

# 基于龙芯平台的 PXI 设备驱动设计

杜影<sup>1</sup>, 朱元元<sup>2</sup>, 刘康丽<sup>1</sup>, 毕硕<sup>1</sup>, 谷静<sup>2</sup>

(1. 北京航天测控技术有限公司, 北京 100041;

2. 北京航天长征飞行器研究所, 北京 100076)

**摘要:** 针对国产龙芯 PXI 测试平台搭载中标麒麟操作系统的软硬件环境, 分析了龙芯平台下 PXI 设备驱动的基本框架、关键数据结构和编译方式, 详细分析了 PXI 设备初始化、数据读写、设备控制、中断处理等典型的 PXI 设备驱动模块, 讨论了龙芯平台下 PXI 设备驱动的设计过程, 并在驱动设计过程中依照仪器软件结构向应用层提供了驱动的使用接口, 简化了应用层软件设计, 为基于龙芯平台和中标麒麟系统的 PXI 仪器软件开发提供了较好的应用案例; 实际验证结果表明, 在测试控制领域, “龙芯平台+中标麒麟”的组合成功地把开源软件和 PXI 硬件结合在一起, 北京航天测控公司在国内率先建立了自主可控的 PXI 通用测试平台。

**关键词:** 龙芯 3A3000 处理器; 中标麒麟操作系统; Linux; PXI 设备驱动

## Design of PXI Device Driver Based on LOONGSON Platform and NeoKylin Operating System

Du Ying<sup>1</sup>, Zhu Yuanyuan<sup>2</sup>, Liu Kangli<sup>1</sup>, Bi Shuo<sup>1</sup>, Gu Jing

(1. Beijing Aerospace Measurement & Control Technology Co., Ltd., Beijing 100041, China;

2. Beijing Institute of Space Long March Vehicle, Beijing 100076, China)

**Abstract:** Concern over the software and hardware environment of domestic LOONGSON PXI test platform with NeoKylin operating system; the basic framework; key data structure and compiling method of PXI device driver under LOONGSON platform are analyzed; Typical PXI device driver modules; such as PXI device initialization; data read-write; device control; interrupt handling are also analyzed in detail; The design process of PXI device driver under LOONGSON platform is discussed; And the driver interface for application layer which is based on the Instrument Software Architecture and simplifies the design of application layer software is provided; The design provides a good application case for the development of PXI equipment software based on LOONGSON platform and NeoKylin operating system; The actual verification results show that in the field of measurement and control; the combination of LOONGSON Platform and NeoKylin successfully combines open source software and PXI hardware; Beijing AMC corporation takes the lead in establishing a self-controllable PXI universal test platform in China.

**Keywords:** LOONGSON 3A3000 processor, NeoKylin operating system, Linux, PXI device driver

## 0 引言

当前, 鉴于军工芯片在信息化装备所处的核心地位, 采用自主研发的国产芯片势在必行, 并且已经得到了国家政策的支持和大力发展。随着国产核心基础软硬件的性能与功能逐步完善, 龙芯平台搭载中标麒麟操作系统的模式已成为国产计算开发平台的首选, 其发展、推广和应用是国产软硬件推广和规模化发展的技术基础和科学依据。

军用 PXI 模块产品一直以地面测试和系统级维修保障为主要任务。在“自主可控”的背景下, 将开源软件和 PXI 硬件平台结合在一起, 打造基于开源软件的通用 PXI 测试平台, 建立基于国产龙芯 CPU 的 PXI 仪器生态是实现国产

化替代的必经之路。如今, PXI 产品国产化正面临着发展瓶颈与挑战。首先, 在硬件平台方面, 嵌入式控制器的 CPU 要更换成国产龙芯 CPU, 其整体性能和稳定性有待提高; 其次, 在软件开发环境方面, 龙芯平台下的仪器控制软件要摒弃对 NI 公司 LabView、CVI、VISA 库以及 Windows API 的依赖, 而自主开发“龙芯平台+中标麒麟系统”下的 PXI 设备驱动程序和应用程序。

本文将详细讨论龙芯平台下 PXI 设备驱动的基本架构和设计方法, 并在龙芯平台下以 PXI 数字多用表模块为例进行试验验证。

## 1 龙芯 PXI 测试平台

### 1.1 龙芯 3A3000

龙芯 3 A 系列处理器是首款国产商用四核处理器, 文中 PXI 龙芯平台所使用的处理器为龙芯 3A3000, 其工作主频为 1.0 GHz, 主要技术特征为: 片内集成 4 个 64 位的高性能处理器核; 8MB 的分体共享三级 Cache; 通过目录协议维护多核及 I/O DMA 访问的 Cache 一致性; 2 个 64 位带

收稿日期: 2019-06-24; 修回日期: 2019-08-29。

**作者简介:** 杜影(1981-), 女, 高级工程师, 主要从事测试测量技术方向的研究。

朱元元(1982-), 男, 高级工程师, 主要从事测试技术方向的研究。

ECC, 667 MHz 的 DDR2/3 控制器; 32 位 33 MHz PCI; 1 个 LPC、2 个 UART、1 个 SPI、16 路 GPIO 接口<sup>[1]</sup>。龙芯 3A3000 处理器采用 MIPS 架构, 因此, 目前运行在 X86 架构的操作系统与应用软件均无法在龙芯平台运行和使用, 需要通过交叉移植与本地移植的技术实现软件环境适配<sup>[2-3]</sup>。

### 1.2 中标麒麟操作系统

国产中标麒麟是一种基于 Linux 内核的桌面操作系统, 支持众多处理器架构。本文使用的 PXI 龙芯平台安装的操作系统为中标麒麟桌面操作系统软件(龙芯版) V5.0, 支持龙芯 3A3000 (64 位) 平台采用的 MIPS 架构, 具有强大的网络优势和极为突出的安全性和稳定<sup>[4]</sup>, 支持主流国产软硬件, 能够最大程度发挥中标麒麟操作系统和龙芯 3A3000 整机平台的整体性能。

### 1.3 PXI 测试平台

如图 1 所示, 本文所搭建的国产龙芯 PXI 测试平台包含软硬件平台两个部分。硬件平台主要由 PXI 机箱、基于龙芯 3A3000 处理器的嵌入式控制器和 PXI 模块产品组成。软件平台主要包括: 中标麒麟 5.0 操作系统软件、集成安装在该操作系统的可视化软件开发工具 QT5.6、以及其他脚本语言。基于此平台可以完成 PXI 模块产品的驱动设计、开发、调试、加载、调用, 编写可视化应用程序, 并进行 PXI 模块的功能性能验证。

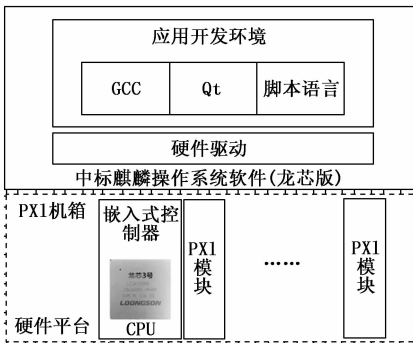


图 1 龙芯 PXI 测试平台

## 2 龙芯平台下 PXI 设备驱动分析

在 Windows 操作系统下, PXI 设备驱动开发人员广泛采用行业标准的 VISA (虚拟仪器系统 I/O 接口软件) 对 PXI 仪器进行配置、控制、编程和调试, 也有小部分开发人员在 WindowsAPI 基础上进一步实现 PXI 仪器驱动开发。但是在国产龙芯平台搭载中标麒麟操作系统的环境下, 以往的 PXI 设备驱动已无法运行和使用(如第 1 节所述), 程序员必须面向 RISC (精简指令集计算机) 体系结构的 Linux 系统重新开发 PXI 设备驱动。

### 2.1 龙芯平台下 PXI 设备驱动基本框架

考虑到 PXI 仪器的应用场合和测试测量对象, 可将 PXI 设备驱动程序实现分为以下几个部分: 初始化设备模块、设备打开模块、数据读写和控制模块、中断处理模块、设备释放模块和设备卸载模块, 如图 2 所示。

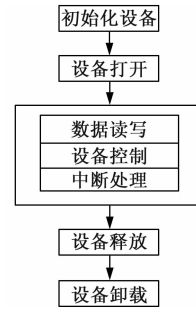


图 2 Linux 下 PXI 设备驱动组成部分

初始化设备需要完成以下工作:

- (1) 检查系统 (Linux 内核) 是否支持 PCI 总线;
- (2) 注册硬件驱动程序, 即检查设备所占用的插槽并保存它所占用的插槽的位置等信息;
- (3) 启动 PCI 设备, 读取所有配置好的 PCI 设备信息中的参数和属性 (包括厂商 ID、设备 ID 等) 以备驱动程序使用;
- (4) 使能 DMA (直接数据存储), 以便在内存操作的时候能够让设备把数据通过总线主 DMA 传至系统内存。

设备打开模块主要是根据当前 PXI 设备号获取其设备参数和属性, 并申请对当前 PXI 设备的控制权。另外, 如果当前 PXI 设备有中断请求, 则需要在该驱动模块里申请中断使能, 并在 Linux 内核注册中断处理程序。

数据读写和设备控制模块主要通过: 读设备操作、写设备操作、I/O 控制操作、内存重映射操作等向应用程序提供对硬件进行控制的接口。面向不同的 PXI 设备驱动程序, 需要程序员重写该模块各个接口函数的方法, 从而实现硬件逻辑控制和数据的收发与存储。数据读写和设备控制模块是整个 PXI 设备驱动程序的核心部分。

中断处理模块即指中断处理程序, 当系统接收到中断后, 程序先要根据中断号判断是否为当前设备发出的中断请求, 如符合条件则立即开始执行中断处理程序, 例如: 在接收到中断后完成数据的接收与存储或复位硬件等。而中断使能如前文所述在设备打开模块中完成, 中断释放则在后面的设备释放模块中完成。

设备释放模块和设备卸载模块比较简单, 它们与设备打开模块和初始化设备模块相对应, 主要是在设备控制结束后释放对设备的控制以及所占用的内存和中断, 并将设备驱动程序从内核中注销。

### 2.2 关键数据结构

类同于 Linux 下的 PCI 驱动程序, 在 PXI 驱动程序中起核心作用的关键数据结构 (struct) 也是 pci\_driver 和 pci\_dev, 它们的声明包含文件在 linux/pci.h 中。其中, pci\_driver 这个数据结构涵盖: 设备模块名称、所有能够驱动的设备列表、查找并初始化设备的函数和卸载设备的函数, 而 pci\_dev 这个数据结构则详细描述了一个 PXI 设备几乎所有的硬件信息, 包括厂商 ID、设备 ID、各种资源等<sup>[5]</sup>, 其数据格式在此不做赘述。

在 2.1 节的初始化设备模块中, 启动 PCI 设备后, 所有配置好的 PCI 设备信息保存在结构体 `pci_driver` 中, 而在设备打开模块中, 当前 PXI 设备的信息保存在 `pci_dev` 中。数据读写和设备控制是基于 `pci_dev` 中的信息执行相关操作的。

### 2.3 驱动程序发布

在 Linux 系统下, 设备驱动程序发布就是将其编译成内核的一部分, 其主要有静态编译和动态编译两种方法: 第一种方法要改动内核的源文件, 直接将驱动程序编译成内核的一部分, 从而增加了内核的大小, 且不利于调试; 第二种方法是将驱动程序编译成可安装模块, 通过使用 `insmod` 命令将驱动程序动态加载到内核中, 使之成为系统内核的一部分, 当不使用该设备驱动时则使用 `rmmod` 命令将其从内核中卸载, 另外, 还可将加载命令放入开机自启动选项, 这样, 系统开机时就能够自动加载驱动。比较两种驱动发布方式, 动态编译更合理也更灵活<sup>[6-7]</sup>。

## 3 龙芯平台下 PXI 设备驱动设计与调用

### 3.1 驱动设计

根据上节所述, 我们首先要在 Linux 下编写和加载驱动模块文件, 即 “.ko” 文件。“ko” 文件是 kernel object 文件, 其意义在于把内核的一些功能移动到内核外, 需要的时候插入内核, 不需要时卸载, 这样可以缩小内核的存储容量, 而且使用方便。

“.ko” 文件的源码是 “.c” 文件, 该文件的核心内容是构造驱动程序的基本框架, 文件中含有 2.1 节所述的几个关键功能模块以及 2.2 节所述的关键数据结构。需要注意的是, 同加载和卸载模块相关的函数或数据结构都要在前面加上 `_ _init`、`_ _exit` 等标志符, 以使同普通函数区分开来。

在编译驱动文件之前要先编写 Makefile 文件, 该文件可使用 `make` 工具自动完成编译工作, 其有利于程序员修改和重新编译。Makefile 文件的基本内容如下 (以 `hello.c` 生成 `hello.o` 为例):

```
obj-m := hello.o
KERNEL := /usr/src/kernels/(uname -r)/
PWD := (shell pwd)
modules :
(MAKE) -C (KERNEL) M=(PWD) modules
.PHONEY:clean
clean :
rm -f *.o *.ko
```

在终端执行 `make` 命令, 即在指定路径下生成了 `hello.o` 文件。然后执行 `insmod` 命令安装 `hello` 驱动模块, 如下所示:

```
insmod hello.ko 相对路径加载
```

为了简化了应用层软件设计, 我们需要进一步生成 “.so” (shared object) 文件。该文件是 Linux 下的动态链接库文件, 类似于 windows 下的 .dll 文件。“so” 文件的生

成需要 “.h” 和 “.c” 文件, 为了方便应用层软件的调用以及不同平台软件的移植, 我们可以对 “.c” 文件的函数进行标准化封装, 函数名和参数可依照 Windows 下 dll 中的函数进行定义, 例如:

某 PXI 仪器在 Windows 下的应用程序调用了 `visa` 库函数 `viIn8`, 那么就可以在 Linux 下封装函数名同为 `viIn8` 的函数, 并生成相应的 .so 文件提供给 Linux 下的应用程序。这样, 那些使用跨平台软件开发环境 (如 Qt) 编写的可视化程序, 既不需要更改界面源码, 也不需要更改逻辑源码, 只需要在编译程序时链接 “.so” 文件, 就实现了跨平台软件移植。

编译 “.so” 文件同样需要先编写 Makefile 文件, 其内容一般为 (以生成 `libhelloio.so` 为例):

```
CFLAGS = -Wall -g -O -fPIC
INCLUDE = helloio.h
TARGET = libhelloio.so
SRCS = helloio.c
OBSJ = (patsubst %.c,%o,(SRCS))
all: helloio
helloio:(OBSJ) (INCLUDE)
gcc -shared -fPIC (OBSJ) -O3 -o (TARGET)
gcc (CFLAGS) -O3 -c < -o *.o
clean:
rm -f *.o
rm -f *.so
```

在终端执行 `make` 命令, 即生成了 `libhelloio.so` 文件。

### 3.2 驱动调用

目前, 龙芯软件生态环境还处于初始发展阶段, PXI 仪器软件的迁移和二次开发存在一定难度, 需要操作系统、应用软件以及硬件兼容的紧密配合。在中标麒麟操作系统中可使用的可视化集成开发环境是 Qt Creator, 选择 Qt 作为开发工具不仅因为其开发功能强大, 更主要是因为它有开源版本且支持跨平台运行<sup>[8]</sup>。

在 Qt Creator 中创建工程后除了工程文件本身还包含: 头文件 (.h)、源文件 (.c) 以及界面文件 (.ui)。在编写逻辑代码调用 “.so” 文件时, 首先添加相应的头文件, 例如 `helloio.h` (对应 `libhelloio.so`), 然后在工程文件 (.pro) 中引用库文件, 代码如下:

```
LIBS += -L/lib64 -l helloio
```

如 3.1 节所述, 如果在驱动设计时将 .so 文件中的函数封装为 Windows 下 .dll 文件的同名函数, 那么在跨平台移植的过程中, 软件人员不再需要进行二次开发, 只需简单的程序移植并配置好构建和运行环境就能够编译出 Linux 下的可执行文件。在 Linux 下的 Qt 环境配置如下图所示。

### 3.3 遇到的问题

本文使用的龙芯平台是试验版, 其中断服务功能并不完善, 机箱上电后, PXI 设备中断号在初始化设备的过程中不能自动分配, 也就是说系统不能获知并列出机箱中所有 PXI 硬件设备的中断号, PXI 机箱中每个插槽所使用的



图 3 Linux 下 Qt 环境配置

硬件设备中断号固定（如：1 号插槽设备的中断号固定为 04H），且需要人工测试获取插槽与中断号的对应关系。因此，在设备打开程序中要根据硬件模块所在的插槽位置来配置当前 PXI 设备所对应的中断号，并使能中断，获取其使用权。

这个问题将导致同一型号的 PXI 硬件设备无法使用同一个驱动程序，不同插槽的 PXI 设备要根据其对应的中断号加载不同的驱动程序，从而影响到同型号 PXI 模块驱动程序的通用性，增加了驱动开发的复杂度。在新版的龙芯平台上，该问题已解决。

### 4 应用案例

本文将以北航测控公司的 PXI 仪器模块产品 AMC4311（5.5 位数字多用表）为对象，基于龙芯平台和中标麒麟 5.0 操作系统进行 PXI 仪器模块的驱动程序和应用程序开发，以验证上述 Linux 下驱动程序设计的正确性和实用性。

AMC4311 的硬件结构图如图 4 所示，其具备直流电压、交流电压、直流电流、交流电流、2 线电阻、4 线电阻和频率测量功能<sup>[9]</sup>。

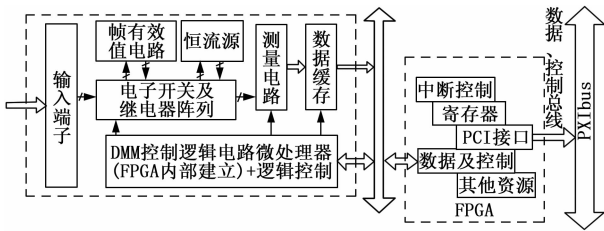


图 4 数字多用表模块硬件结构图

在龙芯平台的中标麒麟 5.0 系统下 AMC4311 驱动程序开发过程如下：

- (1) 编写设备驱动框架函数，形成 Amcdev.c 源文件；
- (2) 编写 MakeFile 文件，使用 make 命令生成 Amcdev.o 文件，生成路径为：/usr/src/kernels/Amcdev.o；
- (3) 编写标准化库函数，形成 amcIO.c 源文件和 amcIO.h 头文件；
- (4) 编写 MakeFile 文件，使用 make 命令生成 libamcIO.so 文件，生成路径为：/usr/local/lib/libamcIO.so。

可视化软件开发环境选用跨平台软件 Qt5.6<sup>[7]</sup>，界面源

码和逻辑代码隔离，这样可以在不修改界面源码的情况下，简单地将 Windows 下的界面程序移植到 Linux 下使用；逻辑代码调用的驱动函数是“libamcIO.so”文件封装的自定义接口函数。在 Linux 下重新配置 QT 的构建和运行环境，编译程序生成 Linux 下的可执行文件，其运行界面如图 5 所示。



图 5 龙芯平台下数字多用表控制程序运行界面

### 5 结束语

采用自主可控软硬件产品对于实现国产化替代、打造自主可控的 IT 环境有着非常重要的意义。在全球开源技术蓬勃发展，在中国政府力主导掌握测量等基础学科核心技术的大背景下，国内对包括 PXI 在内的测试测量技术一定会更加重视，国产 PXI 软硬件产品的市场份额也会保持一个稳定上升的增长态势。北京航测控公司已在国内率先建立了自主可控的 PXI 通用测试平台，军工科研院所等机构也正在通过逆向设计、自主研发等方式实现国产化替代。未来，龙芯平台搭载中标麒麟操作系统的软硬件组合将为我国自主软硬件产业带来良好的发展机遇。

### 参考文献：

- [1] 龙芯中科技术有限公司，龙芯 3A3000/3B3000 处理器用户手册 [Z]. 北京：龙芯中科技术有限公司，2017.
- [2] 王亮亮，杨琴，芮雪. 面向 RISC 体系结构的 Linux 系统移植原理 [J]. 计算机技术与发展, 2014 (8): 76-79, 83.
- [3] 施志林. RISC、CISC 技术比较与研究 [J]. 中国科技信息, 2008 (11): 131-132.
- [4] 中标软件. 中标麒麟桌面操作系统 [EB/OL]. <http://www.cs2c.com.cn/product/2019>.
- [5] 陈莉君. Linux 中 PCI 设备驱动程序的开发 [J]. 青海师范大学学报 (自然科学版), 2006 (2): 57-59.
- [6] 樊东东. Linux 内核源码剖析 [M]. 北京：机械工业出版社，2010.
- [7] 倪继利. Linux 内核分析及编程 [M]. 北京：电子工业出版社，2005.
- [8] 陆文周. Qt5 开发及实例 [M]. 北京：电子工业出版社，2014.
- [9] 北京航测控技术有限公司，PXI 总线 5.5 位数字多用表 (AMC4311A) 用户手册 [Z]. 北京：北京航测控技术有限公司，2011.