

# 在Zephyr RTOS基础上构建兼容AUTOSAR CP的智能车控操作系统

付宝军<sup>1</sup>, 任 慰<sup>2</sup>

<sup>1</sup>华勤技术, 上海

<sup>2</sup>openEuler社区, 湖北 武汉

收稿日期: 2025年12月22日; 录用日期: 2026年1月16日; 发布日期: 2026年1月29日

---

## 摘要

AUTOSAR CP (Classic Platform)平台虽在提升车辆软件的模块化、可重用性、可扩展性以及跨系统集成方面成效显著, 有效缩短了开发周期、提升了软件质量、降低了成本, 但面对软件定义汽车的巨浪, 其仍面临软件复杂度激增、功能快速迭代与开发效率瓶颈等挑战。与此同时, 始于2016年Zephyr项目作为近年来发展迅猛的现代化开源嵌入式系统软件平台, 以其功能强大、可扩展、安全和高度模块化的架构在嵌入式领域崭露头角。本文深入分析Zephyr的核心架构与成功实践, 结合汽车电子应用场景, 提出一种在Zephyr RTOS基础上构建兼容AUTOSAR CP的智能车控操作系统的构想, 并探讨其技术可行性及所面临的关键挑战, 为未来高可靠、高效率、智能化的车控软件平台演进提供新思路。

---

## 关键词

汽车电子, 操作系统, 实时操作系统, AUTOSAR CP, Zephyr

---

# Building AUTOSAR CP Compatible Vehicle Operating System Based on Zephyr

Baojun Fu<sup>1</sup>, Wei Ren<sup>2</sup>

<sup>1</sup>Shanghai Huaqin Technology, Shanghai

<sup>2</sup>openEuler Community, Wuhan Hubei

Received: December 22, 2025; accepted: January 16, 2026; published: January 29, 2026

---

## Abstract

Although the AUTOSAR Classic Platform has significantly advanced vehicle software in terms of modularity, reusability, scalability, and cross-system integration—effectively shortening development

文章引用: 付宝军, 任慰. 在 Zephyr RTOS 基础上构建兼容 AUTOSAR CP 的智能车控操作系统[J]. 嵌入式技术与智能系统, 2025, 2(5): 319-329. DOI: [10.12677/etis.2025.25031](https://doi.org/10.12677/etis.2025.25031)

cycles, improving software quality, and reducing costs—it now faces substantial challenges in the era of software-defined vehicles, such as rapidly escalating software complexity, the need for fast-paced functional iteration, and bottlenecks in development efficiency. Meanwhile, the Zephyr project, launched in 2016, has emerged as one of the fastest-growing modern open-source embedded software platforms, gaining prominence in the embedded domain due to its robust feature set, scalability, security, and highly modular architecture. This paper presents an in-depth analysis of Zephyr's core architecture and successful practices, and proposes a novel concept for a smart vehicle control operating system built upon the Zephyr RTOS that is compatible with AUTOSAR CP. We further discuss its technical feasibility and key challenges, offering a new pathway toward the evolution of future automotive control software platforms that are highly reliable, efficient, and intelligent.

## Keywords

Automotive Electronic, Operating System, Real-Time Operating System, AUTOSAR CP, Zephyr

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

AUTOSAR 的主要目标是提高车辆软件组件的模块化、可扩展性和可重用性, 以及促进不同系统和技术的集成。AUTOSAR 组织的成员单位将包含自身 IP 的技术成果贡献进 AUTOSAR 标准, 这已经在很大程度上避免了大家“重复造轮子”, 一定程度上实现了缩短开发周期, 提升软件质量, 降低开发成本的目标[1]。同时, 面对软件定义汽车对功能快速扩展和开发效率提升的迫切需求, AUTOSAR 也开始尝试采用“代码优先”的创新模式开发 AUTOSAR AP (Adaptive Platform) 平台[2]。不过, 对于 AUTOSAR CP (Classic Platform) 平台, 当前行业内仍以 Vector、ETAS、普华基础软件等提供的商业解决方案为主, 同时, 也有部分企业在尝试通过自研或开源的方式进行实现。

对于 AUTOSAR CP 平台软件的实现, 目前行业最大的痛点有两个:

一是使用成本较高, 无论采购商业软件还是自研, 都需要大量的研发资金投入。同时各家所实现的软件相互不兼容, 技术碎片化特征明显, 经常需要重复购买, 更加剧了这一问题。这个问题主要受 AUTOSAR 组织早期倡导的“在标准上合作, 在实现上竞争”的商业模式的惯性影响。虽然从早期看有其合理性, 但随着当前开源、降本的呼声越来越高, 这一商业模式正面临严峻挑战。

二是技术迭代慢, 相对行业需求存在一定程度的落后。面对越来越复杂的应用场景和功能越来越强大的多核异构处理器、中央/区域控制器, 各家按照 CP 标准 20 年前即开始定义的架构、功能和方法论开发出的软件难以满足行业需求, 这方面需求虽不如 AP 平台多, 但也存在大量标准以外的新功能需要开发, 同时开发工具链也亟待升级以提高开发效率。

Zephyr 作为 Linux 基金会旗下新一代开源的实时操作系统平台, 在 IoT 与边缘计算领域已验证其可靠性, 具备向车规级演进的潜力。本文旨在初步分析“在 Zephyr RTOS 基础上构建兼容 AUTOSAR CP 的智能车控操作系统”的可行性、意义与价值, 希望为解决以上行业痛点提供参考技术方案。

## 2. 为什么是 Zephyr, 而不是其他 RTOS

Zephyr 是一个由 Linux 基金会主导的开源、可扩展、安全且实时的嵌入式实时操作系统, 专为资源受限的嵌入式系统和物联网设备设计。自 2016 年发布以来, Zephyr 已迅速成长为全球主流的开源嵌入式

操作系统之一, 广泛应用于可穿戴设备、工业控制、医疗设备、智能家居、边缘计算以及新兴的汽车电子领域。

Zephyr 绝不仅仅是“又一个开源 RTOS 罢了”, 而是极有可能统一整个嵌入式 RTOS 世界。理由如下[3]:

(1) **极度活跃的社区**。当其它实时操作系统项目停滞不前或出现下滑时, Zephyr 的贡献者基数自 2017 年以来增长了 7 倍, 在今年达到了平均每月 344 名活跃的开发者, 1640 次提交。如下图 1 所示。

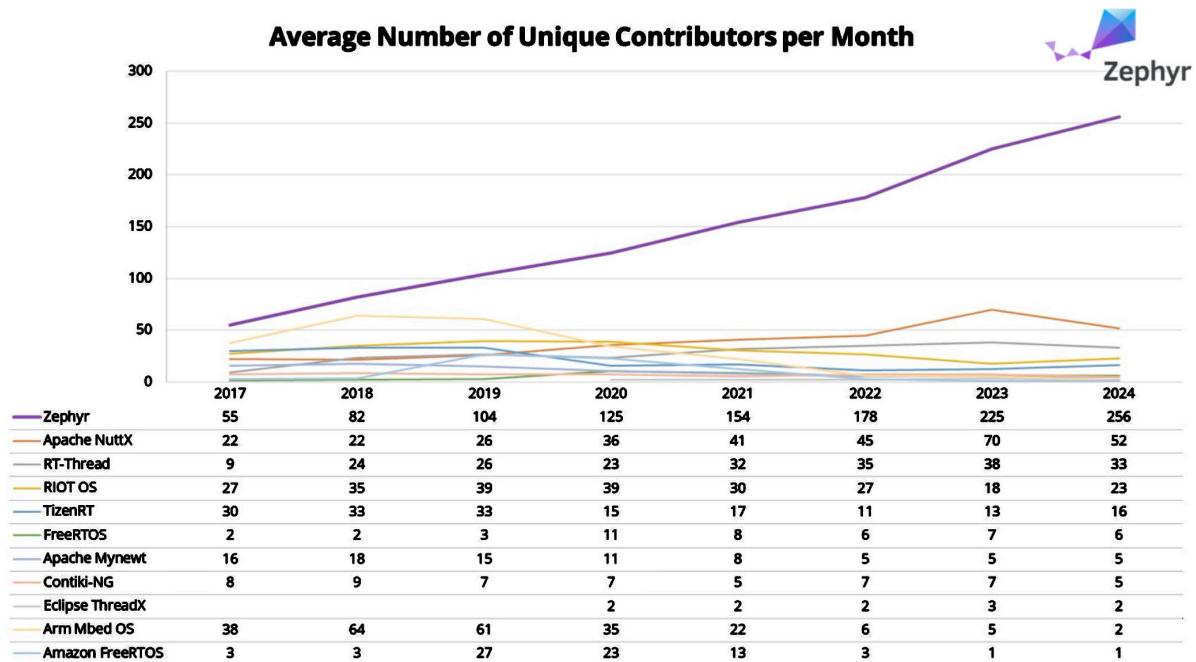


Figure 1. The fast growth of Zephyr contributors

图 1. 快速增长的 Zephyr 开发者及提交

(2) **真正的厂商中立性**。与私有系统或由单一厂商主导的开源项目不同, Zephyr 项目托管在 Linux 基金会旗下, 目前有超过 57 家成员公司和组织参与协作, 包括国内的 openEuler 社区, 湖南大学, 成都菁蓉联创公司等, 而不受单一实体的限制和控制。Zephyr 在开放源码的基础上, 实现了透明开放的开发流程, 以及开放中立的治理。这意味着参与者可以以“贡献越大, 主导权越大”的方式引导它具备高质量和实用的功能集。

(3) **功能丰富, 且高度可配置**。Zephyr 支持 x86、ARM、RISC-V 等几乎所有主流处理器架构, 截至目前已经支持 800 多种开发板、200 多种传感器, 涵盖 NXP、ST、Infineon、Renesas 等众多厂商芯片。同时还提供 RTOS 内核以外的各种不同的通信协议栈、网络安全、文件系统、电源管理等丰富的基础软件功能。

(4) **重视信息安全与功能安全**。Zephyr 在开发时就考虑了功能安全与信息安全, 信息安全方面具有独立的 CVE 编号和完善的治理框架, 功能安全方面目前具备满足行业认证(如 ISO 26262/IEC 61508)的技术条件。

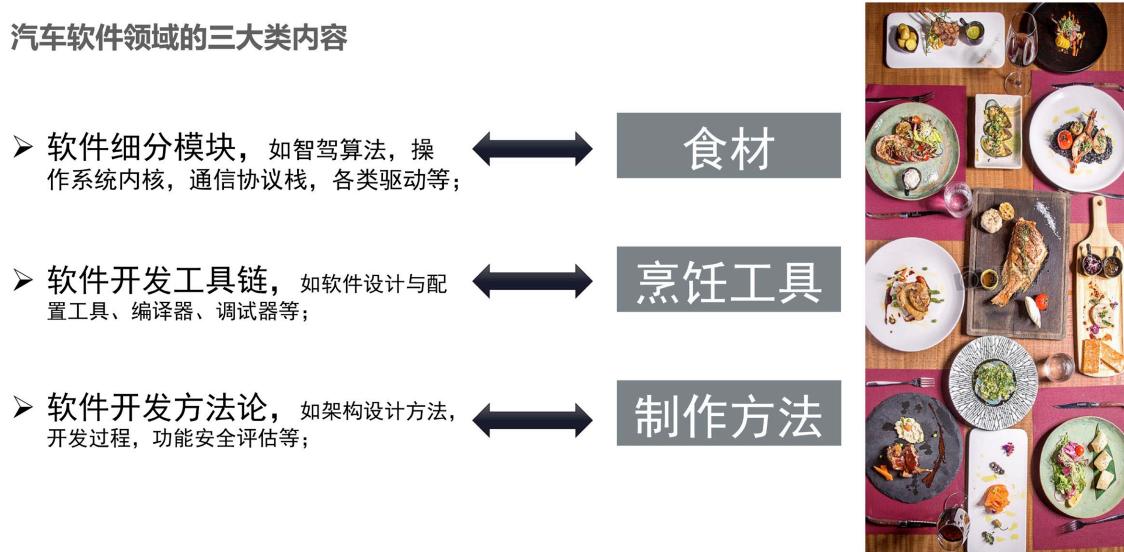
(5) **现代化开发工具链**。Zephyr 不仅是一个操作系统, 更是一套完整的嵌入式开发生态。如基于 Cmake + Python 的构建系统, 使用 Devicetree + Kconfig 对软件进行配置, 采用 West 作为元工具统一管理代码、构建、烧录与调试, 测试框架内置 Ztest 单元测试、覆盖率分析、静态检查工具, 并支持 CI/CD

和容器化开发。

**(6) 对产品开发友好。** Zephyr 项目整体采用对商业友好的 Apache 2.0 许可证, 同时每两年一个长期支持版本(LTS, Long Time Support), 每个 LTS 版本的社区维护周期为五年, 可以高效地支持产品级的开发。

### 3. Zephyr + AUTOSAR CP 中间件 = 面向 MCU 的新一代汽车电子基础软件平台

复杂软件系统的开发如同做一桌大餐, 丰富多样的食材、专业的烹饪工具和熟练的制作方法三者缺一不可, 如下图 2 所示。车载软件的开发也不例外, 要想做出好的汽车软件, 需要在软件功能模块、开发工具链和开发方法论三个方面持续精进。



**Figure 2.** The composition of automotive software  
**图 2.** 汽车软件领域三大内容

Zephyr 的快速发展和 AUTOSAR CP 软件的开源化, 使得汽车基础软件平台围绕以上三个方面的精进有了更具体的目标。

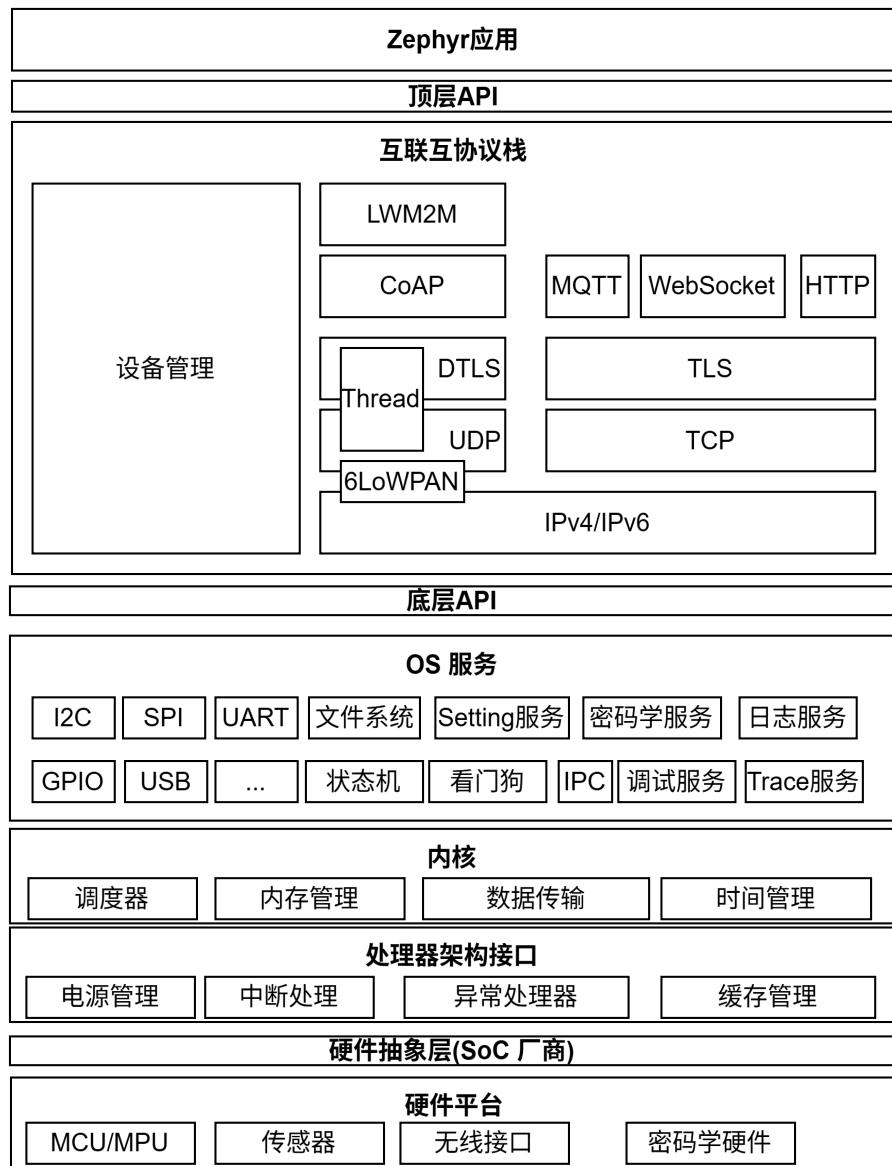
### 3.1. 汇聚 IoT 和汽车两大边缘计算领域平台功能, 实现技术领先

#### 3.1.1. Zephyr 提供丰富多样且极具特色的功能(但“汽车味”不够)

Zephyr 并非通用操作系统, 而是专为 MCU 应用设计的轻量级、硬实时、高安全 RTOS, 同时通过模块化设计兼顾功能完整性。其参考架构图如下图 3 所示。

(1) 全面的内核服务。Zephyr 提供了一系列开发者熟悉的内核服务, 如多线程服务、中断服务、线程间数据传递与同步服务、内存分配服务、电源管理服务等。同时支持协作式与抢占式、时间片轮转、最早截止时间优先等多种调度算法。另外, 还支持大量不同的 CPU 架构和具体芯片型号。虽然不符合 AUTOSAR 所定义的 OS 标准, 但可以通过将线程(Thread)封装为 AUTOSAR 任务(Task)的方式, 实现与 AUTOSAR 软件的兼容。

(2) 提供内存保护机制。Zephyr 实现了可配置的针对特定架构的栈溢出保护、内核对象与设备驱动权限追踪, 以及在线程级内存保护下的线程隔离机制。其内存保护机制主要依赖硬件上 MMU、MPU 和堆栈指针保护机制。



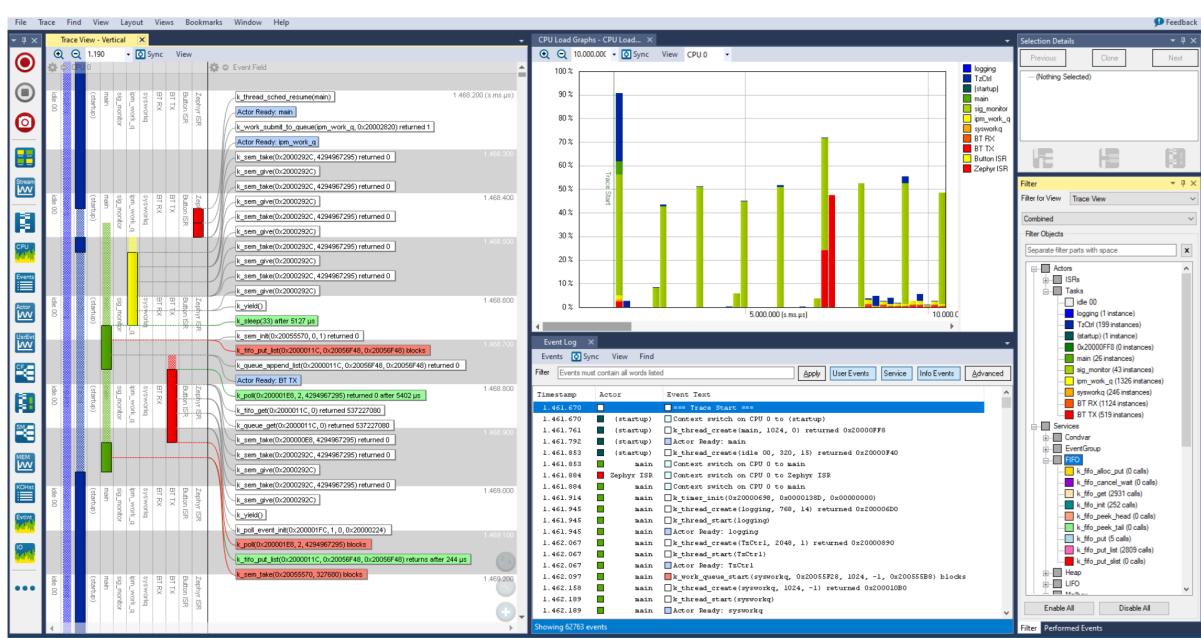
**Figure 3.** The architecture of Zephyr

**图 3.** Zephyr 操作系统架构

(3) 丰富的底层 OS 基础功能服务[4]。Zephyr 不仅仅是一个 RTOS 内核, 还提供了类似 AUTOSAR Memory 技术栈的存储服务, 类似 AUTOSAR MCAL 和 CDD 的各种外设及复杂驱动, 信息安全技术栈, 核间通信, 符合 SOA 思想的 ZBus 机制, 软件远程更新, 以及基于 Zephyr 的嵌入式实时虚拟机 ZVM (Zephyr-Based Virtual Machine)。

(4) 提供支持多种协议的原生以太网技术栈。以太网通信支持功能完整且经过优化, 包含 LwM2M 协议及符合 BSD 套接字标准的支持。同时提供对 OpenThread 的支持。这一特性有利于车载以太网通信功能的开发。

(5) 高效的跟踪与调试功能。Zephyr 内置高级日志记录框架和跟踪框架, 支持 UART、网络、文件系统等多种后端, 配合外部工具可实现内核及其各子系统内部运行状态的可视化, 如下图 4 所示, 以较低的成本即可提升开发效率。



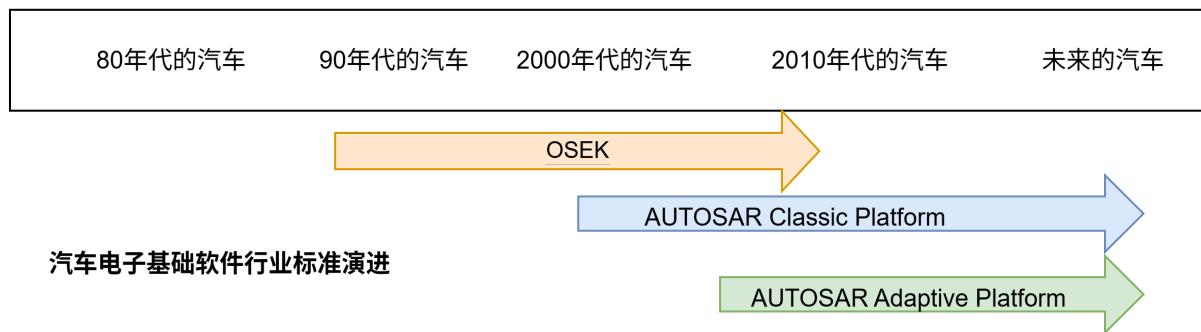
**Figure 4.** Percepio's graphical tool base on Zephyr trace  
**图 4.** Percepio 公司基于 Zephyr Trace 功能图形化分析工具

除以上外, Zephyr 还正在探索 Rust 语言集成、机器学习、大模型轻量化与本地部署等新兴技术领域。

当然,需要注意 Zephyr 所提供的汽车基础软件领域相关标准要求的功能还很少,如诊断协议栈、网络管理、CAN/LIN/Flexray 通信协议栈,不能兼容 AUTOCAR CP 等,尚不足以支撑汽车领域特别是车控领域的广泛应用。

### 3.1.2. AUTOSAR CP 标准定义了丰富的车用基础软件(但“完整性”不够)

相比上世纪 90 年代欧洲汽车行业所定义的 OSEK 标准, AUTOSAR 标准为汽车 ECU 定义了更为完整的基础软件功能。在 OSEK 已定义的操作系统、通信和网络管理三部分标准的基础上,通过横向分层、纵向分技术栈的架构设计,对约 100 个模块的实现进行了标准化,如下图中橙色背景的 OS、Com、NvM 等软件模块。极大地提高了车辆软件组件的模块化、可扩展性和可重用性,并促进了不同系统和技术的集成。如图 5 所示。



**Figure 5.** The evolution of automotive system software  
**图 5.** 汽车电子基础软件行业标准的演变

然而,这种完整性是相对的。一方面,由于所定义的基础软件模块是各家会员单位具有普遍共识的

部分, 而对于缺乏共识, 尤其是自己的“独门秘籍”, 各家其实都“留有后手”, 如复杂驱动。具体可参考《汽车基础软件漫谈2. 汽车基础软件的定义》[5]一文中的介绍。另一方面, 随着汽车智能网联化的快速发展, 产生了 FOTA、Hypervisor 等大量新的需求, 而 AUTOSAR 标准难以及时将这些新需求标准化。因此, 这也就造成当前还有大量的基础软件模块并没有被标准化, 如图 6 中灰色背景的软件模块。

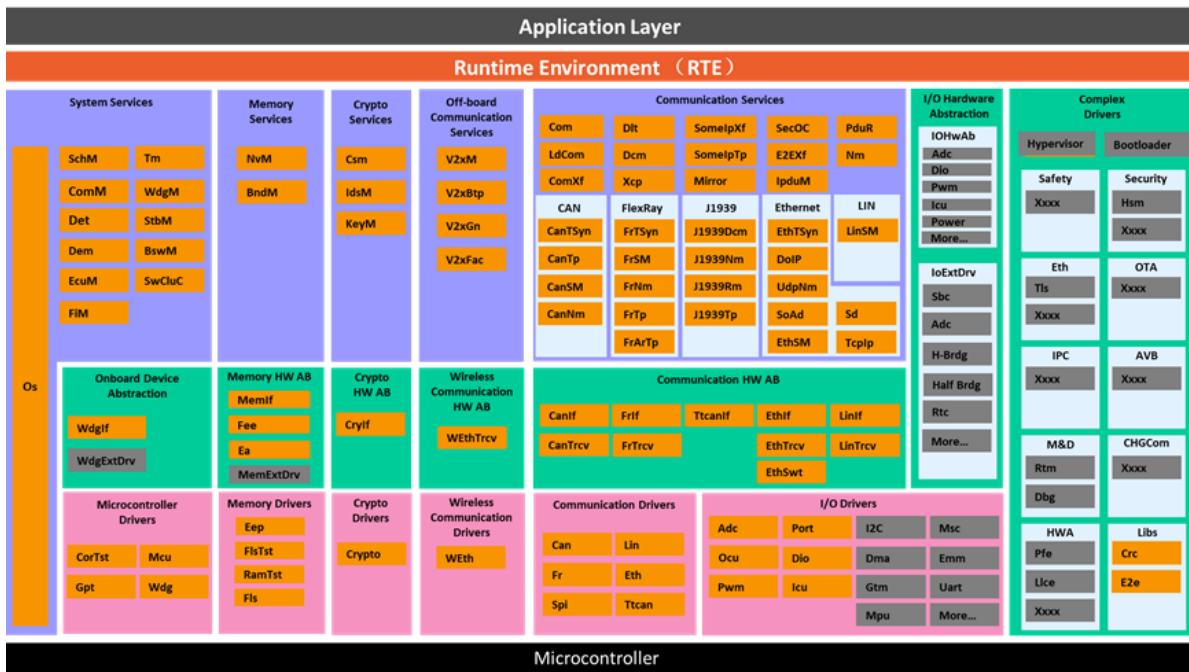


Figure 6. The landscape of AUTOSAR CP  
图 6. AUTOSAR CP 全景图

不过, 从上一节中的分析不难看出, Zephyr 恰好可以大量补充这些未标准化的基础软件模块。如各种硬件驱动、核间通信 IPC, 软总线 ZBus, 软件远程更新 OTA, 以及虚拟化功能等。

### 3.2. Zephyr 提供完整的“现代化嵌入式开发”工具链体系, 大幅提升开发效率

#### 3.2.1. 统一的构建系统: Cmake + Python + West

Zephyr 的构建系统可以形象地概括为“CMake 为骨, Python 为皮”。凭借跨平台、易扩展、驾驭复杂项目结构的先天优势, CMake 早已成为众多顶级开源工程的“基建”——从人工智能(TensorFlow)到计算机视觉(OpenCV), 从桌面框架(Qt)到机器人操作系统(ROS), 几乎无处不在。

Zephyr 在此基础上搭起了一套功能强悍却细节繁复的构建框架。为了不让普通开发者直面 CMake 的“钢筋水泥”, Zephyr 用 Python 包裹出一层简洁的外壳: 元工具 west 一键完成代码拉取、编译、烧录、调试; 配套 Python 脚本则进一步补位, 接管设备树解析、测试调度等杂务。骨子交给 CMake, 面子留给 Python, Zephyr 就这样把“强大”与“友好”一并交到了开发者手里。

#### 3.2.2. 高效的软件配置机制: Kconfig + Devicetree + Yaml

Zephyr 自诞生起就以“把 Linux 验证过的成功范式搬到 RTOS 世界”为己任, 其中最具代表性的两张王牌便是 Kconfig 与 DeviceTree——在实时操作系统里首次做到“全套照搬、完整落地”。

Kconfig 把 Linux 内核“千刀万剐”式的可裁剪能力原封不动搬进 Zephyr: 只要一行 prj.conf, 驱动、协议栈、调试特性均可按需开关, 真正做到“一样源码, 千般身材”。这一方案与 Bosch 在其经典的 Motronic

系列发动机控制器中所使用的“系统常数”机制非常类似。

DeviceTree 则让“板级细节”彻底告别硬编码。Zephyr 的 DT 语法与 Linux 保持兼容, 却玩出了 RTOS 独有的“静态化”花样: 不再由 bootloader 动态解析臃肿的 DTB, 而是借助 Python 脚本在编译阶段直接把设备树翻译成 C 宏 + 初始化代码, 既省资源又省启动时间——思路与 AUTOSAR 的 ARXML 异曲同工, 却胜在肉眼可读、版本可控。

光有这两件利器还不够, Zephyr 又把 YAML 拉进战场: 仓库清单、应用摘要、测试向量一切皆可 YAML 化, 让“配置”成为一等公民。

通过 Kconfig、Devicetree、Yaml 配置文件等一系列措施, Zephyr 实现了相当高程度的代码、配置可复用性, 典型的结果就是在 Zephyr 中基于已有的硬件上去增加类似但又有不同的硬件支持会非常高效率, 这也是 Zephyr 中自带的硬件开发板支持快速增加的根本原因。

### 3.2.3. 强大的调试、测试与优化基础设施

Zephyr 所提供的工具链犹如一把瑞士军刀, 除了提供软件配置与构建工具外, 还提供了大量实用、先进的现代化开发工具。以下挑选其中部分工具进行介绍。

(1) QEMU 模拟器。Zephyr 为 ARM, RISC-V 等各种不同的 CPU 架构提供了 QEMU 模拟器, 使得开发者可以在 PC 环境下, 无需开发板即可对软件进行调试。同时为了进一步提高开发效率, Zephyr 还是在支持 PC 环境下原生运行(Native Support), 也即 Zephyr 应用作为一个 Linux 进程运行, 可以访问网络、可以和蓝牙模拟程序、I/O 模拟程序, 乃至 Matlab 连接, 同时可以比 QEMU 模拟器更快的仿真速度。

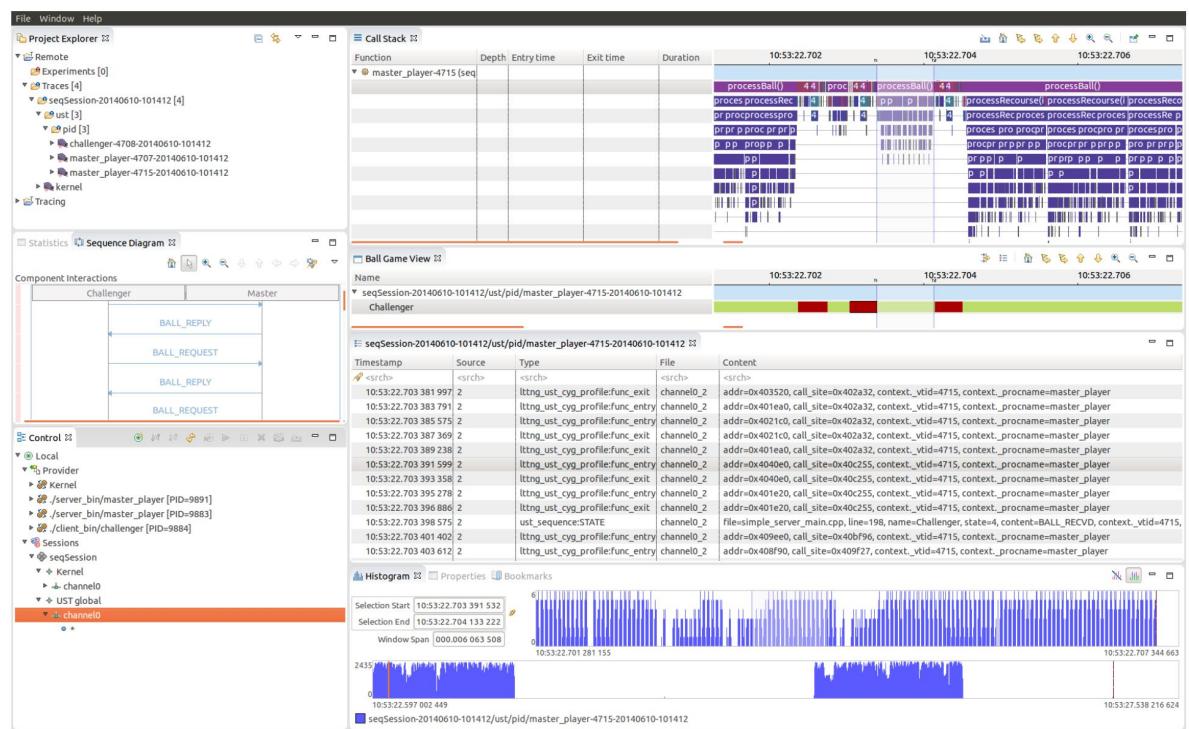


Figure 7. The TraceCompass tool for Zephyr

图 7. Zephyr 中的 TraceCompass 工具

(2) Twister 测试运行器。它是 Zephyr 开发流程中至关重要的一环, 专门用于管理和执行大规模、跨平台的编译与测试。可以将其理解为 Zephyr 的专属持续集成和测试框架。它的核心任务是自动化地验证

Zephyr 代码库的正确性。考虑到 Zephyr 支持超过 800 多块开发板和多种架构(x86、ARM、RISC-V 等), 手动测试所有组合显然是不现实的, Twister 正是为了解决这一难题而生。

(3) 静态代码分析。Zephyr 支持 Coverity、Parasoft C/C++ test、Polyspace、CodeChecker 等近 10 种原生静态代码分析工具, 开发者仅需将自己所使用的工具配置到 Cmake 中即可。

(4) TraceCompass 跟踪调试器。一款由爱立信开发的开源工具, 能够可视化线程调度与中断等并发时序事件, 有助于在复杂系统中发现非预期的交互行为与资源冲突。下图展示了其部分效果, 如图 7 所示。

### 3.3. 关键开发方法: 信息安全管理强, 功能安全有基础

在软件定义汽车和智能网联时代, 功能安全是底线, 信息安全管理是防线。二者均强调系统性、全生命周期、基于风险的开发流程, 共同构成智能汽车可信运行的“双轮驱动”, 是车企技术竞争力、合规准入和用户信任的核心保障。那 Zephyr 在这两个方面有哪些特点, 是否能满足汽车行业的高要求呢?

#### 3.3.1. 信息安全管理

Zephyr 的信息安全管理旨在保护设备在联网环境下的数据机密性、完整性和可用性, 防止恶意攻击。具有以下核心安全特性与机制。

- 安全的启动链与代码完整性

- MCUBoot: Zephyr 官方推荐并深度集成 MCUBoot 作为安全启动引导程序。它通过数字签名验证固件镜像的完整性和真实性, 防止运行被篡改的代码。
- 硬件信任根: 支持利用芯片内的安全硬件(如 TrustZone、TPM、HSM)来安全存储密钥和进行加密操作。

- 加密套件与安全协议

- 丰富的加密库: 内置了稳健的加密算法库, 包括 AES (加密)、SHA (哈希)、RSA/ECC (非对称加密)等, 并支持硬件加速。
- 完整的网络安全协议栈: 原生支持 TLS/DTLS (用于安全通信), 并集成了流行的库如 Mbed TLS。

- 安全的更新机制

- 通过 MCUBoot 实现安全、可靠的固件空中升级。支持 A/B 分区, 确保更新失败后能回滚到旧版本, 保障设备可用性。

- 内核级安全功能

- 内存保护单元支持: 当硬件支持 MPU 或者 MMU 时, Zephyr 可以配置内存区域的访问权限(只读、只执行等), 防止用户态应用程序破坏内核或其他应用的内存。
- 栈溢出保护: 内置栈保护机制(如栈边界检查), 可以有效缓解栈溢出攻击。
- 线程隔离: 支持将线程限制在其专属的内存域中, 限制其可访问的资源, 实现权限分离。

- 信息安全审计与漏洞管理

- Zephyr 项目有一个活跃的信息安全委员会, 负责接收、评估和披露安全漏洞。
- 拥有明确的漏洞报告流程, 并定期发布安全公告。
- 代码库会接受持续的安全审查和静态代码分析(例如, 使用 Coverity Scan)。

总之, 信息安全管理是 Zephyr 的“核心优势”, 其已经具备了工业级应用所需的安全能力。车载软件在很大程度上可以复用, 而无需重新开发。

#### 3.3.2. 功能安全管理

功能安全管理关注的是系统在发生故障或操作失误时, 能避免引发人身伤害、设备损坏或环境危害等风

险。Zephyr 的目标是满足国际功能安全标准。其核心特性与进展如下：

- 功能安全认证准备
  - Zephyr 项目已通过 IEC 61508 (通用功能安全标准) SIL 3 的概念批准(Concept Approval)。正积极推进 IEC 61508 SIL3 的正式认证, 同时准备 ISO 26262 ASIL-D 认证, 以满足汽车电子等高安全等级应用需求。
  - 项目正在为此进行系统性准备, 包括建立完善的开发流程、文档和验证体系。
- 高可靠性内核设计
  - 确定性行为: 作为 RTOS, Zephyr 具有确定性的线程调度和中断响应时间, 这对于安全关键型应用至关重要。
  - 错误处理: 内核包含多种运行时错误检测机制, 如断言检查、空指针解引用检测等, 能及时报告系统异常。
- 测试与验证
  - 强大的测试框架: 如前所述的 Twister 工具, 用于进行大规模、自动化的回归测试, 确保代码更改不会引入新的缺陷。
  - 高测试覆盖率: 项目对内核和关键模块有极高的代码覆盖率要求, 并通过测试用例库和模拟器(如 QEMU)持续执行。
  - 静态代码分析: 强制使用静态分析工具, 在代码合并前发现潜在缺陷。
- 详细的文档与可追溯性:
  - 功能安全认证要求严格的流程。Zephyr 项目正在完善其设计文档、需求规格说明和测试计划, 确保开发过程的可追溯性。

总之, 功能安全是 Zephyr 的“战略方向”, 它正在系统性地为未来的国际安全认证做准备, 虽然目前还不是一个现成的已认证操作系统。

#### 4. 总结



**Figure 8.** The reference software platform combined Zephyr and AUTOSAR CP  
**图 8.** 整合 Zephyr RTOS 和 AUTOSAR CP 中间件的车控操作系统平台

伴随着普华开源小满、理想星环 OS 等开源汽车操作系统的出现, 以 Zephyr 项目为基础, 在其丰富

多样的基础功能、高效现代化的工具链体系和快速提升的安全性的基础上, 将开源汽车操作系统中相关软件模块与其整合在一起, 如图 8 所示, 从而打造一个开放、领先、安全、高效的开源智能车控操作系统的可能性正逐渐增大。尤其是当考虑到 Zephyr 正在以每月近 2000 次的提交速度高速成长时, 更让我们对未来充满期待。

## 参考文献

- [1] AUTOSAR 组织. AUTOSAR 和开源引发热议, 管控车用 OS 复杂度、实现降本增效走哪条路, 看这一篇就够了[EB/OL]. <https://mp.weixin.qq.com/s/0CfGiESYYrU0bu6bJHp8mg>, 2025-04-09.
- [2] AUTOSAR 组织. AUTOSAR CAPI: 构建统一软件定义汽车(SDV)生态的基础[EB/OL]. [https://mp.weixin.qq.com/s/UhisFLGvpiB3\\_VRrdREuWA](https://mp.weixin.qq.com/s/UhisFLGvpiB3_VRrdREuWA), 2025-09-09.
- [3] 嵌入式系统专家之声. Zephyr 实时操作系统将主导嵌入式世界的五个理由[EB/OL]. <https://mp.weixin.qq.com/s/1NpxKHIHtUK8izlnqYoX9oA>, 2025-05-10.
- [4] HalfCoder. Zephyr 2025 开发者大会演讲文档概览[EB/OL]. <https://mp.weixin.qq.com/s/9IcJKeSXi7XOnkwgZ33cGg>, 2025-08-31.
- [5] 汽车软件开发 BBS. 汽车基础软件漫谈|2. 汽车基础软件的定义[EB/OL]. <https://mp.weixin.qq.com/s/d5Gitj3ugiozIHF0J3byw>, 2022-06-01.