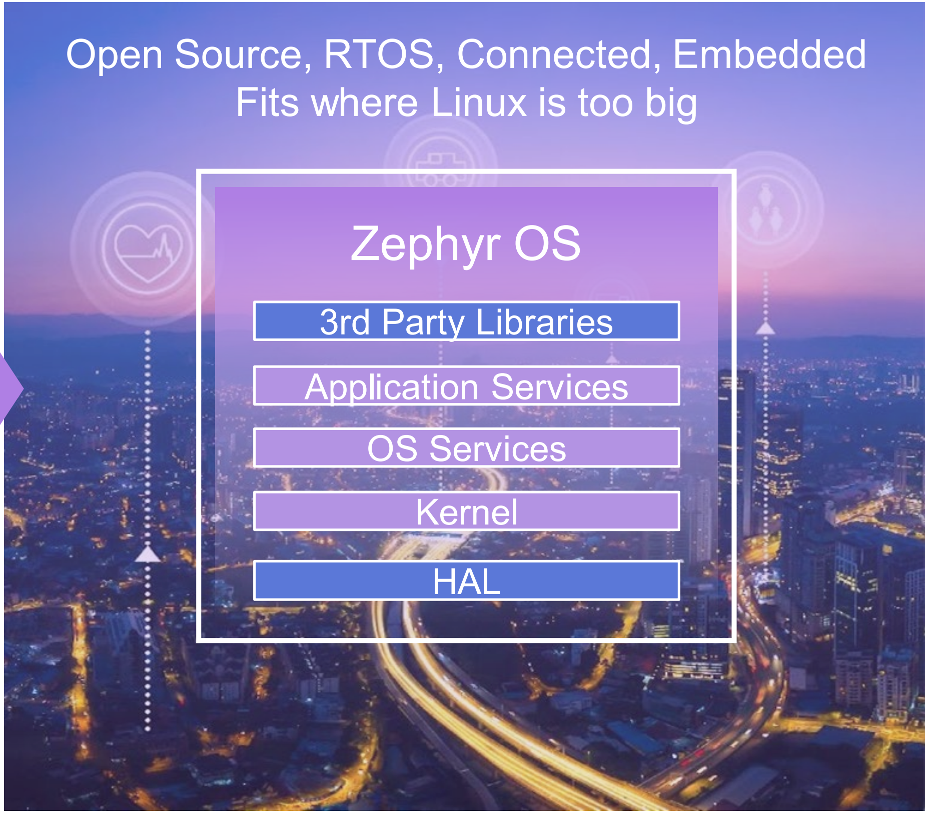
Zephyr Project:
Unlocking Innovation with an Open Source RTOS

Kate Stewart, The Linux Foundation
Carles Cufi, Nordic Semiconductor
Marti Bolivar, Nordic Semiconductor

Zephyr Project

- **Open source** real time operating system
- **Vibrant Community** participation
- Built with **safety and security** in mind
- **Cross-architecture** with broad SoC and development board support.
- **Vendor Neutral** governance
- **Permissively** licensed - Apache 2.0
- **Complete**, fully integrated, highly configurable, **modular** for **flexibility**
- **Product** development ready using LTS includes security updates
- **Certification** ready with Auditable



Open Source, RTOS, Connected, Embedded
Fits where Linux is too big

Zephyr OS

- 3rd Party Libraries
- Application Services
- OS Services
- Kernel
- HAL

THE LINUX FOUNDATION PROJECTS

Products Running Zephyr Today

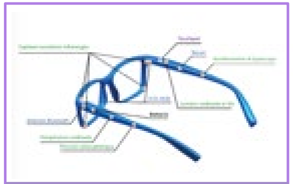
Grush Gaming Toothbrush

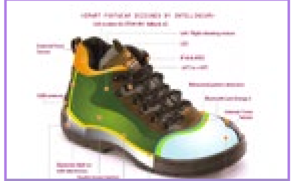
hereO Smartwatch

Proglove

Rigado IoT Gateway

Distancer


Ellcie-Healthy Smart Connected Eyewear


Intellinium Safety Shoes

GNARBOX 2.0 SSD

Adero Tracking Devices

Anicare Reindeer Tracker

Sentrius

GEPS

Point Home Alarm

RUUVI Node

HereO Core Box

Safety Pod

Zephyr Supported Hardware Architectures

ARC Synopsys

arm
Cortex-M, Cortex-R
& Cortex-A

intel®
X86 & x86_64

Nios® II Processor

RISC-V
32 & 64 bit

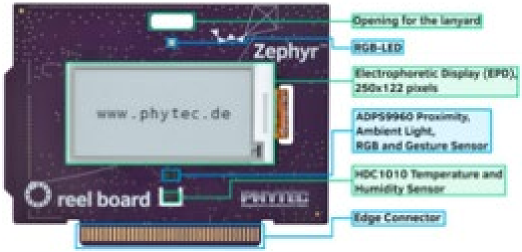
tensilica
Xtensa

Coming soon: SPARC OpenPOWER™

Development Boards Shipping with Zephyr Today

Zephyr™

Nordic Thingy91

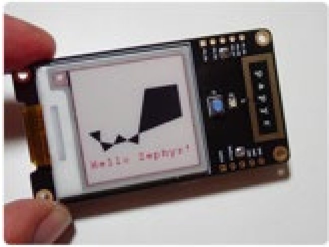

Zephyr
www.phytec.de
reel board
PHYTEC
Opening for the lanyard
RGB-LED
Electrophoretic Display (EPD), 250x122 pixels
ADPS9960 Proximity, Ambient Light, RGB and Gesture Sensor
HDC1010 Temperature and Humidity Sensor
Edge Connector

Phytec Reel Board

Icarus (Adafruit Feather) - Actinius

Antmicro Badge

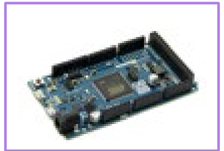
Phytec Link Board BASE



Electronuts Papyr

Board Support – 200+ and growing

Zephyr™

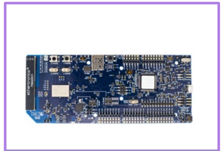

Arduino Due

Nucleo 103RB

Adafruit Feather

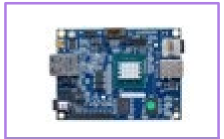
Nucleo64 L476RG

Nucleo F411RE

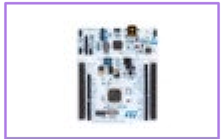

NRF91 pca10090

Nucleo F334R8

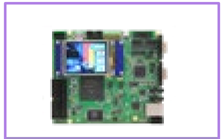
Synopsys EMSK

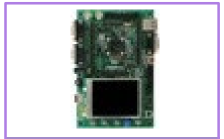

Minnowboard

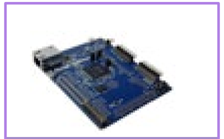
Altera MAX10

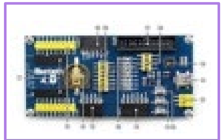

Nucleo 401RE

Vega Board


ARM V2M MPS2

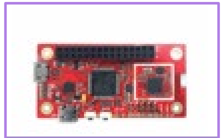

STM3210c


Atmel SAM E70

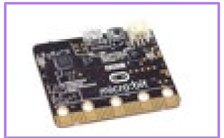

NRF51

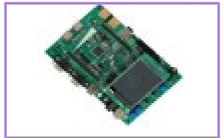
NXP FRDM K64F

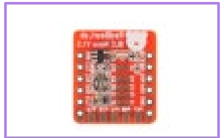
NRF52

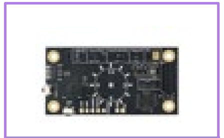

Seed Carbon

TI Launchpad Wifi

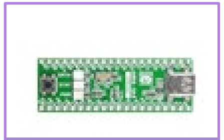

BBC Microbit

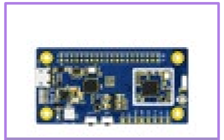

STM32373c

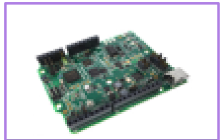

Redbear BLE Nano


96b Neon Key

STM32 Olimexino


STM Mini A15


Seeed Nitrogen


ARM V2M Beetle

Zedboard Pulpino

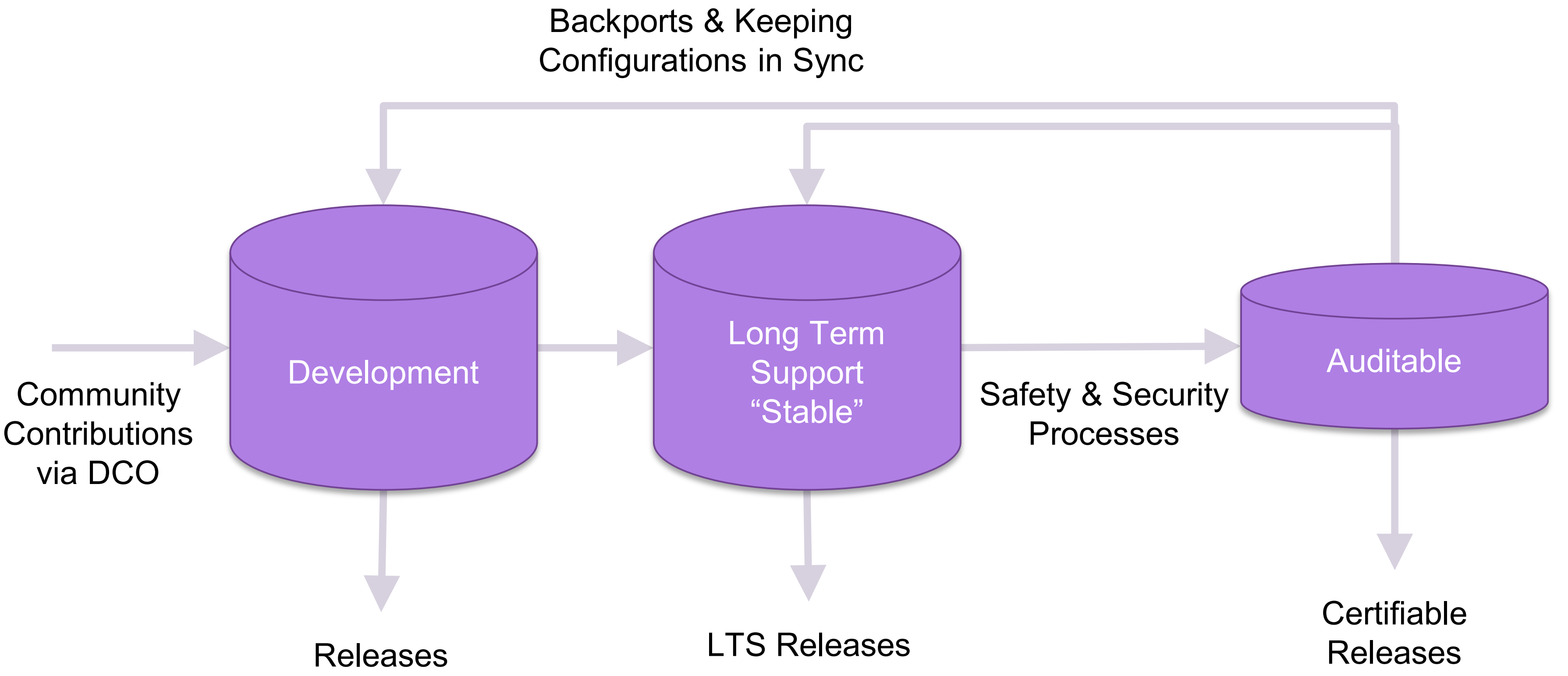
NXP FRDM-KW41Z

SiFive HiFive1

NXP i.MX RT1050

http://docs.zephyrproject.org/boards/boards.html

Code Repositories


Backports & Keeping Configurations in Sync
Community Contributions via DCO
Development
Long Term Support "Stable"
Safety & Security Processes
Auditable
Releases
LTS Releases
Certifiable Releases

Zephyr OS: Long Term Support (LTS - 1.14)

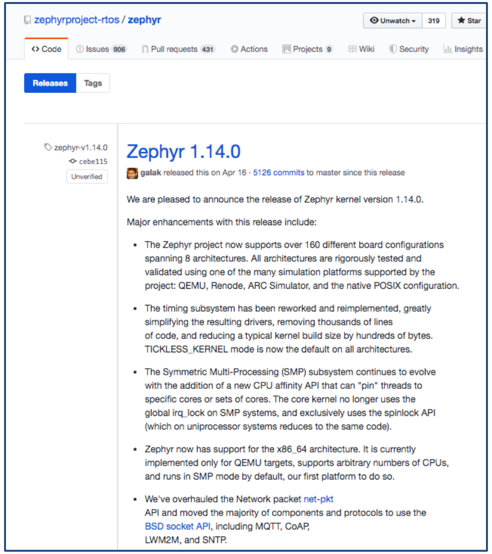
It is:

- **Product Focused**
- **Current with latest Security Updates**
- **Compatible with New Hardware**: We will make point releases throughout the development cycle to provide functional support for new hardware.
- **Tested**: Shorten the development window and extend the Beta cycle to allow for more testing and bug fixing
- **Supported** for 2 years

It is *not*:

- **A Feature-Based Release**: focus on hardening functionality of existing features, versus introducing new ones.
- **Cutting Edge**

Zephyr OS: Long Term Support (LTS - 1.14)



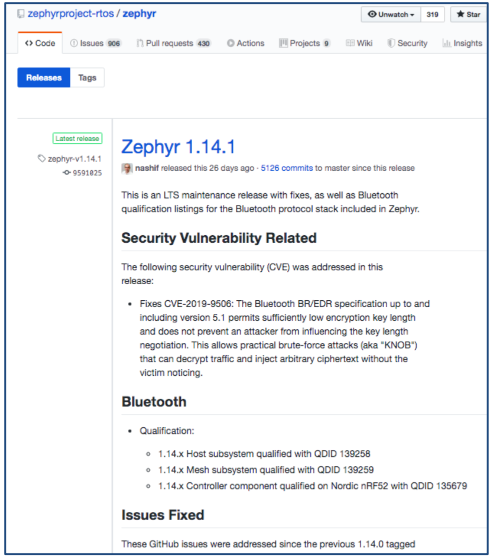
Zephyr 1.14.0

galak released this on Apr 16 · 5126 commits to master since this release

We are pleased to announce the release of Zephyr kernel version 1.14.0.

Major enhancements with this release include:

- The Zephyr project now supports over 160 different board configurations spanning 8 architectures. All architectures are rigorously tested and validated using one of the many simulation platforms supported by the project: QEMU, Renode, ARC Simulator, and the native POSIX configuration.
- The timing subsystem has been reworked and reimplemented, greatly simplifying the resulting drivers, removing thousands of lines of code, and reducing a typical kernel build size by hundreds of bytes. TICKLESS_KERNEL mode is now the default on all architectures.
- The Symmetric Multi-Processing (SMP) subsystem continues to evolve with the addition of a new CPU affinity API that can "pin" threads to specific cores or sets of cores. The core kernel no longer uses the global irq_lock on SMP systems, and exclusively uses the spinlock API (which on uniprocessor systems reduces to the same code).
- Zephyr now has support for the x86_64 architecture. It is currently implemented only for QEMU targets, supports arbitrary numbers of CPUs, and runs in SMP mode by default, our first platform to do so.
- We've overhauled the Network packet net-pkt API and moved the majority of components and protocols to use the BSD socket API, including MQTT, CoAP, LWM2M, and SNTP.



Zephyr 1.14.1

nashif released this 26 days ago · 5126 commits to master since this release

This is an LTS maintenance release with fixes, as well as Bluetooth qualification listings for the Bluetooth protocol stack included in Zephyr.

Security Vulnerability Related

The following security vulnerability (CVE) was addressed in this release:

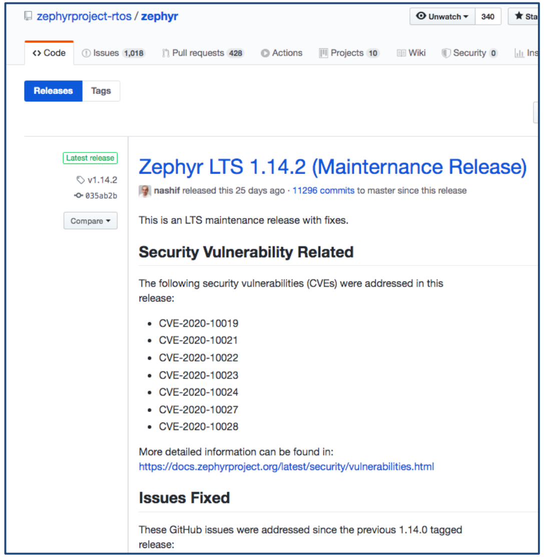
- Fixes CVE-2019-9506: The Bluetooth BR/EDR specification up to and including version 5.1 permits sufficiently low encryption key length and does not prevent an attacker from influencing the key length negotiation. This allows practical brute-force attacks (aka "KNOB") that can decrypt traffic and inject arbitrary ciphertext without the victim noticing.

Bluetooth

- Qualification:
  - 1.14.x Host subsystem qualified with QDID 139258
  - 1.14.x Mesh subsystem qualified with QDID 139259
  - 1.14.x Controller component qualified on Nordic nRF52 with QDID 135679

Issues Fixed

These GitHub issues were addressed since the previous 1.14.0 tagged



Zephyr LTS 1.14.2 (Maintenance Release)

nashif released this 25 days ago · 11296 commits to master since this release

This is an LTS maintenance release with fixes.

Security Vulnerability Related

The following security vulnerabilities (CVEs) were addressed in this release:

- CVE-2020-10019
- CVE-2020-10021
- CVE-2020-10022
- CVE-2020-10023
- CVE-2020-10024
- CVE-2020-10027
- CVE-2020-10028

More detailed information can be found in:
https://docs.zephyrproject.org/latest/security/vulnerabilities.html

Issues Fixed

These GitHub issues were addressed since the previous 1.14.0 tagged release:

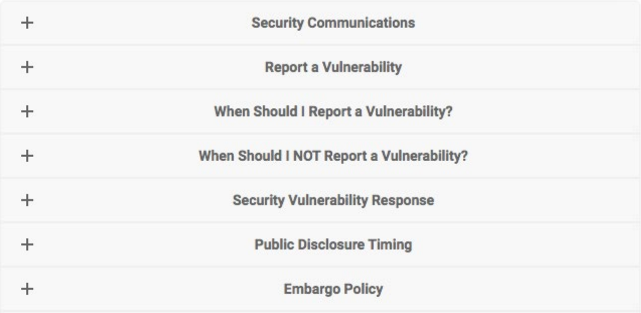
Delivering bug fixes and latest security updates!

Vulnerability Management Process

- Early in 2020 the project received a bulk vulnerability report
- Highlighted need to better document vulnerability management processes
- Added vulnerability reporting to project docs and top level web pages
- Process:
  - Embargo period
  - Stages issue goes through
  - Working with maintainers to see issues fixed
  - Public disclosure at end

Security

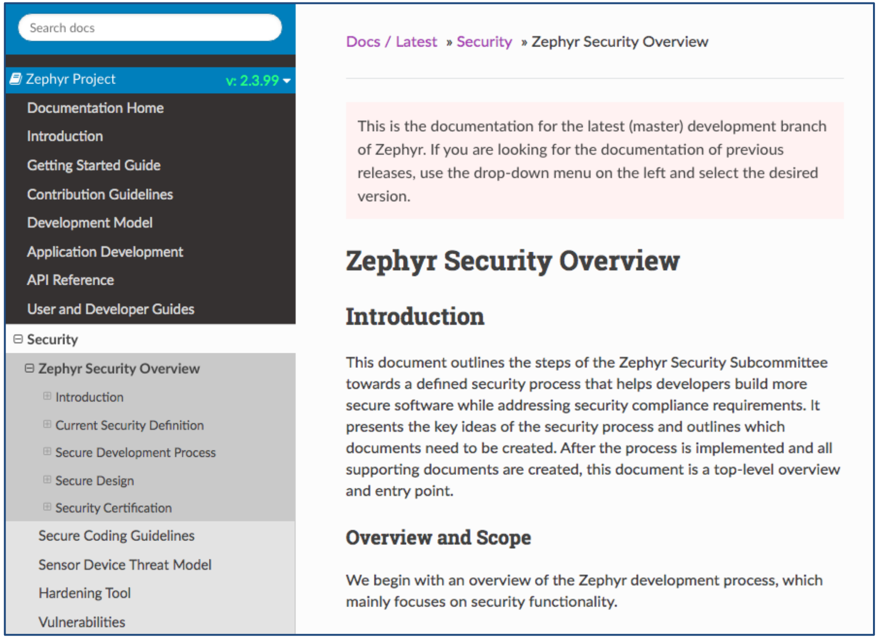
This page describes Zephyr project's security and disclosure information.



+ Security Communications
+ Report a Vulnerability
+ When Should I Report a Vulnerability?
+ When Should I NOT Report a Vulnerability?
+ Security Vulnerability Response
+ Public Disclosure Timing
+ Embargo Policy

Project Security Documentation

- Project Security Overview
- Started with documents from other projects
- Built around Secure Development, Secure Design, and Security Certification
- Ongoing process, rather than something to just be accomplished



This is the documentation for the latest (master) development branch of Zephyr. If you are looking for the documentation of previous releases, use the drop-down menu on the left and select the desired version.

Zephyr Security Overview

Introduction

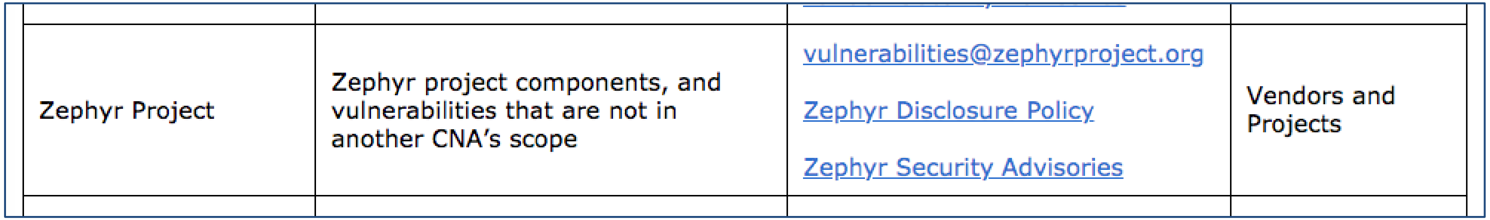
This document outlines the steps of the Zephyr Security Subcommittee towards a defined security process that helps developers build more secure software while addressing security compliance requirements. It presents the key ideas of the security process and outlines which documents need to be created. After the process is implemented and all supporting documents are created, this document is a top-level overview and entry point.

Overview and Scope

We begin with an overview of the Zephyr development process, which mainly focuses on security functionality.

CVE Numbering Authority with PSIRT

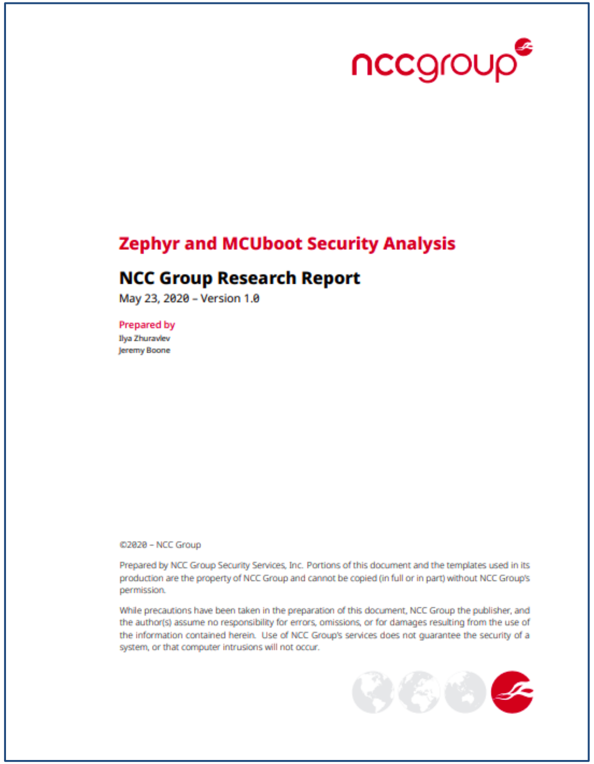
- PSIRT is Subset of Security Subcommittee
- CNA: CVE Numbering Authority
- Registered with MITRE as the numbering authority for the project. We issue our own CVEs
- Must satisfy MITRE documentation and process requirements



Zephyr Project	Zephyr project components, and vulnerabilities that are not in another CNA's scope	vulnerabilities@zephyrproject.org Zephyr Disclosure Policy Zephyr Security Advisories	Vendors and Projects

Recent Security Report

- NCC Group reported ~26 issues
- Critical, High and Medium made into JIRA tickets
- These have now been fixed
- Embargo is past, everything updated now in the vulnerability report page
- Most issues identified resulted in 1 or more CVEs being reported



nccgroup

Zephyr and MCUboot Security Analysis

NCC Group Research Report

May 23, 2020 – Version 1.0

Prepared by
Ilya Zhuravlev
Jeremy Boone

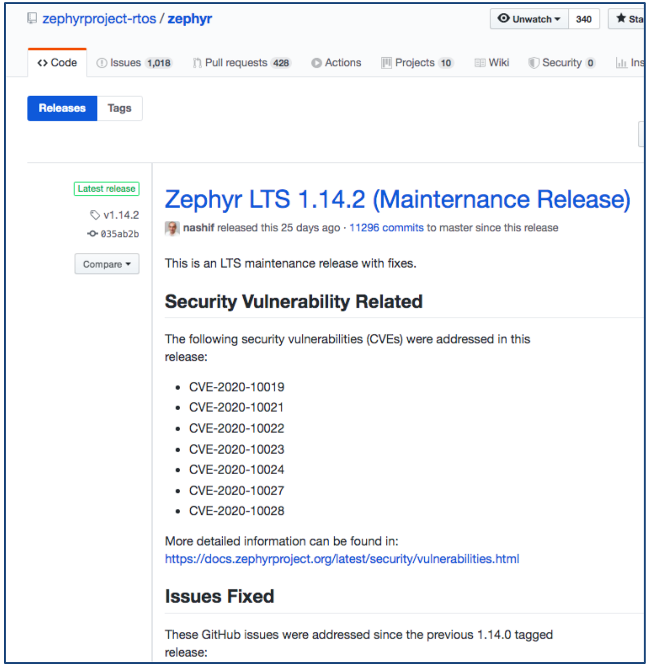
©2020 – NCC Group

Prepared by NCC Group Security Services, Inc. Portions of this document and the templates used in its production are the property of NCC Group and cannot be copied (in full or in part) without NCC Group's permission.

While precautions have been taken in the preparation of this document, NCC Group the publisher, and the author(s) assume no responsibility for errors, omissions, or for damages resulting from the use of the information contained herein. Use of NCC Group's services does not guarantee the security of a system, or that computer intrusions will not occur.

Results from the Report

- Most issues were fixed in reasonable time and included in releases
- One issue, recommendation is to disable
- Increased embargo from 60 to 90 days
  - Zephyr isn't an end product, vendors need time to incorporate fixes into products
  - Zephyr needs alert system to notify vendors
- Continue to improve process



Zephyr LTS 1.14.2 (Maintenance Release)

nashif released this 25 days ago · 11296 commits to master since this release

This is an LTS maintenance release with fixes.

Security Vulnerability Related

The following security vulnerabilities (CVEs) were addressed in this release:

- CVE-2020-10019
- CVE-2020-10021
- CVE-2020-10022
- CVE-2020-10023
- CVE-2020-10024
- CVE-2020-10027
- CVE-2020-10028

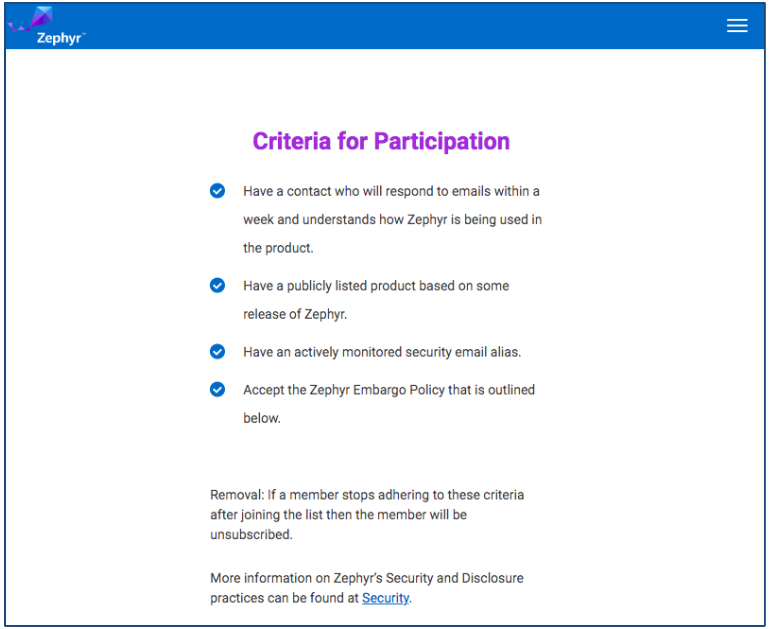
More detailed information can be found in:
https://docs.zephyrproject.org/latest/security/vulnerabilities.html

Issues Fixed

These GitHub issues were addressed since the previous 1.14.0 tagged release:

Vulnerability Alert Registry

- For Embargos to work, product makers need to be notified early so they can remediate
- Created [Vulnerability Registry]() for **vendors to register** to receive these alerts for free
- Goal: Zephyr to fix issues within 30 days and then give product makers 60 days before publication of vulnerability



Criteria for Participation

- ✓ Have a contact who will respond to emails within a week and understands how Zephyr is being used in the product.
- ✓ Have a publicly listed product based on some release of Zephyr.
- ✓ Have an actively monitored security email alias.
- ✓ Accept the Zephyr Embargo Policy that is outlined below.

Removal: If a member stops adhering to these criteria after joining the list then the member will be unsubscribed.

More information on Zephyr's Security and Disclosure practices can be found at [Security]().

Zephyr in RTOS Landscape 2020/08/24

#1

Total Contributors

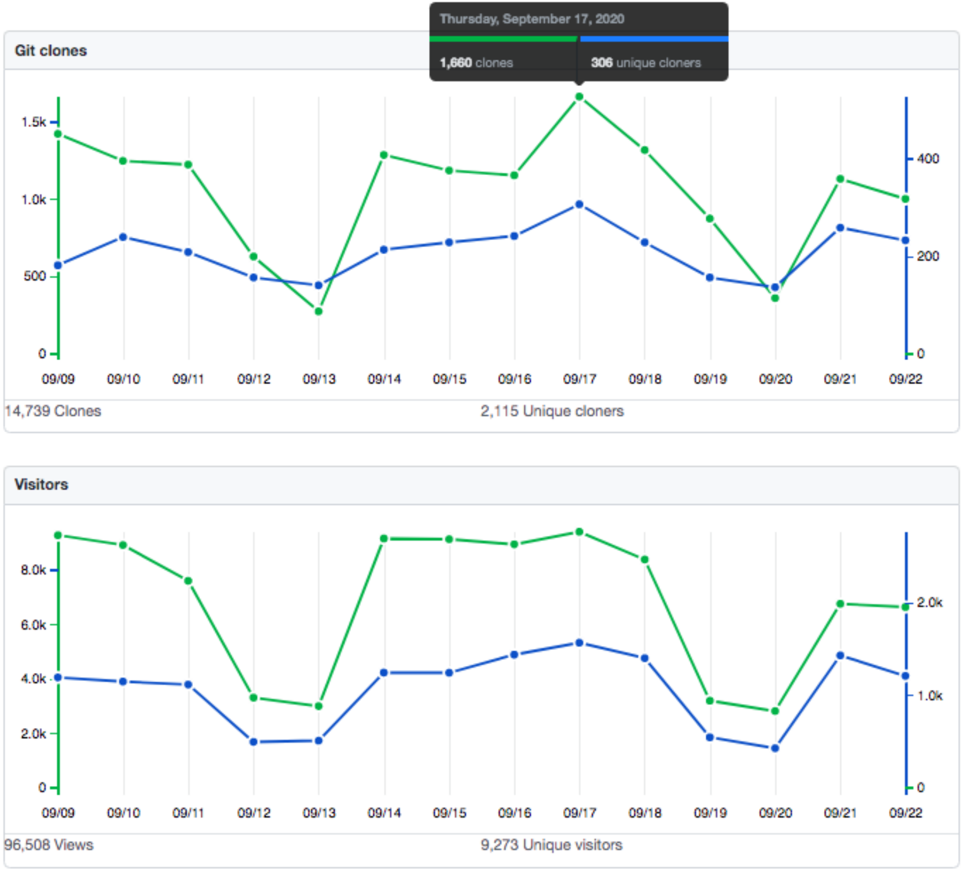
Rank	RTOS	#
1	**Zephyr**	**747**
2	mbed OS	612
3	RT-Thread	288

#1

Total Commits

Rank	RTOS	#
1	**Zephyr**	**43,312**
2	nuttX	38,031
3	RIOT	31,144

Github Web Traffic



Thursday, September 17, 2020

1,660 clones

306 unique cloners

Git clones

14,739 Clones

2,115 Unique cloners

Visitors

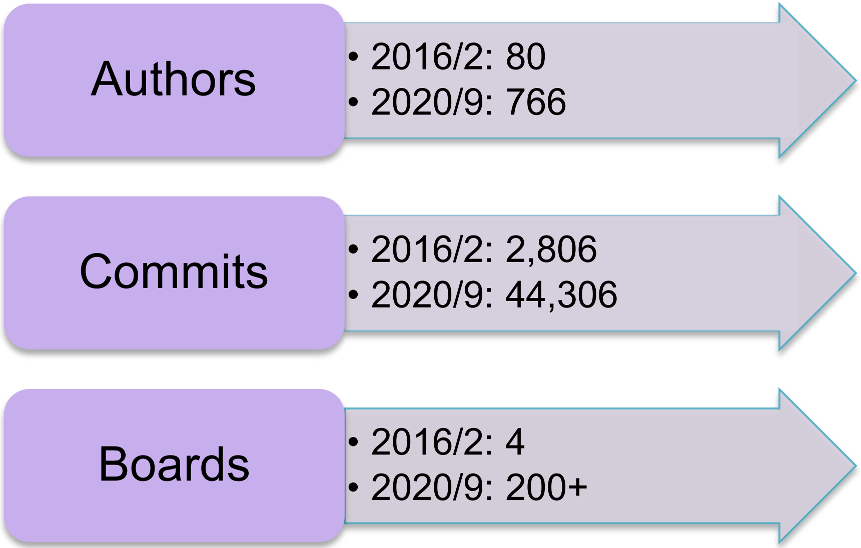
96,508 Views

9,273 Unique visitors

2 weeks of traffic to github.com/zephyr code repository as of **2020/09/22**

Growing a Diverse Community!

Lifetime project participation

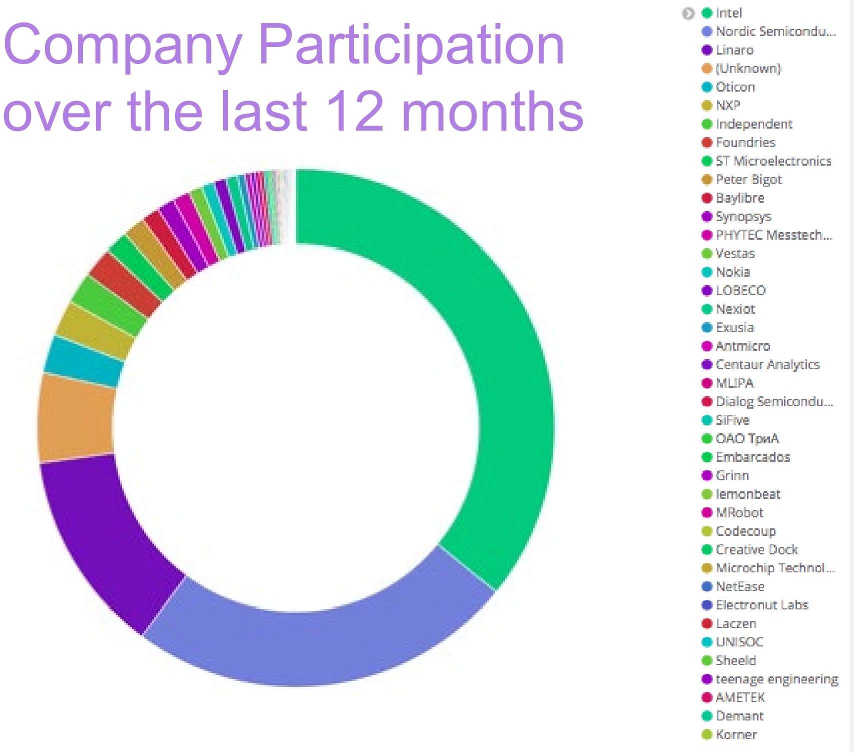


Authors
- 2016/2: 80
- 2020/9: 766

Commits
- 2016/2: 2,806
- 2020/9: 44,306

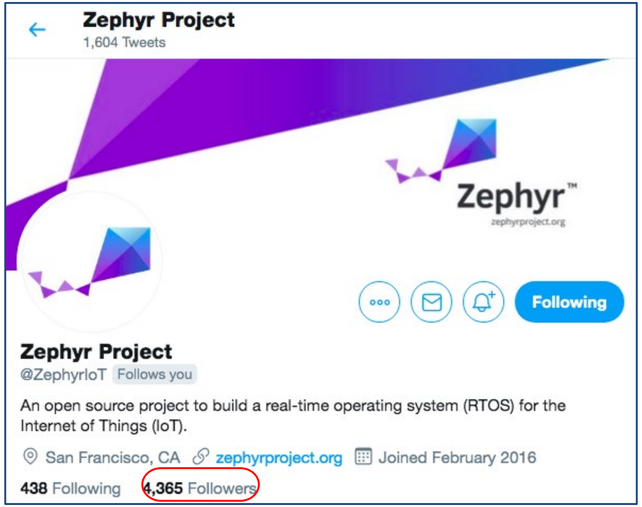
Boards
- 2016/2: 4
- 2020/9: 200+

Company Participation over the last 12 months



Vibrant, Active & Global Community

> 4300 Followers on Twitter



Zephyr Project

1,604 Tweets

Following

Zephyr Project

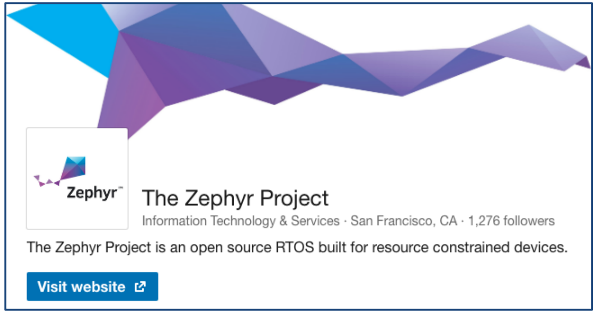
@ZephyrIoT Follows you

An open source project to build a real-time operating system (RTOS) for the Internet of Things (IoT).

San Francisco, CA zephyrproject.org Joined February 2016

438 Following 4,365 Followers

> 1200 Active on LinkedIn



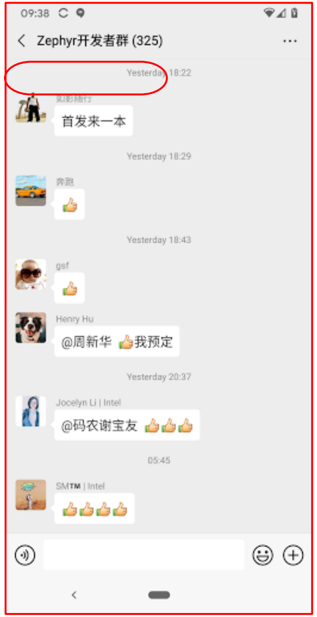
The Zephyr Project

Information Technology & Services · San Francisco, CA · 1,276 followers

The Zephyr Project is an open source RTOS built for resource constrained devices.

Visit website

> 325 Members in WeChat Group



09:38

Zephyr开发者群 (325)

Yesterday 18:22

如影随行

首发来一本

Yesterday 18:29

奔跑

👍

Yesterday 18:43

gsf

👍

Henry Hu

@周新华 👍我预定

Yesterday 20:37

Jocelyn Li | Intel

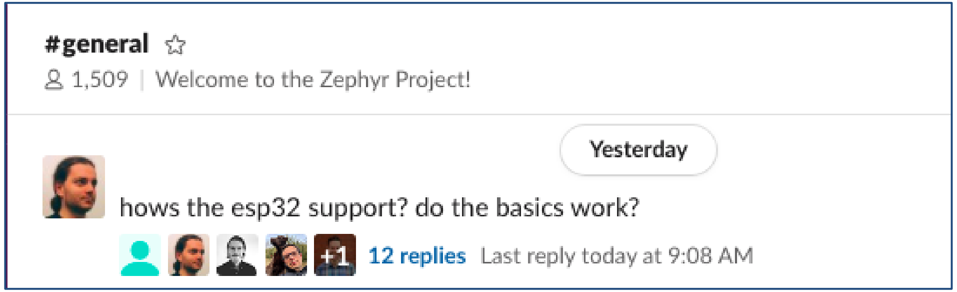
@码农谢宝友 👍👍👍

05:45

SMTM | Intel

👍👍👍👍

> 1500 Active on Slack



#general ☆

1,509 | Welcome to the Zephyr Project!

Yesterday

hows the esp32 support? do the basics work?

+1 12 replies Last reply today at 9:08 AM

Zephyr: Technology Overview

Carles Cufí, Nordic Semiconductor

Zephyr Ecosystem

Zephyr Community

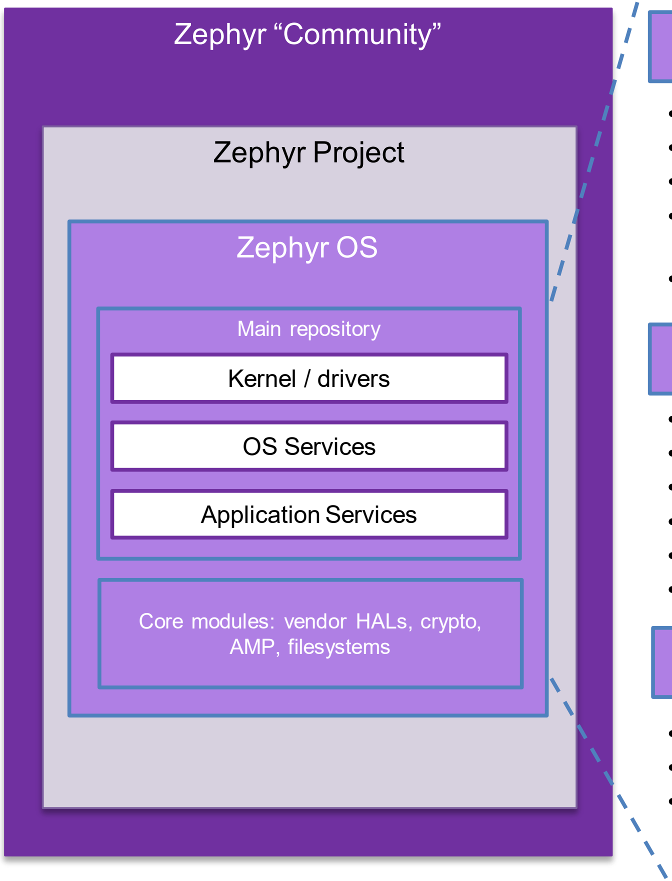
- 3rd Party modules and libraries
- Support for Zephyr in 3rd party projects, for example: micro-ROS, Tensorflow LITE, Micropython, Jerryscript

Zephyr Project

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

Zephyr OS

- Kernel and drivers
- OS services such as IPC, Logging, file systems, connectivity stacks, crypto



Kernel / drivers

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

OS Services and Low level APIs

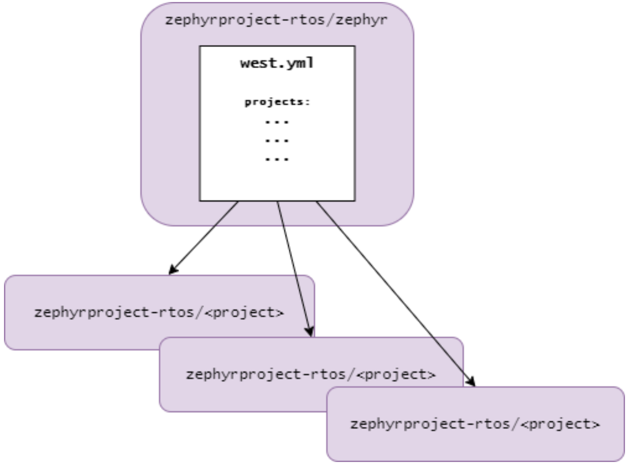
- Platform specific drivers
- Generic implementation of I/O APIs
- File systems, Logging, Debugging and IPC
- Cryptography Services
- Networking and Connectivity
- Device Management

Application Services

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

Zephyr Repositories

- Main (or manifest repository): https://github.com/zephyrproject-rtos/zephyr
    - Contains all core Zephyr code
    - All (or almost) code is Apache v2 licensed
- Externally maintained code is in separate repositories
    - Vendor HALs (CMSIS, stm32cube, mcux, nrfx, etc.)
    - Crypto (mbedTLS)
    - Filesystems (LittleFS, FATFS, etc.)
    - Libraries (OpenAMP, LoRA, LVGL, OpenThread, Canopennode, etc.)
- Clean separation of Zephyr-specific code and 3rd-party projects
- Our meta-tool, west, manages the repos

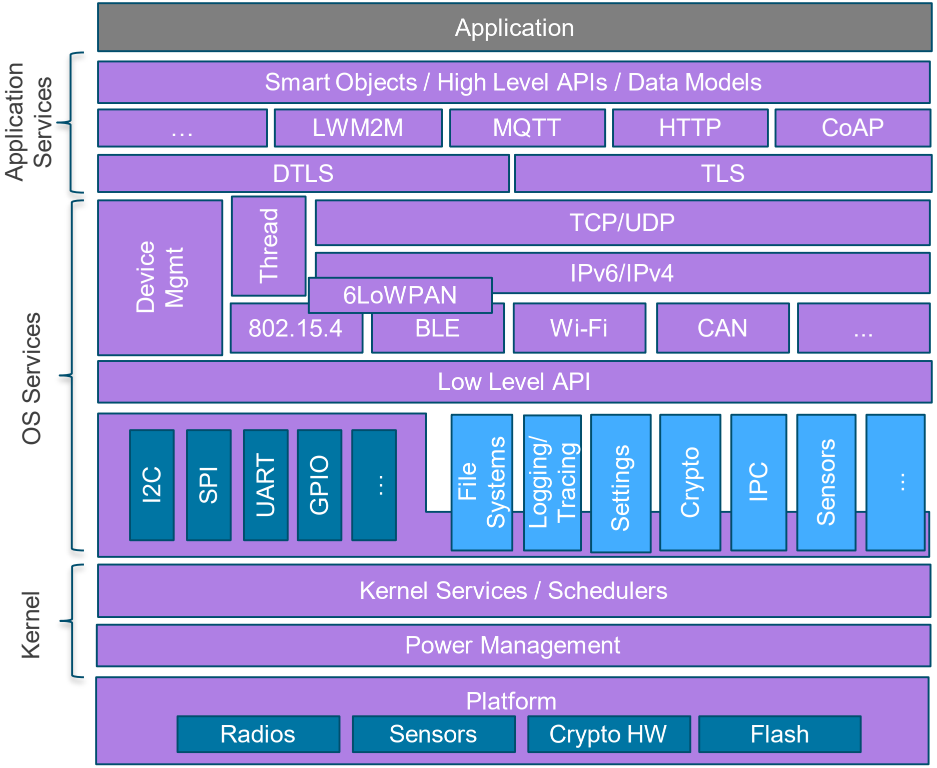

zephyrproject-rtos/zephyr
west.yml
projects:
...
...
...
zephyrproject-rtos/<project>
zephyrproject-rtos/<project>
zephyrproject-rtos/<project>

Build and configuration systems

Zephyr

- Zephyr is cross-platform and can build, flash and debug on all major Operating Systems (Windows, macOS, Linux)
- CMake
  - Industry standard build system
  - Cross-platform
  - Extensible and script-based
- Kconfig
  - Configuration system originally developed for the Linux kernel
  - Implemented in Python (so cross-platform)
- Devicetree
  - Standards-based hardware description language
  - All references resolved at build-time
  - Access from source code via a macro API

Architecture



- Highly Configurable, Highly Modular
- Cooperative and Preemptive Threading
- Memory and Resources are typically statically allocated
- Integrated device driver interface
- Memory Protection: Stack overflow protection, Kernel object and device driver permission tracking, Thread isolation
- Bluetooth® Low Energy (BLE 5.1) with both controller and host, BLE Mesh
- 802.15.4 OpenThread
- Native, fully featured and optimized networking stack

Fully featured OS allows developers to focus on the application

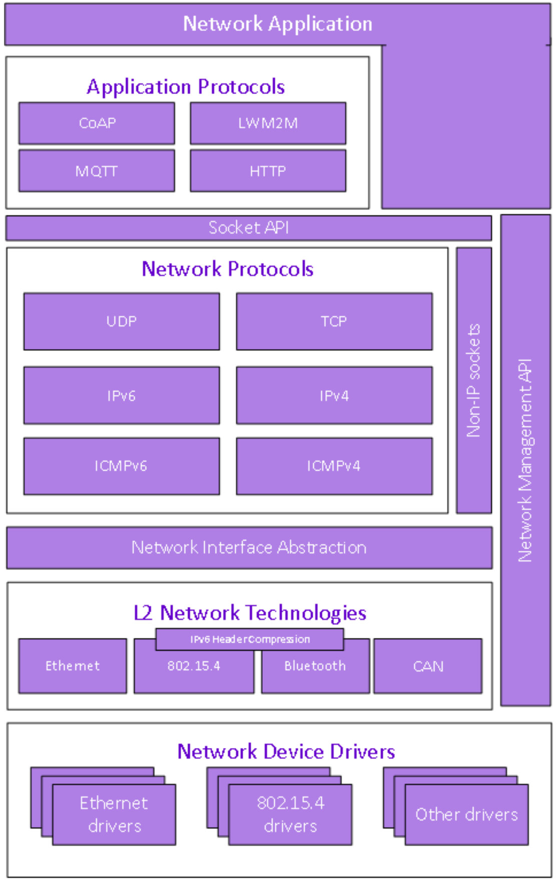
Kernel and drivers

Flexible and modern RTOS kernel

- Single core, SMP and AMP (via IPM driver) support
- Cooperative and preemptible threads
- Extensive set of synchronization and data-passing primitives
- User mode (userspace):
  - Isolates user mode threads from kernel and each other
  - Exposes a subset of kernel and driver primitives to user mode threads
  - Validates all parameters through system calls
- Efficient device driver model
  - Storage of constant driver data in ROM
  - Direct access to Devicetree nodes
  - Common APIs for all hardware implementations

Native IP Stack

- Build from scratch for Zephyr
  - Using Zephyr native kernel concepts
- Dual mode IPv4/v6 stack
  - TCP v4/v6, second generation
  - DHCP v4; IPv4 autoconf; IPv6 SLAAC; DNS; SNTP
- Multiple network interfaces support
- Time Sensitive Networking support
  - 802.1QAV API
  - 802.1AS (gPTP, generalized Precision Time Protocol)
- BSD Sockets-based API
  - TLS/DTLS supported via setsockopt call
  - RAW socket support for IP and non-IP traffic
- Supports IP offloading
  - Transparent for application using Socket API
- Compliance and security tested
  - >500 automated tests for TCP level using commercial products like IWL Maxwell Pro



Network Application

Application Protocols

CoAP

LWM2M

MQTT

HTTP

Socket API

Network Protocols

UDP

TCP

IPv6

IPv4

ICMPv6

ICMPv4

Non-IP sockets

Network Management API

Network Interface Abstraction

L2 Network Technologies

IPv6 Header Compression

Ethernet

802.15.4

Bluetooth

CAN

Network Device Drivers

Ethernet drivers

802.15.4 drivers

Other drivers

Zephyr Networking Features

High-Level Protocols

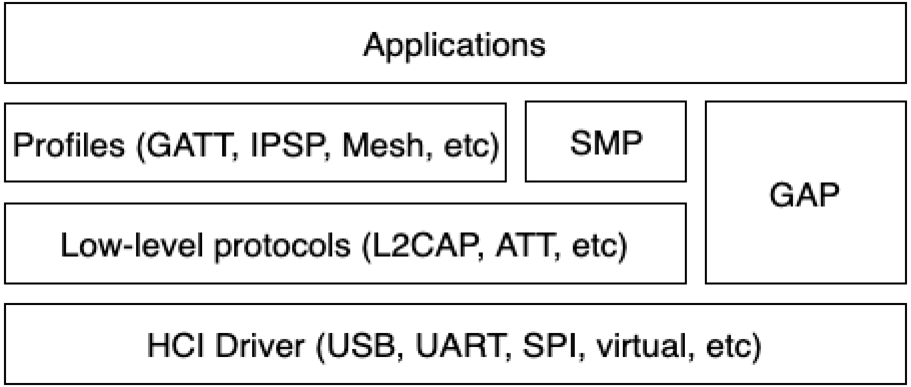
- CoAP v1
- MQTT Client v3.1.1
- HTTP
  - Native HTTP client
  - Server is implemented using CivetWEB library
  - Websocket client
- SOCKS5
- LWM2M
- Thread
  - Supported by OpenThread project

Supported technologies

- Ethernet
- Ethernet over USB
- WiFi with IP offload
- IEEE 802.15.4 with 6Lo
- Bluetooth LE with 6Lo
- CANbus with 6Lo
- PPP
- Serial modem interface

Bluetooth Host and Mesh

- Bluetooth 5.1 compliant
- Low Energy & experimental Bluetooth Classic
- Multiple HCI transports
- Qualified (as of 1.14.1) for LE and Mesh
- Can be built separately or combined with the controller
- Active community developing upcoming standards
- Mesh & GATT reference stack in Bluetooth SIG training materials



Applications

Profiles (GATT, IPSP, Mesh, etc)

SMP

GAP

Low-level protocols (L2CAP, ATT, etc)

HCI Driver (USB, UART, SPI, virtual, etc)

Bluetooth Low Energy Controller

Second-generation open source BLE software Controller:

- Bluetooth 5.1 compliant and qualified (v1.14.1)
- Split design with Upper and Lower Link Layers
- Support for multiple BLE radio hardware architectures
  - Nordic nRF5 on Arm Cortex-M (Arm v6-M, v7-M and v8-M)
  - VEGAboard on RISC-V (RV32)
  - Proprietary radio and ISA (downstream only)
- Support for both Big and Little-Endian architectures
- Asynchronous handling of procedures in the ULL
- Enhanced radio utilization (99% on continuous 100ms scan)
- Latency resilience: Approx 100uS vs 10uS, 10x improvement over 1st gen
- CPU and power usage: About 20% improvement over 1st gen
- Multiple advertiser and scanner instances

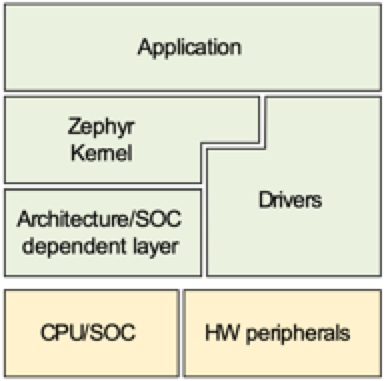
Zephyr USB Device Stack

- Supports multiple MCU families (STM32, Kinetis, nRF, SAM, …)
- USB 2.0 support
- Full and High speed support
- Supported classes:
  - CDC ACM, ECM, EEM
  - RNDIS
  - HID
  - Mass Storage
  - Bluetooth
  - Device Firmware Update
- Tight integration with the RTOS
- Flexible descriptor instancing
- Native execution support for emulated development on Linux
- WebUSB support

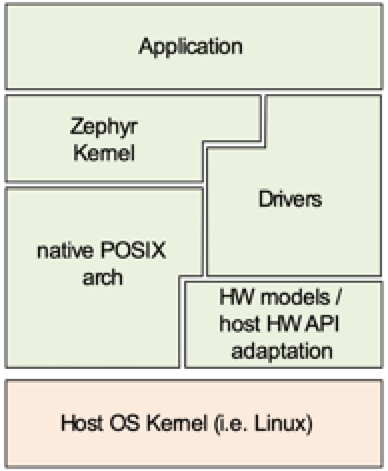
Native Execution on a POSIX-compliant OS

- Build Zephyr as native Linux application
- Enable large scale simulation of network or Bluetooth tests without involving HW
- Improve test coverage of application layers
- Use any native tools available for debugging and profiling
- Develop GUI applications entirely on the desktop
- Optionally connect to real devices with TCP/IP, Bluetooth, and CAN
- Reduce requirements for HW test platforms during development

Normal Zephyr layering



Native build Zephyr layering

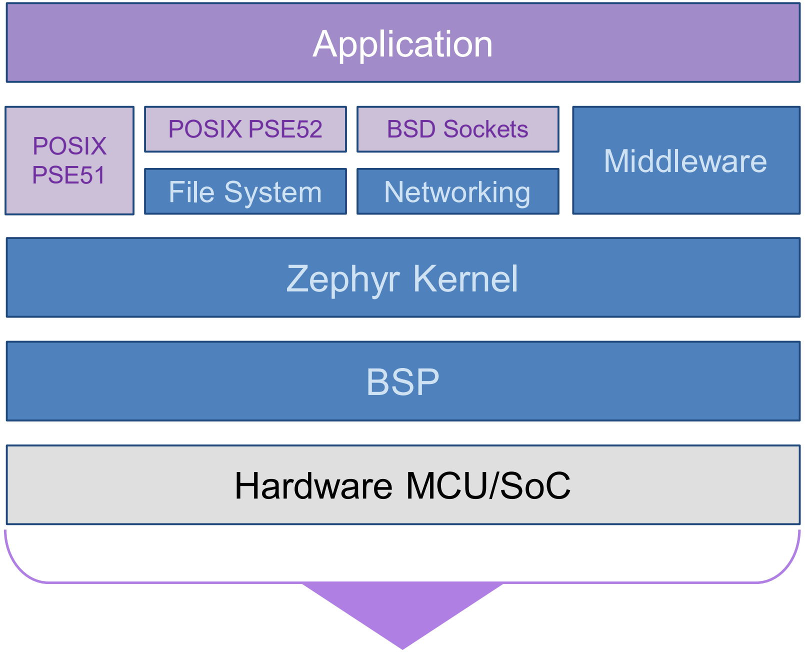

Application
Zephyr Kernel
Drivers
native POSIX arch
HW models / host HW API adaptation
Host OS Kernel (i.e. Linux)

POSIX API on Zephyr

Provides familiar API to non-embedded programmers, especially to Linux developers

Enable re-use (portability) of existing libraries based on POSIX APIs

- Provides efficient subset appropriate for small (MCU) embedded systems
- POSIX API subset is increasingly popular operating system abstraction layer (OSAL) for IoT
- Supports subsets of PSE51, PSE52, and BSD sockets API



POSIX support in Zephyr

https://docs.zephyrproject.org/latest/guides/portability/posix.html

and much more...

- Powerful logging subsystem with multiple backends
- Fully-featured shell for interaction with the system
- Device Firmware Update support via multiple mechanisms
  - MCUMgr (UART, BLE, TCP/IP)
  - Updatehub (TCP/IP)
  - Hawkbit (HTTP)
- Display support with LVGL
- Multiple filesystems and storage mechanisms
- C++ support

Thank you for watching

Carles Cufí, Nordic Semiconductor

Building with Zephyr?

West overview

Marti Bolivar, Nordic Semiconductor

Outline: Presenter: Marti Bolivar

- Zephyr and west developer
- What we're doing today:
    - Overview of what's available in west
    - Good starting points for experimenting and getting help
- Helpful if you've already gone through the Zephyr Getting Started Guide for v2.3.0, but not required: https://docs.zephyrproject.org/2.3.0/getting_started/index.html

What is west?

Zephyr’s “meta-tool” or “swiss army knife,” used for many common development workflows.

An extensible command line tool for managing a Zephyr workspace: https://docs.zephyrproject.org/2.3.0/guides/west/index.html

Recommended but not required: https://docs.zephyrproject.org/2.3.0/guides/west/without-west.html

Developed in its own git repository: https://github.com/zephyrproject-rtos/west

West “proper” vs. Zephyr uses

West “proper”

- Workspace commands (init , update , etc.)
- Configuration command (config)
- APIs for adding extension commands (external plug-ins)

In the west repository

Zephyr’s west usage and extensions

- Zephyr module integration
- Extension commands (build , flash , debug , etc.)

In the zephyr repository

Creating a workspace: west init

```
$ west init -m https://github.com/zephyrproject-rtos/zephyr \
            --mr v2.3.0 zephyrproject
```

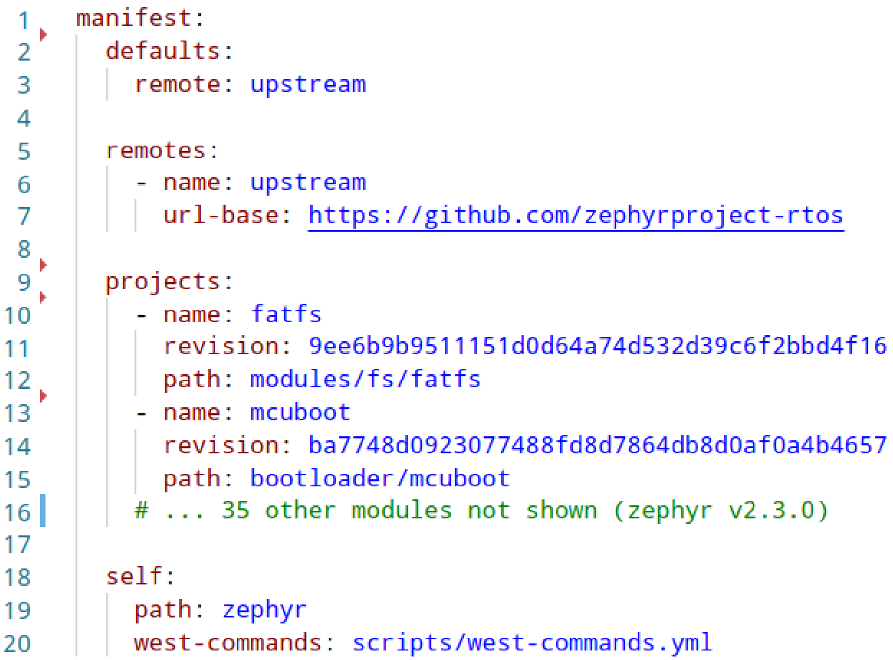
Results (simplified)

```
zephyrproject
├── .west
│   └── config
└── zephyr
    └── west.yml
```

Vocabulary

- zephyrproject: the **west workspace's** top level directory, or **topdir**. The `.west` directory marks the topdir.
- zephyr: the **manifest repository** (in this example)
- .west/config: the workspace **local configuration file**; tells west that "zephyr" is the manifest repository
- west.yml: the **manifest file**; says what other git repositories should be pulled in via `west update`

Manifest file, west.yml

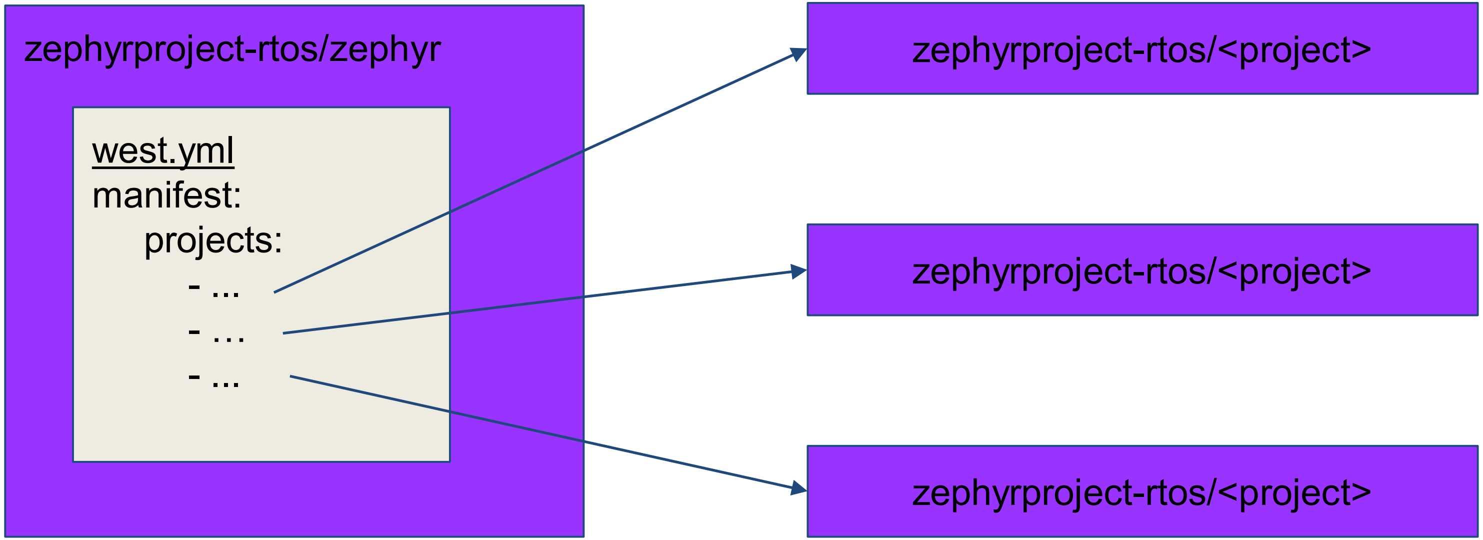


```yaml
manifest:
  defaults:
    remote: upstream

  remotes:
    - name: upstream
      url-base: https://github.com/zephyrproject-rtos

  projects:
    - name: fatfs
      revision: 9ee6b9b9511151d0d64a74d532d39c6f2bbd4f16
      path: modules/fs/fatfs
    - name: mcuboot
      revision: ba7748d0923077488fd8d7864db8d0af0a4b4657
      path: bootloader/mcuboot
    # ... 35 other modules not shown (zephyr v2.3.0)

  self:
    path: zephyr
    west-commands: scripts/west-commands.yml
```

Simplified contents of zephyr/west.yml in v2.3.0.

- remotes : where projects can be fetched
- projects : a list of git repositories in the workspace
- self : configures the manifest repository (zephyr) itself

Manifest file semantics


zephyrproject-rtos/zephyr
west.yml
manifest:
projects:
- ...
- …
- ...
zephyrproject-rtos/<project>
zephyrproject-rtos/<project>
zephyrproject-rtos/<project>

Every element of the projects list is another Git repository that is included in the upstream Zephyr project distribution.

Updating a workspace: west update

```
$ west update        # run inside the “zephyrproject” workspace
```

Results (simplified)

```
zephyrproject
├── bootloader
│   └── mcuboot
├── modules
│   └── fs
│       └── fatfs
├── .west
│   └── config
└── zephyr
        └── west.yml
```

Vocabulary

- **Modules**: third-party code with Zephyr integration, can be integrated into Zephyr applications, device drivers, etc.
- Includes things like file systems, vendor HALs, etc.
- A small number west projects live outside the modules directory. Currently just the mcuboot bootloader and some additional developer tools repositories

Other workspace commands

Some additional workspace management commands:

list	print information about projects in the west manifest
manifest	slice and dice the west manifest
diff	"git diff" for one or more projects
status	"git status" for one or more projects
forall	run a command in one or more local projects

For a complete list of commands, including extensions, run:

```
$  west help
```

Zephyr extension commands

Some additional Zephyr development extension commands:

build	compile zephyr applications
flash	flash a compiled application to a board
debug	flash an application and enter a debugger (usually GDB)
debugserver	flash an application and start a debug server
attach	attach a debugger to a board without flashing

Most common extension issue

If you see this when running an extension like west build :

```
west: error: argument <command>: invalid choice: 'build'
(...)
```

Then you’re not in your workspace, so the extension can’t be found.

See this item on the troubleshooting page: https://docs.zephyrproject.org/2.3.0/guides/west/troubleshooting.html#invalid-choice-build-or-flash-etc

West configuration files

- “INI-like” (Python configparser syntax)
- We saw .west/config earlier; that’s the workspace configuration file
- You can also set user- and system-wide configuration values
- Run west config to manage your configuration
- See also: https://docs.zephyrproject.org/2.3.0/guides/west/config.html

Help and troubleshooting

Documentation: West page in the Zephyr guides and Troubleshooting west. Also:

$ west help

List all commands and one line help for each (including extensions).

$ west <command> help

Help for a specific <command>, like west help init

$ west - v <command>

Enable verbose output for <command>, like west - v init www.zephyrproject.org

Q&A