

## 毕业设计（论文）开题报告

**题 目：适用于Kendryte芯片的组件化系统驱动解决方案**

学 院： 信息工程学院

专业名称： 计算机科学与技术

班级学号： 21045334

学生姓名： 郑龙兵

指导教师： 胡镜林

填表日期 2024 年 12 月 25 日

1. **选题的依据及意义**

在信息技术飞速发展的时代，嵌入式系统已经深入到人们生活的各个角落，包括人工智能和物联网等新兴领域。随着边缘计算、AI芯片等技术的不断发展，嵌入式系统的数据处理需求、开发效率要求激增，芯片硬件设计日趋复杂和多样，传统的嵌入式系统驱动开发方式已经无法满足现代嵌入式市场的需求。因此，本次工作选用了典型的AI异构计算系列芯片Kendryte，开发基于Rust语言的Kendryte芯片组件化系统驱动解决方案；不仅符合技术发展趋势，也满足了市场和社会对高效、安全、可维护的嵌入式系统的需求。

随着物联网和人工智能技术的快速发展，像Kendryte这样的AI芯片已经成为连接智能设备与云端的重要桥梁。传统的嵌入式系统驱动开发方式，如C语言的单片机编程，已经不能满足现代复杂SoC的需求。而基于Rust语言的组件化系统驱动，以其内存安全、并发安全、模块化程度高的特点，正逐渐成为主流。首先，从技术需求来看，随着AI芯片复杂度的不断提高，驱动开发难度和维护成本也在不断增加。同时，嵌入式系统也面临着安全性、可靠性、实时性等挑战。因此，一个能够高效管理复杂硬件资源、保证系统安全性的组件化驱动系统，显得尤为重要。其次，从技术角度来看，随着Rust语言在系统编程领域的快速发展，嵌入式系统驱动开发具备了更多可能性。例如，利用Rust的所有权系统，可以在编译时就避免大部分内存安全问题[1]；通过trait系统，可以实现更灵活的抽象和组件化设计；利用宏系统，可以减少重复代码，提高开发效率。

此外，组件化系统驱动还能为嵌入式开发者提供更加模块化、可重用的开发环境。抽象硬件接口、组件化设计等功能，减少了重复开发工作，降低了开发者的学习成本，同时也为硬件厂商提供了更多的软件生态支持空间。

综上所述，Kendryte芯片组件化系统驱动解决方案的选题具有重要的现实意义和深远的技术影响。它不仅能够满足现代AI芯片和嵌入式开发者的需求，提高开发效率，还能为嵌入式系统构建更安全、更可靠的软件环境，推动嵌入式技术的健康发展。因此，开发一个基于Rust语言的组件化系统驱动解决方案，是一个具有广阔前景和发展空间的选题。

本课题采用Rust语言作为主要开发语言，结合嵌入式系统特有的开发工具链和调试环境。Rust作为一种现代化的系统编程语言，具有内存安全、并发安全、零成本抽象等优点，广泛应用于系统级软件开发。而组件化设计方法论作为软件工程中的重要概念，具有高内聚、低耦合、可重用性强等特点，适用于构建复杂的嵌入式系统驱动,其广泛应用在飞行器控制系统[3]，航电系统[4]等领域。将Rust语言和组件化设计结合使用，可以充分发挥两者在各自领域的优势，构建出安全、高效、易维护的嵌入式系统驱动框架。Rust和组件化设计的组合具有良好的扩展性和可维护性，支持模块化开发、静态类型检查、泛型编程等功能，便于后期维护和升级。Rust提供了强大的并发安全机制和内存管理模型，结合组件化设计的模块化特性，可以确保系统的安全性和稳定性。基于Rust和组件化设计的Kendryte芯片驱动系统的开发依据主要包括语言特性的优势、设计模式的适用性、系统需求与性能要求、可扩展性与可维护性以及安全性与稳定性等方面的考虑。这些因素共同构成了系统开发的基础和支撑，为构建一个高效、稳定、安全的嵌入式系统驱动框架提供了有力的保障。

**二、国内外研究概况及发展趋势**

随着嵌入式技术和人工智能的快速发展，像Kendryte这样的AI芯片正逐渐成为物联网和边缘计算领域的重要组成部分。在这一发展过程中，基于Rust语言的组件化系统驱动解决方案凭借其内存安全、并发安全和模块化的特点，受到了广泛的关注和应用。本文将对基于Rust的Kendryte芯片组件化系统驱动解决方案的国内外研究概况及发展趋势进行深入探讨。

在国外，基于Rust的嵌入式系统驱动研究与应用取得了显著成果。许多开源项目和商业产品例如STM32[5][8]、树莓派[6][7]等成熟产品系列，纷纷采用Rust语言进行嵌入式系统开发，不仅通过引入形式化验证和静态分析技术实现了更高的系统可靠性和安全性，还注重优化资源利用和实时性能，极大提升了嵌入式系统的效率和稳定性。同时，国外的研究机构和高校也在积极探索Rust在嵌入式领域的应用，如美国斯坦福大学的Tock操作系统项目和瑞典皇家理工学院的Rust嵌入式开发课程，为组件化系统驱动的发展提供了坚实的理论基础。基于Rust的嵌入式系统驱动将呈现出更多创新趋势。零成本抽象和泛型编程的深入应用将成为系统发展的重要方向，通过更灵活的组件设计和代码复用，提升开发效率和系统性能。同时，随着AI芯片的普及和发展，系统对异构计算资源的管理和调度将受到更多关注，以满足复杂AI应用的需求。另外，系统的安全性和可靠性始终是研究的重点，通过利用Rust的所有权系统和借用检查器，确保内存安全和并发安全。

在国内，基于Rust的嵌入式系统驱动研究虽然起步较晚，但发展迅速，已经取得了一些重要成果。国内的芯片厂商如平头哥半导体已经开始探索使用Rust语言开发RISC-V架构的嵌入式系统；在SoC芯片方面，先楫、乐鑫和博流等芯片品牌提供了官方的Rust embedded-hal支持包实现。在学术研究方面，清华大学的rCore操作系统教学项目为Rust在操作系统和嵌入式领域的应用提供了宝贵的经验。同时，国内的一些初创公司也开始在物联网和边缘计算领域应用Rust语言，为基于Rust的组件化系统驱动的发展提供了实践基础。在技术研究方面，国内的开发者和研究机构正在积极探索Rust语言在嵌入式系统中的应用。他们不断尝试将Rust的优势特性如trait系统和宏系统应用到驱动开发中，为基于Rust的组件化系统驱动的开发提供了更多的可能性。在市场应用方面，虽然目前大规模商业应用还相对较少，但已有一些前沿项目开始采用Rust进行嵌入式系统开发。这些项目通过引入先进的编程范式和创新的系统架构，为嵌入式系统提供了更加安全、高效、可维护的软件解决方案。

未来，基于Rust的Kendryte芯片组件化系统驱动解决方案将面临更多机遇和挑战。首先，随着AI技术在边缘设备上的广泛应用，如何在驱动层面有效管理和优化AI加速器将成为重要研究方向。其次，实时性能优化将继续受到关注，特别是在需要严格时间保证的应用场景中。再次，跨平台和可移植性将成为重要议题，如何设计一个既能充分利用Kendryte芯片特性，又具有良好可移植性的驱动框架将是一个挑战。最后，随着物联网安全问题日益突出，利用Rust的安全特性构建更安全的嵌入式系统将成为未来研究的重点方向。

总的来说，基于Rust的Kendryte芯片组件化系统驱动解决方案代表了嵌入式系统开发的一个重要发展方向。它不仅能够提供更高的系统安全性和可靠性，还能通过组件化设计提高开发效率和代码复用率。随着技术的不断进步和应用场景的拓展，这一解决方案必将在物联网、边缘计算和人工智能等领域发挥越来越重要的作用。

**三、研究内容及实验方案**

1. **研究内容**

研究Kendryte K230芯片架构并阅读相关技术文档，使用Rust编程语言和汇编语言，设计并实现一个组件化系统驱动框架。该框架应支持K230芯片的启动、初始化和主要功能，包括系统底层初始化、外设驱动、系统服务和中间件等模块。具体研究内容描述如下：

（1）研究Kendryte K230芯片架构和特性，深入分析其启动流程和固件格式。

（2）研究TF卡镜像文件格式，设计并实现自动化固件生成工具。

（3）研究RISC-V汇编语言和Rust语言在嵌入式系统中的应用，特别是在系统底层初始化过程中的作用。

（4）研究K230芯片外设寄存器的访问方法，设计并实现符合embedded-hal标准的硬件抽象层。

（5）研究嵌入式系统服务和中间件的设计原则，探索RustSBI在K230芯片上的集成方法。

1. **实验方案**

本课题采用的开发环境为Visual Studio Code配合Rust插件，使用CanMV-K230-V1.0开发板与CKLINK-LITE调试器进行硬件调试。开发使用的Rust工具链版本为rustc stable-1.83.0，RISC-V GNU工具链用于汇编代码编译。具体开发步骤如下：

（1）分析Kendryte K230启动流程和固件格式：

* 研究K230技术手册，理解芯片启动过程
* 分析现有固件文件结构，确定各个段的作用和排列顺序
* 绘制启动流程图，明确从上电到操作系统加载的完整过程

（2）使用Rust xtask完成TF卡镜像.img固件文件的创建：

* 分析固件文件结构，包括引导加载程序、主程序和配置信息等
* 实现自动打包各个组件，生成符合K230要求的.img文件
* 添加校验和加密功能，确保固件的完整性和安全性

（3）编写汇编和Rust代码实现系统底层初始化：

* 使用RISC-V汇编语言编写启动代码，设置栈指针和全局指针
* 实现异常向量表和中断处理程序的基本框架
* 使用Rust编写运行时初始化代码，包括.bss段清零和.data段复制
* 实现内存管理的基本功能，为Rust运行时环境做准备

（4）手动编写外设寄存器访问代码，完成embedded-hal traits的实现：

* 研究K230芯片手册，明确各外设寄存器的地址和功能
* 使用Rust的volatile\_register包中的RW、RO和WO等结构实现底层寄存器读写函数
* 为常用外设（如UART、GPIO、I2C、SPI等）实现驱动程序

（5）开发系统服务组件和中间件，集成RustSBI：

* 实现基本的RISC-V SBI系统服务，如中断管理、时钟管理和电源管理
* 移植RustSBI到K230平台，实现SBI标准接口

（6）测试和优化：

* 使用CKLINK-LITE调试器在CanMV-K230-V1.0开发板上进行硬件测试
* 编写单元测试和集成测试，验证各个组件的功能
* 使用Rust的性能分析工具优化关键路径代码

进行功耗和实时性能测试，确保系统满足嵌入式应用要求

（7）文档编写：

* 使用Rust的文档注释功能，为所有公共API编写详细文档
* 编写系统架构文档，包括组件关系图和数据流图
* 编写示例代码，详细说明如何使用该驱动框架开发应用

**四、目标、主要特色及工作进度**

**1 课题目标**

该Kendryte K230芯片组件化系统驱动解决方案的开发基于Rust语言和嵌入式设计原则，旨在简化开发流程，提高系统安全性和可靠性，为开发者提供一个高效、灵活的驱动框架。系统需要实现以下核心功能：

（1）实现K230芯片的底层初始化，包括启动流程、内存管理和中断处理等基础功能。

（2）开发符合embedded-hal标准的硬件抽象层，支持UART、GPIO、I2C、SPI等常用外设的操作。

（3）集成RustSBI，提供标准化的系统调用接口，为上层操作系统和应用程序提供支持。

（4）开发TF卡镜像生成工具，简化固件打包和部署过程。

（5）通过组件化设计，实现高效、稳定的硬件资源管理，为应用开发者提供灵活、易用的编程接口。

**2 课题特色**

（1）该系统采用Rust语言作为主要开发语言，充分利用Rust的内存安全和并发安全特性，显著提高系统的可靠性和安全性。Rust的零成本抽象和高性能特性也确保了驱动框架的效率。

（2）采用组件化设计方法，将系统功能模块化，提高代码复用率和可维护性。这种设计方法使得不同模块可以独立开发、测试和更新，大大提高了开发效率和系统的可扩展性。

（3）实现了符合embedded-hal标准的硬件抽象层，为上层应用提供统一的接口。这不仅简化了应用开发，还提高了代码的可移植性，使得应用软件包在不同硬件平台之间迁移变得更加容易。

（4）集成了RustSBI，为上层软件提供标准化的系统调用接口。这为将来集成全功能操作系统或开发复杂应用奠定了基础，同时保证了软件生态的兼容性。

（5）开发了专门的TF卡镜像生成工具，简化了固件的打包和部署过程。这大大提高了开发和测试的效率，使得开发者可以更快地迭代和验证他们的设计。

（6）采用集成测试、单元测试等软件工程常用测试手段验证关键组件的正确性，提高了系统的可靠性和安全性，这在嵌入式系统和物联网设备中尤为重要。

**3 工作进度**

第01周至第03周：资料检索，完成英文技术资料翻译和开题报告

第04周至第09周：需求分析，进行系统的设计与开发

第10周至第12周：软件测试，进行系统的调试

第13周至第16周：整理毕业设计相关文档，撰写和修改毕业设计论文

**五、参考文献**

1. 胡霜,华保健,欧阳婉容,等.Rust语言安全研究综述[J].信息安全学报,2023,8(06):64-83.DOI:10.19363/J.cnki.cn10-1380/tn.2023.11.06.
2. 杨斌,蒋维,常泽海.Rust语言在核电安全级仪控研发应用探讨[J].仪器仪表用户,2023,30(06):65-68.
3. 陈中祥.一种基于组件化驱动的飞行控制系统设计方案[J].中国高新科技,2019,(24):89-91.DOI:10.13535/j.cnki.10-1507/n.2019.24.31.
4. 杜建华,万菁昱,瞿海娜.航电系统驱动软件标准组件技术的研究[J].信息通信,2017,(03):34-36.
5. HaoboGu. (2024). rmk [Computer software]: Mechanical keyboard firmware for stm32/rp2040, supports vial/dynamic keymap/eeprom, written in Rust. GitHub. [**https://github.com/HaoboGu/rmk**](https://github.com/HaoboGu/rmk)
6. dotcypress. (2023). μLA [Computer software]: Micro Logic Analyzer for RP2040. GitHub. [**https://github.com/dotcypress/ula**](https://github.com/dotcypress/ula)
7. fu5ha. (2023). LuLuu [Computer software]: Firmware for a custom RP2040-based display controller that streams animated images from a microSD card to a small LCD display. Github. [**https://github.com/fu5ha/luluu**](https://github.com/fu5ha/luluu)
8. cbruiz. (2023). prinThor [Computer software]: 3DPrinter/CNC/Engraver firmware framework powered by rust embassy for stm32 families and rp2040. Github. [**https://github.com/cbruiz/printhor**](https://github.com/cbruiz/printhor)