

doi:10.19306/j.cnki.2095-8110.2021.01.020

面向功能的测试设备驱动器设计与实现

秦振汉¹, 胡广明¹, 张辉², 郭双红¹

- (1. 航天科工惯性技术有限公司, 北京 100074;
2. 陆军装备部驻北京地区航空军事代表室, 北京 100074)

摘要:在各种定制化测试设备中,由于很多仪器驱动不符合现有的标准,从而降低了测试软件的开发效率和质量。为此,提出了一种面向功能的测试设备驱动器的设计与实现方法。通过对特定类别测试设备所需实现功能的分析,获得了脱离硬件环境的功能项目集合,并建立了规范化的设备驱动接口。在驱动组件的内部,封装了对硬件仪器的控制,并描述了实际测试设备的功能实现。测试设备驱动器为上层软件提供了统一的开发和运行基础,便于测试程序针对这些虚拟化的功能接口进行开发,避免了硬件的差异性对业务逻辑的影响。应用实例表明,该方法提高了测试程序的移植性和开发质量,尤其适用于各种系列化的、含有非标准驱动程序的测试软件的开发。

关键词:驱动器;功能项目;自动测试设备;驱动接口

中图分类号:TP311.5

文献标志码:A

开放科学(资源服务)标识码(OSID):

文章编号:2095-8110(2021)01-0176-07



Design and Implementation of Function Oriented Test Equipment Driver

QIN Zhen-han¹, HU Guang-ming¹, ZHANG Hui², GUO Shuang-hong¹

- (1. Aerospace Science & Industry Inertial Technology Co., Ltd., Beijing 100074, China;
2. Aviation Military Representative Office of the Army Armaments Department in Beijing Area, Beijing 100074, China)

Abstract: As some instrument drivers do not meet the existing standards in a variety of customized test equipment, which reduces the test software development efficiency and quality, a design and implementation method of function oriented test equipment driver is proposed. A set of functional items separated from the hardware environment is obtained and a standardized device driver interface is established via the analysis of the functions of the specific type of test equipment. The control of the hardware instrument is encapsulated in the driver component and the function realization of the actual test equipment is described. The test equipment driver provides a unified development and operation basis for the upper software enabling the test program development for these virtualized functional interfaces to avoid the impact of hardware differences on business logic. The application shows that this method improves the portability and development quality of test program, especially for the test software development of various series with non-standard driver.

Key words: Driver; Functional item; Automatic test equipment; Driver interface

收稿日期:2020-05-12;修订日期:2020-06-11

作者简介:秦振汉(1981-),男,硕士,工程师,主要从事测试设备测试软件的研发。E-mail:qzh13651056530@126.com

0 引言

在自动测试系统中,一般都会在测试程序和硬件仪器的驱动程序之间建立一个软件中间层,以连接计算机和各种不同测试仪器^[1-2]。仪器驱动器是一组用于控制程控仪器的软件单元,提供了规范化的软件接口,简化了具体的编程步骤,便于实现仪器功能的操作^[3-4]。

测试仪器的种类、功能众多,针对仪器驱动的标准问题,国内外进行了大量的工作,陆续推出了多种标准和规范,目的是提高仪器的可互换性和通用性水平。常用的仪器驱动标准包括 SCPI、VPP、IVI-C、IVI-COM 等,并在测控领域得到了广泛的应用^[5-7]。这些标准的基本思想是把仪器的操作封装成相同的接口,其函数名称和参数完全相同,解决了仪器之间互换的问题^[8-9]。

但在实际工程项目中,存在各种定制化的测试系统,内部集成了大量的非标准仪器和板卡。这些测试仪器和板卡来自不同的厂商,总线形式不一,其驱动程序未遵循现有的标准和规范,多使用自定义的指令集和控制函数,驱动程序的接口定义差别明显,相互间很难兼容。

为此,在工程应用中,一般会对厂商提供的非标准驱动程序进行二次封装,形成专门针对特定仪器的封装库^[9-10]。这种开发方式在独立的软件项目中问题不明显,但是对于需要部署在多种测试设备上,用于解决多型号产品测试问题,以及开发、使用、维护周期较长的系列化软件产品而言,则带来了许多的问题。例如,缺乏统一的硬件控制方案,各测试程序的随意性较强;软件的移植性较差,当硬件升级和改造后,已有的测试程序无法使用;软件的复用性不高,重复开发的情况经常出现。

本文从实际工程实践出发,参考已有的研究成果^[3,11-12],提出了一种面向功能的测试设备驱动器的设计和开发方法。测试设备驱动器是沟通底层硬件驱动程序与上层测试软件的桥梁,为软件开发提供了统一的软硬件接口,并规范了驱动器内部的实现方式。通过对特定类别测试设备所需实现的硬件功能需求的抽象,最大程度地凝练共性内容,并将它们整合在一起,形成脱离硬件环境的测试功能项目集合。上层测试程序针对这些完全虚拟化的功能项目进行开发,不必了解底层各种异构的测试仪器、板卡及其驱动程序所带来的差异,可根据

不同的业务需求,方便、灵活地实现对硬件资源的间接控制。同时,将不同设备间的差异性内容封装在驱动器组件的内部,随测试设备一起发布,共同构成了上层测试软件开发和运行的基础资源。

1 测试设备驱动器的设计原理

以用于实现电子产品功能和性能检测的测试设备为例,该类设备的内部集成了各种测试仪器和板卡,是测试功能的实现载体。这些仪器和板卡的种类、形式多样,包括了 PCI、CPCI、PXI 等形式的模块化板卡、外置的组合仪器、标准的台式仪器等;实现的功能也各不相同,其中很多仪器都是针对特定需求而单独开发的。

在设计测试设备驱动器时,目前经常使用的是以测试仪器为中心的设计方式,即直接针对各种测试仪器和板卡的驱动程序进行测试程序的开发。此时,上层应用软件直接控制底层的硬件资源,当测试程序在不同设备之间移植或测试仪器改变时,测试程序需要进行大量改动,可移植性和重用性较差^[14],显然无法满足定制化设备的软件开发要求。

为此,本文采用了针对功能的驱动器设计方式,其关注点不是各种异构的仪器或板卡,而是更高层的设备层面。通过对测试设备硬件功能的分析,抽象出该类设备的各种测试功能。测试设备驱动器针对的是该类设备需要实现的系统级测试功能,而不再是各种异构的仪器或板卡。这种以功能为中心的设计方式,有效屏蔽了底层硬件的差异性内容,使得驱动器具有更好的稳定性和扩展性。

2 测试设备驱动器的接口设计

驱动器接口的设计内容包括:在驱动器的内部建立功能项目驱动接口,用于表示各种测试功能;在驱动器的外部为上层应用提供统一的对外操作接口,以及该驱动器自身应包含的、辅助性的功能性接口。

2.1 功能项目驱动接口的定义

功能项目驱动接口描述了某个功能项目的具体细节,代表了一种完全抽象的系统级功能,与硬件(包括测试仪器、板卡、测试设备等)无关,因而更加具有普遍性。该接口的定义模型如图 1 所示。

所有的项目驱动接口都继承自基础接口 IBasedSpecificDriver,由后者反映各驱动接口具有的共性内容,在定义后不会被修改,对外提供了驱动接口的一致性描述信息。

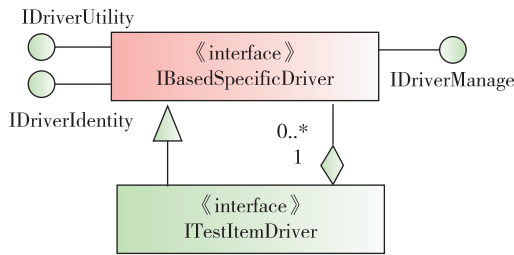


图1 测试功能项目驱动接口的定义

Fig. 1 Definition of test functional item driver interface

1) 基础接口的定义

基础接口 `IBasedSpecificDriver` 不涉及任何具体的测试功能,体现了公共性,其内容包括:

`IDriverUtility`: 提供一组最基本的操作方法,如使能、复位、错误查询等;

`IDriverIdentity`: 对外提供了该测试功能项的识别信息,其中最重要的内容是一个用以标明每个实现组件功能类别的唯一标识符;

`IDriverManage`: 完成各种功能项目驱动接口的创建、集成和检索,在驱动器中提供了一个独立的驱动接口列表,集中管理该驱动器含有的各种测试功能项目。

2) 功能项目驱动接口的定义

每个功能项目驱动接口 (`ITestItemDriver`) 表示了一个独立的测试功能项目,众多驱动接口组织在一起,构成了一个虚拟化的测试设备,具备了该类测试设备的各种硬件测试功能。

2.2 测试设备驱动器的对外接口定义

测试设备驱动器的对外接口 (`IAteDriver`) 是对外提供的唯一端口,屏蔽了驱动器内部的具体实现,该接口的定义模型如图2所示。

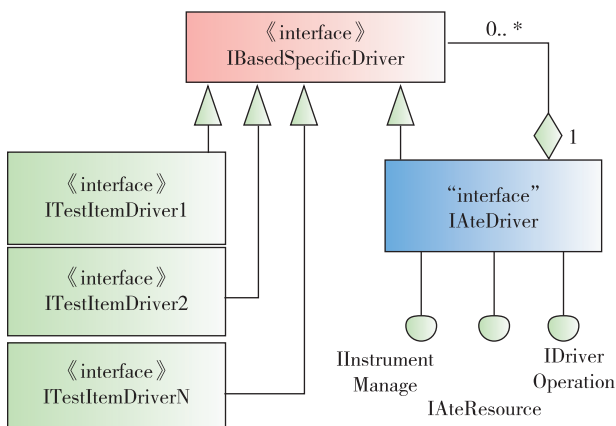


图2 测试设备驱动器的接口模型

Fig. 2 Interface model of test equipment driver

驱动器接口采用了组合模式 (Composite Pattern), 内部包含了 N 个测试功能项目接口。从物理意义上分析,该模型表示自动化测试设备具有了 N 种不同的独立测试功能,并可根据设备发展情况随时进行扩展,但不会影响测试设备的对外描述。

该驱动器的对外接口可以应用在不同类别、不同型号的测试软件开发中,换言之,对于上层应用软件而言,可忽略因硬件设备和仪器的差异性所带来的影响,使得应用开发人员可以将工作重心集中于业务逻辑本身,更加方便、灵活地应对上层软件的需求和变化。

2.3 测试设备驱动器的接口设计

测试设备驱动器的对外接口继承了 `IBasedSpecificDriver`, 自身已经具备了功能项目驱动接口定义的各种操作。除此之外,还提供了3个额外的辅助接口,表示了驱动器需要具有的特殊操作。

1) 测试仪器和板卡的管理

`IInstrumentManage` 接口负责管理该测试设备中的各种测试仪器和板卡,不涉及具体的测试功能。

2) 设备资源的管理

在驱动器设计时,采用数据模型的方式,描述了测试设备和被测产品的物理特性^[13]。资源管理接口 `IAteResource` 的核心任务是提供被测产品的输入输出端口和测试仪器的资源端口之间的映射关系。

3) 设备驱动器基本操作

`IDriverOperation` 接口中定义了测试设备驱动器自身独有的基本操作,用于驱动器的初始化。

3 测试设备驱动器的实现

测试设备驱动器的对外接口为上层应用软件提供了一个统一的操作端口,但测试软件的执行必须建立在实际的测试设备硬件上,每个驱动器组件代表了某个具体的测试设备,描述了该设备如何实现各种规定的功能项目。

驱动器组件直接实现了测试设备驱动器接口 `IAteDriver`, 具体实现过程可以划分为三部分: 1) 驱动器模型中定义的、具有公用性质的基础功能的实现; 2) 该测试设备应具备的各种测试功能项目的实现; 3) 驱动器与上层应用软件间的动态交互过程的实现。

3.1 基础功能的实现

通过对测试设备驱动器接口模型的分析,驱动器需要实现的基础功能包括以下内容。

- 项目驱动接口的创建

在实现 IBasedSpecificDriver 接口时,核心的功能是实现各种功能项目驱动接口的实例化,并将其组装在驱动接口列表中。该功能由 IDriverManage 接口的 CreateDrivers 函数实现,其伪代码如下:

```

Function: CreateDrivers
Input: dict: 驱动接口列表
Output: count: size of dict
1: clear dict;
2: 建立串口测试项目驱动接口的实例 D[0]
3: add D[0] to dict;
4: 建立 CAN 测试项目驱动接口的实例 D[1]
5: add D[1] to dict;
6: /* 参考 D[0], D[1], 建立其他项目驱动实例,
并增加到列表中 */
7: for each IBasedSpecificDriver D[i] in dict do
8: if D[i]. DriverManager ≠ null then
9: D[i]. DriverManager. CreateDrivers();
10: end if
11: end for
12: return dict. Count;

```

在实现 IAteDriver 接口时,主要的开发内容包括:

- 测试仪器和板卡的初始化

这些硬件初始化完毕后,分门别类地保存在驱动器中,作为后续各种功能项目的开发基础。

- 设备资源管理功能的实现

该功能的主要目的是实现被测产品的测试需求、测试设备提供的测试资源、测试仪器的测试能力三者之间的匹配,从而获得被测产品和测试设备之间的测试路径。本文参考现有研究成果^[14-15],并结合工程实践,建立了如图 3 所示测试设备的连接关系模型,以描述自动测试设备内外的电气特性。

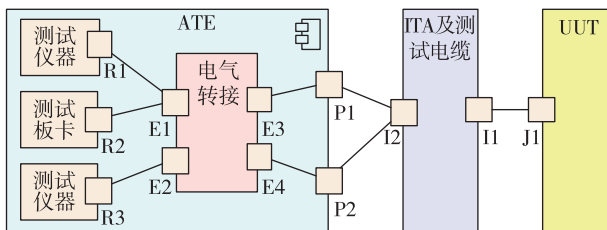


图 3 测试设备的连接关系模型

Fig. 3 Connection relation model of test equipment driver

3.2 测试功能项目的开发

对于驱动接口列表中的每项内容,都需要在驱动器组件开发时一一予以实现。每个功能项目最终都要落实到硬件仪器上,因而驱动器组件是和自动测试设备紧密耦合在一起的。在驱动器组件内部,通过调用各种异构的驱动程序,实现该测试设备的设计要求,从而将对外提供的各种虚拟测试项目转换为真实的测试功能。

测试功能项目的开发步骤如下:

Step1: 建立与功能项目驱动接口对应的实现类。

功能项目与设备驱动器是局部与整体的关系,功能项目实现类一般以内部嵌套类的形式存在,并由驱动器组件进行实例化。

Step2: 驱动基础接口的开发。

完成基础接口中定义的各项内容,其中尤其注意的是测试功能项的识别信息。

Step3: 功能项目的具体实现。

- 测试路径的查询,以获得完整的测试路径。

以图 3 为例,从右至左,获得从产品端口 J1 映射到测试板卡 R2 通道的测试路径,并以该板卡为真正的硬件执行单元,利用通道 R2 进行信号的发送和采集。

- 仪器的操作

对于非标仪器,使用硬件厂商提供的 API 函数,或者经过二次开发而形成的封装库;对于标准仪器,可直接调用 IVI 类驱动或专用驱动组件。在驱动器组件开发和调试完毕后,除非硬件发生变化,否则不会修改该组件。

- 功能的实现

功能项目面向的是测试设备的系统级功能,不仅仅是对仪器的简单操作。以模拟量采集为例,当使用数字万用表时,测量数值已经由仪器处理,可以直接读取;但使用 PCI 板卡时,需要对高速采集的多组连续数值进行必要的信号处理(如平滑、去除野点等),才可以得到比较准确的数据。

3.3 交互过程

图 4 以序列图(Sequence Diagram)的形式,描述了在典型应用场景下,应用软件与测试设备驱动器之间的交互过程,阐述了在测试软件运行过程中,各方的执行顺序及所起到的作用。

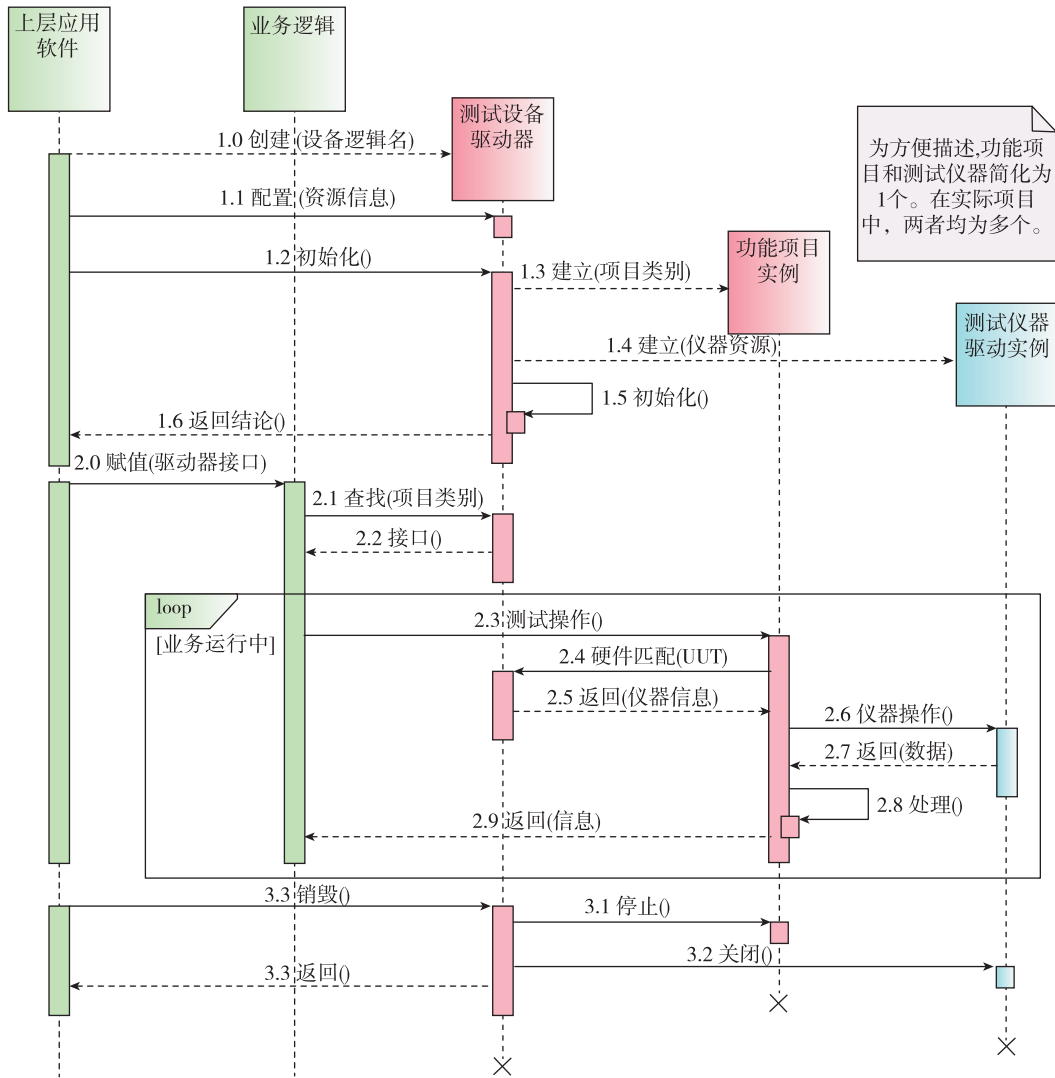


图 4 测试设备驱动器的交互过程

Fig. 4 Interaction process of test equipment driver

从图 4 可以明显看出,在上层应用软件和测试仪器驱动程序之间增加了测试设备驱动器(功能项目实例是驱动器的一部分,为说明方便,特将其单独列出),进而隔断了两者的直接关联。

在序列图中,消息 1.0~1.6 描述了驱动器组件的创建过程,驱动器由上层应用软件建立,并配置各种资源信息;各功能项目的建立、硬件仪器的初始化等由驱动器自身实现,与外部无关。消息 2.0~2.9 描述了测试业务的执行过程,业务逻辑只能通过应用软件赋予的驱动器接口进行开发;在业务逻辑内部,查找到需要的测试功能项目驱动接口,并通过后者,根据匹配的仪器信息,实现对某个测试仪器的操作。消息 3.0~3.3 描述了驱动器的销毁过程,该过程同样由上层应用软件发起。

4 应用实例

该测试设备驱动器的设计和开发方法已经在多个导航计算机测试软件开发项目中得到了应用。导航计算机测试设备是一种典型的、用于电子产品测试的系列化产品,内部集成了多种 PCI、CPCI 等板卡和其他专用仪器,其驱动程序大部分都是由生产厂商自己定义的。该类软件产品的整体结构示意图如图 5 所示。

针对每个自动测试设备(ATE1, ATE2, ATE3),在设备驱动层中都提供了一个虚拟的设备驱动器组件(Driver1, Driver2, Driver3),以屏蔽底层硬件驱动和硬件电气设计的差异,并在内部集成了该设备所具有的各种功能(I1, I2, …, Im)。设备驱动器组件反映了各种测试设备具有的硬件功能,作为

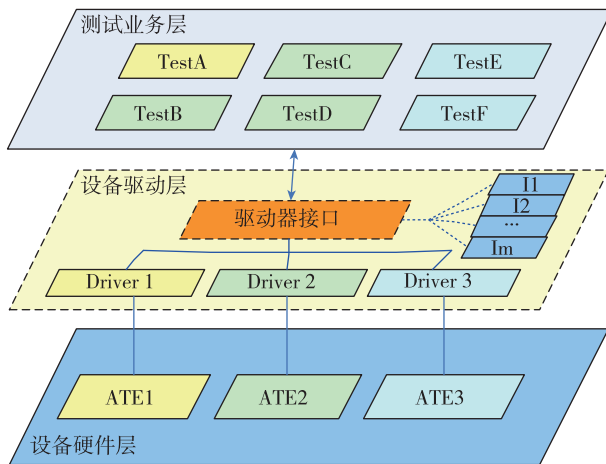


图 5 驱动器的应用实例示意图

Fig. 5 Application example of driver

上层应用程序的运行基础,在开发完后可全局复用。

测试业务层中采用了增量开发方法,可根据不同被测产品的特定测试要求而扩展新的业务组件(TestA, TestB, …… , TestF)。这些业务组件只能通过测试设备驱动器接口完成对硬件层的功能操作。在该类测试软件中,应用软件开发人员只需了解测试设备具有的测试功能,从而将主要工作集中在业务逻辑部分,提高了开发效率和质量。

5 结论

本文针对测试软件开发中存在的非标准驱动程序问题,提出了一种面向功能的驱动器设计与开发方法。实际项目的应用表明,该方法具有以下特点:

1)测试设备驱动器位于业务层和设备硬件层之间,有效降低了各测试程序与设备硬件层之间的直接耦合性,便于测试程序在多个测试设备上重用和移植;

2)测试设备驱动器对外提供了统一的规范化接口,提高了测试软件的标准化水平,并具有良好的延续性,这对于周期较长的系列化软件产品尤其重要;

3)采用上层业务程序与测试仪器驱动程序分离的设计方法,促进软硬件开发的分工与合作,提高了专业化水平,并将对硬件的全部操作集中在驱动器组件中,避免了重复开发,提高了开发质量。

通过测试设备驱动器,为各种系列化软件产品的开发提供了统一的硬件控制接口定义和规范化的实现方案,减少了非标准驱动程序的差异性对软

件项目的影响,从而提高了测试程序的移植性和开发质量,满足工程应用中系列化测试软件的设计与开发的需要。

参考文献

- [1] Lopes T P. Leveraging IVI for instrument wrapper development[C]// Proceedings of 2012 IEEE Autotestcon. IEEE, 2012: 144-146.
- [2] 许爱强, 文天柱, 李文海. 基于 IVI 技术的仪器驱动程序开发[J]. 现代电子技术, 2011, 34(2): 173-175.
Xu Aiqiang, Wen Tianzhu, Li Wenhai. Development of instrument drive program based on technology of IVI[J]. Modern Electronics Technology, 2011, 34(2): 173-175(in Chinese).
- [3] 赵鹏鹏, 崔少辉, 张文泉. 测试软件平台的仪器可互换性机制研究[J]. 中国测试, 2012, 38(2): 118-120.
Zhao Pengpeng, Cui Shaohui, Zhang Wenquan. Research of instrument interchangeability in test software platform [J]. China Measurement and Test, 2012, 38(2): 118-120(in Chinese).
- [4] 王国华, 孟晓风, 赵瑞贤. 一种基于开放式系统的仪器互换性实现[J]. 中国测试技术, 2006, 32(4): 11-14.
Wang Guohua, Meng Xiaofeng, Zhao Ruixian. An implementation of instrument interchangeability based on an open architecture [J]. China Measurement Technology, 2006, 32(4): 11-14(in Chinese).
- [5] 秦凡, 韦高. 基于 VISA 库及 SCPI 命令的仪器程控测量[J]. 现代电子技术, 2011, 34(11): 126-131.
Qin Fan, Wei Gao. Instrument program-controlled measurement based on VISA and SPCl [J]. Modern Electronics Technology, 2011, 34(11): 126-131(in Chinese).
- [6] Cui X, Peng Y, Peng X. A universal test system framework and its application on satellite test[C]// Proceedings of International Conference on Compute and Data Analysis (ICCCA'17), 2017: 283-288.
- [7] Jia H, Shi J. Development of interchangeable virtual instrument driver for cannon-chamber test instrument [C]// Proceedings of International Conference on Computer Science and Information Processing. IEEE, 2012: 299-302.
- [8] 陈宝华, 席泽敏, 王俊茂, 等. 自动测试系统互换性技术研究[J]. 电子测量技术, 2010, 33(7): 1-3.
Chen Baohua, Xi Zemin, Wang Junmao, et al. Study

- for interchangeability technology of auto test system [J]. *Electronic Measurement Technology*, 2010, 33(7): 1-3(in Chinese).
- [9] 朱坤, 文天柱, 许爱强, 等. 面向信号的智能激励信号控件设计[J]. *测控技术*, 2015, 34(6): 100-103.
Zhu Kun, Wen Tianzhu, Xu Aiqiang, et al. Design of signal-oriented intelligent source signal control[J]. *Measurement Control Technology*, 2015, 34(6): 100-103(in Chinese).
- [10] 陈鑫友, 郭海帆, 陈建华, 等. 可互换仪器程控命令封装库设计与实现[J]. *太赫兹科学与电子信息学报*, 2017, 15(1): 110-114.
Chen Xinyou, Guo Haifan, Chen Jianhua, et al. Design and implementation of interchangeable instrument driver[J]. *Journal of Terahertz Science and Electronic Information Technology*, 2017, 15(1): 110-114(in Chinese).
- [11] 刘福军, 汤宫民, 孙香冰, 等. 面向服务的自动测试系统仪器驱动器设计与实现[J]. *仪表技术*, 2016(7): 5-8.
Liu Fujun, Tang Gongmin, Sun Xiangbing, et al. Design and realization on instrument driver of service-oriented ATS[J]. *Instrumentation Technology*, 2016(7): 5-8(in Chinese).
- [12] 安幼林, 杨锁昌, 黄考利. 基于功能接口的网络化ATS软件平台研究[J]. *计算机测量与控制*, 2007, 15(6): 114-116.
An Youlin, Yang Suochang, Huang Kaoli. Research of networked ATS software platform based on functional interface[J]. *Computer Measurement & Control*, 2007, 15(6): 114-116(in Chinese).
- [13] 孟翰, 吴际, 胡京徽, 等. 硬件系统自动化测试的多视角建模及案例研究[J]. *计算机科学*, 2018, 45(9): 75-80.
Meng Han, Wu Ji, Hu Jingwei, et al. Modeling in multiple views and industrial case study of automatic test for hardware system [J]. *Computer Science*, 2018, 45(9): 75-80(in Chinese).
- [14] 张文, 杨京礼. 一种面向信号的自动测试系统资源分配方法设计[J]. *现代电子技术*, 2013, 36(19): 116-119.
Zhang Wen, Yang Jingli. Design of a method of resource allocation in the signal-oriented ATS[J]. *Modern Electronics Technology*, 2013, 36(19): 116-119(in Chinese).
- [15] 张桂英, 侯倩, 范利花, 等. 基于ATML的可移植TPS开发技术研究[J]. *测控技术*, 2017, 36(1): 117-120.
Zhang Guiying, Hou Qian, Fan Lihua, et al. Research on developing technology for transplant test program set based on ATML[J]. *Measurement Control Technology*, 2017, 36(1): 117-120(in Chinese).

(编辑:孟彬)