

# 嵌入式系统体系架构研究现状与发展趋势

赵庆玲, 周江南

南京理工大学计算机科学与工程学院, 江苏 南京

收稿日期: 2025年10月28日; 录用日期: 2025年11月20日; 发布日期: 2025年12月1日

## 摘要

嵌入式系统作为信息技术的神经末梢, 经历了从1971年微处理器出现到半个世纪的演进, 形成了硬件中心、RTOS支撑、SoC集成、AI边缘等多阶段发展。文章系统阐述了嵌入式系统体系架构的研究现状与发展趋势, 回顾了国外从Apollo制导计算机到现代AI异构平台的演进历程, 梳理了国内自80年代引进模仿到近年来鸿蒙微内核、RISC-V和边缘AI的自主创新路径, 对比了中外在技术起步、生态成熟度和创新模式上的差异, 最后展望了异构计算、微内核模块化、边缘智能和分布式协同等未来发展方向, 指出嵌入式架构将向高性能、安全、可扩展和智能协同演进。

## 关键词

嵌入式系统, 体系架构, 实时操作系统, SoC, 异构计算

# Current Research Status and Development Trends of Embedded System Architecture

Qingling Zhao, Jiangnan Zhou

School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing Jiangsu

Received: October 28, 2025; accepted: November 20, 2025; published: December 1, 2025

## Abstract

Embedded systems serve as the neural endpoints of information technology. Since the introduction of microprocessors in 1971, they have evolved over half a century through stages including hardware-centric design, real-time operating system (RTOS) support, System-on-Chip (SoC) integration, and edge AI applications. This paper systematizes the research status and development trends of embedded system architectures. It reviews the international evolution from the Apollo Guidance Computer to contemporary AI-driven heterogeneous platforms, and outlines China's developmental trajectory—from technology introduction in the 1980s to recent indigenous innovations such as

the HarmonyOS microkernel, RISC-V architecture, and edge intelligence. A comparative analysis is conducted between China and other countries regarding technological origins, ecosystem maturity, and innovation models. Finally, the paper prospects future trends—including heterogeneous computing, modular microkernel design, edge intelligence, and distributed coordination—highlighting the ongoing evolution of embedded architectures toward higher performance, security, scalability, and intelligent collaboration.

## Keywords

Embedded Systems, System Architecture, Real-Time Operating System, SoC, Heterogeneous Computing

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

嵌入式系统作为现代信息技术的“神经末梢”，是嵌入到物理设备中的专用计算系统，通过软硬件协同实现对设备功能、性能及交互的智能控制。从 1971 年 11 月第一款微处理器的推出[1]，经过半个世纪的发展，嵌入式系统体系架构已经囊括虚拟化、微服务、异构并行计算和边缘计算等多个领域。

## 2. 国外发展现状

图 1 为美国 NASA 为阿波罗登月任务设计的 Apollo Guidance Computer，这是最早的嵌入式设备之一，具有定制微控制器和只读程序存储器，用于实现特定的飞行控制功能[2]。此阶段嵌入式架构以硬件为中心，系统功能固定，控制方式以轮询和简单中断为主，尚未形成操作系统支持的体系结构。

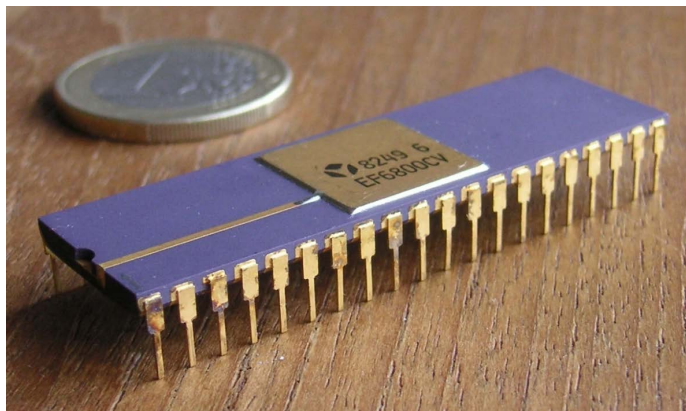


Figure 1. Apollo guidance computer

图 1. 阿波罗制导计算机

进入 70 年代末至 80 年代，随着微处理器的快速发展，嵌入式系统开始普及。Intel 推出的 8080、8051 以及 Motorola 6800 (如图 2)等处理器成为工业自动化、医疗电子等领域嵌入式系统的主力平台[3]。系统软件方面，实时操作系统(RTOS)的出现使得嵌入式系统首次具备了多任务调度、中断优先级、时间管理

等功能[4], 体系结构开始向分层设计演进。

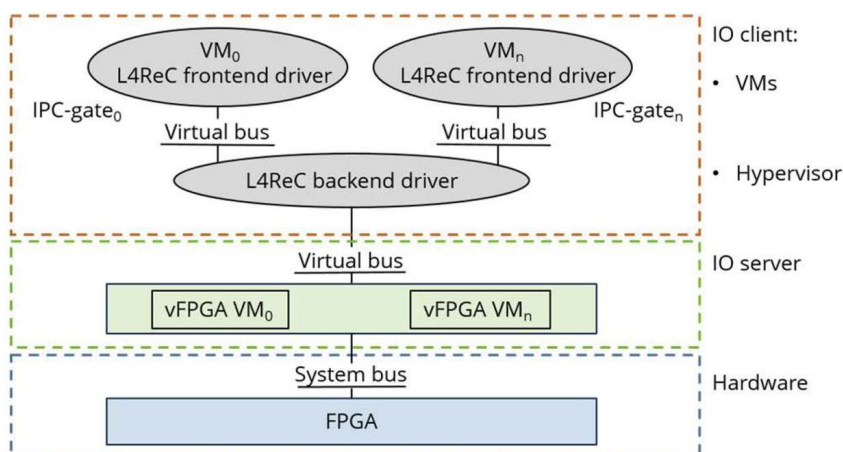


**Figure 2.** Motorola 6800 chip  
**图 2.** Motorola 6800 芯片

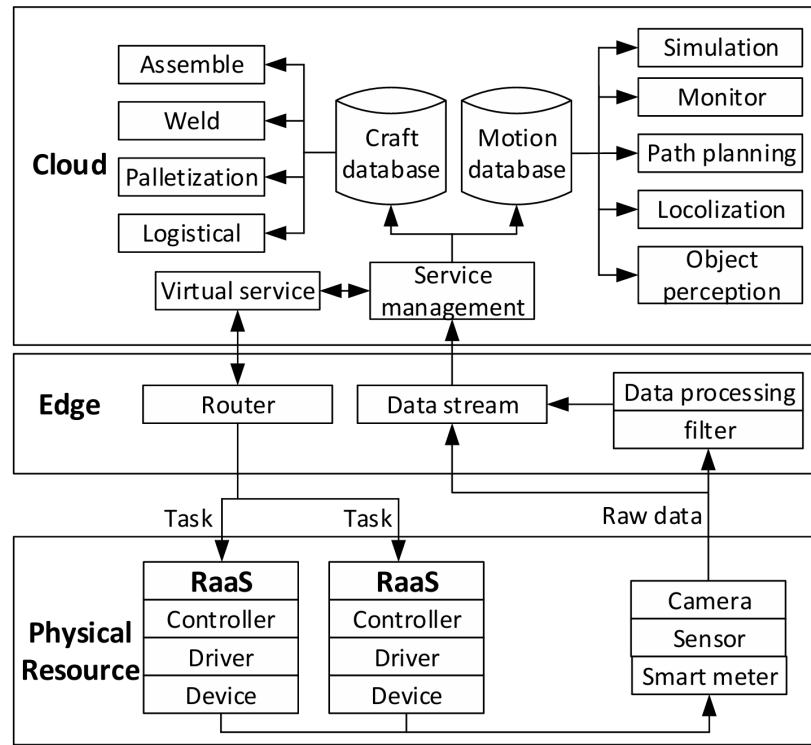
1990 年代是嵌入式架构发展的关键时期。System-on-Chip (SoC)设计理念兴起, 处理器厂商开始将 CPU 核、DMA、内存控制器等功能集成于单芯片, 大幅提升了性能与集成度[2]。同时, ARM 架构以其低功耗、高性能的优势迅速崛起, 成为嵌入式处理器的主流。

21 世纪初, 嵌入式系统迈向网络化与智能化阶段, 体系架构呈现出以硬件抽象层、RTOS 内核、中间件、应用层组成的分层结构, 并广泛应用轻量级协议(如 MQTT)与远程交互机制, 适应物联网的需求[4][5]。此阶段的嵌入式系统已不再局限于本地控制, 而是成为边缘智能节点, 参与更大范围的协同计算体系, 基于物联网与边缘计算的设计被广泛提出, 如图 3 就是一款基于边缘计算的工业机器人系统设计架构。

近年来, 人工智能和边缘计算进一步推动嵌入式体系结构升级。以 NVIDIA Jetson、Google Edge TPU 等平台为代表的新一代嵌入式系统, 采用异构多核架构(CPU + GPU + NPU), 具备强大的 AI 推理能力[5]。操作系统则向微内核、分布式和虚拟化方向发展(如 L4ReC)(见图 4), 这是一款基于微内核设计的操作系统, 旨在为嵌入式系统和虚拟化提供支持。操作系统强调实时性、安全性与可扩展性[6], 系统结构设计更注重模块解耦、资源隔离与 AI 能力的深度融合, 使嵌入式系统在自动驾驶、工业视觉、智能终端等领域发挥关键作用。



**Figure 3.** L4ReC Embedded reconfigurable system  
**图 3.** L4ReC 嵌入式可重构系统



**Figure 4.** Edge-computing-based industrial robot system architecture diagram  
**图 4.** 基于边缘计算的工业机器人系统架构设计图

总体来看, 国外嵌入式系统体系架构的发展历程体现了从专用向通用、从简单向智能、从单核向异构的演进趋势。随着 AI、IoT 与通信技术融合不断深入, 嵌入式架构将持续在性能、安全、可扩展性与智能协同等方面推进创新, 继续支撑下一代智能系统的发展。

### 3. 国内发展现状

中国嵌入式系统体系架构的发展历程始于 20 世纪 80 年代, 早期嵌入式系统主要依赖引进国外平台和器件[7], 这一阶段的嵌入式架构尚未形成完整体系, 主要为高校和研究所基于单片机进行的开发实验与科研探索。

进入 90 年代中后期, 随着电子信息技术在国内的快速发展, 嵌入式系统逐步进入产业应用阶段。高校和企业开始引入国外嵌入式处理器和开发平台, 如 ARM7、ARM9 等, 同时接触并应用一些主流实时操作系统(RTOS), 包括 uC/OS-II、VxWorks [8]等。

进入 2000 年代, 嵌入式系统在国内的应用领域迅速扩大, 涵盖工业控制、家电控制、通信设备、消费电子等。体系架构进一步发展, 开始采用基于 SoC 的设计模式, 集成度和复杂性显著提升。国产处理器如龙芯、飞腾、君正等逐步推向市场[9], RT-Thread、MiniGUI、嵌入式 Linux 等国产操作系统及图形界面库也在不断完善[4]。

2010 年以后, 随着物联网和智能硬件兴起, 嵌入式系统体系结构进一步向网络化、平台化和智能化演化。国产芯片厂商如兆易创新、全志科技、瑞芯微等崛起, 推出面向物联网和边缘计算的 MCU、MPU 平台[10], 嵌入式处理器架构向 ARM Cortex-M (如图 5)和 Cortex-A 系列靠拢, 支持多核并发、外设接口丰富, 具备较高实时性与低功耗特性。操作系统方面, RT-Thread 发展出 Nano、Smart 等不同形态, 华为 LiteOS、阿里 AliOS Things 也加入国产 RTOS 阵营, 支持 BLE、Wi-Fi、NB-IoT 等通信协议栈, 形成以

RTOS 为核心、多层中间件协同的完整体系结构[11]。

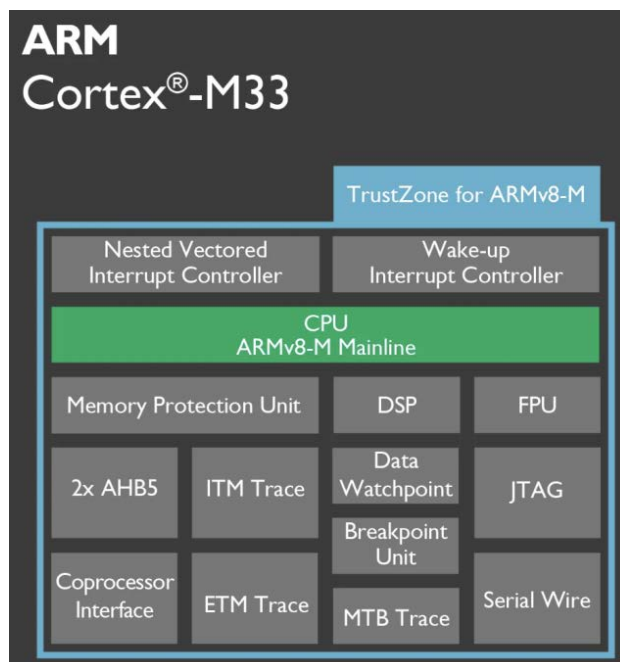


Figure 5. ARM Cortex-M architecture

图 5. ARM Cortex-M 架构

近年来,随着鸿蒙操作系统的发布以及 RISC-V 开源指令集在国内的推广,中国嵌入式体系架构正进入融合创新的新阶段[12]。鸿蒙 OS 以微内核与分布式架构为核心理念,针对多设备协同运行、端云协同构建和高可靠实时控制场景进行了架构设计,推动国产嵌入式系统向泛终端、泛网络协同方向演进。同时,赛昉、芯来、全志等公司纷纷推出 RISC-V 架构芯片,结合轻量化操作系统与推理框架(如 Tengine [13], 见图 6),在智能终端、边缘 AI 等场景中形成新的体系结构路径。翼辉信息自主研发的 ECS 嵌入式安全容器通过  $\mu\text{s}$  级实时响应、MB 级轻量化资源隔离和全架构兼容特性,为航空航天、智能制造等关键领域提供自主可控的容器化基础设施,填补了传统容器在实时嵌入式系统的技术空白,推动任务关键系统向云原生架构安全演进。翼辉 VSOA 微服务框架凭借全双工通信、内置服务发现和 SIL4 安全认证等核心能力,为嵌入式实时系统提供高可靠、低延迟的分布式服务架构,全面满足汽车电子和工业控制等关键领域对实时性、安全性和灵活性的严苛要求,加速嵌入式系统向云原生架构的智能化升级。总体来看,中国嵌入式系统体系架构的发展实现了从引进模仿到自主构建的转变,未来将在异构计算、低功耗 AI、安全可信计算等领域持续创新,构建以国产软硬件为核心的新型嵌入式技术生态。

在嵌入式系统体系结构中,混合关键性系统代表了一种从传统的“一功能一硬件”的联邦式架构向基于资源共享的集成式架构演进的核心范式。其根本目标是在单个高性能硬件平台(如多核处理器或片上系统 SoC)上,整合运行具有不同安全关键性等级(如安全关键、任务关键、非关键)的软件组件,从而显著降低系统的尺寸、重量、功耗和成本。这种架构的核心设计原则是保证时空隔离性。与传统架构依赖物理隔离来实现安全不同,混合关键性系统通过硬件与系统软件层面的协同设计,在共享资源上构建逻辑隔离。在时间维度,采用时间分区调度或增强型实时调度算法,确保高关键性任务的计算时间预算不被低关键性任务抢占;在空间维度,利用内存管理单元(MMU)或内存保护单元(MPU)为每个分区或任务提供独立受保护的地址空间,防止错误的内存访问造成跨任务数据污染。因此,在嵌入式系统架构中,



混合关键性系统是实现“软件定义”功能集中化(如汽车域控制器、航空综合模块化航电)的核心基石,它通过在集成化与安全性之间取得精密平衡,推动了嵌入式系统向更高效、更灵活、更经济的方向发展。国内在工业(如 OpenEuler Embedded [14])和学术界[15]-[18]都有大量的关于混合关键性系统的工作。

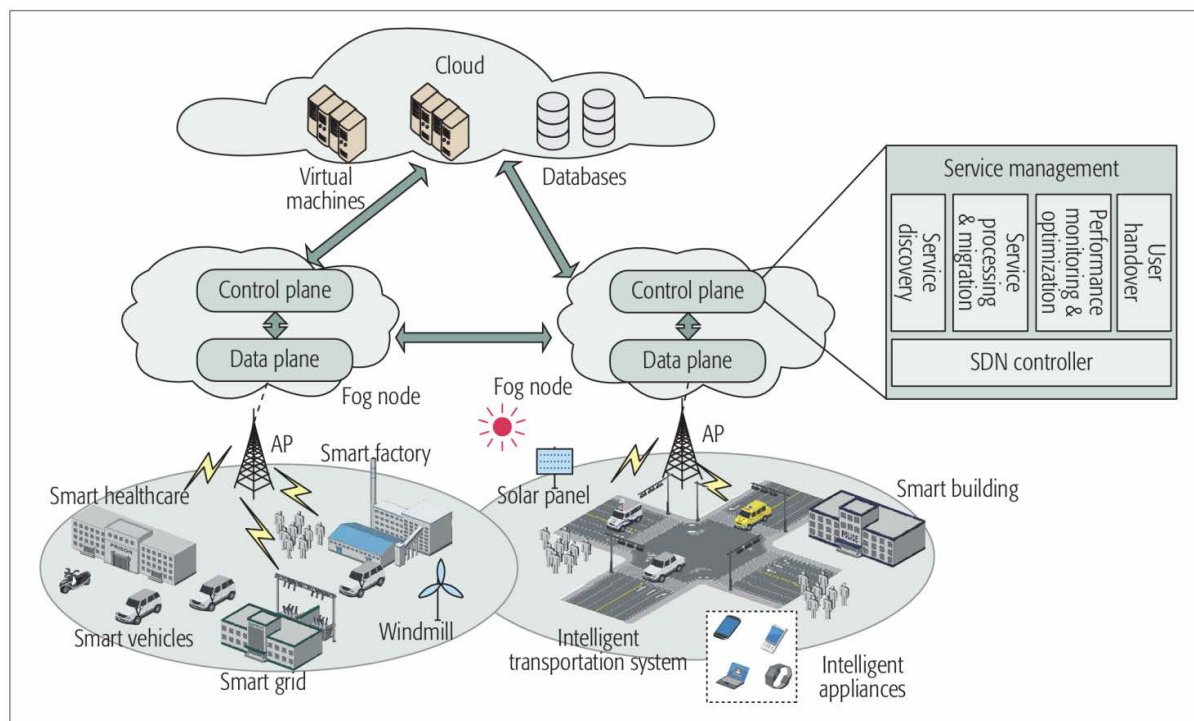


Figure 6. ECIoT architecture  
图 6. ECIoT 架构

#### 4. 国内外发展对比

嵌入式系统作为信息技术与工业应用融合的关键技术,其体系架构的发展在国内外呈现出不同的路径与节奏。国外嵌入式体系架构的发展起步早,以美国 NASA 的 Apollo 导航计算机为代表,随着微处理器的出现,嵌入式系统逐渐形成以微处理器+实时操作系统为核心的架构形态。进入 90 年代,国外体系结构迅速迈入 SoC 集成阶段,ARM、MIPS 等处理器架构与嵌入式 Linux、VxWorks、QNX 等操作系统紧密结合,形成软硬件高度耦合、支持多任务和网络通信的成熟平台,广泛应用于通信、汽车、消费电子等领域。

相比之下,中国的嵌入式系统体系架构起步于 20 世纪 80 年代,初期以引进国外芯片与开发板为主,系统功能主要集中在工业控制、仪器仪表等场景。90 年代后,随着高校教学与科研的推广,ARM 架构逐渐在国内普及,嵌入式软件架构逐步从裸机控制向分层设计演进。进入新世纪后,中国在政策支持和产业驱动下,大力推动自主嵌入式体系的发展,体系结构日趋规范化与平台化。

当前,国外嵌入式体系架构已广泛实现多核异构、AI 协同、虚拟化等先进特性,嵌入式设备具备强大边缘计算能力,并通过微内核架构与分布式机制实现设备间协同控制。而国内在保持稳定控制能力的基础上,正加速向智能化、网络化方向迈进,鸿蒙 OS 的提出标志着国内嵌入式体系在分布式架构方面的重大突破,同时 RISC-V 指令集的兴起也为中国构建开放、自主的体系结构提供了新路径。总体而言,国外嵌入式体系以长期积累和技术成熟为优势,形成了完整生态和标准体系,而中国则通过政策引导和技

术追赶, 逐步建立起以自主可控为目标的本土嵌入式架构体系。

## 5. 发展趋势与展望

随着新一代信息技术的迅速发展, 嵌入式系统体系架构正面临前所未有的机遇与挑战, 从传统的单芯片控制逐步走向异构融合、智能协同和边缘自治, 呈现出多个显著的发展趋势。

首先, 嵌入式体系架构正朝着异构计算方向快速发展。传统嵌入式系统以单一 CPU 或 MCU 为核心, 难以满足当前对计算密度、低功耗和多任务并行处理的需求。为了应对人工智能、图像识别、实时决策等高复杂度应用, 嵌入式平台逐步集成多种异构计算单元, 如 GPU、DSP、FPGA、NPU 等, 形成 CPU+X 的多核协同架构[2]。

其次, 操作系统和软件体系正向微内核、模块化、组件化方向演进。嵌入式操作系统正在逐步摒弃传统的单体内核设计, 采用更具安全性和可扩展性的微内核架构, 能够支持动态加载、服务隔离、权限分离等机制, 提高系统的可管理性和容错能力。同时, 中间件技术和轻量级虚拟化工具也被广泛引入, 以便于跨平台部署和设备间协同。操作系统还不断向下适配资源受限设备, 向上对接物联网平台和云服务, 成为连接设备、网络和服务的核心软件支撑。

随着人工智能技术的普及, 嵌入式系统正逐步具备边缘智能的能力。过去需要通过云端完成的图像识别、语音处理、目标检测等任务, 现今越来越多地在终端设备本地完成, 降低了延迟, 减少了带宽占用, 同时提升了系统的隐私保护能力。这一趋势推动了边缘 AI 芯片、TinyML 算法和小模型加速库的快速发展, 也对架构设计的灵活性与可裁剪性提出了更高要求[19]。

未来, 嵌入式系统体系架构还将继续向分布式与协同计算演化。在车联网、工业物联网、智慧城市等复杂应用场景中, 多个嵌入式节点需要以高可靠、低延迟的方式协同工作。这要求体系架构不仅关注单一设备的功能实现, 还需考虑网络通信协议、分布式资源调度、实时同步与容灾机制等系统性问题。

综上所述, 嵌入式系统体系架构的发展正从功能单一、结构固定的传统模式, 迈向智能化、异构化、安全化与开放化的新阶段。未来的嵌入式系统将不仅仅是“设备的大脑”, 更将成为边缘智能协同网络中的关键节点, 其体系架构设计也将持续融合多领域技术, 成为推动智能社会发展的核心支柱。

## 参考文献

- [1] Lenhardt, D., O'Malley, M., Payne, C., *et al.* (2004) A Look at Intel Processors from the 4004 to the Pentium Pro. <https://docsbay.net/doc/562505/a-look-at-intel-processors-from-the-4004-to-the-pentium-pro>
- [2] 何小庆, 何灵渊. 嵌入式软件开发的三个趋势[J]. 嵌入式技术与智能系统, 2024, 1(1): 1-10.
- [3] Wray, W.C. and Greenfield, J.D. (1994) Using Microprocessors and Microcomputers the Motorola Family. Prentice-Hall Inc.
- [4] 何小庆. 国产嵌入式操作系统发展思考[J]. 单片机与嵌入式系统应用, 2019, 19(12): 4-5+10.
- [5] Capogrosso, L., Cunico, F., Cheng, D.S., Fummi, F. and Cristani, M. (2024) A Machine Learning-Oriented Survey on Tiny Machine Learning. *IEEE Access*, **12**, 23406-23426. <https://doi.org/10.1109/access.2024.3365349>
- [6] Wu, M., Duan, Z. and Li, J. (2024) Performance Optimization of Microkernel-Based Embedded I/O Virtualization Techniques. *International Conference on Artificial Intelligence, Automation and High Performance Computing*, Zhuhai, 19-21 July 2024, 134-141. <https://doi.org/10.1145/3690931.3690955>
- [7] 王福刚, 杨文君, 葛良全. 嵌入式系统的发展与展望[J]. 计算机测量与控制, 2014, 22(12): 3843-3847.
- [8] Behnam, M., Nolte, T., Shin, I., *et al.* (2008) Towards Hierarchical Scheduling in VxWorks. *Proceedings of the 4th International Workshop on Operating Systems Platforms for Embedded Real-Time Applications*, Prague, 1 July 2008, 63-72.
- [9] 平安证券. 国产 CPU 引领者, 构建自主可控生态体系[EB/OL]. <https://www.vzkoo.com/document/20221213c77b3a451721616c054a00d7.html>, 2022-12-12.
- [10] 华福证券. 全志科技首次覆盖: AIoT 创新迭出, 智能硬件 SoC 厚积薄发[EB/OL].

- [https://m.hibor.com.cn/wap\\_detail.aspx?id=a0681ac3f7ef31601edbb5bcd3a27b08](https://m.hibor.com.cn/wap_detail.aspx?id=a0681ac3f7ef31601edbb5bcd3a27b08), 2024-08-07.
- [11] 何小庆. 3 种物联网操作系统分析与比较[J]. 微纳电子与智能制造杂志, 2020, 2(1): 65-72.
  - [12] 刘畅, 武延军, 吴敬征, 等. RISC-V 指令集架构研究综述[J]. 软件学报, 2021, 32(12): 3992-4024.
  - [13] Sakamura, K. and Koshizuka, N. (2003) T-Engine: The Open, Real-Time Embedded-Systems Platform. *IEEE Micro*, **22**, 48-57. <https://doi.org/10.1109/mm.2002.1134343>
  - [14] openEuler 在 RISC-V 生态论坛发布 AI 边缘计算解决方案[J]. 自动化博览, 2024, 41(2): 5.
  - [15] Zhao, Q.L., Gu, Z.H., Zeng, H.B. and Zheng, N.G. (2018) Schedulability Analysis and Stack Size Minimization with Preemption Thresholds and Mixed-Criticality Scheduling. *Journal of Systems Architecture*, **83**, 57-74. <https://doi.org/10.1016/j.sysarc.2017.03.007>
  - [16] Zhao, Q.L., Gu, Z.H. and Zeng, H.B. (2017) Design Optimization for AUTOSAR Models with Preemption Thresholds and Mixed-Criticality Scheduling. *Journal of Systems Architecture*, **72**, 61-68. <https://doi.org/10.1016/j.sysarc.2016.08.003>
  - [17] Zhao, Q.L., Al-Bayati, Z., Gu, Z.H. and Zeng, H.B. (2016) Optimized Implementation of Multirate Mixed-Criticality Synchronous Reactive Models. *ACM Transactions on Design Automation of Electronic Systems*, **22**, 1-25. <https://doi.org/10.1145/2968445>
  - [18] Zhao, Q.L., Gu, Z.H. and Zeng, H.B. (2015) Resource Synchronization and Preemption Thresholds within Mixed-Criticality Scheduling. *ACM Transactions on Embedded Computing Systems*, **14**, 1-25. <https://doi.org/10.1145/2783440>
  - [19] Dutta, D.L. and Bharali, S. (2021) Tinyml Meets IoT: A Comprehensive Survey. *Internet of Things*, **16**, Article 100461. <https://doi.org/10.1016/j.iot.2021.100461>