

# 面向物联网时代的新一代嵌入式软件平台 Zephyr

任 慰, Synopsys 武汉研发中心

## 起源

相对于其他常见的嵌入式软件平台, Zephyr 的历史并不长, 最初起源于风河公司 (Windriver) 面向物联网设备的操作系统 Rocket。2016 年 2 月由 Intel, Synopsys, NXP 等几家公司发起, 创立了面向物联网时代的新一代嵌入式软件平台 Zephyr, 并由 Linux 基金会管理。Zephyr 的中文意思为“微风”, 从名字可以看出, Zephyr 并不算成为一个像通用 OS 如 Linux 那样的重型平台, 而是主打轻量级。Zephyr 旨在成为物联网时代资源有限的中小型设备的最好的开源软件平台, 同时十分强调安全设计。Zephyr 和 Linux 同属 Linux 基金会, 可以有效地和 Linux 形成互补, 可以用在因为 Linux 过大过重而不适合使用的场合, 例如微控制器。一个比较形象的说法是, Zephyr 希望成为一个针对微控制器的 Linux, 而且熟悉 Linux 内核开发流程的开发者可以方便地上手 Zephyr 开发。

Zephyr 的几家创立公司在芯片领域、嵌入式领域都有十分雄厚的实力, 都拥有自己的比较成熟的平台, 那么为什么仍然要合作从新创立一个新的平台呢? 其主要目的在于创建一个大家都可以参与的良好生态。物联网时代的快速发展对嵌入式软件提出了更高的要求, 虽然嵌入式软件的一个特点就是碎片化, 但整体上正向复杂化、平台化的方向发展, 这是信息化、数字化不断向底层、向终端延伸的结果, 其趋势可以从第一波 PC 时代和第二波移动互联网时代可以看出。嵌入式软件正从“春秋时代”迈入“战国时代”, 在不同的具体应用领域或逐渐出现对应的一个或几个平台, 一个很好的例子便是汽车电子应用中 AUTOSAR 平台。在此背景之下, 一方面由某个公司开发维护一个专属的平台, 其意义、投入产出比等都需要仔细斟酌, 特别对于“非巨头”公司; 另一方面, 一些由个人或者机构开发的简单平台如不跟上趋势, 也会逐渐消亡。

## 主要特点

在物联网发展的热潮下, 一方面涌现了许多新的嵌入式软件平台, 如 ARM Mbed、Alios Things、Huawei LiteOS 等, 另一方面很多传统的平台在获得投资或被收购后也在不断强化和升级以应对物联网时代的要求, 如 Amazon FreeRTOS、RT-Thread、uCrium OS 等。在强手如林的情况下, Zephyr 有其独到之处, 主要可以归结为以下几点:

## 开源

开源是形成良好生态的前提, Zephyr 的代码、文档和工具等绝大部分资料都开源托管在 github 上。虽然很多平台都选择了开放源代码, 但 Zephyr 的开源和开发体现在细节之上。从 Zephyr 的代码仓库(repo)中, 可以看到详尽的开发活动记录, 包括代码提交、bug/issue 列表、讨论记录、测试记录与结果、发布计划、路线图等等。Zephyr 中目前没有直接开源的

部分很少，只限于最新的蓝牙协议栈实现、安全相关的内容等。除 github 上相关内容，用户还可以从 zephyr 的官方网站、邮件列表，社交平台（slack）等多个途径获得有效的信息。

## 活跃的社区

Zephyr 成立至今发展迅速，不断有新的机构加入成为会员，不断有新的开发者和用户使用并贡献代码，已经形成了一个相当活跃的社区。截至 2019 年 10 月 24 日，根据 github 上的统计数据，Zephyr 的有效代码提交者为 559 名，在所有 GitHub 上类似平台中排名第一，总共代码提交数为 34571 个，只名列老牌平台 nuttx 之后（估计很快就会超越）。而这个两个数据中，相当大一部分来自 Zephyr 会员机构的专业开发人员，极大地保证了 Zephyr 的质量。同时对 zephyr 官方网站的访问、开发者在线会议和邮件列表及 Slack 上的讨论也是非常活跃。依托于活跃的社区，Zephyr 每三个月就会有一次正式发布，为支持产品级的开发每两年会有一次长期支持版的（LTS，Long Term Support）的发布。

## 安全导向的设计

安全作为物联网时代的热点之一，Zephyr 从设计之初就把安全融入了进来。首先 Zephyr 对安全之基的质量保证（QA，Quality Assurance）非常重视，建立了一套完善的质量保证体系，应用测试驱动开发（Test-Driven-Development）理论，为集成测试和单元测试提供了易用的测试框架和执行脚本。在功能代码之外，构建了大量的开源测试用例，并充分利用 github 的基础设施以及多种持续集成(Continuous Integration)工具构建了一个自动化的流程，涵盖了代码风格检查、代码静态检查、单元测试、集成测试和文档生成等多方面；其次，Zephyr 在运行时（runtime）上依托硬件上各种保护机制，实现了类似 Linux 的用户态/内核态隔离、堆栈溢出检测和可信执行环境（TEE）等功能，可以有效地实现任务隔离、数据保护等功能，这一点在众多平台中是少有的；第三，在信息安全(Cyber Security)方面集成了两种框架 mbedtls 和 tinycrypt(适用于无动态内存分配的应用)，可以对通信和数据进行加解密，并有密码学驱动框架方便集成各类硬件加速单元；第四，在功能安全方面（Functional Safety），Zephyr 正在紧锣密鼓地筹备通过相关认证，包括代码上应用 MISRAC 2012 标准，核心部件符合 IEC61508 标准，这对于诸如汽车电子、工业控制、医疗设备等领域内的应用是十分必要的。

## 跨平台

不可否认的是，当前基于 ARM 架构的微控制器在嵌入式应用中占据了主流，ARM mbed 也是一个相当不错的新一代物联网软件平台，但如其名所示，其只支持 ARM 架构。Zephyr 则是一个跨平台的物联网软件平台，支持多种处理器架构，包括 x86、ARM、ARC、Tensilica 以及最近发展迅速的 RISC-V（SiFive 公司已经成为了 Zephyr 的会员），支持 200+ 以上的开发板。Zephyr 在设计上充分考虑了硬件的抽象性，使得其具有良好的可移植性，未来会支持越来越多的处理器架构和开发板。在开发方式上，Zephyr 支持在 Linux/Unix、Windows 和 Mac OS 下开发。

## 中立

虽然 Zephyr 是由 Intel、Synopsys、NXP 等大公司共同发起，特别是 Intel，投入了大量的资源、贡献了大量的代码，但 Zephyr 是一个中立的平台，不属于任何一家公司，其托管在 Linux 基金会下，实行会员制，每一个会员乃至每一个普通开发者都可以合理地表达自己的意见，参与决策，重大事项由会员通过投票决定。中立是各个利益相关方的最大公约数，避免了平台过多的受某个公司的处于商业利益上的影响，才可能吸引广泛的参与，构建一个良好的生态。Linux 的成功就是一个明证。

## 宽松的许可证

Zephyr 采用 Apache 2.0 许可证，Apache 2.0 许可证是一种商业友好的许可。用户可以把 Zephyr 用于商业目的而无需开放源码。在具体实践中，除 Zephyr 自身代码外，对于外部集成模块，Zephyr 也十分注意，尽可能选择 Apache 2.0 兼容的模块，避免许可证污染。

## 完整的软件平台

对于立志成为“微控制器 Linux”的 Zephyr，在注重轻量级的同时，也是一个非常完整的软件平台，是一个“一站式”的解决方案。Zephyr 包含了操作系统内核、设备驱动、蓝牙协议栈、网络协议栈、测试框架、文件系统框架等众多组件。借鉴了 Linux 内核的成功经验（kconfig 和 device tree），Zephyr 内的组件以及组件内的功能是高度模块化、灵活、可配置的。在开发工具方面，zephyr 提供并定期发布 Zephyr SDK 以提供标准的开发环境。Zephyr 还支持以模块（module）的形式集成第三方的库、框架或者硬件驱动等。

## 支持产品级的开发

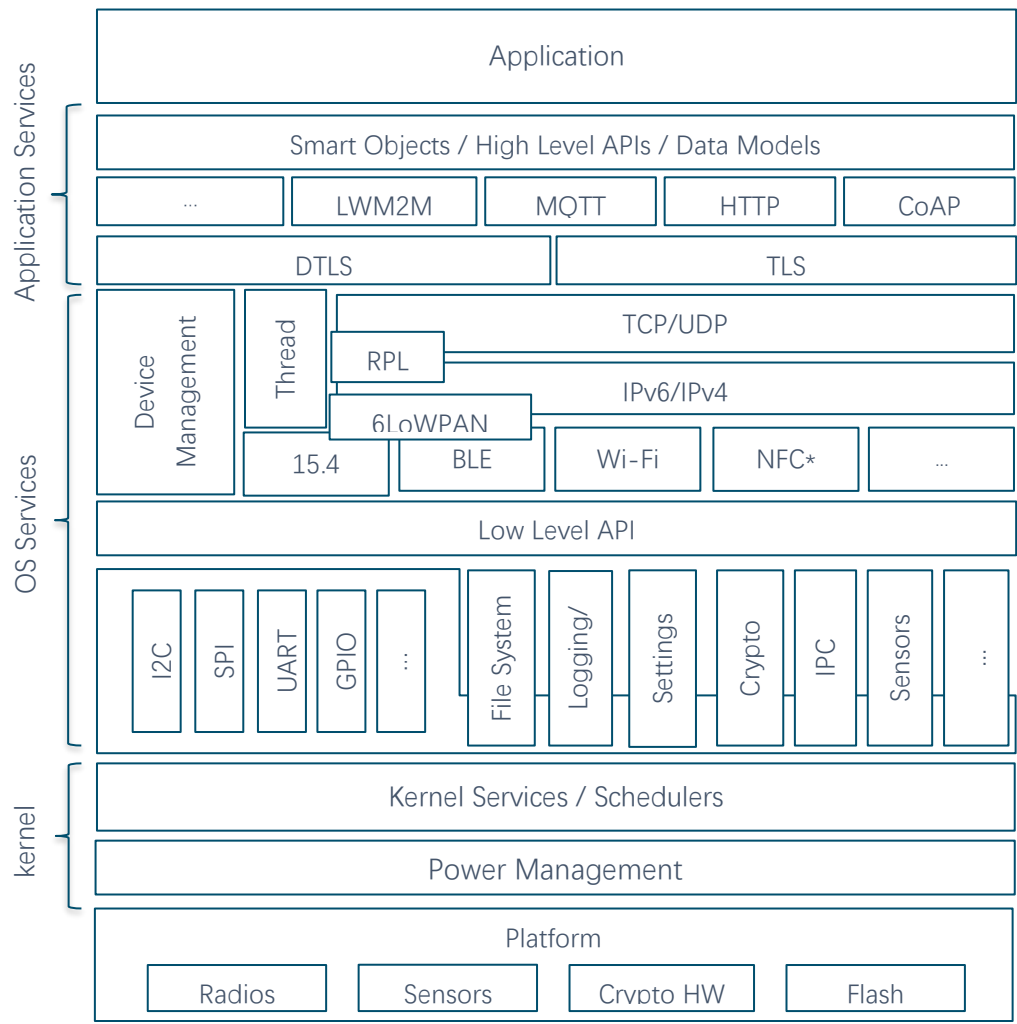
一些 Zephyr 所面向的应用如工业控制、仪器仪表、汽车电子往往生命周期比较长，对于可靠性、稳定性、可维护性等提出了更高的要求，为此 Zephyr 推出了长期支持版 LTS 以满足此类应用，LTS 的发布周期为 2 年，为保证稳定在一个 LTS 周期之内，不会引入新的特性、API 的变化，但会持续进行安全更新、漏洞修复，另外相对于正常版本，LTS 版会经过更多更广泛的测试。

## 通过认证

Zephyr 会尝试通过一系列的认证，如功能安全认证(FuSa)，蓝牙协议栈认证等等。这些认证可以大大减轻开发者和用户在这方面工作，增强使用 Zephyr 的信心。目前 Zephyr 已经通过了 ARM PSA Level 1 的认证。在 LTS 版本之上，Zephyr 还会进一步提供通过相关认证可审计（Auditable）的版本，可以应用于汽车电子等要求高可靠、高稳定的场合。

# 系统框架

Zephyr 的整体的框架如下图所示，大体可以分成三大部分：



Zephyr 系统框图

## 内核

Zephyr 的内核是一个实时内核，包含了实时内核所应有的绝大部分功能， 在此概要列举如下：

- **任务调度。**支持基于优先级的可剥夺 (Preemptive) 和协作(Cooperative)调度，支持基于时间分片 (Time-slicing) 的轮询调度 (Round-robin)，支持 Earliest Deadline First (EDF) 调度算法，可选三种任务就绪队列的实现以满足不同应用需求：简单的单级队列，传统的多级队列，基于红黑树的队列。
- **任务间同步。**支持二进制信号量 (Binary semaphore)，普通信号量 (Counting semaphore) 互斥量(Mutex) 和快速用户态互斥量 (Futex)。
- **任务间数据传输。**支持基本数据队列，增强数据队列和邮箱，支持工作队列(workqueue)
- **内存分配。**所有内核资源编译时静态分配，支持基于可变和固定长内存池的动态内存分

配和释放。

- **内核扩展。**支持内核 Tickless 和功耗管理，支持内存保护、堆栈溢出检测 and 用户/内核态，支持同构多核 SMP 和异构多核 AMP，支持 POSIX API (PSE51 和 PSE52) 同时支持在符合 POSIX API 上的 OS 上，例如 Linux，作为应用运行。

## 内核服务

构建在内核之上，Zephyr 包含了大量的内核服务，在此列举其中比较重要的服务：

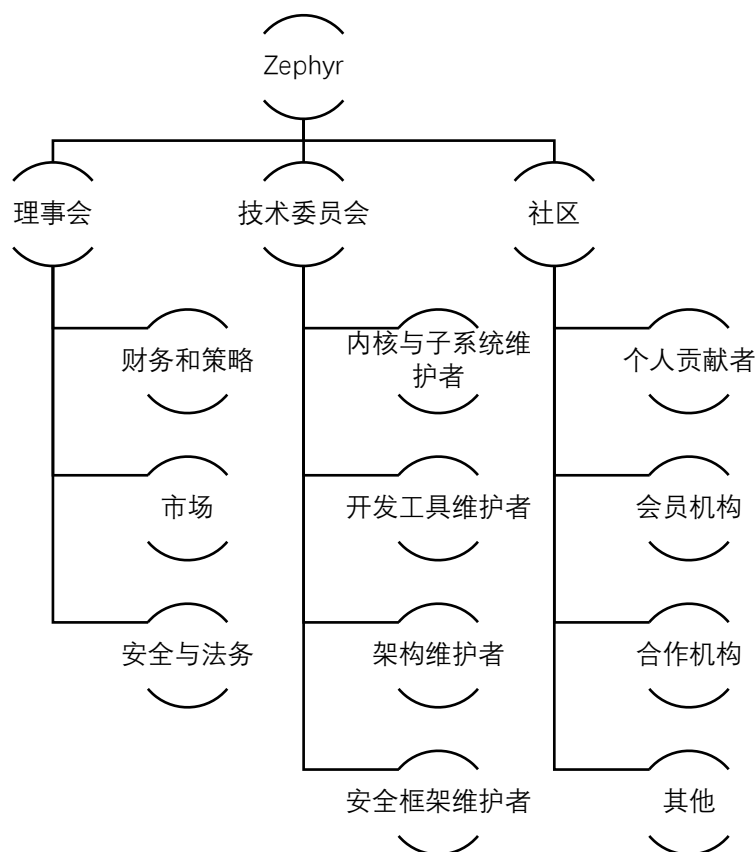
- **设备驱动框架。**如 Linux 内核一样，Zephyr 中有一个比较完善的设备驱动框架，包含了大量的各式各样的设备驱动，驱动实现上借鉴了 Linux 设备驱动的成功经验，并采用了设备树 (Device tree) 和 Kconfig 进行管理和配置，从而可以简化设备驱动的开发和使用。
- **网络协议栈。**在物联网时代，互联互通是重中之重，为此 Zephyr 并没有选择移植一个嵌入式网络协议栈，而是从头开始设计实现。Zephyr 的网络协议栈同时支持 IPv4 和 IPv6 协议，支持多个网络接口，支持多种 L2 连接方式如 ethernet、802.15.4、蓝牙、802.11，CAN 等，这使得在 IPv6 协议下可以统一不同的底层连接方式，支持 SAL(Socket Abstract Layer)可以集成不同网络协议栈的实现，例如集成外部 Wi-Fi、5G、NB-IoT 模块。在应用协议上，Zephyr 包好了 MQTT、COAP 和 LWM2M 等新一代物联网应用协议。
- **蓝牙协议栈。**Zephyr 的蓝牙协议栈是少数几个完整的开源实现，支持蓝牙 5.1 标准，支持蓝牙 Mesh 协议。Zephyr 与蓝牙技术联盟 (Bluetooth SIG) 有着良好合作，蓝牙技术联盟推荐 Zephyr 的蓝牙协议栈作为新一代蓝牙标准的参考实现之一。

## 应用服务

在内核服务之上，Zephyr 也包含了大量的应用层服务和参考例程，可以大大加速应用的开发，例如信息安全库 ARM MBED TLS，图形库 little vgl、新一代物联网协议、和不同云平台的连接支持等等。Zephyr 支持以模块 (module) 的形式集成第三方代码，进一步扩展了 Zephyr 的生态。

## 治理模型

作为一个中立的平台，Zephyr 托管在 Linux 基金会下，其运营由会员公司的会费支持，并且技术导向与管理是分开，其治理模型如下图所示，可以分为三大部分：



## 理事会(Governing Board)

Zephyr 理事会由 Linux 基金会代表、各会员机构代表和技术委员会的代表共同组成，主要负责项目目标与战略的制订，商务、市场和法务上的决策，投资和预算的监管，为技术委员会提供指导，会员需求的协调等事务。理事会定期召开会议，处理相关事务。Zephyr 会员分为付费的白金（Platinum）、银牌（Sliver）和免费的伙伴（Associate）三级，不同级别的会员每年的会费不一样所享受的权力也不一样。

## 技术委员会(Technical Steering Commit)

技术委员会是 Zephyr 的主要执行结构，由各会员机构代表、内核与子系统维护者、开发工具维护者、安全框架维护者和各处理器架构维护者组成，涵盖了 Zephyr 代码的方方面面。技术委员会是 Zephyr 的最高技术决策机构，通过投票的形式做出各种技术上的决策。在具体工作上，技术委员会负责技术方向、路线图的制订，协调各种开发活动，协调和外部项目的合作，负责组织发布。技术委员会下设多个工作组负责协调特定领域的开发活动，如 API、安全、工具等。技术委员会及其下属各工作组会定期举行线上开放会议处理技术相关事务。

## 社区 (Community)

社区是整个 Zephyr 的基础，由 Zephyr 的开发者和用户组成。社区在技术委员会的指导之

下，进行开发活动。Zephyr 各模块的维护者会审阅开发者和用户所提交的各种相关代码、文档、bug 报告等等，并提供意见。具体细节可以从 Zephyr 的 Pull request 列表和 Issue 列表查看。Zephyr 正在构建 Maintainer->Collaborator->Contributor 三级开发者体系，以指导和管理开发活动。开发者可以依据自身贡献成为相应模块的维护者，进而加入技术委员会发挥更积极的作用。

## 机遇与挑战

自成立以来，Zephyr 发展非常迅速，世界范围内越来越多开发者和用户选择了 Zephyr，其正向成为“微控制器的 Linux”这一目标一步一步迈开坚实的脚步。Zephyr 的快速发展对于开发人员和相关机构都可能是一个机遇，这个机遇可能类似于“在 Linux 发展之初就参与到 Linux 的开发使用”。

在 Zephyr 发展过程中，也存在着诸多挑战。ARM 公司依赖于 ARM 架构在微控制器机构上的主导地位，不断强化 ARM Mbed 平台，希望把其在移动互联时代的成功延续到物联网时代。众多巨头公司特别云计算的巨头为了延申其云战略，也把手伸进了嵌入式这个领域，微软收购了 ThreadX，亚马逊收购了 FreeRTOS，阿里推出了 AliOS things，华为推出了 LiteOS。Zephyr 并不能保证一定会笑到最后，其成功之匙就是中立和开放。

在国内，由于其历史较短，使用门槛较高、当前只支持 GNU 工具链而无 IDE 的支持、语言因素（开发资料主要是英文，开发者大多在国外），Zephyr 的受众还很少，目前国内与 Zephyr 相关开发力量主要来至 Intel 亚太研发中心、Synopsys 武汉研发中心、NXP 上海研发中心以及不多个人开发者。中国未来在物联网时代的影响力，如市场、产品等，决定了 Zephyr 在中国的成功是 Zephyr 成功的一个重要因素。针对存在的问题和挑战，一方面 Zephyr 不断在完善自身，同时 Zephyr 中国开发者正与 Zephyr 社区紧密合作不断加强在 Zephyr 在国内的推广，已经建立了微信群与 QQ 群用于在线沟通，并积极组织参加多种相关线上线下活动。

## Zephyr 相关在线资源

官方网站: <https://www.zephyrproject.org>

在线文档: <https://docs.zephyrproject.org/latest>

Github: <https://github.com/zephyrproject-rtos/zephyr>

邮件列表: <https://lists.zephyrproject.org/g/main>

社交频道: <https://zephyrproject.slack.com>

国内: 微信群（须邀请），QQ 群（580070214）