Zephyr™ Project
What is Zephyr™ Project?

**Small Footprint RTOS for IoT**
- As small as 8KB
- Enables application code to scale
- Configurable
- Modular

**Truly Open Source**
- Hosted by Linux Foundation
- Transparent development
- Apache 2.0 License

**Cross Architecture**
- ARM
- x86
- ARC
- RISC-V
- Tensilica
- Nios II
Why Does the Industry Need RTOS Consolidation?

- High NRE hindering Mass adoption
- Fragmentation with large number of choices
- No single RTOS for IoT use cases x-platforms
- OEMs and Devs need a solution they can influence
- Limited options that include security for connected, constrained devices
  Collaborate on security with others
- Roll your own demands high level of maintenance
Zephyr Project Membership:

Platinum:
- Linaro
- NXP
- Intel
- Synopsys

Silver:
- runtime.io
- Nordic Semiconductor
Zephyr Architectures
Zephyr Community: 1 Year later...

Zephyr Launch
Feb 17, 2016
5 Repositories ➔ 10
80 Authors ➔ 207
5,806 Commits ➔ 14,239
4 Boards ➔ 57
gerrit ➔ github
Goal: Separate business decisions from meritocracy, technical decisions

<table>
<thead>
<tr>
<th>Governing Board</th>
<th>Technical Steering Committee</th>
<th>Community</th>
</tr>
</thead>
</table>
| • Decides project goals  
  • Sets business, marketing and legal decisions  
  • Prioritizes investments and oversees budget  
  • Oversees marketing such as PR/AR, branding, others  
  • Identifies member requirements | • Serves as the highest technical decision body consisting of project maintainers and voting members  
  • Sets technical direction for the project  
  • Coordinates X-community collaboration  
  • Sets up new projects  
  • Coordinates releases  
  • Enforces development processes  
  • Moderates working groups  
  • Oversees relationships with other relevant projects | • Code base open to all contributors, need not be a member to contribute.  
  • Path to committer and maintainer status through peer assessed merit of contributions and code reviews  
  • Ecosystem enablement |

Zephyr Project Governance

- Financial & Policy Oversight
- SW Certification Oversight
- Marketing Oversight
- Kernel & Subsystem Maintainers
- Security Maintainer
- Developer Tools Maintainers
- Architecture Maintainers
- Individual Contributors
- Member Organizations
- Supporting Organizations
- Others
Security and Global IoT
“… to maintain and address all security concerns in the sector, both software and hardware security chips should be used.” – Technavio, January 2017

Global internet of things security market is expected to grow at a CAGR of nearly 48% during the period 2017-2021 – Technavio, January 2017

Zephyr Project & Securing IoT
● Focus on addressing security needs of connected, resource constrained devices
● Work group focused on defining the safety & security strategy and development plans
● Membership marries HW & SW security expertise and investment through open source development
● The goal of working group to develop a safety & security auditable version of the OS
Zephyr OS: Auditable Code Base

- Initial and subsequent certification targets to be decided by Governing Board.
- An auditable code base will be established from a subset of Zephyr OS.
  - Code bases will be kept in sync from that point forward.
  - More rigorous processes (necessary for certification) will be applied before new features move into the auditable code base.
- Processes to achieve selected certification to be determined by Security Working Group and coordinated with Technical Steering Committee.
Code Repositories

- Development
  - Community Contributions via DCO
  - Releases

- Long Term Support "Stable"
  - Safety & Security Processes
  - Products

- Auditable
  - Products ready to be Certified
  - Audit Ready Documentation (Platinum Members Only)

Backports and keeping configurations in sync
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 Drivers

Zephyr Community
- Bootloader
- 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
Architecture

- Highly Configurable, Highly Modular
- Cooperative and Pre-emptive Threading
- Memory and Resources are typically statically allocated
- Cross architecture (IA32, ARM*, ARC, NIOS-II, RISC V, Xtensa, others WIP and under discussion)
Zephyr Roadmap

<table>
<thead>
<tr>
<th>Zephyr Releases</th>
<th>2016</th>
<th>2017</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Aug</td>
<td>Sept</td>
</tr>
<tr>
<td>Zephyr 1.5</td>
<td>♦</td>
<td>♦</td>
</tr>
<tr>
<td>Zephyr 1.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zephyr 1.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zephyr 1.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt; Future &gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Zephyr 1.6**
- QMSI 1.3
- Unified Kernel
  - New Kernel APIs with shims for old Kernel APIs
- HW Crypto Offload
- Native CoAP
- Shell Improvements
- USB Mass Storage
- Power Management Enhancements for Quark SE
- Nordic BLE Controller

**Zephyr 1.7**
- Unified Kernel continued
- Direct Interrupt handling
- CMSIS support
- Native IP Stack
  - New Net and L2 APIs
  - Native protocols: MQTT, HTTP Library, CoAP (Zoap), NATS
- IoTivity/OCF
- Device Tree
- Bootloader Support
- Openocd support
- RISC V Port
- Xtensa Port

**Zephyr 1.8**
- Transition to github
- Infrastructure improvements
- Tick-less Kernel
- BT 5.0 Features
- Eco System: Tracing, debugging support through 3rd party tools
- Memory Management
  - Memory Protection (MPU)
  - Thread Isolation
  - Paging
- Expand device support

**< Future >**
- Improved Build and Debug
  - 3rd Party Compilers Support
  - Build on Mac/Windows
- Asymmetric Multi Processing (AMP) - OpenAMP
- Initial Thread Stack components
- Expand LLVM Support
- Dynamic runtime modules
- Precision Time Protocol
- Time Sensitive Networks
- SMP Support
- CanBUS, SocketCAN
- Kernel data integrity
- BLE Mesh
- LWM2M
- FOTA Updates
- Additional architectures
- IDE Integration
Benefits of Adopting Zephyr Project

• Roll your own is expensive & difficult to develop & maintain
• Permissively licensed corresponds to ease of adoption
• Multiple corporate sponsorships assures long term commitment and longevity
• Long standing reputation of open source investment among current membership
• Community innovation has proven faster for progression
• Project development is a collaboration of industry experts
Membership has its Privileges

- Guide market segment focus priorities
- Board seats are limited to members only
- Board sets priority for audit and certification activities
- Platinum members have access to reference audits and certification artifacts
Participate!

Examine the code, contribute and join us!

Impact architecture
Direction
Marketing / Advocacy
Decision making
Backup - Technology Information
Zephyr Project

- **SDK**
  - Toolchains
  - Host tools
- Tools and development environment
- Testing:
  - Automated Hardware Testing
  - Test harnesses
- Documentation
- Management Console and Bootloader
- Binding and Frameworks, i.e:
  - Zephyr.js
  - Python bindings
  - Iotivity
- Cloud Connectivity and Gateways
Advanced Usages

AMP

SMP
IP stack in Zephyr 1.7

- Dual mode native stack.
- UDP/TCP
- Elements of RPL, 6LoWPAN
- Protocols:
  - HTTP, CoAP, MQTT, DNS, DHCPv4, DTLS
- 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)
  - BLE/6LoWPAN (IPSP node)
  - WPAN-USB/WPAN-Serial (Zephyr as a 15.4 adapter/serial radio for Linux)
IP stack: Next (1.8+)

- Thread enablers (MLE, IP-IP tunneling, 6LoWPAN and IP routing etc.)
- VLAN support for Ethernet, NTP/SNTP
- Samples:
  - Offload, mesh
- Optimization and Profiling
- Testing:
  - IPv6 ready logo,
  - TAHI full coverage
- Thread 1.2
- 15.4 IOP testing
- ...
Bluetooth support in Zephyr 1.7

- Combined Controller + Host support for nRF5 boards
  - 1.6 was limited to Controller-only on nRF51
- Memory optimizations (threads & buffers) to easily fit 16k targets
- New SPI HCI driver for ST BlueNRG module
  - Carbon board will try to reuse this
- Improved BR/EDR (Bluetooth Classic) support
- Fix UART HCI driver to never drop packets by taking advantage of HW flow control
- Latency reductions in Controller code
- IPSP/6LoWPAN ported & working on the new IP stack
Bluetooth: Next (1.8+)

- Bluetooth 5.0 Controller & Host features
- Bluetooth Mesh
  - As soon as the specification gets adopted
  - Collaboration is open to companies with existing spec access
- Vendor HCI commands & events
  - Mesh optimizations
  - Various things that standard HCI doesn’t provide, like default static address reading
- Further memory consumption reduction with the help of k_poll()
- Completion of basic BR/EDR protocols (RFCOMM, L2CAP & SDP)
Integrated Security

- Standardized building block and robust communication stacks
- Cryptographic library based on TinyCrypt2
- TLS/DTLS: Provided by mbedTLS
- Now: Static and single binary applications, Single address space, No loadable modules
- Planned security features:
  - Device Management and Updates
  - APIs to support vendor specific Crypto implementations (software/hardware)
  - Secure Key Storage