# 基于SCDAYU800A开发板的OpenHarmony 移植与适配研究

王 剑1,闻 飞2,孙庆生1

<sup>1</sup>长江大学计算机科学学院,湖北 荆州 <sup>2</sup>江苏润开鸿数字科技有限公司,江苏 南京

收稿日期: 2025年4月1日; 录用日期: 2025年4月23日; 发布日期: 2025年4月30日

# 摘 要

文章详细介绍了针对润开鸿鸿锐开发板SCDAYU800A的OpenHarmony操作系统的移植适配过程。润开鸿鸿锐开发板SCDAYU800A基于平头哥高性能RISC-V架构曳影TH1520芯片,具备强大的AI算力和丰富的功能接口。通过对OpenHarmony的适配,实现了从产品定义、工具链适配到内核移植、驱动适配等关键环节的完整流程。并重点探讨了RISC-V架构下针对标准系统Linux内核的移植以及触摸屏驱动的开发,为基于RISC-V架构的嵌入式系统研究提供了实践参考。

# 关键词

SCDAYU800A开发板,OpenHarmony,RISC-V架构,移植适配,触摸屏驱动

# Research on the Porting and Adaptation of OpenHarmony Based on the SCDAYU800A Development Board

Jian Wang<sup>1</sup>, Fei Wen<sup>2</sup>, Qingsheng Sun<sup>1</sup>

<sup>1</sup>School of Computer Science, Yangtze University, Jingzhou Hubei

<sup>2</sup>Jiangsu Runkaihong Digital Technology Co., Ltd., Nanjing Jiangsu

Received: Apr. 1<sup>st</sup>, 2025; accepted: Apr. 23<sup>rd</sup>, 2025; published: Apr. 30<sup>th</sup>, 2025

#### **Abstract**

This paper provides a detailed introduction to the porting and adaptation process of the OpenHarmony operating system for the RunKaiHong Hongrui development board SCDAYU800A. The SCDAYU800

文章引用: 王剑, 闻飞, 孙庆生. 基于 SCDAYU800A 开发板的 OpenHarmony 移植与适配研究[J]. 嵌入式技术与智能系统, 2025, 2(2): 123-132. DOI: 10.12677/etis.2025.22010

development board, based on the Pingtouge high-performance RISC-V architecture Yieying TH1520 chip, features powerful AI computing capabilities and a rich set of functional interfaces. By adapting to OpenHarmony, the entire process from product definition to toolchain adaptation, kernel transplantation, and driver adaptation has been successfully implemented. The paper focuses on the adaptation of the transplantation of the Linux kernel under the RISC-V architecture, and the development of the touchscreen driver, providing practical references for embedded system development based on the RISC-V architecture.

# **Keywords**

SCDAYU800 Development Board, OpenHarmony, RISC-V Architecture, Porting and Adaptation, Touchscreen Driver

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

随着 RISC-V 指令集架构的快速发展,基于该架构的计算机系统在工业控制、智能设备等领域得到了广泛应用[1]-[3]。润开鸿鸿锐开发板 SCDAYU800A 作为一款高性能的 RISC-V 开发平台,具备强大的 AI 算力和丰富的功能接口,适用于多种应用场景。OpenHarmony 是一个由华为公司贡献给开放原子基金会(OpenAtom Foundation)的开源项目。它是一个全场景分布式操作系统,旨在为各种智能设备提供统一的操作体验。本文将重点探讨如何将 OpenHarmony 4.1 Release 版本移植到润开鸿鸿锐开发板 SCDAYU800A上,并实现系统的稳定运行。

# 2. 润开鸿鸿锐开发板 SCDAYU800A 硬件特性

润开鸿鸿锐开发板 SCDAYU800A 具备平头哥高性能 RISC-V 架构曳影 TH1520 芯片,集成 4 核高性能 RISC-V 处理器玄铁 C910 架构。润开鸿鸿锐开发板 SCDAYU800A 的外观如图 1 所示。



Figure 1. Runkaihong Hongrui development board SCDAYU800A appearance 图 1. 润开鸿鸿锐开发板 SCDAYU800A 的外观

该开发板具备以下硬件特性:

- 1) AI 算力达 4TOPs,支持蓝牙、音频、视频和摄像头等功能。
- 2) 支持多种视频输入输出接口,如 MIPI、HDMI等。
- 3) 提供丰富的扩展接口,如 GPIO、SPI、I2C 等。
- 4) 适用于工控平板、智慧大屏、智能 NVR、信息发布系统、云终端、车载中控等多种场景。
- 图 2显示了润开鸿鸿锐开发板 SCDAYU800A Mipi 屏幕安装的方法。



Figure 2. Runkaihong Hongrui development board SCDAYU800A Mipi screen installation 图 2. 润开鸿鸿锐开发板 SCDAYU800A Mipi 屏幕安装

# 3. OpenHarmony 移植流程

移植就是把程序从一个运行环境转移到另一个运行环境。在主机-开发机的交叉模式下,即是把主机上的程序下载到目标机上运行。图 3 显示了 OpenHarmony 移植到 SCDAYU800 的流程情况。OpenHarmony 版本选择必须大于 3.2 Beta2 以上版本。

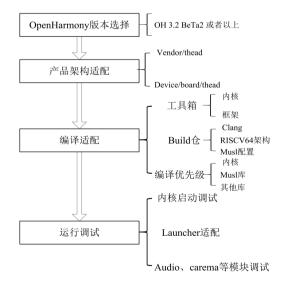


Figure 3. Flowchart of porting OpenHarmony to SCDAYU800 图 3. OpenHarmony 移植到 SCDAYU800 的流程图

OpenHarmony 的移植过程包括产品定义、工具链适配、musl 库适配、内核移植、init 启动子系统移植、显示适配等关键环节[4]。以下是移植步骤。

# 3.1. 产品定义

在 OpenHarmony 的移植过程中,产品定义是基础步骤。针对 OpenHarmony 标准系统,适配新的产品的方法是首先要在 vendor、device/board 和 device/soc 三个目录下创建产品相关的三个目录 vendor/hihope/\${product\_name}、device/board/hihope/\${product\_name}和 device/soc/\${chip\_product}/\${chip},以 SCDAYU800A 为例,在当前产品中 product\_name 设置为 dayu800,chip\_product 设置为 thead,chip 设置为 th1520。

vendor/hihope/dayu800 目录主要存放厂家资料以及产品的配置文件,包括描述产品的 config.json、产品的 hcs 配置文件以及其他配置文件等。

device/board/hihope/dayu800 目录主要用于存放开发板相关的文件,包括外设驱动、启动参数、内核编译相关文件、烧录相关文件、uboot 相关文件以及升级相关文件。

device/soc/thead/th1520 目录主要用于存放和芯片 SoC 相关的文件和库,这些文件只会因芯片 soc 改变才会修改,而不会因为开发板变化而进行修改。

快速适配这三个仓的方法是学习能编译通过的产品,例如 rk3568 产品,然后将其拷贝一份,并重命名为当前产品名,然后按照以下步骤根据自己产品的特性进行逐步修改。根据上述分析,首先需要在 //vendor/hihope 目录下创建 dayu800 产品目录,并创建 config.json 文件,用于描述产品所使用的 SOC 及所需的子系统。配置文件内容如下[5]:

```
"product_name": "dayu800",
"device_company": "thead",
"device build path": "device/board/hihope/dayu800",
"target_cpu": "RISCV64",
"type": "standard",
"version": "3.0",
"board": "dayu800",
"api_version": 11,
"enable ramdisk": true,
"build selinux": false,
"build_seccomp": false,
"inherit": [
  "vendor/hihope/dayu800/rich.json",
  "vendor/hihope/dayu800/chipset_common.json"
"subsystems": [
    "subsystem": "security",
    "components": [
       {
         "component": "selinux",
```

该部分根据 vendor 仓的定义,在//device/board/hihope 目录下创建一个 dayu800 的产品,然后创建一个 ohos.build 文件定义产品的 subsystem 以及 BUILD.gn 定义产品的编译项目。

在//device/soc 下根据芯片公司名称创建一个 thead 的仓,存放 th1520 soc 相关的代码和闭源库,通过 //device/board/hihope/BUILD.gn 添加模块进行关联编译。该仓下的 ohos.build 文件描述的是设备子系统,这里使用的是 device\_th1520,相关名称需要将拷贝过来的文件修改为当前设备的子系统名。同时需要将 ohos.build 的 module\_list 下的模块及子模块中涉及的 part\_name 和 subsystem\_name 全部修改为 device\_th1520。

至此,可以通过使用如下命令,启动 dayu800 的构建:

./build.sh --product-name dayu800

#### 3.2. 工具链适配

因为官方的 OpenHarmony 不支持 RISCV64 架构,因此在工具链架构的配置上相对来说配置起来比较困难,首先解决的是 build 仓中的架构适配,这部分仿照 ARM64 配置的部分进行相应的 RISCV64 架构添加,同时对于 musl 的编译配置根据 RISCV64 架构进行对应添加,内核的编译部分根据平头哥提供的编译工具链,使用 llvm 会有相关私有指令集无法编译[6]。

#### 3.3. musl 库适配

musl 库是一个轻量级、高性能的 C 标准库(libc)实现,专为 Linux 系统设计。它旨在提供标准兼容性、代码简洁性和高效性,尤其适合嵌入式系统、容器化环境(如 Docker)或需要高度可移植性的场景。这里首先编译 musl 库是因为该库依赖的模块相对最少,能最快测试 RISCV64 架构编译配置结果是否符合要求,该部分的编译主要涉及到 build 仓的配置、musl 仓的 RISCV64 架构配置,以及 musl 库中有架构的代码,这一块部分从 glibc 中获取。

# 3.4. 内核移植

内核移植是 OpenHarmony 移植过程中的核心环节。润开鸿鸿锐开发板 SCDAYU800A 采用平头哥提供的 Linux 内核,适配 OpenHarmony 特性。该部分的移植既要对 Linux 内核本身很熟悉,同时也要对鸿蒙的内核特性宏配置很熟悉。本内核移植工作的主要思路是:

第一步是配置内核的编译流程脚本。

内核的编译是 device/board/hihope/dayu800/kernel/BUILD.gn 中 kernel 模块调用对应目录下的 build\_kernel.sh 脚本进行编译的。该脚本的核心思想是首先拷贝内核源码到 out/kernel/src\_tmp/linux-5.10 路径下,给内核源码中打上 HDF (Hardware Driver Foundation)的补丁,然后在内核源码中创建 vendor 路径的软链接(主要是设备树 dts 和产品的外设驱动),然后将编译的 config 文件拷贝到 arch/arm64/config 路径下,将产品的 dts 文件拷贝到 vendor/arch/arm64/boot/dts/hihope 路径下,接下来将外设驱动链接到内核

中,编译内核。最后拷贝编译成功的镜像到 out/dayu800/packages/phone/images 中。

第二步是将产品的设备树 dts 完整地移植过来, 然后编译内核并解决相应的编译问题。

第三步是将架构代码即 arch/arm64 下的代码尽可能地替换成产品 linux 中的内容, 然后编译内核并解决相应的编译问题。

在 device/board/hihope/dayu800/kernel 目录下进行内核编译。除内核以外的代码编译采用的是 OpenHarmony 版本自身的 llvm 工具链。

这里只列出核心编译命令。

# 首次编译请先执行下面命令

./build/prebuilts\_download.sh

# 全量代码编译

./build.sh --product-name dayu800 --gn-args full\_mini\_debug=false --ccache

# 单模块编译

# module\_name 举例: "kernel:kernel",表示编译 kernel 目录下的 kernel 模块,所有后面的 kernel 是 module name

./build.sh --product-name dayu800 --ccache --build-target module

# 内核模块编译

./build.sh --product-name dayu800 --ccache --build-target kernel

编译成功有如下信息打印

[OHOS INFO] c overall build overlap rate: 1.05

[OHOS INFO]

[OHOS INFO]

[OHOS INFO] dayu800 build success

[OHOS INFO] cost time: 0:45:57

=====build successful=====

2025-01-18 13:06:52

将编译后的内核打包成 boot.ext4 镜像文件。

第四步是将产品 linux 的 config文件内的宏移植过来。

第五步是运行内核,并根据运行时问题进行逐个解决。

#### 3.5. init 启动子系统移植

在成功适配 musl 库和内核后,需要适配和启动第一个 init 进程及其初始化配置文件。主要工作包括:

- 1) 编译适配 startup init 模块。
- 2) 配置启动参数,确保系统能够正常启动。

# 3.6. 显示适配

显示适配是系统启动的最后一步, 主要包括以下内容:

- 1) 在 shell 中运行 bootanimation, 确保启动动画正常显示。
- 2) 调试 OpenHarmony 的 Launcher 模块,完成系统启动。
- 3) 该模块涉及的仓为 graphic\_graphic\_2d, 代码路径为 foundation/graphic/graphic\_2d, 该模块是针对

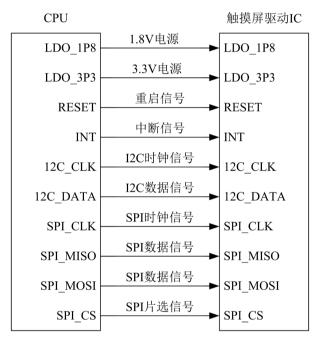
当前产品而适配。

# 4. 触摸屏驱动开发

触摸屏驱动是润开鸿鸿锐开发板 SCDAYU800A 的重要组成部分,其开发过程基于 HDF 驱动管理框架中的 Input 模型。触摸屏的驱动被放置在//drivers/hdf\_core/framework/model/input/driver/touchscreen 目录中。移植触摸屏驱动主要工作是向系统注册 ChipDevice 模型实例[7]。触摸屏驱动的主要工作包括:

- 1) 对触摸屏驱动 IC 进行上电、配置硬件管脚并初始化其状态。
- 2) 注册中断、配置通信接口(I2C或SPI)。
- 3) 设定 Input 相关配置、下载及更新固件等操作。

图 4 显示了润开鸿鸿锐开发板 SCDAYU800A 触摸屏的常用管脚。



**Figure 4.** Common pins of the development board SCDAYU800A touch screen **图 4.** 开发板 SCDAYU800A 触摸屏的常用管脚

润开鸿鸿锐开发板 SCDAYU800A 触摸屏驱动的开发过程分为以下三个步骤,这里仅列出部分代码。

#### 4.1. 设备描述配置

在 OpenHarmony 中,配置(Configuration)通常指对系统、硬件或应用程序的参数、选项和功能进行定制和调整的过程,以满足特定设备或场景的需求。在 HDF 驱动框架下,需要在配置文件中注册驱动信息。设备描述配置主要包含 Input 驱动模型各模块层级信息配置文件路径为:

vendor/hihope/dayu800/hdf\_config/khdf/device\_info/device\_info.hcs, HDF 框架依据该配置信息实现对 Input 模型各模块的依次加载。配置内容如下:

```
input :: host {
    hostName = "input_host";
    priority = 100;
    device_input_manager :: device {
```

```
device0 :: deviceNode {
    policy = 2;  // 向外发布服务
    priority = 100;  // 加载优先级
    preload = 0;  // 加载该驱动
    permission = 0660;
    moduleName = "HDF_INPUT_MANAGER";
    serviceName = "hdf_input_host";
    deviceMatchAttr = "";
    }
}
// 其他设备配置
```

# 4.2. 板级配置及器件私有配置

板级配置及器件私有配置文件路径为 vendor/hihope/dayu800/hdf\_config/khdf/input/input\_config.hcs, 配置内容如下:

```
root {
    input_config {
         touchConfig {
             touch0 {
                  boardConfig {
                      match_attr = "touch_device1";
                      inputAttr {
                                                    //0代表触摸屏
                          inputType = 0;
                           solutionX = 800;
                           solutionY = 1280;
                          devName = "main_touch"; // 设备名称
                      }
                      // 其他配置
         }
}
```

#### 4.3. 添加器件驱动

在器件驱动中,主要实现平台预留的差异化接口。以数据获取及解析为例,润开鸿鸿锐开发板 SCDAYU800A 使用了 gt911 触摸屏,代码路径为:

drivers/hdf\_core/framework/model/input/driver/touchscreen/touch\_gt911.c, 部分代码如下:

static void ParsePointData(ChipDevice \*device, FrameData \*frame, uint8\_t \*buf, uint8\_t pointNum)

```
{
    int32_t resX = device->driver->boardCfg->attr.resolutionX;
    int32_t resY = device->driver->boardCfg->attr.resolutionY;
    // 数据解析逻辑
//器件私有配置解析
//器件设备注册到平台驱动
//调用 Input HDI 接口
}
图 5 显示了调用 Input HDI 的步骤。
```

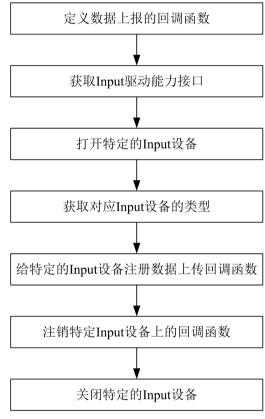


Figure 5. Steps to call Input HDI 图 5. 调用 Input HDI 的步骤

# 5. 结论

本文介绍了润开鸿鸿锐开发板 SCDAYU800A 的 OpenHarmony 操作系统的移植适配过程。在分析 OpenHarmony 标准系统的前提下,成功实现了 OpenHarmony 在润开鸿鸿锐开发板 SCDAYU800A 上的稳定运行。该研究为基于 RISC-V 架构的单板机系统开发提供了实践参考,具有一定的应用价值。

# 参考文献

DOI: 10.12677/etis.2025.22010

- [1] RISC-V 架构手册[EB/OL]. https://riscv.org/, 2024-12-11.
- [2] 刘畅, 武延军, 吴敬征, 赵琛. RISC-V 指令集架构研究综述[J]. 软件学报, 2021, 32(12): 3992-4024.

- [3] 林金龙,何小庆. 深入理解 RISC-V 程序开发[M]. 北京: 航空航天大学出版社, 2021.
- [4] OpenHarmony 官方文档[EB/OL]. https://www.openharmony.cn/docs/zh-cn/overview/, 2024-12-13.
- [5] 润开鸿鸿锐开发板 SCDAYU800A 技术手册[EB/OL]. https://www.hoperun.com/, 2024-12-10.
- [6] HDF 驱动开发框架[EB/OL]. OpenHarmony 开发者社区. <a href="https://www.openharmony.cn/docs/zh-cn/overview/">https://www.openharmony.cn/docs/zh-cn/overview/</a>, 2024-12-12.
- [7] ITOPEN/DAYU800 [EB/OL]. https://gitee.com/itopen/dayu800, 2025-01-23.