

基于 TestStand 的自动测试系统软件设计

吴伟¹, 张宏军¹, 刘康², 朱迪¹, 吴捷¹

(1. 上海机电工程研究所, 上海 201109; 2. 同济大学 经济与管理学院, 上海 200092)

摘要: 为满足测试系统多功能测试状态的需求, 开发了一种基于 TestStand 的自动测试系统软件; 该软件使用 TestStand 引擎对复杂的自动测试流程和测试资源进行管理; 实际应用表明, 该软件能够简化测试流程, 有效提高了测试系统的测试效率, 减少了软件维护成本, 同时, 该系统软件具有良好的兼容性和灵活性。

关键词: TestStand; 自动测试系统软件; 测试序列

Design of ATE Software Based on TestStand

Wu Wei¹, Zhang Hongjun¹, Liu Kang², Zhu Di¹, Wu Jie¹

(1. Shanghai Electro-Mechanical Engineering Institute, Shanghai 201109, China;

2. College of Economic Management, Tongji University, Shanghai 200092, China)

Abstract: In order to meet the performance test requirements in multimode, a new ATE software based on TestStand was designed. This software can manage the complex test process and resource via the TestStand engine. The practical application shows the ATS software can facilitate test development process, highly increases test development efficiency and exhibits well compatibility and agility.

Keywords: TestStand; ATS software; testing sequence

0 前言

在编写传统自动测试系统测试软件时, 主要以测试流程为依据, 自上而下一步一步完成, 所有的测试参数、程控指令、测试结果等都内置于测试软件中, 任何一个改变时, 都需要重新编译与链接, 软件的开发时间长, 系统的可靠性低。而当面对比较复杂的测试项目时, 测试软件会迅速变得十分庞大, 软件的维护也越来越困难, 当这种方式编写的软件进行功能扩展或者应用于另外一个系统时, 软件几乎需要重新编写。为此, 在 TestStand 测试管理环境下, 本文设计了一种新型测试软件, 采用虚拟仪器技术开发的自动测试系统可以配置不同的测试序列以满足不同测试状态的需求, 大大提高了自动测试系统的测试效率, 降低了软件的维护使用成本。

1 系统结构及原理

某装备自动测试系统如图 1 所示, 系统分为测控组合机箱、电源机箱、射频微波机箱、专用设备。被测产品通过专用接口适配器及通用接口实现与测控组合机箱的信号转接和互连。通用接口采用 VPC90 标准, 按照规范定义了所有常用测试资源及基本开关资源, 包含 25 个信号接口模块, 可根据实际系统的测试需求选用测控组合机箱内部的测试资源, 专用资源根据实际系统的测试需求单独配置。

测控组合机箱为整个自动测试系统的控制核心, 自动测试系统软件安装在零槽控制器内, 每一个测试项目被进一步细化、分解成若干个测试步骤, 这些测试步骤功能相互独立, 可以是单个数据判别、信号测量、激励动作或命令通讯。测试步骤使用测试编程语言 LabVIEW 2011 编写, 并按照规定的函数定义编译为 DLL 文件中的一个库函数。

收稿日期: 2014-03-14; 修回日期: 2014-04-15。

作者简介: 吴伟(1980-), 男, 研究生, 高工, 主要从事测试系统总体设计、故障诊断等方向的研究。

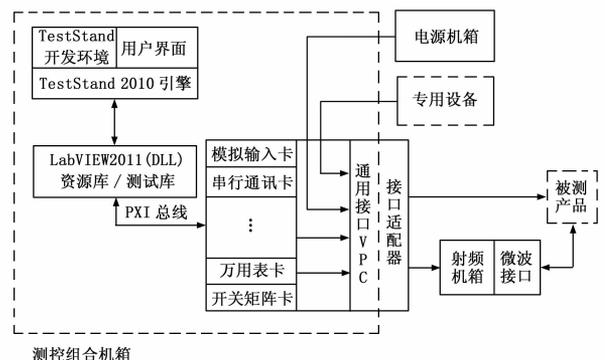


图 1 某装备自动测试系统框图

用户通过用户界面配置这些测试步骤形成一个个测试项目, TestStand 2010 引擎加载 DLL 模块适配器函数使 TestStand 在执行测试序列的同时协调控制电源机箱、射频机箱和专用设备的各种资源实现被测产品的功能性能测试。

2 软件设计与实现

本文涉及的被测产品是一个复杂的黑盒系统, 测试参数众多, 种类繁多, 在对不同特性参数测试时, 需要施加不同的激励信号和工作命令, 同时, 该产品有多种测试状态, 相同的测试项目在不同的工作模式下, 测试方法和测试流程也有很大的差异。因此测试序列复杂繁多, 测试流程管理就非常重要。

TestStand 是一个对自动测试系统测试任务进行管理的商业软件, 由测试执行引擎执行由多个测试步骤形成的测试序列来实现测试管理^[1]。TestStand 为测试开发和执行提供了一个核心的框架, 省去了通常的测试执行的开发任务, 减少开发工作量、加强测试系统兼容性。将各个测试项目做成单独可以调用的测试序列, 工作模式配置、不同仪器的调用、读取等都作为公共服务测试程序集成到 TestStand 的环境中, 用户只需要通过测试序列对话框选择需要的测试项目, 就可以开始不同工

作模式产品的测试。

基于 TestStand 建立测试系统的步骤如下^[2-4]：

(1) 定制和配置 TestStand：用户可以根据测试系统的实际情况对 TestStand 环境变量进行配置。通过 TestStand ActiveX API，用户可以自定义测试变量，测试项目类型，可以定义测试报告生成格式和路径，可以定义测试序列执行模式。此外，用户还可以通过修改相应目录下 TestStand 配置文件，完成对 TestStand 的语言汉化。

(2) 建立测试序列：应用 TestStand 的序列编辑器来建立测试序列，以文件扩展名 .seq 为标识。一个测试序列文件由若干个测试步骤 (Step) 组成，测试步骤可以是 TestStand 内置的类型，也可以是从 DLL 文件中导出的函数或者 LabVIEW 开发的 VI 在被执行时调用相应的测试代码模块。

NI TestStand 2010 版本提供 7 大类 52 种测试步骤。本文的测试系统软件主要应用了测试 (Test) 类型里的成功失败型测试 (Pass/Fail Test)、内置类型里的动作 (Action) 和调用序列 (SequenceCall) 3 种测试步骤。

在 TestStand 2010 中，首先建立 Main Sequence 工程文件，然后根据被测产品的测试要求进行测试项目分类，在 Main Sequence 目录下应用 SequenceCall 插入需要各个测试项，每个测试项又包含很多测试小项，按照层次，从大项到小项逐级建立测试序列，一个测试序列就是一种被试产品的测试状态。

(3) 测试代码模块：自动测试系统软件的测试代码模块在 LabVIEW 2011 编程环境下开发，首先建立一个工程项目文件，用来纪录保存项目中的项目文件引用、配置信息、部署信息、程序生成信息等等。然后根据测试步骤 (Step) 编写对应的测试代码模块，调试完毕后通过工程项目文件中的程序生成规范将测试代码模块生成 DLL 的不同函数。

测试代码模块由测试库和资源库组成。测试库是所有测试小项函数的集合，包含在 MyDll 文件中，将测试结果与技术指标进行比较，给出合格或者不合格的结论。资源库是自动测试系统中可以用软件控制的所有公用硬件资源函数的集合，包括开关矩阵的通/断、数字万用表的测量、AD 模块的采集等等。

图 2 是一个典型测试序列的代码框图，由于许多测试项目的测试步骤和测试过程相同，只是激励的选择或信号参数选择的不同，因此在资源库函数编写时，需要对原有的测试代码进行数据封装和功能抽象，提供测试序列中的测试步骤 (Action) 调用。资源库的函数在编写时都预留了对应的命令接口，用户只需要更改相应的命令参数，就能进行功能的切换而不需要重新编译。

测试库则按照测试小项的测试内容和技术指标完成数据的判读任务，每一个测试小项对应一个函数，提供测试序列中的测试步骤 (Pass/Fail Test) 调用。DLL 函数内部的数据交互

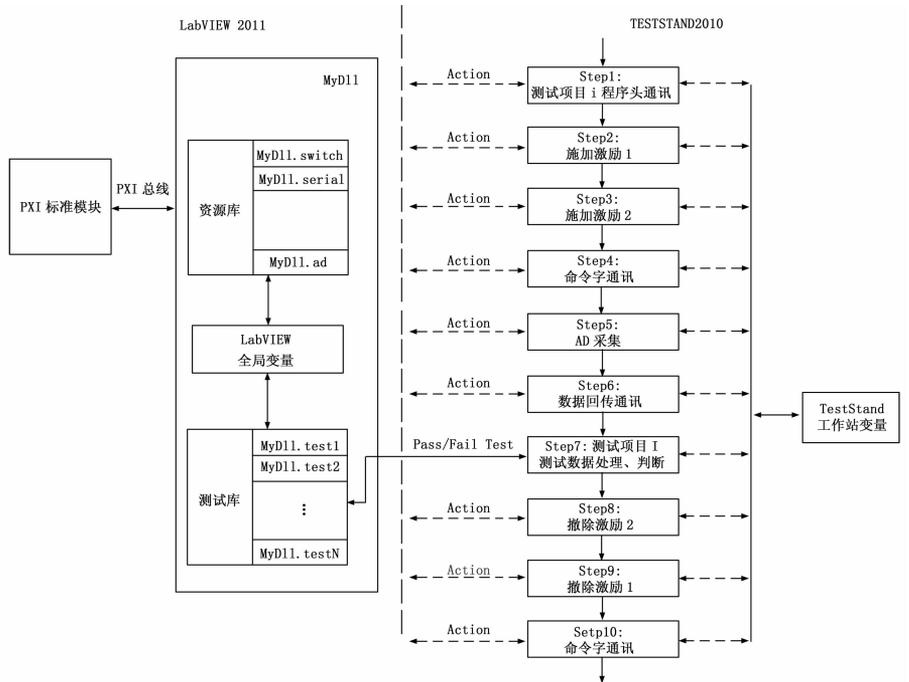


图 2 某测试序列测试代码框图

通过 LabVIEW 全局变量实现，测试步骤之间的数据交互通过 TestStand 工作站变量实现。

(4) 用户界面：TestStand 集成了 4 个通用的测试界面并提供了这些界面的源代码，基于本系统的其他使用要求，本系统采用第三方软件 (VS6.0)，编程实现自动测试软件的测试主界面，并根据测试的具体要求，完成自动测试软件系统的其他辅助功能，例如数据报表、数据回放等。

(5) 执行流程：自动测试系统软件执行步骤如图 3 所示。首先，运行软件，进行必要的用户登陆，进入测试主界面，点击测试序列配置菜单，进入测试序列配置界面，在测试步骤集中选择本次测试需要的测试步骤，并对每个测试步骤的输入输出参数进行设置，对于常用的测试内容可以保存为一个测试序列文件，简化配置过程。然后，输入必要的被试产品的测试信息，点击测试开始按键，启动自动测试流程，TestStand 引擎自动加载用户测试序列并完成对应的测试步骤。在此期间，测试库中的测试步骤会弹出运行界面，显示被试产品的各种测试信息，包括信号波形、合格指示、实测数据等等。最后，自动测试软件根据测试步骤执行情况自动生成测试报告，用户可通过数据回放功能再现测试过程中的相关信息。

3 实例应用

某装备自动测试系统原来只具有对被试产品进行整机综合测试，现在需要在原有硬件基础上增加振动试验和功能测试两种新型工作模式。在构建新的测试系统时，原有的资源库中的代码模块通过改变不同的输入参数可直接使用，同时增加了振动试验和功能测试对应的测试库函数，采用序列配置对话框将它们按照测试要求组合成两个新的测试序列 (振动测试 .seq 和功能测试 .seq)，原有的综合测试 .seq 保持不变。实践证明，采用 TestStand 为核心的自动测试系统能够快速完成软件功能的扩展，可以提供不同被试产品、不同测试状态的可扩展性、可剪裁性和可移植性。

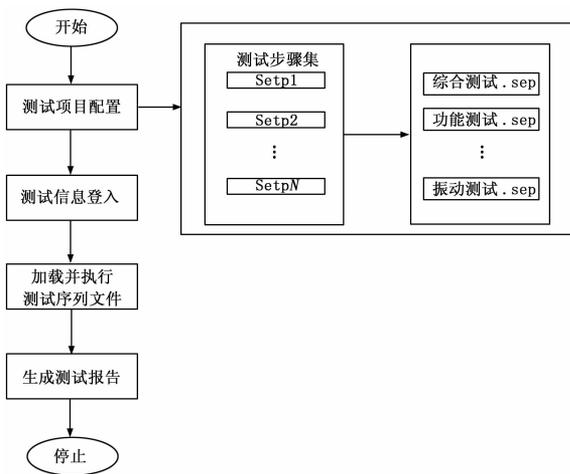


图 3 自动测试系统软件执行流程

4 结论

基于 TestStand 的自动测试系统软件采用 TestStand 构建自动测试系统软件的框架, 通过序列配置定制自动测试流程, 使用 LabVIEW 进行测试代码模块的开发, 操作简单, 使用方便, 有效提高了自动测试系统的测试效率, 减少了软件维护成本, 同时, 该系统软件具有良好的兼容性和灵活性。

参考文献:

[1] 赵 鑫, 等. 基于综合优先级的并行测试调度算法设计及实现 [J]. 计算机测量与控制, 2007, 15 (4): 423-425.
 [2] 芦 俊, 等. 基于 TestStand 的 DDS 特性参数自动测试管理 [J]. 半导体技术, 2009, 34 (10): 957-959.
 [3] 于建林, 等. 基于 TestStand 的测试系统及应用 [J]. 国外电子测量技术, 2004, 2: 6-9.
 [4] 王伟浩, 等. 基于 TestStand 的计量软件介绍 [J]. 计算机技术及应用, 2009, 29 (6): 50-53.

(上接第 2311 页)

```
void Task_IMU(int arg1,int arg2, int arg3)
{
    //do something
}
...
if(ERROR == tsk_create(&.TskID[0], "T1", 5, 0, TskStack, sizeof(TskStack), (TSKFUNCPTR)Task_IMU, 0, 0, 0))
    Task_IMU();
```

3.4.3.2 中断管理

在中断服务程序中先关闭该中断, 再在对应的任务中处理完中断事件后开启, 这样保证在一个处理周期中, 若有该中断发生, 不会响应, 避免频繁响应中断造成系统无法处理其他事件, 也无法发送遥测信息。

3.5 试验结果

在某型弹载计算机平台上应用 OS 微内核设计的系统软件, 进行系统级半实物仿真, 弹载计算机、惯测、舵机等弹上系统闭环工作, 通过了半实物仿真验证试验, 具体的测试数据如表 1。

表 1 基于 OS 的弹载计算机软件的性能数据

序号	项目	实测值
1	中断响应时间 < 100 μs	79 μs
2	任务上下文切换时间 < 200 μs	114 μs
3	操作系统 bin 文件 < 30 kB	28 kB
4	整个系统软件 < 128 k 双字	50 k 双字
5	上电软件初始化到用户程序 < 10 ms	6.177 5 ms
6	制导控制指令曲线相似度达到 90%	一致

从表 1 的序号 1~5 的数据上可以看出, 按弹载计算机应

用条件定制的 OS 微内核性能指标满足设计需求, 最终形成的弹上计算机系统软件执行时间、占用空间均满足应用指标要求。从表 1 的序号 6 的数据表明, 弹载计算机系统能运行正常, 运行曲线与数值仿真一致, 证明基于 OS 微内核的弹载计算机软件满足设计要求。

4 结论

通过对嵌入式操作系统和战术导弹弹载计算机系统的研究, 制定出符合弹载软件特性的 OS 微内核系统, 同时在此基础上设计了多任务架构的弹载计算机软件, 屏蔽底层驱动功能, 合理划分系统软件任务项, 充分验证基于嵌入式实时操作系统的战术导弹弹载计算机软件的可行性, 为后续导弹电子设备综合化的多任务软件系统设计奠定了技术基础。相比传统的弹载软件设计方法, 基于实时操作系统的软件设计对提高弹载计算机软件的重用性、降低软件开发的复杂性, 提升系统任务的处理能力具有重要的意义。

参考文献:

[1] 毛伟云. 嵌入式实时操作系统 μC/OS-II 在弹载计算机上的应用 [J]. 火箭与制导学报, 2007, 27 (4): 349-356.
 [2] 夏克寒, 许化龙. 弹载计算机中的实时操作系统研究 [J]. 微电子学与计算机, 2004, 21 (12): 97-100.
 [3] 邓 昀. 微内核结构嵌入式实时操作系统的研究与设计 [J]. 微电子学与计算机, 2012, 29 (10): 133-139.
 [4] 苏玉强, 刘 颖. 基于消息的嵌入式系统应用框架研究 [J]. 计算机测量与控制, 2011, 19 (5): 1142-1144.
 [5] SZOS 软件用户手册 V3.0 [Z]. 北京: 北京神州航天软件技术有限公司, 2010.
 [6] 解月江. 基于 SPARC 与 RTEMS 的飞行控制软件开发技术 [J]. 航天控制, 2009, 27 (4): 101-106.