

通用半实物仿真资源管理器的设计

金鹏飞, 李晓明

(浙江理工大学机械工程学院, 浙江 杭州 310018)

✉ 1714168788@qq.com; lxmzist@zstu.edu.cn



摘要: 半实物仿真系统包含各种各样的前端模块, 如何管理这么多复杂的前端模块就成为该系统的一个重要难题。文章提出通用半实物仿真资源管理器的设计方案, 将前端模块的属性分为动态属性和静态属性, 并设计出与其对应的前端设备的 XML 文档, 制定出前端模块与主机模块的通信协议, 为前端模块的通用化制定了标准。经过实验表明, 该管理软件能够实时对各种前端设备进行管理, 并且不需要过多的操作, 就能在管理软件中展示出前端设备的属性信息, 达到即插即用的效果。

关键词: 静态属性; 动态属性; XML 文档; 前端模块

中图分类号: TP311.5 **文献标识码:** A

Development of General Semi-physical Simulation Resource Manager

JIN Pengfei, LI Xiaoming

(College of Mechanical Engineering, Zhejiang Sci-Tech University, Hangzhou 310018, China)

✉ 1714168788@qq.com; lxmzist@zstu.edu.cn

Abstract: Semi-physical simulation system contains a variety of front-end modules, and how to manage so many complex front-end modules has become an important challenge of the system. The paper proposes a design scheme for a general semi-physical simulation resource manager, which divides the attributes of the front-end module into dynamic and static attributes. An XML document corresponding to the front-end device is also designed, and the communication protocol between the front-end module and the host module is developed, which sets standards for the generalization of the front-end module. The experimental results show that the proposed management software can manage various front-end devices in real-time and display the attribute information of front-end devices in the management software without excessive operations, so as to achieve the effect of plug and play.

Keywords: static attributes; dynamic attributes; XML document; front-end module

1 引言(Introduction)

模块化、通用化、可扩展是虚拟仪器^[1]的重要特征, 一台虚拟仪器系统的不同功能模块可能是由不同的仪器厂家生产的。如果想确保市场上各种各样不同功能、不同厂家生产的仪器模块能够相互组合成为一台完整的仪器系统并能正常运行, 那么对这些仪器模块进行管理就起到了关键性的作用。虽然各类管理软件产品类型及数量众多, 但是仍主要采取各自的体系架构, 在技术内部融合及接口一致性处理方面面临的技术难度比较大^[2-5]。

针对上述问题, 杨浩前的研究实现了设备资源的自我控制和自我管理, 使用户摆脱了烦琐的底层硬件接口驱动命令,

但是杨浩前的方案却没有包含所有的数据类型, 并且在数据传输的过程中会出现中断的错误^[6]; 赵俊设计的 IVI(互换虚拟仪器)设备驱动分成仪器类驱动和专用驱动两层, 更改函数时, 只需要改变其配置文件, 而不需要改变上层程序, 但是赵俊的方案没有实现相同仪器间的互换, 没有实现各个仪器之间的通用^[7]。为了解决上述设备管理器方案中存在的缺点, 本文设计出一种资源管理器的方案, 包括资源管理器的功能, 前端模块^[8]资源的设计和与之对应的 XML^[9]配置文件, 以及前端模块和主机模块的通信协议, 实现了半实物仿真的标准化和通用化^[10]。

2 资源管理器的设计内容(Design content of resource manager)

本文研究的内容是通用半实物仿真平台资源管理的设计。这些资源的主要目的是作用于大部分前端设备。其中，需要实现的功能包含设备树的自动生成、设备运行状况的实时监测和前端设备的管理。通过该资源管理系统可以在开发平台对前端硬件进行查看、监测数据文件传输及更新前端设备。因此，需要一套符合整个管理软件开发平台的前端设备管理器，用于在开发时对当前所连接的前端设备进行管理和状态监控。基于此，课题的研究内容主要包括以下两点。

(1)前端模块管理功能：本系统具备硬件模块化特点，主机模块是系统的控制中心。管理软件运行于主机模块之上，对前端模块进行控制，与前端模块进行信息交互，实现平台的测试测量和监测功能。该管理软件还需要对当前系统内的前端模块及其运行状态信息进行管理。

(2)状态记录功能：系统中存在很多的特性或状态需要量化表达。这些状态有些用于前端模块的管理，例如模块的ID、生产厂家ID等；有些状态用于配置硬件参数，控制平台的运行，例如与总线相关的前端模块地址滤波器值及基本时间周期值等；有些状态与总线等硬件状态密切相关；有些状态仅用于底层的功能调整或者表达软件的运行状态；有些状态表征系统的运行状态，需要实时更新获取，例如某个前端模块的总线丢帧情况等。因此，需要建立一套标准机制记录整个系统的状态运行参数。

3 资源管理器的实现(Implementation of resource manager)

3.1 前端模块的建模

对于以上需要实现的功能，整个资源管理器中关键的部分就是对其设备的管理，而设备管理器系统中最主要管理的对象之一就是形形色色的前端模块。因此，针对该系统建立了相应的模型图，如图1所示。

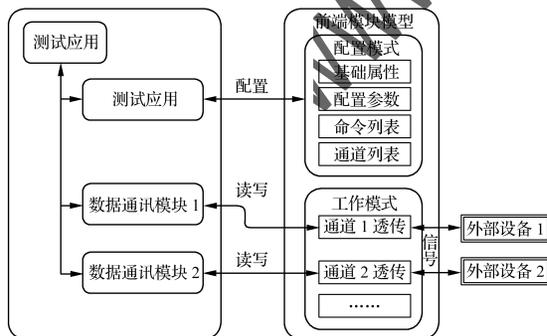


图1 设备管理器前端模型
Fig.1 Front-end model of the device manager

前端模块^[10]是组合式仪器平台里的一个组成部分，由DSP(数字信号处理)控制板、总线板和外部接口模块组成，其中总线板实现了前端模块与主机及其他前端模块之间的通信，总线板里的总线是系统内置的仪器总线；DSP控制板主要是一个DSP控制系统，包括DSP和存储系统等；外部接口模块是前端模块与被测实物之间的信号交流接口，例如

RS232^[11]、1553B总线接口^[12]及CameraLink接口^[13]等。

3.2 前端模块资源的设计

前端模块资源用户管理前端应用扩展模块的属性信息和状态，记录模块的属性状态，提供获取和配置属性状态的接口。需要针对前端模块的硬件特征和测试系统的软件特征，抽象模块特性，在总线驱动中量化表达。对属性特征进行分类，不同类型的属性有着不同的获取和配置方式。结合前端模块的特性，将属性分为静态属性和动态属性，静态属性包含前端模块的厂家ID等所有前端模块都具有的属性，而动态属性是某些前端模块独有的，就需要专门的代码实现，进而实现资源管理的通用化。现阶段对前端模块资源属性的分类见表1。

表1 前端模块属性分类表

Tab.1 Classification table of front-end module property

静态属性	动态属性
模块厂家ID	消息通道基本时间周期
版本号	消息通道固定时间窗长度
RAM	前端模块通道的个数
DDR2	各个通道中波特率

3.2.1 XML的设计

通用化设计标准就是设计有标准的接口，能够让所有的其他组件在使用这个数据传输组件时，只要根据标准的接口进行设计就能完成数据的传输。那么，对于各式各样的组件，传输的数据格式和要求是不一样的，那么至于这些参数的传入就可以使用XML文档进行配置^[9-10]。XML可配置参数有数据类型、数据属性、输入端口及输出端口等。本文用XML语言描述前端模块提供的资源与功能，包括基础属性、配置参数、命令列表、通道列表。因此，为了实现数据标准化，设计了对应的XML文档，具体实现代码(节选)如下：

```
<module id="1" available="true">
  <name>IO板1</name>
  <type>IO板</type>
  <handleType>DSP6748,FPGA</handleType>
  <handleTypeFre>450MHz</handleTypeFre>
  <version>1.2</version>
  <factory>先进研究所</factory>
  <state>运行正常</state>
  <RAM>128MB</RAM>
  <DDR2>2GB</DDR2>
  <FLASH>2GB</FLASH>
  <channelType>16路输出口,32路输入口</channel-
Type>
  <channels>
    <channel id="1">
      <name>输入1</name>
      <state>正常</state>
      <baudrate>3v</baudrate>
    </channel>
    <channel id="2">
      <name>输出2</name>
      <state>正常</state>
      <baudrate>3v</baudrate>
    </channel>
  </channels>
</module>
```

3.2.2 JavaBean 的设计

数据经过 XML 文档传输之后会被相应的解析器解析，而数据经过解析后应该有其进行标识，这样可以在之后的数据处理中提供映射。基于此，设计了对应的 JavaBean 作为数据的载体，解析数据字段表述见表 2。

表2 解析数据字段表述表

Tab.2 Parsing data field description table

属性名	描述信息
frameName	数据帧名称
frameAddr	数据帧地址
blockName	字段名称
valueName	数据名称
valueType	数据类型
deviceId	设备 ID
Buffer	数据原字节

3.3 前端模块与主机模块的通信协议设计

3.3.1 协议需求分析

为了满足以上两种工作模式的需求，设计了前端设备与应用软件之间的传输层协议。该协议主要是根据传输的数据类型与功能指定的。传输的内容大致可以分为两大类(图 2)：一是工作模式，这种模式下可以与主机模块一对一地通信、进行模块注册、接收资源查询和配置命令、接收状态查询命令、接收 DSP 应用安装命令及接收 DSP 程序信息查询命令。二是用户模式，在该模式下能接收消息日志、数据和退出用户模式的命令。当前端模块执行 DSP 程序后会自动进入用户模式。

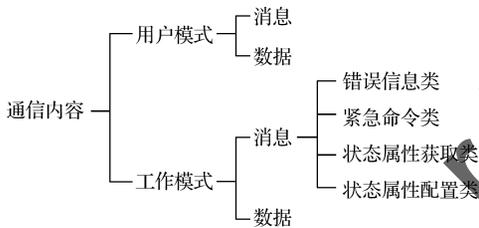


图 2 数据传输图

Fig. 2 Data transmission diagram

3.3.2 通信协议的制定

通信协议的数据链路层帧结构如图 3 所示，主要包含仲裁场和数据场。仲裁场的功能是首先根据数据的优先级进行判断，然后判断该数据是否可以被接收。数据场主要是发送数据，用户在数据场中可以撰写自己需要的协议，之后形成 TLV(Tag、Length、Value)格式。对于在前端模块上传输的数据，将仲裁场定义为数据帧头，将数据场定义为数据体。数据的数据场是一个长度为 1 024 字节的字符数组，消息的数据场是由 3 个无符号整型数构成的，长度为 12 字节。

	1 bit	16 bit	2 bit	32 bit	1 024 byte	16 bit
数据通道	帧起始位	仲裁场	应答场	数据长度	数据场	CRC 场
	1 bit	16 bit	2 bit	32 bit	12 byte	16 bit
消息通道	帧起始位	仲裁场	应答场	数据长度	数据场	CRC 场

图 3 总线数据链路层帧结构

Fig. 3 Layer frame structure of bus data link

根据资源管理器分析的协议需求和通信特性，将不同功能、不同类型的数据和消息按照不同的通信方式和优先级发

送。针对前端模块的应用情况，对仲裁场和数据场进行定义。对仲裁场的定义如图 4 所示，图 4 中的 type 部分是提供给用户的应用软件使用的，实现了通用化和标准化。

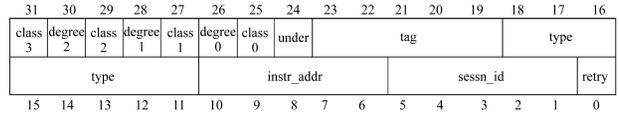


图 4 消息和数据仲裁场协议

Fig. 4 Message and data arbitration field protocol

根据仲裁场中的协议规定，数据和消息的优先级主要是依靠协议中的前 7 位来决定(31—25 位)。在仲裁场的协议中，信息的类别 class 和优先程度 degree 共同影响着数据的优先权。尽管低位的值可能会对优先权产生影响，但它的影响力不及高位的值，因此最后选择前 7 位作为重要的评估标准。数据信息的优先传输级一般来自数据信息大类 class，但是通过改变 degree 的方式，也能够有效地调整默认数据类型的优先程度。将数据信息大类 class 和 degree 创新性地结合起来，促使它们之间的关系变得平衡，从而取得最佳的效果。class 的取值范围能够从 0 到 F 不等，它能够支持 16 种消息类型和 16 种数据类型，这样能够允许低优先级的数据类型在某些情况下，具有更高的优先权，从而使数据传输更为便捷，更具有层次感。

仲裁场中的 under 字段用来判断通信是上位机管理软件(上层)还是虚拟仪器嵌入式系统(底层)。1 代表底层通信，0 代表上层通信。tag 用来指定底层通信中的指令类型，type 用来指定上层通信中的指令类型。用户可以在 type 中根据自身的需求，设置对应的协议。instr_addr(10—6 位)和 sessn_id(5—1 位)分别是底层通信和上层通信的地址，取值范围是 0—F(Hex)。retry 用来显示数据的发送是否发生了丢帧，正常情况下数字为 0，当从 0 变成 1 时，就表示出现了丢帧，前端模块通信就会对数据丢帧进行检测。如果检测后发现数据丢失或者出现错误，前端模块驱动会自动进行重发。

数据场的具体设计如图 5 所示。其中，包含关于消息的数据和前端模块采集的数据。消息总线数据场的首半个字节为消息长度，其余为消息内容；数据总线数据场的前两个字节为数据长度，其余为数据内容。在数据内容的最后会跟着一个编号，它是一个递增数，用于丢帧检测。每当数据传输成功后，这个编号就会自动加 1，当接收端接收到的数据是断断续续的，说明出现了数据丢失，就需要通知发送端重新发送丢失的数据帧。

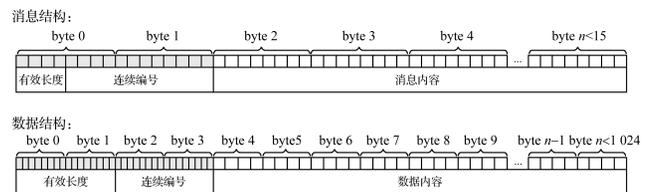


图 5 消息和数据总线数据场协议

Fig. 5 Message and data bus data field protocol

3.3.3 通信协议的设计

设备管理器连接到主机模块，通过主机模块接口调用 VISA(虚拟仪器软件结构)接口，连接到前端模块进行获取。

需要与前端模块通过仪器总线进行交互，按照前文对协议进行分析和制定，前端模块与主机通信协议节选见表3。

表3 前端模块与主机通信协议节选

Tab. 3 Excerpts from the communication protocol between the front-end module and the host

数据	标签	长度	数据类型	值说明
资源描述数据	0x01	N	Byte[N]	具体内容与前端模块的配置有关
状态描述字	0x02	1	Byte	0:正常;1:DSP异常; 2:接口异常;4:电源异常; 8:总线异常;16:其他异常
模块配置数据	0x03	N	Byte[N]	参考资源表
操作结果	0x04	1	Byte	0:成功;其他:错误字
资源ID	0x05	1	Byte	参考资源表
资源值	0x06	N	Byte[N]	参考资源表
长度	0x07	4	UINT32	参考资源表
包数	0x08	4	UINT32	参考资源表
包索引	0x09	4	UINT32	参考资源表
包数据	0x10	N	Byte[N] N<1 024	参考资源表
DSP接收状态	0x11	1	BYTE	0:未收到;1:接收中; 2:丢数据;3:超时; 4:校验错;5:成功
DSP应用信息	0x12	N	BYTE[N]	“版本号;安装序列号”
工作模式	0x13	1	Byte	0:系统;1:用户

4 实例演示(Example demonstration)

本文所提软件是基于浙江大学课题组研发的通用测试仪器系统，实物如图6所示。



图6 仪器系统实物图

Fig. 6 Physical diagram of instrument system

通过前端模块和上位机软件进行通信后，运行上位机管理软件，可以实时展示前端设备的数据，能够对前端设备进行管理。如图7所示，界面左侧能够显示前端设备的可用状态，界面右侧是对单个前端设备具体信息和数据的展示，界面的下方是日志的输出。

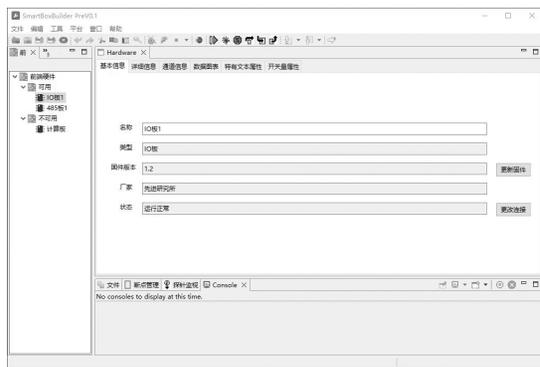


图7 上位机管理软件界面展示

Fig. 7 PC management software interface display

5 结论(Conclusion)

本文分析了资源管理器技术的研究现状，针对浙江大学课题组研发的多功能仪器组件前端设备的复杂性，设计相对应的资源管理器方案。对前端模块进行抽象建模，设计前端模块的资源管理应用程序，并模拟实际测试案例，验证通用半实物仿真资源管理器及通信协议标准化方案的可行性，可以实现软件中关于各种前端模块中设备树的自动生成、IO通信、属性管理、事件通知等全部功能，满足了资源管理器的应用需求。最重要的是，本设计方案只要配置对应的XML文档，就可以对前端模块即插即用，应用程序开发非常简单，具备标准化、通用性、避免软件重复开发的特点。

参考文献(References)

- [1] 郭荣斌,赵秀才.自动测试系统的发展趋势[J].国外电子测量技术,2014,33(6):1-4.
- [2] 齐永龙,宋斌,刘道煦.国外自动测试系统发展综述[J].国外电子测量技术,2015(12):1-4.
- [3] BALLESTEROS J, FERNANDEZ PALOP J I, HERNANDEZ M A. LabView virtual instrument for automatic plasma diagnostic[J]. Review of Scientific Instruments, 2004, 75(1):90-93.
- [4] BAROTH E, BRUNZIE T, HARTSOUGH C. An update on visual programming data acquisition software tools[J]. EE: Evaluation Engineering, 1996, 35(2):26-36.
- [5] 路亚峰,陈义军,温新歧,等.虚拟仪器技术研究现状与展望[J].国外电子测量技术,2010(11):35-37.
- [6] 杨浩前.基于VISA架构的物联网资源驱动层研究与实现[D].西安:西安石油大学,2015.
- [7] 赵俊.通用测试仪器可互换引擎IVI-C的研究与实现[D].成都:电子科技大学,2013.
- [8] 李佳林.模块化测试仪器底层通用软件的标准化方法研究[D].杭州:浙江大学,2017.
- [9] 何卓桁,刘志勇,李璐,等.异构文本数据转换中XML解析方法对比研究[J].计算机工程,2020,46(7):286-293,299.
- [10] 冯安,冯福来.虚拟仪器技术的测试系统设计与标准化运用[J].航空标准化与质量,2007(6):35-38.
- [11] 宋丫,许亚星,李庆楠.一种基于RS232总线的通信网络数据传输监测方法[J].数码世界,2020(11):26-27.
- [12] 张志国,刘学.1553B总线数据快速处理软件的设计与实现[J].电子设计工程,2022,30(19):6-9,14.
- [13] 单彦虎,张晋项,任勇峰,等.基于FPGA的CameraLink图像数据接口设计[J].仪表技术与传感器,2020(9):51-54.

作者简介:

金鹏飞(1996-),男,硕士生.研究领域:机电系统集成,软件设计.

李晓明(1976-),男,博士,副教授.研究领域:机电系统集成.