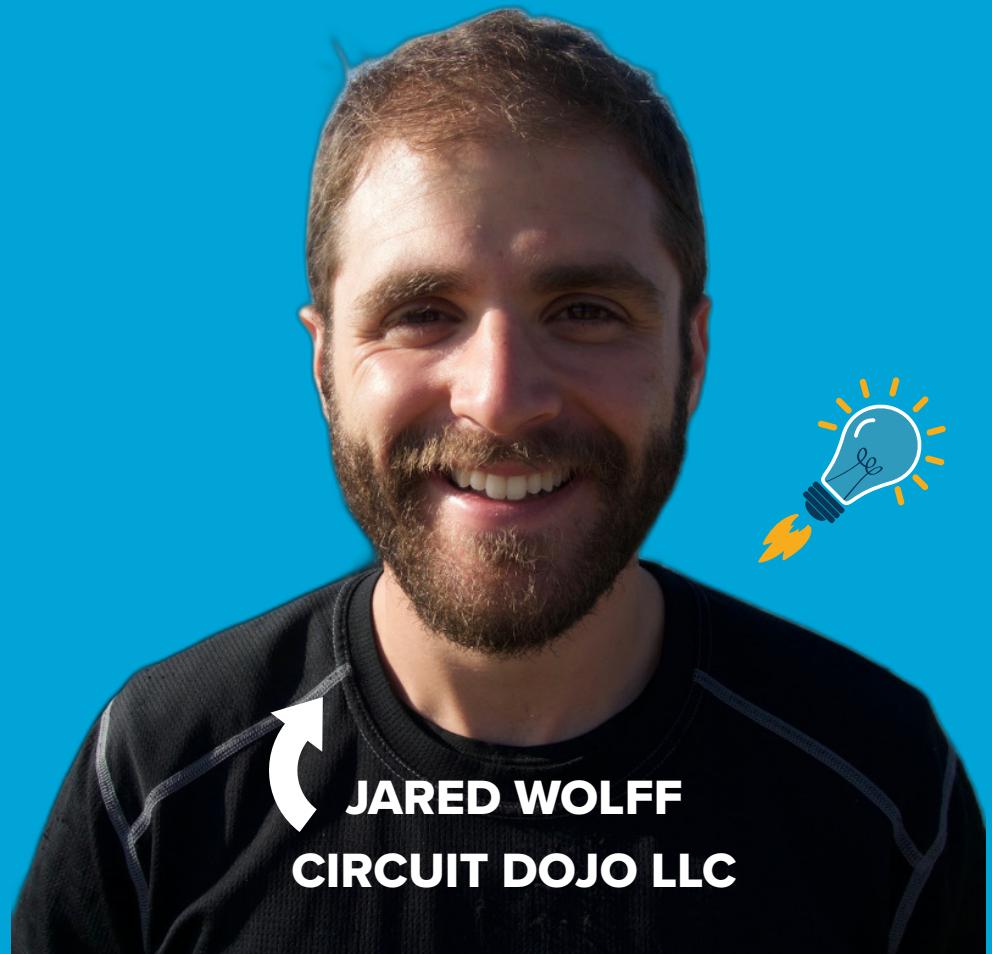




ZDS 2022

ONE CODE BASE
MULTIPLE TARGETS



WEEKLY LIVE SESSIONS

YOUTUBE.COM/C/CIRCUITDOJO



DESIGNING FOR MULTIPLE TARGETS

- Core tenants of Zephyr
- It's as simple as changing a board type
- Does require a little more work re: configuration



PREP WORK

- Do your research!
- Make sure that:
 - All devices used support the peripherals used
 - There is support in Zephyr for these peripherals



PREP WORK

- For this presentation we'll be using 3 different boards/chips
- nRF9160 Feather (Nordic nRF9160)
- Particle Xenon (Nordic nRF52840)
- LORA E5 Dev Board (ST STM32WLE5JC)

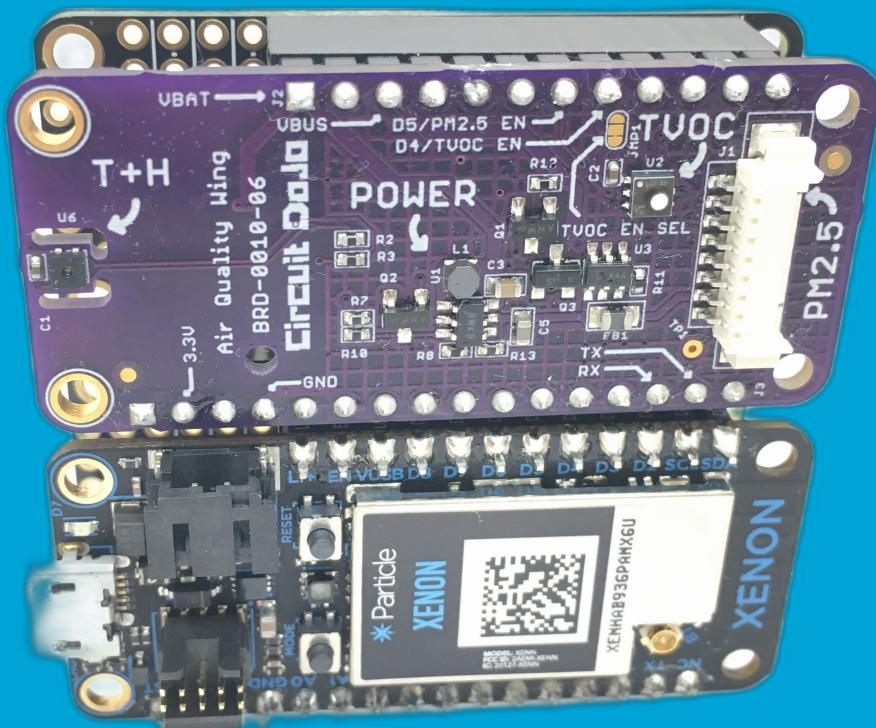


PREP WORK

- For the feathers, we'll plug into a doubler while using the Air Quality Wing (AQW)
- The AQW uses:
 - 1 UART interface
 - 1 I2C interface
 - A couple control GPIOs



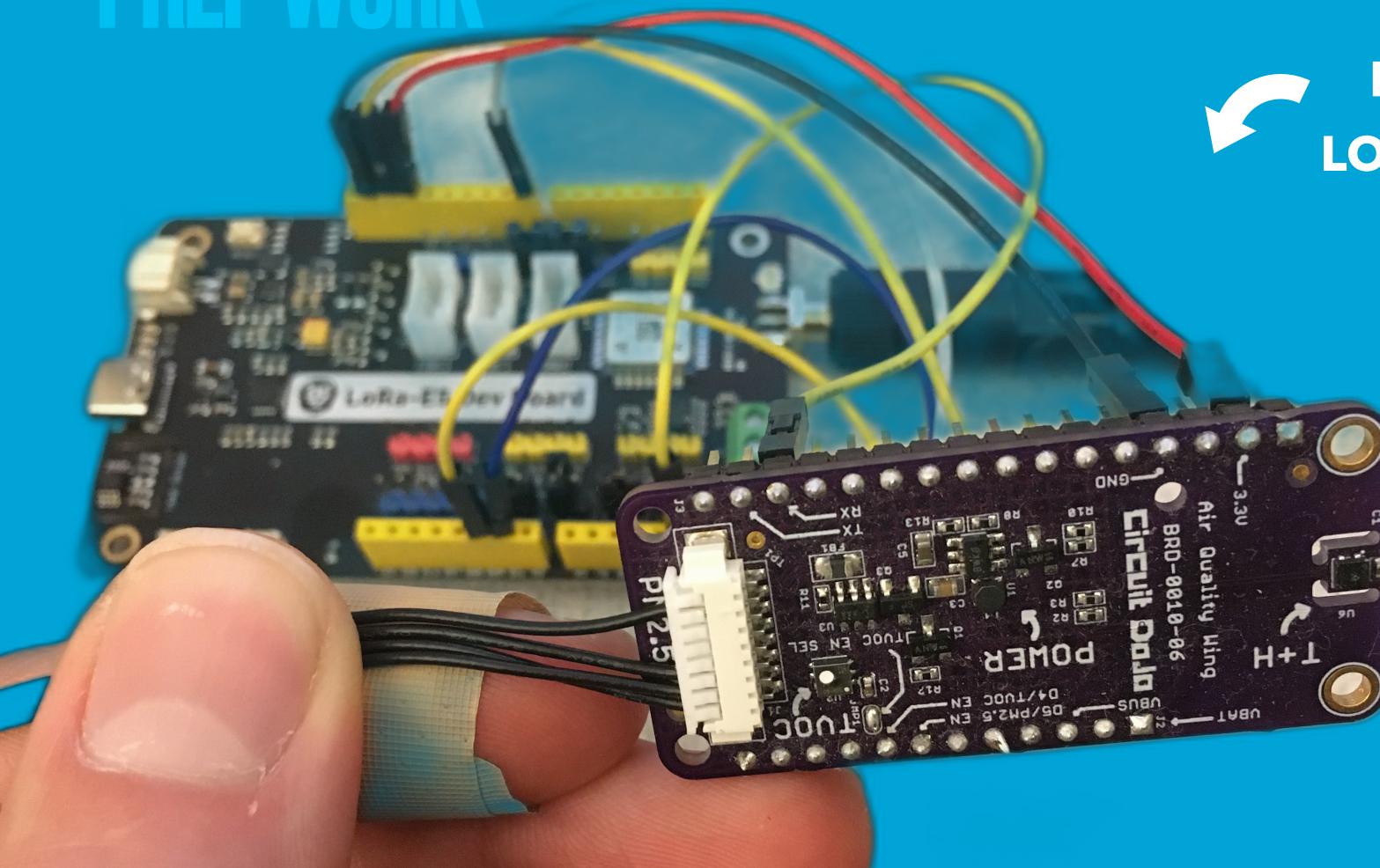
PREP WORK



WHAT WE'RE
WORKING WITH



PREP WORK



HERE'S THE
LORA E5 SETUP



PREP WORK

- We'll be measuring
- PM2.5
- Temp
- Humidity
- VOC



PREP WORK

```
[00:00:00.004,000] <inf> aqw_basic_demo: Air Quality Wing Demo  
[00:00:04.040,000] <inf> aqw_basic_demo: Temperature: 22.0°C  
[00:00:04.040,000] <inf> aqw_basic_demo: Humidity: 45.0%  
[00:00:09.050,000] <inf> aqw_basic_demo: PM2.5: 8.0 ppm
```



PREP WORK - MANIFEST

- Manifest files
- Differentiating between different flavors of Zephyr
 - Vanilla (west-vanilla.yml)
 - nRF Connect SDK (west-ncs.yml)
- Prevents build conflicts and loads only what you need



PREP WORK - NCS WEST MANIFEST



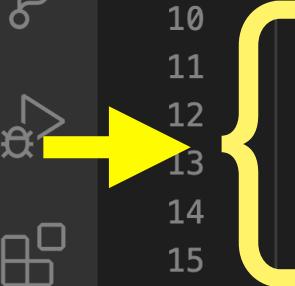
west-ncs.yml — air-quality-wing-zephyr-example

```
! west-ncs.yml ×
```

demo > ! west-ncs.yml

```
7 manifest:
8   remotes:
9     - name: nrfconnect
10    url-base: https://github.com/nrfconnect
11  projects:
12    - name: nrf
13      repo-path: sdk-nrf
14      remote: nrfconnect
15      revision: v1.7.1
16      import: true
17      # Drivers repository
18      - name: air-quality-wing-drivers
19
```

PREP WORK - “VANILLA” MANIFEST



west-vanilla.yml — air-quality-wing-zephyr-example

```
! west-vanilla.yml ×
```

demo > ! west-vanilla.yml

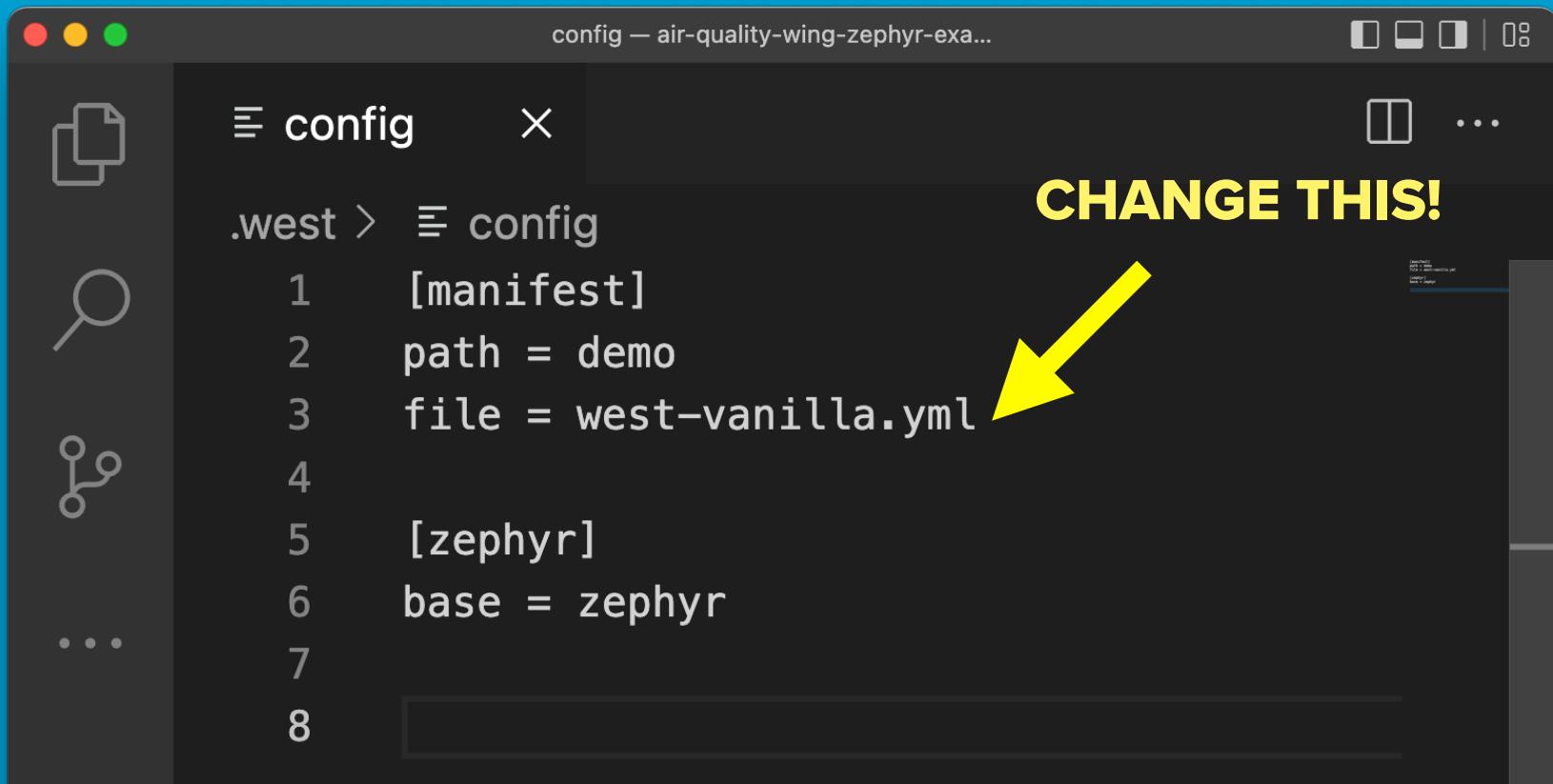
```
7 manifest:
8   remotes:
9     - name: zephyrproject
10    url-base: https://github.com/zephyrproject-rtos
11  projects:
12    - name: zephyr
13      repo-path: zephyr
14      remote: zephyrproject
15      revision: v3.0.0
16      import:
17        name-allowlist:
18          - cmsis
19          - hal_st
20          - hal_stm32
```

PREP WORK- MANIFEST

- Switching between each manifest is fairly easy
- Edit your .west/config
 - Change manifest.file to west-vanilla.yml or west-ncs.yml depending on your needs
- Re-run west init and west update to pull requirements



PREP WORK - MANIFEST



A screenshot of a Mac OS X desktop environment showing a terminal window titled "config — air-quality-wing-zephyr-exa...". The terminal displays a configuration file with the following content:

```
.west > config
1 [manifest]
2 path = demo
3 file = west-vanilla.yml
4
5 [zephyr]
6 base = zephyr
7
8
```

A large yellow arrow points from the word "file" in line 3 to the ".yml" extension of the file name "west-vanilla.yml". To the right of the arrow, the text "CHANGE THIS!" is displayed in yellow capital letters.

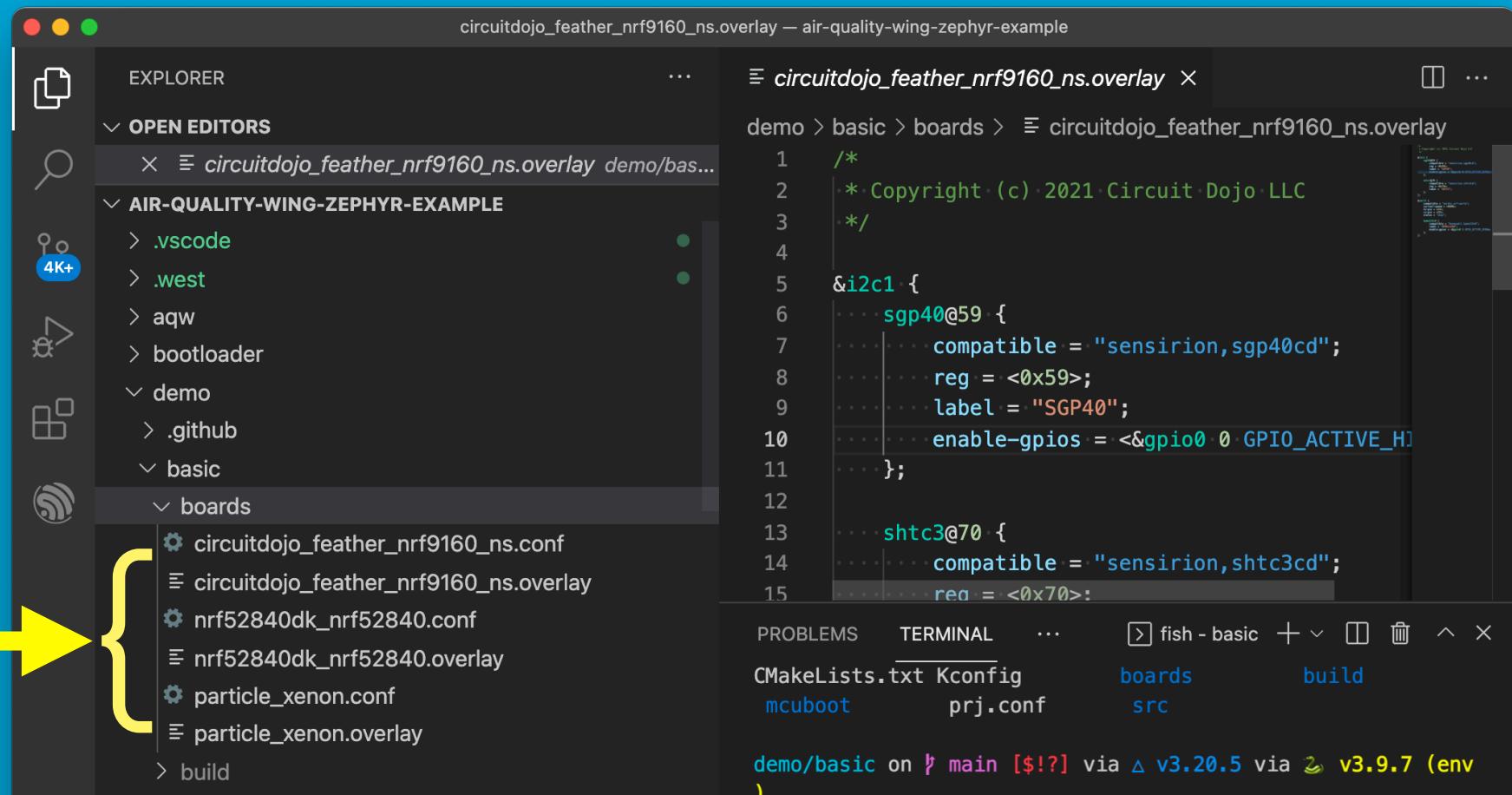


DESIGN WORK

- Next, let's get to the design work:
 - The base code stays the same but your device specific overlays and configurations change
 - These files get merged into your application when *west build* is invoked



DESIGN WORK



A screenshot of the VS Code interface. The title bar says "circuitdojo_feather_nrf9160_ns.overlay — air-quality-wing-zephyr-example". The left sidebar shows a file tree under "OPEN EDITORS" and "AIR-QUALITY-WING-ZEPHYR-EXAMPLE". A yellow arrow points from the bottom-left towards the file tree. The main area shows a code editor with the following content:

```
demo > basic > boards > circuitdojo_feather_nrf9160_ns.overlay
1  /*
2   * Copyright (c) 2021 Circuit Dojo LLC
3   */
4
5 &i2c1 {
6     sgp40@59 {
7         compatible = "sensirion,sgp40cd";
8         reg = <0x59>;
9         label = "SGP40";
10        enable-gpios = <&gpio0 0 GPIO_ACTIVE_HIGH>;
11    };
12
13    shtc3@70 {
14        compatible = "sensirion,shtc3cd";
15        reg = <0x70>;
```

The bottom right corner of the code editor has a small lightbulb icon with a lightning bolt.

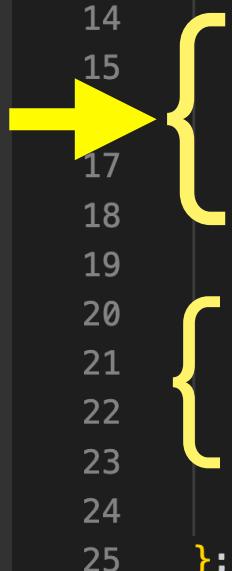


DESIGN WORK - COMMONALITIES

- Commonalities include
- Sub-device labels, addresses, configuration
- Same *compatible* entries for each sub-device



DESIGN WORK - COMMONALITIES



lora_e5_dev_board.overlay — air-quality-wing-zephyr-example

demo > basic > boards > lora_e5_dev_board.overlay > ...

```
12
13     &i2c2 {
14         sgp40@59 {
15             compatible = "sensirion,sgp40cd";
16             reg = <0x59>;
17             label = "SGP40";
18         };
19
20         sh10@70 {
21             compatible = "sensirion,sh10cd";
22             reg = <0x70>;
23             label = "SH10";
24         };
25     };
```

DESIGN WORK - DIFFERENCES

- Differences include:
 - Different phandle for I2C and UART
 - Enabling of other interfaces like DMA
 - Using device specific bindings in the *compatible* entry
 - Pin mappings are also different



DESIGN WORK - DIFFERENCES

particle_xenon.conf — air-quality-wing-zephyr-ex...



```
particle_xenon.conf X
demo > basic > boards > particle_xenon.conf
1 # Increase stack size
2 CONFIG_MAIN_STACK_SIZE=4092
3
4 # UART for HPMA
5 CONFIG_UART_ASYNC_API=y
6
7 # USB
8 CONFIG_USB=y
9 CONFIG_USB_DEVICE_STACK=y
10 CONFIG_USB_DEVICE_PRODUCT="Air Quality Wing"
11 CONFIG_USB_UART_CONSOLE=y
12
13 CONFIG_UART_0_INTERRUPT_DRIVEN=n
14 CONFIG_UART_0_ASYNC=y
```

**ENABLING
BOARD/PROCESSOR
SPECIFIC SETTINGS**



DESIGN WORK - DIFFERENCES

- The nRF52
 - Enabling the Async API
 - Configuring/enabling USB
 - Also some important settings to allow UART0 and USB to coexist



DESIGN WORK - DIFFERENCES

lora_e5_dev_board.conf -- air-quality-wing-zephyr-example

lora_e5_dev_board.conf X

demo > basic > boards > lora_e5_dev_board.conf

```
1 CONFIG_LOG=y
2
3 # DMA
4 CONFIG_DMA=y
5 CONFIG_DMA_STM32=y
6
7 # ASYNC Uart
8 CONFIG_UART_STM32=y
9 CONFIG_UART_ASYNC_API=y
10
11 # Enable bootloader
12 CONFIG_BOOTLOADER_MCUBOOT=y
13 CONFIG_MCUBOOT_SIGNATURE_KEY_FILE="bootloader/mcuboot/root-rsa-2048.pem"
```

**ENABLING
BOARD/PROCESSOR
SPECIFIC SETTINGS**



DESIGN WORK - DIFFERENCES

- The STM32
 - Requires DMA for ASYNC UART to work
 - DMA requires manual configuration within your overlay
 - Different phandles since it's a different processor/ HAL
 - Still works out of the box with the AQW drivers!



DESIGN WORK - DIFFERENCES

BOARD/
ARCHITECTURE
SPECIFIC
PROPERTIES

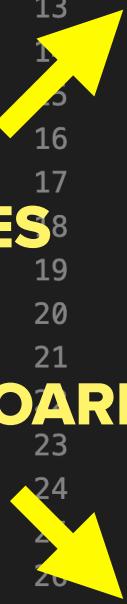


lora_e5_dev_board.overlay — air-quality-wing-zephyr-example

```
lora_e5_dev_board.overlay 9+ X
demo > basic > boards > lora_e5_dev_board.overlay > serial@40008000 > dma-names
23     label = "SHTC3";
24 };
25 };
26
27 &lpuart1 {
28     current-speed = <9600>;
29     dmas = <&dmamux1 11 22 0x20440
30         &dmamux1 1 21 0x20480>;
31     dma-names = "tx", "rx";
32
33
34     hpma115s0 {
35         compatible = "honeywell,hpma115s0";
36         label = "HPMA115S0";
37         enable-gpios = <&gpiob 5 GPIO_ACTIVE_HIGH>;
38     };
39 }
```

DESIGN WORK - DIFFERENCES

PHANDLES
DIFFER
BOARD BY BOARD



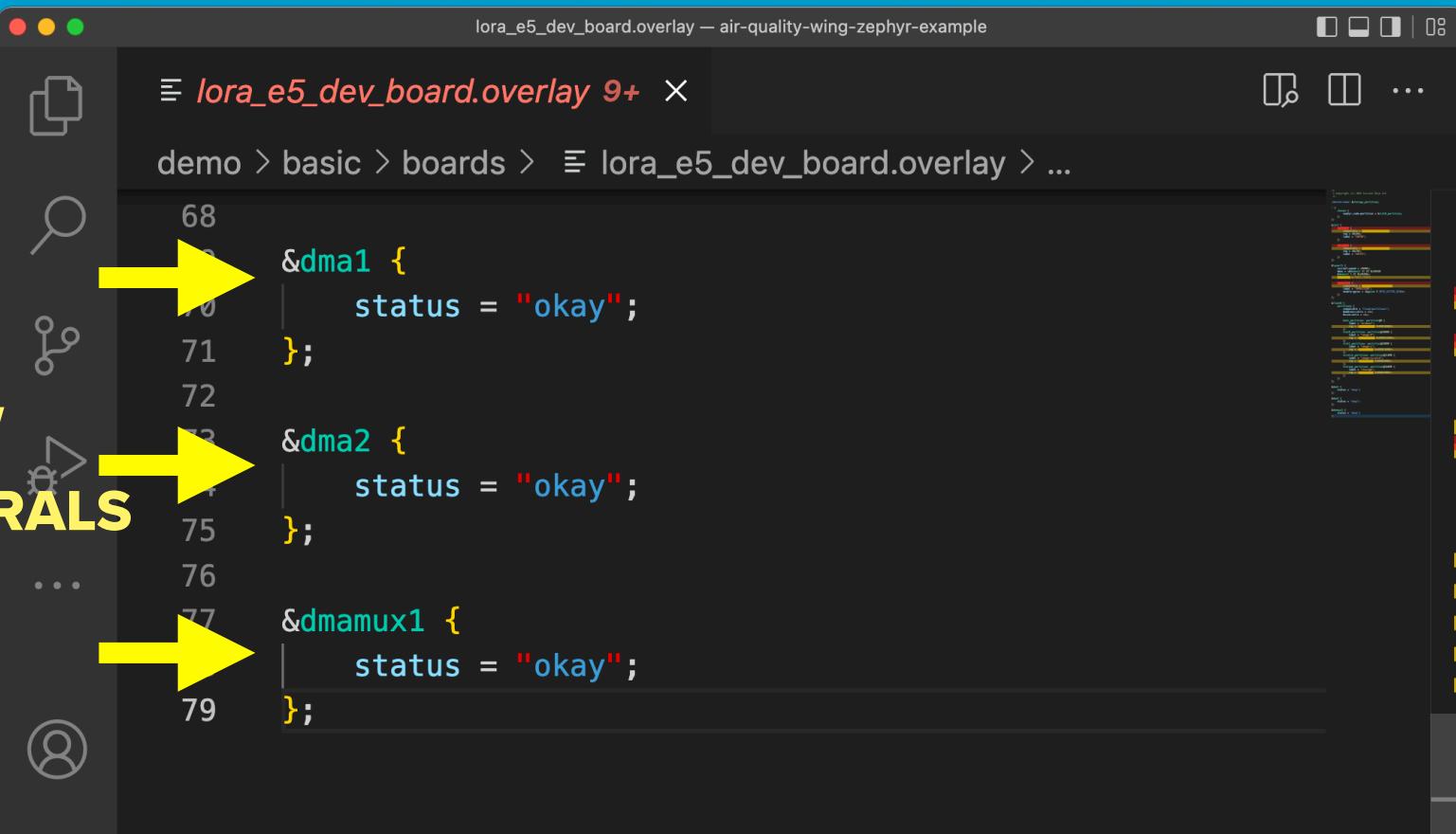
```
lora_e5_dev_board.overlay — air-quality-wing-zephyr-example
lora_e5_dev_board.overlay 9+ X
demo > basic > boards > lora_e5_dev_board.overlay > serial@40008000 > hpma115s0

12
13     &i2c2 {
14         sgp40@59 {
15             compatible = "sensirion,sgp40cd";
16             reg = <0x59>;
17             label = "SGP40";
18         };
19
20         shtc3@70 {
21             compatible = "sensirion,shtc3cd";
22             reg = <0x70>;
23             label = "SHTC3";
24         };
25
26     &lpuart1 {
27         current-speed = <9600>;
```



DESIGN WORK - DIFFERENCES

NEW
PERIPHERALS



```
lora_e5_dev_board.overlay — air-quality-wing-zephyr-example
demo > basic > boards > lora_e5_dev_board.overlay > ...
68
69     &dma1 {
70         status = "okay";
71     };
72
73     &dma2 {
74         status = "okay";
75     };
76
77     &dmamux1 {
78         status = "okay";
79     };
...
```



BUILDING

- Choose your target!
- Make sure it's supported by both Zephyr and the sample you're trying to build
- Or, create your own board definition!
- Make sure that between building for different platforms you do a *pristine* build
- `west build -b circuitdojo_feather_nrf9160_ns -p`



TRY IT

LETS PLAY WITH IT



CODE AVAILABLE

[GITHUB.COM/CIRCUITDOJO/AIR-QUALITY-WING-ZEPHYR-DEMO](https://github.com/circuitdojo/air-quality-wing-zephyr-demo)



QUESTIONS?



COMMUNITY.JAREDWOLFF.COM

HELLO@JAREDWOLFF.COM