Working with Zephyr community.

A timeline for adding support for a new board

Daniel Baluta

April 2024

Content

Goals of the presentation

Zephyr release cycle

Toolchain support

Board support

Working with the community

Timeline for contributing upstream

Goals of the presentation

- Newcomer guidance
  - Offer guidance on the initial steps for newcomers to begin with Zephyr development
- Board integration
  - Provide insight into the process of adding a new board in Zephyr
  - Use the "Hello World" sample to demonstrate functionality
- Community collaboration
  - Illustratre how Zephyr community works looking at the timeline for a change to get accepted
- Code acceptance
  - Show the best practices to speed up the process of getting your code accepted

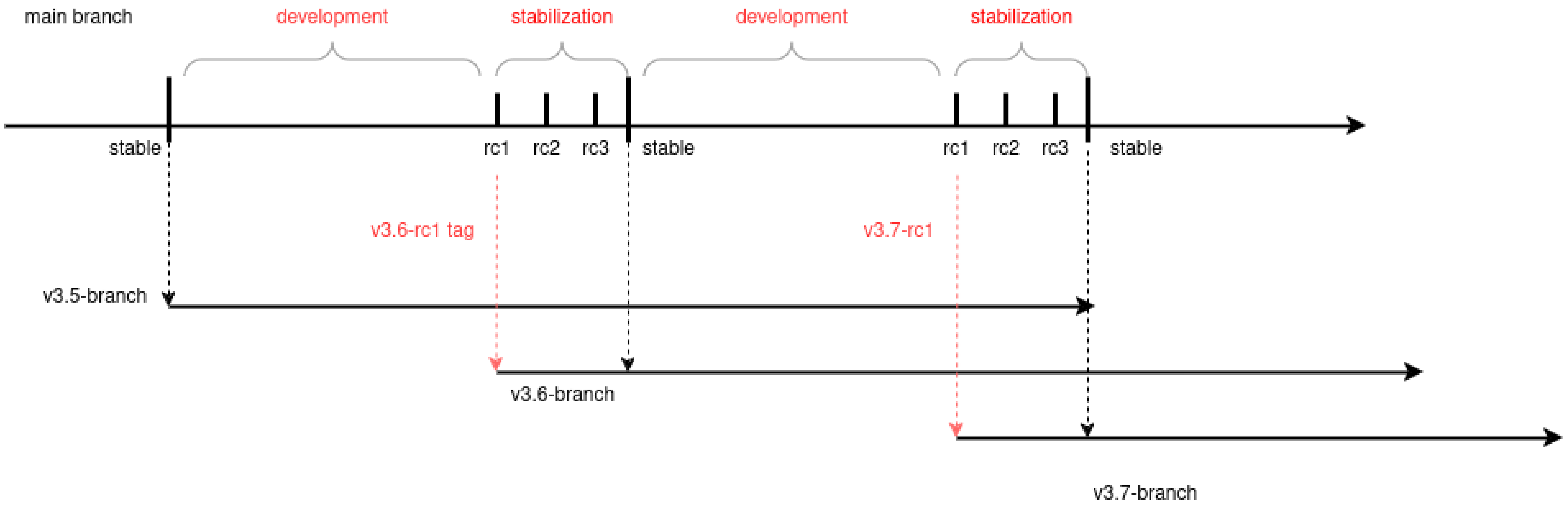
Zephyr release cycle

Zephyr Release Cycle (1)

- Development happens on Github
- Release phases
  - *Development* phase, takes 3 months
  - *Stabilization* phase, takes 1 month (RC1, RC2, RC3)
  - At the end, stable release is out
- Maintenance
  - Periodic releases get updates, bugfixes and security fixes at least 2 cycles
- Long term support
  - Release created every 2 years
  - Is maintained at least 2.5 years after release

Zephyr Release Cycle (2)

- Periodic releases every 4 months
- LTS releases every 2 years

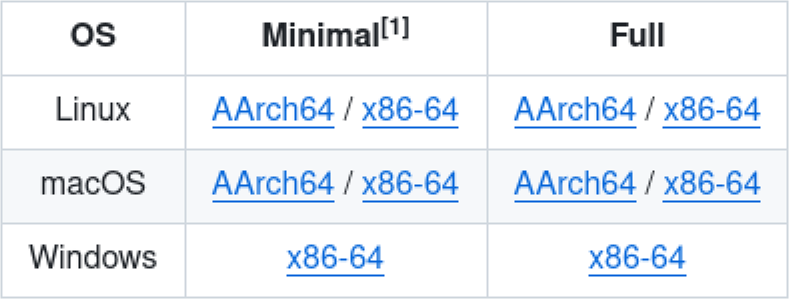


Notes and sources section

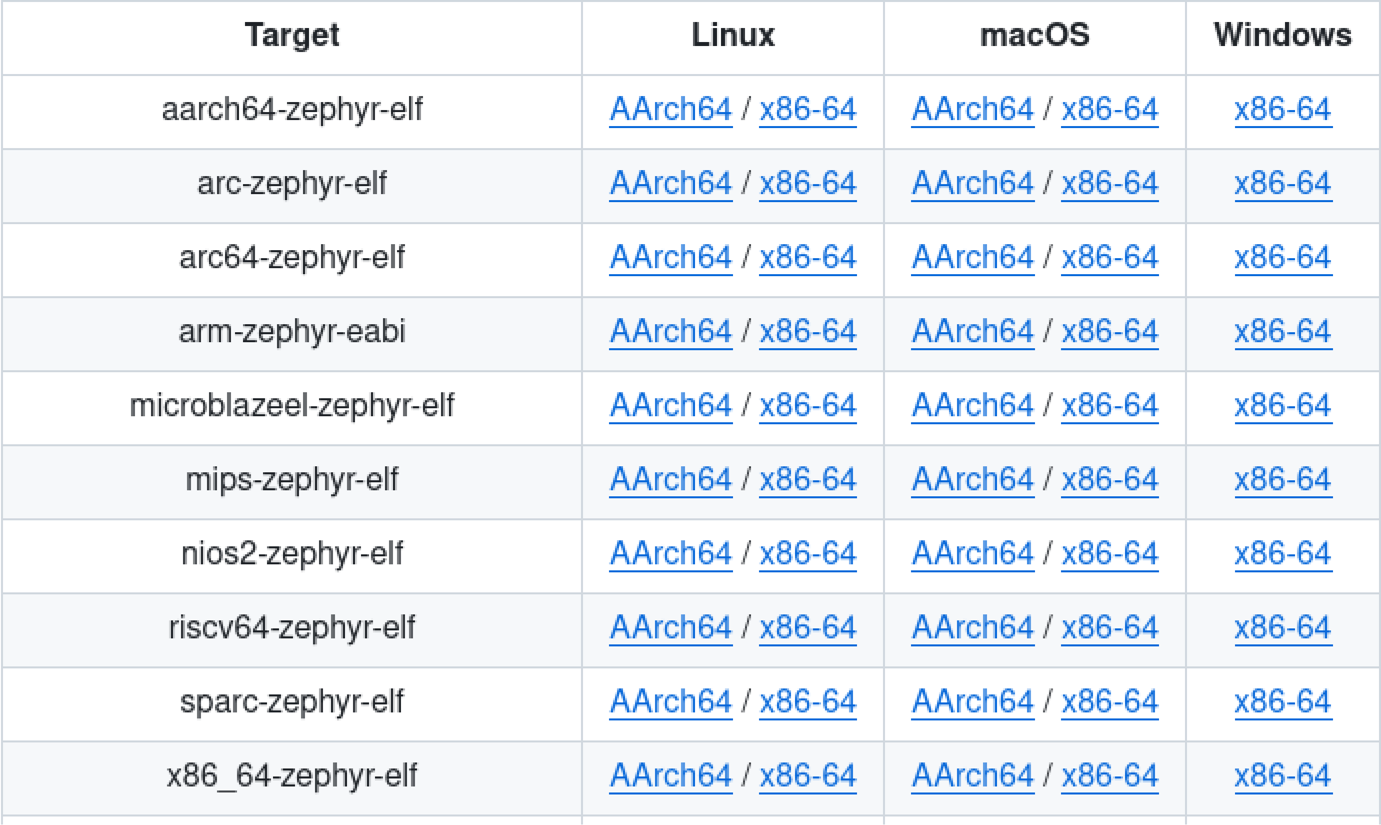
Toolchain support

Toolchain support (1)

- https://github.com/zephyrproject-rtos/sdk-ng/releases

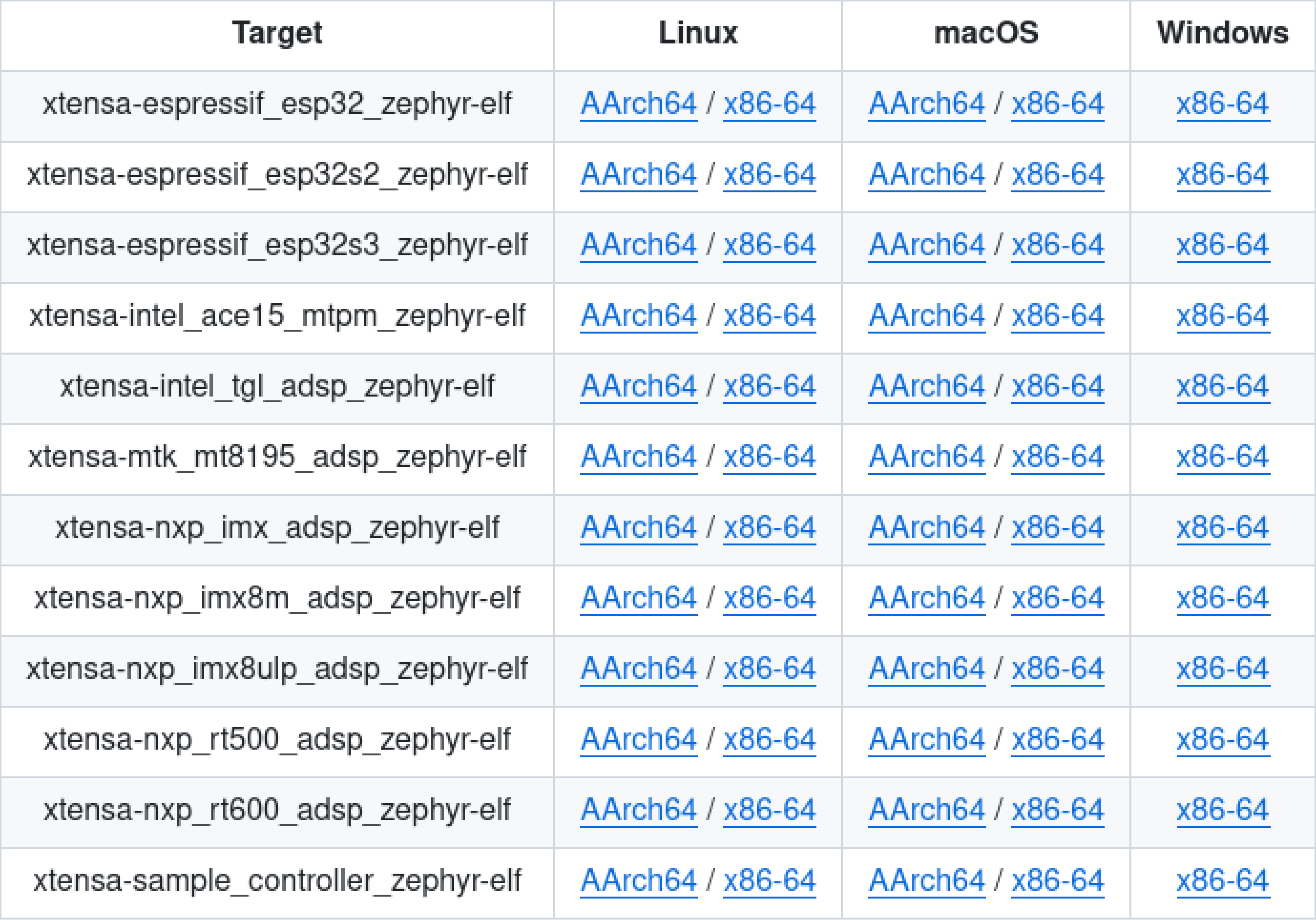


OS	Minimal[1]	Full
Linux	AArch64 / x86-64	AArch64 / x86-64
macOS	AArch64 / x86-64	AArch64 / x86-64
Windows	x86-64	x86-64



Target	Linux	macOS	Windows
aarch64-zephyr-elf	AArch64 / x86-64	AArch64 / x86-64	x86-64
arc-zephyr-elf	AArch64 / x86-64	AArch64 / x86-64	x86-64
arc64-zephyr-elf	AArch64 / x86-64	AArch64 / x86-64	x86-64
arm-zephyr-eabi	AArch64 / x86-64	AArch64 / x86-64	x86-64
microblazeel-zephyr-elf	AArch64 / x86-64	AArch64 / x86-64	x86-64
mips-zephyr-elf	AArch64 / x86-64	AArch64 / x86-64	x86-64
nios2-zephyr-elf	AArch64 / x86-64	AArch64 / x86-64	x86-64
riscv64-zephyr-elf	AArch64 / x86-64	AArch64 / x86-64	x86-64
sparc-zephyr-elf	AArch64 / x86-64	AArch64 / x86-64	x86-64
x86_64-zephyr-elf	AArch64 / x86-64	AArch64 / x86-64	x86-64

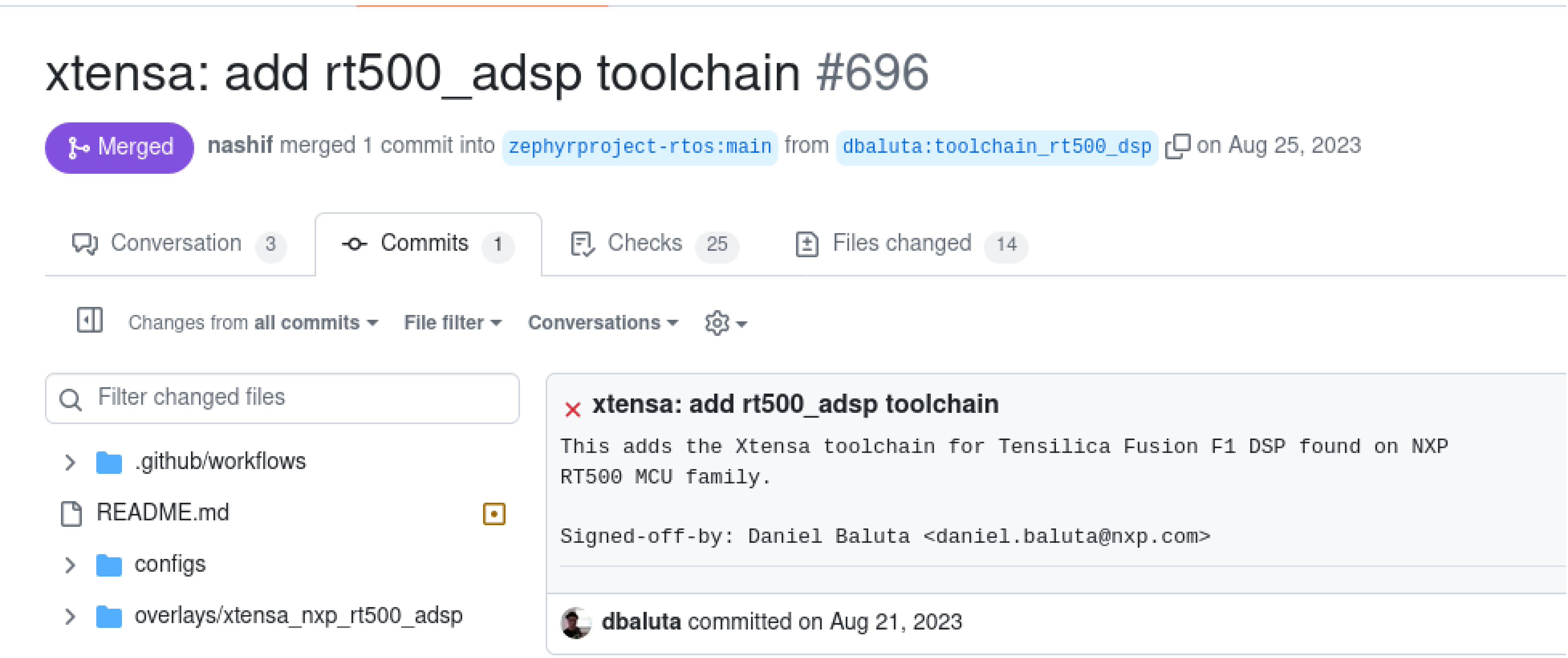
Toolchain support (2)



Target	Linux	macOS	Windows
xtensa-espressif_esp32_zephyr-elf	AArch64 / x86-64	AArch64 / x86-64	x86-64
xtensa-espressif_esp32s2_zephyr-elf	AArch64 / x86-64	AArch64 / x86-64	x86-64
xtensa-espressif_esp32s3_zephyr-elf	AArch64 / x86-64	AArch64 / x86-64	x86-64
xtensa-intel_ace15_mtpm_zephyr-elf	AArch64 / x86-64	AArch64 / x86-64	x86-64
xtensa-intel_tgl_adsp_zephyr-elf	AArch64 / x86-64	AArch64 / x86-64	x86-64
xtensa-mtk_mt8195_adsp_zephyr-elf	AArch64 / x86-64	AArch64 / x86-64	x86-64
xtensa-nxp_imx_adsp_zephyr-elf	AArch64 / x86-64	AArch64 / x86-64	x86-64
xtensa-nxp_imx8m_adsp_zephyr-elf	AArch64 / x86-64	AArch64 / x86-64	x86-64
xtensa-nxp_imx8ulp_adsp_zephyr-elf	AArch64 / x86-64	AArch64 / x86-64	x86-64
xtensa-nxp_rt500_adsp_zephyr-elf	AArch64 / x86-64	AArch64 / x86-64	x86-64
xtensa-nxp_rt600_adsp_zephyr-elf	AArch64 / x86-64	AArch64 / x86-64	x86-64
xtensa-sample_controller_zephyr-elf	AArch64 / x86-64	AArch64 / x86-64	x86-64

Add support for a new toolchain (1)

- Pull Request on https://github.com/zephyrproject-rtos/sdk-ng



xtensa: add rt500_adsp toolchain #696

Merged nashif merged 1 commit into zephyrproject-rtos:main from dbaluta:toolchain_rt500_dsp on Aug 25, 2023

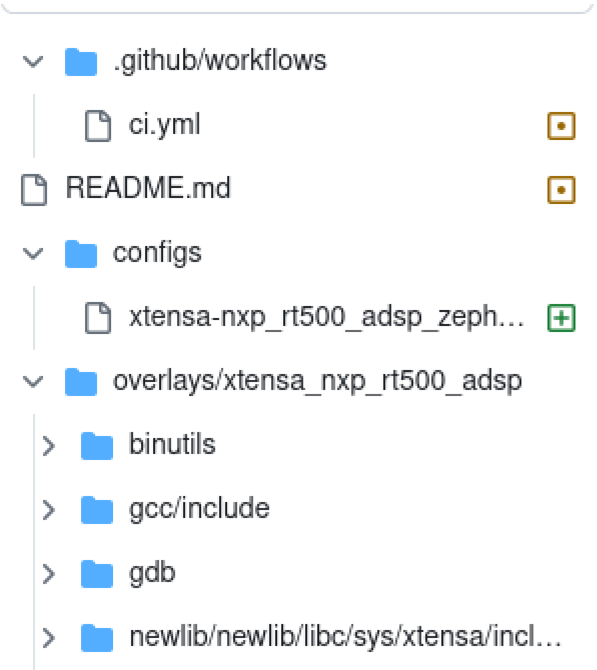
✕ xtensa: add rt500_adsp toolchain

```
This adds the Xtensa toolchain for Tensilica Fusion F1 DSP found on NXP
RT500 MCU family.

Signed-off-by: Daniel Baluta <daniel.baluta@nxp.com>
```

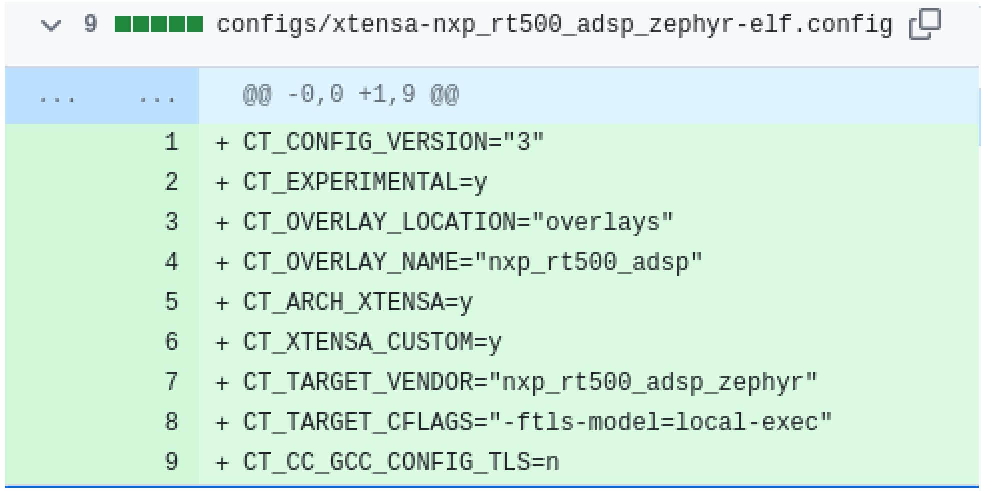
dbaluta committed on Aug 21, 2023

Notes and sources section

Add support for a new toolchain (2)

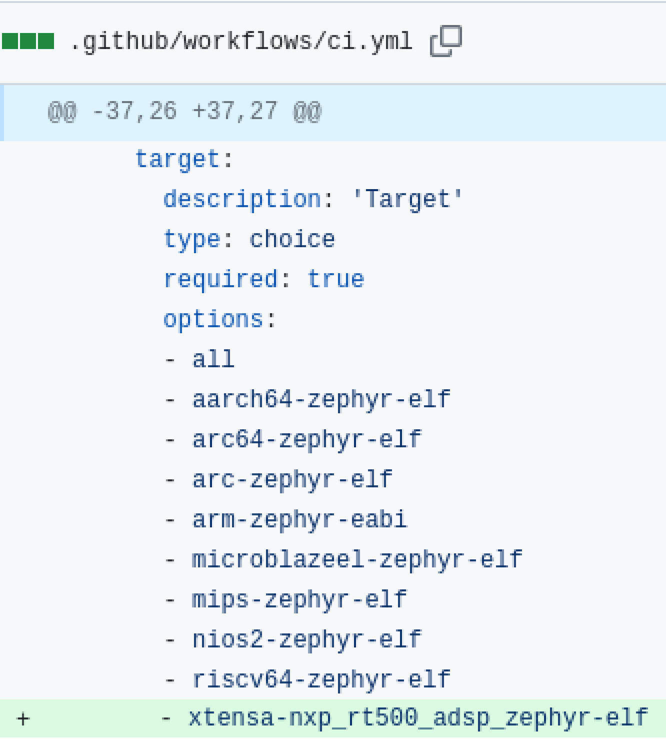


```
.github/workflows
    ci.yml
README.md
configs
    xtensa-nxp_rt500_adsp_zeph…
overlays/xtensa_nxp_rt500_adsp
    binutils
    gcc/include
    gdb
    newlib/newlib/libc/sys/xtensa/incl…
```



9 configs/xtensa-nxp_rt500_adsp_zephyr-elf.config

```
@@ -0,0 +1,9 @@
+ CT_CONFIG_VERSION="3"
+ CT_EXPERIMENTAL=y
+ CT_OVERLAY_LOCATION="overlays"
+ CT_OVERLAY_NAME="nxp_rt500_adsp"
+ CT_ARCH_XTENSA=y
+ CT_XTENSA_CUSTOM=y
+ CT_TARGET_VENDOR="nxp_rt500_adsp_zephyr"
+ CT_TARGET_CFLAGS="-ftls-model=local-exec"
+ CT_CC_GCC_CONFIG_TLS=n
```



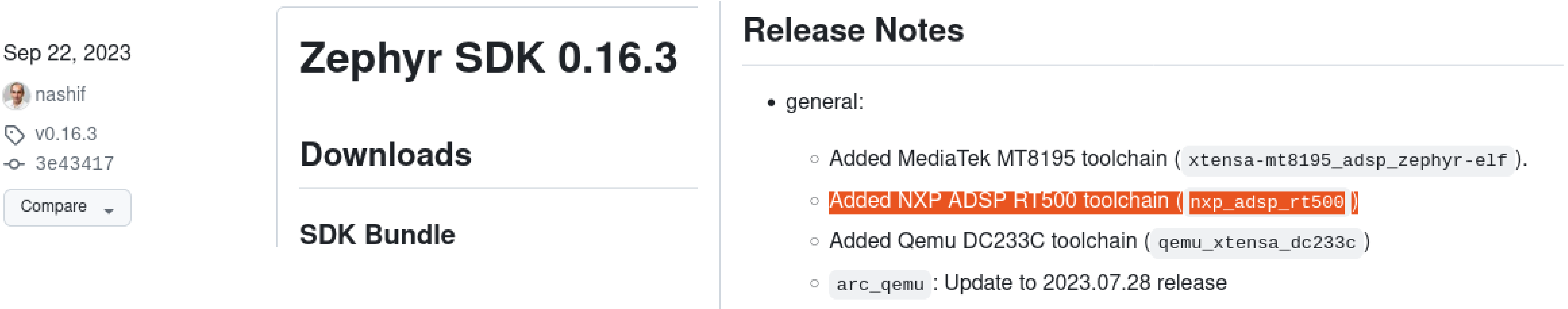
.github/workflows/ci.yml

```
@@ -37,26 +37,27 @@
      target:
        description: 'Target'
        type: choice
        required: true
        options:
        - all
        - aarch64-zephyr-elf
        - arc64-zephyr-elf
        - arc-zephyr-elf
        - arm-zephyr-eabi
        - microblazeel-zephyr-elf
        - mips-zephyr-elf
        - nios2-zephyr-elf
        - riscv64-zephyr-elf
+       - xtensa-nxp_rt500_adsp_zephyr-elf
```

Notes and sources section

Add support for a new toolchain (3)

- At least 2 weeks to get your changed merged
- At least 1 Release Candidate
- Typically allocate 1 month



Sep 22, 2023

nashif

v0.16.3

3e43417

Zephyr SDK 0.16.3

Downloads

SDK Bundle

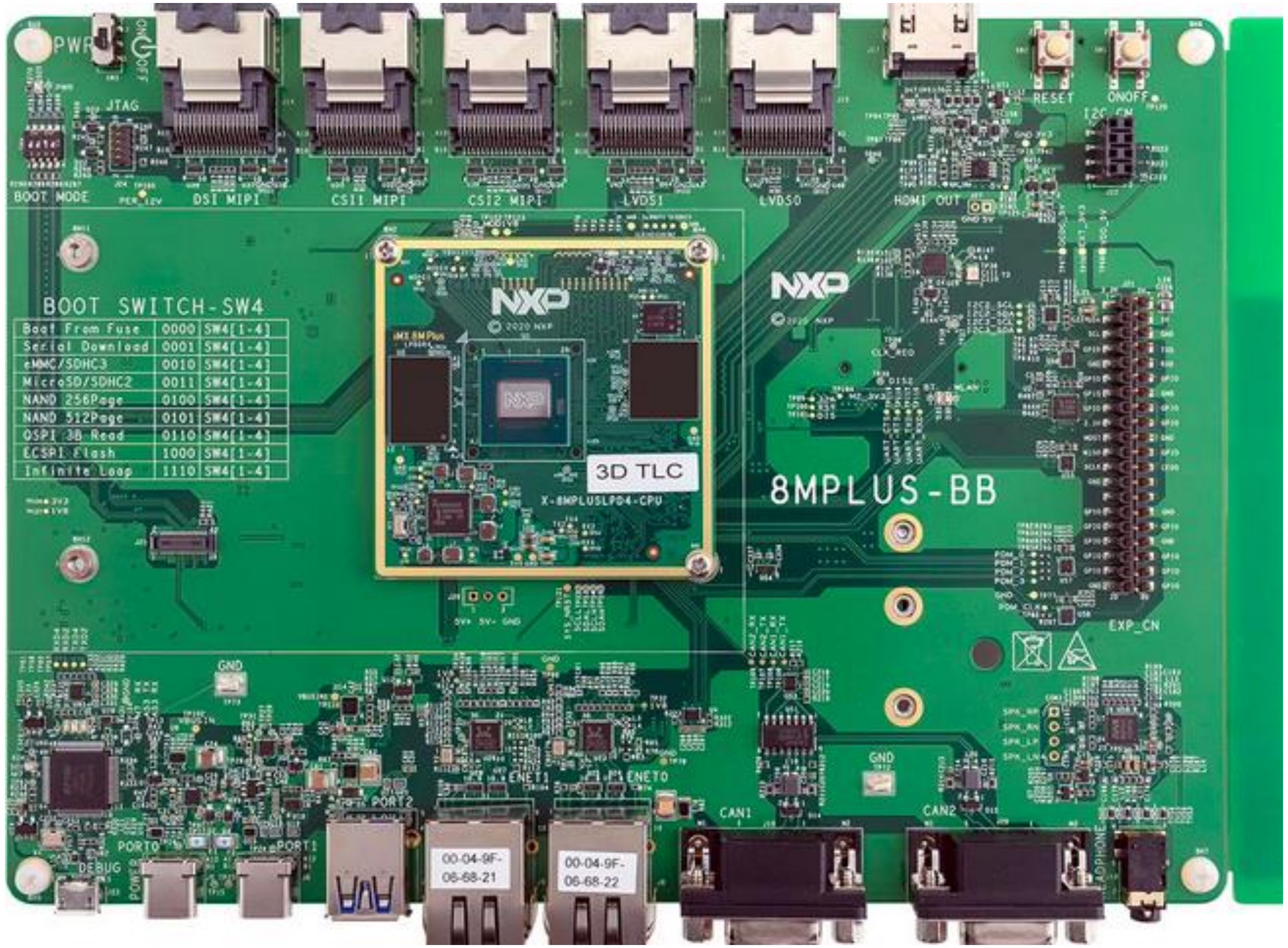
Release Notes

- general:
  - Added MediaTek MT8195 toolchain (`xtensa-mt8195_adsp_zephyr-elf`).
  - Added NXP ADSP RT500 toolchain (`nxp_adsp_rt500`)
  - Added Qemu DC233C toolchain (`qemu_xtensa_dc233c`)
  - `arc_qemu` : Update to 2023.07.28 release

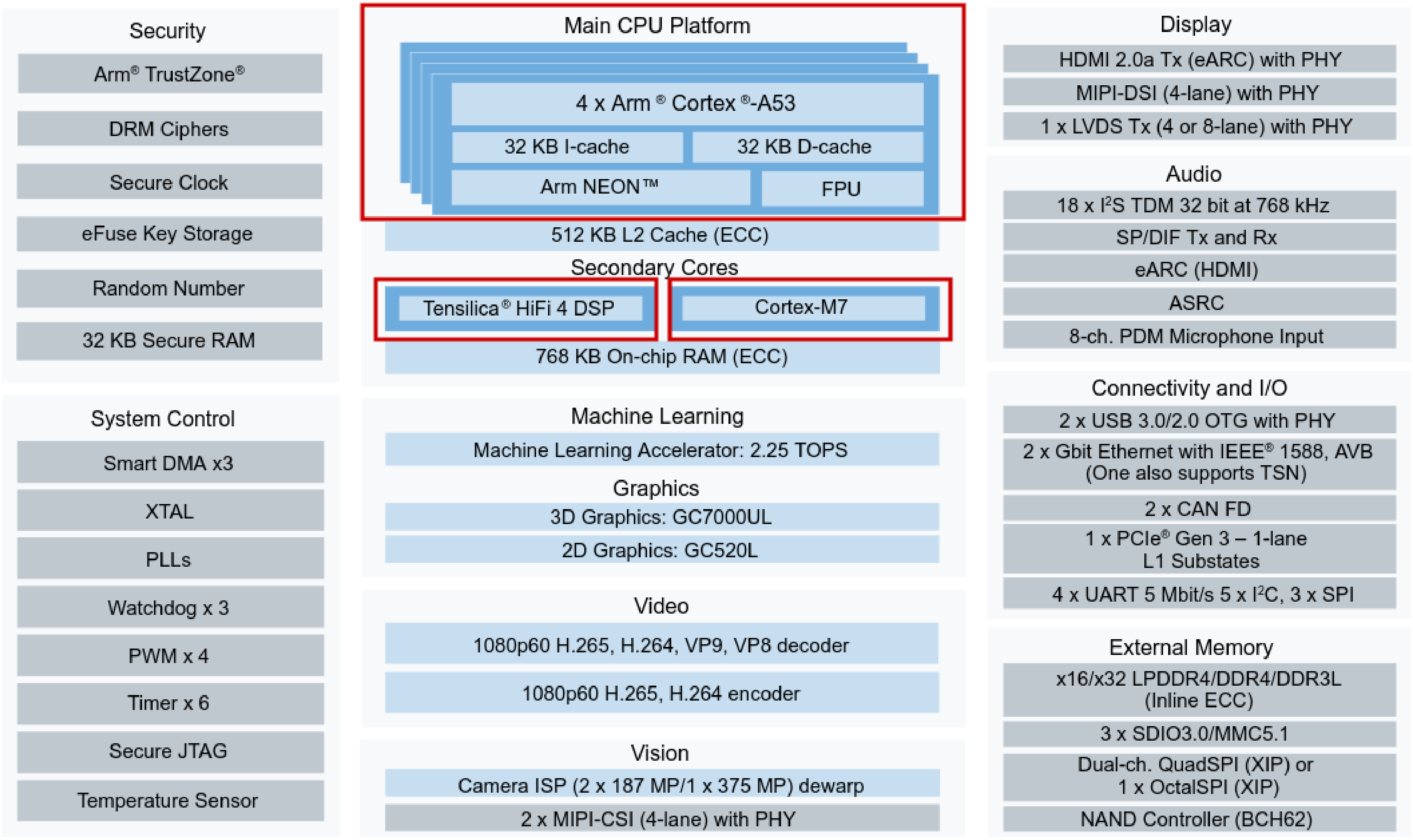
Board support

Get to know your hardware: NXP i.MX8M Plus EVK

- NXP i.MX8M Plus EVK
  - 4 x Arm Cortex A53
  - Arm Cortex-M7
  - Tensilica HIFI4 Audio DSP
- Existing Zephyr support
  - Architecture (arm64, arm, xtensa)
  - CPU (Cortex-A53, Cortex-M7, HIFI4 DSP)
  - SoC
  - Board



I.MX8M Plus – building blocks



Security

- Arm® TrustZone®
- DRM Ciphers
- Secure Clock
- eFuse Key Storage
- Random Number
- 32 KB Secure RAM

System Control

- Smart DMA x3
- XTAL
- PLLs
- Watchdog x 3
- PWM x 4
- Timer x 6
- Secure JTAG
- Temperature Sensor

Main CPU Platform

- 4 x Arm ® Cortex ®-A53
- 32 KB I-cache
- 32 KB D-cache
- Arm NEON™
- FPU
- 512 KB L2 Cache (ECC)

Secondary Cores

- Tensilica® HiFi 4 DSP
- Cortex-M7
- 768 KB On-chip RAM (ECC)

Machine Learning

- Machine Learning Accelerator: 2.25 TOPS

Graphics

- 3D Graphics: GC7000UL
- 2D Graphics: GC520L

Video

- 1080p60 H.265, H.264, VP9, VP8 decoder
- 1080p60 H.265, H.264 encoder

Vision

- Camera ISP (2 x 187 MP/1 x 375 MP) dewarp
- 2 x MIPI-CSI (4-lane) with PHY

Display

- HDMI 2.0a Tx (eARC) with PHY
- MIPI-DSI (4-lane) with PHY
- 1 x LVDS Tx (4 or 8-lane) with PHY

Audio

- 18 x I^{2}S TDM 32 bit at 768 kHz
- SP/DIF Tx and Rx
- eARC (HDMI)
- ASRC
- 8-ch. PDM Microphone Input

Connectivity and I/O

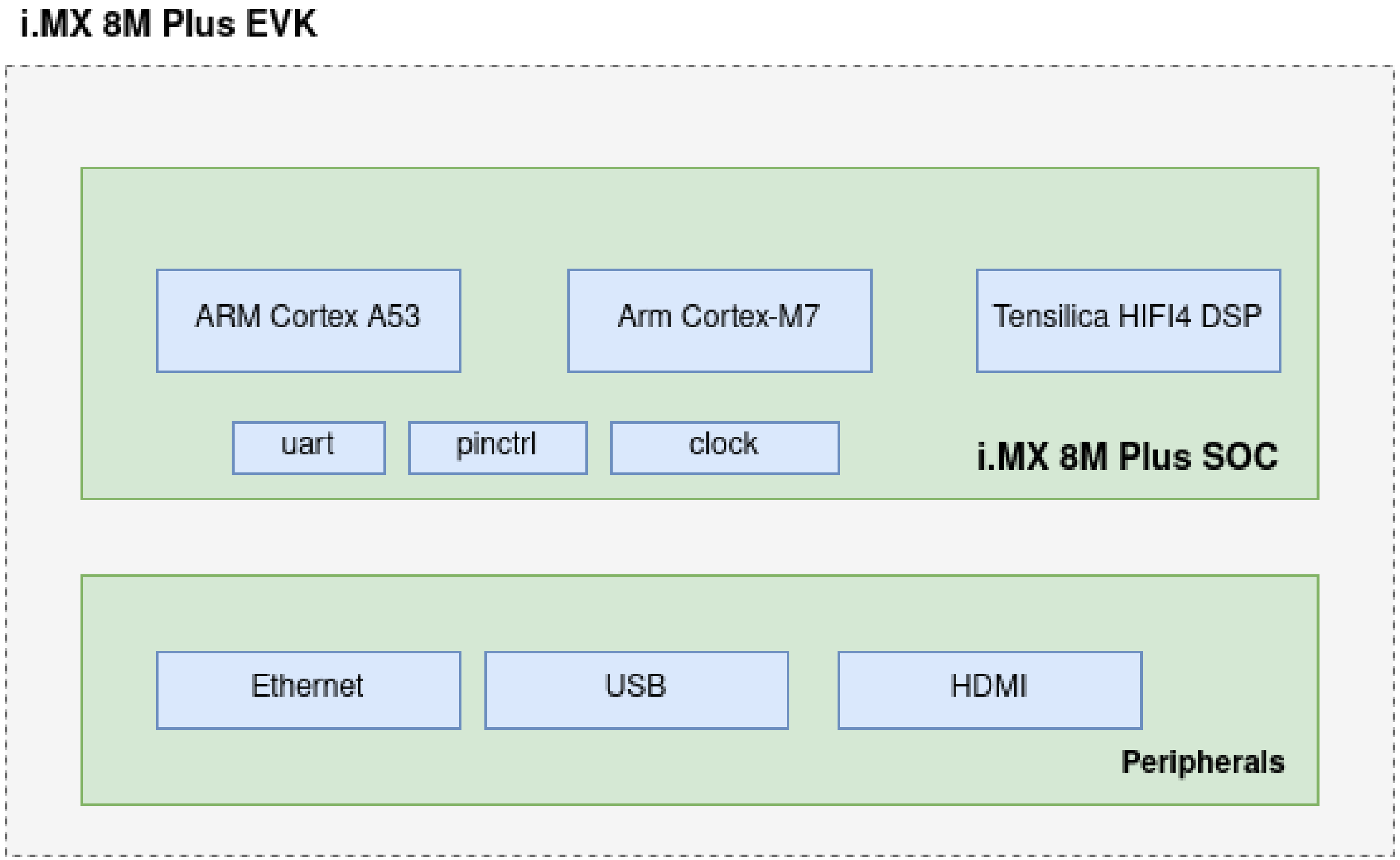
- 2 x USB 3.0/2.0 OTG with PHY
- 2 x Gbit Ethernet with IEEE® 1588, AVB (One also supports TSN)
- 2 x CAN FD
- 1 x PCIe® Gen 3 – 1-lane L1 Substates
- 4 x UART 5 Mbit/s 5 x I^{2}C, 3 x SPI

External Memory

- x16/x32 LPDDR4/DDR4/DDR3L (Inline ECC)
- 3 x SDIO3.0/MMC5.1
- Dual-ch. QuadSPI (XIP) or 1 x OctalSPI (XIP)
- NAND Controller (BCH62)

i.MX8M Plus – simplified view

- What is the minimal hardware support for running "Hello World" sample on HIFI4 DSP?
    - CPU
    - Uart
    - Pinctrl
    - Clocks

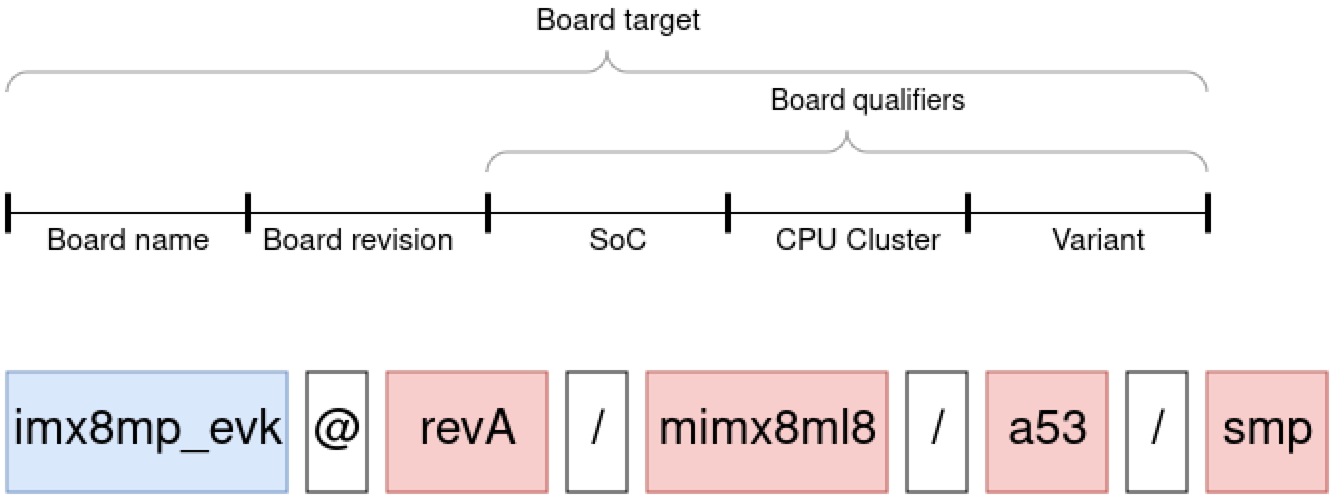


Introducing Hardware Model v2 (HWMv2)

- A model oriented toward Asymmetric Multi-Processing (AMP)
  - Identical CPUs are grouped into a **cpucluster**
  - Support cpuclusters of different architectures (AMP)
- Model the hardware into multiple layers
  - Architecture
  - CPU
  - SoC (family, series, cpu proper)
  - Board
- Merged into current working version Zephyr (3.7.0)
- Adding support for a board is
  - 80% configuration (Kconfig, defconfig, device tree)
  - 20% actual code .c code

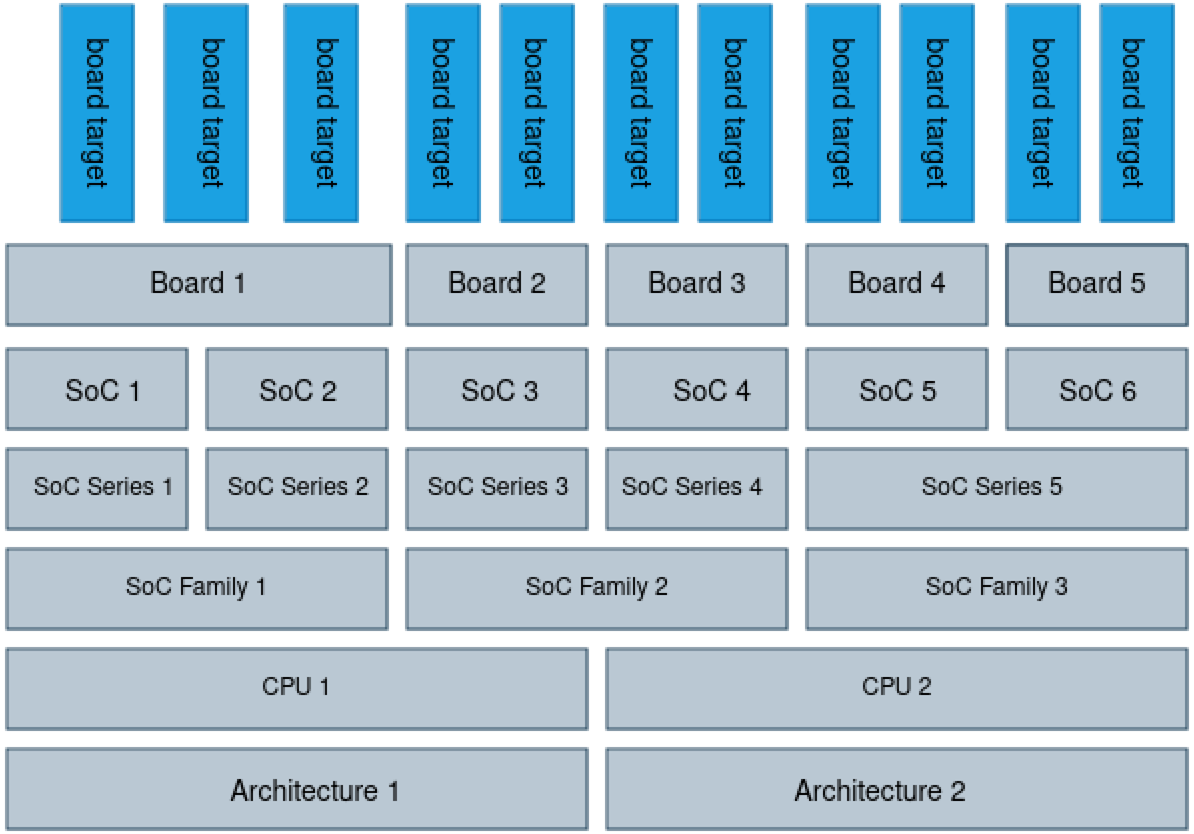
Board terminology

- Board – target system that can load and execute an application image
- Board target – full string used to compile an image for a particular hardware
- Board name – human readable name of board
- Board qualifiers – additional tokens to form the board target
- **west build –b imx8mp_evk@revA/mimx8ml8/a53/smp**



Hardware support hierarchy

- One board can contain multiple SoCs
- One SoC an contain multiple cpuclusters



board target board target board target board target board target board target board target board target board target board target board target

Board 1 Board 2 Board 3 Board 4 Board 5

SoC 1 SoC 2 SoC 3 SoC 4 SoC 5 SoC 6

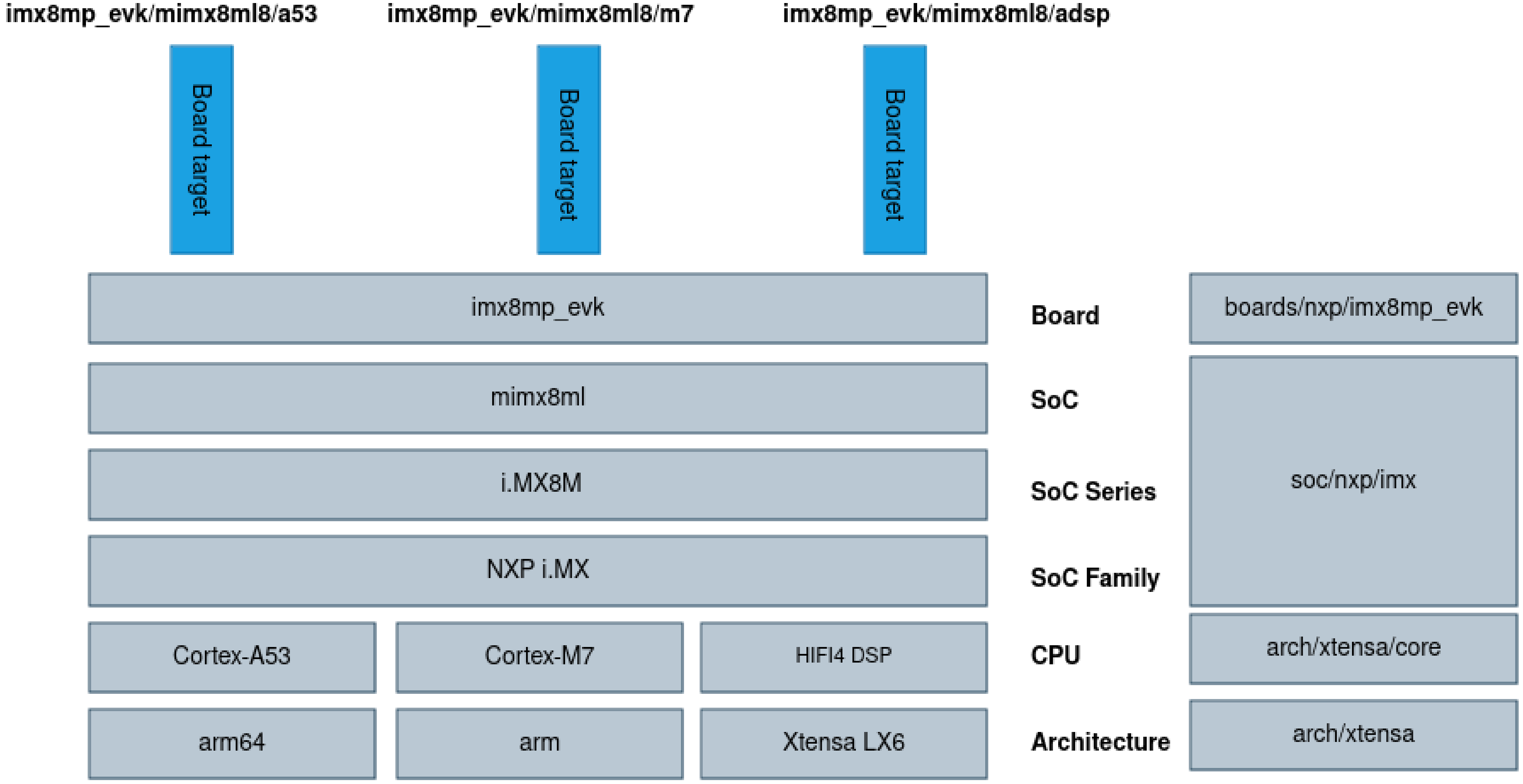
SoC Series 1 SoC Series 2 SoC Series 3 SoC Series 4 SoC Series 5

SoC Family 1 SoC Family 2 SoC Family 3

CPU 1 CPU 2

Architecture 1 Architecture 2

Mapping hardware hierarchy on i.MX8M Plus instance



imx8mp_evk/mimx8ml8/a53

imx8mp_evk/mimx8ml8/m7

imx8mp_evk/mimx8ml8/adsp

Board target

Board target

Board target

imx8mp_evk — **Board** — boards/nxp/imx8mp_evk

mimx8ml — **SoC** — soc/nxp/imx

i.MX8M — **SoC Series** — soc/nxp/imx

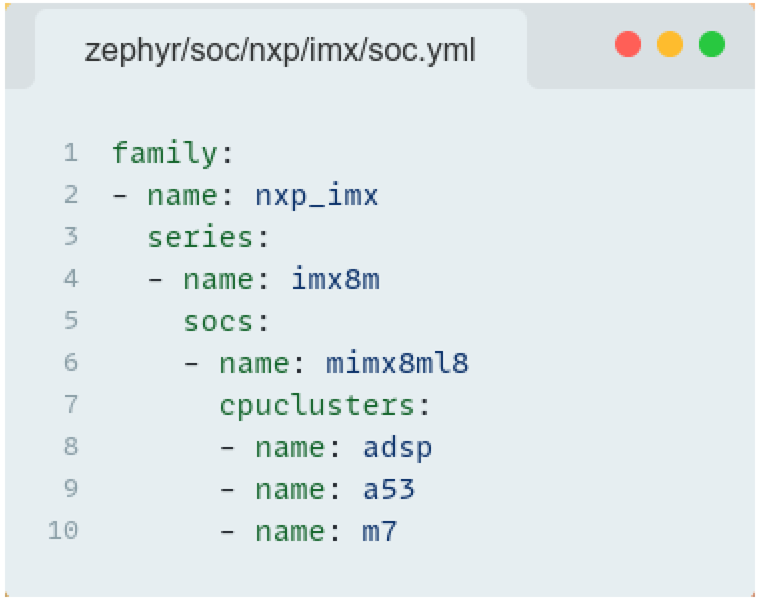
NXP i.MX — **SoC Family** — soc/nxp/imx

Cortex-A53, Cortex-M7, HIFI4 DSP — **CPU** — arch/xtensa/core

arm64, arm, Xtensa LX6 — **Architecture** — arch/xtensa

SoC hierarchy description

- soc.yml file
- Describes: family, series, SoC
- A SoC can have multiple CPU clusters
- A CPU cluster contains multiple CPU of the same type

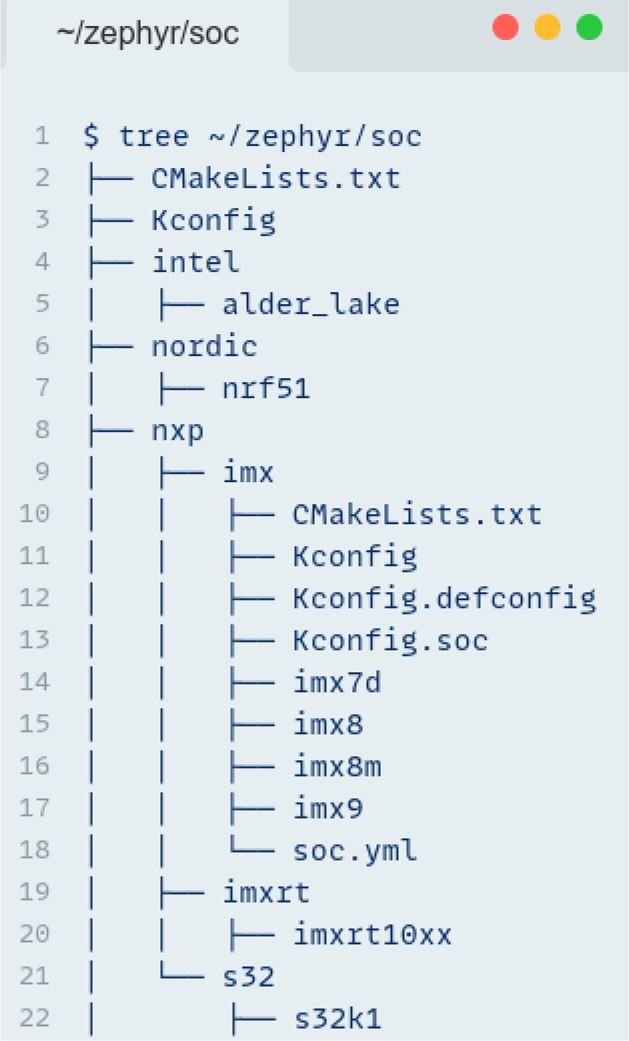


zephyr/soc/nxp/imx/soc.yml

```yaml
family:
- name: nxp_imx
  series:
  - name: imx8m
    socs:
    - name: mimx8ml8
      cpuclusters:
      - name: adsp
      - name: a53
      - name: m7
```

SoC family

- group of related chips from a vendor that share
  - similar architecture
  - similar features
  - similar design principles
- zephyr/soc/<vendor>/<soc-family>
- SoC families
  - NXP i.MX
  - NXP i.MXRT
  - NXP S32
  - Intel Alder Lake
  - Nordic NRF51

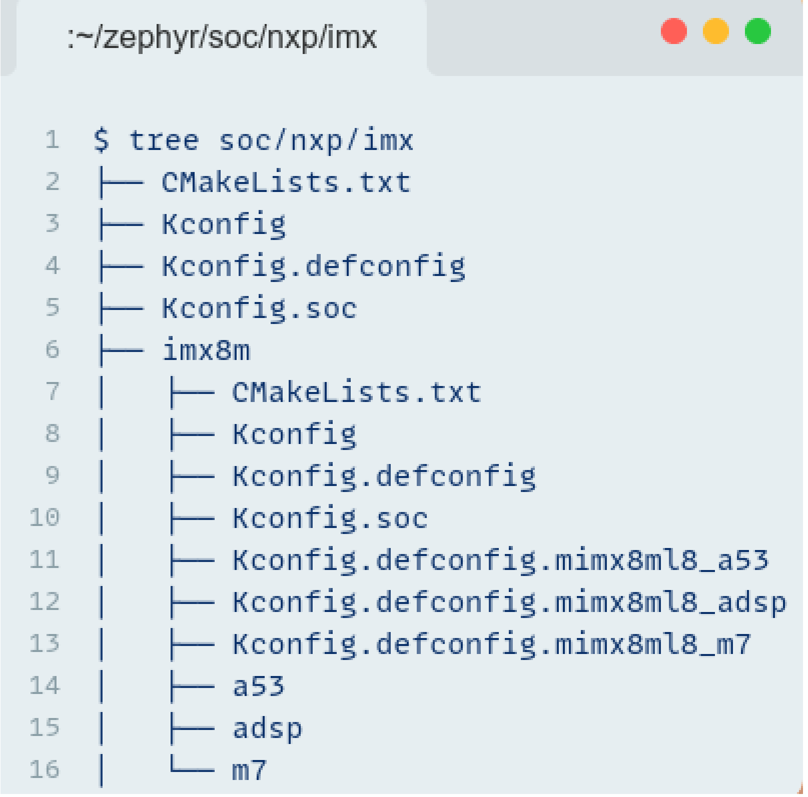


~/zephyr/soc

```
$ tree ~/zephyr/soc
├── CMakeLists.txt
├── Kconfig
├── intel
│   ├── alder_lake
├── nordic
│   ├── nrf51
├── nxp
│   ├── imx
│   │   ├── CMakeLists.txt
│   │   ├── Kconfig
│   │   ├── Kconfig.defconfig
│   │   ├── Kconfig.soc
│   │   ├── imx7d
│   │   ├── imx8
│   │   ├── imx8m
│   │   ├── imx9
│   │   └── soc.yml
│   ├── imxrt
│   │   ├── imxrt10xx
│   └── s32
│       ├── s32k1
```

SoC series

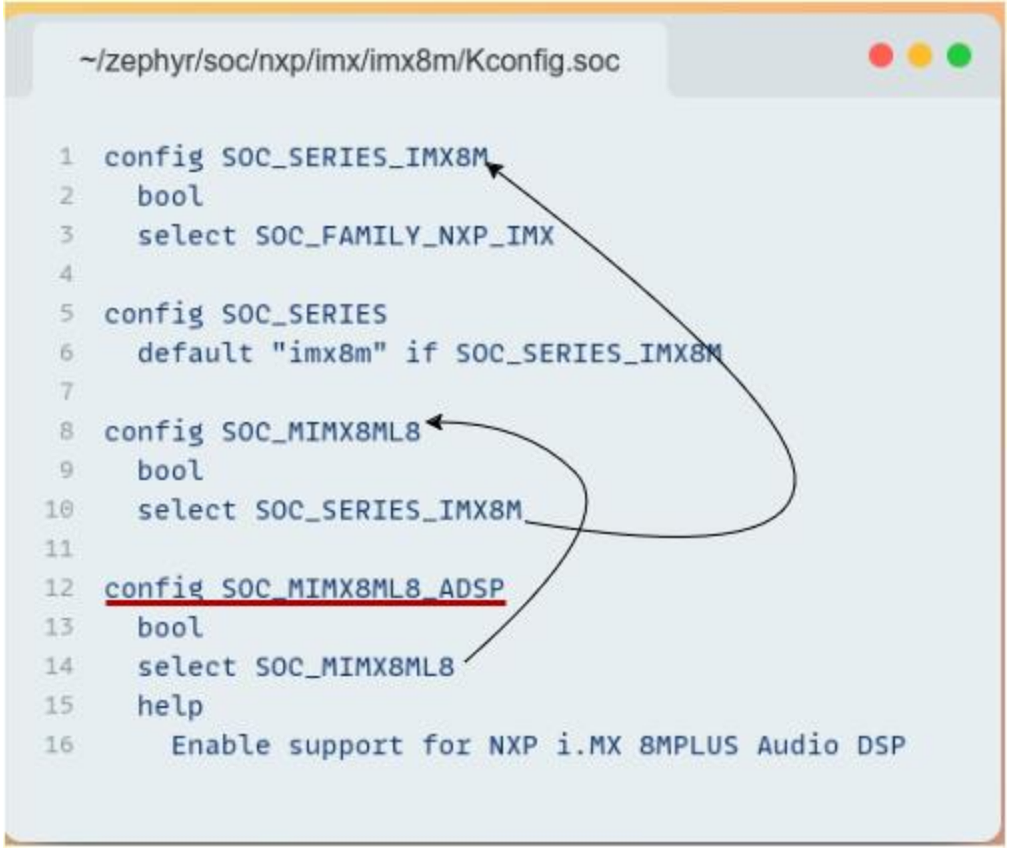
- Refers to specific iteration / version with a SoC family
- SoC series
  - NXP i.MX8M
  - NXP i.MX8ULP
  - NXP i.MX9



```
$ tree soc/nxp/imx
├── CMakeLists.txt
├── Kconfig
├── Kconfig.defconfig
├── Kconfig.soc
├── imx8m
│   ├── CMakeLists.txt
│   ├── Kconfig
│   ├── Kconfig.defconfig
│   ├── Kconfig.soc
│   ├── Kconfig.defconfig.mimx8ml8_a53
│   ├── Kconfig.defconfig.mimx8ml8_adsp
│   ├── Kconfig.defconfig.mimx8ml8_m7
│   ├── a53
│   ├── adsp
│   └── m7
```

Navigating through kconfig selection

- Enable support for ADSP
  - This brings in SoC
    - Which in turns brings in soc series
      - Which in turn brings in soc family
- But where and how enables the ADSP
  - Answer:
    - A specific board!

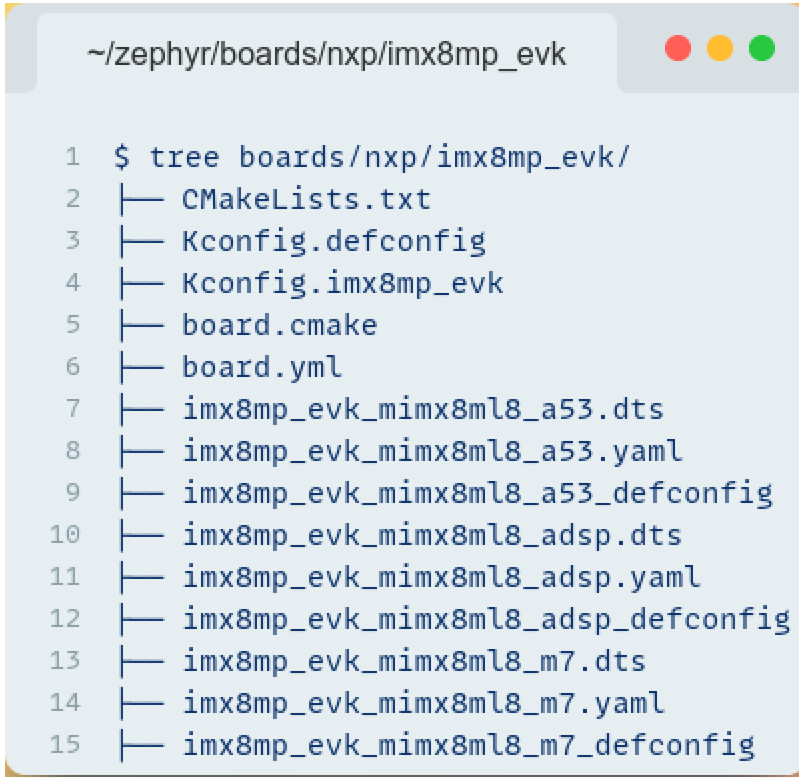


~/zephyr/soc/nxp/imx/imx8m/Kconfig.soc

```
config SOC_SERIES_IMX8M
  bool
  select SOC_FAMILY_NXP_IMX

config SOC_SERIES
  default "imx8m" if SOC_SERIES_IMX8M

config SOC_MIMX8ML8
  bool
  select SOC_SERIES_IMX8M

config SOC_MIMX8ML8_ADSP
  bool
  select SOC_MIMX8ML8
  help
    Enable support for NXP i.MX 8MPLUS Audio DSP
```

What is a board?

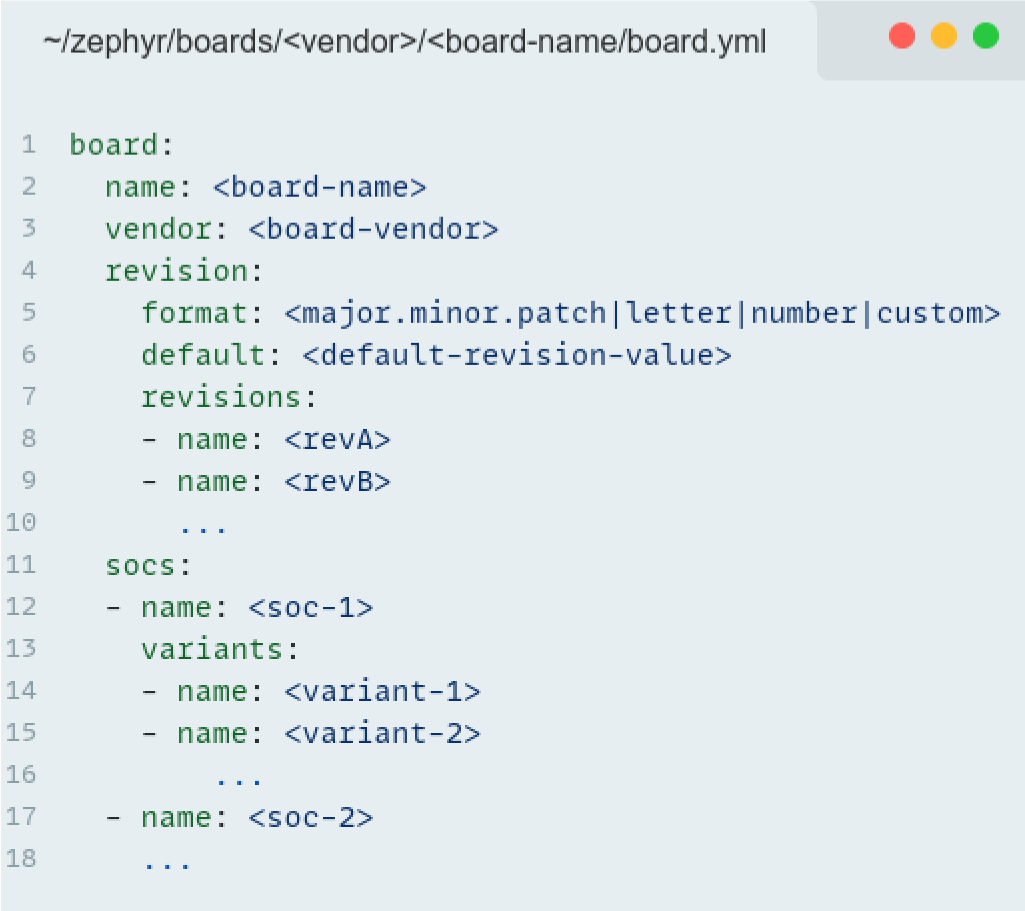
- Target hardware we want to build an application for
  - Hardware is decoupled from the application
  - Location: zephyr/boards/<vendor>/<Your-Board-Name/
  - Board directory contains
    - board.yml
    - Kconfig.imx8mp_evk
    - imx8mp_evk_<qualifiers>.dts
  - Optional files
    - Kconfig, Kconfig.defconfig
    - Imx8mp_evk_<qualifiers>_defconfig
    - Imx8mp_evk_<qualifiers>.yaml
    - Board.cmake for debugging



```
$ tree boards/nxp/imx8mp_evk/
├── CMakeLists.txt
├── Kconfig.defconfig
├── Kconfig.imx8mp_evk
├── board.cmake
├── board.yml
├── imx8mp_evk_mimx8ml8_a53.dts
├── imx8mp_evk_mimx8ml8_a53.yaml
├── imx8mp_evk_mimx8ml8_a53_defconfig
├── imx8mp_evk_mimx8ml8_adsp.dts
├── imx8mp_evk_mimx8ml8_adsp.yaml
├── imx8mp_evk_mimx8ml8_adsp_defconfig
├── imx8mp_evk_mimx8ml8_m7.dts
├── imx8mp_evk_mimx8ml8_m7.yaml
├── imx8mp_evk_mimx8ml8_m7_defconfig
```

Board.yaml

- Describes the metadata of the board
  - Board name
  - Vendor
  - SoCs and their variants
  - Note: cpuclusters are not described here
    - They are inherited from soc.yml
  - It is possible to have multiple boards in board.yml

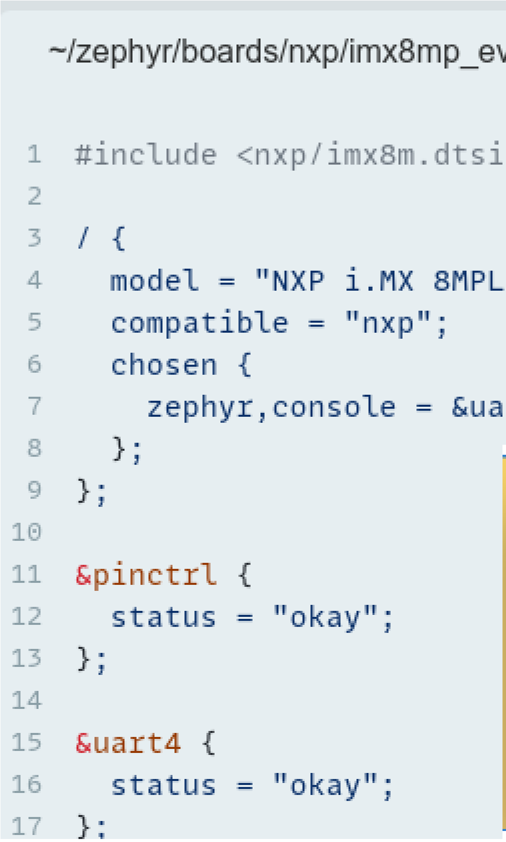


~/zephyr/boards/<vendor>/<board-name/board.yml

```
board:
  name: <board-name>
  vendor: <board-vendor>
  revision:
    format: <major.minor.patch|letter|number|custom>
    default: <default-revision-value>
    revisions:
    - name: <revA>
    - name: <revB>
      ...
  socs:
  - name: <soc-1>
    variants:
    - name: <variant-1>
    - name: <variant-2>
        ...
  - name: <soc-2>
    ...
```

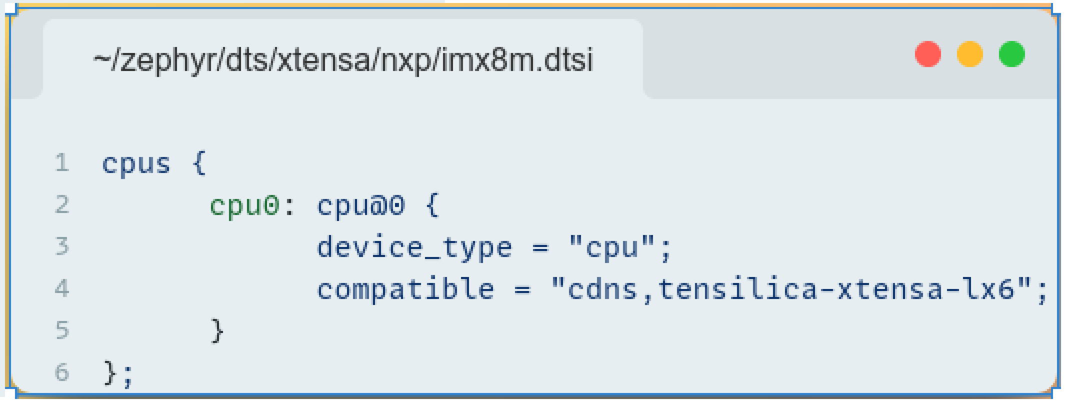
Board target device tree

- Hardware description
- cpus and minimal soc devices to run “Hello world”



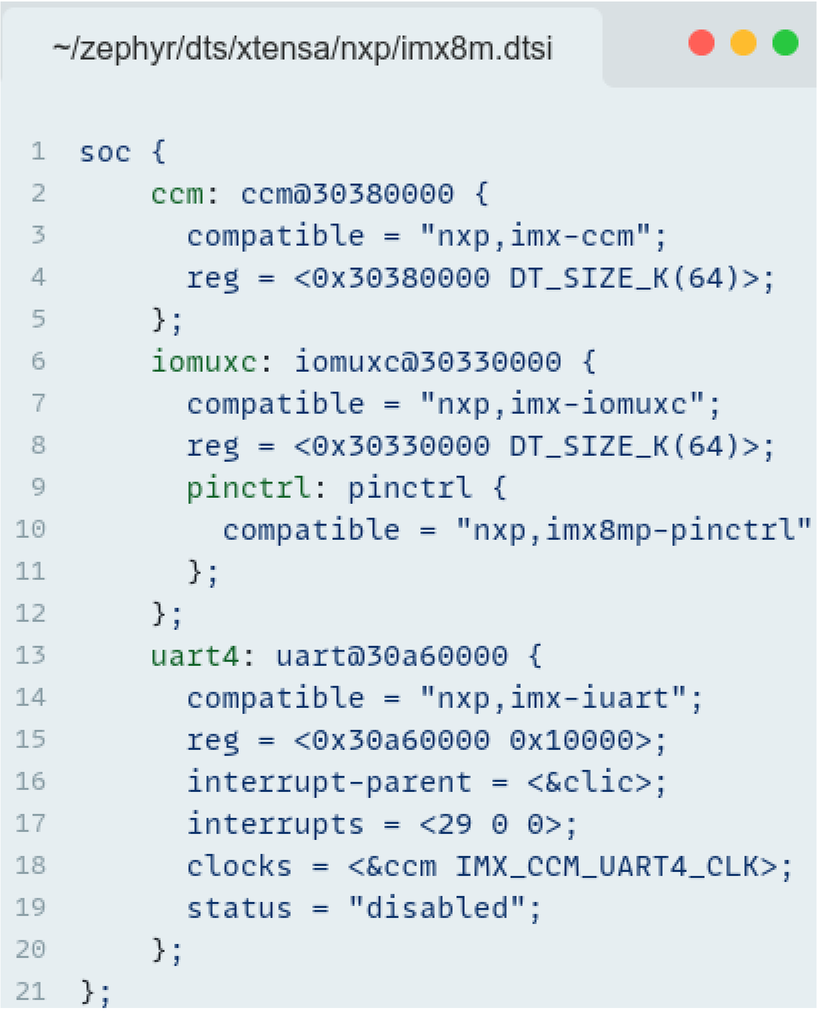
~/zephyr/boards/nxp/imx8mp_evk/<target>.dts

```
#include <nxp/imx8m.dtsi>

/ {
  model = "NXP i.MX 8MPLUS Audio DSP";
  compatible = "nxp";
  chosen {
    zephyr,console = &uart4;
  };
};

&pinctrl {
  status = "okay";
};

&uart4 {
  status = "okay";
};
```



~/zephyr/dts/xtensa/nxp/imx8m.dtsi

```
cpus {
        cpu0: cpu@0 {
                device_type = "cpu";
                compatible = "cdns,tensilica-xtensa-lx6";
        }
};
```

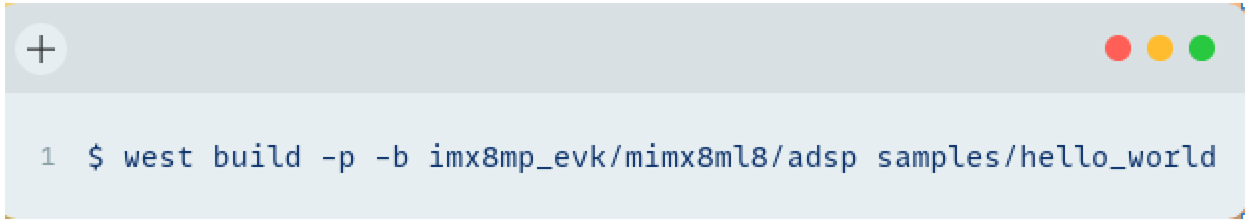


~/zephyr/dts/xtensa/nxp/imx8m.dtsi

```
soc {
    ccm: ccm@30380000 {
      compatible = "nxp,imx-ccm";
      reg = <0x30380000 DT_SIZE_K(64)>;
    };
    iomuxc: iomuxc@30330000 {
      compatible = "nxp,imx-iomuxc";
      reg = <0x30330000 DT_SIZE_K(64)>;
      pinctrl: pinctrl {
        compatible = "nxp,imx8mp-pinctrl"
      };
    };
    uart4: uart@30a60000 {
      compatible = "nxp,imx-iuart";
      reg = <0x30a60000 0x10000>;
      interrupt-parent = <&clic>;
      interrupts = <29 0 0>;
      clocks = <&ccm IMX_CCM_UART4_CLK>;
      status = "disabled";
    };
};
```

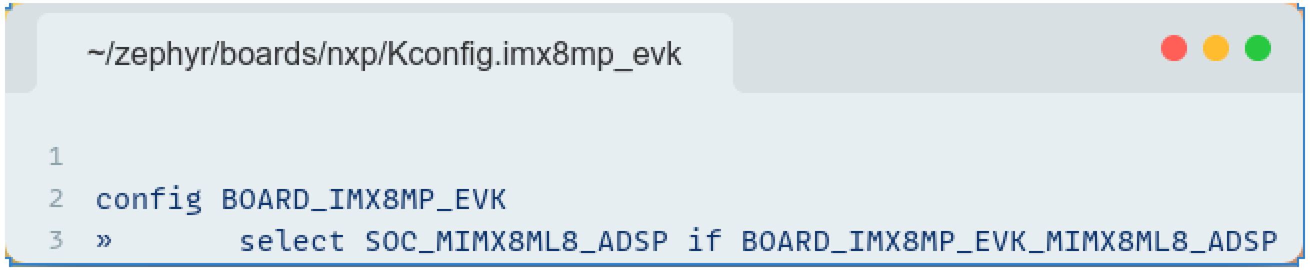
Board Kconfig.imx8mp_evk

- Final piece of the puzzle
- Base software configuration for selecting the SoC
- How does BOARD_IMX8MP_EVK_MIMX8ML8_ADSP gets constructed?



```
$ west build -p -b imx8mp_evk/mimx8ml8/adsp samples/hello_world
```

- Boards/Kconfig.v2 creates BOARD_${BOARD_TARGET_STRING}



~/zephyr/boards/nxp/Kconfig.imx8mp_evk

```

config BOARD_IMX8MP_EVK
»       select SOC_MIMX8ML8_ADSP if BOARD_IMX8MP_EVK_MIMX8ML8_ADSP
```

Working with Zephyr community

Working with the community

- https://docs.zephyrproject.org/latest/contribute/index.html

- Read the guidelines
  - Contribution guidelines
  - Coding guidelines
  - Proposals and RFCs
  - Pull Requests and issues
  - Documentation guidelines

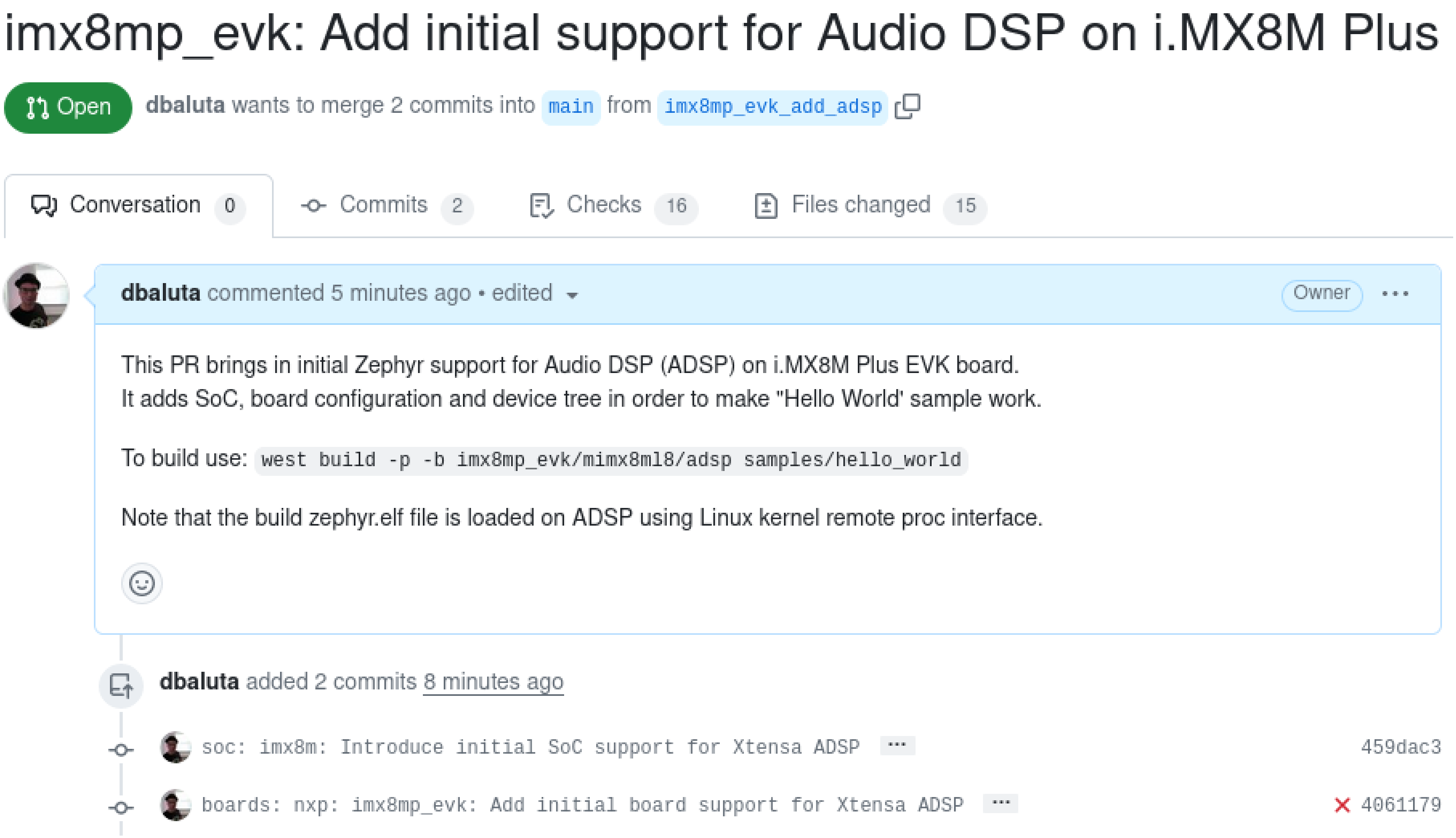
Contribution guidelines

- Get familiar with Zephyr tree structure and development environment
- Check “Good First Issue” label
  - Sending a small contribution can walk you through the entire submission process
- Check Coding Style
  - Always run ./scripts/checkpatch.pl to verify your change
- Check "How to write a good commit message"
  - Run gitlint
  - explain WHY the change is needed
  - Add your Signed-off-by line. Use git commit –s

How to speedup Pull Request acceptance

- A Pull Request should be fairly small
  - Up to 3 – 5 patches
  - Each patch should contain a single logical change
- Make sure you always test your code
- Keep CI clean
  - Enable at least build tests for your new board
- Work with reviewers to address comments
  - Answer to each comment and agree with reviewer on a resolution
- Write a small changelog of your changes after each new version of the PR

Example PR



imx8mp_evk: Add initial support for Audio DSP on i.MX8M Plus

dbaluta wants to merge 2 commits into `main` from `imx8mp_evk_add_adsp`

dbaluta commented 5 minutes ago • edited

This PR brings in initial Zephyr support for Audio DSP (ADSP) on i.MX8M Plus EVK board.
It adds SoC, board configuration and device tree in order to make "Hello World' sample work.

To build use: `west build -p -b imx8mp_evk/mimx8ml8/adsp samples/hello_world`

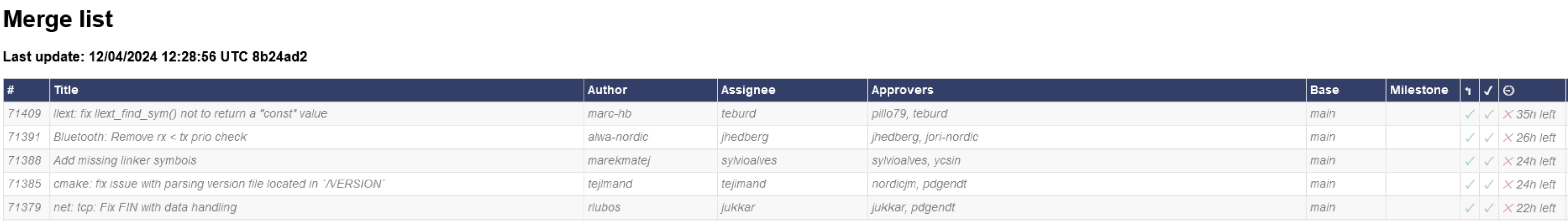
Note that the build zephyr.elf file is loaded on ADSP using Linux kernel remote proc interface.

dbaluta added 2 commits 8 minutes ago

```
soc: imx8m: Introduce initial SoC support for Xtensa ADSP             459dac3
boards: nxp: imx8mp_evk: Add initial board support for Xtensa ADSP  ✕ 4061179
```

After sending the Pull Request

- Be patient!
- PR merged no sooner than 2 days but typically can take up to 2 weeks
- Need 1 R-b from asigneed and 1 R-b from other reviewer
- Ping asignee/reviewers after 1 week
- Ask for help on Discord #pr-help channel after 2 weeks
- Check status: https://merge-list.zephyrproject.io/



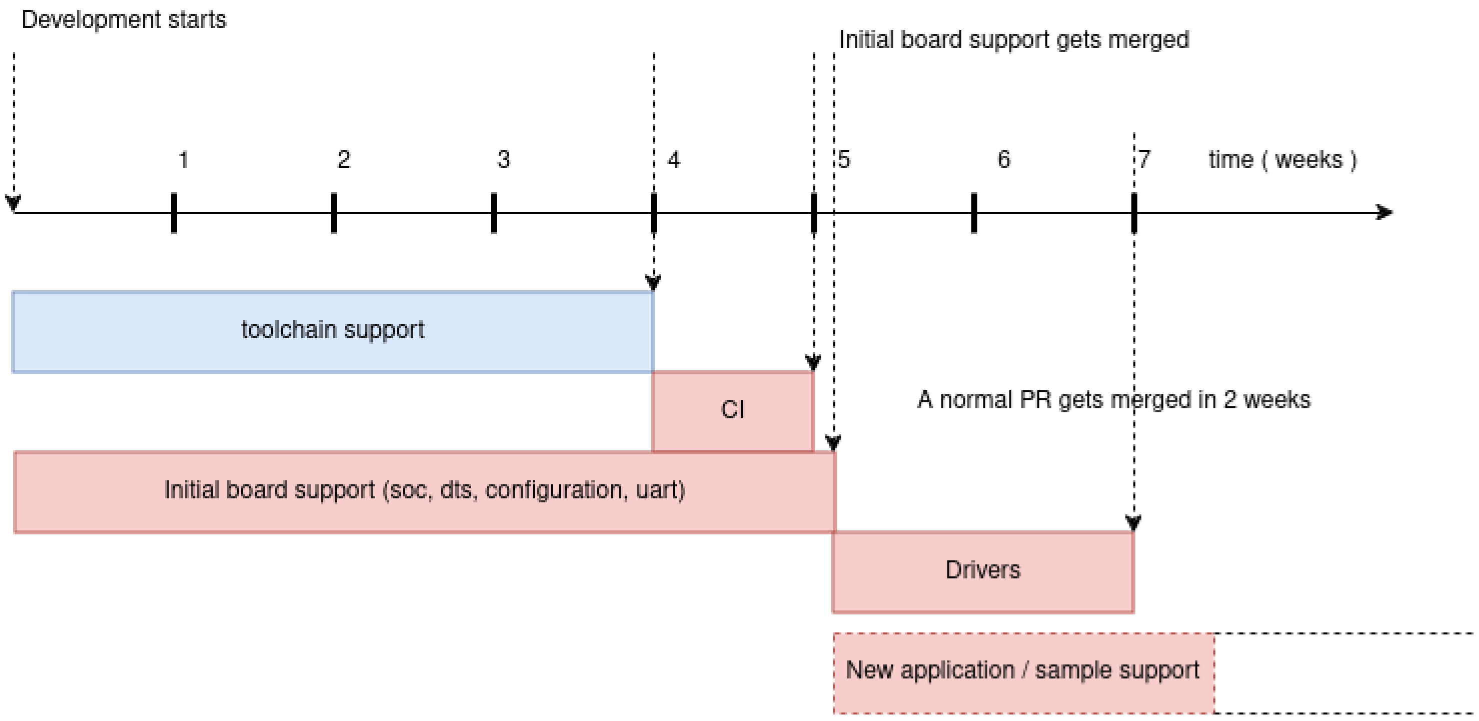
Merge list

Last update: 12/04/2024 12:28:56 UTC 8b24ad2

#	Title	Author	Assignee	Approvers	Base	Milestone	↰	✓	⊝
71409	llext: fix llext_find_sym() not to return a "const" value	marc-hb	teburd	pillo79, teburd	main		✓	✓	✗ 35h left
71391	Bluetooth: Remove rx < tx prio check	alwa-nordic	jhedberg	jhedberg, jori-nordic	main		✓	✓	✗ 26h left
71388	Add missing linker symbols	marekmatej	sylvioalves	sylvioalves, ycsin	main		✓	✓	✗ 24h left
71385	cmake: fix issue with parsing version file located in `/VERSION`	tejlmand	tejlmand	nordicjm, pdgendt	main		✓	✓	✗ 24h left
71379	net: tcp: Fix FIN with data handling	rlubos	jukkar	jukkar, pdgendt	main		✓	✓	✗ 22h left

Final thoughts and timeline recap

- Be patient!
- Work with reviewers and maintainers
- Address comments
- Explain in detail your decisions
- Trival PRs take up to 2 days
- Normal PRs take up to 2 weeks


Development starts
Initial board support gets merged
1
2
3
4
5
6
7
time (weeks)
toolchain support
CI
A normal PR gets merged in 2 weeks
Initial board support (soc, dts, configuration, uart)
Drivers
New application / sample support

Get in touch

Daniel BALUTA
daniel.baluta@nxp.com

nxp.com

nxp.com