

嵌入式系统测试环境体系结构的研究

任 冀¹, 樊晓光¹, 田 涛²

(1. 空军工程大学空军工程学院航空电子工程系, 西安 710038; 2. 空军装备软件测评中心, 北京 100076)

摘要: 针对嵌入式系统测试缺乏通用平台的难点问题, 提出了一种通用测试平台方案。该方案以 PXI 为底层设备总线, 通过 PXI 转换接口实现设备互连, 克服了不同设备之间接口的不一致性; 采用硬件在回路仿真的方法对被测单元进行测试, 满足了系统测试的实时性要求; 采用商业软件 Matlab/Simulink、Labwindows/cvi 及开源软件 TCL/Tk 实现体系构建, 以 TCP/IP 协议实现网络互联, 完成对各个分系统的测试前的数据准备和测试后的数据备份、分析及处理, 具有很好的可重用性、扩展性和通用性。此外, 综合使用 UML 和 CSR (条件 - 激励 - 响应) 对测试需求进行了规范化描述。

关键词: 嵌入式系统; 通用测试平台; 硬件在回路; Simulink; Labwindows/cvi

Research on Structure of Test Environment for Embedded System

REN Ji¹, FAN Xiao-guang¹, TIAN Tao²

(1. Department of Avionics and Electronic Engineering, Air Force Engineering Institute, University of Air Force Engineering, Xi'an 710038;

2. Airforce Software Testing and Evaluating Center, Beijing 100076)

【Abstract】 Since there is a lack of universal test platform for embedded system, a new solution is presented. It is based on the PCX low layer equipment bus with which equipments can be connected by adapters, so that differences between interfaces of equipments is prevented. The unit under test is tested with the method of HILS (hardware in the loop simulation) to fulfill realtime requirement. The commercial software, Matlab/Simulink, Labwindwos/cvi, and open source software TCL/Tk is introduced to realize the construction of the platform. The data preparation before the test and data backup, analysis and processing after the test on the connected systems are realized based on the TCP/IP network, so that good reusability, extensibility and universality are acquired. UML and CSR (condition-signal-response) are introduced to make the test requirement normalized.

【Key words】 embedded system; universal test platform; hardware in the loop; Simulink; Labwindows/cvi

随着航空航天通信与控制、工业过程自动控制、军事、医疗设备等领域对嵌入式设备的需求不断增加, 为了保证嵌入式设备的可靠性、正确性, 相应的测试理论、测试方法也需要不断改进^[1]。测试是对所建造的系统是否满足所要求标准进行的检验, 同时通过测试可以发现现有系统的缺陷, 从而可以指导系统改进。嵌入式系统作为一种软硬件结合的系统, 由于其自身的特点, 对此类系统的测试必须建立系统的方法学(包括测试理论、测试手段、测试环境搭建、测试工具链的建立与使用、测试结果的分析处理方法)^[2]。

1 当前测试系统的主要问题及解决方法

(1) 测试需求的规范化描述: 使用 UML 的机制描述测试需求, 使描述规范化^[3];

(2) 不同设备接口不统一: 采用 PXI (PCI eXtension for instrumentation) 作为底层总线, 通过不同的 PXI 转换接口实现不同接口之间的统一^[4];

(3) 由于某种原因, 外部设备无法实际构建: 采取 HILS (硬件在回路仿真) 的半实物仿真平台^[5], 可以使用的硬件设备可直接接入系统中, 而难以构建的设备可以采用 Matlab/Simulink 仿真的方法进行构建^[6];

(4) 准备好测试用例后, 测试过程的自动执行: 采用 TCL/Tk 脚本语言, 及其脚本解释器作为测试前的系统配置和测试过程的自动执行^[7];

(5) 测试过程中, 测试信息的捕获: 利用 Labwindows/cvi 根据相应的测试需求, 设计实现相应的虚拟仪器, 捕获所需要的测试信息^[8]。

2 具体解决方案及实施步骤

2.1 测试过程

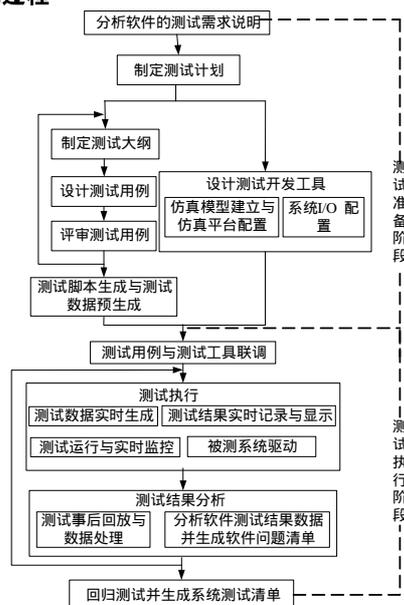


图1 嵌入式软件测试过程

作者简介: 任 冀(1982 -), 男, 硕士研究生, 主研方向: 分布式系统与智能检测; 樊晓光, 博士、教授; 田 涛, 博士、工程师
收稿日期: 2006-09-30 **E-mail:** jackstevenson@sohu.com

整个测试过程(如图 1)包括准备阶段和执行阶段。测试准备阶段主要是根据测试需求设计测试用例,同时为了使设计好的测试用例能够顺利执行,需要构建测试环境,这可以与测试用例设计并行进行。

测试执行阶段是在将设计好的测试用例作用于已经建立好的测试环境的基础上,进行实际的测试工作,主要包括激励数据的生成,测试过程的实时监控,测试结果的记录分析。

2.2 测试环境

测试环境(如图 2)是为了使 UUT(被测单元)能够正常运行而建立的外围系统,包括设备、总线、接口。首先,某些设备(如导弹、卫星)实验代价高昂,可采用 HIL 仿真的方法建立测试系统;其次,不同设备间的接口千差万别(如 MIL-1553B、ARINC-429、RS232、RS422、RS485、GPIB、SCXI、MXI、USB、FireWire 等),这对于建立统一的测试环境构成了巨大障碍。因此,可采用 PXI 作为测试总线,利用各种 PXI 转换接口实现不同设备接口的统一性,而对于一些非常规接口,只能进行手动配置。

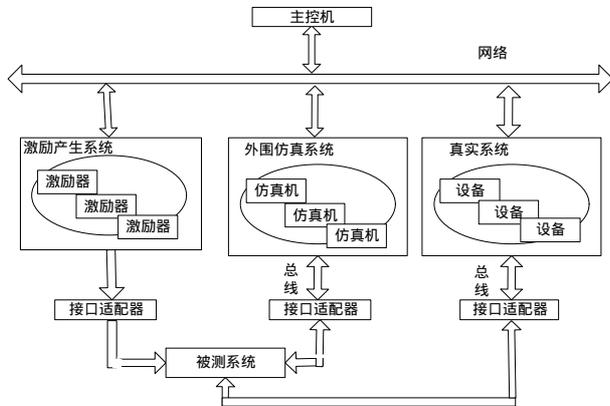


图 2 嵌入式软件仿真测试环境

2.3 测试任务类型

在整个测试过程中,测试任务具有不同的属性(如表 1),主要包括测试阶段和实时性;准备阶段,完成测试用例的设计和测试环境的配置;执行阶段,又可分为实时任务和非实时任务^[9],实时任务主要完成数据采集与监控,非实时任务主要完成对数据的后期分析与处理。这对应着测试系统软件结构中的实时模块与非实时模块并存,前者优先级高于后者。

表 1 测试任务类型

测试阶段	测试任务	
准备测试	测试脚本生成与测试数据预生成	
	仿真模型建立与仿真平台配置	
	系统 I/O 配置	
	测试用例与测试工具联调	
执行测试	实时任务	非实时任务
	测试数据实时生成	测试事后回放与数据处理
	测试运行与实时监控	分析软件测试结果数据并生成软件问题清单
	被测系统驱动	回归测试并生成系统测试清单
	测试结果实时记录与显示	

2.4 测试系统软件层次

为了建立测试环境,需要一系列的软硬件支持,其中测试系统软件结构(如图 3)为测试系统的相关工具的开发提供了便利,分为接口适配层、外部实体层、服务层和应用层。

服务层为应用层的开发提供相应的支持,而应用层开发出来的模块可以进一步扩充服务层的各个模块。各个模块之间具有相对的独立性。服务层的各种库支持来源于 ANSI C

的标准库、Matlab 的扩展库、Matlab/Simulink 的模型库、Labwindows/cvi 的数据采集库、数据处理库、仪器库、驱动库,同时为了实现某一特殊应用,可扩展已有的库。

外部实体层和接口适配层主要是为了建立被测单元的外部环境,使其能正常的运作。



图 3 测试系统软件层次

2.5 测试工具开发

测试工具开发(如图 4)是为了使测试工作能顺利进行,包括测试脚本开发、模型开发、仪器开发、驱动开发。

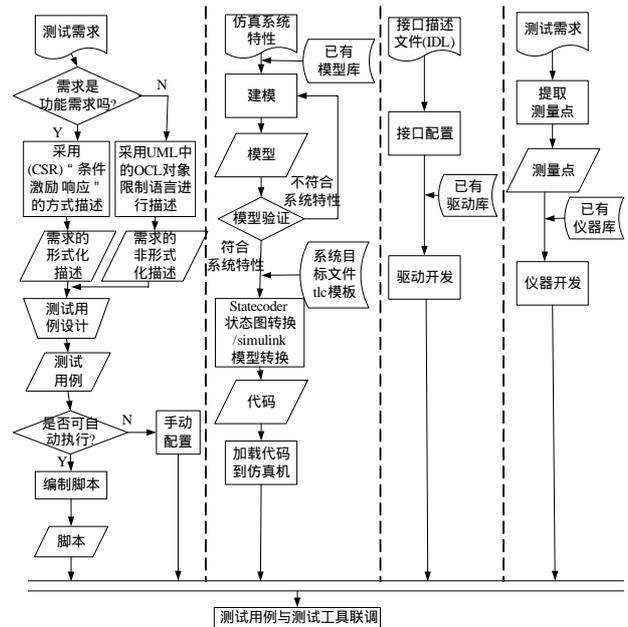


图 4 测试工具开发

(1)测试脚本开发。脚本开发采用 TCL/Tk,它具有以下几个方面的优点:1)可移植性:脚本可实现 Windows/Unix 的完全移植;2)流程控制:循环、选择等语法结构已完全实现;3)内建命令:如可用 after 命令完成延时,用 puts 进行打印;4)用例连接:使用 source 命令连接脚本,使多个用例实现测试自动化;5)命令扩展:TCL 提供命令扩展接口,可使用 C/C++编写新命令,编译、注册后在脚本中像内建命令一样使用;6)完全免费:TCL 解释器和图形库(统称为 TCL/Tk)是完全免费的资源,遵从 GPL。

脚本编制好后,具有相对的独立性,对于多个系统的测试,可以不用重新编制脚本即可运行,节省了人力、时间,提高了效率。

(2)模型开发。模型开发采用 Matlab/Simulink,它为模型的开发提供了一系列的工具,有利于连续系统、离散系统、混杂系统的建模,同时它为一些特殊的系统建模提供了相应

的工具箱，支持状态图模型转换；此外支持离线仿真，以便进行模型的正确性验证，之后可以将模型映射为目标系统的代码，修改后可直接加载到目标系统实现在线仿真。

(3)仪器开发。仪器开发采用 Labwindows/cvi，它具有以下几个方面的优点：1)NI（美国国家仪器）公司为 Labwindows/cvi 提供了丰富的硬件支持，包括信号调理设备、数据采集卡等，同时 Labwindows/cvi 自带了一系列设备的驱动；2)Labwindows/cvi 为数据的分析处理提供了丰富的工具，如 FFT、频谱分析等；3)丰富的仪器、仪表面板为数据的实时监控和回放提供可视化元素；4)在数据存储与备份方面，Labwindows/cvi 提供了文件和数据库方式。

这些特点极大地方便了数据的采集、分析和备份工作。

(4)驱动开发。由于 NI 公司为大多数设备提供了驱动程序，因此只需对自行开发的设备编制驱动。

2.6 测试执行过程

在测试准备工作(包括测试用例设计、测试工具开发、系统接口配置)完成后，将进入测试执行阶段(如图 5)。

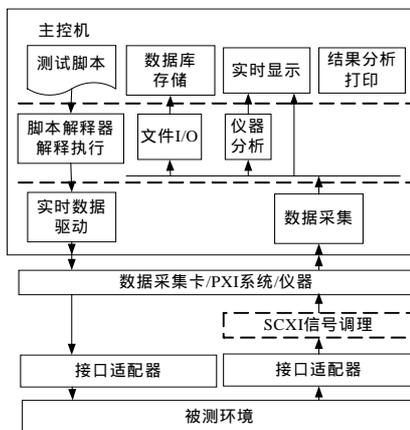


图 5 测试执行过程

该阶段主要完成将测试用例或编制好的测试脚本作用于测试环境，同时通过数据采集和监控得到实际结果，然后将得到的结果与预期结果进行对比，或者对数据进行处理，以便分析系统的特性。

(1)测试环境的实时性。测试过程中一个很重要的问题是实时性的问题。嵌入式系统一般为实时性系统，因此建立的测试环境也应具有实时的特性，同时必须是闭环的、非侵入式的。为了满足实时的要求，可以将 Matlab/simulink 开发的模型转换为实时操作系统(如 VxWorks)中可执行的程序，或其它硬件设备中(如 DSP)中可实时执行的程序。

(2)故障注入。通过修改模型的参数设置故障，可以测试被测单元在外部设备故障情况下运行的健壮性。

(3)数据采集。在数据采集过程中，为了实现数据采集的完整性，可以先将采集到的数据存储到文件，在测试完后再导入到数据库。

3 结束语

本文提出了一种嵌入式系统测试环境的构建方法和框架，由 Matlab/Simulink 建立外部设备模型，Labwindows/cvi 构建的虚拟仪器设备，以及 TCL/Tk 编写的脚本具有相对的对立性，同时可以在实践过程中不断地积累各种库，包括模型库、仪器库、驱动库等，从而整个系统具有很强的重用性、扩展性和通用性。

参考文献

- 1 陆民燕, 韩峰岩. 装备软件质量和可靠性管理[M]. 北京: 国防工业出版社, 2006.
- 2 章亮, 刘斌, 陆民燕. 嵌入式软件测试开发环境的框架设计[D]. 北京: 北京航空航天大学, 2005.
- 3 Pilone D, Pitman N. UML 2.0 in a Nutshell[M]. [S.l.]: O'Reilly, 2005.
- 4 PXI 体系结构[EB/OL]. (2006-05). <http://www.Friendess.com>.
- 5 徐德华. 实时半实物仿真系统关键技术的研究[D]. 哈尔滨: 哈尔滨工程大学, 2005.
- 6 姚俊, 马松辉. Simulink 建模与仿真[M]. 西安: 西安电子科技大学出版社, 2003.
- 7 Ousterhout J K. TCL and the TK Toolkit[M]. [S.l.]: Addison-Wesley Publishing Company, Inc., 1993.
- 8 李海, 武小栋. LabWindows/CVI 在半实物分布式仿真系统中的应用[J]. 电子技术应用, 2005, 31(4): 13-16.
- 9 谈琳, 罗永红. 实时软件的仿真测试平台的研究[J]. 计算机仿真, 2005, 22(1): 247-250.

(上接第 232 页)

RDR 是一种非常朴素而实用的方法，其应用前景是非常广阔的，但这种方法也存在一些局限性，例如，设计知识冗余造成搜索速度慢、多专家知识源之间冲突检测等问题。为了解决这些问题，应进一步研究相应有效的解决策略，为企业提供高效的设计知识管理。

参考文献

- 1 邱晖, 孙正顺. 知识管理系统的构建和策略[J]. 计算机工程与应用, 2001, 37(1): 52-54.
- 2 孔繁胜. 知识库系统原理[M]. 杭州: 浙江大学出版社, 2000.
- 3 Andy D, Agogino A M. Managing Design Information in Enterprise-wide CAD Using Smart Drawings[J]. Computer-Aided Design, 1998, 30(6): 425-436.
- 4 Ha S, Pahang G, Chang M, et al. Managing Design Knowledge: Active

Document System[J]. Annuals of the CIRP, 1999, 48(1): 89-92.

- 5 Compton P, Horn R, Quinlan R, et al. Maintaining an Expert System in J R Quinlan[C]//Proc. of the 4th Conf. on Applications of Expert Systems. 1988.
- 6 Wada T, Horiuchi T, Motoda H. Adaptive Ripple Down Rules Method Based on Minimum Description Length Principle[C]//Proc. of IEEE International Conference on Data Mining, 2002.
- 7 Debbie R, Compton P. Uncovering the Conceptual Models in RDR KBS[C]//Proc. of International Conference on Conceptual Structures. 1997.
- 8 Khan A S, Hoffmann A. Acquiring Adaptation Knowledge for CBR with MIKAS[C]//Proc. of Artificial Intelligence, Berlin, Germany. 2001.