为RustSBI做开源贡献

蒋周奇

华中科技大学 网络空间安全学院

关于我

- 研一学生（系统安全-物联网安全-软硬件协同防御，导师：周威老师）
- 开源兴趣：RISC-V、固件和底层软件、物联网安全
- 做过的开源项目：RustSBI，BL-PAC，Coruscant，Nukkit
- 本科以来坚持开源贡献已5年左右

RustSBI是什么软件?

- RISC-V机器态固件实现
  - 特权架构模型“积木”的机器态部分
- 运行于RISC-V架构底层
  - 机器态为内核态提供环境调用服务
  - 提高安全性，并在后台支持内核的基本功能
- RustSBI原型设计系统
  - 包括启动、机器和内核三个部分
  - 内核态部分提供UEFI或LinuxBoot引导
  - 加速产品设计和选型
- 编程语言：100% Rust语言
- 市场份额50%左右，主项目星标约700+

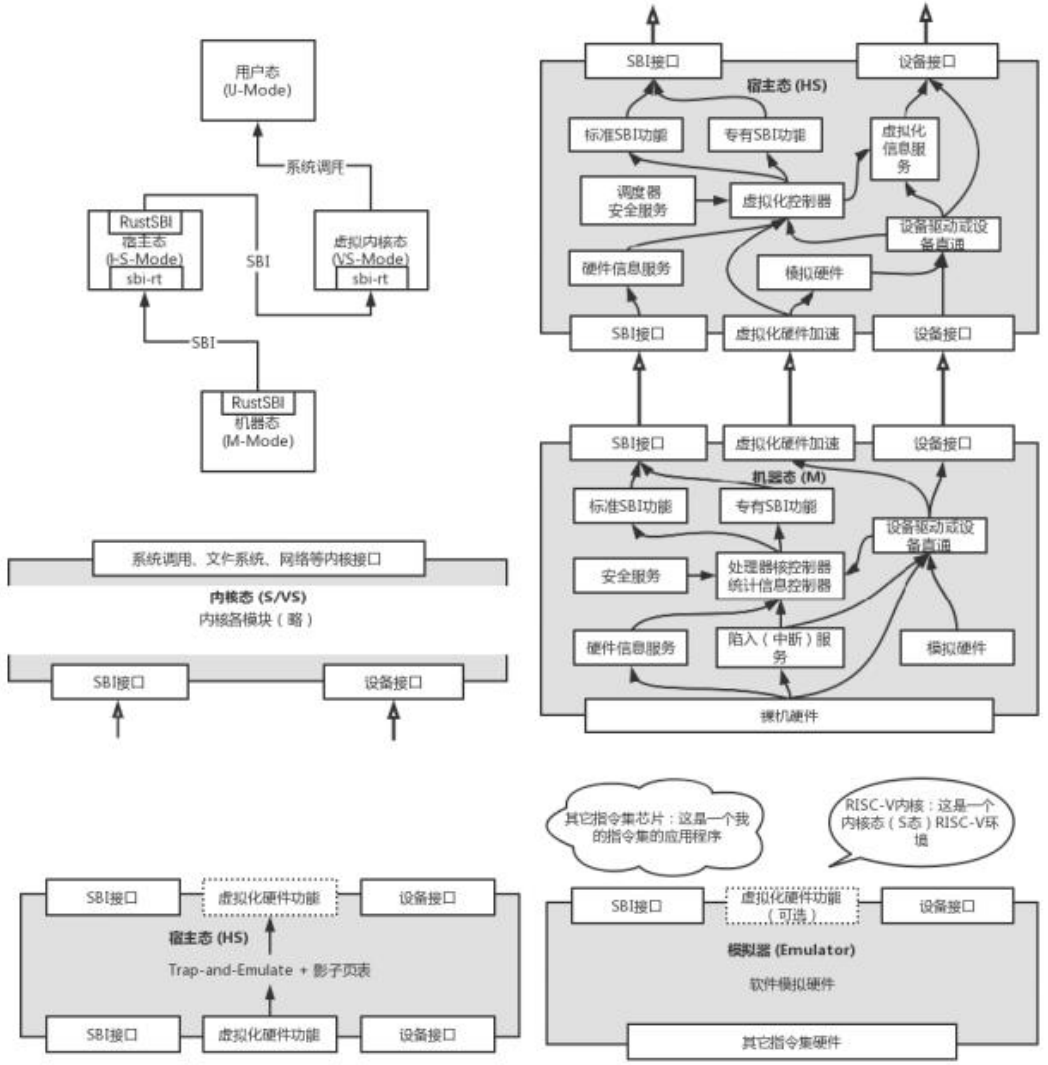


图 1　考虑虚拟化的 RISC-V 特权架构模型。a) 特权态架构，其中宿主态既是 SBI 的用户，又是 SBI 服务的提供者。b) 宿主态模块、机器态模块的逻辑框图，其中设备接口通常以 MMIO 形式体现。c) 模拟器模块，模拟器可以是其它架构上的应用程序，而其上的 RISC-V 内核以 RISC-V 指令集运行。d) 宿主态模块，其中若存在嵌套虚拟化，虚拟化硬件功能由软件模拟。

RustSBI社区

- 成立时间：2021年6月3日
- 目前成员数：十余人，来自各校科研团队和企业
- 管理项目：RustSBI主仓库、独立包仓库和原型设计系统、RustSBI教程
- 社区活跃的维护者会检查各仓库的合并请求
- 向Rustcc社区日报投新闻稿
- 社区间合作：主要有OSFW社区、Rustcc嵌入式社区和rCore社区
- 国际标准制定：RISC-V PRS会议

RustSBI 0.3.0正式版现已发布

洛佳 发表于 2022-11-04 00:42

Tags：嵌入式,RISC-V

RustSBI是RISC-V下SBI标准的实现，旨在为裸机平台、虚拟化和模拟器软件提供良好的SBI接口支持。它有机结合了Rust嵌入式生态与RISC-V系统软件，加快开发速度的同时，保证Rust语言具备的良好安全性和运行性能。本次0.3.0版本主要包括增加了实例化的SBI接口支持及相关的构造器结构，可以在stable Rust编译，去除了对堆内存和全局变量的依赖，完善了相关文档，以及若干的小修复。0.3.0版本更新将为Rust编写的RISC-V虚拟化软件和RISC-V模拟器提供良好的支持，并进一步完善裸机RISC-V开发的实用性，可以启动Linux等在内的成熟操作系统和zCore等在内的科研操作系统。

随着RustSBI 0.3.0正式版的发布，RustSBI的生态链项目趋于成熟，正在酝酿的“RustSBI原型设计系统”也在活跃开发中。内核运行工具sbi-rt、常数与结构包sbi-spec和规范测试集sbi-testing都已完成定型、发布预览版，并进入实际项目的依赖选项中。“RustSBI原型设计系统”并非专注于原型设计，而是提供一种快速开发的解决方案，开发完成后，它将允许厂家在最短的时间内适配SBI接口到自己的RISC-V主板和平台，并且直接获得蓬莱TEE、@dram的软件模拟虚拟化以及Raven固件调试器等高级功能。与此同时，贡献者和用户群体也反馈了对RustSBI及其新版本的评价。

活跃的社区贡献者@YdrMaster认为，RustSBI软件是社区力量在RISC-V SBI生态中的表现。“RustSBI帮助我探索‘内核之下（M态）’和‘内核之前（bootloader）’；相比OpenSBI，它的实现更简洁、干净，构建方式更现代，能提供更好的开发体验和操作空间”，YdrMaster说，“它除了具备所有Rust的优势之外，还具有库 + 实现的抽象，不必将所有实现塞进一个仓库，对一个硬件也有针对不同需求的不同实现。如果需要一个新实现，可以只重做关心的部分，复用其它部分。另外，它的运行速度快，在连续的内核测试时十分明显。”

长期贡献Oreboot项目的Daniel Maslowski说，RustSBI简化了完整引导程序的开发工作。“RustSBI是Rust生态中的SBI实现，它有助于记住RISC-V中（的SBI服务）需要什么，并且已经定义了所有的常量和结构”，丹尼尔说，“Rust是它特长的一方面，（在引导程序开发中）我不需要额外的组件或者代码库。这样，对于相当多的SoC，我们可以为固件提供单个的初始化阶段，只要它能够放入SRAM中，就像我为JH7100（128K）做得一样。”

UltraOS团队的@LoanCold认为，RustSBI就它为RISC-V SBI生态所做的贡献来说，它可以继续蓬勃发展下去，给开发者更多的选择空间。“我所参与的UltraOS团队用Rust实现撰写的操作系统，使用了RustSBI项目。从项目来说，更好的开发者支持以及更强大的K210开发板支持，是我受益的最大部分”，LoanCold说，“我们团队也自身更改过RustSBI以实现更好的功能，这是开源或者进一步开源带来的好处，或者说RustSBI较为完备的注释带来的好处。它同时使得我们能够更好地支持K210平台的开发，这是OpenSBI所不能做到的。未来的RustSBI可以做到垂直整合，吸引稳定的使用者，完善平台支持和自动化测试，来保障系统级别的应用长期稳定运行。”

“今年相比过去的两年，RustSBI生态和用户在进一步扩大。除了科研和教学界，我们乐于见到更多产业界的公司贡献到RustSBI生态中”，洛佳说，“BL808的官方Rust支持库就是一个好的开始。大小核支持、虚拟化和模拟器支持以及安全特性，这些都是RustSBI擅长的部分。无论用户选择创新的全栈Rust实现还是兼顾U-Boot、UEFI或者EDK II等传统软件的实现，RustSBI都可以良好地支持和配合产业软件的发展。在我们应用于模拟器的性能测试中，RustSBI体现出非凡的性能，部分性能指标达到了竞争对手的20至30倍。我们希望将RustSBI卓越的特点分享给所有的引导程序软件，无论是C或者Rust都可以——生态的参与者能够一起合作，共同提高引导程序产业的安全和稳定性。”

本次更新的主要贡献者有@duskmoon314，@OrangeCMS，@YdrMaster和@luojia65。

RustSBI的主要子项目

- RustSBI主仓库（https://github.com/rustsbi/rustsbi）
  - 提供RISC-V SBI与具体硬件平台无关的功能，紧随RISC-V SBI标准更新而更新，并参与标准制定工作
- RustSBI原型设计系统（https://github.com/rustsbi/standalone，*快速迭代项目*）
  - 为主板研发者快速选型引导程序而设计
  - 提供从机器态到内核态UEFI、LinuxBoot的完整解决方案，支持多个厂家主板，提供快速的移植方式
- RustSBI独立包支持（rustsbi-d1、rustsbi-qemu和rustsbi-k210等项目）
  - 从前RustSBI的主力发展方向，目前逐渐整合到RustSBI原型设计系统中
- RustSBI教程（https://github.com/rustsbi/rustsbi-tutorial）
  - 为教学目的设计，从零实现一个类似于RustSBI的机器态固件，作为大学操作系统课程的补充
- SBI生态有关仓库：sbi-spec规范常量和类型、sbi-rt运行环境和sbi-testing测试集
- 其它仓库：外设组件化支持库、ROM运行环境库和嵌入式开发基础库等

贡献须知

- Git贡献、软件代码协议和代码格式
  - 使用MulanPSL-v2.0和MIT双协议
  - 每个Git commit要求Signed-off-by签名
  - Rust代码要求使用cargo fmt默认配置的代码格式
  - 要求每个Git commit都能够编译，否则应当squash commits（使用git rebase -i）
- 代码测试风格
  - 每个项目细节上不同。在真板上运行的项目，必须编写测试用例，测试通过后方可合并。
- 暂时无需签订贡献者协议
- 合并请求流程：Pull Request提交代码→维护者检查→代码合并
  - 若维护者建议修改代码，修改后应当squash commits；无需新建pull request。
  - 合并请求的响应时间：常规项目工作日5天以内，休息日7天以内。快速迭代项目工作日48小时以内，休息日72小时以内。

版本发布流程与版本兼容性

- RustSBI主项目：跟随RISC-V SBI发布而发布，稳定为主
  - 大版本大致为每年一次，小版本随着漏洞修复而更新
- RustSBI原型设计系统
  - 目前尚未达到发布第一个版本的技术要求
  - 内核态运行成功后，发布0.0.0版本。未来（初步规定）6个月一次大版本，每个漏洞一次小版本
- 跟随Cargo包管理器的版本兼容性要求
  - 0.0.z版本允许破坏性更新，0.y.z版本破坏性更新必须增加y，x.y.z版本破坏性更新必须增加x
- 版本发布流程
  - 梳理贡献者清单、修改清单（changelog）
  - crates.io版本发布，git网页新版本发布
  - 主项目和原型设计系统的大版本发布：原则上应当编写新闻稿

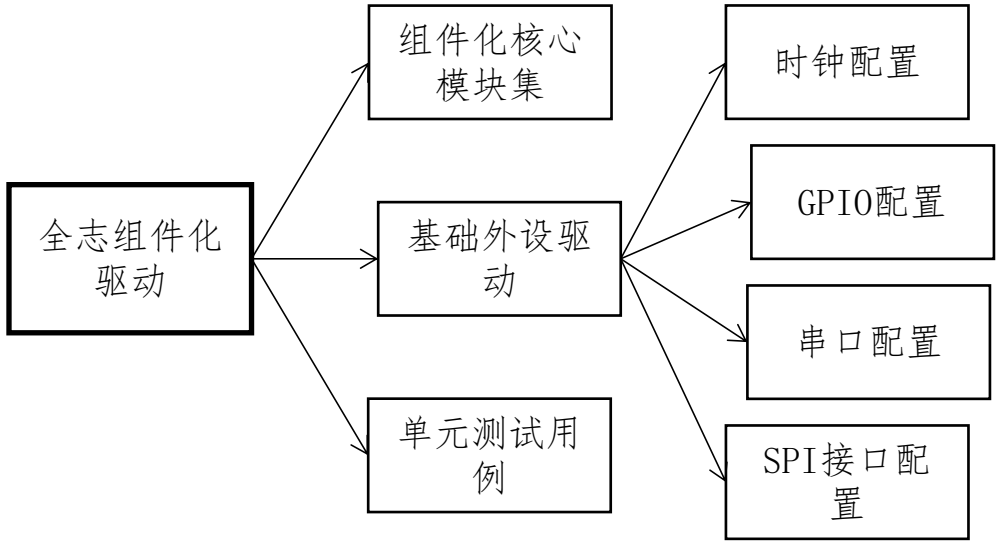
原型设计系统仍然需要的初步工作

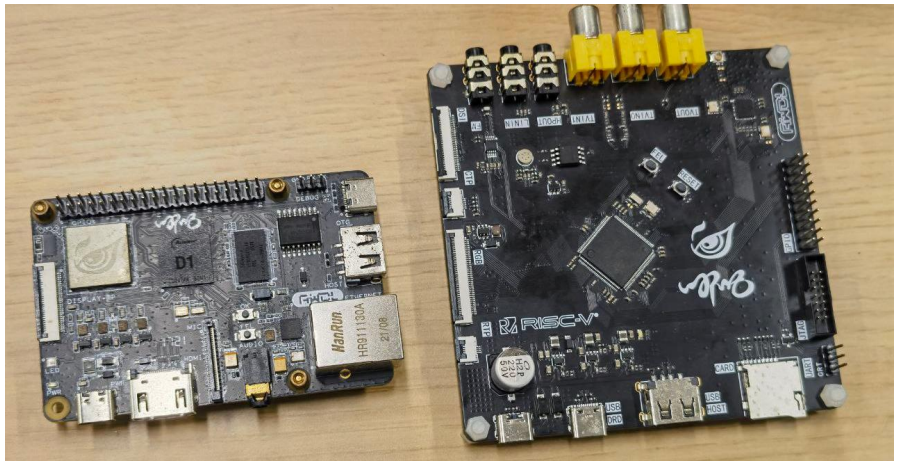
- 组件化外设驱动
- 各个芯片的ROM启动流程
- 机器态SBI基础功能
- 应用内详细的帮助文档

- 注：较复杂的工作在初步工作之后逐渐开展

全志芯片组件化基础外设驱动（aw-soc项目）

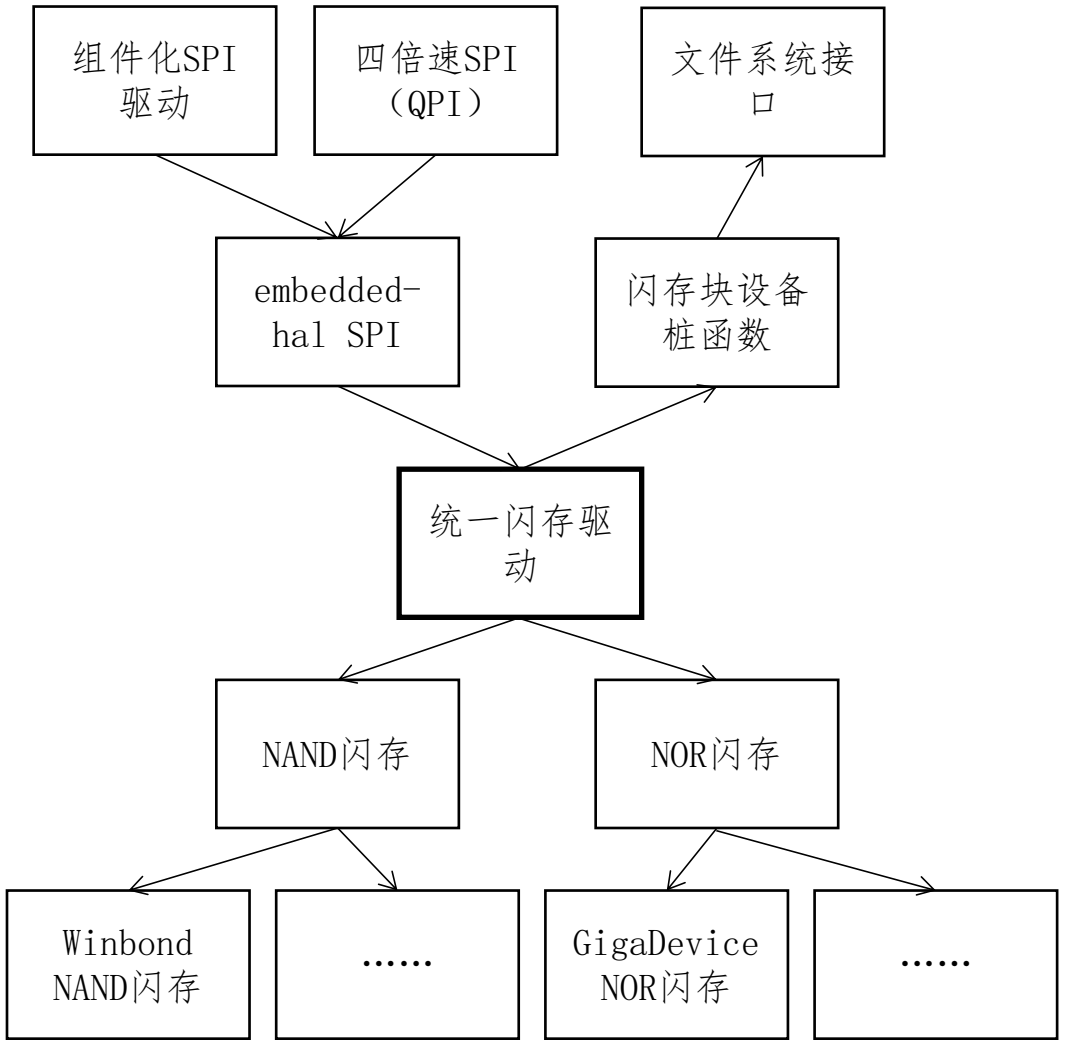
- 组件化驱动是什么？
  - 清华大学rCore团队提出“组件化操作系统”的一部分
  - 利用21世纪编程语言理论的特点，构造动静态一体、高效和灵活兼具的外设驱动
  - 编写一次就能同时运用于嵌入式、固件和操作系统内核
- 全志芯片简介
  - 全志D1是RISC-V架构的嵌入式芯片
  - 全志全系列芯片的外设大致相同，写一个项目就能复用在未来所有的全志RISC-V芯片上
- 项目目标：完成全志芯片基础的组件化外设驱动





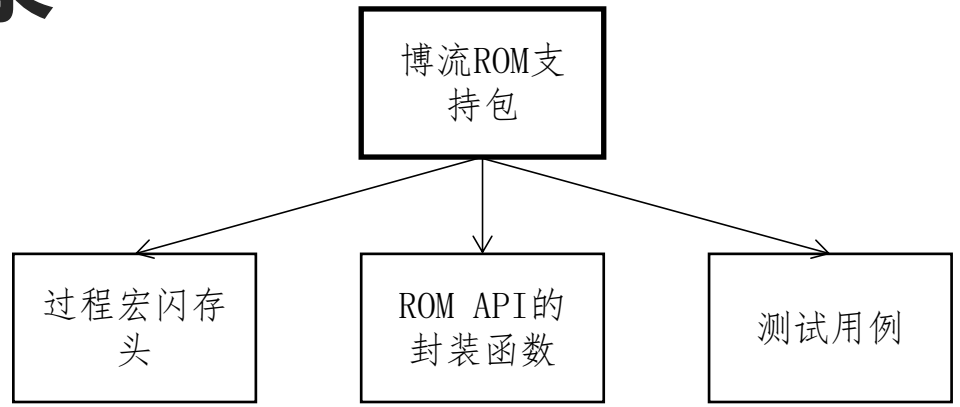
统一闪存驱动

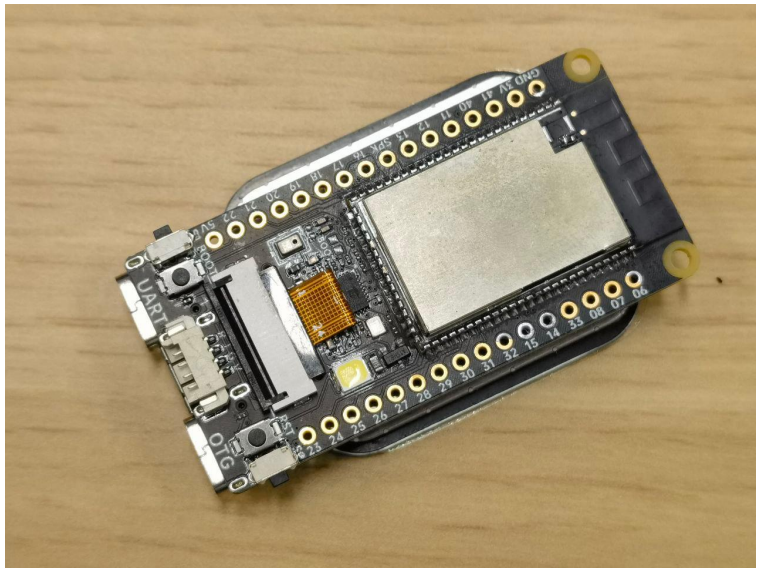
- 为什么需要统一闪存驱动？
  - 嵌入式Rust提供了功能抽象的SPI接口，而大多数NAND和NOR闪存都通过SPI读写数据
  - 在配置妥当的前提下通过SPI接口统一完成闪存的基本操作
  - 用户无需过多了解厂家品牌闪存的细节，就可完成闪存读写和数据预取等基本操作
- 根据妥善配置的SPI功能抽象接口，提供NAND和NOR闪存的读写操作
- 适配各个厂家的NAND和NOR闪存
- 统一闪存驱动对引导程序开发有较大的帮助



博流芯片的ROM启动和烧录

- 过程宏闪存头是什么?
  - 嵌入式Rust常见的函数包装方法，使用过程宏封装main函数，直接得到ROM可识别的闪存头
- ROM支持包的主要功能
  - 提供可包装闪存头的过程宏
  - 操作ROM API内的程序功能
- ROM支持包必须和具体芯片的ROM代码适配后才可烧录使用
- 博流芯片BL808（三核异构RISC-V芯片）
- 阅读ROM手册，完成可直接烧录到闪存的过程宏闪存头，并封装可使用的ROM API





致谢

- 感谢学院、老师和同学们
- 感谢一路支持RustSBI项目的贡献者们，包括@YdrMaster、@duskmoon和@OrangeCMS

开源愉快！

为RustSBI社区做开源贡献

华中科技大学 网络空间安全学院 蒋周奇