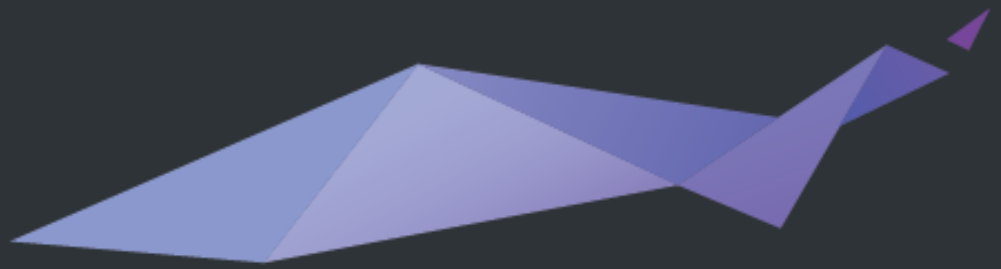


Using SoC Vendor HALs in the Zephyr Project

Maureen Helm, NXP

What is Zephyr™ Project?

Zephyr™

Small Footprint RTOS	Truly Open Source	Cross Architecture
• As small as 8KB • Enables applications code to scale • Configurable • Modular	• Apache 2.0 License • Hosted by Linux Foundation • Transparent development	• ARM • x86 • ARC • NIOS-II • RISC-V • Xtensa

Zephyr Eco-System

Zephyr OS

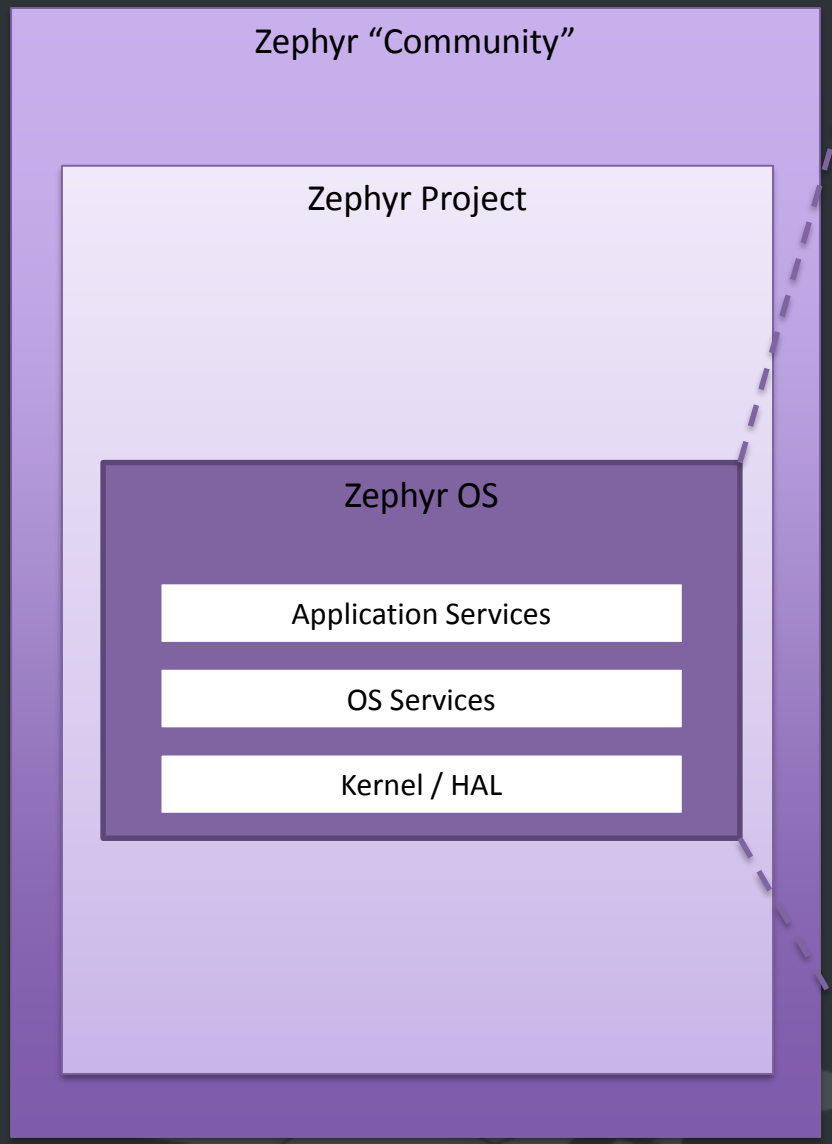
- The kernel and HAL
- OS Services such as IPC, Logging, file systems, crypto

Zephyr Project

- SDK, tools and development environment
- Additional middleware and features
- Device Management and
- Bootloader

Zephyr Community

- 3rd Party modules and libraries
- Support for Zephyr in 3rd party projects, for example: Jerryscript, Micropython, Iotivity



Kernel / HAL

- Scheduler
- Kernel objects and services
- low-level architecture and board support
- power management hooks and low level interfaces to hardware

OS Services and Low level APIs

- Platform specific drivers
- Generic implementation of I/O APIs
- File systems, Logging, Debugging and IPC
- Cryptography Services
- Networking and Connectivity
- Device Management

Application Services

- High Level APIs
- Access to standardized data models
- High Level networking protocols

Why Use SoC Vendor HALs?

- Core and peripheral register definitions
- Low-level stateless peripheral drivers
- Bare metal transactional drivers

- Maintained by the SoC vendor
- License is often permissive
- Used in other projects, not just Zephyr
  - Greater maturity and QA testing

- Simplifies adding new SoCs and drivers to Zephyr

Tradeoffs

- Code is maintained elsewhere
  - More difficult to update upstream
  - License new to Zephyr, or not compatible with Apache 2.0
- Code is used elsewhere
  - APIs not compatible
  - Features not implemented

HALs Currently in Use

Vendor	HAL	SoC Family
Arm	CMSIS	SAM, nRF5, Kinetis, EFM32, STM32
Atmel	ASF	SAM
Intel	QMSI	Quark
Nordic	MDK	nRF5
NXP	MCUXpresso SDK	Kinetis
SiLabs	Gecko SDK	EFM32
ST	STM32Cube SDK	STM32
TI	SimpleLink SDK	SimpleLink

Levels of Abstraction

- Transactional HAL driver
  - Thin Zephyr shim driver
  - MCUX and QMSI
- Low-level, stateless HAL driver
  - Larger Zephyr shim driver
  - STM32
- Register definitions only
  - Near-native Zephyr driver
  - Atmel, Nordic

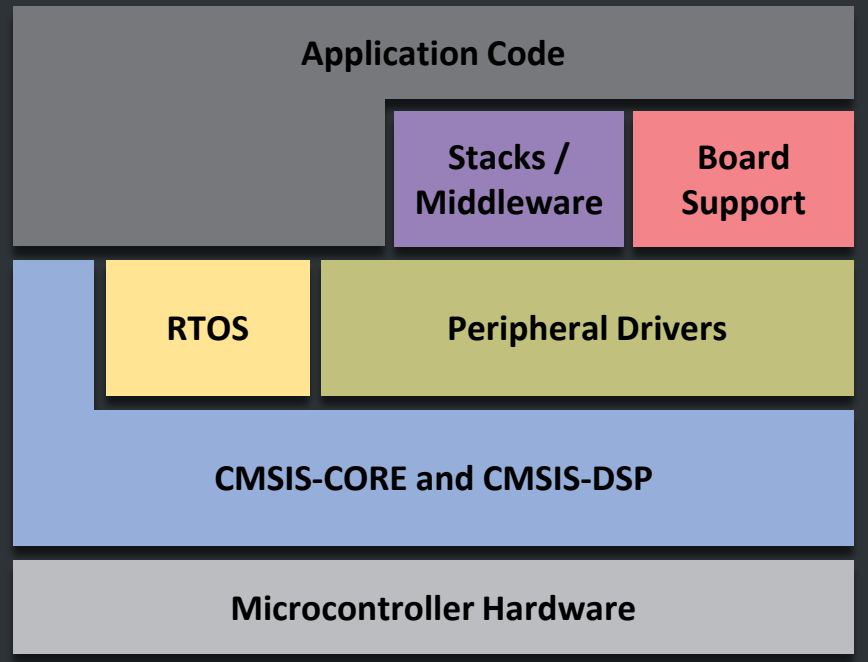
CMSIS

- CMSIS = Cortex Microcontroller System Interface Standard
  - CMSIS-Core, -SVD, -DSP, -Driver, etc.
  - Defined by Arm
- CMSIS-Core standardizes processor core access and peripheral definitions
  - Arm provides generic Cortex-M header files
  - Vendors provide device header files
- Zephyr kernel port uses CMSIS to access NVIC, SCB registers
- Zephyr drivers use CMSIS to access peripheral registers

MCUXpresso SDK

Zephyr™

- Common enablement for NXP Cortex-M MCUs
- Peripheral register definitions
  - CMSIS-Core compatible
  - Generated from same source as hardware documentation
- Bare metal peripheral drivers
  - Similar APIs across Kinetis and LPC families (UART/LPUART/LPSCI, SPI/DSPI/LPSPI, etc.)
  - Stateless and transactional abstraction levels
  - Minimal inter-dependencies
  - Tested on all hardware platforms
- IDE Example projects
  - Demonstrate peripheral driver APIs



The ext/ Folder

- Externally maintained source code lives in ext/
- Includes SoC vendor HALs, mbedTLS, TinyCrypt, FatFS, Segger RTT
- Permissively-licensed, but not necessarily Apache 2.0
  - Many are BSD 3-clause
- Imported as-is with minimal modification
  - If modifications are needed, make them in a separate commit
- Exempt from Zephyr coding style

Import Process

1. Submit “ Code Component README” with source code patch to the TSC
2. Are the TSC members agreeable (via TSC vote) to the proposal?
   - **NO** – software is rejected
   - **YES** – TSC chair forwards README to the Governing Board for review
3. Does any member of the Governing Board raise concern over inclusion in 2 week review period
   - NO – software is accepted, and README is included within the project’s documentation (in tree and external sites as appropriate)
   - YES – Governing Board will meet to discuss whether to override the TSC approval or identify other approaches

Code Component README

Origin: XYZ (project that hosts original code)

Status: The current version supported in Zephyr is XYZ 1.4. See https://github.com/xyz/releases for more details.

Purpose: Hardware Abstraction Layer (HAL) for ABC Microcontroller products

Description: XYZ is a Hardware Abstraction Layer (HAL) for ABC Microcontroller products. It currently supports the following SoCs:

- ZBC DFG Microcontroller
- ZBC DFH Microcontroller

Dependencies: XYZ assumes ….. is available to link to.

URL: https://github.com/xyz

commit: 08ded7f21529c39e5133688ffb93a9d0c94e5c6e

Maintained-by: External

License: BSD-3-Clause

License Link: https://github.com/xyz/LICENSE

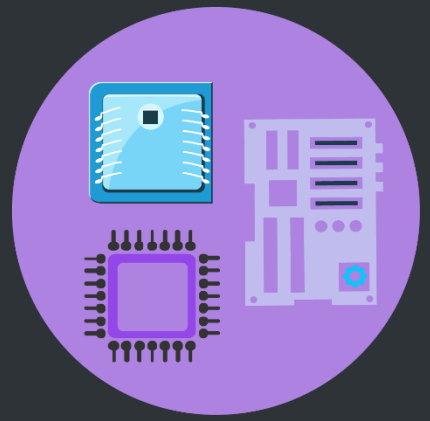
Adding a new Arm SoC

- Does the SoC belong to an existing family or series?
- Are there CMSIS headers available?
  - Is the license compatible?
- Are there transactional or stateless drivers available?
  - Is the license compatible?
  - Are the APIs compatible?
  - Can they be used for other SoCs in the same family?

Summary

- SoC vendor HALs reduce the amount of custom Zephyr code
- Can have different levels of abstraction
- Successfully used by many SoCs in Zephyr
- Review the Input Process and git history
- Talk with maintainers

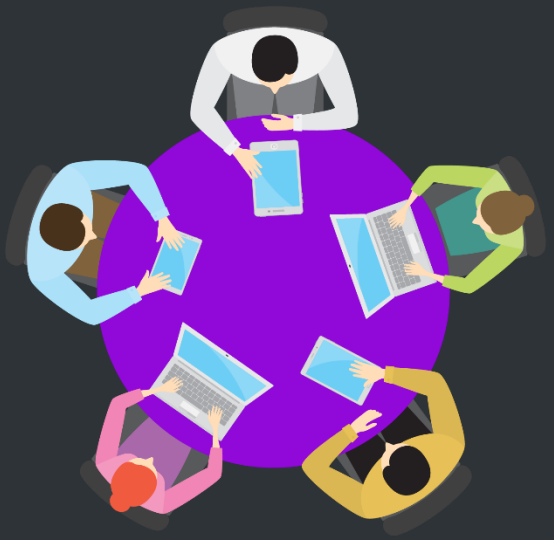
Participate!



Impact architecture

Direction

Marketing / Advocacy



Decision making

Examine the code and join!

Zephyr™

www.zephyrproject.org