

doi: 10.11823/j.issn.1674-5795.2022.06.14

鸿蒙操作系统在智能测量仪器中的应用

王晶星, 李新良*, 李维*

(航空工业北京长城计量测试技术研究所, 北京 100095)

摘要: 嵌入式操作系统在仪器智能化过程中起着至关重要的作用。通过分析鸿蒙系统的技术架构和分布式及微内核特性, 并与现有的国内外嵌入式操作系统进行对比, 结合测量仪器数据采集的实际应用案例, 得出了鸿蒙系统在自主可控、实时性和连接流畅方面具有优越性, 满足测量仪器对操作系统的需求。

关键词: 计算机控制; 嵌入式操作系统; 智能仪器; 鸿蒙系统; 微内核

中图分类号: TB9; TP316 **文献标识码:** A **文章编号:** 1674-5795(2022)06-0096-12

Application of Harmony system in intelligent measuring instruments

WANG Jingxing, LI Xinliang*, LI Wei*

(Changcheng Institute of Metrology & Measurement, Beijing 100095, China)

Abstract: Embedded operating system plays a crucial role in the process of instrument intellectualization. By analyzing the technical architecture and distributed and microkernel characteristics of Harmony OS, comparing with the existing domestic and foreign embedded operating systems, and combining with the actual application cases of data acquisition of measuring instruments, it is concluded that Harmony OS has superiority in independence and controllability, real-time and smooth connection, and meets the needs of measuring instruments for operating system.

Key words: computer control; embedded operating system; intelligent instrumentation; Harmony system; microkernel

0 引言

测量仪器^[1]是科学研究和科学发展的基础条件, 现代科技的重大突破依赖于领先的测量技术^[2]。美国商务部研究报告显示: 其测量仪器与设备的生产总值约为美国制造业总量的4%, 但对美国国民生产总值的带动效应却超过66%。也就是说, 测量技术具有价值撬动、有效赋能以及创新带动经济的重要功能。当前, 测量仪器逐渐朝着数字化和网络化的总体方向发展, 而测量仪器产品性能的发展目标是高精度度、高可靠性、高环境适应性。随着计算机技术的发展, 智能化是

测量仪器发展的重要方向。智能测量仪器利用计算机系统实现传统测量仪器的功能, 包括自动采集、实时修正测量误差、综合处理测量数据、控制管理输入输出设备等功能, 促进了仪表的准确性和便捷度, 测量仪器的智能化离不开嵌入式操作系统(Embedded Operating System, EOS)的支撑。

随着国际科研环境的变化, 发展具有自主知识产权的嵌入式操作系统已经成为了共识。目前, 国内的嵌入式操作系统研究和开发的主要目标是为航空航天、城市轨道交通、信息通信装备等产业形成自主可控的操作系统。华为的鸿蒙操作系统(Harmony OS)在国产化道路上迅速发展, 其分

收稿日期: 2022-10-31; 修回日期: 2022-11-13

基金项目: 国家“十四五”技术基础项目(KBC52102580); 航空计量所创新基金项目(ZC02201586)

引用格式: 王晶星, 李新良, 李维. 鸿蒙操作系统在智能测量仪器中的应用[J]. 计测技术, 2022, 42(6): 96-107.

Citation: WANG J X, LI X L, LI W. Application of Harmony system in intelligent measuring instruments[J]. Metrology and measurement technology, 2022, 42(6): 96-107.



布式和微内核特性,让测量系统中各模块的连接更流畅,数据传输也有了统一的语言。通过拓展总线接口,有利于实现测量设备之间的智能互联。将鸿蒙系统植入测量仪器中,可以扩大测量仪器的智能化水平,使测量仪器的功能进一步扩展提高。同时,鸿蒙系统作为国产操作系统,提供了自主可控、国产化的软件选择,提升了现有仪器的功能与自主化水平,为计量仪器搭载国产平台完成了初步的探索。

通过分析智能测量仪器的技术背景和需求,基于嵌入式操作系统的特点和分类,研究鸿蒙系统的基本特性,对鸿蒙系统在智能测量仪器中的应用进行归纳总结,验证鸿蒙系统的可用性,最后分析和展望鸿蒙系统在测量领域的发展前景。

1 嵌入式操作系统的概念和特点

1.1 嵌入式操作系统的概念

操作系统(Operating System, OS)是计算机系统中基础的系统程序,它对整个计算机控制系统的硬件和软件信息进行控制 and 处理,以合理配置整个计算机控制系统的能力和资源,促进计算机与其他硬件和软件的连接和通信。如图1所示,操作系统处于硬件之上,直接面向硬件,统一管理硬件资源;处于用户和应用程序之下,帮助应用程序访问各种物理资源并提供抽象的接口。



图1 计算机体系的层次结构

Fig.1 Hierarchy of computer system

嵌入式系统运行在计算机平台中,用于控制和管理其他设备,其硬件和软件可进行裁剪和配置,软件代码短小而精悍,可以固化,并具有低功耗、高可靠性、高稳定性的特征。嵌入式系统结构通常由四部分组成,包括嵌入式微处理器、外设硬体装置、嵌入式作业系统和应用程序。嵌入式系统在电信系统、电子产品、仪表、航空电子、医疗设备、智能家居等领域都有广泛的应用。

嵌入式操作系统^[3]在嵌入式芯片环境中运行,并且对整个芯片系统及其可操作、管理的所有单元设备、部件装置等资源实现统一的协调、调度、指挥和控制^[4]。在实现智能测量仪器的过程中,嵌入式操作系统的作用有:为测量仪器的硬件模块提供底层驱动,以传递控制数据信号;协调测量系统应用层各个模块的功能调度,保证整个系统的实时性。嵌入式操作系统通常分为分时操作系统和实时操作系统。

1.2 分时操作系统

在分时操作系统中,多个用户共同使用一台主机,在一定程度上节省了资源。系统服务请求采用时间片轮转的方式处理,多个终端通过通信线路连接,多个用户交替地占用主机服务器上的时间片来处理作业。

分时操作系统的典型代表是Linux操作系统,也是目前较为普及的开源操作系统。Linux操作系统的系统内核性能高、稳定、安全、网络特性好,具有良好的可移植性、可靠的系统安全以及丰富的驱动以支持广泛的周边硬件设备。

2015年,董思乔^[5]等人基于Linux和Wi-Fi构建了智能家居控制系统,系统采用PC主机和智能手机作为基本硬件平台,辅助Wi-Fi智能插座、Wi-Fi智能传感器和Wi-Fi设备控制电路,实现了运行稳定且响应快速的智能家居控制系统。2017年,孙忠祥^[6]基于Linux系统完成了联网与数据转发、传感器数据采集以及远程监控的功能,实现了智能农业温室大棚远,程监控系统的设备云平台。2021年,郭堂伟^[7]等基于Linux和ARM芯片设计了医院空气质量实时检测系统,采用Lora无线技术作为通信支撑,设计了主控、采集和通信三大模块,空气质量检测效率显著提升。

分时操作系统能够向用户提供良好的界面和及时的响应,每次应用中得到系统资源的机会比较平等,而且出现重复垃圾任务的机会较小,相对来说也能够缩短CPU的闲置时间。不过分时操作系统还面临着可靠性问题和数据通信的困难,因此需要特别注意用户程序 and 用户数据的安全与完整性,而且分时操作系统不能处理一些紧急任务,它循环地为每个用户/作业服务一个时间片,对每个用户/作业完全公平,所以实时处理用户任

务的能力比较欠缺。

1.3 实时操作系统

实时操作系统,是指能够在有限的响应时间内,将所需紧急性等级的服务提供给应用程序的操作系统。“实时”是指运行操作系统的计算机对于输入信息可以做出及时的处理,并在被处理对象指导的时间范围内迅速做出反应。因此,实时操作系统在内核设计、调度策略选择、任务优先级的分配等方面与其他操作系统均有显著差异,典型例子是 VxWorks, RT-Thread、鸿蒙系统等。

VxWorks 是一个具有高性能的内核以及友好的用户开发环境的嵌入式实时操作系统。VxWorks 的特性是:可裁剪的微内核结构、允许按需选择的操作系统组件、高效的任务管理、灵活的任务间通讯和微秒级中断处理。VxWorks 系统由于公用程序精练且效率很高,使延迟时间很短,所以系统自身开销很小,实现了 VxWorks 优越的实时性。

RT-Thread 是一款国产的物联网操作系统,开发简易、超低功耗、安全性高,具备嵌入式操作系统所需的所有关键组件(包括 GUI、网络协议栈、安全传输、低功耗组件等),因此具有大量的可支持的芯片、合作厂商、踊跃的社区开发者和广泛的应用领域。表 1 对比了 VxWorks 和 RT-Thread 实时性能方面的主要指标。

表 1 VxWorks 和 RT-Thread 实时性能主要指标

Tab.1 VxWorks/RT-Thread real-time performance

操作系统	中断响应时间/ μs	任务切换时间/ μs
VxWorks	< 3	3.8
RT-Thread	< 10	< 20

杨文豪^[8]等人设计了基于 VxWorks 的 PXI 实时读写接口,对设备驱动进行了开发、封装和整合,解决了高速数据采集控制卡在分时操作系统下存在的实时性差、不稳定、控制复杂等问题。刘恩凡^[9]等人设计了基于 VxWorks 的某装备自动测试系统,软件采用虚拟仪器技术,系统采用标准化信号接口,满足了装备测试的实时性需求。张明利^[10]等人设计了基于 VxWorks 的嵌入式多协议通信控制器,实现了效率高并且可靠性好的管理和控制,适用于通讯控制器系统接口多、功能复杂、信息处理能力要求高的场景。

2014 年,李明^[11]在 RT-Thread 上设计了全自动生化分析仪测控系统,系统使用 CAN 总线,RT-Thread 提供的实时多任务处理功能满足了仪器对复杂控制的要求。2018 年,盛蔚^[12]等人设计了基于 RT-Thread 的无人机数据记录仪,对传统无人机数据记录仪存在的功耗大、存储效率低、实用性差的问题做出改进,实现了多通道、大数据流输入环境下的长时间准确存储。2019 年,莫小凡^[13]等人利用 RT-Thread 和生物传感器设计了葡萄糖检测仪,满足了分析仪线性检测范围宽、检测速度快、检测精度高的检测需求。

实时操作系统的显著特点是具有较高的及时性,根据任务的实时需求,执行优先调度,在规定时间内完成对该事件的处理;同时也具有较高的可靠性。而鸿蒙操作系统不仅提供了自主可控的选择,还采用了微内核的架构,很好地解决了裁剪嵌入式系统时面临的问题,并且采用分布式技术来统一协议和数据接口,解决了测量仪器设备语言不统一的问题。

2 鸿蒙操作系统概述

2.1 鸿蒙操作系统概述

鸿蒙操作系统是华为自主研发的一款基于微内核的实时多任务操作系统。于 2019 年 8 月 9 日正式发布,目前已更新至鸿蒙 3.0 系统。鸿蒙系统的分布式概念是从传统的单一设备的系统能力出发,对多种终端形式的特性进行适配,能够同时对多种终端设备进行支持,解决了很多割裂智能终端体验的难题。从用户的角度来看,鸿蒙系统可以让应用场景中的各类终端实现能力互相融合,从而构成一个能够在不同终端设备之间实现快速连接、能力互助、资源共享的“虚拟超级终端”,匹配合适的设备,提供全场景的流畅体验。从应用开发者的角度来看,鸿蒙系统引用了各种分布式技术的开放方式,使得开发实现应用程序不受终端设备形态差异的影响,大大降低了开发复杂度和开发成本,使开发者将精力专注在上层业务逻辑上,更加轻松高效地开发应用程序。从设备开发者的角度来看,鸿蒙系统采用模块化的设计方式,能够灵活裁剪设备的资源能力和业务特性,适应操作系统对终端设备的各种形态要求。

鸿蒙系统整体遵从分层设计，从下到上依次为：内核层、系统服务层、框架层和应用层。内核层包括内核子系统和驱动子系统。系统服务层包括系统基本能力子系统集、基础软件服务子系统集、增强软件服务子系统集、硬件服务子系统集，可根据实际应用情况对子系统进行裁剪。框架层为开发应用程序提供了多语言框架。应用层由一个或多个FA（Feature Ability）或PA（Particle Ability）组成。鸿蒙系统的技术架构如图2所示。



图2 鸿蒙系统技术架构

Fig.2 Technical architecture of Harmony system

2.2 鸿蒙对传输层策略的优化

传统分布式操作系统的不同终端之间属于点对点连接，带宽较为固定，远端到中心计算机的延迟时间较高。若整体链路中有设备突然离线，会导致与其相连的分布式节点受到影响，从而导致整体网络不稳定。而不同节点的节点能力也不对等，资源分配不均匀的情况下会导致算力、带宽受到限制，从而导致整体效率的下降。为优化这些不足，低功耗传感网一般采用ZigBee或蓝牙进行组网，拓展整体带宽，并保证单一设备支持热插拔。然而这些方式仍然不能解决单一设备算力过低、组网困难的情况，同时也无法保证传输时钟的对等。

鸿蒙系统为解决上述问题，首先从连接协议入手。以蓝牙为例，传统的终端为BLE协议，利用3个固定信道来回扫描进行查找和匹配，效率较低，而鸿蒙则利用对占空比进行控制调速，以特殊频率匹配，加速发现过程。同时，启用带宽更大的Wi-Fi与蓝牙配合连接，进行异构混组网，用户无须设置连接方式，无需复杂的操作即可对不同的设备进行匹配。当需匹配1台设备时，鸿蒙和传统分布式操作系统匹配速度时间差异如表2所

示。当需匹配设备为4台时，鸿蒙和传统操作系统匹配速度时间差异如表3所示。

表2 匹配1台设备时间差异

Tab.2 Matching time difference of one device

操作系统	首次连接时间/s	非首次连接时间/s
鸿蒙	0.5	无时延
传统分布式	> 3	> 3

表3 匹配4台设备时间差异

Tab.3 Matching time difference of four device

操作系统	首次连接时间/s	非首次连接时间/s
鸿蒙	1.5	无时延
传统分布式	> 10	> 10

为保证Wi-Fi与蓝牙同时开启的情况下尽量降低功耗，鸿蒙引入了策略控制，即根据当前设备状态控制设备接入的耗能模块是否工作。同时采用快速唤醒与启动芯片，控制Wi-Fi传输速率，将芯片性能发挥到极致。

如图3所示，绿线代表鸿蒙的Wi-Fi调度策略，蓝线则为传统操作系统的Wi-Fi调度策略。在初期的相应阶段，鸿蒙快速分配带宽与存储资源，使Wi-Fi速率快速达到网络带宽的上限，并一直接近满速率波动。同样，在稳定阶段，网络限制产生变化时，鸿蒙的调度策略相较于传统操作系统来说也不会产生过度地上冲或下冲，并持续保持接近满速率运行。

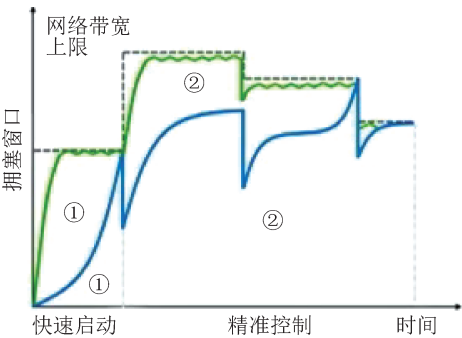


图3 鸿蒙与传统系统Wi-Fi效率对比

Fig.3 Comparison of Wi-Fi efficiency between Harmony system and traditional system

Wi-Fi与蓝牙共同启用的另一个好处是最大程度地避免了自身通信产生的数据阻塞。在Wi-Fi速率跑满的状态下，鸿蒙可将控制帧分配给蓝牙传输，Wi-Fi则负责大型数据流的传输，减少Wi-Fi

处理的管理消息。

与蓝牙相似,为最大程度地避免其他Wi-Fi设备抢占信道,该Wi-Fi采用了2.4 GHz和5 GHz多径双路并发的机制,且不局限于传统2.4 GHz无线网的20 MHz/40 MHz窄带宽,扩展了国内对于5 GHz避让频段的限制,提升了文件的吞吐能力。

尽管鸿蒙Wi-Fi和蓝牙的积极调度策略相较于传统Wi-Fi和蓝牙共同使用的功耗降低了10%,但也可以舍弃Wi-Fi从而进一步降低功耗,单通道传输则势必会遇到运力不足等问题,于是鸿蒙也加入了多级动态缓冲池作为到端运力的调度策略,在高带宽设备未找到其他网内高带宽设备并对低带宽设备发送数据时,将无法迅速接收到的数据先缓存至本地存储,再根据低带宽设备的接收速度逐步将全部数据发送,防止了数据的重传,同时解决了老设备对新传输协议的兼容问题。

鸿蒙操作系统使用近端控制,摆脱了极端条件下对云端网络的要求。同时引入了基于软时钟的软总线架构,不但对信道分时复用,防止连接设备过多导致的信号冲突问题,也可以最大程度地降低组网设备掉线导致的数据丢失、大面积设备掉线等风险。设备之间也可进行信息交换,可

信设备自组网,并感知上下线及完善信息管理。

2.3 基于软时钟的软总线网络架构

总线(Bus)是一种内部结构,是CPU、内存、输入、输出设备传递信息的公用通道,传统的终端通过总线连接主机各部件,再将外部设备与总线通过相应的接口电路连接,从而构成计算机硬件系统的公用通道^[14]。通过这种方式产生了三层线路,分别为地址线、数据线和控制线,且不同设备之间相互独立,控制者无法利用组网设备与被控设备共享地址线而单独调试该设备。单独的设备也由于端对端传输而无法在不影响整体网络的情况下实现热插拔。

针对传统分布式的网络不可靠、延迟受环境影响可能较高、由于设备未形成网络而导致的有限带宽、网络拓扑改变导致的组网失败等,鸿蒙的分布式架构设计了基于任务总线(sendByte)和数据总线(sendMessage)的软总线。分布式软总线的架构如图4所示。进行分布式软总线的改造后,全部入网设备被视为超级终端,通过感知、计算、调度将高速通道(Lane)引入资源,将所有终端并为共享资源的集合,同在异构组网内的pad、手机、pc等则可通过LaneHub获取共享资源。

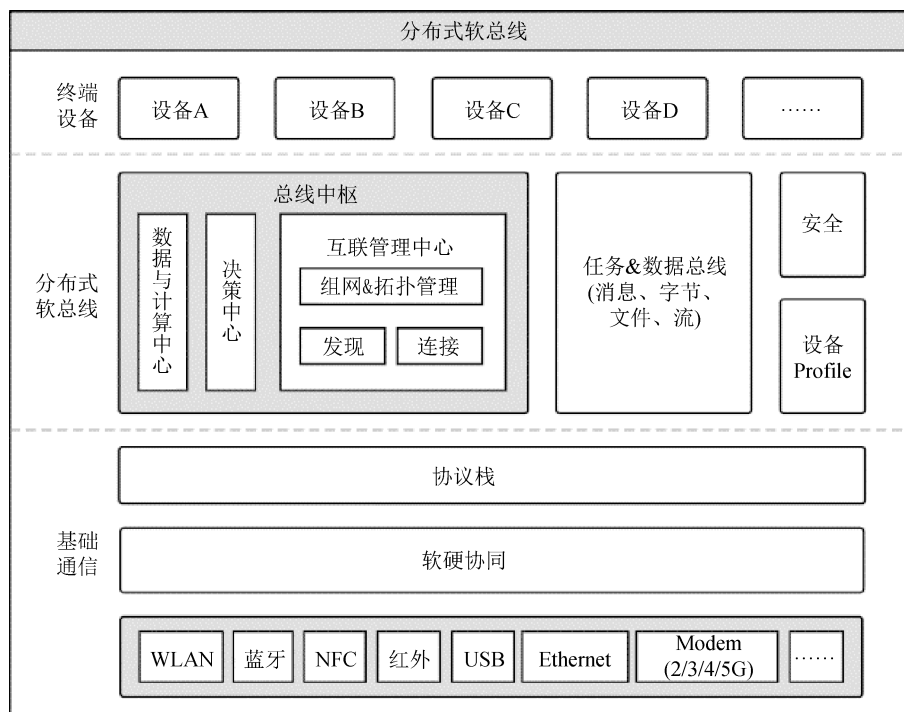


图4 分布式软总线

Fig.4 Distributed soft bus

除此之外,为最大程度地利用分布式的空口资源,避免频段冲突和竞争,鸿蒙系统也舍弃了分布式设备的内部时钟。这些时钟由于内部晶振的不同质量决定时钟偏移不同,很难保证时间的同步性,持续同步又会带来功耗与通信消耗。而

鸿蒙的超级终端则共用一条时钟,以异构网络拓扑与结构为基础,以时间同步的方式为超级终端分布式系统构建统一的时钟源,协调各设备业务时钟,并保持高精度时钟进行分布式业务,即为软时钟。软时钟的同步算法如图5所示。

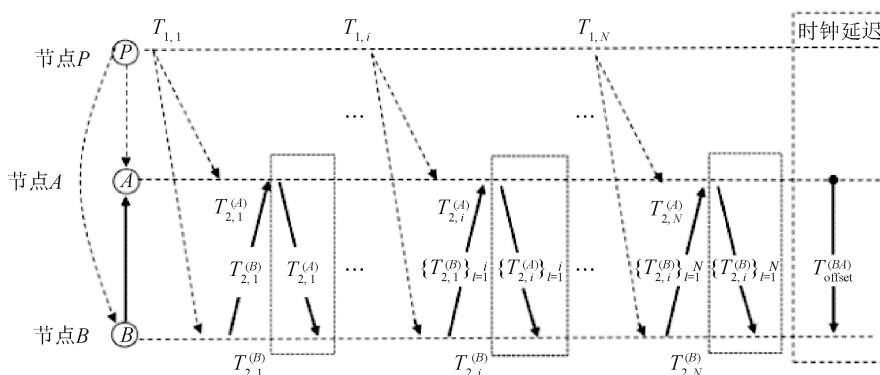


图5 软时钟同步算法

Fig.5 Soft clock synchronization algorithm

以节点 P 作为基准的时钟源,其他节点(例如节点 A 和节点 B)与节点 P 之间,除了本地时间差异外,还存在晶振不同所导致的时间漂移和时钟频率差异等。由此可得出节点 A 和节点 B 上的数据包接收时间分别为

$$T_{2,i}^{(A)} = T_{1,i} + \theta_{\text{offset}}^{(PA)} + \theta_{\text{skew}}^{(PA)} \cdot (T_{1,i} - T_{1,1}) + d^{(PA)} + X_i^{(PA)} \quad (1)$$

$$T_{2,i}^{(B)} = T_{1,i} + \theta_{\text{offset}}^{(PB)} + \theta_{\text{skew}}^{(PB)} \cdot (T_{1,i} - T_{1,1}) + d^{(PB)} + X_i^{(PB)} \quad (2)$$

式中: $T_{1,1}$ 和 $T_{1,i}$ 为节点 P 的数据包发送时间; $\theta_{\text{offset}}^{(PA)}$, $\theta_{\text{offset}}^{(PB)}$, $\theta_{\text{skew}}^{(PA)}$ 和 $\theta_{\text{skew}}^{(PB)}$ 为传输时随机干扰等因素带来的补偿和偏离值; $d^{(PA)}$ 和 $d^{(PB)}$ 为数据包通信时延的确定性部分; $X_i^{(PA)}$ 和 $X_i^{(PB)}$ 为数据包通信时延的不确定部分。

节点之间的数据包接收时间差值为

$$T_{2,i}^{(A)} - T_{2,i}^{(B)} = \theta_{\text{offset}}^{(BA)} + \theta_{\text{skew}}^{(BA)} \cdot (T_{1,i} - T_{1,1}) + d^{(PA)} - d^{(PB)} + X_i^{(PA)} - X_i^{(PB)} \quad (3)$$

传输时随机干扰等因素带来的漂移值为

$$\theta_{\text{offset}}^{(BA)} = \theta_{\text{offset}}^{(PA)} - \theta_{\text{offset}}^{(PB)} \quad (4)$$

$$\theta_{\text{skew}}^{(BA)} = \theta_{\text{skew}}^{(PA)} - \theta_{\text{skew}}^{(PB)} \quad (5)$$

通过调整时间差,可实现多节点之间的时钟同步。引入时钟后,时分复用让极少的频谱资源支持更多设备的连接成为可能。根据每个设备的特征、账号、承载的业务等信息,就可以自动组装成一个超级终端。同时,可以将现有BLE的3个广播信道基于分布式软时钟的不同时间戳来连接

不同设备,使未来几百台智能设备连接在一起成为可能。

多设备自动组网后,监测设备可以通过高速通道监测到设备的发生业务、业务特征、需要的带宽、时延、功耗等,也能从单设备看到设备与全局的关系。这样,当超级终端中的多对设备发生业务时,就可以通过全视角、单设备视角为其选择合适的频点、频宽的信道为其工作,再加上分层控制及QoS管理,就能最大程度地降低干扰。

除了以上给出的分布式软时钟常用的场景,随着业务复杂性越来越多,分布式软时钟还可以在更多场景中发挥更多的作用。比如多设备所带传感器信号的采集,分布式软时钟就可以使得不同传感器上时钟保持一致,以便于保证不同信号的正确性。

分布式软时钟对于物理层协同时延小于300 μs ,传输层时延为500 μs ,窗口协同则为1 ms,在日常使用中基本可以忽略该时延产生的误差。同时,随着不同设备属性不同,所承载业务的网络拓扑不同,其所需要的时钟不同,软时钟将引入复杂网络时钟,为业务提供更加精准的时间同步。

2.4 分布式数据管理

在分布式软总线的能力基础上,分布式数据管理可以实现测量数据的分布式管理。用户数据和测量数据不再绑定单一的物理设备,业务逻辑

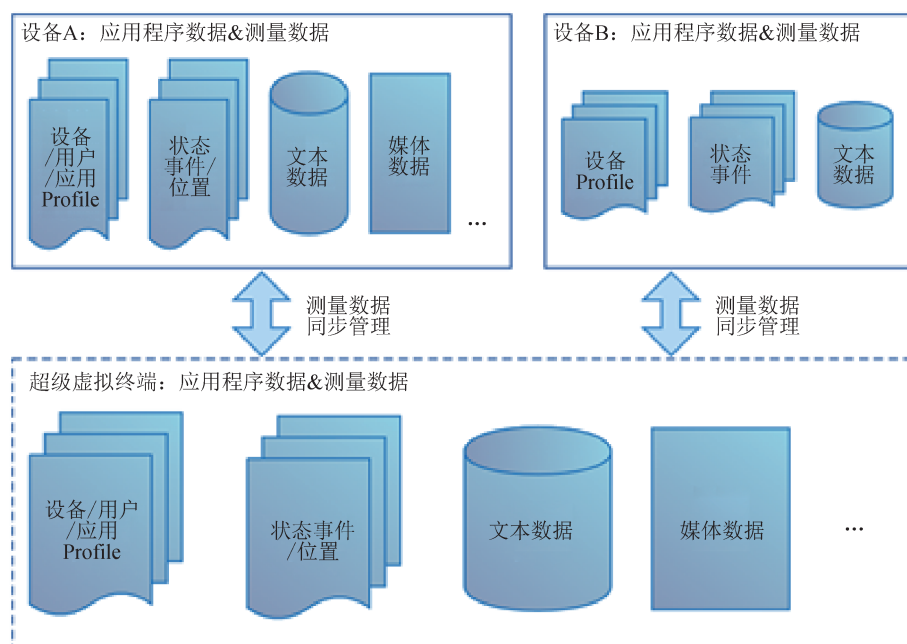


图6 分布式数据管理

Fig.6 Distributed data management

脱离数据存储，数据在跨设备运行的应用中无缝连接，使测量设备操作更加流畅。图6为分布式数据管理示意图。

2.5 分布式任务调度

分布式任务调度构建统一的分布式服务管理（发现、同步、注册、调用）机制，基于分布式软总线、分布式数据管理、分布式Profile等技术特性，支持远程启动、远程调用、远程连接以及跨设备应用的迁移等操作，能够根据不同设备的能力、位置、业务运行状态、资源使用、用户习惯和意图等，实现跨设备应用的一体化，选择合适的设备运行分布式任务。图7以应用迁移为例，对分布式任务调度能力进行了简要展示。

总的来说，分布式的软总线可实现测量硬件能力虚拟化、多终端的能力共享、彼此之间进行外设。分布式数据管理及任务调度使操作使用更流畅，跨设备的数据处理方便快捷，对硬件能力进行跨设备的调度，从而做到数据共享、算力共享、人工智能共享。

2.6 Linux、鸿蒙优劣势分析

在内核方面，鸿蒙采用微内核结构，内核的可移植性比Linux更好，内核稳定性更高，减少了内存的开销，通过IPC（进程间通信机制）实现进

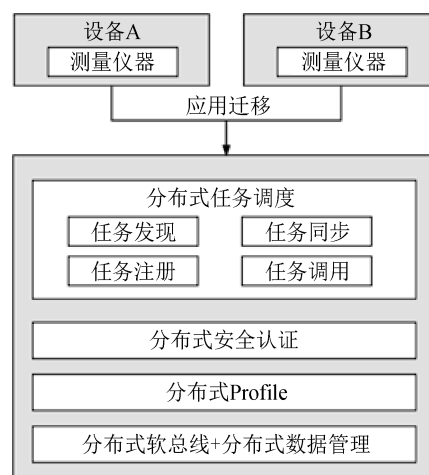


图7 分布式任务调度

Fig.7 Distributed task scheduling

程间通信；而Linux采用宏内核结构，内核的可移植性较鸿蒙差，内核维护量大，系统内核相对较大，因为包含了文件系统、设备驱动，对内存的需求增加，但是通过函数调用直接实现内核模块之间的通信，提高了内核效率。

在实时性方面，鸿蒙采用硬实时中断响应机制，真正做到了在规定的时间内完成事件处理的任务响应要实时；而Linux采用软实时机制，在时间的确定性上比较没有保障。

3 鸿蒙系统在智能测量仪器中的应用

利用鸿蒙系统在分布式软总线和软时钟中展现出的易于组网、定时准确、实时性高的优越性,对传统电子测量仪器进行智能化改造。

整个测量系统由下至上包含了三个部分:①感知层:传感器与专用数据采集设备;②传输层:Hi3861开发板;③应用层:鸿蒙终端。感知层中,测量数据由大量散布在一定区域内的各类传感器节点感知物理量的变化,经过传感器接口电路对信号进行放大后,由ADC芯片完成模拟到数字信号的转换,将测量数据送入专用的数据采集设备。传输层中,由数据采集设备处理后通过RS485串口通信线路传递至海思Hi3861芯片,该芯片将作为数据传输模块,形成一个数据汇聚节点,向下进行串口通讯,向上连接Wi-Fi网络;接着,传感器的测量数据通过Wi-Fi无线通信,借助鸿蒙组网技术,建立近场无线通信,实现UDP协议传输。应用层中,由搭载鸿蒙系统的设备终端实现应用层功能,完成测量数据的可视化显示和用户交互、WEB联网、数据采集的控制指令、提供开放API等。整体上,由鸿蒙终端完成数据的采集、监控、记录、控制,实现整体功能逻辑,系统结构如图8。为将鸿蒙系统移植入智能测量仪器中,进行了以下应用研究。

3.1 改进任务调度算法

针对测量系统的特点,优化并移植鸿蒙系统内核,改进其任务调度^[15]及中断处理算法。

鸿蒙系统采用的分布式任务调度是多级反馈队列(Multi-Level Feed-back Queue, MLFQ),不同于Linux使用的完全公平调度模式。设有N个队列($Q_1, Q_2, \dots, Q_N$),其中各个队列对于处理机的优先级不同,一般来说,优先级 $\text{Priority}(Q_1) > \text{Priority}(Q_2) > \dots > \text{Priority}(Q_N)$,那么位于 Q_1 中的任何进程相对于CPU的优先级,都要高于 Q_2 中的任何进程,依次类推其他队列。进程调度在优先级最低的队列 Q_N 中,按照时间片轮换法进行;对于其他队列,遵循先来后到的服务算法,每个进程分配一定的时间片,如果时间片运行完毕,进程还没有结束,就会进入下一个优先级队列的末尾。并且,每个队列的时间片随着优先级的增加而递减,其时间片在优先级越高的

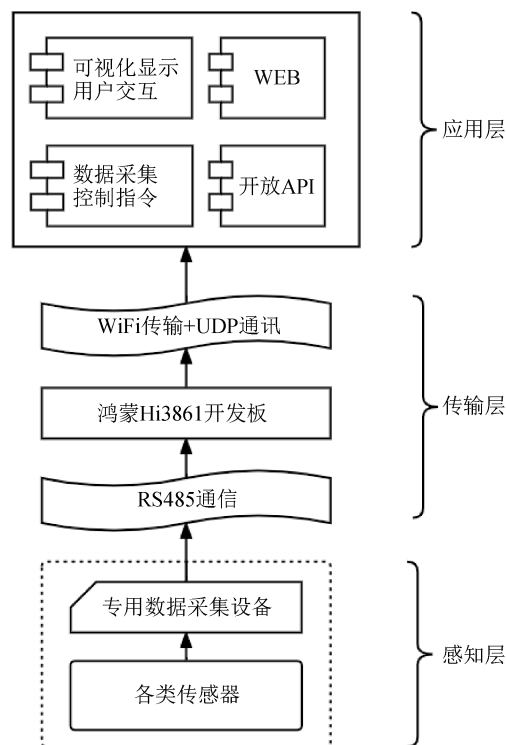


图8 智能测量系统结构

Fig.8 Intelligent measurement system architecture

队列中会缩短。优先级最低的队列 Q_N 为了方便那些超大作业的完成,时间片一般都很大。

MLFQ调度的重点在于调度器如何设定优先任务。MLFQ并不固定地赋予每项任务优先级。而是随着任务的运行动态,对优先级进行调整。如果一个任务不停地放弃CPU,比如在等待I/O完成,MLFQ会保持它的高优先级,因为它是个交互式任务。相反如果一个任务占用CPU很长时间,也就是长任务,MLFQ会降低其优先级。通过这种方式MLFQ会随着任务的运行,不停地学习,进而预测任务未来可能的运行方式。当一个长任务和交互式任务同时运行时,交互式任务一直在最高优先级队列中,长任务不停地运行到最低优先级队列中,交互式任务仿佛穿插运行在长任务中一样。这样互动式的任务,反应时间很好,而长任务也获得了良好的周转时间。所以MLFQ的4个基本规则可以提炼如下:

规则1: $\text{Priority}(Q_1) > \text{Priority}(Q_2)$, 则运行A, 而B不运行。

规则2: 如果A, B优先级相同, 那么A和B都轮流运行, 采用轮询调度策略。

规则3: 当任务到达系统时, 把它放在最高优先级队列中。

规则4: 如果任务用完整个时间片, 把它从高优先队列往下移动一级; 如果工作在其时间片以内主动释放CPU, 则优先级不变。

目前的MLFQ, 看起来似乎是公平的, 对长任务来说会公平地分享CPU, 对短任务或者IO敏感的交互式任务也能更快地运行。然而, 这种实现方式还有严重的问题, 就是“饿死”问题, 如果系统中有太多的交互式任务运行, 他们加起来会消耗掉所有的CPU时间, 则长任务不可能获得任何的CPU时间(称为“饿死”)。

随着时间的改变, 任务的类型也会发生改变。CPU密集型的程序可能会在后面变成交互型的程序, 按照这种策略, 到了最低优先级后无法再往上移动, 也就无法重新被系统当作是交互式任务。

这个问题基本的解决思路就是周期性地提升系统中任务的优先级, 把任务放到最高优先级队列中, 因此可以提出一条新规则5: 在一段时间S后, 将所有任务移至最高优先级队列。

S的大小值得关注, 若S值设置过大, 长任务“饿死”时间会变长; 若S值设置过小, 交互式程序就不会获得本该公平的CPU资源, 如图9所示。

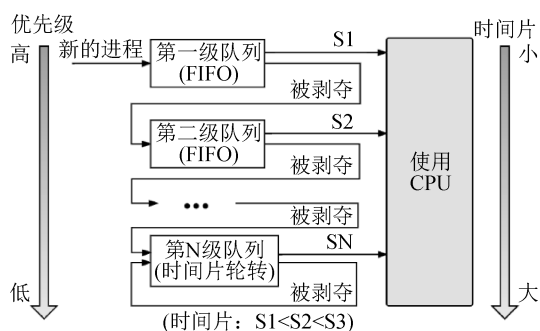


图9 设置S值

Fig.9 Setting of S value

引用新规则后解决了两个问题, 保证不会有任务被饿死。当CPU密集型的程序变成交互式程序时, 能再次获得调度器的信任。首先, 这个过程不会饿死, 在最高优先级队列中, 它会以轮转的方式将CPU分享给其他高优先级的工作, 这样才能最终得到执行; 其次, 如果一个CPU密集型的工作变成了交互, 那么当它的优先级提高时, 能再次获得调度器的信任。

3.2 中断控制设计

鸿蒙操作系统的中断均采用中断向量表的方式进行处理, 中断号是一个特定的中断请求信号

标志, 与中断号对应的是中断向量, 中断向量是一个入口地址, 用于中断服务程序。中断向量表是所有中断处理程序的入口, 处理过程是将一个函数与中断向量表中的中断向量在一个虚拟中断向量表中连接起来。当中断向量对应中断发生时, 会调用已挂断的用户中断服务程序执行。当一个中断触发时, 处理器会跳转到中断源相应的位置进行处理, 并且每个中断服务程序必须排列在一个统一的地址上。中断向量表一般是由一个数组定义出来的, 或者在起始代码里给出中断向量表。如图10所示。

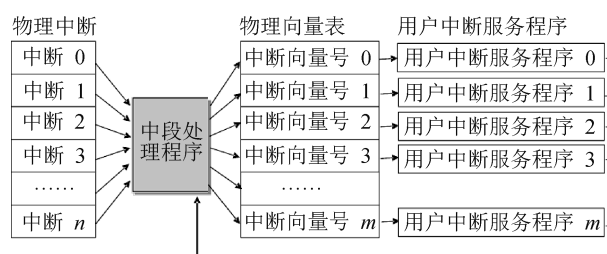


图10 中断向量表

Fig.10 Interrupt vector table

为方便测量仪器的使用, 特别选择LOSCFG_PLATFORM_BSP_GIC_V2作为中断控制器, 对应中断控制过程如图11所示。

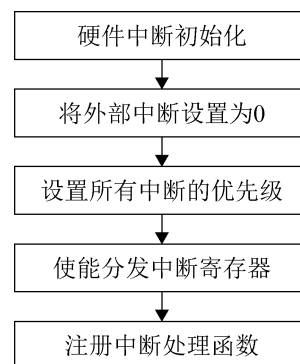


图11 中断控制过程图

Fig.11 Interrupt control diagram

3.3 测量方法逻辑判别

当前的自动化检定软件普遍是由上位机控制、读取测量仪器传输回的数据并进行数据处理, 大部分操作必须由计量人员手工完成, 并且无法进行规程要求的环境实时监测等功能。

在对鸿蒙系统进行任务调度处理和中断控制设计后, 测量系统可以在计量任务中合理的调度如逻辑判断、命令识别、自采集、自适应、自检验与故障自诊断等。经过对鸿蒙开发板的模块化

改造与本次任务的数字化计量规程规范的分析，程序可对传感器、环境条件进行自检与故障自诊断，然后提取规程规范中的操作步骤，包括提供计量参量的仪器控制、计量过程控制、通过算法的逻辑判别完成命令的识别和数据的自动采集与数据处理。在测量过程中可对测量仪器进行自动切换量程、自动适应测量参数，并在监控终端对测量结果进行自适应显示。通过软时钟进行高精度的时序控制，利用优化后的任务调度和中断控制算法增强仪器软件实时测量控制的性能。

3.4 自校准与实时校准

传统上位机一般只能记录测量仪器传回的原始数据，并且无法集成对仪器的校准工作，而智能测量仪器可借助鸿蒙系统集成的传感器自校准方法在检定前对传感器进行零位校准，标准校准等。在计量过程中，根据当前接入的仪器数据对传感器采用校正算法消除系统误差，实现自校准和自补偿，完成对传感器测量误差的实时修正，并且提供数字滤波算法，消除随机误差，实现测量数据的噪声抑制。还可以利用数据处理技术对仪表进行非线性特征的自校正，自动按照对应的非线性特性进行转换，实现测量系统输入与输出的理想直线关系，实现方法多采用分段线性插值法和曲线拟合法。

引入鸿蒙系统后，利用其软时钟的特性，为测量系统设置一个整体时钟信号，来保证每个测量模块时钟的准确性和同步性，可以避免不同测量时用到的仪器接收指令和处理数据的时钟不一致，实现在测量条件迅速变换下的实时自校准。同时，由于测量系统需要定时修正误差，普通智能测量仪器在这部分的开销较大，而鸿蒙系统可通过分布式软总线合理分配资源，使自校准任务不会过度占用采集资源与数据处理的资源。

3.5 测量数据的扩展性与准确性保证

鸿蒙系统对计量设备的智能化改造大幅度减少了整体系统的开销，并加入了自组网、软总线等功能，可以拓展计量任务的空间范围、被检数量等。同时，可以在测量仪器到鸿蒙开发版之外拓展高精度环境信息监控等，与规程规范中的要求进行比对，保证量值的准确性。由于鸿蒙分布式节点造价低的特点，在特殊计量条件下，例如恶劣环境、非标计量要求等，可以设置多台计量设备

终端作为冗余，保证链路的通畅以及数据的准确。

同时，分布式软总线的软时钟可以确保远端计量设备传输数据的实时性，外加对计量参量的仪器控制，使数据更加准确。用户可通过手机、平板等汇总设备实时监控每一台设备的运行情况以及环境信息，也可以单独查看某一台远端设备的状态，同时可以控制此次计量任务的进行。

得益于鸿蒙对任务优先级的分配，一般计量任务所用到的业务也可根据重要性进行优先级分布，如表4所示。

表 4 计量任务优先级排布
Tab.4 Measurement task prioritization

分配业务	信道/ 个	数量/ 个	RSSI	负载	优先级
数据采集	36	6	-35	600	优先
环境监测	23	12	-35	300	非优先
参量仪器控制	155	1	-75	0	非优先
校准计算	50	48	-75	600	优先

对于数据采集设备，尽量提升其资源占用以保证计量数据顺利地快速回传，同时调用鸿蒙系统的任务总线，将原始记录 and 校准修正计算保存至分布式终端中处理性能较强的手机、平板设备，同时可以设计对应专业的计量软件，进行原始记录表格的生成以及证书的上传。在全程的低功耗环境监测下，整体任务完毕后，用户便可选择将数据上传至云端或快速出具校准证书，大大降低了数据处理的时间和人力成本，同时大幅提高了数据的准确性。

3.6 用户界面显示

传统的仪表由于专注于测量的准确性，往往只提供最基本的字符显示功能，如果对仪器不熟悉，需要耗费很长时间学习操作。而鸿蒙系统则可通过向仪器直接发送指令，而将晦涩难懂的仪器指令变成可视化操作界面，通过增加辅助功能，大大增加了仪器的可用性。

开启鸿蒙终端 APP 后，完成主窗口的创建。首先创建事件，初始化定时器，定时器每隔 1 s 显示时间，向开发板发送指令。UDP 开始监听特定端口，UDP 接收到数据产生中断事件，进行显示。设置时钟 1，使其每隔 1 s 在右上角显示当前时间，并通过 UDP 以“GBK”的数据格式发送命令“sendcmd”，IP 地址为当前 IP 地址 192. 168. 1. 169，

UDP在192.168.1.-网段内监听,监听到数据后触发收到数据事件,将接收到的字节数据转化为文本,分别放到各个标签中,即可完成显示。

使用标签、按钮、时钟、数据报等组件进行页面布局设计,鸿蒙终端APP的主界面如图12所示。

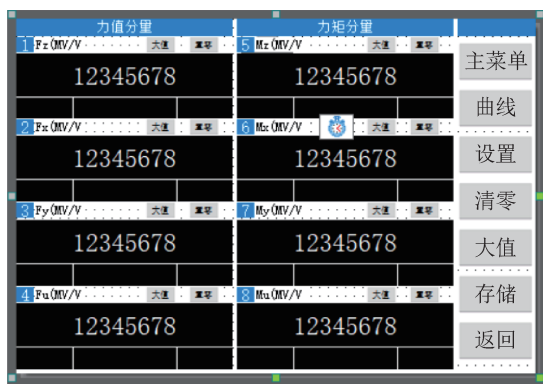


图12 鸿蒙终端APP的主界面

Fig.12 Main interface of Harmony terminal APP

3.7 硬件功能拓展

在鸿蒙操作系统环境中实现智能测量仪器的开发,脱离了传统的C51, STM32等嵌入式平台,采用可运行鸿蒙系统的海思Hi3861芯片作为主控芯片。该芯片集成了32位高性能微处理器,包含硬件安全引擎,并且具备丰富的外设接口,包括SPI, UART, I2C, PWM等,可通过对GPIO口的编程与配置,实现与测量系统中的各种测量仪器、数据采集卡、信号放大器、主传感器、附加传感器、传感器模拟器等设备的连接,通过串行和并行的通信接口组成测量控制系统,可提高智能测量仪器的硬件拓展性和兼容性。同时,基于鸿蒙系统的分布式软总线特性,合理分配并实时监控各任务和线程的通信接口,使硬件设备更便捷地与上位机终端设备进行连接和通信,实现对输入输出设备的自动控制与管理。利用Wi-Fi结合UDP协议传输的网络特性,增强智能测量仪器分布式测量与控制性能。

串口通信^[16]是最基本的单片机通信,在鸿蒙开发板的海思Hi3861模块中,用于串口通信的UART2接口使用GPIO11和GPIO12管脚。在通信过程中,首先实现UART2的初始化,编写初始化函数、读数据函数以及收发数据函数,将接收到的数据再次发送出去。在开发过程中,需对UART2进行使能,初始化GPIO为UART2,对GPIO11和GPIO12进行初始化,接着配置UART2

的波特率、数据位、停止位、奇偶校验和流控,设计数据传输的协议和格式,然后在任务中调用UART2串口读写函数和任务处理函数,最后对业务代码和模块的脚本进行编译。

4 总结

在分析了鸿蒙系统的技术架构以及分布式特性后,发现其在可靠性、安全性、连接性方面均有较高的优越性。和已有的Linux系统进行比较,鸿蒙系统的内核移植性好、系统稳定且开销较小。本文设计了基于鸿蒙系统的智能测量系统,验证了鸿蒙系统可以在任务调度与中断控制、自动采集、自动校准、数据传输、用户界面和硬件通信方面,满足智能测量仪器对嵌入式操作系统的需求。

将鸿蒙系统根据测量系统的特点进行优化,结合其分布式软总线与软时钟的特性,可为测量系统提供更智能、实时性更强的操作平台。将来可在长、热、力、电等测量领域进行功能拓展,利用鸿蒙系统的多任务开发模式,不断更新配置功能,从而实现全面的智能测量云平台。

参考文献

- [1] 周宁. 关于计量科学技术发展若干问题的思考[J]. 计测技术, 2022, 42(1): 8.
ZHOU N. Perspectives on the development of metrological science and technology [J]. Metrology and measurement technology, 2022, 42(1): 8. (in Chinese)
- [2] 何旋, 吴霞, 周自力. 对计量内涵及其发展重点的思考[J]. 计测技术, 2022, 42(3): 79-85.
HE X, WU X, ZHOU Z L. Understanding of metrology connotation and thoughts on development focus [J]. Metrology and measurement technology, 2022, 42(3): 79-85. (in Chinese)
- [3] 王腾飞. 对计算机嵌入式实时操作系统的研究及分析[J]. 科技创新与应用, 2020(36): 2.
WANG T F. Research and analysis of computer embedded real-time operating system [J]. Technology innovation and application, 2020(36): 2. (in Chinese)
- [4] 宋然, 严国萍. 嵌入式实时操作系统(RTOS)研究[J]. 计算机测量与控制, 2001(z1): 3.
SONG R, YAN G P. Research on embedded real-time operating system(RTOS)[J]. Computer measurement and control, 2001(z1): 3. (in Chinese)
- [5] 董思乔, 赵荣建, 孙通. 基于Wi-Fi构建的智能家居控制系统的设计[J]. 电视技术, 2015, 39(4): 89-91.

- DONG S Q, ZHAO R J, SUN T, et al. Design of a smart home control system built on Wi-Fi[J]. TV technology, 2015, 39(4): 89-91. (in Chinese)
- [6] 孙忠祥. 基于设备云平台的智能农业温室大棚远程监控系统的实现[D]. 哈尔滨: 哈尔滨理工大学, 2017.
- SUN Z X. Implementation of remote monitoring system for intelligent agricultural greenhouses based on equipment cloud platform[D]. Harbin: Harbin University of Technology, 2017. (in Chinese)
- [7] 郭堂伟, 周涵, 俞啸, 等. 基于 ARM 嵌入式的医院空气质量实时检测系统设计与研究[J]. 计算机测量与控制, 2021, 29(4): 5.
- GUO T W, ZHOU H, YU X, et al. Design and research of ARM embedded-based hospital air quality real-time detection system[J]. Computer measurement and control, 2021, 29(4): 5. (in Chinese)
- [8] 杨文豪. 基于 PXI 总线的高速图像采集模块的设计与实现[D]. 太原: 中北大学, 2017.
- YANG W H. Design and implementation of high-speed image acquisition module based on PXI bus[D]. Taiyuan: North Central University, 2017. (in Chinese)
- [9] 刘恩凡, 于洪涛. 基于 VxWorks 的某装备自动测试系统设计与实现[J]. 计算机测量与控制, 2016, 24(3): 3.
- LIU E F, YU H T. Design and implementation of an equipment automatic test system based on VxWorks[J]. Computer measurement and control, 2016, 24(3): 3. (in Chinese)
- [10] 张明利, 许清清, 李阿明, 等. 基于 VxWorks 的嵌入式多协议通信控制器设计[J]. 仪器仪表用户, 2022, 29(3): 4.
- ZHANG M L, XU Q Q, LI A M, et al. VxWorks-based embedded multi-protocol communication controller design[J]. Instrumentation user, 2022, 29(3): 4. (in Chinese)
- [11] 李明. 全自动生化分析仪测控系统设计[D]. 南京: 南京理工大学, 2014.
- LI M. Automatic biochemical analyzer measurement and control system design[D]. Nanjing: Nanjing University of Technology, 2014. (in Chinese)
- [12] 盛蔚, 贺彪, 高彤. 基于 RT-Thread 的无人机数据记录仪设计与实现[J]. 电子技术应用, 2018, 44(4): 4.
- SHENG W, HE B, GAO T. Design and implementation of RT-Thread-based UAV data logger[J]. Electronic technology application, 2018, 44(4): 4. (in Chinese)
- [13] 莫小凡, 王晓荣, 储震宇, 等. 基于 STM32 和生物传感器的葡萄糖检测仪设计[J]. 仪表技术与传感器, 2019(11): 4.
- MO X F, WANG X R, CHU Z Y, et al. Design of glucose detector based on STM32 and biosensor[J]. Instrumentation technology and sensors, 2019(11): 4. (in Chinese)
- [14] 薛东亮. 基于嵌入式的物联网底层数据交换系统的设计与实现[J]. 自动化与仪器仪表, 2018(10): 4.
- XUE D L. Design and implementation of an embedded-based IoT underlying data exchange system[J]. Automation and instrumentation, 2018(10): 4. (in Chinese)
- [15] 李欣, 白兴武. 基于 Linux 的嵌入式实时操作系统任务调度算法优化[J]. 自动化与仪器仪表, 2020(9): 4.
- LI X, BAI X W. Optimization of task scheduling algorithm for Linux-based embedded real-time operating system[J]. Automation and instrumentation, 2020(9): 4. (in Chinese)
- [16] 杨文豪, 焦新泉, 储成群. 基于嵌入式操作系统 VxWorks 的 PXI 实时读写接口设计[J]. 仪表技术与传感器, 2016(10): 98-101.
- YANG W H, JIAO X Q, CHU C Q. Design of PXI real-time read/write interface based on embedded operating system VxWorks[J]. Instrumentation technology and sensors, 2016(10): 98-101. (in Chinese)

(本文编辑: 郑燕)



第一作者: 王晶星(1997—), 女, 硕士研究生, 研究方向为嵌入式测量仪器。



通讯作者: 李新良(1972—), 男, 研究员, 主要从事动态力学量计量溯源领域相关研究。



通讯作者: 李维(1986—), 男, 高级工程师, 博士, 硕士生导师, 主要从事芯片化计量与传感相关研究。