

# 基于银河麒麟操作系统的 PXI 仪器控制软件设计

胡宇

(成都天奥测控技术有限公司, 成都 611731)

**摘要:** 当前 PXI 仪器控制软件主要基于 Windows 操作系统开发, 通过调用标准 VISA I/O 库实现仪器驱动控制, 这些软件和 I/O 库通常不能在国产操作系统上运行, 在一定程度上限制了 PXI 系统在国产化进程中的发展; 针对此问题, 对 PXI 仪器驱动控制技术进行了研究, 分析国产操作系统硬件驱动原理和虚拟仪器软件架构, 采用基于内存映射的 PXI 总线寄存器访问技术, 并提出仪器分类驱动控制和互换方法, 研制了一套可在银河麒麟操作系统上运行的软件系统, 包括仪器控制界面程序、驱动程序和底层 I/O 库等; 经实验测试在银河麒麟操作系统上实现了 PXI 仪器的参数设置、测量控制和测试结果返回显示等, 解决了现有控制驱动软件、VISA I/O 库不支持国产银河麒麟操作系统, 导致 PXI/PXIe 仪器不能使用的问题。

**关键词:** 银河麒麟操作系统; PXI/PXIe; VISA; 基址寄存器; 内存映射

## Design of PXI Instrument Control Software Based on Kirin Operating System

HU Yu

(Chengdu Spaceon T&C Technology Co., Ltd., Chengdu 611731, China)

**Abstract:** Current PXI instrument control software is mainly developed based on Windows operating system, and instrument driver control is achieved by calling standard VISA I/O library. These software and I/O libraries usually cannot run on domestic operating systems, which to some extent limits the development of PXI systems in the localization process. In response to this issue, research on PXI instrument driver control technology is conducted to analyze the hardware driver principles and virtual instrument software architecture of domestic operating systems, adopt PXI bus register access technology based on memory mapping, propose an instrument classification driver control and swapping method, and develop a software system that can run on the Kirin operating system, including instrument control interface program, driver program, and underlying I/O library. Through experimental testing, the parameter setting, measurement control, and test result return display of PXI instruments are implemented on the Kirin operating system, solving that existing control driver software and VISA I/O library can not support the domestic Kirin operating system, resulting in PXI/PXIe instruments to be unusable.

**Keywords:** Kirin OS; PXI/PXIe; VISA; Base address register; memory mapping

## 0 引言

近年来, 随着国内电子工业、航空航天、科学研究等领域对测试与控制设备需求的日益扩大和性能要求的不断提升, 以及国家层面自主可控的战略发展, 提高软硬件国产化率成为了迫切需求<sup>[1-2]</sup>。银河麒麟是我国自主研发的操作系统, 具有较强的易用性、稳定性和安全

性, 支持嵌入式设备、桌面计算机和服务器。然而, 由于起步较晚, 银河麒麟操作系统的软件生态相对薄弱, 对测试仪器硬件驱动还不够全面, 难以覆盖测试系统所需。因此, 研究面向银河麒麟操作系统的 PXI 仪器控制软件, 对于推动全国产化的测试系统生态发展具有重大意义。

PXI/PXIe 仪器是模块化的虚拟仪器, 其通信传输

收稿日期: 2024-10-27; 修回日期: 2024-11-20。

作者简介: 胡宇 (1981-), 男, 大学本科, 工程师。

引用格式: 胡宇. 基于银河麒麟操作系统的 PXI 仪器控制软件设计[J]. 计算机测量与控制, 2025, 33(3): 131-137.

基于 PCI/PCIe 总线<sup>[3]</sup>, 通过计算机软件操控、显示实现仪器功能。PXI 仪器自诞生以来就得到迅猛发展和推广, 广泛应用于电子、能源、交通等各个行业的测试、控制、采集系统中<sup>[4-5]</sup>。目前, 国内外在 PXI 设备驱动控制技术方面的研究、应用主要基于国外主流操作系统, 诸如 Windows、Linux 等<sup>[6]</sup>, 而在国产操作系统上的研究相对较少。国外方面, 在 PXI 设备驱动的开发与优化方面积累了丰富的经验, 并形成了较为完善的驱动开发框架和标准<sup>[7]</sup>。测控行业内全球著名的公司均开发了 Windows 操作系统下的 VISA 库, 且已十分成熟, 包括 NI VISA 和 KEYSIGHT VISA 在国内外得到了广泛的应用<sup>[8-10]</sup>。VISA-Virtual Instrument Software Architecture, 即虚拟仪器软件结构, 是 VXI plug&play 联盟制定的测控软件 I/O 接口标准, 及其规范的总称<sup>[11]</sup>。VISA 库提供用于仪器编程的标准 I/O 函数库<sup>[12]</sup>, 通常以动态链接库的方式存在。VISA 动态库驻留在计算机操作系统内, 是计算机与仪器的标准软件通信接口, 计算机通过调用 VISA 库来控制各类仪器, 其中也包括了 PXI/PXIe 仪器。VISA 库是 PXI 仪器控制软件的基础, 但不能在银河麒麟操作系统上使用。国内方面, 随着国产操作系统的逐步普及和持续发展, 越来越多的学者和企业开始关注在国产操作系统上进行 PXI 设备驱动控制的研究<sup>[13]</sup>, 但整体还不够成熟全面, 仍处于起步阶段。

## 1 软件系统架构及原理

银河麒麟操作系统是一款开源操作系统, 基于 Linux 内核。Windows 系统和 Linux 操作系统支持的文件格式不同, Windows 操作系统下的可执行程序以及动态链接库不能直接复制到 Linux 操作系统下的运行。现有 Windows 系统下的 VISA I/O 库是“.dll”格式, 可通过静态和动态调用。Linux 支持动态链接库为“.so”格式不能调用“.dll”, 需要将源码在系统下进行重新编译才能使用。目前, NI VISA 和 KEYSIGHT VISA 并没提供 Linux 版本程序和源代码, 导致 VISA I/O 库不能在 Linux 和银河麒麟操作系统下使用。

为提高系统安全性和稳定性, Linux 将用户程序和内核程序进行了隔离, 分离成两种运行环境, 即用户空间和内核空间。前者一般包含图形界面、应用程序, 后者主要负责内存管理、进程管理、文件系统管理和硬件底层驱动等内容。内核空间处于用户空间和硬件之间, 因此用户空间不能直接访问硬件。为保护内核安全, 用户空间程序同样不能直接访问内核空间的数据, 只能通过系统调用内核空间给出的标准接口来访问, 上层软件通常使用 lib 库文件来封装这些系统调用。类似 Win-

dows 操作系统下的 VISA I/O 库, 便是系统调用的封装。针对银河麒麟操作系统, 本研究拟参考 VISA 规范开发 VISAlib 库, 对仪器功能和使用特点对内核空间接口进行了进一步封装, 最终为用户程序提供各种仪器基本操作方法。VISAlib 库作为应用程序和底层驱动的桥梁, 是 PXI 仪器控制软件的核心, 也是本文的重点研究对象。

Linux 系统通过设备驱动程序完成对硬件设备的控制和管理, 驱动程序为用户程序提供访问硬件设备的接口。驱动程序包含用户驱动程序和底层驱动程序, 用户驱动程序属于应用程序, 实现仪器功能函数, 通过调用 VISAlib 向底层驱动传输控制指令和数据交互。底层驱动运行在内核模式下, 是内核的一部分。银河麒麟操作系统本身提供通用的 PCI 驱动, 可作为 PXI 仪器的内核底层驱动。为简化硬件设置操作复杂度, 在 PCI 设备通用驱动框架中, 所有设备在 Linux 内核中都被抽象为文件, 通过 read ()、write () 等文件操作接口来访问硬件设备。这种统一的硬件、文件访问方式提高了 Linux 系统下的软件可移植性和可扩展性。PXI 设备作为 PCI 总线设备, 驱动程序同样遵循 Linux 下的 PCI 设备驱动框架, 通过相应的数据结构和方法来实现对硬件设备的控制和管理。

PXI 仪器种类多样, 一般包括示波器、万用表、函数发生器、直流电源、交流电源、开关、功率计、频谱分析仪、射频信号源、计数器、上变频器、下变频器和数字化仪等 13 种类型。其中变频器通常内置于射频微波仪器内部, 不以独立仪器的形态出现。因此, 本文将上变频器和下变频器整合到频谱分析仪、射频信号源中, 共计归纳为 11 种 PXI 仪器类型。不同类型仪器功能的、参数设置存在较大差异, 需开发不同的界面程序, 本文将根据上述仪器的分类, 设计 11 种仪器控制界面程序。同时, 为使界面程序适应同一类型中不同生产厂家的仪器, 本研究也设计了 11 种仪器的类驱动程序, 使界面程序与仪器本身的功能驱动程序解耦, 实现仪器的可互换。类驱动对功能驱动的管理、映射技术也是本研究的关键技术之一。

综上所述, 基于银河麒麟操作系统的控制软件系统架构包括用户界面程序、类驱动程序、仪器功能驱动程序和 VISAlib 库等, 各部分根据 Linux 系统软件架构划分, 驻留于用户空间之中, 各程序从上至下依次调用, 最后通过 VISAlib 库调用内核空间的系统底层驱动框架, 实现仪器硬件的操控, 如图 1 所示。

## 2 软件系统设计与实现

### 2.1 用户界面程序设计

用户界面程序是 PXI 仪器的人机交互界面, 也称

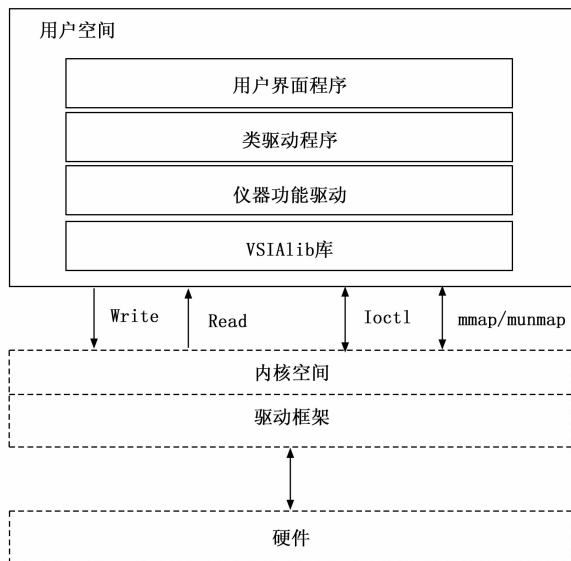


图 1 软件系统架构图

为仪器软面板。Linux 系统下图形界面一般采用 QT 和 Python 软件开发，Python 语言具有良好的跨平台特性，可实现程序在 Windows、Linux、银河麒麟等操作系统之间的无缝移植。Python 开发语言具有丰富的开源扩展库支撑，包括数据处理、数据可视化等，同时，还可直接调用其他编程语言开发的库程序，无需再次编译。基于上述优点，本研究选择 Python 语言开发开发 PXI 仪器用户界面程序，采用 tkinter 模块实现 GUI 编程，使用文本输入框、按钮控件、消息控件和画布控件等完成人机信息交互。

针对上文的 11 种仪器类型，本研究设计了相应的 Python 界面程序，分别为：ScopeUI.py、DmmUI.py、FgenUI.py、DCPwrUI.py、ACPwrUI.py、SwitchUI.py、PwrMeter.py、SpecAnUI.py、RFsiggenUI.py、CounterUI.py 和 Digitizer.py。仪器用户界面程序实现功能包括仪器管理、仪器功能参数配置、仪器测量控制、测量结果显示和波形绘制等。仪器管理功能主要包含仪器选择、仪器地址输入和仪器初始化等功能。仪器功能参数配置和测量控制等功能则根据仪器自身功能特点进行设计。下面以万用表为例进行说明，万用表的功能配置包括：直流电压测量、交流电压测量、直流电流测量、两线电阻测量和频率测量等，在界面中使用组合选择框控件实现功能选择。万用表参数属性配置包括量程、触发源、触发延时等参数，通过文本输入框控件输入设置值或通过组合选择框控件选择设置参数。万用表测量控制功能包括：测量时间设置、启动测量、返回测量值等，通过按钮控件进行控制。各按钮控件与万用表类驱动函数进行绑定。万用表测量结果显示通过一个文本输入框显示。

用户打开界面程序时，软件会扫描系统中该类型的驱动程序，并将仪器型号显示在组合选择框中，选择仪器型号后，软件则加载相应型号的仪器驱动。用户选择仪器总线类型并输入仪器总线地址，点击“初始化”，软件则打开仪器进行初始化操作。用户界面程序的输入、输出控件与仪器类驱动函数和参数相对应和关联，用户设置参数将被传递到类驱动函数，当按下控制按钮时则调用相应的类驱动函数，进行仪器参数设置，按压“启动测量”按钮则开始测量，按压“返回测量值”则读取测量结果并显示到界面上。测量完毕后，用户点击“关闭”按钮，软件则关闭仪器，释放加载的驱动程序。

驱动调用的总体思路为：界面程序对驱动程序的调用通过 ctypes 库实现，运行“lib = cdll.LoadLibrary”加载驱动程序动态链接库 so 文件，在 python 代码中运行“lib.xxx\_function”调用驱动程序动态链接库中的功能函数。

## 2.2 类驱动程序设计

类驱动程序用于实现界面程序与仪器的无关性，在不改变界面程序代码的情况下实现同类仪器的互换。

类驱动和仪器功能驱动包含相同功能的接口函数，通过函数前缀进行区别，类驱动函数前缀为仪器类别，仪器驱动函数前缀为仪器型号。仪器型号、厂家等信息由配置工具写入配置文件，用于指示当前使用的仪器。应用程序调用类驱动接口函数，类驱动读取配置文件，加载对应仪器驱动，并运行驱动内相同功能的接口函数，实现仪器的控制。更换仪器时只需改动配置文件，无需更改应用程序代码，从而实现仪器的可互换。

类驱动设计的关键包括两点：1) 上层应用程序调用类驱动函数时映射执行仪器功能驱动函数。2) 是类驱动函数要覆盖不同厂家同一类型的仪器功能。

针对关键点一，实现思路为：通过配置文件存放需要使用的仪器厂家、型号等仪器信息，类驱动打开配置文件读取仪器信息，根据厂家和型号使用 dlopen() 函数加载对应的仪器功能驱动库装入内存，再通过 dlsym() 函数将内存中的仪器功能驱动函数指针和变量地址传给类驱动，完成仪器功能驱动到类驱动的映射。本研究类驱动程序基于 C/C++ 语言开发，程序架构如图 2 所示。

针对关键点二，本研究参考 IVI 规范设计类驱动函数，包括通用接口函数和仪器类驱动函数。通用接口函数适用于所有仪器种类，主要实现仪器初始化和关闭等基本操作。其中，init() 用于仪器初始化，通过仪器名称（使用总线类型和地址表示）打开仪器，获取并返回仪器句柄。close() 用于关闭仪器，断开连接，清除

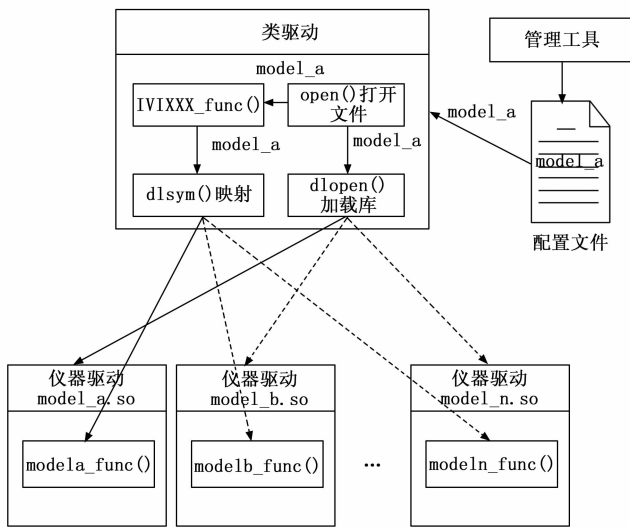


图 2 类驱动架构

内存，释放句柄。各类仪器的类驱动因功能差异使得其接口函数各不相同，采集、测量类仪器一般包括退出、触发设置、采集参数设置、通道设置、测量和读取结果等。针对 11 种仪器，本研究开发的类驱动分别为 IviScope、IviDmm、IviFgen、IviDCPwr、IviACPwr、IviSwitch、IviPwrMeter、IviSpecAn、IviRFsiggen、IviCounter 和 IviDigitizer，以万用表为例，IviDmm 类驱动函数设计如下：

- 1) IviDmm\_Abort () 函数用于停止测试并退出测量。
- 2) IviDmm\_ConfigureMeasurement () 函数用于配置测量属性，设置属性包括测量功能、量程、分辨率等。
- 3) IviDmm\_ConfigureTrigger () 函数用于配置触发属性，包括选择触发源和设置触发延迟时间。
- 4) IviDmm\_Fetch () 函数用于获取测量值，并设置超时等待时间，若超时未获取到测量值，则返回超时状态。
- 5) IviDmm\_Initiate () 函数用于启动测量开始测试。
- 6) IviDmm\_IsOverRange () 函数用于检测测量值是否超量程，传入量程值，返回是否超量程。
- 7) IviDmm\_Read () 函数用于启动测量并获取测量值，并设置超时等待时间，若超时未完成测量，则返回超时状态。
- 8) 上述函数分别调用仪器功能函数中对应的 Abort ()、ConfigureMeasurement ()、ConfigureTrigger () 等函数实现仪器控制功能。

### 2.3 仪器功能驱动设计

PXI (Pci Extensions For Instrument) 仪器是 PCI

总线在面向仪器领域的扩展<sup>[14]</sup>，其通信和数据传输均基于标准 pci 总线实现。PCI 总线设备 PCI BAR (Base Address Register) 空间是一组寄存器，用于映射 PCI 总线设备的内存地址或 I/O 端口，操作系统和其他设备通过访问配置寄存器，可以确定该设备所需的资源，再将设备资源映射到系统地址空间，应用程序便可通过访问地址空间与设备通信<sup>[15]</sup>。因此，PXI 仪器的驱动控制主要基于 BAR 空间的映射和数据传输操作<sup>[16]</sup>，向寄存器发送指令进行仪器控制，从寄存器读取数据获得测量结果。本设计中对寄存器的读写操作通过开发 VISAlib 库实现。

仪器功能驱动需要实现类驱动中规定的各种仪器操控功能，基本实现思路为：调用 VISAlib 库中的 viOpen () 函数获取仪器句柄，通过 viIn32 () 函数向特定的寄存器地址发送控制字，完成仪器参数配置和启动测量，再通过 viOut32 () 函数从内存空间读取仪器状态字或测量结果数据。下面以 SC58411PXI 万用表模块为例，说明本设计中的仪器功能驱动各函数实现过程。SC58411 主要使用的寄存器包括 REG\_DATAIO: 0x001C、REG\_TRIGGER: 0x0040、REG\_TP6C595: 0x0060 和 REG\_SYSTEM\_CTRL: 0x0018，分别用于数据读取、触发使能、参数配置和运行控制，这些寄存器均位于 BAR0 空间。各功能函数使用型号“SC58411”作为前缀名，主要函数设计如下：

- 1) SC58411\_Abort ()，调用 viOut32 () 函数先向 REG\_SYSTEM\_CTRL 寄存器写停止控制字，再向 REG\_TRIGGER 写 0 关闭触发。
- 2) SC58411\_ConfigureMeasurement ()，调用 viOut32 () 函数向 REG\_TP6C595 写入功能和量程配置控制字。
- 3) SC58411\_Fetch ()，根据数据长度循环调用 viIn32 () 函数从 REG\_DATAIO 读取测试数据，直至读取全部数据，并进行实数转换。
- 4) SC58411\_Initiate ()，调用 viOut32 () 函数先向 REG\_SYSTEM\_CTRL 寄存器写启动测试控制字，再向 REG\_TRIGGERL 寄存器写启动触发控制字。
- 5) SC58411\_Read ()，先调用 SC58411\_Initiate () 启动测试，再调用 SC58411\_Fetch () 获取测量值。

### 2.4 VISAlib 库设计

现有 Windows 操作系统中的 VISA 库提供了丰富的仪器操作函数，形成了标准 I/O 库函数，涵盖 GPIB、VXI、PXI 和 LXI 等仪器总线，提供统一的输入输出接口函数。消息基和寄存器基是控制虚拟仪器两种基本方式，VIAS 库通过调用各类总线接口驱动向仪器传输字

符串消息或访问寄存器、内存空间，实现仪器的控制。目前，银河麒麟等国产操作系统下尚未形成统一、标准的 VISA 库，本文所述的 VISAlib 库便是适用于国产操作系统的仪器标准 I/O 库。

VISAlib 库是应用程序和 Linux 系统底层驱动之间的桥梁，包括资源管理函数和仪器控制函数。资源管理类函数用于管理仪器操作包括：打开、关闭、设置/获取属性等，仪器控制类用于仪器操作控制指令交互和数据传输。两类函数之间通过“DEVICE”结构传递数据，如图 3 所示，为实现多台仪器的操作，利用数组 Device [n] 存放“DEVICE”数据，n 为设备序号。

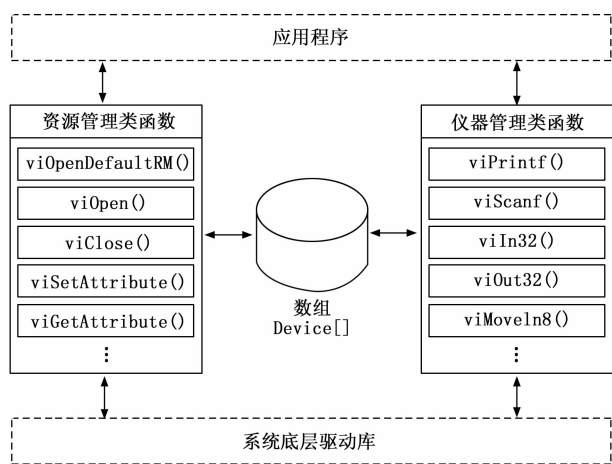


图 3 VISAlib 库程序架构图

Device [] 结构体数组包含 Opened、MmapAddr 等成员变量，Opened 用于指示仪器是否打开，MmapAddr 用于记录 BAR 空间映射的内存地址<sup>[17]</sup>。规划 256 个仪器总数量，设置变量 DevNo（取值范围为 0~256），用于记录仪器数量。DevNo 初始值置 0，每完成一个仪器的 BAR 地址空间映射则视为打开一个仪器，此时将变量 DevNo 加 1，并将 Opened 置 1，将映射后的地址赋给 MmapAddr。DevNo 作为该仪器后续操作、关闭的唯一句柄，并作为 Device [] 数组下标，获取对应的仪器信息。关闭仪器时将 Opened 置 0，释放 MmapAddr 地址<sup>[17]</sup>。当 DevNo 计数超过 256 时，则将 DevNo 置 0，从小到大依次查找 Opened 为 0 的编号，找到的第 1 个编号作为新的仪器句柄<sup>[17]</sup>。

VISAlib 库实现功能主要包括仪器打开关闭功能和数据读写功能等，对应 viOpen、viClose、viIn8 \ 16 \ 32 和 viOut8 \ 16 \ 32 函数<sup>[18]</sup>，各函数实现方式如下。

1) viOpen 函数：

viOpen 函数用于根据仪器名称打开指定的仪器，返回仪器操作句柄<sup>[19]</sup>，其本质为将设备 PCI 总线地址映射到用户空间内存。为了更高效的块操作，使用

mmap 系统调用来把设备地址映射到用户层，再直接读写。首先使用系统底层 open () 函数获取 PXI 设备文件句柄，例如在银河麒麟操作系统下某 PXI 仪器总线地址为“0001: 00: 07.0”，打开设备的方法为使用 open () 函数打开“/sys/bus/pci/devices/”系统路径下“0001...00...07.0/”文件夹中的 resource 文件，将返回句柄保存到“fd”。然后使用 mmap 函数映射设备地址到用户程序，即“mmap (0, map\_size, PROT\_READ | PROT\_WRITE, MAP\_SHARED, fd, target\_base)”。mmap 映射的最小单位是页面 (page)，需要先获取当前系统的页面大小，进行页对齐，可使用宏 \_SC\_PAGE\_SIZE 获取页面大小。Mmap 返回的虚拟地址即可以用来读写设备数据。

因 PCI 总线具有 6 个 BAR 空间，需要依次打开 6 个 resource 文件保存句柄 fd，获取 BAR 基址，获取 BAR 空间大小，依次调用 mmap 函数映射 6 个 BAR 的内存地址 map\_base，并存入数组 MmapBases []，将 Device [n] 中的仪器地址指针指向数组 MmapBases []，n 则为 viOpen 函数返回的仪器句柄。

2) viClose 函数：

viClose 函数用于关闭资源管理器和仪器设备<sup>[19]</sup>，实现思路为，使用 munmap () 来解除地址映射，并调用 close () 依次关闭 6 个 resource 文件句柄。

3) viIn8 \ 16 \ 32 函数：

viIn8 \ 16 \ 32 函数用于从仪器读取 8 位 \ 16 位 \ 32 位数据，实现过程为：从 Device [VI] 仪器地址指针获取 6 个 BAR 的映射地址，根据入参 space 值确定需要读取的 BAR 地址 map\_base；调用 readdw 函数读取地址为 map\_base 的数据。

4) viOut8 \ 16 \ 32 函数：

viOut8 \ 16 \ 32 函数用于从仪器发送 8 位 \ 16 位 \ 32 位数据，实现过程为：从 Device [VI] 仪器地址指针获取 6 个 BAR 的映射地址，根据入参 space 值确定需要写入的 BAR 地址 map\_base；调用 writedw 函数向地址 map\_base 写入数据。

为防止仪器被多个线程同时操作，产生无法预计逻辑混乱与冲突。VISAlib 在实现仪器操作 I/O 函数（打开、读写、关闭等）时，需首先调用 pthread\_mutex\_lock () 函数，保护该段代码在执行时不会被其他线程调用执行，避免多线程同时对设备进行访问，代码执行完成后调用 pthread\_mutex\_unlock () 函数去除线程锁定状态。

3 实验结果与分析

为验证上述 PXI 仪器驱动控制方法的有效性，以

万用表模块为例进行实物测试验证。实验搭建了一套 PXI 平台,平台中包含一台 PXI 机箱 SC58505、一张 PXI 控制器 SC58109 和一张 PXI 万用表 SC58411,使用电压源、电流源和电阻作为测试激励源。在 SC58109 上安装银河麒麟操作系统,将 VISAlib 库编译成动态链接库 visalib.so 文件,万用表类驱动编译为 ividmm.so, SC58411 功能驱动编译为 spaceon\_sc58411.so。SC58411 应用程序通过调用 ividmm.so 库函数,实现仪器初始化、功能配置、量程配置和测量等功能。应用程序界面和实验平台如图 4 所示。



图 4 PXI 万用表控制实验

实验分别对上述功能进行验证,并控制 SC58411 进行电压、电流和电阻参数进行测量,实验步骤、方法如下。

**仪器初始化验证:**将 PXI 控制器和待测万用表模块插入 PXI 机箱,在银河麒麟终端中运行 python3 DmmUI.py,打开程序控制界面;在界面“选择仪器”下拉框中选中陪测驱动的类型 SC58411;在资源名称中选择相应总线类型 PXI,填写 PXI 总线号及设备号;点击“初始化”按钮,观察程序控制界面面板“显示文本”和银河麒麟终端输出的打印信息,“显示文本”输出“仪器初始化成功”,且终端无错误信息报出。

**参数设置功能验证:**在界面“设置—功能”处下拉选择陪测万用表支持的功能,如“DC\_VOLTS”;在界面“设置—量程”处输入陪测万用表支持的量程,如 10 V;在界面“设置—触发源”处选择陪测万用表支持的触发源,如“IMMEDIATE”;在界面“设置—触发延时”处设置触发延时时间,如“—1”;点击界面“配置属性”按钮,观察“显示文本”是否显示设置成功,且正确返回当前设置值;改变上述配置参数,分别测试电压、电流、电阻测量功能参数设置。

**测量功能验证:**在界面“测量—最长测量时间”处填入“4 000”;设置直流电源输出 6.6 V 电压,并连接到万用表;依次点击“启动测量”“返回测量值”按钮,观察界面显示测量结果,误差在预期范围内;改变测量功能及参数,依次对电流源输出电流和电阻器电阻值进行测试,界面应能显示测量结果,且误差应在预期范围内。

相关实验数据如表 1 所示。

表 1 PXI 万用表控制实验数据

验证内容	要求	输入/设置	期望值	测量值	结论
仪器初始化	成功打开仪器并返回句柄	03~09.0	仪器初始化成功	仪器初始化成功	通过
电压量程设置	设置成功	10 V	10 V	10 V	通过
电压测量	测量值与输入值一致	6.6 V	6.47~6.73 V	6.68 V	通过
电流量程设置	设置成功	3 A	3 A	3 A	通过
电流测量	测量值与输入值一致	1.5 A	1.48~1.52 A	1.502 A	通过
电阻量程设置	设置成功	1 000 000 $\Omega$	1 000 000 $\Omega$	1 000 000 $\Omega$	通过
电阻测量	测量值与输入值一致	10 000 $\Omega$	9 900~10 100 $\Omega$	9 967.6 $\Omega$	通过

实验数据表明软件能够对被试 PXI 模块进行初始化、参数设置和测量控制,电压、电流和电阻误差在预期范围内,其中包括激励信号输出误差和测量误差。

#### 4 结束语

PXI/PXIe 仪器已成为当前虚拟仪器的主流产品<sup>[20]</sup>,针对当前各行业对计算机和操作系统国产替代使用的迫切性,以 PXI/PXIe 为代表的虚拟仪器在国产生态下的驱动控制问题亟待解决。文章针对银河操作系统,研究了其内核 Linux 框架的 PCI 总线驱动技术,提出了基于 pci 总线寻址和配置空间内存映射的方法,进而完成操作系统应用程序与 PXI/PXIe 仪器间的数据传输,达到仪器驱动控制的目的。并开发了相应的 I/O 库和界面程序,成功实现了仪器初始化、量程设置和信号测量功能,对研发国产化自动测试系统具有一定的借鉴意义。目前,本文的研究还处于原理样机阶段,为实现工程化应用还需进行更多的完善,包括用户空间和内核空间软件功能的合理划分,扩展 LXI、GPIB、VXI、RS232、USB 等更多的仪器总线,形成完备的国产化 VISA I/O 库,以及开发完整的总线、资源管理软件等。

#### 参考文献:

- [1] 唐永学,朱桂梅,汤浩.基于龙芯平台的无源检测装置设计与实现[J].计算机测量与控制,2021,29(4):5-9.
- [2] 彭晓飞,李杰,刘路扬,等.国产航天元器件自主可控应用验证方法研究[J].计算机测量与控制,2022,30

- (5): 268-273.
- [3] 汪新舜, 吴星云, 沈辉, 等. 基于虚拟仪器的航天器外热流软件设计 [J]. 计算机测量与控制, 2022, 30 (7): 166-172.
- [4] 史峰, 何晓凤. 基于PXI平台的通用火化机质量检测系统设计 [J]. 计算机测量与控制, 2021, 29 (4): 80-85.
- [5] 李光, 潘晓彤, 汪灏, 等. 批生产卫星的桌面电接口快速自动化测试系统设计 [J]. 计算机测量与控制, 2021, 29 (4): 86-89.
- [6] 赵昕, 郭恩全, 李小杰. LINUX系统下 GPIB 驱动优化设计与实现 [J]. 计算机测量与控制, 2020, 28 (3): 163-167.
- [7] 徐利, 张良国. 基于VISA\_COM的测量仪器通用测试系统的开发 [J]. 计算机测量与控制, 2015, 23 (10): 10.
- [8] 吴宗卓. 基于NI-VISA的野外靶场设备远程控制系统软件开发设计 [J]. 计算机测量与控制, 2022, 30 (6): 138-143.
- [9] 刘磊, 付存文, 张光山. 基于VISA的PXI设备访问新方法 [J]. 计算机测量与控制, 2016, 24 (7): 162-163.
- [10] 彭映成, 孙震宇, 邢园丁, 等. 基于LabWindows/CVI与TekVISA的Tek示波器远程测控软件设计 [J]. 测控技术, 2018, 37 (2): 123-125.
- [11] 刘阳, 罗显志, 周杨罗, 等. 基于FPGA的IEEE488接口协议实现及驱动程序开发 [J]. 计算机测量与控制, 2022, 30 (8).
- [12] 苟江川, 朱峰, 刘光辉, 等. 基于虚拟仪器的电磁干扰近场测试系统设计 [J]. 计算机测量与控制, 2015, 23 (1): 16-18.
- [13] 刘康丽, 谷静, 杜影, 等. 中标麒麟系统下基于QT的PXI仪器软件开发 [J]. 计算机测量与控制, 2019, 27 (10): 159-162.
- [14] 孟珞珈, 郭元兴, 盛强强, 等. 模块类网络安全产品的智能测试系统设计 [J]. 计算机测量与控制, 2024, 32 (1): 37-44.
- [15] 胡广浩, 阮福明, 赵希昉, 等. 基于VxWorks的CPCI多通道卡驱动设计与实现 [J]. 计算机测量与控制, 2022, 30 (5): 209-214.
- [16] 肖文君, 刘万松, 刘伍丰, 等. 基于WDF的PCIe总线驱动程序设计与实现 [J]. 测控技术, 2015, 34 (7): 101-104.
- [17] 胡宇, 唐小峰, 马雅男, 等. 一种PXI/PXIe仪器的控制方法及存储介质 [P]. 中国: CN116719762A, 2023. 09. 08.
- [18] 蒋荣华, 陈光祜. 虚拟仪器软件结构—VISA函数的设计与实现 [J]. 测控技术, 2006, 25 (3): 63-65.
- [19] 卫红春, 杨浩前, 李宥谋. 虚拟仪器软件架构VISA资源管理的实现原理研究 [J]. 测控技术, 2015, 34 (8): 96-99.
- [20] 孟珞珈, 郭元兴, 盛强强, 等. 模块类网络安全产品的智能测试系统设计 [J]. 计算机测量与控制, 2024, 32 (1): 37-44.
- (上接第112页)
- [5] 马宏伟, 张焯, 王鹏, 等. 多机械臂煤矸石智能分拣机器人关键共性技术研究 [J]. 煤炭科学技术, 2023, 51 (1): 427-436.
- [6] 冯凌云, 曾祥苹, 李琴, 等. 基于输送带及视觉跟踪的机器人分拣包装工作站仿真 [J]. 包装工程, 2023, 44 (13): 227-235.
- [7] 贾超广, 肖海霞. 机器视觉的食品包装快速分拣系统 [J]. 食品工业, 2021, 42 (5): 276-279.
- [8] 陈亚琳, 宋增禄. 袋装食品快速自动分拣系统设计 [J]. 食品工业, 2021, 42 (4): 299-302.
- [9] 贾庆轩, 段嘉琪, 陈钢. 机器人在轨装配无标定视觉伺服对准方法 [J]. 航空学报, 2021, 42 (6): 628-637.
- [10] 原劲鹏, 葛连正, 李德伦. 双臂空间机器人闭链系统的协同柔顺控制策略研究 [J]. 空间控制技术与应用, 2023, 49 (2): 42-50.
- [11] 林洪, 邓艳. 改进贪心算法求解扩展简化折扣 {0-1} 背包问题 [J]. 西南师范大学学报: 自然科学版, 2022, 47 (11): 63-71.
- [12] 张文玥, 娄小平, 陈福笛. 移动机器人多传感器融合定
- 位仿真研究 [J]. 计算机仿真, 2023, 40 (3): 436-441.
- [13] 丁伟利, 杨庆, 张恒. 基于虚拟仿真的龙门机器人板材智能分拣系统及方法 [J]. 高技术通讯, 2023, 33 (6): 630-642.
- [14] 刘诚, 孙远升, 花军, 等. 基于贪心算法及局部枚举策略的人造板排样方案研究 [J]. 木材工业, 2021, 35 (6): 55-61.
- [15] 赵明阳, 杨揆, 王璟旻, 等. 基于贪心算法的气象修正表自动分区拟合算法设计 [J]. 火炮发射与控制学报, 2022, 43 (4): 75-79.
- [16] 薛旭升, 杨星云, 齐广浩, 等. 煤矿带式输送机分拣机器人异物识别与定位系统设计 [J]. 工矿自动化, 2022, 48 (12): 33-41.
- [17] 肖瑶星, 刘立新, 胡柳, 等. 基于三维模型的食品分拣机器人抓取控制方法 [J]. 食品与机械, 2023, 39 (4): 77-82.
- [18] 廉胤东, 刘彦汝, 杨启帆, 等. 基于信息物理系统的多分拣移动机器人调度策略 [J]. 控制与决策, 2023, 38 (1): 281-288.