文章编号:1671-4598(2026)01-0009-07 DOI:10.16526/j.cnki.11-4762/tp.2026.01.002 中图分类号:TP274 文献标识码:A

基于 RTX 的飞行陀螺性能测试系统研制

周 强,闫锐桐
(北京航空航天大学 自动化科学与电气工程学院,北京 100191)

摘要:针对飞行陀螺性能测试系统高实时性的需求,设计并实现了基于 PCI 总线架构与 RS422 总线的硬件系统,同时提出基于 RTX 的测试软件开发方案;测试系统依托 Windows + RTX 平台,兼具良好实时性与 Windows 操作系统在测控软件开发上的优势;系统采用模块化、层次化设计,分为驱动层、逻辑层和用户层;驱动层与底层对接,负责与底层 FPGA 开发相应驱动;逻辑层承担业务处理,对各个板卡接口函数进行封装;用户层实现与用户交互,构建稳定可靠的上层界面程序以满足用户需求;最后,在研发的飞行陀螺测试系统装置上开展功能与性能测试;经验证,实时性响应迅速,能够满足测试需求,为未来测试设备研发提供了更多参考。

关键词:Windows;RTX;实时性;驱动开发;接口函数

Development of Flight Gyroscope Performance Testing Systems Based on RTX

ZHOU Qiang, YAN Ruitong
(School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China)

Abstract: In order to meet the high real-time requirements of flight gyroscope performance testing systems, a hardware testing system based on PCI bus architecture and RS422 bus is designed and implemented, and a test software development solution based on RTX is proposed. Through a Windows+RTX platform, the system has good real-time performance while maintaining the advantages of Windows operating system in measurement and control software development. The system adopts a modular and hierarchical design and consists of 3 layers: a driving layer, which connects with the bottom layer for the development and drivers of the underlying FPGA, a logical layer responsible, which is responsible for business processing and encapsulating the interfaces of various boards, and a user layer, which interacts with users and constructs a stable and reliable upper-layer interface program to meet user needs. Finally, functional and performance tests are conducted on the developed flight gyroscope system. After verification, the system meets test requirements and provides more references for the development of test equipment in the future.

Keywords: Windows; RTX; real-time; driven development; interface functions

0 引言

随着全球航天事业的蓬勃发展,陀螺组件测试技术研究备受关注。美国早在 20 世纪便在先进卫星项目中应用高精度陀螺组件与成熟测试体系,其飞控测试系统多采用硬实时系统 VxWorks,但存在成本高、开发难的问题。国内自“东方红一号”发射后,卫星测试技术从依赖人工与简单设备的单一化模式,逐步向高度自动化、集成化转变,取得显著突破[1]。当下,我国研发重点聚焦通用兼容性、冗余设计与实时响应,以应对复杂测试需求。但卫星陀螺组件测试面临传统设备与型号同步研发致资源消耗大、平台式惯性导航系统硬件复杂且依赖进口等问题[2]。在实时操作系统选择上,国内研究大多是基于 Linux 的 RTLinux,其实时性略差且图形界面体验相对 Windows 系统有待完善[3]。

飞行陀螺性能测试系统基于 RTX+ Windows 软实时系统开发测试软件,实现了测试软件与硬件的有机融合。该测试系统通过融合硬实时系统的实时性与 Windows 的图形化界面优势[4],在保持软件良好的可维护性与可升级性的同时,解决了测控系统的高实时性需求。在此基础上,本文总结了实时操作系统下测试软件开发的技术难点,为未来军用测试设备研发奠定坚实

收稿日期:2024-11-22; **修回日期**:2025-02-25。
作者简介:周 强(1972-),男,博士,副教授。
引用格式:周 强,闫锐桐.基于 RTX 的飞行陀螺性能测试系统研制[J].计算机测量与控制,2026,34(1):9-15.

基础[5]。

1 测试系统硬件平台设计

1.1 总体设计

某型飞行陀螺性能测试系统架构及组成框图如图 1 所示，整个测试系统通过测控单元、三轴转台、装夹装置及用户上位机等组件的紧密协同工作，实现对飞行陀螺的全面测试。

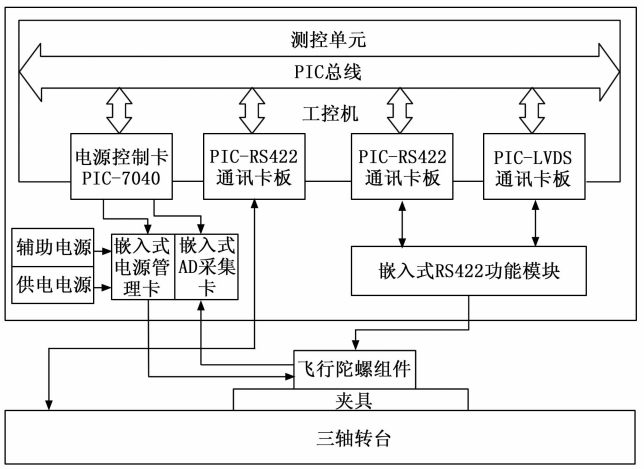

图 1 飞行陀螺性能测试系统组成图

测控单元是整个测试系统的中枢，将产品和外设相联系，实现测试中的交互部分。测控单元具有以下功能：支持 6 路独立电源管理，监控电压、电流及功率；集成指令控制，实现产品开关机与转台飞行模式自动化；采集模拟/数字量，通过 422/LVDS 接口与产品及转台高效通讯。

三轴转台组成包括内框、中框和外框及转台控制系统，能够提供灵活的运动控制平台，模拟复杂环境下的飞行姿态，其主要功能有：实现独立或联动框架运动自控，接收计算机反馈的转台信息，在仿真模式下按要求的响应实际转发转台信息至测控单元。用户上位机提供 1 ms 的周期方波作为同步信号，转台在同步信号上升沿锁存三轴位置，所存时间小于 10 μs，可以确保时间同步，随后于 1 ms 内应答并完成数据打包发送。

装夹装置连接飞行陀螺组件和三轴转台，确保飞行陀螺组件在安装过程中的精准定位，保障了飞行陀螺组件在模拟环境下的稳定运动。

1.2 测控单元设计

测控单元设计以工控机为核心，集成自主研发的 PCI 总线接口 RS422 通信卡、LVDS 通讯卡和电源控制卡，另外系统还配置了嵌入式 RS422 功能模块，实现飞行陀螺的性能测试。

嵌入式电源管理卡经 OC 指令，实现陀螺组件多路供电管控，以继电器操控其他组件的电源输出。其配置 18 个 TTL 输入接口、18 个 TTL 输出通道的 7040 板卡输出控制信号，还设有 16 路光耦隔离与 12 路 OC 输出。鉴于 PCI-7040 采集范围为－10 V 至 10 V，产品需 20 V 至 120 V 可调供电、电流驱动不小于 20 A，故而需对电压、电流信号予以处理并隔离输出。

嵌入式 RS422 功能模块按控制模块分为实时监控与数据采集两个子模块。实时监控模块可以实现转台控制与脉冲数据采集，支持高精度差分脉冲处理与同步脉冲发送，其原理图如图 2 所示。数据采集子模块主要负责遥测信号采集，确保高测量精度与分辨率，其原理图如图 3 所示。

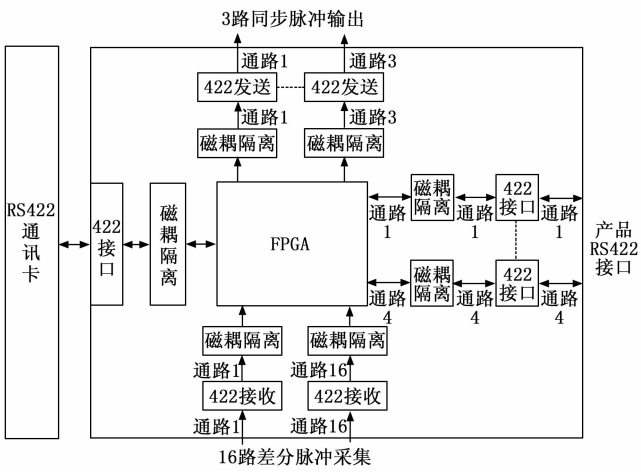

图 2 实时监控功能子模块

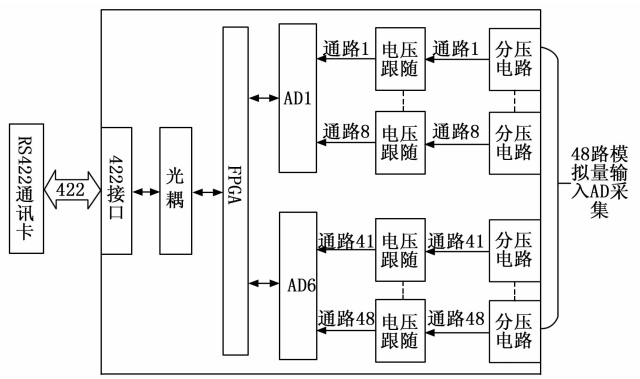

图 3 数据采集子模块

为满足实时性需求，本研究优化了两块 RS422 通信板卡的硬件和驱动设计，分别负责监控三轴转台和监控陀螺组件，通过双卡板并行工作的方式，高效传输三轴转台与陀螺组件数据，原理图如图 4 所示。

此外，测控单元中还配置了 4 通道 LVDS 通信卡，以 FPGA 为核心支持 LVDS 协议，具备高速数据传输与缓存能力。该设计不仅满足了当前的测试需求，还预留了未来接口升级的空间，增强了系统的灵活性与可扩展性。

2 测试系统驱动程序设计

测试系统的软件自底向上分为三个层次，第一层为

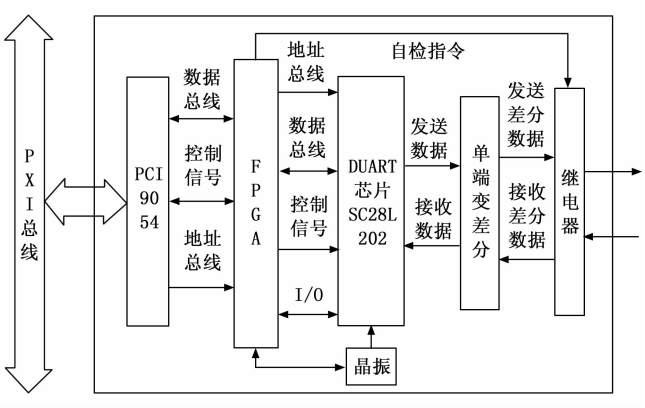

图 4　RS422 通信卡原理框图

内核级设备驱动程序，直接与 FPGA 交互，实现硬件控制；第二层承上启下，向下封装底层驱动中的基本函数，向上提供简洁易用的上位机编程接口；第三层为交互层，是用户进行交互的上位机测试软件。整体的软件架构如图 5 所示[6]。

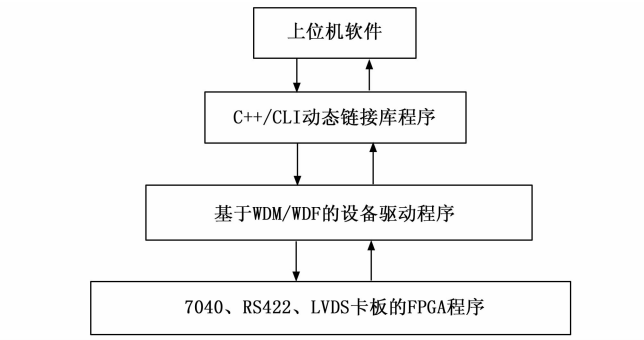

图 5　软件的基本构架

2.1　实时操作系统选型

实时操作系统强调任务完成时间的确定性，确保任务在严格的时间约束内完成，而不仅是逻辑正确。实时操作系统的任务调度受当前负载影响，高负载会导致延迟，例如“打开过多软件造成卡顿”，这在实时操作系统中是不允许的[7]。

常用的实时操作系统有：RTLinux、Ardence RTX、Windows CE，VxWorks 等，对比后发现 VxWorks 操作系统具有很高的硬实时性，中断响应与进程切换高效，但成本高昂，缺乏上位机对接系统，开发难度大且周期长。RTLinux 操作系统虽然开源，但图形界面不及 Windows，更适用于嵌入式开发。Windows CE 操作系统是高度模块化与可定制的，但也同样面临 RTLinux 操作系统存在的问题。相比之下，基于 Windows 扩展的 RTX 实时操作系统，可以提供类似 Win32 的 RTX 库，融合 Windows 的开发优势与实时性需求。

飞行陀螺组件的性能测试需要满足实时性、通用性、稳定性及人机交互友好性等需求。因此通过系统软件分层设计，分为实时层和非实时层，结合 RTSS 与 Win32 进程，利用高性能实时通信，达到了毫秒级内精准比对三轴转台与陀螺状态的需求，以指导后续测试动作。该系统不仅周期短，成本低，还保留了 Windows 测控设备优势[8]。

2.2　Windows 下 RS422 基于 SC28L202 的实时扩展

在 Windows 环境下，基于 WDM/WDF 的驱动框架在后台任务较多时，并不能保证整个操作系统的实时性，处理器性能的不断提升也不足以从根本上改变系统的非确定性。由于软件计时策略受限于可靠性，可对 RS422 板卡进行优化，通过其高精度硬件计时增强实时性，精确执行操作，从而提升了系统计时与执行效率。

在 RS422 板卡的实时性拓展中，核心组件 SC28L202 芯片利用其 14.745 6 MHz 晶振与可调预分频器，可以适配多种不同波特率的系统时钟，具有广泛兼容性；通过 SC28L202 中断定时来实现的定时发送功能主要步骤如下：上位机经应用与内核层交互，通过 PCI 总线向 RS422 卡发送指令，实现数据写入 RAM 及定时发送控制。数据通过直接写入或缓存区拷贝的方式传输至板卡 RAM。计时指令及参数设置后，计时模块启动，按周期中断，主模块从 RAM 取数据，经 422 解码后通过总线发送。接收到停止指令后，计时与发送同步停止。

2.3　驱动程序开发

针对实时控制需求提出两种解决方案：一是对现有操作系统底层硬件芯片进行拓展，增强特定操作的硬实时能力；二是采用 RTX 实时操作系统，利用其软实时特性与确定的最坏响应时间接口开发，达到实时性目的[9]。

2.3.1　驱动程序关键技术

2.3.1.1　应用层和内核层之间通信

1）打开设备。打开设备的方式有两种，一是通过 GUID 接口，二是通过符号链接名方式；本研究采用第一种。首先，利用 GetDevicePath 函数精准定位设备路径，随后，通过 CreateFile 函数实现设备的开启，并获取其操作句柄。此过程中，CreateFile 的第六参数灵活设定，以适配不同的操作模式：FILE _ ATTRIBUTE _ NORMAL 确保同步操作，结合 FILE _ FLAG _ OVERLAPPED 后则支持异步处理[10]。

2）关闭设备。在结束测试任务并退出应用程序之前，为确保资源正确释放，需调用 CloseHandle 函数以关闭设备并销毁其句柄。此过程会触发 EvtFileCleanup 与 EvtFileClose 例程的自动执行，确保资源清理的完整性与系统性。

3）DeviceIoControl 通信函数。打开设备获取设备句柄后，通过 DeviceIoControl 函数可实现应用层到内核层之间的相互通信。其控制命令通过整合设备类型、

功能、数据传递方法及访问权限，形成单一控制码，设计于头文件，遵循 CTL _ CODE 宏定义。数据传递方式分四类：METHOD _ BUFFERED 通过分配系统缓冲平衡输入输出，便于管理；METHOD _ IN _ DIRECT 与 METHOD _ OUT _ DIRECT 可实现直接读写，在 WDF 框架下，编程的效率和可移植性大大提升；METHOD _ NEITHER 仅提供可能失效的虚拟地址，且地址随着进程的切换可能随时会失效，编程复杂度高[11]，故应谨慎使用。

4）ReadFile 和 WriteFile 函数。应用程序可以通过 ReadFile 和 WriteFile 函数来与驱动程序通信，同样分同步和异步操作，进行数据传输时也有不同的缓冲区地址获取方式。

2.3.1.2 硬件访问

在硬件交互的架构中，硬件访问被明确区分为 I/O 访问与寄存器访问两大范畴。针对 I/O 映射地址芯片的访问需求，KMDF 框架在任意 IRQL 级别下，提供了诸如 WRITE _ PORT _ UCHAR 及其缓冲区版本等函数，用以向特定端口写入多样化数据结构。同时，READ _ PORT _ UCHAR 及其缓冲版本函数支持从相应端口安全读取数据。针对 PCI 设备开发，其配置空间寄存器 0～5 的关键设置，直接决定了地址空间类型与容量大小。

2.3.1.3 中断处理

中断处理因其 DIRQL 执行级别限制调用函数且需快速响应，常采用延迟过程调用（DISPATCH _ LEVEL 优先级）处理数据。中断服务例程异步调用，同步访问共享数据可以通过硬件中断自旋锁，将级别提升至 DIRQL 或利用中断同步例程实现，但需保证执行时间尽可能短。Windows 内核设 32 级中断优先级，高优先级中断屏蔽低优先级并暂存，恢复时按序响应，两个相同优先级中断的响应顺序和其发生的先后顺序相同[6]。

2.3.1.4 DMA 传输

在 DMA 操作领域，WDF 框架提供了 WDFDMAENABLER、WDFDMA TRANSACTION 和 WDFCOMMONBUFFER 对象，来实现 DMA 操作。不同对象说明不同 DMA 通道的特性。

2.3.2 Windows 系统下 RS422 驱动开发

RS422 驱动程序可以分接收和发送配置模块、定时\非定时发送模块、中断接收模块三部分。

1）接收和发送配置模块主要向底层相应寄存器值精准写入必要值，在驱动过程中通过 OCTUPLE422 _ WRITE _ Handler 函数来实现。

函数调用时，采用一维数组将数据从应用层传输至内核层，首元素为寄存器偏移，其余为待写数据。通过预检数据元素数量确保至少两个 ULONG，不足则报错并终止执行。利用 do {} while (0) 实现错误处理时的即时退出，模拟 goto 功能而规避风险。

2）非定时发送模块采用 FIFO 策略，涉及 OCTUPLE422 _ READ _ ONE _ Handler 函数调用，确保从指定寄存器安全读取数据，利用系统缓冲区配置，严格验证数据大小与缓冲区有效性，通过 WdfRequestRetrieveInputBuffer 与 WdfRequest RetrieveOutputBuffer 处理数据传递，确保数据准确性。动态对比 FIFO 空间与待发送数据，优化发送流程。普通发送则复用配置模块，简化操作流程，仅需指定寄存器与数据即可实现高效传输。整体设计提升了代码复用性、封装性及开发效率，为 RS422 通信提供了稳定可靠的驱动支持。

3）定时发送模块中，针对 Windows 实时性不足，本研究提出 RS422 实时扩展方案，软件架构中定义 IOCTL _ TIMED _ SEND 采用 METHOD _ IN _ DIRECT 方式，DeviceIoControl 函数调用中的 lpOutBuffer 作为输入缓冲，驱动内由 IOCTL _ TIMED _ SEND _ Handler 处理，满足项目需求[12]。

驱动层验证输入后，使用 KSpinLock 确保数据同步访问。检查通道有效性与数据整除性后，依据 FIFO 空余量决定单次发送量。若 FIFO 充足，则直接写入；不足时，仅发送可用量。定时发送基于中断机制，设定 SC28L202 中断级别，启动发送后等待完成事件。中断服务例程中，确认中断源后，延迟处理发送中断，禁止中断并操作数据，随后重新使能。在延迟处理中，根据中断标志位（FIFO 非空或 SC28L202 定时中断），更新发送状态，调整 FIFO 写入量，并重新使能 FPGA 中断。若发送完成，则重置 SC28L202 为全空中断并置位等待事件，使线程继续执行，完成整个定时发送流程[13]。

4）中断接收模块分三部分：一是开启接收配置 FIFO 并使能中断；二是中断服务中检测通道状态，标记并屏蔽中断，延迟调用读取数据至 FIFO；三是接收部分，根据测试需求与单元，从 FIFO 中按帧格式读取数据。

2.3.3 Windows＋RTX 系统下 RS422 驱动开发

本研究将 Win32 与 RTSS 框架相结合，针对实时性需求，将关键任务放在 RTSS 进程内执行，以优化系统响应。为解决跨进程通信问题，采用共享内存机制，通过映射非分页物理内存至各进程的虚拟空间，实现数据无缝共享。同时，采用集成事件、互斥锁及信号量等同步性，确保进程间通信的准确与一致。下面将详细地介绍整个驱动程序。

在驱动程序初始化阶段，首先利用 RtCreateMutex 函数获取互斥体，用于进程间的同步，函数中第三个参数的长度受限于 RTX _ MAX _ PATH，是一个以 NULL 结尾的字符串，该参数名称需唯一。通过 GetLastError 返回 ERROR _ INVALID _ HANDLE。无名

互斥体仅适用于单进程内部同步。在成功创建或引用现有互斥体后，通过句柄进行访问，若尝试创建的互斥体已存在于系统中，则返回 ERROR _ ALREADY _ EXISTS 错误码，表明操作互斥体已存在而未执行预期的新建操作。对于非此类特定冲突情况将返回 0，表示先前调用成功且未发生错误。随后，通过调用 RtCreateEvent 函数，创建两个事件对象，作为同步核心机制的核心组成部分，用于 RTSS 与 Win32 进程间的通信。这个调用的第二个参数设定了自动或手动重置模式，其中手动模式在创建时需显式调用 RtResetEvent 进行管理，以确保通信的精确控制。可以通过进程映射的物理内存的命名区域创建可以通过 RtCreateSharedMemory 实现。尽管在 RTSS 环境中，第一个参数 PAGE _ READWRITE 或 PAGE _ READONLY 的选择相对次要，但内存大小及命名参数则需要严格定义，其中第二个和第三个参数指定了共享内存的大小。第四参数与创建互斥锁、事件时字符串参数的要求具有相似，需有效且格式正确。成功调用后，最后一个二级指针获共享内存虚拟地址，用途同互斥锁、事件返回值，用于后续资源引用[14]。

驱动程序的核心线程持续监测 Win32 环境下的时间状态，是一个循环等待的过程。在 Win32 通知 RTSS 进程的事件被激活，即触发对内存空间的访问，通过读取共享内存中相应的 DevCtrlArray［DevNum］.pDevCtrl Share－＞CmdId 来判断并响应四种不同的控制指令：RtxCmd _ Open、RtxCmd _ Close、RtxCmd _ StartRecv、RtxCmd _ StopRecv；

RtxCmd _ Open 初始化板卡硬件，封装于 InitializeOneDevRtss 函数中。调用 PciFind Device 获取 PCI 设备信息，调用 PciGetMem BaseAddress 获 9 054 寄存器物理地址存于共享内存。Local 寄存器操作类似。随后通过 RtMapMemory 映射物理内存至用户虚拟空间，作为操作基址。调用 RtAttach InterruptVectorEx 将 IST/ISR 与中断关联，接着创建通道接收/发送事件及锁。最终，启用 PCI 与 LOCAL 中断，完成硬件初始化流程[15]。其流程图如图 6 所示。

```
ULONG intcsr =
* PVULONG(DevCtrlArray[DevNum].VirBas eAddr9054 +PLX9054_INTCSR);
if(!(intcsr &PLX9054_INTCSR_LIIA))
{return FALSE;}
```

IST 为 RTX 中断服务程序，通过比对中断状态与预设值，确认中断源是否为本设备产生，不是则退出服务程序。

```
(UCHAR) * PVUSHORT(DevCtrlArray[DevNum].VirBaseAddrLocal + IsrAddr[Ch]);
```

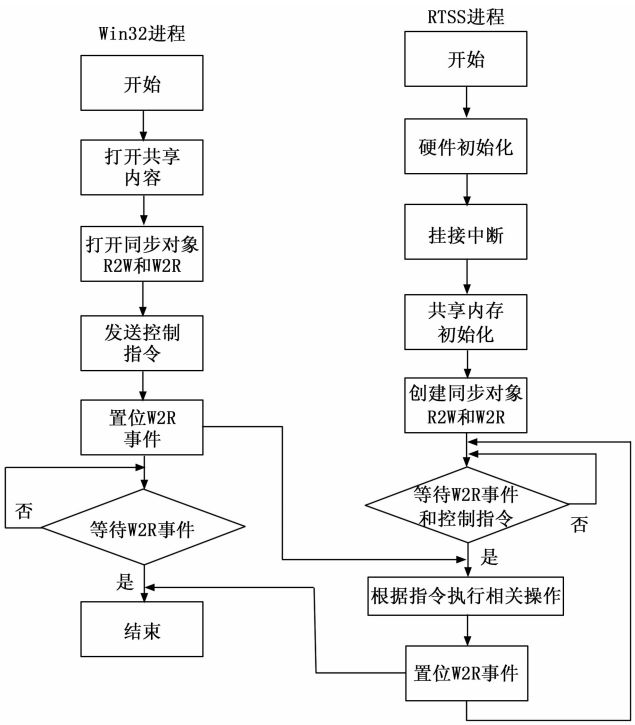

图 6　Win32 和 RTSS 驱动流程图

```
if(!(Isr&(1<<1)))continue;
```

后续步骤中，针对不同板卡号所映射的特定通道号，分别接收不同的中断事件实施精细化处理策略。检测到中断后，通过即时锁定共享数据区域，确保数据一致性。随后，验证中断通道状态，若符合预设条件，则精确计量底层 FIFO 数据量，并高效迁移至 RTX 驱动层 FIFO，否则跳出本次循环[16]。

3　测试例程设计

针对飞行陀螺性能测试需求，开发的上位机软件共包含 9 个独立且协同的模块，测试系统软件框架如图 7 所示。

在飞行陀螺测试系统中，人机接口是用户与设备交互的关键，还可作为二次开发的接口，对设备可维护性和可升级性至关重要[17]。基于面向对象编程，硬件被抽象为对象，操作封装于对象中，相似操作统一接口处理，可以显著提升开发效率。

人机接口功能上细分为自动与手动测试接口。手动模式下，测试前需进行预配置测试任务，并根据实际测试时的测试目标、测试场景及测试需求下发测试任务。自动测试则分为全项与单项，前者全面执行选定任务，若参数不合格或出现测试指标异常，则对相应测试任务进行重测并记录全部数据以便后续分析调整；在实际测试中针对特殊数据测试，用户可调整自动测试配置或结合手动模式。系统具有数据库记录操作与错误数据，通过数据分析，软件能自动判别问题并提供调整建议，优

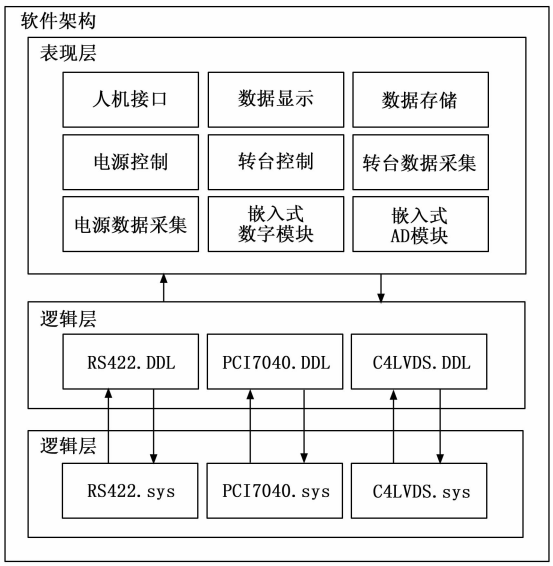

图 7 测控系统软件框图

化测试流程。

数据显示模块中，在使用原生控件进行数据显示过程中，数据显示频率较高时容易导致视觉上的闪烁。为此采用 ListView、TextBox 及 MeasurementStudio 控件，引入 DoubleBuffered List View 并进行二次开发，使用双缓存技术来显示数据。该方案的数据显示分为两个部分：一是测试数据显示，使用不同颜色对正常、偏大、偏小及超调数据进行编码，供辅助测试人员即时分析调整；二是操作显示，用于实时反映测试系统运行状态，方便测试人员进行故障排查和结果评估[18]。

数据存储领域中的高效性与准确性是研究关注的焦点，在存储方式上，数据存储分为本地数据与本地数据库存储两类。在实际项目中，面对大量测试数据的文件读写需求，现有文件操作类虽提供基本支持，但接口局限且灵活性不足，特别是在处理多样化、大规模数据时，直接混合存储管理复杂，容易导致数据混乱与误读，且文件损坏时难以预知。鉴于此，需要设计并创建一种面向特定测试需求的线程安全文件操作类 SingleEncoderAndDecoder，该类涵盖特定的测试数据信息，支持按需读取特定类别或时段数据，优化了数据存取流程。

在实际测试中，多线程同步不当易致文件存储数据错误。为降低模块耦合、提升开发效率，本系统将文件读写的多线程同步操作封装在不同的类中，调用者不需要考虑同步与互斥问题[19]。针对数据库存储，因为测试项不同会导致数据量和类型有所差别，因此开发了适配的数据库操作类。根据数据传输和存储需求，灵活采用按需连接或持续连接、多级缓冲写入等策略，确保数据存储高效且准确，避免丢帧等问题。

转台控制模块因产品差异而采用不同控制方式。普通测试模式下，测控系统通过上位机 RS232 与转台通信，将速度、角加速度、位置等信息发送至转台上位机，带动转台运动；另一种为动态仿真模式，可任意配置转台运动曲线，测试软件解算曲线为控制命令，经测控系统下位机 RS422 接口发至转台下位机，实现转台控制。通过 RTX 开发及 PCI-422 板卡二次开发，可以达到最高精度 1 ms 的定时发送[20]。转台数据采集模块同样分为普通和仿真双重模式，普通模式基于 RS232 协议实现实时数据交互并即时显示到系统界面，仿真模式下采集的数据若入库则采用多级缓冲机制[21]存储。

此外，测试系统中的电源控制模块发生故障时，不断电仅报警，并由工作人员决定是否断电，通过软件配置电平信号控制 OC 门上电，可兼容多种指令及不同系统；电源采集模块分指令与产品电源采集，前者用 PCI-7040 的 AD 采集，后者经远端采集盒的 422 接口获取。嵌入式数字模块与嵌入式 AD/IO 模块通过特定命令帧格式通信；系统还具备自动化测试功能，在自动化测试任务的设计中，可以通过构建自动测试接口来实现，这些接口会通过调用原本用于手动测试的函数来执行具体的测试步骤[22]，测试过程中应该减少人为干预。

4 测试结果与分析

测试结果表明，惯性测量单元标定综合测试设备在功能性和性能稳定性方面均达到设计要求。在功能测试中，测试软件通过主界面实现了参数配置、测试项目选择和实时数据显示等操作。测控系统主界面如图 8 所示。测试过程中，设备能够准确响应用户指令，完成供电控制、模式切换和数据采集等核心功能。

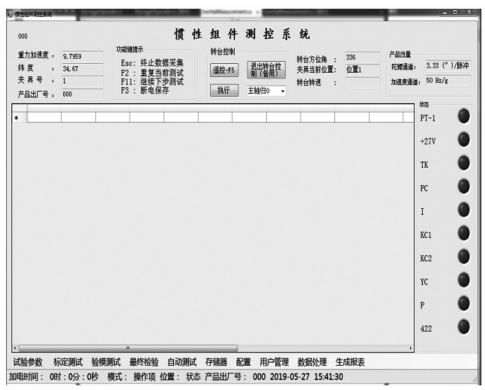

图 8 测控系统主界面

性能测试显示，系统在实时性和精度方面达到了预期目标。下图展示的分别是速率测试、位置测试，分别见图 9、图 10。

实时性测试结果表明，系统在高负载条件下依然能够保证指令响应迅速，满足惯性测量单元性能测试的高实时性需求。精度测试结果显示，设备在模拟与数字量

图 9　速率测试界面图

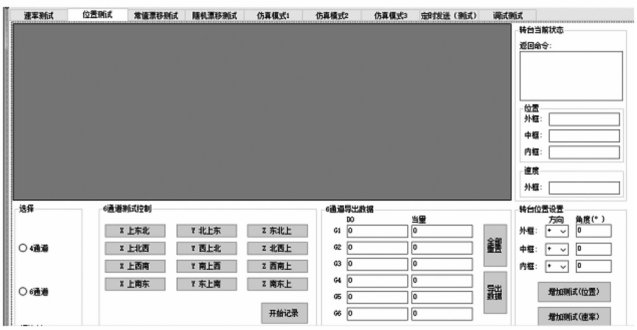

图 10　位置测试界面图

的测量误差范围均满足要求，且在宽温范围内表现稳定。整体来看，测试系统具备良好的稳定性、可靠性和扩展性，为后续型号产品的测试需求提供了坚实保障。

5　结束语

本研究聚焦于某型号飞行陀螺性能测试系统，基于 RTX 平台攻克测试程序关键技术难点，满足了系统的高实时性需求，实现了稳定可靠的测试软件开发。主要研究内容包括：协同硬件选型与设计、模块化开发 RTX 驱动、构建上层界面，并全面测试系统性能。在未来，RTX 测试软件开发将是军用测试领域的重要趋势。测试系统的研制不仅实现了飞行陀螺性能测试的突破，更为后续同类设备的软件开发提供了技术支撑。未来工作可以从提升软件稳定性、解决潜在技术问题，并拓宽 RTX 测试软件在多元测试场景中的应用等方面着手，推动技术得到更广泛的应用。

参考文献：

[1] 张　玮. 航天装备测试技术发展与应用 [J]. 科技与创新，2023 (16)：170 - 172.

[2] ZHOU C D，TANG B P. Development of PCI bus based data acquisition card [A] // Proceedings of the Second International Symposium on Instrumentation Science and Technology [C]，2002：174 - 177.

[3] 杨阿锋，吴　帅，刘　凯，等. PCIe 接口高速数据传输卡的驱动程序开发 [J]. 中国测试技术，2008 (2)：67 - 68.

[4] 张文辉，陈　欣，曹立佳，等. 基于 RTX 的无人机地面站系统 [J]. 兵器装备工程学报，2017，38 (9)：91 - 94.

[5] 唐　寅. 实时操作系统应用软件开发指南 [M]. 北京：中国电力出版社，2002.

[6] WU Q，XU J M，LI X W，et al. The research and implementation of interfacing based on PCI express [A] // International Conference on Electronic Measurement & Instruments [C]. 2009 (3)：116 - 121.

[7] WILEN ADAM，JUSTIN P，THORNBURG RON. Introduction to PCI express：a hardware and software developers guide [M]. US：INTEL PRESS，2003：83 - 100.

[8] 于继超，刘经宇. 基于 RTX 的工程飞行模拟器数据采集与存储系统设计 [J]. 系统仿真技术，2014，10 (1)：73 - 80.

[9] DHAWAN S K. Introduction to PCI Express-A new high speed serial data bus [J]. IEEE Symposium on Nuclear Science，2005：687 - 691.

[10] 张沛露，张宇鹏. 基于 RTX 系统 PCI 总线驱动程序设计实例 [J]. 经贸实践，2015 (8)：344 - 345.

[11] 党　赟. 基于 WDF 框架的航天测控数字基带板驱动程序设计 [D]. 西安：西安电子科技大学，2015.

[12] 黄　键，宋　晓，薛顺虎. RTX 平台下实时仿真系统的设计方法 [J]. 计算机应用与软件，2009 (4)：167 - 169.

[13] RENAUX D，DASIEWICZ P. RTX-Parlog：Real-Time extended parlog [J]. Real-Time Systems，1993：147 - 153.

[14] WANG S A，LI J，JIANG S M，et al. Study of Real-time data collection and release with in-vehicle system [J]. Computer Science and Electronic Technology International Society，2012：1472 - 1475.

[15] RTX7.1 documentation [S]. US：Ardence Inc，2012.

[16] 李宏科. 一种基于 RTX 的实时系统的实现 [J]. 装备制造技术，2006 (3)：55 - 56.

[17] DAVIS CLIVE. Digital oscilloscopes take strain out ofbus analysis [J]. Electronics Weekly，2006，22 (39)：26 - 27.

[18] 郑人杰. 计算机软件测试技术 [M]. 北京：清华大学出版社，2000.

[19] National instruments NI-DAQmx help [S]. US：National Instruments，2008.

[20] Tektronix，MSO2000B，DPO2000B，MSO2000 and DPO-2000 series oscilloscopes programmer manual [S]. US：Tektronix，Inc，2012.

[21] 王立红，刘建立，周丽芹. 多通道 PXI 总线数据采集系统设计 [J]. 电子质量，2004 (11)：23 - 25.

[22] National Instruments NI-DAQmx C Reference Help [S]. US：National Instruments，2008.