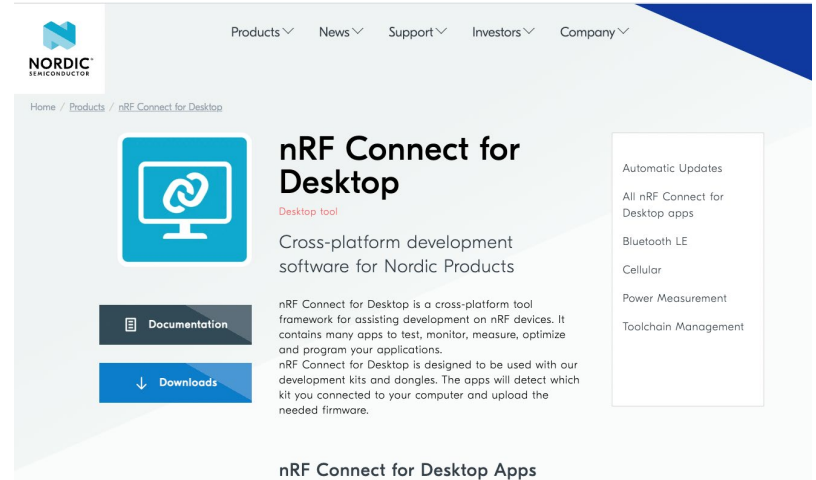
The Zephyr Project RTOS

- It's not Linux!
- “Zephyr”
- It's an RTOS
 - Direct access to hardware
 - No rings/privileged mode*
- Organization is different from other RTOSes
 - Collection of repositories
 - Uses a “meta-tool” called West



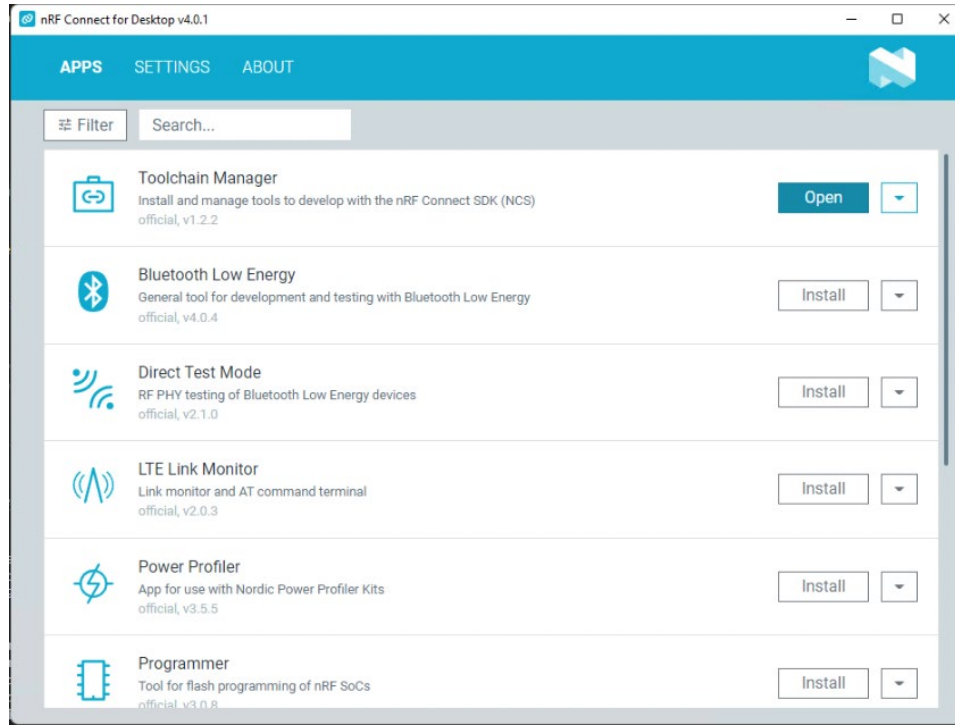
Getting Started

- nRF Connect for Desktop
- An application that sets up a complete Zephyr development environment
- VS Code Plugins!
 - Create a new project
 - Based on an example
 - Open existing project
 - Configure Zephyr
 - Build and debug a project

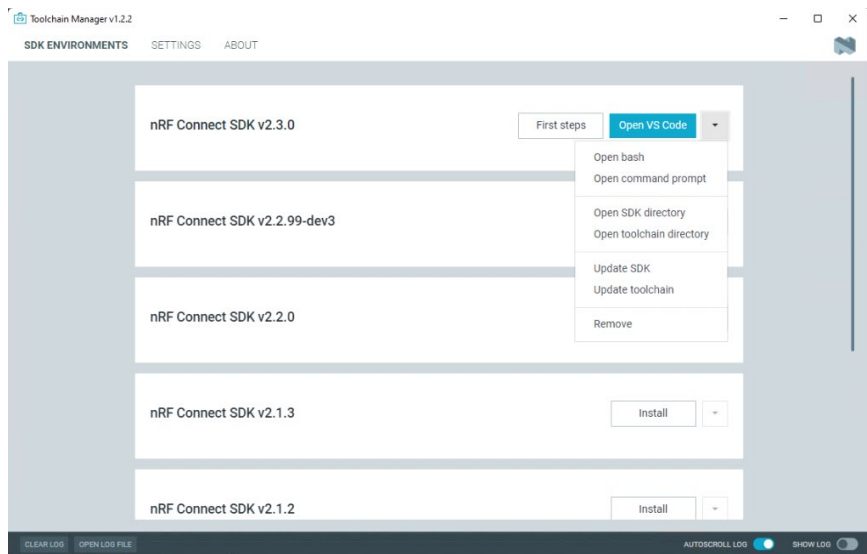


The screenshot shows the product page for nRF Connect for Desktop on the Nordic Semiconductor website. The page features the Nordic Semiconductor logo in the top left, a navigation menu with links for Products, News, Support, Investors, and Company, and a breadcrumb trail: Home / Products / nRF Connect for Desktop. The main content area includes a large blue icon of a monitor with a Zephyr logo, the product name 'nRF Connect for Desktop' in bold, and the subtitle 'Desktop tool'. Below this is a description: 'Cross-platform development software for Nordic Products'. A 'Documentation' button and a 'Downloads' button are visible. A detailed description of the tool's capabilities is provided, along with a list of supported features: Automatic Updates, All nRF Connect for Desktop apps, Bluetooth LE, Cellular, Power Measurement, and Toolchain Management. The page also includes a section for 'nRF Connect for Desktop Apps'.

nRF Connect For Desktop



nRF Connect For Desktop



nRF Connect for VS Code

Quick Setup

The nRF Connect extension requires the nRF Connect toolchain to be present.

You can use the [Toolchain Manager](#) from [nRF Connect for Desktop](#) to manage your nRF Connect SDK installations. Alternatively, you can follow the [manual installation instructions](#).

Once installed, set the default nRF Connect version and toolchain here or in the [extension settings](#) (User Settings → Extensions → nRF Connect). The default settings act as a fallback if the current workspace does not override them.

The following settings apply to the [User Settings](#) scope.

nRF Connect SDK

2.3.0 (c:\ncs\v2.3.0) Install...

Changes saved.

nRF Connect Toolchain

2.3.0 (c:\ncs\toolchains\v2.3.0) Refresh

Helpful Links

- Open walkthrough...
- Open Nordic DevAcademy...
- Open an existing application...
- Create a new application...
- Create a new board...

Hello World!

nRF Connect for VS Code

Quick Setup ▾

The nRF Connect extension requires the nRF Connect toolchain to be present. You can use the [Toolchain Manager](#) from [nRF Connect for Desktop](#) to manage your nRF Connect SDK installations. Alternatively, you can follow the [manual installation instructions](#).

Once installed, set the default nRF Connect version and toolchain here or in the [extension settings](#) (User Settings ⇒ Extensions ⇒ nRF Connect). The default settings act as a fallback if the current workspace does not override them.

The following settings apply to the [User Settings](#) scope.

nRF Connect SDK ⓘ

2.3.0 (c:\ncs\v2.3.0) Install...

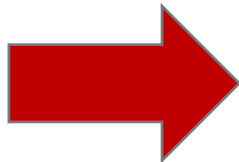
Changes saved.

nRF Connect Toolchain ⓘ

2.3.0 (c:\ncs\toolchains\v2.3.0) Refresh

Helpful Links

- Open walkthrough...
- Open Nordic DevAcademy...
- Open an existing application...
- Create a new application...**
- Create a new board...



New Application

Application type ⓘ

Freestanding Workspace

Freestanding applications require and use a locally installed nRF Connect SDK. [More information](#)

nRF Connect SDK ⓘ

2.3.0 (c:\ncs\v2.3.0) Install...

nRF Connect Toolchain ⓘ

2.3.0 (c:\ncs\toolchains\v2.3.0) Refresh

Application location ⓘ

c:\Users\mbilloo\ecoc2023 ...

Application template ⓘ

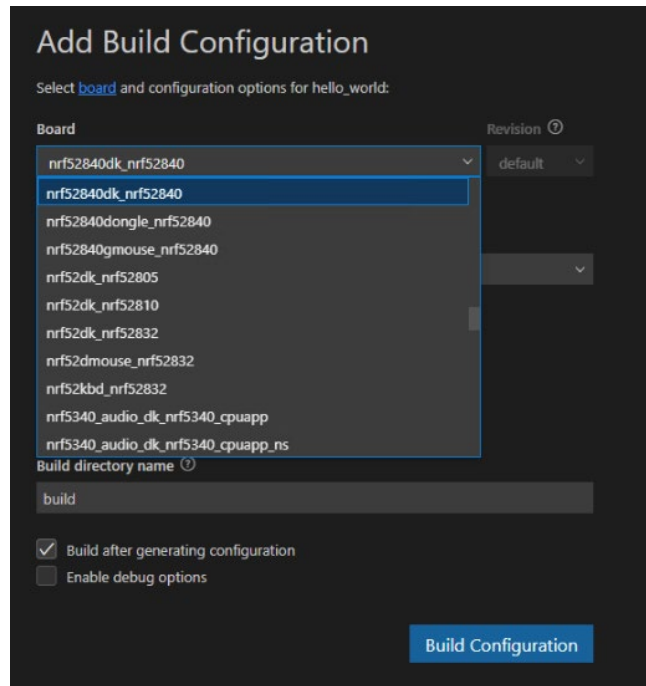
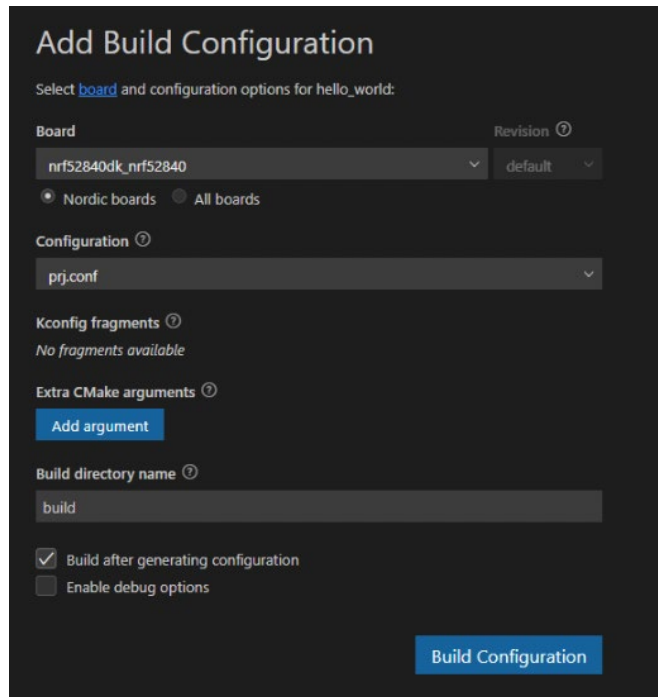
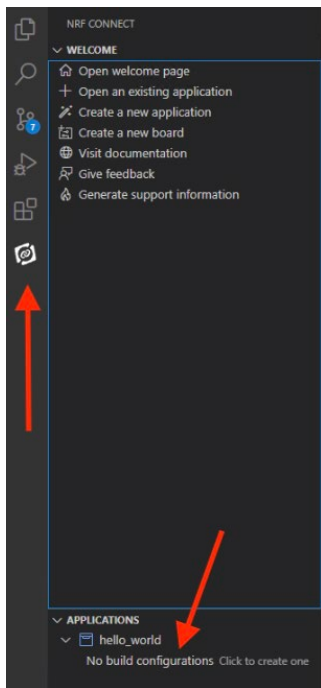
zephyr/samples/hello_world Browse...

Application name ⓘ

hello_world

Create Application

Hello World (nRF Connect Plugin)



Hello World!

Add Build Configuration

Select [board](#) and configuration options for hello_world:

Board Revision ⓘ
nrf52840dk_nrf52840 default

Nordic boards All boards

Configuration ⓘ
prj.conf

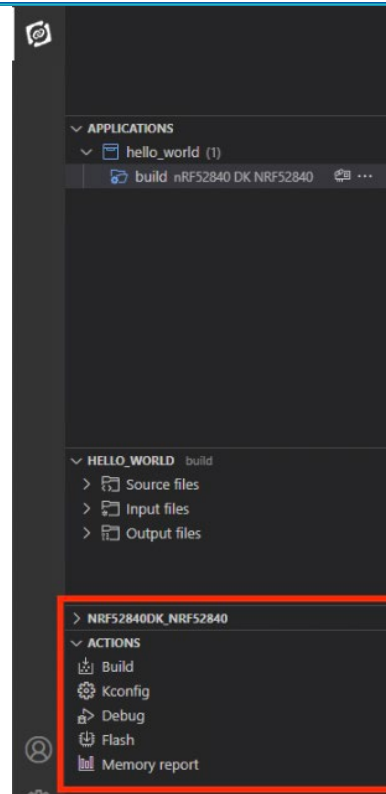

Kconfig fragments ⓘ
No fragments available

Extra CMake arguments ⓘ
[Add argument](#)

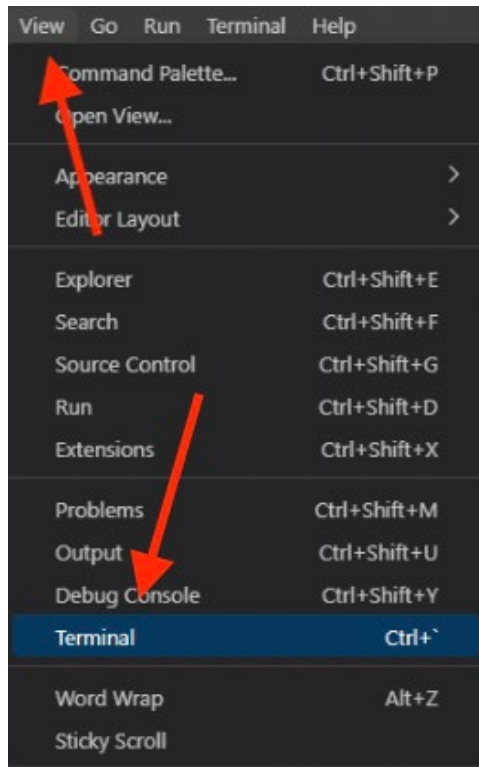
Build directory name ⓘ
build

Build after generating configuration
 Enable debug options

[Build Configuration](#)



Hello World!



A screenshot of the Visual Studio Code terminal window. The terminal shows the output of a build process for a 'hello_world' project. The output includes building C objects and linking the executable. A table shows memory usage for FLASH, RAM, and IDT_LIST. The terminal also displays messages about reusing the terminal and executing a task.

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL NRF TERMINAL

[165/168] Building C object zephyr/CMakeFiles/zephyr_final.dir/misc/empty_file.c.obj
[166/168] Building C object zephyr/CMakeFiles/zephyr_final.dir/dev_handles.c.obj
[167/168] Building C object zephyr/CMakeFiles/zephyr_final.dir/isr_tables.c.obj
[168/168] Linking C executable zephyr\zephyr.elf
Memory region      Used Size  Region Size  %age Used
      FLASH:      22868 B      1 MB      2.18%
      RAM:         7616 B      256 KB      2.91%
      IDT_LIST:      0 GB      2 KB      0.00%
Terminal will be reused by tasks, press any key to close it.
Executing task: nRF Connect: Build: hello_world/build (active)

Building hello_world
west build --build-dir c:\Users\mbilloo\eoc2023\hello_world\build c:\Users\mbilloo\eoc2023\hello_world

ninja: no work to do.
Terminal will be reused by tasks, press any key to close it.
```

Hello World!

```
File Edit Selection View Go Run Terminal Help
main.c - hello_world - Visual Studio Code

Welcome to nRF Connect
C main.c A X
src > C main.c > ...
1  /*
2  * Copyright (c) 2012-2014 Wind River Systems, Inc.
3  *
4  * SPDX-License-Identifier: Apache-2.0
5  */
6
7 #include <zephyr/kernel.h>
8
9 void main(void)
10 {
11     printk("Hello World! %s\n", CONFIG_BOARD);
12 }
13
```

HELLO_WORLD build

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL NRF TERMINAL

[166/168] Building C object zephyr/DtsKefFiles/zephyr_final_dir/misc/empty_file.c.obj
[166/168] Building C object zephyr/DtsKefFiles/zephyr_final_dir/dev_handles.c.obj
[167/168] Building C object zephyr/DtsKefFiles/zephyr_final_dir/irq_tables.c.obj
[168/168] Linking C executable zephyr/zephyr.elf

Memory region	Used Size	Region Size	%age Used
FLASH:	22868 B	1 MB	2.18%
RAM:	7616 B	256 KB	2.91%
IDT_LIST:	8 KB	2 KB	0.90%

Terminal will be reused by tasks, press any key to close it.

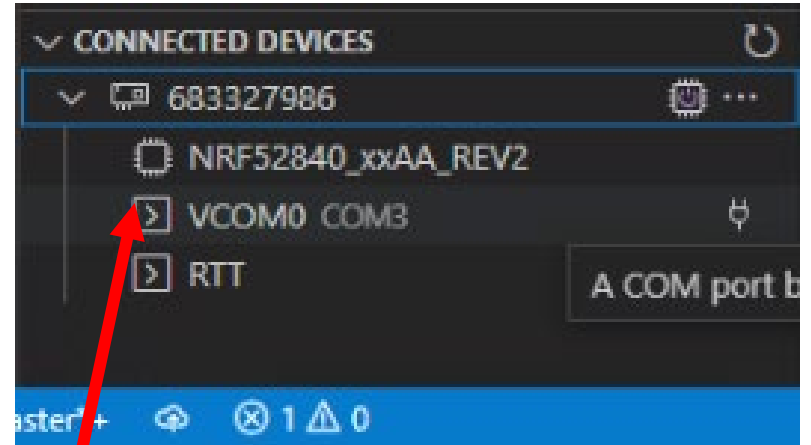
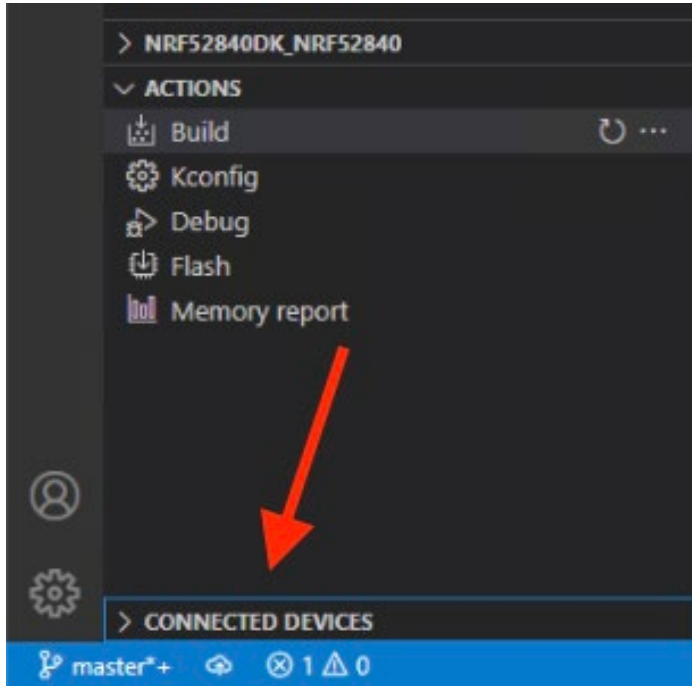
Executing task: nRF Connect: Build: hello_world/build (active)

```
Building hello_world
west build --build-dir c:\Users\mbillioo\ecoc2023\hello_world\build c:\Users\mbillioo\ecoc2023\hello_world
ninja: no work to do.
Terminal will be reused by tasks, press any key to close it.
```

master+ 1/0

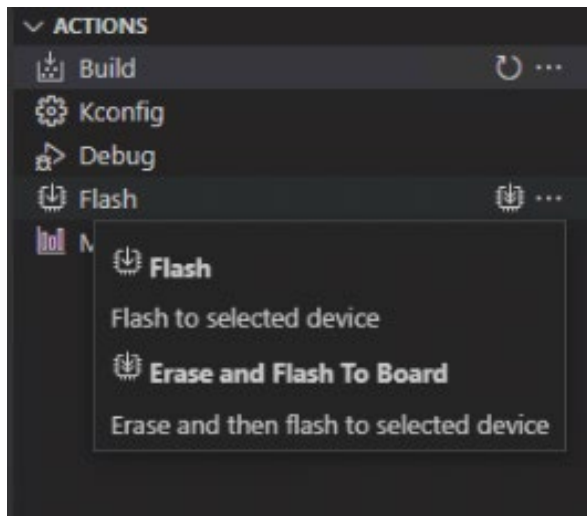
In 13, Col 1 Tab Size 4 UTF-8 CRLF C Win32 hello_world/build

Hello World!



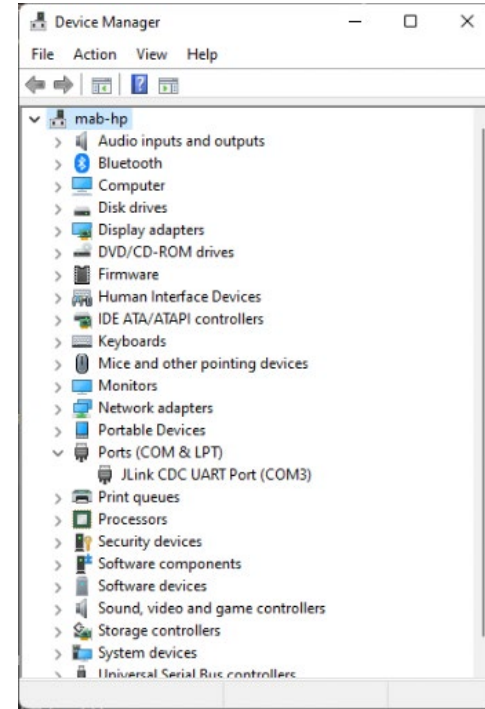
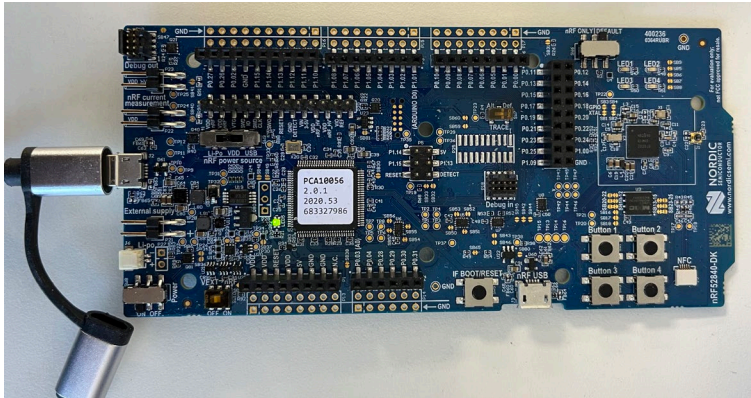
MY BOARD!

Hello World!

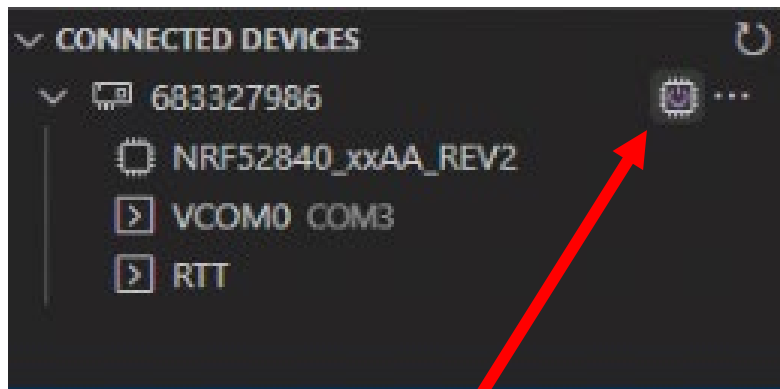


```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL NRF TERMINAL
ninja: no work to do.
Terminal will be reused by tasks, press any key to close it.
Executing task: nRF Connect: Flash: hello_world/build (active)
Flashing build to 683327986
west flash -d c:\Users\mbilloo\ec2023\hello_world\build --skip-rebuild -r nrfjprog --dev-id 683327986
-- west flash: using runner nrfjprog
-- runners.nrfjprog: Flashing file: c:\Users\mbilloo\ec2023\hello_world\build\zephyr\zephyr.hex
[ ##### ] 1.107s | Erase file - Done erasing
[ ##### ] 0.318s | Program file - Done programming
[ ##### ] 0.287s | Verify file - Done verifying
Enabling pin reset.
Applying pin reset.
-- runners.nrfjprog: Board with serial number 683327986 flashed successfully.
Terminal will be reused by tasks, press any key to close it.
```

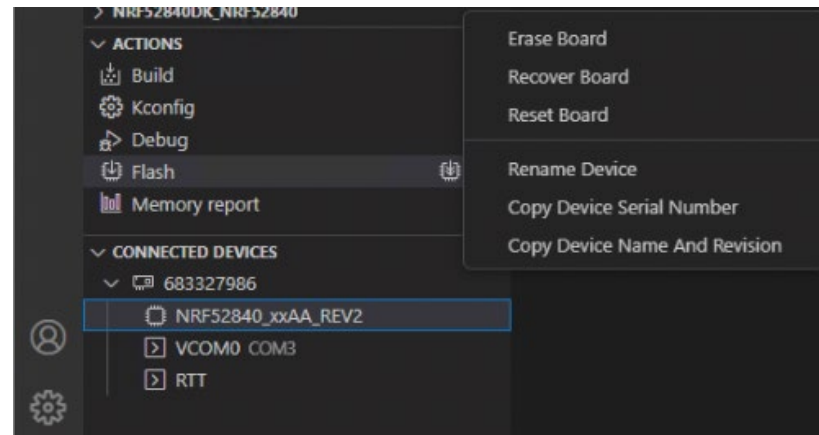
Hello World!



Hello World!



BOARD RESET



COM3 - PuTTY

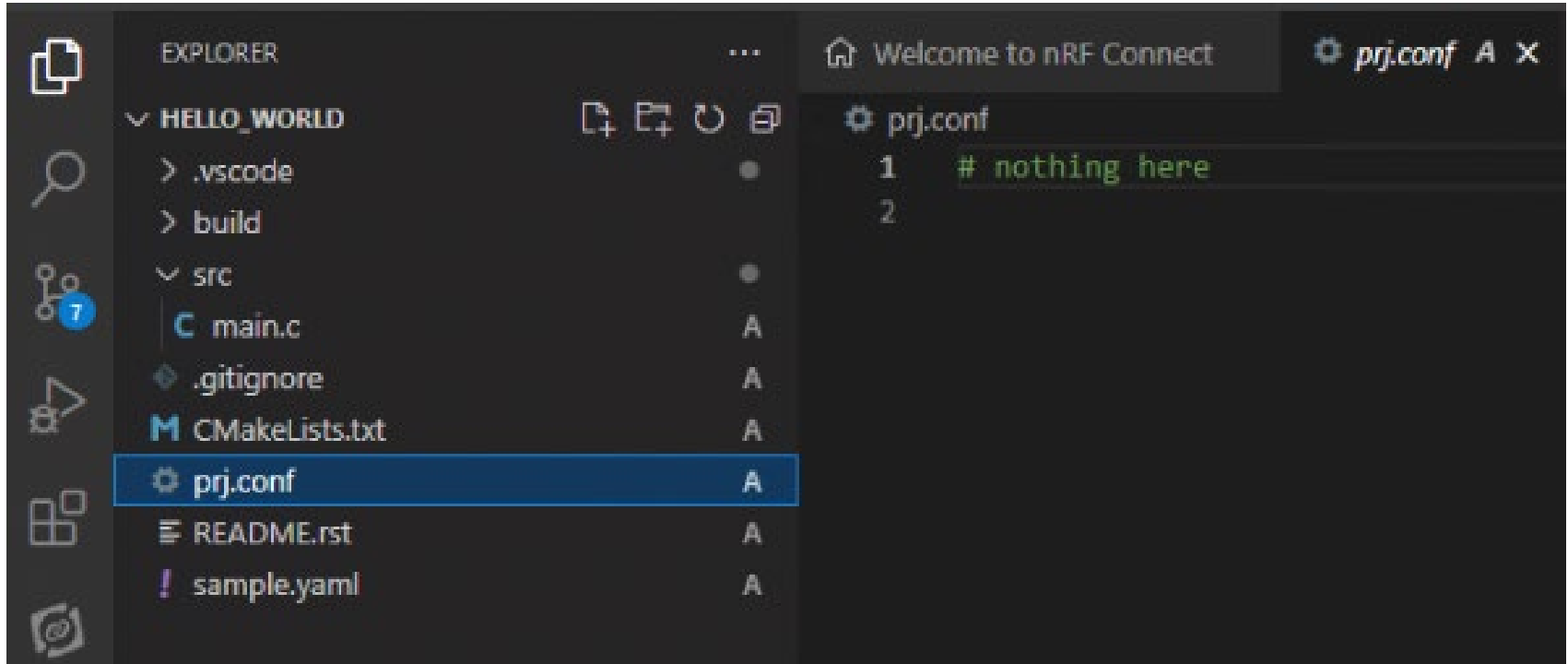
```
*** Booting Zephyr OS build v3.2.99-ncs2 ***  
Hello World! nrf52840dk_nrf52840  
*** Booting Zephyr OS build v3.2.99-ncs2 ***  
Hello World! nrf52840dk_nrf52840  
█
```

KConfig

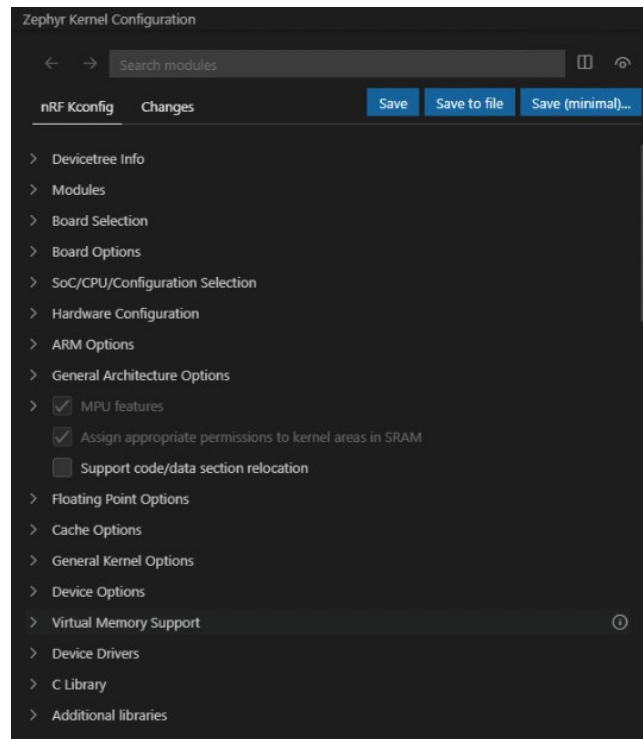
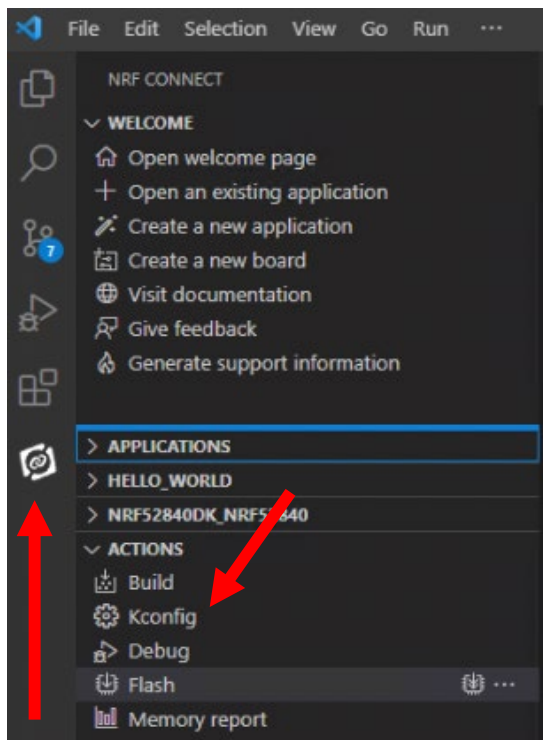
- Borrowed from Linux
- Used to configure Zephyr features
 - Subsystems
- Similar to specifying macros in other RTOSes
 - All features (not just kernel options)!
- Configuration
 - Can be file based
 - “prj.conf”
 - Or use GUI



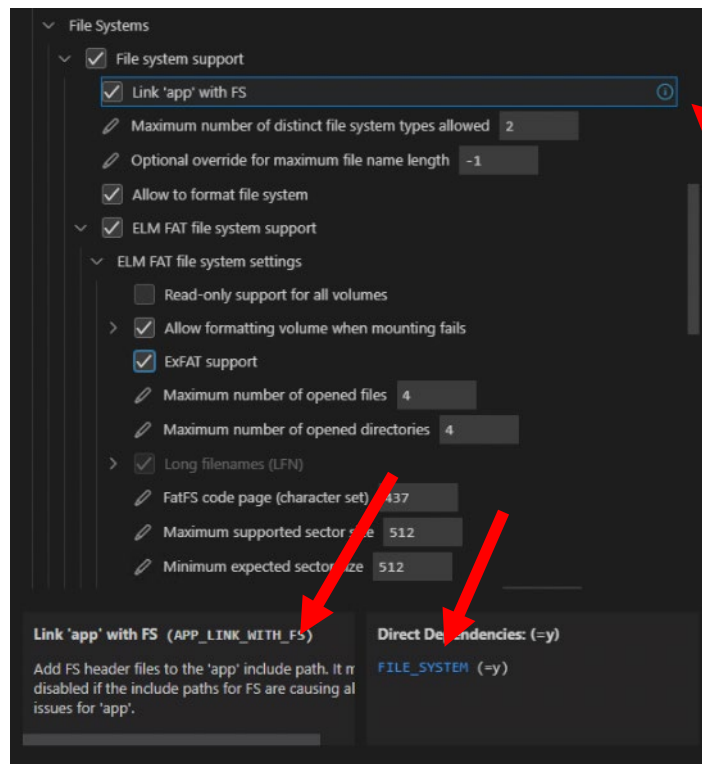
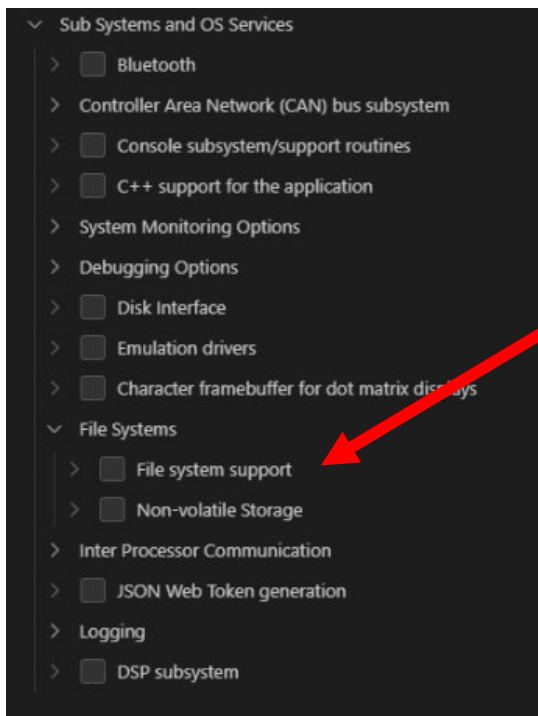
KConfig



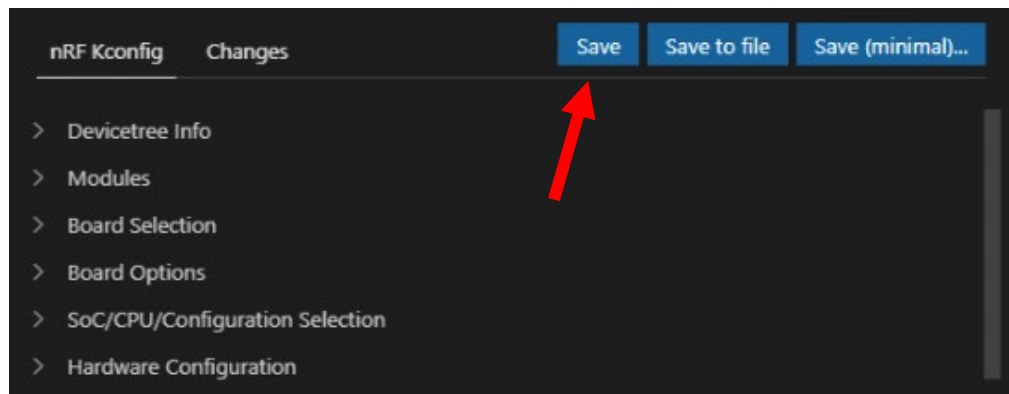
KConfig



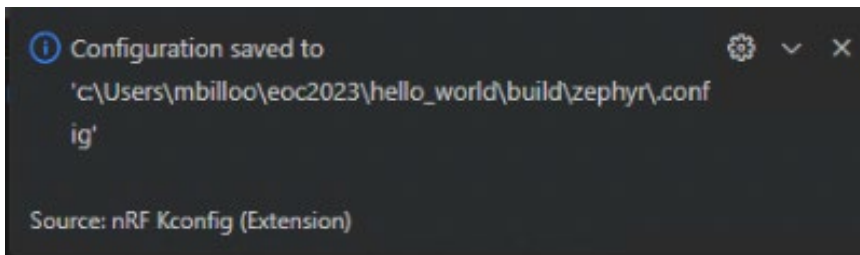
KConfig



KConfig



**CHANGES WILL
NOT BE SAVED
TO prj.conf!**

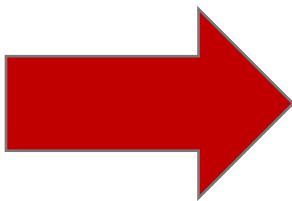


**BAD FOR
VERSION
CONTROL!**

KConfig

```
#  
# File Systems  
#  
CONFIG_FILE_SYSTEM=y  
CONFIG_APP_LINK_WITH_FS=y  
CONFIG_FILE_SYSTEM_MAX_TYPES=2  
CONFIG_FILE_SYSTEM_MAX_FILE_NAME=-1  
CONFIG_FILE_SYSTEM_MKFS=y  
CONFIG_FAT_FILESYSTEM_ELM=y  
  
#  
# ELM FAT file system settings  
#  
# CONFIG_FS_FATFS_READ_ONLY is not set  
CONFIG_FS_FATFS_MKFS=y  
CONFIG_FS_FATFS_MOUNT_MKFS=y  
CONFIG_FS_FATFS_MAX_ROOT_ENTRIES=512  
CONFIG_FS_FATFS_EXFAT=y  
CONFIG_FS_FATFS_NUM_FILES=4  
CONFIG_FS_FATFS_NUM_DIRS=4  
CONFIG_FS_FATFS_LFN=y  
CONFIG_FS_FATFS_LFN_MODE_BSS=y  
# CONFIG_FS_FATFS_LFN_MODE_STACK is not set  
CONFIG_FS_FATFS_MAX_LFN=255  
CONFIG_FS_FATFS_CODEPAGE=437  
CONFIG_FS_FATFS_MAX_SS=512  
CONFIG_FS_FATFS_MIN_SS=512  
CONFIG_FS_FATFS_WINDOW_ALIGNMENT=1  
# end of ELM FAT file system settings  
  
# CONFIG_NVS is not set  
# end of File Systems
```

CAN
COMMIT
CHANGES
TO prj.conf



ADD build
DIRECTORY TO
.gitignore

```
prj.conf \, M X  
prj.conf > ...  
1 #  
2 # File Systems  
3 #  
4 CONFIG_FILE_SYSTEM=y  
5 CONFIG_APP_LINK_WITH_FS=y  
6 CONFIG_FILE_SYSTEM_MAX_TYPES=2  
7 CONFIG_FILE_SYSTEM_MAX_FILE_NAME=-1  
8 CONFIG_FILE_SYSTEM_MKFS=y  
9 CONFIG_FAT_FILESYSTEM_ELM=y  
10  
11 #  
12 # ELM FAT file system settings  
13 #  
14 # CONFIG_FS_FATFS_READ_ONLY is not set  
15 CONFIG_FS_FATFS_MKFS=y  
16 CONFIG_FS_FATFS_MOUNT_MKFS=y  
17 CONFIG_FS_FATFS_MAX_ROOT_ENTRIES=512  
18 CONFIG_FS_FATFS_EXFAT=y  
19 CONFIG_FS_FATFS_NUM_FILES=4  
20 CONFIG_FS_FATFS_NUM_DIRS=4  
21 CONFIG_FS_FATFS_LFN=y  
22 CONFIG_FS_FATFS_LFN_MODE_BSS=y  
23 # CONFIG_FS_FATFS_LFN_MODE_STACK is not set  
24 CONFIG_FS_FATFS_MAX_LFN=255  
25 CONFIG_FS_FATFS_CODEPAGE=437  
26 CONFIG_FS_FATFS_MAX_SS=512  
27 CONFIG_FS_FATFS_MIN_SS=512  
28 CONFIG_FS_FATFS_WINDOW_ALIGNMENT=1  
29 # end of ELM FAT file system settings  
30  
31 # CONFIG_NVS is not set  
32 # end of File Systems
```

Devicetree

- Borrowed from Linux
- Used to configure hardware
 - Pinmux
 - Pinout
 - Board peripherals
- Part of the build process
 - Start with “top-level” device tree file (provided by vendor)
 - Contains references to other device tree files
 - Pinout configuration
 - Can customize devicetree using “overlays”
 - Custom hardware



Zephyr™



Devicetree Overlay

- For custom hardware
- Change peripheral pinout
- Enable subsystem
 - Sometimes enablement is done via devicetree (instead of KConfig)

Devicetree Overlay

- Example: Enabling ADC and SPI-SD card support

```
/ {
    aliases {
        adcctrl = &adc;
    };
};

&adc {
    status = "okay";
};
```

```
sdhc0: sdhc@0 {
    compatible = "zephyr,sdhc-spi-slot";
    reg = <0>;
    status = "okay";
    label = "SDHC_0";
    mmc {
        compatible = "zephyr,sdmmc-disk";
        status = "okay";
    };
    spi-max-frequency = <24000000>;
};
```

Devicetree Overlay

- Example: Changing default pins

```
&spi1 {
    status = "okay";
    pinctrl-0 = <&mab_spi>;
    pinctrl-1 = <&mab_spi>;
    pinctrl-names = "default", "sleep";
    cs-gpios = <&gpio0 2 GPIO_ACTIVE_LOW>;
};
```

```
&arduino_i2c {
    pinctrl-0 = <&mab_i2c>;
    pinctrl-1 = <&mab_i2c>;
    pinctrl-names = "default", "sleep";
};
```

```
&pinctrl {
    mab_spi: mab_spi {
        group1 {
            psels = <NRF_PSEL(SPIM_SCK, 0, 26)>,
                <NRF_PSEL(SPIM_MOSI, 0, 27)>,
                <NRF_PSEL(SPIM_MISO, 1, 8)>;
        };
    };

    mab_i2c: mab_i2c {
        group1 {
            psels = <NRF_PSEL(TWIM_SDA, 1, 15)>,
                <NRF_PSEL(TWIM_SCL, 1, 14)>;
        };
    };
};
```

What's Next?

- Typical RTOS data structures and paradigms
- Threads/tasks
- Thread/task synchronization
 - Mutex
 - Semaphore
- Thread/task communication
 - Workqueue
 - FIFO

Intermediate Topics

- Connectivity
- BLE
 - Custom advertised name
 - Standard services/characteristics
 - Battery
 - Custom services/characteristics
- WiFi
 - AP mode
 - Station mode

Advanced Topics

- West Manifest
- nRF Connect for Desktop uses Nordic's own West manifest under the hood
- Maybe beneficial to bring West manifest in-house
 - Ensure that all source code is from a particular version
 - Unknown updates may break functionality
 - Helpful for client projects
- Blog post
 - Getting Started With Zephyr: West Manifest Customization
 - <https://www.embeddedrelated.com/showarticle/1505.php>

Thank You!

June 20–22, 2023 | San Clara, CA



#SensorsConverge