THE ZEPHYR PROJECT

Johan Hedberg

Intel Open Source Technology Center

What is Zephyr?

Zephyr is a small, scalable, open-source, real-time operating system (RTOS) for use on resource-constrained systems supporting multiple architectures.

Small footprint

- Zephyr Kernel can be configured to run in as little as 8k RAM
- Enables application code to scale
- Configurable and Modular

Cross Platform

- Native support for multiple architectures:
    - x86, ARM*, ARC, NIOS-II, Tensilica, RISC-V

Open Source

- Zephyr is licensed under Apache 2 License
- Managed by the Linux Foundation*
- Transparent development

Project goal: To become the “Linux” of microcontrollers

Project Members

Platinum Members

intel

Linaro

NXP

SYNOPSYS

Silver Members

NORDIC SEMICONDUCTOR Smarter Things

runtime.io

And others…

Supported Architectures

intel®

ARM

ARC® Synopsys

RISC-V

Nios® II Processor

tensilica

Supported Boards

FRDM K64F

Arduino Due

Nucleo 103RB

NRF51

Nucleo64 L476RG

Nucleo F411RE

NRF52 pca10040

Nucleo F334R8

Arduino 101

Minnowboard

Altera MAX10

Nucleo 401RE

Hexiwear

ARM V2M MPS2

STM3210c

Atmel SAM E70

Galileo

Synopsys EMSK

NRF52

Seeed Carbon

TI Launchpad Wifi

BBC Microbit

STM32373c

Redbear BLE Nano

Quark D2000

STM32 Olimexino

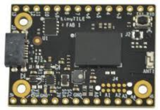
STM Mini A15

Seeed Nitrogen

ARM V2M Beetle

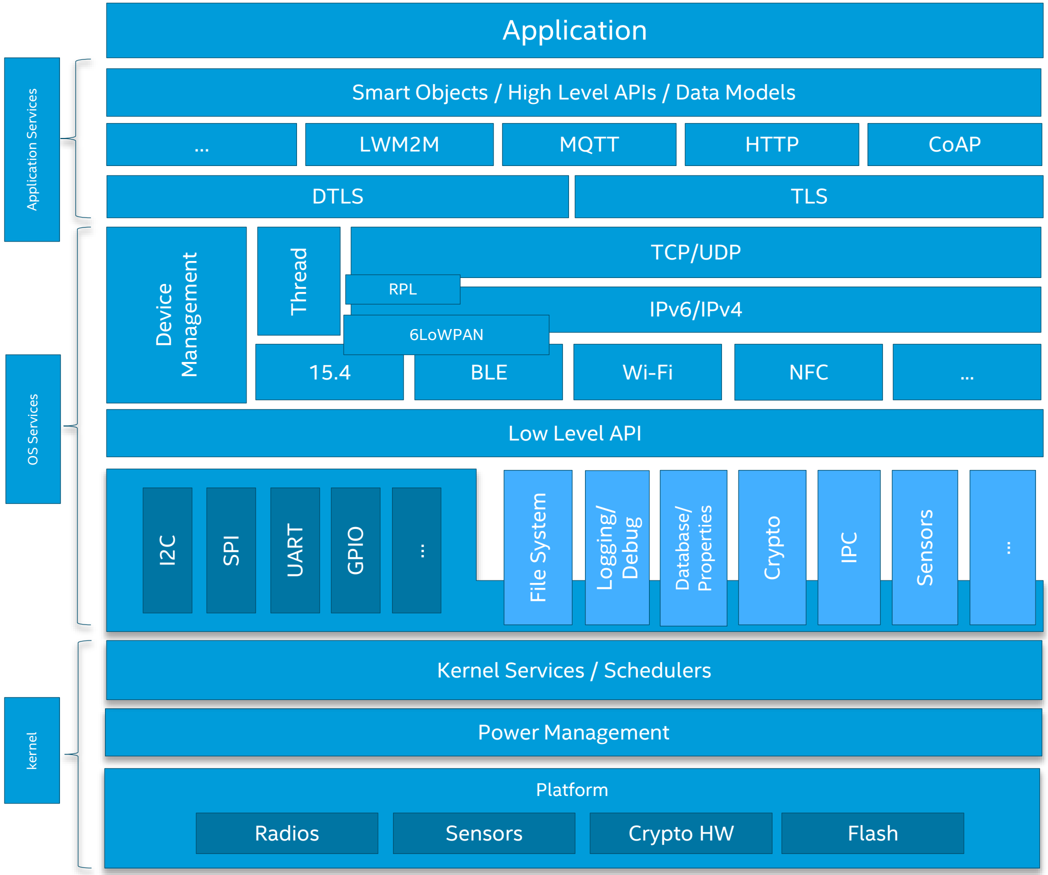
Zedboard Pulpino

FRDM-KW41Z


tinyTILE

Architecture

- Highly Configurable, Highly Modular
- Cooperative and Pre-emptive Threading
- Memory and Resources are typically statically allocated
- Bluetooth® Low Energy (5.0) with both controller and host, Bluetooth Mesh
- Native and optimized IP stack



Zephyr Ecosystem

Zephyr OS

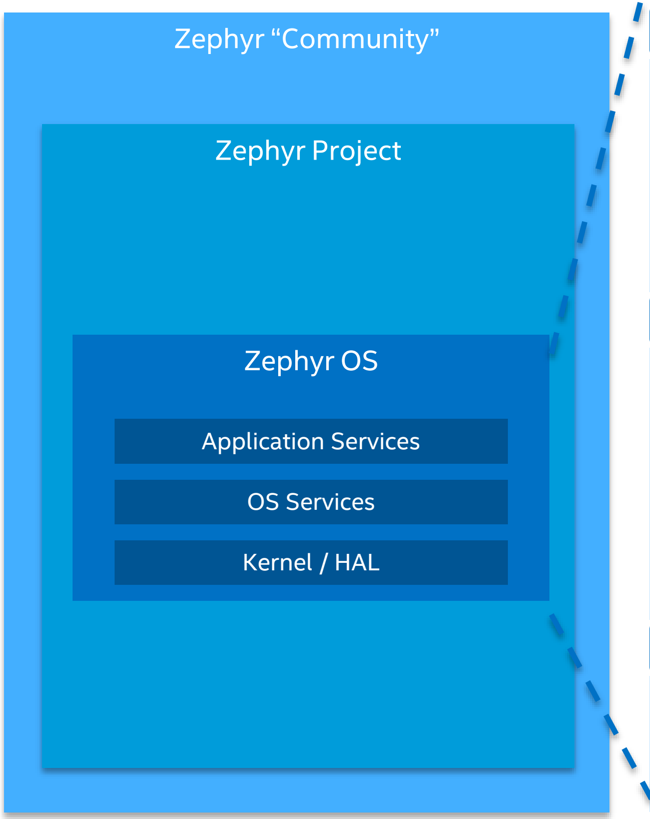
- The kernel and HAL
- OS Services such as IPC, Logging, file systems, crypto

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: Zephyr.js, Micropython, Iotivity



Zephyr “Community”

Zephyr Project

Zephyr OS

Application Services

OS Services

Kernel / HAL

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

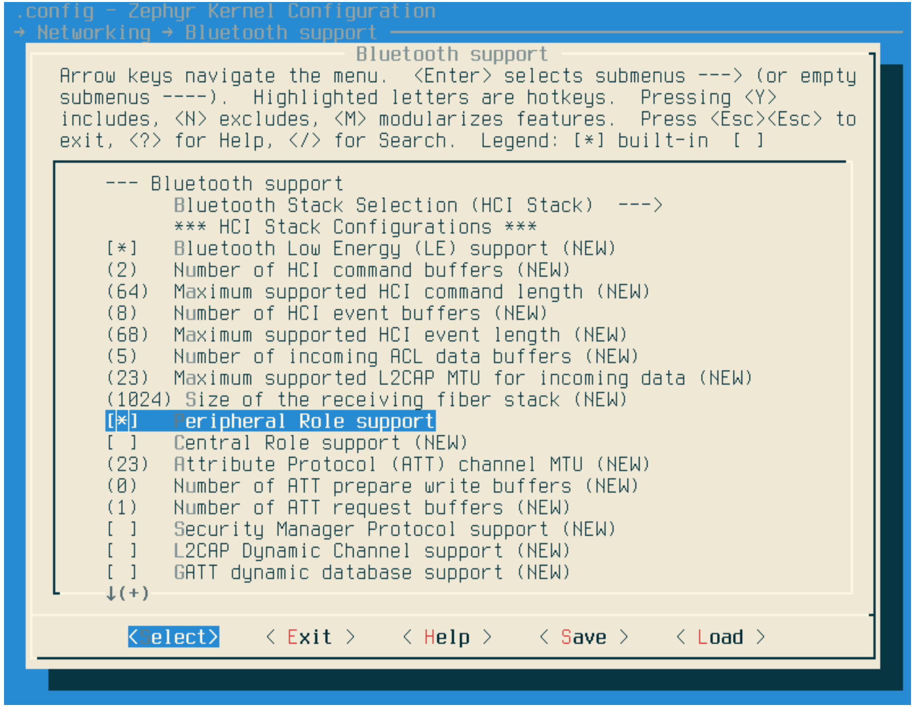
- High Level APIs
- Access to standardized data models
- High Level networking protocols

Developing with Zephyr

- Code on github, contributions through pull requests
- Linux, Mac & Windows SDKs supported
- Lots of sample applications in the source tree
- Qemu support in Linux
- Flashing boards usually just “make flash”

Familiar to Linux developers

- KConfig – based build configuration (e.g. “make menuconfig”)
- Linux coding style
- Device-tree used for board definitions
- Integrated qemu support
  - No special HW needed to get started



```
.config - Zephyr Kernel Configuration
→ Networking → Bluetooth support
                          Bluetooth support
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty
  submenus ----).  Highlighted letters are hotkeys.  Pressing <Y>
  includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to
  exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ]

      --- Bluetooth support
            Bluetooth Stack Selection (HCI Stack)  --->
            *** HCI Stack Configurations ***
      [*]   Bluetooth Low Energy (LE) support (NEW)
      (2)   Number of HCI command buffers (NEW)
      (64)  Maximum supported HCI command length (NEW)
      (8)   Number of HCI event buffers (NEW)
      (68)  Maximum supported HCI event length (NEW)
      (5)   Number of incoming ACL data buffers (NEW)
      (23)  Maximum supported L2CAP MTU for incoming data (NEW)
      (1024) Size of the receiving fiber stack (NEW)
      [*]   Peripheral Role support
      [ ]   Central Role support (NEW)
      (23)  Attribute Protocol (ATT) channel MTU (NEW)
      (0)   Number of ATT prepare write buffers (NEW)
      (1)   Number of ATT request buffers (NEW)
      [ ]   Security Manager Protocol support (NEW)
      [ ]   L2CAP Dynamic Channel support (NEW)
      [ ]   GATT dynamic database support (NEW)
      ↓(+)

        <Select>    < Exit >    < Help >    < Save >    < Load >
```

Zephyr Project Roadmap and Vision

- Expand use cases and application areas
  - Industrial, safety and security features
  - Deep Embedded usages
  - Advanced Configurations and use cases: SMP, AMP, ..
- Eco System
  - Improve support on Mac* and Windows*
  - IDE integration
  - 3rd Party Tools: Tracing, Profiling, Debugging…
  - LLVM, Commercial compilers, ..

- Introduce and support Zephyr as an E2E platform:
  - Bootloader
  - Device Firmware Updates
  - Cloud Connectivity
  - Development Tools
- Safety and Security
  - Development model and process with security and safety in mind
  - Secure and harden the Kernel
  - MISRA-C 2012 Compliance
  - Standard APIs and Portability: POSIX Layer (PSE54), BSD Socket,

Tentative Roadmap

2017				2018							
Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	June	Jul	Aug
◆ 1.9			◆ 1.10			◆ 1.11			◆ 1.12 (LTS Candidate)		

Zephyr 1.9

- POSIX API Layer (Pthread)
- BSD Socket Support
- Expand Device Tree support to more architectures
- Bluetooth Mesh
- Bluetooth 5.0 Support
- LWM2M
- MMU/MPU (Cont.): Thread Isolation, Paging (→)
- Revamp Testsuite, Increase Coverage

Zephyr 1.10

- FOTA Updates (LWM2M, BLE (→))
- Integration with MCUBOOT Bootloader
- MMU/MPU (Cont.)
- Build and Configuration System (CMake)
- LLVM Support
- NFFS File system support
- Thread Protocol
- Revamp Testsuite, Increase Coverage (Cont.)

Zephyr 1.11

- SMP Support (←)
- POSIX API Layer (PSE54)
- Eco-System:
  - Tracing,
  - Profiling
  - debugging support through 3rd party tools
- Source Code modularisation: Support external modules, boards, SoCs
- Support the kernel (scheduler + objects) as a separate module
- IDE Integration(←)

Backlog

- CanBUS, SocketCAN
- AMP
- Native Port
- MISRA-C 2012: Kernel
- Enhanced Sensor support (support HW FIFOs)

(←) potentially pull in
(→) potentially push out

SLIGHTLY DEEPER LOOK INTO ZEPHYR’S CONNECTIVITY SUBSYSTEMS

Networking Overview

- Zephyr-optimized native stack
- IPv4 & IPv6 (also simultaneously)
- UDP &TCP (also simultaneously)
- RPL & 6LoWPAN
- Protocols:
  - HTTP/WebSocket, CoAP, MQTT, DNS, mDNS, DHCPv4, LWM2M, SNTP
  - DTLS, TLS
- 802.15.4 Restricted Functionality Device support

- Drivers for
  - CC2520 (802.15.4)
  - ENC28J60 (802.3)
  - NXP FRDM K-64F (802.3)
  - NXP MCR20A (802.15.4)
  - NXP KW41Z (802.15.4)
  - BLE/6LoWPAN (IPSP node)
  - WPAN-USB/WPAN-Serial (Zephyr as a 15.4 adapter/serial radio for Linux)

Bluetooth Overview

- Bluetooth 5.0 Controller & Host
- Bluetooth Mesh
- Memory optimizations (threads & buffers) to easily fit 16k targets
- Multiple HCI drivers: UART, SPI, USB
- Basic Bluetooth Classic (BR/EDR) support
- IPSP/6LoWPAN working together with the native IP stack
- Planned features
  - Vendor HCI extensions (Mesh, custom address handling, etc)
  - Bluetooth 5.0 Advertising Extensions

Three more (short) talks on Zephyr on Saturday

- Implementing Bluetooth Mesh with Zephyr
- Zephyr Networking overview
- Building Zephyr Bluetooth Controller

Resources

Web: www.zephyrproject.org

Twitter: @ZephyrIoT

IRC: #zephyrproject @ freenode.net

Email: zephyr-devel@lists.zephyrproject.org

Code: github.com/zephyrproject-rtos

QUESTIONS?