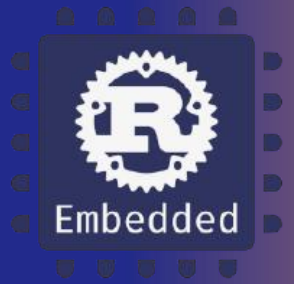
RUST CHINA CONF 2021 - 2022

第二届中国Rust开发者大会

2022.07.31 Online



嵌入式Rust与RustSBI固件设计

洛佳 / 华中科技大学

2022年7月

我是……

- 洛佳
- 华中科技大学网络空间安全学院
- 学习Rust时长约4年
- 主攻方向：软硬件协同防御
- 社交媒体账号：@luojia65
- 古琴爱好者

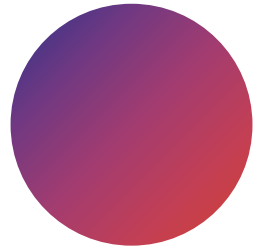
目录

嵌入式领域的Rust

作为用途广泛的内存安全语言，Rust将在物联网、能源等领域大放异彩。厂商、社区如何参与嵌入式Rust的开发体系？

Rust语言固件开发

固件事关RISC-V系统的安全、可靠，其稳定性不容闪失。该如何用Rust语言保证这一点？怎样落实Rust固件到具体芯片产品中？



生态圈中的RustSBI

RustSBI诞生于对RISC-V固件可移植和开放、易用性的呼声，也丰富了Oreboot等优秀的引导程序项目。RustSBI如何能与生态圈较好融合？

嵌入式领域的Rust语言

拓展Rust语言到嵌入式开发，是实践证明可行和实用的途径。社区、厂商积极参与后，嵌入式Rust将得到空前的发展。

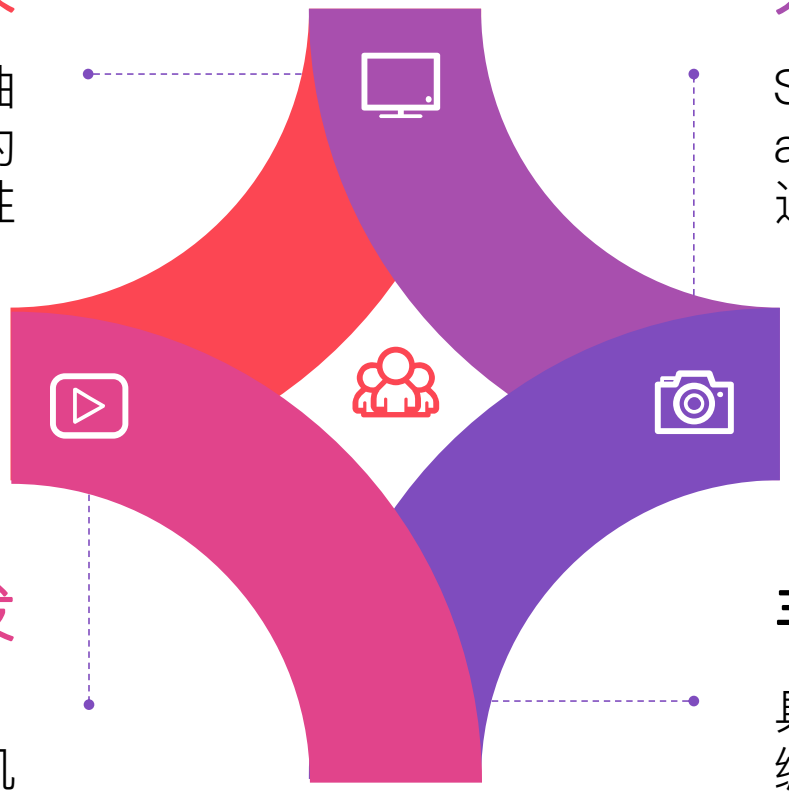
裸机上的Rust语言

丰富的软约束

所有权语义、借用系统，零开销抽象，trait接口、常量泛型，严格的编译期检查，降低人为出错可能性

无惧并发

Send、Sync接口，使用async/await的异步语义，海量的运行时和操作系统内核



灵活的模块化开发

基于模块和包的成熟开发模式，cargo包管理系统，与版本管理机制相容，可定义内部镜像

丰富开放的生态

具有嵌入式调试、烧录工具，裸机编译目标齐全，社区、厂家库支持完整多样，Rust for Linux

设备模型

固定地址模型

外设基地址固定，常用于无虚拟地址的嵌入式应用

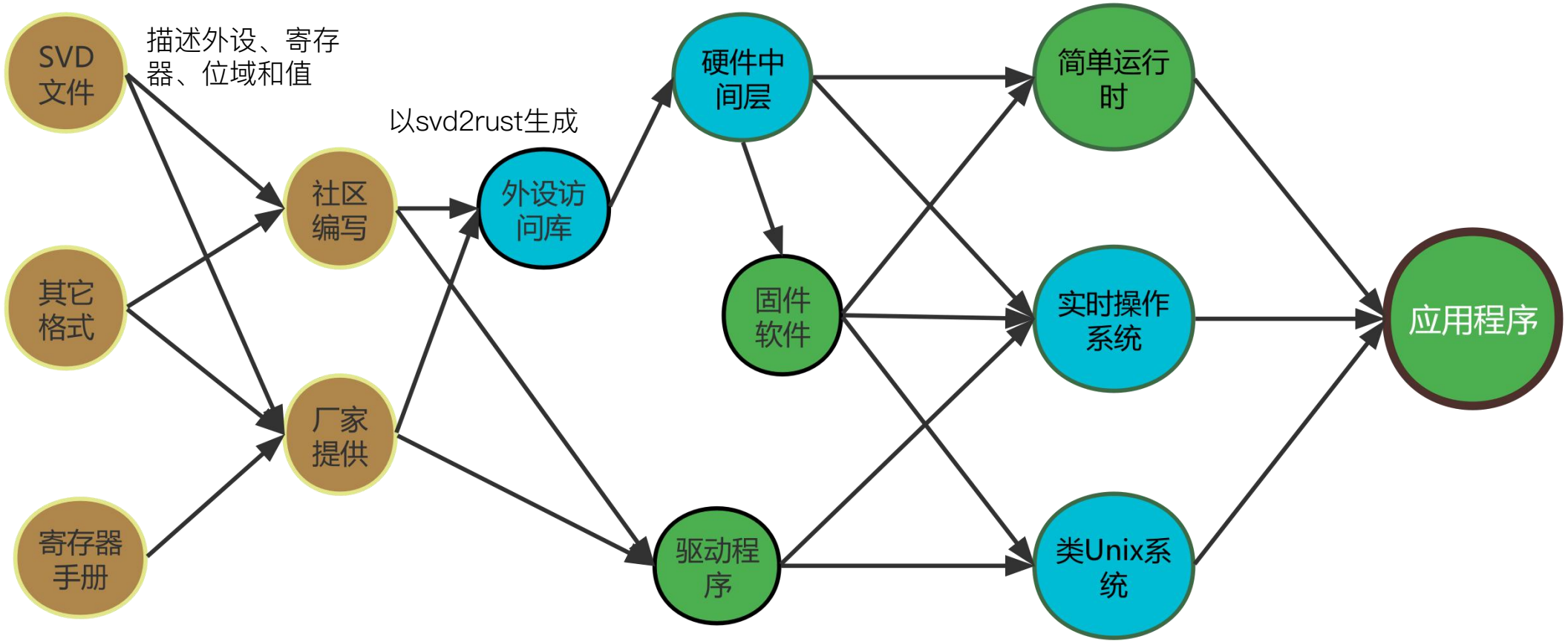
可使用“外设–中间层–应用”的模式以嵌入式Rust开发

可变地址模型

基地址可变，寄存器偏移固定，常用于操作系统内核

具体的设备模型灵活、多样，通常取决于操作系统设计

生成外设访问库（2022年）



embedded-hal

嵌入式外设的统一抽象

外设功能

描述和抽象外设，快速上手，无需阅读繁琐的文档和手册

零抽象开销

紧凑的嵌入式平台，又想要精巧的高级语法？完全没问题！

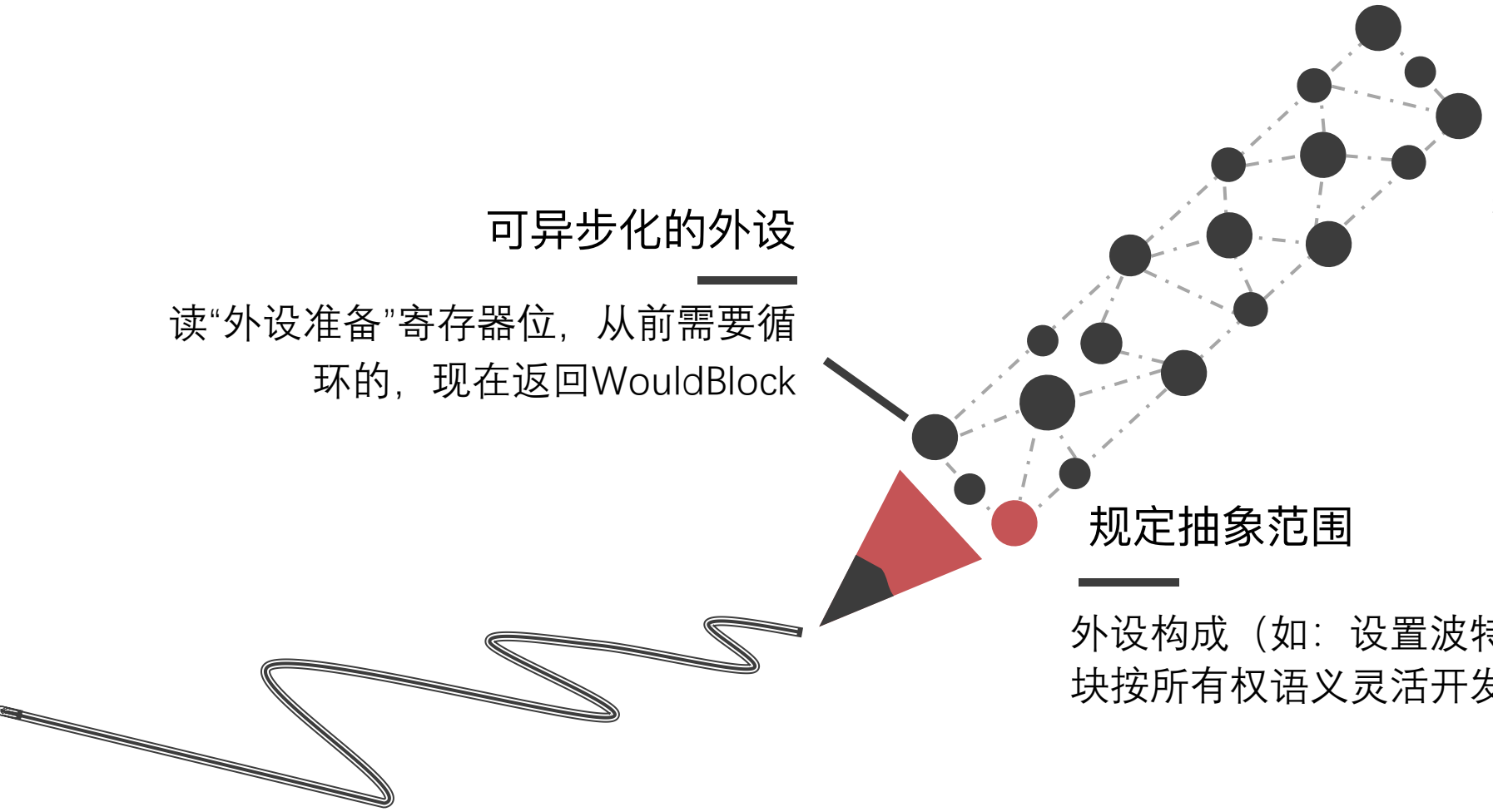
异步语义

与具有中断的外设良好结合，构成运行效率更高的嵌入式软件

级联外设

无需胶水代码，Rust语言的泛型和trait允许外设和文件系统、网络栈等模块联合

实现嵌入式硬件中间层



统一的抽象标准

按照embedded-hal规定编写，可复用、级联，便于与生态配合

可异步化的外设

读“外设准备”寄存器位，从前需要循环的，现在返回WouldBlock

规定抽象范围

外设构成（如：设置波特率）等模块按所有权语义灵活开发

编写运行时

所有的Rust软件必须运行在运行时上，嵌入式Rust也不例外

准备一个栈，即可构成嵌入式Rust中最简单的零运行时；而实时操作系统、复杂操作系统等都可看作丰富的运行时

Rust语言与固件开发

引导程序固件是安全性、可靠性要求更高的嵌入式应用，而Rust语言非常适合开发RISC-V平台下的固件产品。

用Rust编写的引导程序环境

Rust编写的引导程序

开机启动时，固件应被芯片内部机制加载。固件最终应当引导操作系统启动。嵌入式Rust可为精炼的引导程序做有益的尝试；

固件支持环境

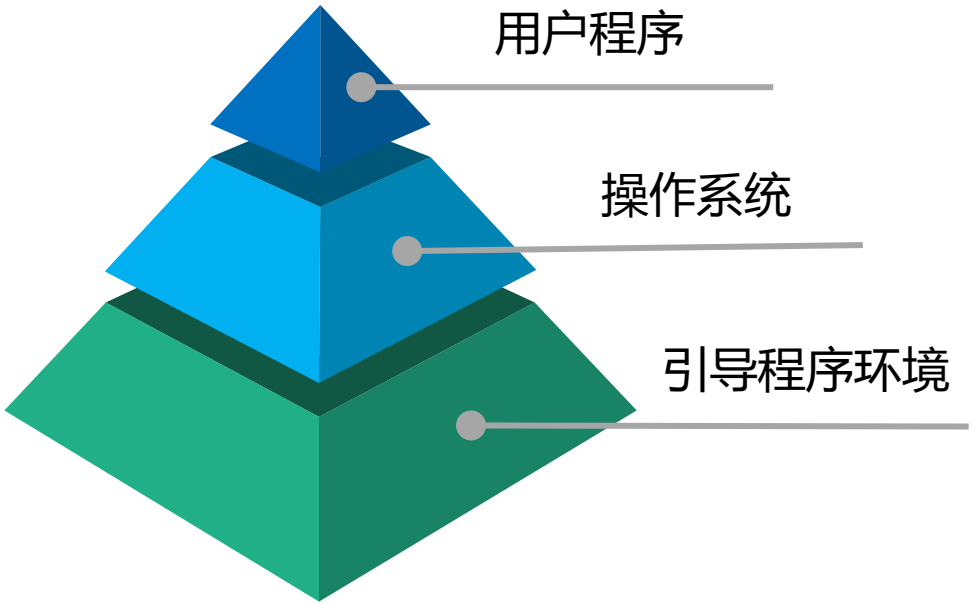
系统启动完成后，固件仍然保持在后台运行，持续提供内核需要的必要功能。维持安全稳定是Rust能提供的开发目标；

内核与固件专有功能

一些内核需要特定的接口和结构才能启动，这些专属的结构将由固件提供。Rust丰富的生态将简化此类功能的开发步骤。

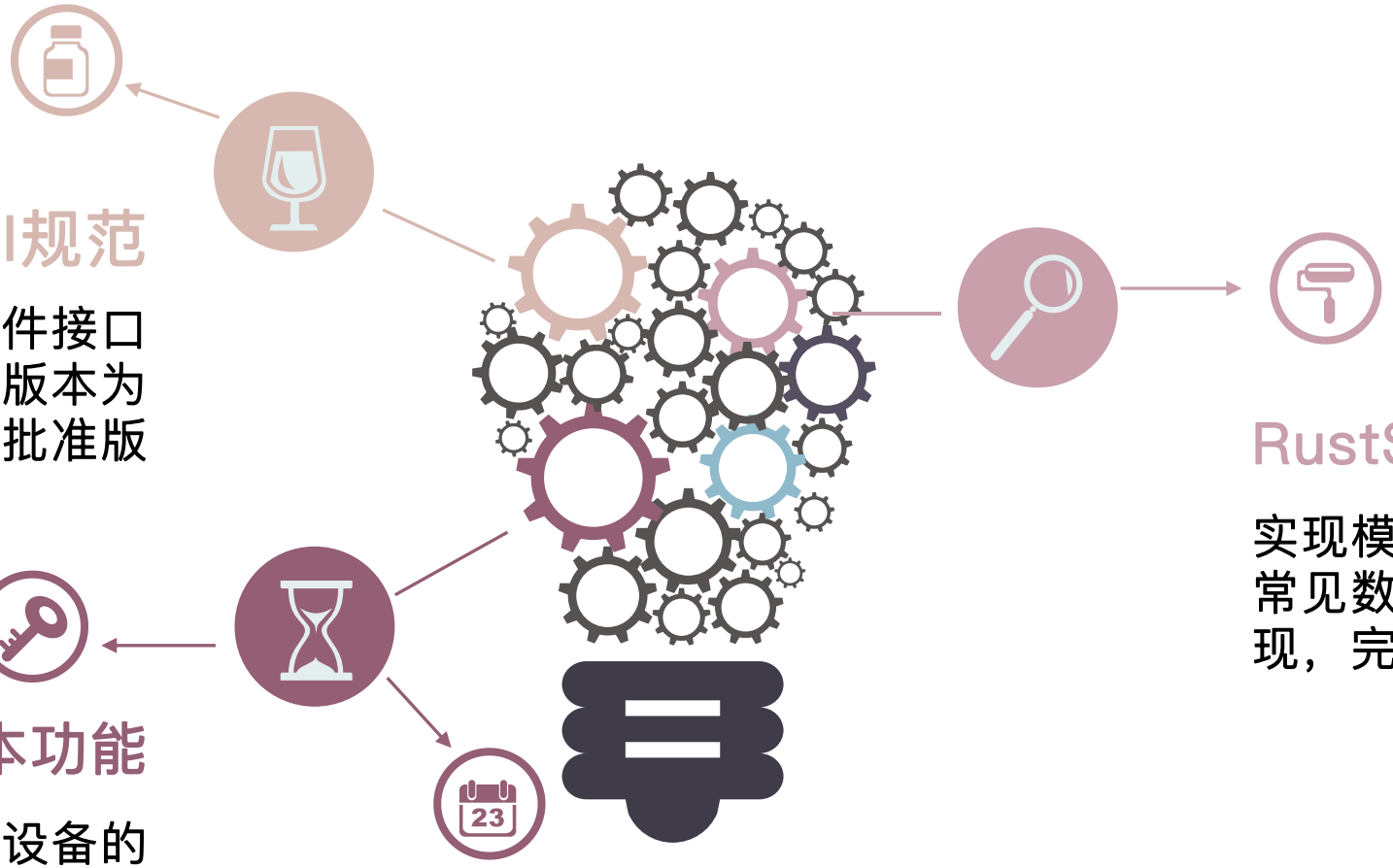
对比：不同架构的固件运行框架

RISC-V架构



其它传统架构

RISC-V的内核支持接口



SBI规范

RISC-V平台的固件接口规范标准，最新版本为v1.0.0正式批准版

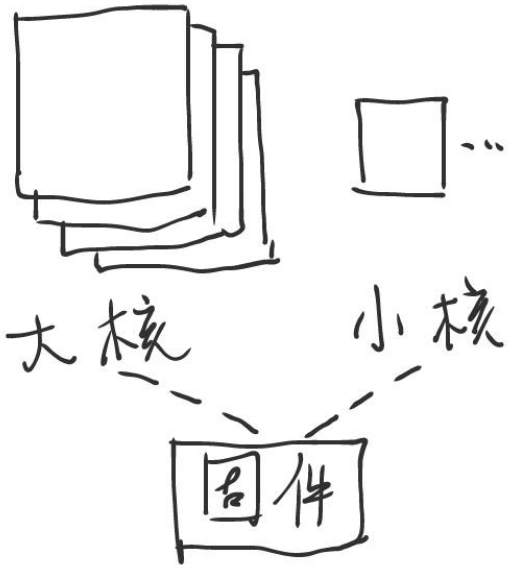
平台基本功能

高级功能和各个设备的抽象由其它标准完成

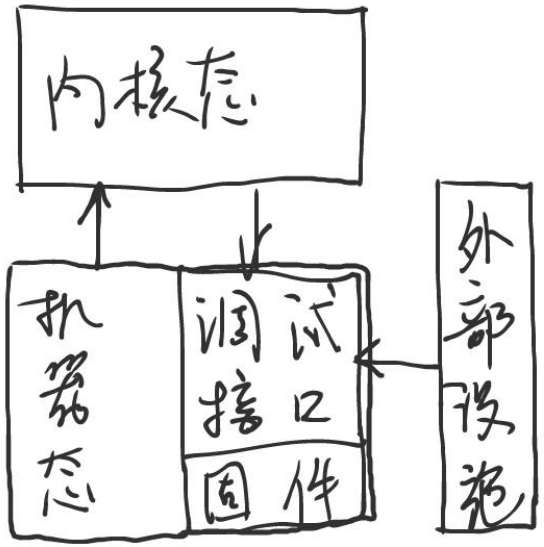
RustSBI

实现模型包括接口规范、常见数据结构、参考实现，完全使用Rust编写

用Rust实现高级固件功能

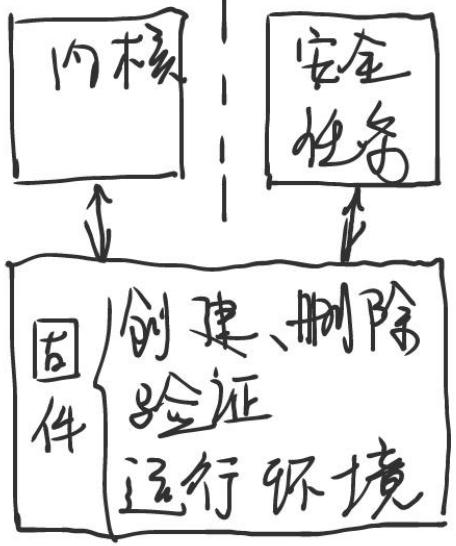


非对称多核处理器



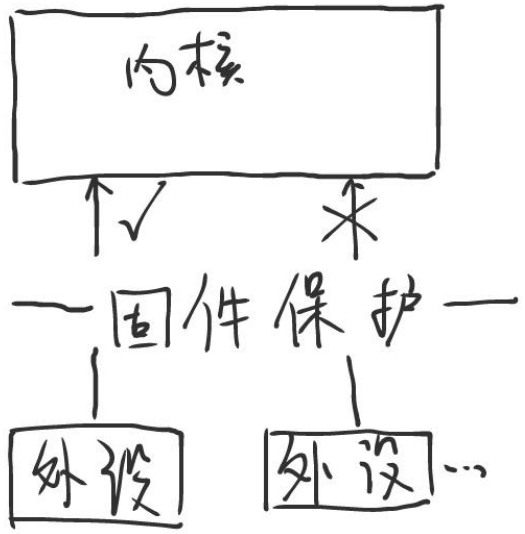
调试固件

如 { Raven
调试大师SBI



安全孤岛固件

如：蓬莱



固件内存保护

汤佳 2022.7.15

非对称多核处理器固件

- 技术难点：同时管理大小核、不同微架构间的同步操作
  - 如FU740下S7型管理核不支持DDR上的LR/SC，所用的同步数据结构必须用内联汇编自己编写
  - 不屏蔽管理核的用法：监视程序或TEE运行时等。可用HSM扩展等管理
  - 选择合适的运行时，如实时系统等，高可靠性下可做DDR失效处理程序
- 启动流程因芯片平台而异
  - 通常是小核先启动，但JH7100芯片是大核先启动
  - 必须适配ROM源码，建议芯片厂家在ROM中读取闪存erofs等文件系统，以魔术数为主并非良好的解决方案

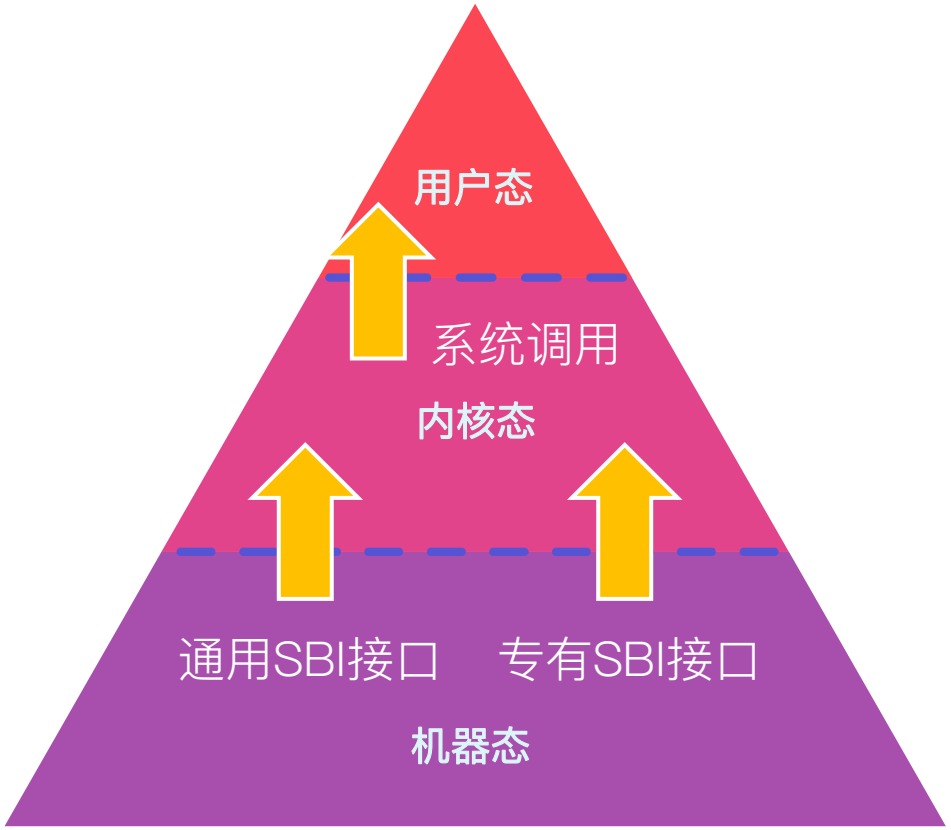
调试用固件

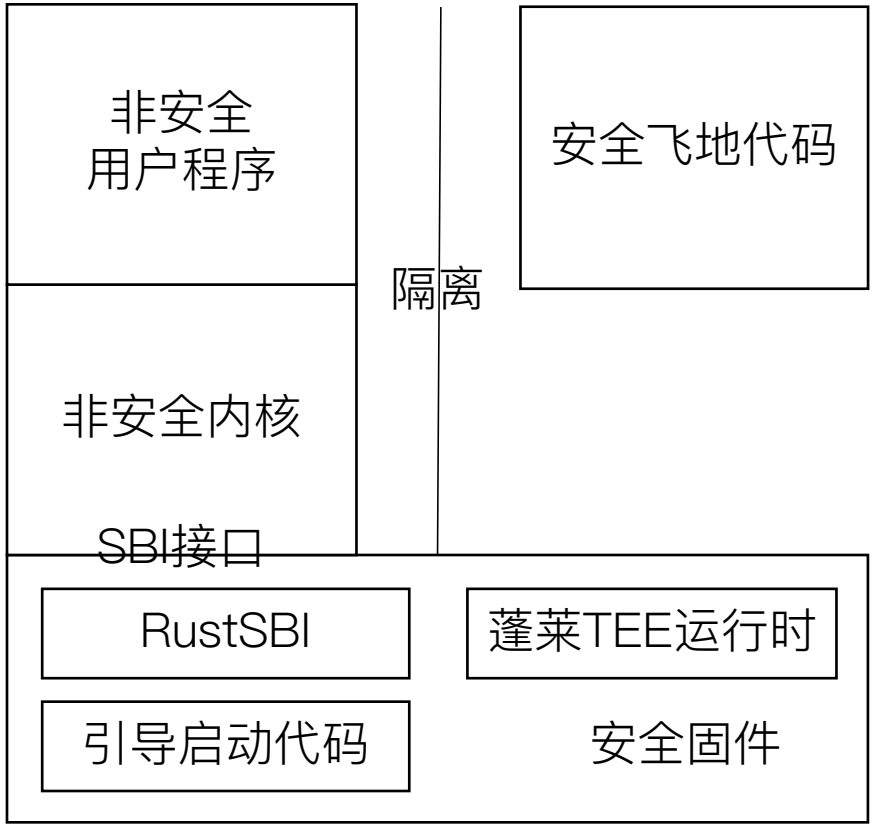
- 现有植入式内核调试接口与操作系统密切相关
  - RustSBI是RISC-V上运行于操作系统之下的环境，与操作系统无关
- 探索一种将其与现有内核调试机制共同运作的使用方法
- 调试大师SBI（https://github.com/luojia65/tiaoshi-dashi-sbi）。扩充这个程序或重新编写，完成无需了解具体操作系统而能调试内核的目的

安全孤岛固件：简介

- M态和S态是不同的处理器状态，可用于代码隔离
  - 在M态上运行固件，并向上暴露运行时接口，是良好的隔离工程方案
  - 暴露的接口可在SBI扩展空间中定义
- 实现机制有Penglai、Keystone等等
- 接口抽象的开发模式
  - RustSBI仅是SBI标准扩展的接口抽象。每次ecall判断扩展编号a7，当属于专有扩展时，由专有扩展模块处理，否则由RustSBI处理
- RustSBI希望能与Penglai社区在技术实现上展开广泛沟通

通用、专有SBI接口的并存方法





安全孤岛固件：常用技术技巧

- 读写特权层地址：如果参数数量太多，需要用指针形式传递……
  - 用mstatus.MPRV读取S态内存，可能出现缺页或权限异常
  - SBI标准规定：固件实现访问特权级内存时若发生缺页和权限异常，将回到特权级，并填写sepc寄存器为ECALL指令的地址
- 检测指令法：封装为函数，Rust语言探测指令常见技巧
  - 原理是临时切换中断处理函数
  - pub unsafe fn try_read<T>(src: SupervisorPointer<T>) –> Result<T, mcause::Exception>
- 专有的SBI扩展与标准SBI扩展共存
  - 建议保存到：供应者专有的SBI扩展空间，扩展编号从0x09000000到0x09FFFFFF

生态圈中的RustSBI

灵活运用嵌入式Rust，为具体的RISC-V平台、引导程序实现固件，支持运行于其上的桌面和服务器操作系统。

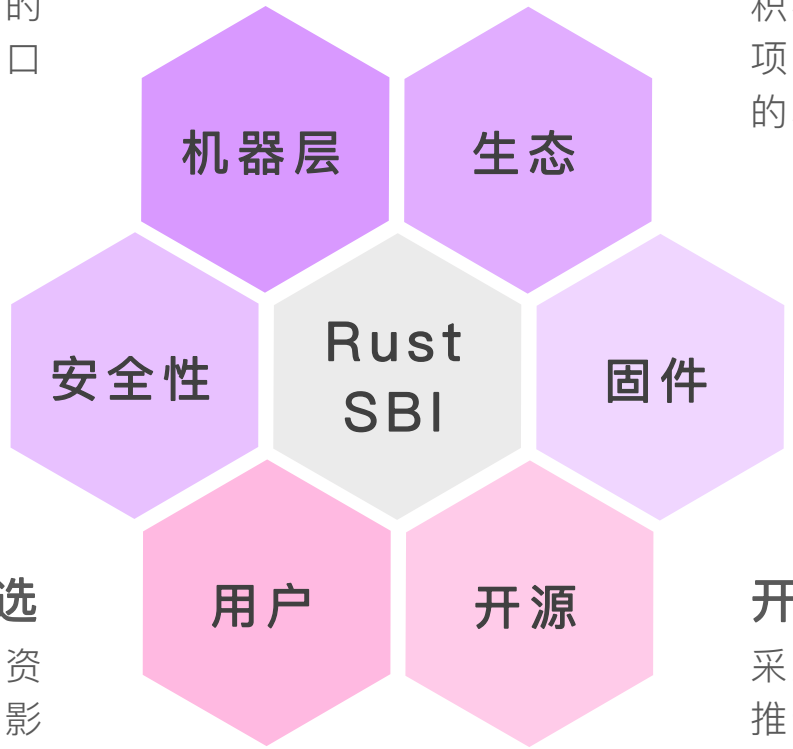
RustSBI是什么？

RISC-V机器层接口抽象

能够支持类Unix系统内核运行的RISC-V固件通常提供SBI接口

引导程序生态的参与者

积极参与Oreboot等大型引导程序项目，成为支持成熟引导解决方案的必需模块



安全稳定的运行时

SBI固件保持在后台运行，生命周期长于内核，安全性不容闪失

固件开发的参考指南

RustSBI和生态共同提供一系列固件实现，为自主可控固件提供参考

科研产业界固件开发首选

支持平台广、可定制性强，社区资源丰富，不受上下游非技术因素影响

开源开放固件的推动者

采纳MulanPSL-v2协议，致力于推动更多厂家开放硬件细节

例：全志D1 Oreboot引导zCore内核

为Oreboot适配D1平台，注意芯片系统的启动流程和具体板卡设计，不增加额外的启动阶段。

专有外设

C906的PLIC外设设计与通常的PLIC不同

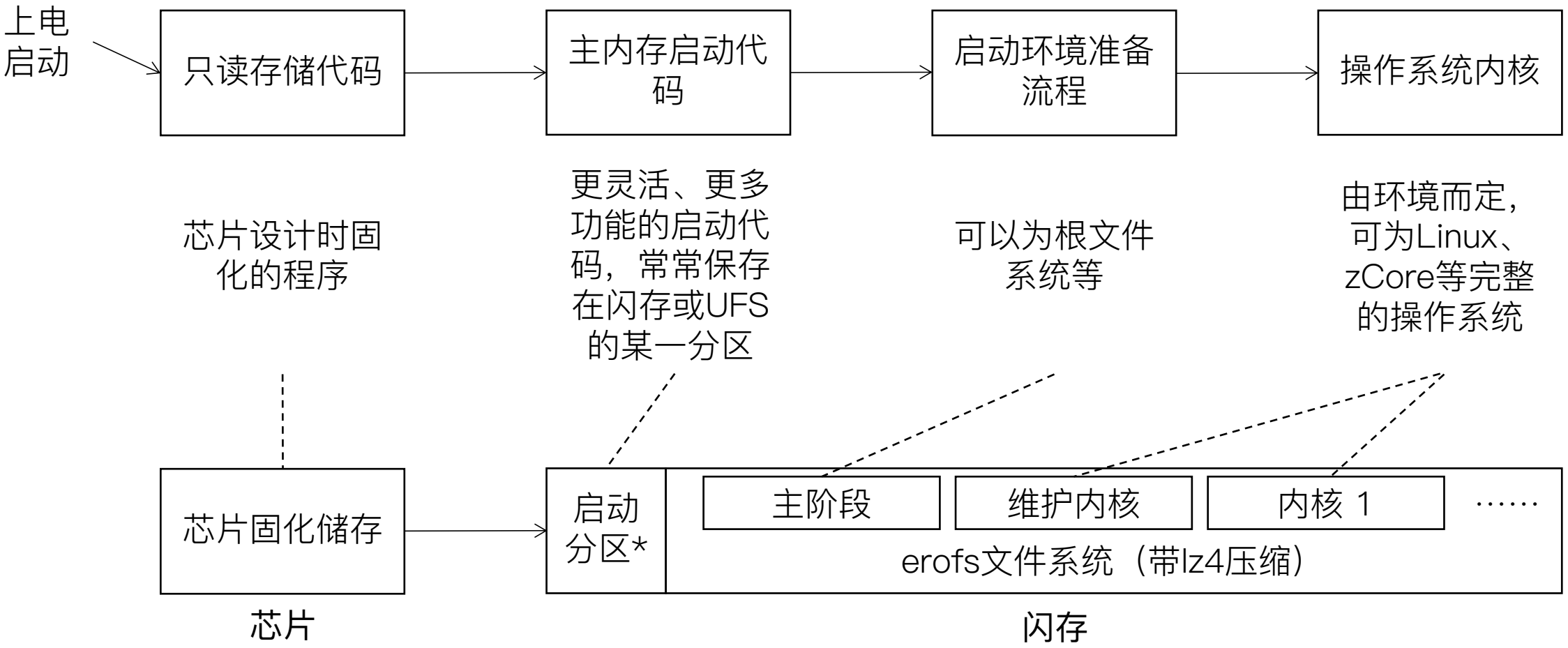
固件实现

严格按照RustSBI接口实现每个SBI模块

FEL固化代码

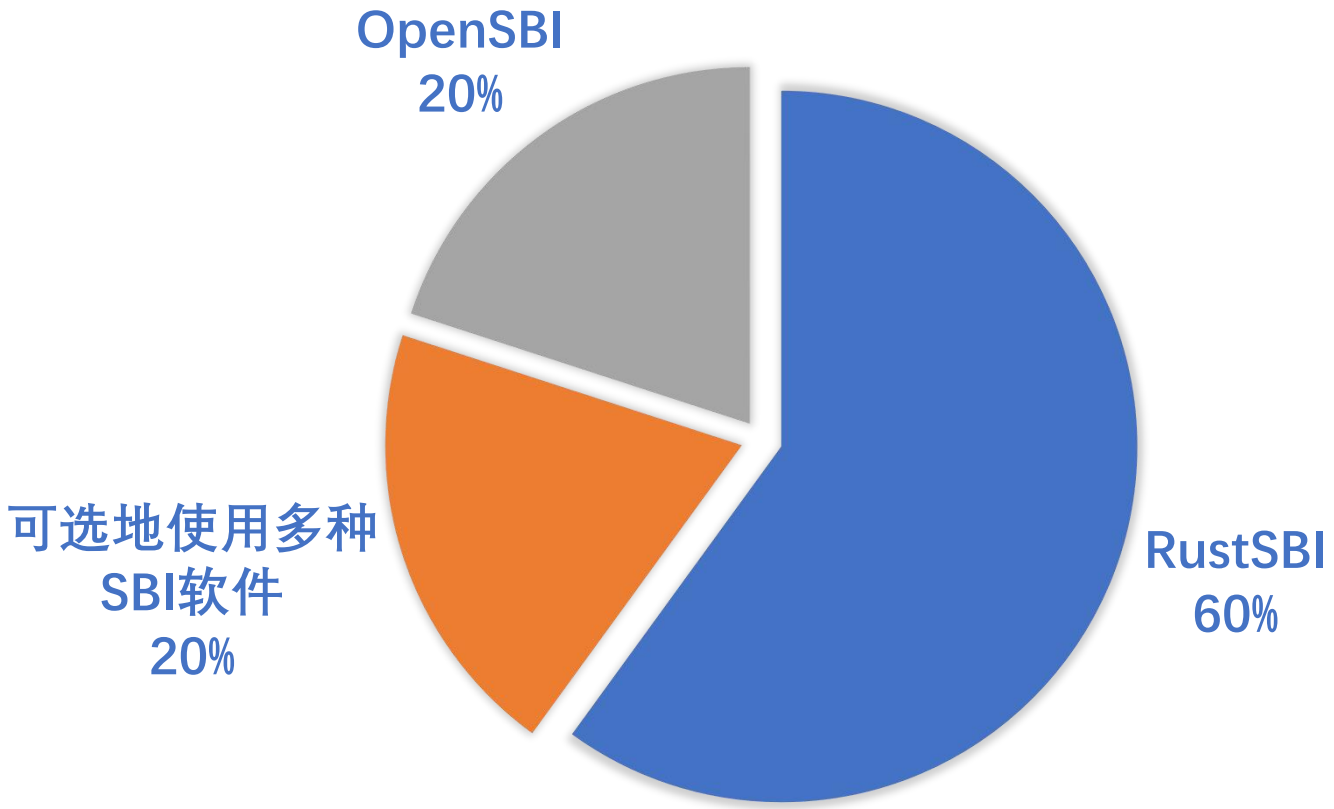
芯片固化代码可为调试提供便利

RISC-V固件引导的引导阶段



*也有的芯片不要求启动分区，而是能直接识读文件系统。

60%的国家一等奖赛队选择RustSBI



数据来源：整理自2021年全国大学生系统能力竞赛操作系统赛一等奖作品公开仓库（https://github.com/oscomp/2021oscomp-best-kernel-design-impl）

RustSBI是RISC-V SBI的官方标准实现

3.9. SBI Implementation IDs

Table 4. SBI Implementation IDs

Implementation ID	Name
0	Berkeley Boot Loader (BBL)
1	OpenSBI
2	Xvisor
3	KVM
4	RustSBI
5	Diosix

原始链接：https://github.com/riscv-non-isa/riscv-sbi-doc/blob/5b856c26dc6b12d5b15ee4f93651592803ac4c50/riscv-sbi.adoc#39-sbi-implementation-ids

RustSBI与其它SBI实现的功能对比

功能	RustSBI	以O开头的其它SBI实现
类Unix内核的运行环境	✓支持	✓支持
SBI 0.2版本IPI、TIMER功能	✓支持	✓支持
提供平台的设备描述	✓支持，通过灵活的serde设备树	✓支持，通过硬编码的设备树
跨平台编译和构建	✓支持，使用xtask框架	✓部分支持，需要配置环境
SBI 0.3版本HSM功能	✓支持，框架已定义	✗不支持
与Rust生态相容性	✓相容性良好	✗较难与Rust生态结合
向后兼容旧特权版本硬件	✓支持，以K210为例	✗不支持
启动管理核	✓支持，如Unmatched	✗不支持，会屏蔽管理核
扩展和定制高级功能	✓支持	✗不支持，较难合并到主分支

*通过代码和文档比较得到。RustSBI文档：https://github.com/rustsbi/rustsbi-hifive-unmatched/wiki。OpenSBI文档：https://github.com/riscv-software-src/opensbi。

欢迎使用RustSBI 0.3.0-alpha.1版本

- 完整支持RISC-V SBI 1.0.0正式批准版
- 两年运营经验，深受用户好评
- 完全使用Rust语言开发，安全高效
- 上下游固件生态完整
- 获厂家和社区支持

Rust
SBI

开发进程中为Rust生态做的贡献

- Oreboot引导程序（https://github.com/oreboot/oreboot）
  - 它可以启动固件中的LinuxBoot引导链，与仅有busybox的Linux固件结合（可实现维护功能），使用kexec启动真正的Linux内核
  - RustSBI是Oreboot支持RISC-V内核运行的接口实现
- 完善自旋提示函数（https://github.com/rust-lang/rust/pull/91548）
  - 添加对RISC-V平台的支持，由Zihintpause的PAUSE指令实现
  - 目前已经被合并到Rust语言标准库中并稳定，用于实现标准库的自旋锁
- 完善外设支持库生成器（https://github.com/rust-embedded-svd2rust/pull/627）
  - 生成枚举类型时正确导出枚举类型的名称，而不是由寄存器位域名决定

话语权

生态建设

RustSBI和竞争产品技术差异不大，速度、体积相仿，但架构更不制约上下游软件发展

软件抽象方法

RustSBI软件有意设计为实现与接口分离，利于可控软件建设

辅助系统运维

若内核、硬件支持受阻，或硬件版本不符，可采用固件模拟或虚拟化技术介入

骨牌效应

厂家、科研参与下，掌握定价权，提高固件领域影响力

RustSBI

致谢

- 感谢Rust中文社区和组委会提供演讲机会，Rust中文社区的嵌入式社区以及TUNA嵌入式社区提供的交流空间
- 感谢这段时间与我交流的社区伙伴，他们有：@YdrMaster，@dramforever，@双倍多多冰，@duskmoon314和更多朋友们
- 感谢丹尼尔Daniel Maslowski和Open Source Firmware社区在开源固件尤其是Oreboot上的贡献
- 感谢在我成长路上帮助和教导过我的所有老师们

Thanks

Rust China Conf 2021-2022 – Online, China