

# 导弹逼近告警器及无源干扰设备测试软件设计

张秋霞

(黄河科技学院 现代教育技术中心, 郑州 450063)

**摘要:** 为解决导弹逼近告警器及无源干扰设备测试成本高、维护保障难的问题, 研制了一套导弹逼近告警器及无源干扰设备自动检测系统; 设计了系统测试软件, 按照模块化设计方式, 针对设备检测需求, 开发不同的软件测试模块并集成; 该系统集成度高, 配置灵活, 操作智能化, 功能完整, 测试项目全, 测试和排故障效率高; 经试用表明, 该检测系统具有良好的操作性和维护性, 可以为导弹逼近告警器及无源干扰设备检测维修提供技术支持, 有效节省测试成本和维护保障费用。

**关键词:** 导弹逼近告警器; 无源干扰设备; 模块化; 自动检测系统; 故障诊断

## Design of Test Software for Missile Approach Warning System and Chaff/Flare Dispenser

Zhang Qiuxia

(Modern Education Technology Center, Huanghe Science and Technology College, Zhengzhou 450063, China)

**Abstract:** In order to solve the high cost in test and the hard in maintenance of Missile Approach Warning System and Chaff/Flare Dispenser (MAWS-CFD) integrated supports, Automatic Test System for MAWS-CFD has been designed. The system-level test software was designed in the way of modeling. The integration of different software parts was processed according to different requirements of equipment inspection. The characteristic features of system were high integration, flexible configuration, intelligent operation, perfect function, complete testing project and increasing test and obstacle avoidance efficiency. The test system was tried on. The results show that the system can offer technical support for maintenance and testing of MAWS-CFD because of the advanced operability and maintainability, which can save cost in test and maintenance efficiently.

**Key words:** missile approach warning system; chaff/flare dispenser; modularity; automatic test system; fault diagnosis

## 0 引言

导弹逼近告警器及无源干扰设备是飞机的重要电子战装备, 能在作用距离内对空一空和地一空导弹威胁进行逼近告警, 对敌导弹威胁进行干扰, 有效提高飞机对空一空、地一空导弹攻击的对抗能力。目前, 广泛使用的自动检测设备还比较落后<sup>[1-2]</sup>。为了满足新机导弹逼近告警器及无源干扰设备快速检测维修的需要, 同时也考虑到检测维修设备向综合化、自动化、集成化方向发展的要求<sup>[3-4]</sup>, 结合目前较为成熟的自动测试与故障诊断技术, 本文提出研制一种高集成度, 以软件测试为主并结合故障诊断功能的导弹逼近告警器及无源干扰设备自动检测系统。

## 1 测试系统软件设计要求及核心思想

### 1.1 测试系统软件设计要求

(1) 从系统功能上划分软件模块, 明确各模块具有什么功能, 应放在软件体系结构的那个位置, 保持“功能独立”的基本原则。

(2) 软件在任何情况下都不能死机和进入死循环, 应能屏蔽掉各种误操作, 对各种关键操作及错误操作给出警告信息, 当测试结果不合格时, 提示使用者是否继续进行等。

(3) 软件应尽可能保证在不同平台和不同操作系统之间的可移植性, 不同测试接口之间最大的兼容性及互换性以及不同测试系统之间的通用性。

(4) 无论是使用 PXI 内嵌式计算机控制<sup>[5]</sup>, 还是通过 MXI 总线控制<sup>[6]</sup>, 软件都应能实现其功能, 且改动很小。当再次开发类似的系统时, 软件的主程序应尽可能改动很小或不改动, 以达到避免软件重复开发的目的, 并且可以在 Windows2000/XP, UNIX, MacOS 等平台使用而不必做较大的改动<sup>[7]</sup>。

### 1.2 软件设计核心思想

为了克服传统测试软件的通用性、可维护性和可扩充性差的缺点, 本文采用适合绝大多数测试系统的通用软件框架, 如图 1 所示。

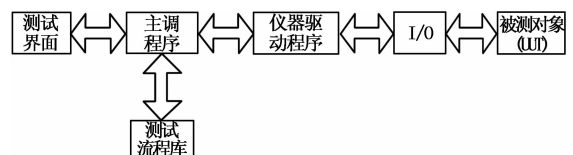


图 1 通用测试软件框架

该框架分为两部分: (1) 用户界面, 这部分是按用户要求根据测试对象编写的, 主要是显示测试结果和完成人机对话; (2) 后台测试程序, 完成对 PXI 模块的操作, 取得测试结果。软件设计的核心部分是编写测试单元需要完成的功能函数, 针对系统所使用的 PXI 模块编写功能函数。当以后系统需要用

收稿日期: 2014-01-18; 修回日期: 2014-02-24。

基金项目: 河南省教育厅科技攻关重点项目(13A520780)。

作者简介: 张秋霞(1980-), 女, 河南偃师人, 硕士, 讲师, 主要从事计算机应用技术等方面的研究。

到新的 PXI 模块时，只需再编写新的功能函数，然后把它添加到函数库中去，可以减少以后组建系统软件的工作量。对于不同的测试系统，只需编写用户界面程序，填写测试数据库（测试流程）和添加专用函数，从而保证了按该框架编制的软件具有较好的通用性、可维护性、可扩充性、移植性、互换性和独立性等特点<sup>[8]</sup>。

## 2 开发平台的选择

软件开发环境的选择，要考虑其先进性、灵活性、成熟性、扩展性、可维护性和高性价比。Visual C++ 6.0 是 Microsoft 公司开发的基于 C/C++ 语言的集成开发工具，集代码编辑、编译、连接、调试等功能于一体，并提供了多种有用的开发工具，它不但大大提高了应用程序的开发效率，还给编程人员提供了一个完整又方便的开发环境<sup>[9]</sup>。采用 C++ 的编程方式已经成为 Visual C++ 开发 Windows 应用程序的主流。

## 3 测试系统软件总体设计

根据软件模块化设计思想，按功能将系统软件分为：人机交互界面和主调度模块、测试任务模块、数据分析模块、系统维护模块、故障诊断模块和在线帮助模块，如图 2 所示。

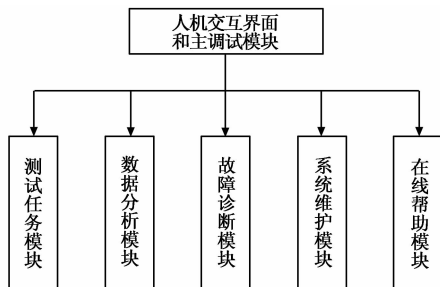


图 2 自动检测系统的软件组成模块

### 3.1 人机交互界面和主调度模块

人机交互界面提供用户登录系统的入口、各子软件包的控制菜单。当某指定任务被选择后，与之相应的工作界面将弹出并被激活。用户应该能够通过通用的界面机制访问所有功能。

应提供汉字的图形化操作界面，并满足下列要求：

- (1) 操作者可随时获得帮助；
- (2) 操作者在工作中应得到足够的提示（包括当前及前后的测试步骤和测试项目、测试参数上下限及其实测值等，测试不合格项应突出显示），重要指令输入前要求重复确认；
- (3) 尽量减小对操作者技能的要求，使操作者工作量减到最小。

主调模块完成测试项目流程的读取、解释、执行、测量结果的获取、判断、显示等操作，主调模块流程如图 3 所示。

### 3.2 测试任务模块

测试任务模块的作用是：选择被测单元（UUT, Unit Under Test），加载相应的测试程序运行，实现可更换单元（简称 LRU）的自动测试、人工测试功能。

主要功能包括如下。

系统初始化：针对特定的 UUT 测试所需的资源，对 PXI 模块进行检测和复位操作，对 PXI 模块故障给出提示信息。

UUT 信息：通过判读在 UUT 适配器上的特征识别电阻，自动判断当前的 UUT，并进行人工确认。

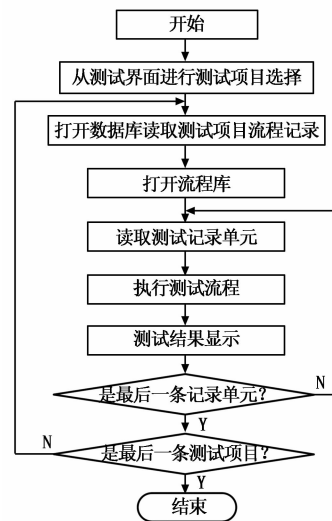


图 3 主调模块流程图

自动测试：按 UUT 的测试程序自动、顺序地执行全部测试项目并记录测试结果。

人工测试：用户可以选择执行多步或单步测试任务，并记录测试数据。

测试任务的流程控制逻辑包括：遇故障停止；单步测试；循环测试。

自动生成测试报告，内容至少应包括：UUT 名称；UUT 产品序列号；测试日期和时间；测试步骤和结果。

此外，任务模块还具备应急处理功能。当被监控的信号超出系统设定的门限时，按照故障等级进行应急处理。

### 3.3 数据分析模块

UUT 选择：提供对历次试验的 UUT 的名称选择、系列号选择等数据库查询功能。数据分析功能都是依据对 UUT 的选择来执行的。

数据浏览：用户可按 UUT 号、UUT 序列号、试验日期等关键字方式，显示及打印 UUT 测试结果等。

故障统计：按 UUT 名称、UUT 序列号等关键字将历次的测试结果进行自动故障统计。

### 3.4 系统维护模块

操作人员管理：记录自动检测系统操作人员和系统管理员的信息数据库，如果是普通操作员登录进入，将不能进入“系统维护”功能。

使用信息：记录系统工作的操作信息，包括操作员信息。

故障维护：对系统工作过程中出现的故障及故障恢复信息进行记录。

### 3.5 故障诊断模块

建立故障数据库和专家诊断意见库，执行测试诊断程序集中相应的测试诊断程序，采集故障数据，判断专家诊断意见数据库中的诊断信息，对被测组件实施故障自动查找，判断出故障区域。

### 3.6 在线帮助模块

在线帮助模块在各个功能软件运行时都能激活，以随时供用户参考。在线帮助采用类似于 Win Help 形式，由控制菜单激活。所有的 Help 信息集中管理，并建立相应的链接关系。

## 4 测试软件开发流程和基本结构

### 4.1 测试软件的开发流程

完整的测试程序集 TPS (Test Program Set) 包括：测试程序、信号调理及电缆转接盒的连线表和测试说明文件。TPS 开发流程如图 4 所示。

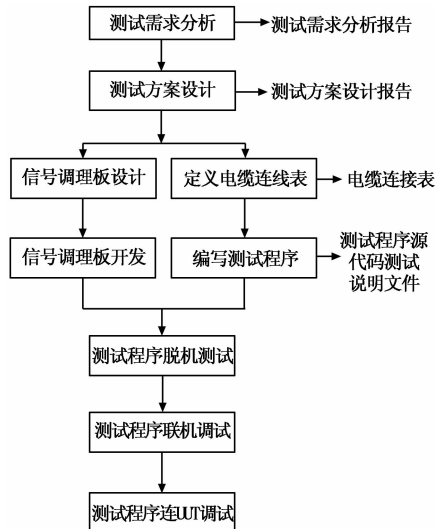


图 4 TPS 开发流程

### 4.2 测试软件的逻辑层次

测试系统软件从逻辑上可以分为 4 个层次。

(1) 仪器驱动层：为 PXI 仪器提供软件编程接口。各模块均按 VISA 规范编写了仪器驱动，对上层软件而言，驱动程序提供了一套方便调用的函数，借助其完成计算机与仪器间的命令与数据传输，完成对测试模块的操纵。测试系统软件对各测试设备的操作是通过各自的驱动程序实现的。

(2) 动作逻辑层：在测试系统软件中，根据测试需求归纳了测试动作，动作逻辑层利用驱动层提供的调用接口实现了测试动作的集合，这些测试动作被流程调度层所用。

(3) 调度与监控层：测试流程以数据库的形式存储在计算机上，测试系统软件读取静态的流程库，由流程调度线程进行解释，并调用相应的动作。

(4) 用户界面层：测试系统软件采用微软推荐的文档视图结构 (Document-View Architecture) 来组织上层程序。为了避免软件界面因硬件操作而死锁，流程调度及数据监测均采用工作线程的方式执行，与软件界面分离。三者之间通过消息、同步事件以及全局变量等方式来进行协调执行。

## 5 测试系统软件设计实现

### 5.1 用户界面层

测试系统软件用户界面设计分为两个层次：负责测试项目的管理主界面和负责测试项目显示的测试界面。

1) 测试项目的管理：测试系统软件功能绝大部分与“测试项目”有关，故采用以“测试项目”为中心的设计方法。测试系统软件提供多个“测试项目”，用户可以选择测试项目、进入测试项目、运行测试项目、退出测试项目。但同一时刻只能有一个测试项目处于打开状态。

测试系统软件采用 MFC 文档视图结构作为主架构。每个“测试项目”相当于一个“文档”。初始情况下，没有打开任何

“文档”，用户可以选择打开不同的“文档”、对“文档”进行操作、关闭“文档”。由于“测试项目”具有互斥的特性，故需要限制打开的“文档”数目，同一时刻只能有一个“文档”处于打开状态。

2) MFC 文档视图框架结构：应用程序采用文档视图架构 (Document-View Architecture)，在 MFC 文档视图结构下，应用程序包括四部分内容如下。

应用程序对象：对应于进程本身，负责创建和维护后三者。

框架窗口对象：是应用程序的用户界面和视图窗口的容器，通常包含工具条、菜单栏等。对于 MDI 类型 (多文档界面) 的应用程序，有一个主框架窗口和若干子框架窗口。

文档对象：负责记录应用程序数据并封装与数据相关的操作。

视图窗口对象：负责在界面中体现数据，并按用户操作来修改维护数据。

应用程序对象包含一个或多个“文档模板”，“文档模板”记录了主窗口、文档、视图窗口的配置。

3) 文档功能设计：测试系统软件需要两类用户界面：测试界面和历史纪录界面。故采用两个文档模板。相应的，有两种类型的文档，测试界面用的文档和历史记录用的文档，分别命名为 CTestDoc 和 CHistoryDoc。前者负责从配置文件、流程数据库及用户输入中读取项目信息，提供流程运行控制功能 (开始、暂停、终止流程)，维护测试期间的流程调度、数据监控等；而后者则只需要提供历史记录信息。

4) 软件配置管理：配置信息用 Windows 配置文件格式 (.ini 文件) 存储。配置信息分为两类，一类为软件全局的配置信息 (如 PXI 模块地址等)，存储于 CFDS.ini 文件中。另一类为各测试项目的配置信息 (如名称、流程表、PXI 模块的启用情况等)，每一个测试项目对应一个配置文件，配置文件名登记于 CFDS.ini 文件中，而各种测试参数采用二进制文件的方式存储。

为了提供配置管理的用户界面，创建多个参数设置对话框，对话框建立时由全局的配置信息对象读出原始设置并加以显示。用户可以在对话框上修改设置，设置完成后，关闭对话框，所作修改存储到配置信息对象，应用程序退出时，配置信息被储存到硬盘。

5) 对话框：在软件的测试过程中需要用到很多的对话框，将他们进行分类设计，包括附属对话框、流程对话框和调试对话框三大类。

(1) 附属对话框：在测试界面和历史记录界面建立、销毁过程中，需要用到多个对话框。项目向导用于在打开测试项目时给出提示、收集用户信息、初始化测试设备。由 3 个页面组成：准备页面、信息收集页面和初始化页面。其中，初始化页面使用启动初始化线程。初始化线程使用动作逻辑层提供的初始化服务。存盘提示对话框用于保存测试结果时提示存盘路径、追加附注及确认储存。历史记录列表对话框用于选择查看历史记录。

(2) 流程对话框：测试项目执行过程中需要大量对话框给出交互界面。分为两类：一类用于操作提示、错误提示等。另一类对应具体的测试动作，有操作人员参与，对测试项目中的参数做

修改和设定。

(3) 调试对话框：为了便于调试，需要设计一些调试面板，使其可在运行期间通过调试菜单打开，脱离流程检查动作执行情况，以便发现、排除程序中存在的错误。在实现新的测试动作时，可先在调试面板中添加按钮，调用动作，经调试无误后，再添加到流程中使用。

6) 日志记录：测试项目存储记录内容较多，包含参数记录、日志记录、流程记录、数据图表等。日志记录用于收集程序各部分发来的着色字符串信息，加上时间，并输出到日志窗口和存盘作为记录文件。日志向用户汇报程序执行进展，提示错误信息。在调试阶段，还可以利用它来辅助调试。

### 5.2 调度和监控层

每个测试项目对应一个或多个“测试流程”，“测试流程”以流程表的形式驻留在数据库中。表名登记到测试项目配置文件中。当测试流程需要包含分支时，按多个流程表处理。流程表定义如表 1 所示。

流程执行过后，流程执行结果用与流程表大致相同的表格存储到结果数据库中。流程的运行独立于用户界面，由单独的线程实现。其执行过程如图 5 所示。

个别测试项目需要在流程执行过程中采集记录电平信号和通信数据包。对于此类测试项目，在建立流程调度线程的同时还将建立数据监测线程。数据监测线程不断读取 1553B 和 RS422 通信数据包并将其以文件的方式进行存储。

### 5.3 动作逻辑层设计

在测试系统软件中，根据测试需求归纳了测试动作，动作逻辑层利用驱动层提供的调用接口实现了测试动作的集合，这些测试动作供流程调度层使用。设备管理包括各 PXI 模块的上电、初始化、自检、关闭和状态查询。测试系统软件采用对象数组的方式进行设备管理，每个设备有一个 ID 标识。初始化线程根据测试项目配置信息访问设备管理服务，初始化该项目所需要的设备。设备初始化后，设备数组中存储了其访问句柄，各测试动作通过对该句柄的访问得以实现。用户关闭测试项目时，查询设备数组中各设备的状态，分别予以关闭或断电。测量动作如下。

(1) 电源开启、关闭：通过串口与电源控制组合通信，开启或关闭其某路电源。需对串口操作进行多线程锁定。

(2) 电源测量：通过串口与电源控制组合通信，读取其

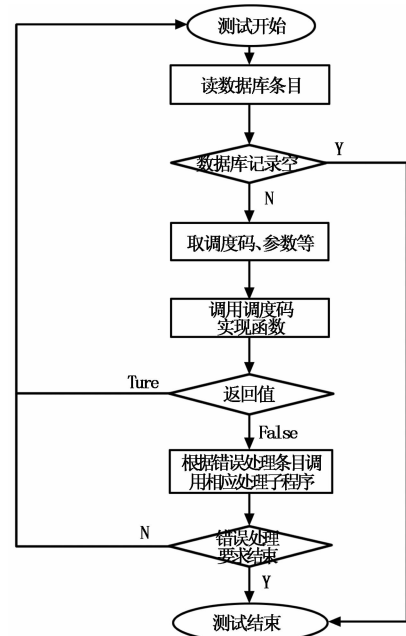


图 5 流程执行图

某路电源的电压或电流值。需对串口操作进行多线程锁定。

(3) 万用表测试：通过万用表模块测量某路电压、电阻、频率等。

## 6 软件运行实例

测试程序被安装到系统后，用户就可在测试程序选择窗口内选择要运行的测试程序。测试程序运行过程中，当测试结果为正常，“结论”一栏的值为“通过”。当测试结果为故障，“结论”一栏的值为“不通过”。当有测试结果为“不通过”，则在测试结束之后给出隔离到 SRU 的故障隔离信息。

## 7 结束语

本文主要研究导弹逼近告警器及无源干扰设备自动检测系统的软件设计与实现，可以得出以下结论：

(1) 采用标准化、模块化、层次化的设计思想，把系统软件划分为人机交互界面和主调度模块、测试任务模块、数据分析模块、系统维护模块、故障诊断模块和在线帮助模块六部分，分析了每个模块功能。它具有很强的通用性、开放性，用

表 1 流程表定义

字段	名称	类型	备注
cxh	序号	整数	数据库条目 ID (自动编号)。
cbz	步骤编号	整数	一个步骤包含多个子步骤，执行时按递增排序，步骤编号从 0 开始连续递增。
czbz	子步骤编号	整数	若为 0，表示此条目为步骤描述，czbzm 中放步骤名称，其他字段无效；若大于 0，表示子步骤编号执行时按递增排序。
czbzm	子步骤描述	字符串	
cdz	调度码	整数	动作的编号。
ccs	参数	字符串	将传给调度码实现函数的参数，如万用表测量中的索引号等。
chg	错误处理	字符串	0, 1, 2, 3 四个数字分别对应：超时处理序号；硬件错误处理序号；软件错误处理序号；状态否处理序号。
cxs	是否显示	整数	1：显示 (打印) 该项；0：不显示 (不打印)。
cjg	结果	字符串	供程序存放结果的字符串。
cru	返回码	整数	供程序存放动作执行的返回码。
note	备注	字符串	

户可以根据需要扩展,使其可对其他同类被测对象进行测试和故障诊断。

(2) 针对设备检测需求,开发不同的软件测试模块并集成,实现了测试程序的重用性和可移植性,提高了软件编程、使用和维护的效率。

(3) 系统集成度高,配置灵活,操作智能化,功能完整,测试项目齐全,并带有故障诊断功能,同时符合地面测试设备标准化、模块化、系列化的发展思路。

(4) 经试用表明,该检测系统具有良好的操作性和维护性,可以满足导弹逼近告警器及无源干扰设备检测维修的迫切需要,具有良好的军事效益。

虽然完成了软件系统的设计和实现,但系统故障诊断子系统处于初步构建中,有待于进一步分析和完善,主要表现在模糊知识库不够充实,不能进行自主学习,需要适时对模糊知识库的内容进行更新。

#### 参考文献:

[1] 肖明清,胡斌,薛辉辉,等.某型导弹发射装置通用检测设备自

检与校准装置设计[J].计算机测量与控制,2013,21(10):2723-2725.

[2] 樊国创.某型飞机机载武器控制系统自动测试设计与实现[D].西安:空军工程大学,2005.

[3] 陶东香,郑海波,茹东生.基于PXI和LabVIEW的无源干扰设备测试诊断系统构建[J].现代电子技术,2011,34(7):120-122.

[4] 于劲松,李行善.下一代自动测试系统体系结构与关键技术[J].计算机测量与控制,2005,13(1):1-3.

[5] 李润生.基于PXI总线的机载计算机通用测试系统的设计和实现[J].航空计算技术,2004,34(4):54-56.

[6] 王宝龙,黄考利,魏忠林.基于LabVIEW的通用自动测试系统设计[J].兵工自动化,2005,24(6):30-32.

[7] 张毅刚,彭喜元,姜守达,等.自动测试系统[M].哈尔滨:哈尔滨工业大学出版社,2001.

[8] 彭芳.PXI技术在某型雷达测试系统中的应用研究[D].西安:空军工程大学,2004.

[9] 乔文军,万晓冬.嵌入式软件覆盖测试工具的研究[J].计算机测量与控制,2007,15(9):1238-1242.

(上接第 1652 页)

动态链接库的调用式有两种:静态调用和动态调用。静态调用(隐式调用)就是在编译和链接应用程序时提供头文件(\*.h)和导入库文件(\*.lib),这样链接器将向系统提供加载DLL所需的信息,并在加载时解析需要调用的DLL函数的位置。应用程序像调用本地函数一样对DLL中的函数进行调用,这种方式操作简单,适用于内存空间足够,要求效率高的程序中。动态调用(显式调用)就是应用程序调用LoadLibrary函数或LoadLibraryEx函数以在程序运行时加载DLL。成功加载DLL后使用GetProcAddress函数获得要调用的DLL函数地址。在使用动态调用时,无需使用导入库文件(\*.lib),依次对要调用的函数进行加载。这种方式的加载速度要比使用静态调用的应用程序快,但使用起来复杂,适用于编写大型应用程序<sup>[8]</sup>。

前面介绍了用VC++完成了DLL建立及生成了导出函数库\*.lib的过程和测试程序的设计,结合天线测试与控制平台软件的特点,在这里选择静态调用即隐式调用的方式来实现DLL的调用。这种方式比较简单易操作并且执行效率高。现在以同步机板的动态链接库ZY\_TBJ\_USB.dll为例,描述一下在界面测试程序下动态链接库具体的调用过程。

(1) 首先在Labwindows/CVI10.0环境下,打开File/Open/Project(\*.prj)选项打开天线系统测试与控制平台界面测试软件名为ST1.prj总工程,所有的测试界面程序都在这个工程中,再把TBJ.h、TBJ.dll、TBJ.lib文件拷到ST1工程所在的目录文件中。

(2) 然后右击ST1.prj工程图标,在出现的菜单下选择Add Existing File选项,把同步机板的TBJ.h、TBJ.dll、TBJ.lib文件分别添加到ST1.prj工程中去。这样就完成了动态链接库的隐式调用。

## 6 结束语

本文将一种LabWindows/CVI和VC++混合编程的软件设计方式应用到雷达天线系统测试与控制平台,充分利用两者的优点构建了雷达天线系统测试与控制平台的整个软件,实现了人性化、友好化的测试界面,同时又保证了对底层硬件设备准确实时地控制。通过对雷达天线进行静态、动态测试,表明该软件能实时准确地将指令信号传输到雷达天线系统并能处理与显示反馈的测试数据。应用表明,该测试与控制平台适合应用于对体积和功能都有较高要求的某型雷达天线系统的故障检测,解决了某型雷达天线系统外场检测的难题。

#### 参考文献:

[1] 初晓军.XX雷达设备[Z].青岛:海军航空工程学院青岛分院,2005.

[2] 张风均.LabWindows/CVI开发入门和进阶[M].北京:北京航空航天大学出版社,2001.

[3] 习郑虎,年夫顺,梁胜利.基于LabWindows/CVI的USB总线微波功率计软件设计[J].计算机测量与控制,2011,19(1):214-215.

[4] 陈凯,赵刚.便携式鱼雷发控系统测试仪研制[J].鱼雷技术,2009,17(5):2-3.

[5] 朱中锐,蔡志明,郭岩.LabWindows/CVI数据采集系统[J].电子测量技术,2007,30(1):105-106.

[6] 耿娟,原亮,吴彩华.DLL在LabWindows/CVI的虚拟仪器中的应用[J].微计算机信息,2006,22(8):115-116.

[7] 理查特.Windows核心编程[M].王建华,等译.北京:机械工业出版社,2000.

[8] 宁娅娜.通过动态链接库实现VC和LabWindows/CVI的混合编程[J].重庆教育学院学报,2011,24(6):76-77.