

分类号 TP311.1

学 号 20211060668

密 级

西安石油大学

Xi'an Shiyou University

全 日 制 学 术 型

硕 士 学 位 论 文



题 目 基于协议符合性的AFDX端系统

自动化测试软件设计

作 者 姓 名 黎万山

导师姓名、职称 王魁生 教授

学 科 (专 业) 名 称 计算机科学与技术

提 交 论 文 日 期 2023年6月9日

论文题目：基于协议符合性的 AFDX 端系统自动化测试软件设计

专 业：计算机科学与技术

硕 士 生：黎万山(签名) _____

导 师：王魁生(签名) _____

摘 要

AFDX 端系统是实现 AFDX 协议的关键设备之一，负责控制和管理飞机各个子系统之间的数据传输。在数据传输中，AFDX 协议是各个端系统设备之间进行数据交换的约定和规范，而符合性测试则是测试这些 AFDX 端系统设备是否遵循规范。对于航空电子设备而言，由于其应用场景的特殊性和对数据传输的高要求，只有确保设备符合协议规范，才能够保证其安全、可靠、稳定地运行。由于 AFDX 协议复杂且当前设备交付需求不断增高，AFDX 端系统测试工作为测试人员带来很大负担。传统的人工测试方式存在人力成本高，效率低、重复性差等缺点，因此，实现 AFDX 端系统协议符合性的自动化测试从而提高测试效率和测试质量是十分重要的。

本文设计实现的测试软件以 Windows 系统为软件平台，以 AFDX 端系统仿真卡为硬件平台，针对端系统的协议符合性实现自动化测试。论文的主要研究工作如下：

(1)本文对 AFDX 协议进行了深入的研究和分析，总结了协议符合性的测试内容。根据端系统功能，测试分为采样端口测试、队列端口测试、SAP 端口测试、余度管理测试和时间同步测试，在此基础上加入安全性测试和可靠性测试。

(2)为了实现易于操作的自动化测试，本文设计了基于 TR-TE 测试策略的 AFDX 端系统自动化测试方案，并搭建了测试环境。针对各项测试内容，本文进行了自动化测试方法研究，并设计了详细的测试步骤。

(3)基于测试内容和测试方法的研究，本文设计并实现了基于协议符合性的端系统自动化测试软件，软件由测试控制软件和测试服务软件组成。本文按照软件的功能和非功能需求设计了软件的总体实现方案，并实现了硬件控制模块、板卡管理模块、测试管理模块、配置表管理模块、测试数据显示模块、测试报告管理模块、程序通信模块、测试响应模块以及用户图形界面。

(4)最后本文验证了测试软件的功能，并将自动化测试与手工方式进行了对比，分析了自动化测试程序的优势。

关键词：AFDX，协议符合性测试，自动化测试，端系统

论文类型：应用研究

Subject: Design of AFDX End System Automated Testing Software Based on Protocol Compliance

Speciality: Computer Science and Technology

Name: Li Wanshan(signature) _____

Instructor: Wang Kuisheng(signature) _____

ABSTRACT

The End System is one of the key devices for implementing the AFDX protocol, responsible for controlling and managing data transmission between various subsystems of the aircraft. During data transmission, the AFDX protocol serves as the agreement and standard for data exchange among End System devices, while conformity testing is conducted to verify if these End System devices comply with the standards. For aviation electronic equipment, ensuring that the devices conform to the protocol standards is essential to ensure their safe, reliable, and stable operation. Due to the complexity of the AFDX protocol and the increasing demand for device delivery, automated testing of AFDX End Systems has posed a significant burden on testing personnel. Traditional manual testing methods suffer from high labor costs, low efficiency, and poor repeatability. Therefore, implementing automated testing for conformity to the AFDX protocol in End Systems is of great importance to improve testing efficiency and quality.

The testing software designed and implemented in this paper is based on the Windows operating system as the software platform, with AFDX End System simulation cards as the hardware platform, aiming to achieve automated testing for protocol conformity in End Systems. The main research work of this paper includes:

(1) In-depth research and analysis of the AFDX protocol, and summarization of the testing content for protocol conformity. Based on the functionality of End Systems, the testing is divided into sampling port testing, queue port testing, SAP port testing, margin management testing, and time synchronization testing, with the addition of security testing and reliability testing.

(2) In order to achieve easy-to-operate automated testing, this paper designs an automated testing approach for AFDX End Systems based on the TR-TE testing strategy and builds a testing environment. For each testing content, this paper conducts research on automated testing methods and designs detailed testing steps.

(3) Based on the research of testing content and testing methods, this paper designs and implements an automated testing software for End Systems based on protocol conformity, which consists of testing control software and testing service software. According to the functionality and non-functional requirements of the software, the overall implementation plan of the software is designed, and the hardware control module, board management module, testing management module, configuration table management module, testing data display module, testing report management module, program communication module, testing response module, and user graphical interface are implemented.

(4) Finally, the functionality of the testing software is validated, and a comparison is made between automated testing and manual methods, analyzing the advantages of automated testing procedures.

Keywords: AFDX, Protocol Compliance Test, Automated Test, End System

Thesis: Application Study

目 录

第一章 绪论.....	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.2.1 AFDX 网络应用现状.....	2
1.2.2 AFDX 测试研究现状.....	3
1.3 论文研究内容.....	4
1.4 论文组织结构.....	5
第二章 相关理论与技术	6
2.1 AFDX 协议介绍.....	6
2.1.1 AFDX 网络组成.....	6
2.1.2 AFDX 协议栈.....	7
2.1.3 虚拟链路.....	8
2.1.4 冗余管理.....	8
2.1.5 完整性检查.....	9
2.1.6 端系统数据发送处理机制.....	9
2.2 协议符合性自动化测试介绍	10
2.2.1 自动化测试概念.....	10
2.2.2 TR-TE 测试策略	12
2.2.3 协议符合性测试.....	13
2.2.4 安全性测试.....	14
2.3 本章小结.....	14
第三章 软件需求分析与设计	15
3.1 测试对象与测试操作流程介绍	15
3.1.1 测试对象.....	15
3.1.2 测试操作流程.....	15
3.2 端系统协议符合性测试内容	16
3.2.1 采样端口测试.....	16
3.2.2 队列端口测试.....	18
3.2.3 SAP 端口测试	18
3.2.4 余度管理测试.....	19
3.2.5 时间同步测试.....	20
3.2.6 端系统安全性测试.....	21
3.2.7 端系统可靠性测试.....	21
3.3 软件需求分析.....	22

3.3.1 功能性需求.....	22
3.3.2 非功能性需求.....	22
3.4 自动化测试软件设计.....	23
3.4.1 协议符合性测试方案.....	23
3.4.2 自动化测试软件架构设计.....	24
3.5 本章小结.....	25
第四章 端系统自动化测试软件实现	26
4.1 软件开发环境.....	26
4.2 测试控制软件实现.....	26
4.2.1 端系统硬件控制模块.....	27
4.2.2 端系统板卡管理模块.....	33
4.2.3 测试管理模块.....	34
4.2.4 配置表管理模块.....	36
4.2.5 测试数据显示模块.....	37
4.2.6 测试报告管理模块.....	38
4.2.7 程序通信模块.....	39
4.2.8 软件图形界面.....	40
4.3 测试服务软件实现.....	40
4.4 本章小结.....	41
第五章 软件测试与分析	42
5.1 测试环境搭建.....	42
5.2 软件功能测试.....	43
5.2.1 硬件控制模块.....	43
5.2.2 板卡管理模块.....	43
5.2.3 配置表管理模块.....	44
5.2.4 测试管理模块.....	46
5.2.5 报告管理模块.....	47
5.3 自动化测试与人工测试方式对比	47
5.4 本章小结.....	48
第六章 总结与展望	49
6.1 论文工作总结.....	49
6.2 未来工作展望.....	49
参考文献.....	51

第一章 绪论

1.1 研究背景及意义

随着航空电子系统的发展,AFDX(Avionics Full Duplex Switched Ethernet,航空电子全双工交换式以太网)^[1]作为一种高速度、高可靠性的总线网络协议^[2],在现代航空电子系统中得到广泛的应用。AFDX 端系统^[3]作为实现 AFDX 网络通信的关键设备之一,是机载设备间交流信息的“关口”,对整个航空电子系统来说都尤为重要。航空电子系统关系着飞机的飞行性能和安全性能,具有一定的特殊性,其对网络通信设备的可靠性要求极高。因此,AFDX 端系统的协议符合性测试工作对于保障航空电子系统的可靠性是必不可少的。

端系统协议符合性测试是为了确保端系统符合 AFDX 协议要求,使其满足正确有效、稳定可靠、完整无误的服务质量要求。AFDX 协议内容复杂,相关测试工作非常繁琐,随着当前 AFDX 端系统的测试需求不断增加,AFDX 端系统的协议符合性测试工作给测试人员带来很大负担。而以往的测试工作是测试人员根据测试需求手动操作不同软件和工具,逐项编写测试程序进行测试,在经过较长时间测试后测试人员才能够确定端系统的合格性。测试过程中存在大量枯燥无味的重复操作,且人工方式容易受到人为因素的干扰使得误操作率提高,最终会导致测试结果出现偏差。以往的测试手段经过长时间的使用和反馈,已经不能满足端系统的测试需求。例如目前的相关测试程序存在测试功能单一,功能的扩展性不足,程序的操作性较差等问题。所以目前急需一款测试功能多样,测试效率较高,灵活度较强的测试软件,用于解决 AFDX 端系统测试需求。

自动化测试指的是测试人员根据测试需求制定相应测试策略,使用测试软件或其他工具实现局部或者全部替代人工,自动地执行测试、比较测试结果以及自动生成测试报告这一过程^[4]。基于对 AFDX 协议的测试特点和测试需求全方位的分析和总结,本文从端系统协议符合性测试的角度出发,设计一款 AFDX 端系统的协议符合性自动化测试软件。相对于人工手动方式的端系统的协议符合性测试,本文设计的自动化测试软件可以在保证测试效果和提高测试的效率的情况下,更加有效利用资源,运行更繁琐的测试项目,有效降低端系统的测试成本,避免人为性因素的失误,使得测试结果更加可靠。

使用自动化测试软件对 AFDX 端系统进行测试的速度将远远快于往常的人工方式,并且自动化测试软件能够整合各类测试项目,测试功能更加全面,测试人员通过测试软件还可以看到被测对象的实时的详细数据,并且软件可以自动地对测试过程产生的数据进行记录和分析,最终自动地生成测试报告^[5]。因此,AFDX 端系统自动化测试软件对于降低测试成本,减少测试人员劳动量,提高测试全面性和精确度,缩短端系统的测试周期,提高测试重用性以及提供规范化测试流程具有重大的意义^[6]。

1.2 国内外研究现状

1.2.1 AFDX 网络应用现状

早在 20 世纪 90 年代, 由欧洲多个国家联合成立的欧洲空中客车公司就启动了研究计划, 直到 2005 年航空电子技术委员会 AEEC(Airlines Electronic Engineering Committee) 的飞机数据网络工作组将 AFDX 标准化, 形成了 ARINC664 标准^[7]。在同一时期, 空客公司的军用运输机项目 A400M 也采用 AFDX 总线作为其航电内部骨干网络进行数据传输, A400M 也是首个使用 AFDX 总线的大型军用运输机, 该型飞机上电缆重量大幅度减轻^[8]。

随后使用 AFDX 总线网络的飞机越来越多, 加拿大庞巴迪宇航公司生产的庞巴迪环球 5000(Bombardier Global 5000), 苏霍伊航空集团设计制造的 SSJ100(Sukhoi Superjet 100), 美国波音公司的最新型宽体中型客机 B787, 以及我国商飞的国产大飞机 C919 及 ARJ21 支线客机等相当数量的国内外民航飞机使用 AFDX 总线网络作为飞机中航空电子系统主干连接网。随着 AFDX 网络的广泛应用, 我国航空电子通信系统迈入了高度综合化的新时代^[9]。

国外对 AFDX 的应用已经非常成熟, 已形成了完整的解决方案。国外已有不少公司在进行 AFDX 相关设备和技术的研发。例如成立于 1991 年 SYSGO 公司, 主要致力于 AFDX 协议栈与实时嵌入技术的研究, 其获得 DO-178B 认证的 AFDX 协议栈在空客 A380、A400、波音 787 等大型客机成功应用。还有 ARINC 产品供应商阿尔塔数据技术有限公司(Alta Data Technologies LLC)提供各种 AFDX 相关航电设备, 曾连续两年荣获 2020 年和 2021 年最佳航空航天供应商奖。还有 Goebel 公司提供相关 AFDX 软件和硬件解决方案定制。除此之外, 还有相关 FPGA 优势企业如美国 ACTEL 也有针对 AFDX 网络的相关解决方案。

当前, 国内大型民机的电子通信设备研究水平仍然比较低, 很多关键技术尚未完全掌握, 部分设备严重依赖于国外进口, 与欧美等发达国家还有很大的距离。随着 AFDX 网络的在国内的应用逐渐增多, AFDX 网络开始在部分飞机上使用, 目前已自主研发出了 AFDX 端系统传输卡、TAP 卡、监控卡等航空电子设备, 以及完成了 AFDX 网络综合实验, 国内对 AFDX 相关设备的设计生产能力已初具形态^[10]。

如今在各行各业“军民融合”的大背景下, 催生了众多民用企业积极发展探索相关 AFDX 产品, 例如北京石竹科技股份有限公司, 四川瑞航电子科技有限责任公司, 成都成电光信科技股份有限公司, 北京泛华恒兴科技有限公司等公司从事研发 AFDX 网络相关设备产品。产品包括 10M/100MAFDX 端系统仿真卡, AFDX 交换机, AFDX 监控卡等。西安航空计算技术研究所也已实现千兆 AFDX 端系统设计^[11]。然而, 我国在 AFDX 网络的研究与实现方面还处在初级阶段, 应用规模不大, 相比国外还有一定滞后性, 这

制约了我国航空电子系统综合技术的发展^[12]。实现国产 AFDX 网络设备在民用大客机以及军用大飞机等相关领域的广泛应用，对于国家的国防建设以及经济发展都有着非常重要的作用。

1.2.2 AFDX 测试研究现状

国外针对 AFDX 网络测试技术的研究很早就拉开序幕，到如今已经发展得比较成熟，在全世界范围内处于领导地位。如美国 GOEBEL 公司生产的 AFDX 测试卡 GXP1000 和相关套件 Goebelyzer 用于数据采集、数据生成和错误注入等功能并支持最新 EDE 扩展。Excalibur 生产的 EXC-664PCIE 测试和仿真板可以支持多种协议。此外，许多国外的航电供应商比如：AIM、Abaco 等公司，都开发出了一系列的测试卡，以供各航空电子系统的开发人员对 AFDX 上层网络协议进行测试和开发。国外针对 AFDX 端系统的测试仍不断在研究，不管是测试策略和测试设备都在不断地升级迭代。

国外众多企业产品中，其 AIM 公司设计制造的 PBA.pro 测试系统和德国 TechSat 公司的 ADS2 产品系列具有较高的知名度。这些测试系统集成了自主研发的测试硬件，通过扩展也可适用于其他第三方厂商的硬件，这些测试系统还提供二次开发的接口。但是由于价格高昂，且本地化服务支持不够及时，所以国内相关公司机构很少购买使用。

在国内，也有很多公司例如旋极信息科技、上海爱韦讯信息技术、北京润科通用科技、北京华力智飞科技等，积极研发相关自主的 AFDX 测试系统。目前除了国内的几家公司外，国内的部分高校和学者也在积极开展 AFDX 有关的测试技术研究。

2021 年朱晓飞^[13]提出了 AFDX 总线传输标准指标的推导方法，给出了总线传输特性测试的要求标准指导体系。李晓庆，张志平^[14]研究了 ARINC664 中完整性检查及冗余管理等规则，针对于 AFDX 端系统完整性检查单元及冗余管理单元的测试要点，提出了一种基于 AFDX 端系统的冗余功能的测试系统，设计了测试用例，并对其进行了验证。

2019 年侯溪溪，周卓等^[15]通过将思博伦公司的 TestCenter 软件和 i-Test 软件进行二次开发，使用思博伦测试板卡和测试主机，实现了一种综合化自动化的 AFDX 网络性能测试工具，工具可以对测试进程进行管理，自动生成测试报告。

2017 年王磊，陈琳^[16]等针分析了传统 AFDX 网络测试技术方法存在的问题，提出了 AFDX 网络测试 3+1 整合法的设计，为 AFDX 网络测试技术方法应用领域未来发展提供新的思路。

2016 年宁毅^[17]通过对 ARINC664 协议分析，提出符合性验证方案和阶段策略，明确 ARINC664 中定义的需求。

2015 年付军立^[18]针对 AFDX 总线 10M 接口电器参数极限适应性的测试需求，提出一种 10M 标准 AFDX 总线信号发生器实现方案。

国内的相关研究对 AFDX 测试的发展有着一定的推动作用，但是仍然有一定的局限

性，这些研究或者趋于理论方面，或者测试功能相对单一。综上所述，我国 AFDX 产品测试行业的发展并不够完善，在 AFDX 测试理论研究和测试产品的技术积累上，都存在着很大的发展空间。目前，AFDX 协议的自动化测试尚未建立起一个业界级别的检测规范，因此，我国企业及产业研究机构都是按照自己的实际状况来设计测试方案，有时还不得不花费高昂的价格采购国外的测试系统和设备。我国在自动化测试的研究主要集中在汽车、信息软件等方面，对于 AFDX 的端系统的测试还没有形成一个高度自动化的、高度国产化的测试保障体系，所以，开展 AFDX 端系统的测试研究能够弥补相关领域的不足。

1.3 论文研究内容

基于协议符合性的 AFDX 端系统自动化测试软件在端系统测试领域具有广阔的需求，在端系统生产以及交付的时候，AFDX 端系统需要进行大量机械重复的测试。传统的测试采用人工方式，测试效率和测试质量都有很大不足。采用自动化方式实现端系统自动化测试将有效提高 AFDX 端系统的测试效率和测试质量。自动化测试软件在开发之前需要开发人员对 AFDX 协议标准充分研究并理解，充分考虑未来测试软件的扩展性、易用性和可靠性，以及解决按照什么方案来测，需要从哪些方面测，如何实现自动化测试等问题。根据以上问题，本文具体的研究内容如下：

(1) AFDX 协议符合性测试内容

根据 AFDX (ARINC664 Part7) 协议相关文档，结合实际测试业务的调研，本文分析并总结 AFDX 端系统的协议测试内容，并加入了端系统的安全性测试和可靠性测试。在此基础上，针对每个测试项目研究并设计了完整的测试步骤。

(2) AFDX 协议符合性自动化测试总体方案设计

为了实现测试规范性和测试可操作性，通过研究并分析端系统测试方法、相关测试工具以及已有测试平台，本文根据 TR-TE 测试策略，设计了自动化测试实现方案并搭建端系统的测试环境。

(3) AFDX 自动化测试软件的设计与实现

根据 AFDX 测试方案和测试步骤以及 AFDX 端系统的硬件数据手册和测试需求，以 Windows 平台的 C/S 架构，本文按照用户友好的交互体验和具有现代化用户界面风格的原则设计开发基于协议符合性的端系统自动化测试软件。软件分为测试控制软件和测试服务软件。

(4) 软件功能验证与分析

本文搭建了软件的测试环境，对本测试软件的功能模块分别进行了验证。通过对比自动化测试与人工测试方式，总结并分析了基于协议符合性的 AFDX 端系统自动化测试软件的优越性。

1.4 论文组织结构

本文分为 6 个章节，全文安排如下：

第一章为绪论。本章首先介绍了基于协议符合性的 AFDX 端系统自动化测试软件的研究背景，接着分析并阐述了 AFDX 网络应用现状和 AFDX 测试的国内外研发现状，最后简要介绍了本文的主要研究内容和组织结构。

第二章为相关理论与技术综述。本章首先对 AFDX 网络协议进行简要介绍，包括 AFDX 网络组成、AFDX 协议栈、虚拟链路、冗余管理、完整性检查和端系统数据发送处理机制。其次介绍了自动化测试和端系统测试的相关概念，包括自动化测试周期方法学、TR-TE 测试策略、协议符合性测试和安全性测试。以上相关理论为课题后续研究奠定了基础。

第三章为 AFDX 端系统协议符合性软件需求分析与设计。首先介绍了测试对象和端系统的一般测试操作流程。接着根据 AFDX 协议本章总结了端系统协议符合性测试内容，对每个测试项目进行了分析并设计了详细的测试步骤。最后，本章对软件的功能性需求和非功能需求进行了分析，基于以上研究设计了协议符合性测试方案和自动化测试软件的架构。

第四章为 AFDX 端系统协议符合性自动化测试软件的实现。基于前一章软件需求分析与设计，本章完成了基于协议符合性的 AFDX 端系统自动化测试软件的实现。自动化测试软件包括测试控制软件和测试服务软件。测试控制软件由硬件控制模块、板卡管理模块、测试管理模块、配置表管理模块、测试数据显示模块、测试报告管理模块和程序通信模块组成。测试服务软件由测试响应模块、硬件控制模块和程序通信模块组成。

第五章为软件的测试与分析。本章按模块测试了软件各个模块的相关功能并通过对比人工测试方式分析了自动化测试的优势。

第六章为总结与展望。在这一章节中对基于协议符合性的 AFDX 端系统自动化测试软件这一课题以及相关工作内容进行了总结，提出本文有待发展的地方，对此课题后续研究工作展望。

第二章 相关理论与技术

针对 AFDX 端系统的协议符合性进行测试,研究 AFDX 网络协议是必不可少的。本研究课题的相关理论与技术主要包括两方面,首先是 AFDX 网络协议相关理论,其次是协议符合性自动化测试相关理论,本章对这两方面进行论述。

2.1 AFDX 协议介绍

AFDX 网络协议是一个用于航空电子网络中高集成度、确定性的通信协议,作为 ARINC 664 协议族的一个子集^[19],ARINC 664 Part7 定义了航空电子交换式以太网网络的一般特性和要求,而 AFDX 则定义了该网络在空客飞机上的具体实现,并增加了一些额外的特性以满足空客航空电子系统的要求。

目前广泛使用的商用以太网 MAC 层协议是 CSMA/CD,该协议在网络上存在冲突,导致以太网的确定性、实时性等有着明显缺陷^[20]。而这对于数据的实时性、可靠性、安全性等要求比较高的航空电子系统的网络通信系统中是不可接受的^[21]。

为了满足航空电子更广泛和更特殊的要求,AFDX 网络的设计理念与传统的以太网大不相同,通过在以太网基础上进行一系列设计改进,使 AFDX 网络具有高实时性、高安全性、高可靠性、低时延等特点,以满足航空电子系统在鲁棒性、兼容性、扩展性等方面的需求^[22]。

2.1.1 AFDX 网络组成

AFDX 网络主要由三部分结构组成:进行数据存储转发的 AFDX 交换机、作为通讯节点的端系统和用于传输数据的链路^[23]。

AFDX 传输链路(Link):链路作为数据传输介质连接设备,分为地面和空中两种类型,在地面使用非屏蔽双绞线,在空中使用同轴电缆。

AFDX 网络交换机(Switch):AFDX 交换机^[24]用于实现数据交换,各交换机之间可以通过级联扩展。交换机按照预先配置的路径,实现帧在多个端口的全双工无阻塞地转发。通过警管、过滤等手段,将数据包高效、可靠发送到正确的目的端,同时保证数据的正确完整。

AFDX 端系统(End System, ES):端系统^[25]作为数据交换接口设备,负责发送和接收航电子系统中的数据。数据经过端系统协议栈处理,转变为符合 AFDX 协议的数据包,数据包再通过交换机转发送给其他的端系统。反之,AFDX 端系统同样可以接收链路中的有效数据并在转换后传输给航电子系统。

AFDX 网络拓扑结构仅使用星型拓扑^[26],同时 AFDX 网络通过互为备份的两条链路增强数据传输的可靠性,AFDX 网络双冗余星型拓扑结构如图 2-1 所示。

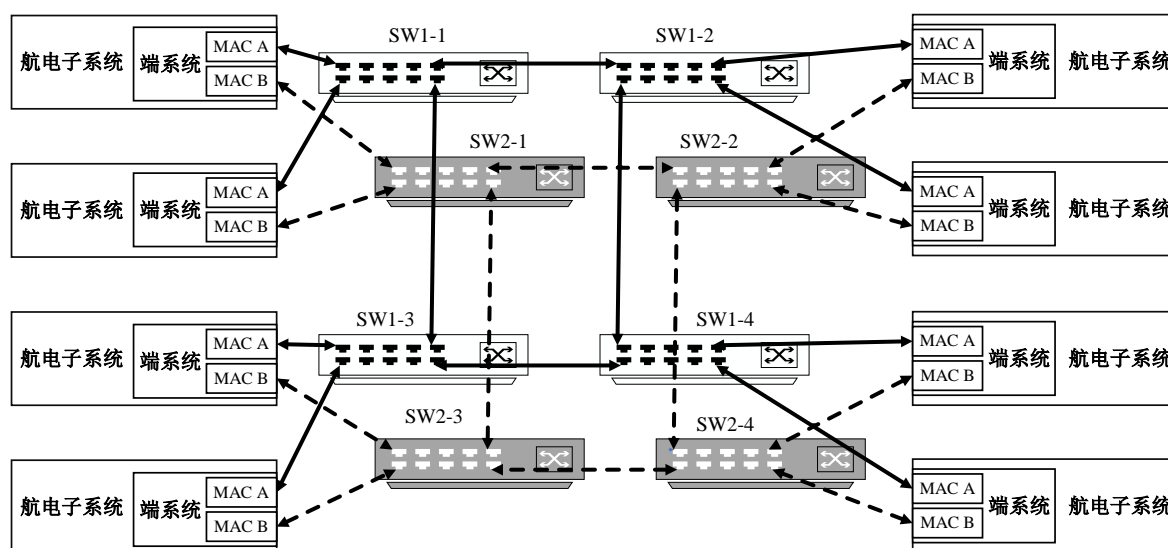


图 2-1 AFDX 网络拓扑

2.1.2 AFDX 协议栈

AFDX 协议是在商业以太网的基础上改进而来的，但与其还有所不同，与 OSI 七层模型相比，AFDX 协议将应用层、表示层及会话层设计为应用层，AFDX 协议栈分层概述^[27]如下：

(1)应用层：应用层主要负责提供应用通信接口，组织数据传输。AFDX 提供三类端口：采样端口、队列端口和 SAP(Service Access Point, 服务访问点)接口。应用程序通过调用不同的端口收发消息，从而实现特定应用。采样端口只能存储单个消息，采样端口读取消息完成后不刷新缓冲区，单条可以重复读取。队列端口可以存储指定个数的消息(该参数可配置)，数据发送顺序和接收顺序保持一致，消息在队列中被读取后会被删除。SAP 端口主要用于实现 TFTP 和 SNMP 等服务。为了和传统网络协议兼容，通过端口设置，可以使用 TCP 协议来传输。

(2)传输层：传输层提供下层网络与上层应用之间的接口，负责向低层网络协议解析来自一个或多个应用的数据，AFDX 在传输层采用的是 UDP 协议。

(3)网络层：AFDX 网络层使用 IP 协议，使用 IPV4 格式地址，负责数据包寻址和路由，实现数据包的分片、重组功能。

(3)数据链路层：数据链路层使用虚拟链路来定义多个单向逻辑链路，在 MAC 地址中包含了 16 位的虚拟链路地址。数据链路层还实现冗余管理，提高了数据传输的可靠性。

(5)物理层：物理层负责激活、维护和停用设备之间的物理电路，并提供时钟信号，负责将信号识别为二进制 1 和 0。物理层功能由端系统上的电路连接网络端口的 PHY 芯片实现。传输介质须满足飞机飞行环境要求，要求在 ARINC664 part2 定义。

2.1.3 虚拟链路

虚拟链路(Virtual Link, VL)定义了两个端系统之间在逻辑上的单向连接^[28]。一条 VL 消息的源只有一个,目的可以有一个或者多个。虚拟链路是建立在网络路由中的虚拟路径,与过去航电设备点对点连接的方式相比,大大节约了物理线缆的开支。

虚拟链路通过设定最大可用带宽,满足实际带宽的分时复用需求^[29]。虚拟链路通过流量整形来减少突发流量情况对虚拟链路正常运行造成的影响以及提供链路的逻辑隔离,任何一条的 VL 的可用带宽不受其他 VL 影响,进而满足 AFDX 网络的确定性。在建立每条虚拟链路时会用到以下参数:

带宽分配间隔(Bandwidth Allocation Gap, BAG)^[30]:即两个连续 AFDX 帧的起始位之间的最小时间间隔。用于防止发送方过于频繁地发送数据包引起的网络拥塞和冲突。每个虚拟链路被分配了特定的带宽分配,确定了可用于数据传输的带宽。

最大帧长度(Lmax): Lmax 指在 VL 上能够传输的最大帧的长度,协议中要求为 1518 字节。

BAG 和 Lmax 保障了 AFDX 网络传输数据的实时性和确定性,根据发送信息的特征来给每条虚拟链路来选择一个合适的 BAG 和 Lmax,从而保证每条消息能在规定的时间内发出^[31]。

VL 的流量将会被规整,抖动(Jitter)^[32]由端系统进行流量整形所引起。当端系统的多条 VL 在同一时间有数据满足条件需要被发送,当 BAG 达到后,VL 可能会等待其他数据发送完毕,这样会产生了一定的延迟时间,这个延迟时间叫做抖动。协议对抖动的最大值(max_jitter)作出了规定,其必须满足公式(2-1)。

$$\begin{cases} \max_jitter \leq 40\mu s + \frac{\sum_{i \in \{VL \text{ 的集合} \}} (20 \text{ bytes} + L_{\max, i} \text{ bytes}) \times 8 \text{ bits/bytes}}{Nb w \text{ bits/s}} \\ \max_jitter \leq 500\mu s \end{cases} \quad (2-1)$$

公式中 max_jitter 指最大抖动时间,当抖动大于了这个数值即表示 BAG 与 Lmax 的设定是错误的,需要重新设定。Nb w 指介质带宽,为 100Mbit/s。40μs 是最小的固定技术抖动。

由公式可得,当一个 AFDX 终端系统的虚链路数量很少,而其数据帧较短,所容许的最大抖动就会减小,在任何情况下 AFDX 网络最大允许为 500μs。

2.1.4 冗余管理

AFDX 网络采用冗余管理机制来增强系统的可靠性,该机制通过互为备份的两条链路实现。发送端系统的虚链路层对要发送的数据帧进行编号,每条虚拟链路的数据帧中的帧序号 SN 从 1 开始,每增加一个数据帧序号加 1,当 SN 号达到 255 时,回滚到 1,

帧序号 0 保留, 用来表示发送端系统复位。

端系统基于各个 VL 配置信息中的冗余设置, 对经过处理的数据帧进行传输网络的选择, 如果配置为冗余发送, 则将该数据帧发往 2 个独立的网络。在终端系统的接收过程中, 接收端的两个端口收到来自两个不同网络的消息。两个数据帧分别存储在相应的缓存中, 并在最后分别接受完整性检查。

在每个冗余网络, 完整性检查对 SN 号在 $[PSN+1, PSN+2]$ 区间的数据帧进行检查, 其中 PSN 是该虚拟链路接收到的前一个数据帧的 SN 号^[33]。端系统通过冗余和完整性检查保障 AFDX 传输数据帧的有效性, 如果出现了故障, 完整性检查将丢弃错误帧。

接收端通过比较两个数据帧的 SN 号实现冗余管理。AFDX 采用“先到先处理”的原则^[34], 冗余管理对这两个数据帧做出选择, 端系统记录收到第一帧数据时该数据的 SN 号, 直接丢弃接收到的第二个相同 SN 号的数据帧, 最后将有效数据帧交给协议栈进行处理。

2.1.5 完整性检查

AFDX 网络中数据帧的完整性主要有四个方面: 源完整性, 位完整性, 顺序完整性, 以及时间完整性^[35]。源完整性通过一系列含有源端系统信息确认, 包括: 数据源端口号、源 MAC 地址等。数据的位完整性通过数据的 CRC 校验来确认。数据传输的顺序完整性通过数据帧携带的帧序列号进行确认。数据的时间完整性是通过数据传输时间确认。

从以上可以看出, 对于源完整性、位完整性和顺序完整性的检查可以通过直接从数据帧中的信息的比较来实现, 对于时间完整性的检查, 需要以数据的传送时间为依据来进行, 只有数据传输时间处于合理阈值才表明端系统接收到的数据帧符合时间完整性的要求^[36]。

2.1.6 端系统数据发送处理机制

AFDX 协议由不同的层次构成, 每个层次负责处理不同的任务。从协议层次的角度来看, 端系统发送 AFDX 消息的过程分为以下步骤, AFDX 端系统消息发送处理机制如图 2-2 所示。

首先是应用层。消息发送前, AFDX 端系统加载预先设定的配置表, 主机应用程序调用 AFDX 端系统驱动软件 API, 消息负载由主机总线传输给端系统。通过中断信号响应, 运行在端系统中的底层传输软件开始对消息进行协议栈处理。

其次是传输层。应用层数据被添加 UDP 头, 头中包含了源 UDP 端口号和目的 UDP 端口号, 这些端口号大多数情况下在端系统的配置表中定义。在 SAP 端口中, 目的地址 IP 和目的 UDP 端口号由应用程序定义, 最后传输层将应用层的数据帧传递给网络层。

接着是网络层。网络层决定是否对 UDP 数据分片, VL 配置中定义的最大帧长度参数决定了分片的大小, 然后网络层对每个分片添加 IP 头, 计算 IP 校验和, 添加以太网

头，最后将其传递链路层。

链路层将以太网帧排队到 VL 队列中并添加 SN 号，帧在冗余管理中被复制，最后按照帧发送的物理端口 ID 修改以太网源地址。

最后，物理层使用 PHY 芯片处理数据，数据被转换为电信号在物理媒介上传输。

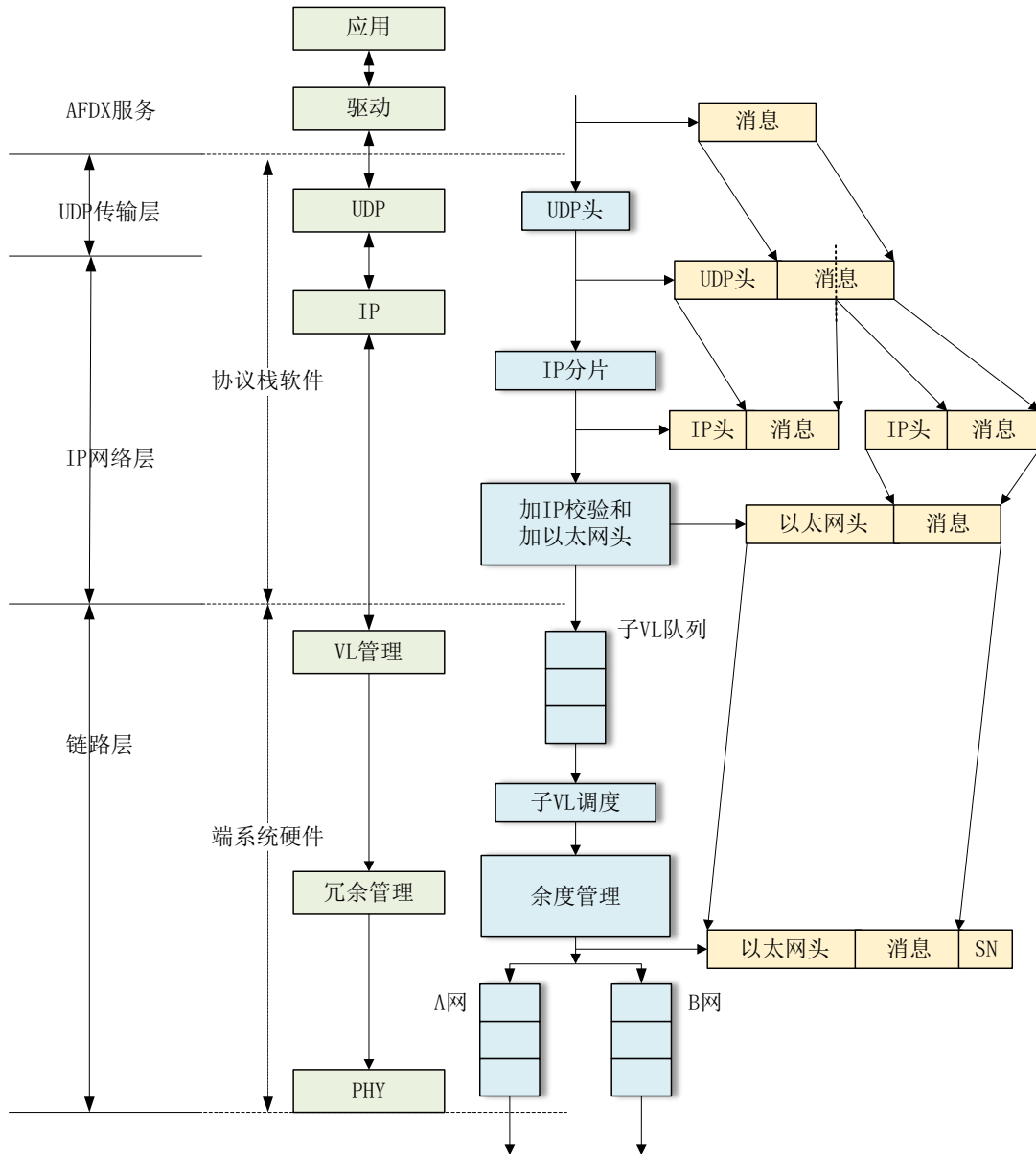


图 2-2 系统数据发送处理机制

2.2 协议符合性自动化测试介绍

2.2.1 自动化测试概念

测试是一种系统性的、有目的的活动，旨在检测和评估一个产品或系统的属性或功能，判断其是否满足的需求和标准。测试可以帮助开发人员和质量保证团队检测和纠正

错误。产品开发的阶段通常需要进行不同的测试。传统的测试工作由测试人员人工操作进行，需要花费大量时间和大量成本。而自动化测试就是使用计算机或其他测试工具自动执行测试的过程。

测试人员进行自动化测试时，首先根据测试需求设计出测试用例并配置好测试环境，之后使用测试工具自动地执行测试步骤。自动化测试可以有效整合软硬件资源，通过使用机器代替人工，不但可以降低测试工作难度，提高测试效率和降低人力成本，还能提高测试结果稳定性和准确性^[37]。

自动化测试的核心是实现测试工作的“自动化”，“自动化”就是要把原来人工操作内容例如：测试数据的录取，测试参数的配置，测试步骤的执行，测试结果的验证等，转变由计算机自动地执行这些测试步骤来完成测试^[38]。

按照一般测试方法的分类，自动化测试可以分为多种不同类型^[39]。按照是否覆盖源代码划分：黑盒测试、白盒测试、灰盒测试。按照测试内容划分：功能测试、性能测试等。按照开发阶段划分：单元测试、集成测试、系统测试、验收测试。此外，还有冒烟测试、随机测试、回归测试、探索测试等不同类型的。

自动化测试获得最佳结果，必须遵循自动化测试的完整周期，从而实现最大化的测试覆盖率。自动化测试周期方法学(Automated Test Lifecycle Methodology, ATLM)^[40]由六个主要过程组成，自动化测试生命周期如图 2-3 所示，每个过程内容如下所述。

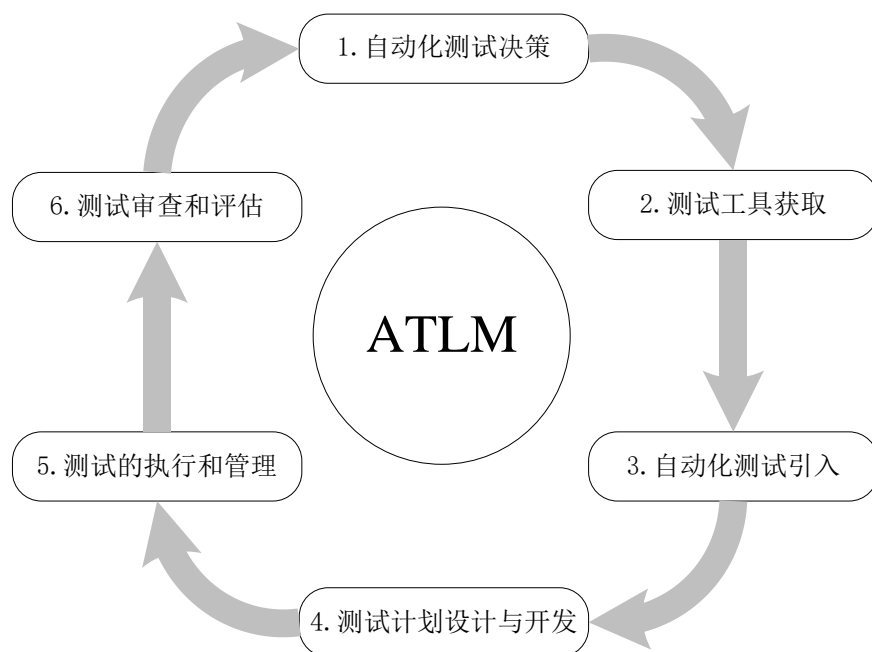


图 2-3 自动化测试生命周期

(1) 自动化测试决策阶段

在这个阶段，测试人员需要明确自动化测试的需求，根据测试项目的实际情况评估来决策是否需要引入自动化测试，团队需要估计自动化测试的成本对于产品的收益是否有利。

(2)测试工具获取阶段

目前没有一个商业上可用的工具可以自动创建一个全面的测试计划的同时支持测试设计和执行，也没有一个单一的测试工具可以用来支持所有的操作系统环境。所以，选择一个合适的自动化测试工具对测试的实施至关重要。

(3)自动化测试引入阶段

项目引入自动化测试，需要对测试过程分析，收集测试指标，选择测试策略，定义测试目标等等。

(4)测试计划的设计与开发阶段

测试人员需要制定完整的测试计划。计划将定义测试人员角色和责任，项目测试时间计划表，测试环境包括软件、硬件以及网络资源，测试验收标准等。最终根据实际情况，开发测试程序。

(5)测试的执行和管理阶段

在测试准备工作完成后，执行为测试项目定义的测试。测试执行时，测试需要遵守测试时间计划，测试执行后，进行测试结果的评估并准备测试报告。

(6)测试项目的审查和评估阶段

在测试执行后，测试团队需要审查测试程序，以确定测试程序哪些方面可以进行改进。最后还要评估测试指标是否满足验收标准，存储测试过程中产生的数据以便日后访问，并且收集积极有效的测试反馈。

2.2.2 TR-TE 测试策略

TR-TE 测试策略即命令-响应测试策略^[41]，TE(Test Equipment)为测试端，是测试命令的发出者和管理者。TR(Test Response)为测试响应端，即被测端，是命令的执行者，没有收到命令之前，不会进行任何操作，TR 根据收到的指令执行不同的测试操作。

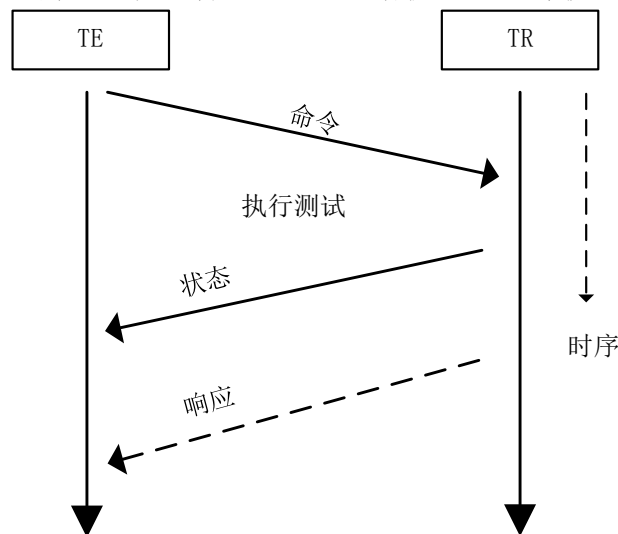


图 2-4 TR-TE 测试时序关系

TR 和 TE 的两个端口之间通过 AFDX 链路连接, TE 发送控制命令, TR 根据命令类型执行该命令, 并将结果返回给 TE, TE 根据 TR 返回的内容判断测试是否通过。TR-TE 测试时序关系如图 2-4 所示。

TR-TE 测试的一般步骤^[42]为:

第 1 步: TE 发送系统复位 Reset 命令, 复位 TR。TR 在接收到复位指令后, 会清空缓存数据, 初始化虚拟链路和端口等配置, 并返回复位完成 ResetOK 状态给 TE, 表示复位完成, 此时, TE 和 TR 之间的通信链路已建立。

第 2 步: TE 发送测试指令, 例如发送采样消息指令。TR-TE 测试策略中用到的所有指令信息和通信端口信息, 都为预先规定好的。TR 接收到指令后, 将待发送消息放入对应发送端口的发送队列中, 随后将发送队列中存放的数据发送出去

第 3 步: TE 解析接收到的信息, 将数据帧中指令字段删除, 将有效负载与期望值进行比较, 验证 TR 发送数据的有效性, 从而判断 TR 的测试结果。

2.2.3 协议符合性测试

协议作为管理计算机网络设备间通信的惯例和规则, 对设备间相互通信以及确保网络的顺利运行至关重要。协议定义了设备之间交换的信息的格式、时间、顺序和错误控制等以确保数据被可靠和有效地传输。

目前的协议一般都是由自然语言来描述的文本, 由于协议标准的实施者在实现协议过程会对这些协议文本有不同的理解误差或其他特殊偏离情况等^[43]。所以, 在协议实现过程中, 非形式化因素将导致出现相同协议的不同实现, 甚至可能出现完全错误的协议实现^[44]。所以为了衡量协议标准和协议实现之间的符合程度, 需要进行协议的符合性测试。

协议符合性测试是一个验证设备或软件的实现是否符合特定网络协议的规范和要求的过程, 是一种功能性测试或性能测试。符合性测试的目的是确保设备或软件的实现符合协议的规格和要求, 是对设备的协议实现的准确性和有效性的重要保障。测试内容包括测试协议的语法、结构和行为, 并验证所有强制性和可选性功能正确实施。

协议符合性测试目前没有统一的测试分类方法, 国内学者从不同角度对其进行了简要分类^[45]。按照测试主体所处的阶段划分: 过程符合性测试与验收符合性测试。按照测试方式的不同划分: 同步标准测试、计划标准测试、追加标准测试等。

符合性测试对于生产企业来说, 促使了企业更加注重产品质量和服务质量, 形成产品促进测试技术提高, 更好的测试技术提高产品品质的良性循环。对于消费者来说, 在一定程度上保障消费者的基本权益, 使消费者可以更好掌握市场的主动权。最后, 协议符合性测试对于我国在产品服务的技术水平, 标准制定等国际竞争中取得优势地位具有重要意义。

2.2.4 安全性测试

飞机的安全性保障对机载设备的稳定运行要求极高，所以对端系统进行安全性测试也是至关重要的。安全性测试的目的是确定系统的安全性能，使其免受安全威胁。与普通测试有所不同的是，安全测试考虑的是有意或者无意的、使得系统产生不安全行为的具有攻击性的操作。

从测试出发点来看，与普通测试不同的是，安全测试是从攻击者的角度出发发现系统的安全漏洞，保证程序不被恶意使用或者破坏。安全测试通常需要做出普通用户不会去做的事情，例如发送长度为 0 的消息。安全测试人员需要像恶意攻击者那样寻找系统的可被攻击的漏洞，更多的是使用反向数据，模拟攻击者精心设计的具有攻击性的数据，更大比例的异常测试用例，安全测试的需求是创造不满足需求的条件，破坏软件的正常运行。安全性测试的几点测试要求如下：

(1) 运行强度测试要求

运行强度测试要求^[46]是指强制软件运行在不正常直到发生故障的条件下(设计的极限状态到超出极限)，检验软件可以运行到何种程度的测试。一般需进行：降级能力的强度测试，系统饱和测试，性能的强度测试，系统健壮能力测试。

(2) 恢复性测试要求

恢复性测试要求是对有恢复或重置(reset)功能的软件的每一类导致恢复或重置的情况，逐一进行的测试，以验证其恢复或重置功能。恢复性测试是要证实在克服故障后，系统能否正常地继续进行工作，且不对系统造成任何损害。恢复性测试一般需进行：探测错误功能的测试，能否切换或自动启动备用硬件的测试，在故障发生时能否保护正在运行的作业和系统状态的测试，在系统恢复后，能否从最后记录下来的无错误状态开始继续执行作业的测试。

(3) 边界测试要求

边界测试要求是对软件处在边界或端点情况下运行状态的测试^[47]。边界测试一般需进行，软件的输入域或输出域的边界或端点的测试，状态转换的边界或端点的测试，功能界限的边界或端点的测试，性能界限的边界或端点的测试，容量界限的边界或端点的测试。

2.3 本章小结

本章首先对 AFDX 网络协议的网络架构、网络协议栈、重点特性进行了相关介绍，对自动化测试软件的测试内容分析提供理论基础。其次本章对 AFDX 协议自动化测试概念和协议符合性测试概念进行了介绍，为 AFDX 协议符合性自动化测试软件的测试方案设计和测试方法设计提供理论基础。

第三章 软件需求分析与设计

软件需求分析和设计是软件开发过程中的一个关键部分，用于保障软件满足预期，有助于开发者开发高质量的软件系统。本章对 AFDX 端系统自动化测试软件进行需求分析和总体设计，从测试对象和测试内容出发，挖掘端系统自动化测试软件的功能性和非功能性需求。最后，基于测试方案进行软件架构设计，为软件实现奠定基础。

3.1 测试对象与测试操作流程介绍

3.1.1 测试对象

本文讨论的自动化测试对象是基于 FPGA 的 AFDX 端系统仿真卡^[48]，其硬件组成如表 3-1 所示。

表 3-1 AFDX 端系统仿真卡硬件组成

HKS664ES 芯片	集成 ARM922T 处理器，实现 ARINC664 协议规定的 AFDX 端系统的主要功能。
电源模块	提供+3.3V、+1.8V 电压。
时钟模块	提供系统时钟，PCI 时钟，RTC 时钟。
复位模块	用于系统上电复位、PCI 复位和 JTAG 复位。
SRAM 模块	用于存储系统配置表信息和底层传输软件。
RS232 模块	使用串行接口用于连接调试设备。
FLASH 模块	端系统的底层传输软件固化在 FLASH 中，系统上电后，程序被搬运至 SDRAM 中运行。
SDRAM 模块	接收发送数据缓存区。
PHY 芯片及网络模块	通过 2 路 RJ45 网口输出 AFDX 数据。

AFDX 端系统仿真卡功能与机载端系统基本一致，端系统的功能依靠端系统硬件模块和端系统底层传输软件协同实现。物理层与数据链路层主要由硬件 FPGA 实现；传输层通过传输软件实现，主要负责接口模块的初始化、发送以及接收；测试程序和驱动程序驻留在 Windows 主机中，测试软件通过 Windows PCI 接口以及驱动程序与端系统硬件进行交互，实现对端系统硬件的控制。

3.1.2 测试操作流程

在对软件进行需求分析之前，需要弄清 AFDX 端系统自动化测试的一般操作流程。在测试准备阶段，首先需要人工地将 AFDX 端系统插入主机并上电，在系统的设备管理

中加载设备驱动。然后测试软件调用驱动接口完成主机上端系统设备查找以及初始化。在测试配置阶段,软件对测试工程、测试用例、测试端和被测试端所加载的配置表进行配置。在测试进行阶段,测试软件进行测试用例的执行和测试过程日志数据的显示和测试进度的展示。在测试完成后,测试软件对测试过程所记录的数据进行分析,判断测试结果并自动地给出测试报告。

3.2 端系统协议符合性测试内容

AFDX 网络功能丰富,需要测试的项目也有很多,在编写自动化程序之前,通过对 AFDX 协议体系的研究,明确测试内容以及完成测试用例的设计和准备。

3.2.1 采样端口测试

AFDX 协议中对采样端口要求如下:

(1)每个采样消息接收时都有一个刷新指示,当读取采样消息时,如果读取的消息为新消息,刷新指示为真,反之为否。

(2)端系统的发送端和接收端能够容纳最大为 1518 字节的 VL 帧,即 VL 帧的负载最大为 1471 字节(帧头大小为 47 字节)。

(3)采样服务不能使用 IP 分片操作,即每条采样消息的长度都要小于或等于对应的 VL 中定义的有效消息长度的要求。所以,发送端系统能够处理应用发送超过 VL 配置最大消息长度的有效载荷消息的请求,当消息长度超过有效载荷长度,该消息直接被丢弃,消息刷新指示为否。

(4)采样服务能发送多播消息,多播消息是指一条消息发出后多个设备可以同时接收,是一对多的传输的消息。协议中定义的多播消息是基于 VL 的,当消息为多播消息,则所有指定 VL 的采样端口都会接收到该条消息。

为了测试端系统采样端口的协议符合性,首先我们对端系统采样端口发送消息进行测试。TE 命令 TR 分别发送消息长度超过 VL 的要求和消息长度不超过要求的消息,当消息长度正确时,TE 能正确接收到 TR 发送的每条消息,并且消息刷新指示为真,接收分片计数递增正确。当消息长度不正确时,TE 不能收到该消息,刷新指示为否,接收分片计数不增加。

其次对测试端系统采样端口接收消息进行测试。TE 向 TR 发送分别发送消息长度超过 VL 中的要求长度和长度不超过要求的消息,TE 能成功收到消息长度正确的消息,且消息刷新指示和数据帧接收计数正确变化。

最后,分别控制 TR 发送或接收多播消息以及 1518 字节长度的消息,通过判断消息刷新指示和对比端系统的数据发送计数和接收计数,判断端系统的正确性。设计的采样端口测试详细步骤如表 3-2 所示。

表 3-2 采样端口测试步骤

序号	测试步骤	期望结果
1	<p>TE 和 TR 配置发送 VL1 和接收 VL1 的允许最大 VL 帧长度为 147B(即最大消息长度为 100B), VL2 消息最大长度为 247B(即最大消息长度为 200B)。</p> <p>配置 TE 发送端口 1 和接收端口 1 使用 VL1, 端口类型为采样端口, 端口消息缓冲最大长度为 247B。配置发送端口 2 和接收端口 2 端口类型为采样端口使用 VL2, 消息缓冲最大长度为 247B。</p> <p>配置 TR 发送端口 3 为采样端口, IP 为多播 IP, 使用 VL3 发送, 配置 TE 接收端口 3 使用 VL3 接收。</p> <p>配置 TR 和 TE 的 VL4 消息最大长度 1518B, 发送端口 4 和接收端口 4 为采样端口, 使用 VL4。</p> <p>TR 配置除发送和接收的 IP 地址, 其他相似。</p>	TE 和 TR 配置成功。
2	TE 命令 TR 分别通过发送端口 1 和发送端口 2 各发送 100 条长度为 200B 的消息。	TR 收到命令。
3	TR 发送 TE 命令要求的消息。	TE 通过接收端口 2 正确收到 100 条长度为 200B 的消息, 且消息刷新指示为真。端口 1 未收到消息, 且消息刷新指示为否。
4	TE 命令 TR 从接收端口 1 和 2 接收消息。TE 使用发送端口 1 和 2, 向 TR 各发送 100 条长度为 200B 的消息。	TR 收到命令。TR 的接收端口 2 正确收到 100 条长度为 200B 的消息, 且消息刷新指示为真, 端口 1 未收到消息, 且消息刷新指示为否。
5	TE 命令 TR 使用端口 3 通过 VL3 发送 100 条采样多播消息。	TR 成功收到命令。
6	TR 使用端口 3 通过 VL3 发送 100 条多播消息。	TE 成功收到多播消息。
7	TE 命令 TR 通过端口 4 发送负载长度从 1 递增至 1471 的消息。	TR 收到命令, 成功发送消息, TE 成功接收到所有不同长度消息。
8	TE 命令 TR 接收消息, TE 发送负载长度从 1 递增至 1471 的消息。	TR 收到命令, 成功接收不同长度消息。

3.2.2 队列端口测试

AFDX 协议中对队列端口要求如下:

(1)队列服务在发送和接收时以先入先出(FIFO)的原则处理消息,在分片操作时,各个片段也应该按照次序发送,且分片接收时只有当整个消息被重组完成被交给应用。

(2)每一个队列服务的实例应该能管理长达 8K 字节的应用数据^[49]。

(3)消息的长度和相关 VL 要求长度不一致时,应使用 IP 分片和重组服务。消息的长度和相关 VL 长度一致时,不应使用 IP 分片和重组。

对于队列端口 FIFO 和支持 8K 字节消息长度测试,可以构造消息内容为递增数据的消息,通过对比数据发送和接收的正确性和完整性来验证。对于队列端口支持数据分片的测试,两个端口发送不同长度的消息,然后对比被测端系统和测试端系统的发送和接收数据帧计数器判断数据是否分片增加。设计的队列端口测试详细步骤如表 3-3 所示。

表 3-3 队列端口测试步骤

序号	测试步骤	期望结果
1	TE 和 TR 配置发送端口和接收端口 5 为队列端口,端口消息支持最大长度为 8K,使用 VL5。VL5 消息最大长度为 1518B(即最大消息负载长度为 1471B)。	TE 和 TR 配置加载成功。
2	TE 命令 TR 构造 100 条长度为 8K 字节,内容数递增数据的消息,使用端口 5 发送。	TR 收到指令成功发送消息,TE 成功接收到消息。TE 和 TR 发送计数和接收计数一致。
3	TE 命令 TR 接收消息。TE 构造 100 条长度为 8K 字节,内容数递增数据的消息,使用端口 5 发送。	TR 成功接收到消息。TE 和 TR 发送计数和接收计数一致。
4	TE 命令 TR 构造 100 条消息内容为数字 1~100 依次递增,长度为 1000 的消息按数字顺序使用端口 5 向 TE 发送。	TE 按顺序收到 100 条内容递增的消息。TE 和 TR 的发送计数和接收计数一致。
5	TE 命令 TR 接收消息。TE 构造 100 条内容为数字 1~100 依次递增的消息,按数字顺序向 TE 发送。	TR 接收到消息。TE 和 TR 的发送计数和接收计数一致。

3.2.3 SAP 端口测试

协议要求端系统能够作为一个服务访问点(Service Access Point, SAP),SAP 端口能够被用来在 AFDX 网络内通信,通过网关或路由器接入兼容网络,提供可与兼容网络通信的 UDP 服务,每个 UDP 服务访问点的实例都应该能够处理 8K 字节的数据。

对 SAP 端口通信测试,可以构造长度为 8KB 长度的消息并使用 SAP 端口发送,发送端和接收端发送消息数与分片发送数据计数比例成固定值。设计的 SAP 端口测试详细步骤如表 3-4 所示。

表 3-4 SAP 端口测试步骤

序号	测试步骤	期望结果
1	TE 和 TR 配置端口 6 为 SAP 端口,最大消息长度为 1518B,使用 VL6,VL6 消息最大长度为 1518B(即最大消息负载长度为 1471B)。	TE 和 TR 配置加载成功。
2	TE 命令 TR 构造 100 条长度为 8K 字节的消息,使用发送端口 6 向 TR 发送。	TR 成功接收指令,发送消息给 TE,TE 和 TR 发送和接收计数正确变化。
3	TE 成功收到完整消息。	接收到的消息内容与发送消息内容一致,发送帧数计数增加 60。
4	TE 命令 TR 接收消息。TE 构造 100 条长度为 8K 字节的消息,使用发送端口 6 向 TR 发送。	TR 收到完整消息,TE 和 TR 发送和接收计数正确变化。

3.2.4 冗余管理测试

端系统之间的通信覆盖在多个独立且冗余的网络,对于任意一个网络组件的失效,例如:一段链路或一台交换机的失效,数据流可以得到冗余保护^[50]。协议中冗余的方案是以每条虚拟链路为基础的,每个帧同时通过两个网络被发出。

在接收端采用“先到有效者胜出”的策略。对于每条 VL 可以自主决定是否采用冗余机制,为了简化接收端系统的算法,帧的冗余副本应该被几乎同时地传输,它们之间的最大时间差不应超过 0.5ms^[51]。

端系统冗余管理测试可以给 TE 和 TR 分别配置三条 VL,一条配置 AB 冗余,一条仅使用 A 网,一条仅使用 B 网,TE 和 TR 的消息能在正确的网络上发送和接收。设计的冗余管理测试详细步骤如表 3-5 所示。

表 3-5 冗余管理测试步骤

序号	测试步骤	期望结果
1	TR 和 TE 分别配置发送端口 7、8、9 分别使用 VL7、8、9,VL7 配置只使用 A 网,VL8 配置只使用 B 网,VL9 开启冗余,同时使用 AB 网。	TR 和 TE 配置成功。
2	TE 命令 TR 使用 VL7 和 VL8 先后发送 100 条消息。	TR 收到命令,TR 成功发送消息,TR 和 TE 的 A 网和 B 网

		计数分别先后增加 100。
3	TE 命令 TR 通过端口 7 和 8 接收消息。TE 使用 VL7 和 VL8 发送 100 条消息。	TR 收到命令，TR 和 TE 的 A 网和 B 网计数分别增加 100。
4	TE 命令 TR 使用 VL9 发送 100 条消息。	TR 收到命令，TR 成功发送消息，TR 和 TE 的 AB 网计数同时增加 100。
5	TE 命令 TR 通过端口 9 接收消息。TE 使用 VL7 和 VL8 发送 100 条消息。	TR 收到命令，TR 和 TE 的 AB 网计数同时增加 100。

3.2.5 时间同步测试

协议中并没有定义为在端系统上运行的各种应用程序提供公共时间参考的任何方法，但是保证端系统消息传输的时间完整性对于机飞行安全至关重要。测试对象 AFDX 端系统上实现了 RTC 时间同步模块。按功能划分将时间同步分为服务端和时间同步客户端，端系统可配置为客户端或服务端，服务端将不断向同一网络内广播 RTC 帧，客户端收到后，将根据收到的 RTC 帧同步自身的 RTC。

针对时间同步模块客户端模式测试，首先将 TR 配置为时间同步客户端，TE 配置为时间同步服务端。连续多次读取 TR 的 RTC，RTC 值依次增加。随后给 TE 设置新的 RTC 值，TE 的 RTC 将从新的 RTC 值开始递增。最后再次读取 TR 的 RTC 值，TR 的 RTC 值在设置后的 RTC 的基础上递增。时间同步服务端的测试与之类似。设计的时间同步测试详细步骤如表 3-6 所示。

表 3-6 时间同步测试步骤

序号	测试步骤	期望结果
1	配置 TE 为时间同步服务端，配置 TR 为时间同步客户端。	TR 和 TE 加载配置成功。
2	TE 命令 TR 连续读取 RTC10 次。	读取的 RTC 有序递增。
3	TE 修改 RTC 为新的 64 位的 RTC 值，TE 命令 TR 连续读取 RTC10 次。	TE 的 RTC 修改成功，读取的 RTC 在修改后值的基础上有序递增。
4	配置 TE 为时间同步客户端，配置 TR 为时间同步服务端。	TR 和 TE 加载配置成功。
5	TE 连续获取 TR 的 RTC 100 次。	TE 的 RTC 随 TR 变化。
6	TE 命令 TR 修改 RTC 为新的 64 位的 RTC 值。	TR 的 RTC 修改成功，RTC 在修改后值的基础上有序递增。
7	TE 获取 TR 的 RTC。	TE 的 RTC 随 TR 在新 RTC 值基础上变化。

3.2.6 端系统安全性测试

根据协议要求, AFDX 端系统可以处理最大 1518B 的网络帧, 即 AFDX 的消息负载不超过 1471B。队列端口和 SAP 端口可以管理长度不超过 8K 字节的消息。根据安全性测试的测试要求和测试的出发点, 针对协议要求边界进行测试。测试消息长度为 0 时、采样消息长度大于 1518B 时以及队列和 SAP 端口长度大于 8K 字节时, 端系统的处理结果。设计的安全性测试详细步骤如表 3-7 所示。

表 3-7 安全性测试步骤

序号	测试步骤	期望结果
1	配置 TE、TR 端口 11 为采样端口最大长度为 1518, 使用 VL11 最大消息负载长度 1518B。端口 12 为队列端口最大长度为 8K, 使用 VL12 最大消息负载为 1518B。端口 13 为 SAP 端口最大长度为 8K, 使用 VL13 最大消息负载 1518B。	TR 和 TE 配置成功。
2	TE 命令 TR 使用端口 11、12、13 分别发送空消息 100 次, 再发送 1518 字节数据 100 次, 8192 字节数据 100 次, 8193 字节数据 100 次。	TE 端口 11、12、13 收到 1518 字节数据。然后端口 12、13 收到 8192 字节数据。
3	TE 命令 TR 接收信息。TR 使用端口 11、12、13 分别发送空消息 100 次, 再发送 1518 字节数据 100 次, 8192 字节数据 100 次, 8193 字节数据 100 次。	TE 收到消息与发送消息相同。

3.2.7 端系统可靠性测试

可靠性测试时为了评估端系统在可用条件下, 保持系统功能可靠性的一种测试, 验证端系统产品质量是否达到预期目标。AFDX 端系统可靠性测试是检测端系统在长时间运行条件下, 三类端口的收发状态的稳定性。设计的可靠性测试详细步骤如表 3-8 所示。

表 3-8 可靠性测试步骤

序号	测试步骤	期望结果
1	配置 TE、TR 使用端口 14、15、16 分别为采样端口、队列端口、SAP 端口使用 VL14、15、16, VL 和端口最大消息长度为 1518。	TR 和 TE 配置成功。
2	TE 分别命令 TR 使用各端口持续发送 1518 字节消息。	TE 收到所有完整消息。
3	TE 命令 TR 接收消息, TE 使用各端口持续发送 1518 字节消息。	TR 收到所有完整消息。
4	TE 命令 TR 发送设备状态信息。	TR 状态正常。

3.3 软件需求分析

3.3.1 功能性需求

根据测试软件的测试方法和测试流程分析，自动化测试软件需要实现的功能有测试管理、端系统板卡管理、测试数据显示、测试配置表管理、测试报告生成功能。

(1)测试管理功能

用于管理测试工程和测试用例，支持用户新建测试工程，添加和管理测试用例，修改测试项执行顺序，执行测试用例。核心是实现对 AFDX 端系统协议符合性自动化测试过程的控制，方便用户进行测试过程启动和停止。

(2)端系统板卡管理功能

用于实现对工控机上的端系统硬件的查找和初始化，包括测试端系统和被测端系统。用户在界面能够直观地获取到端系统状态信息。

(3)测试数据显示功能

通过在测试控制软件界面向用户实时展示测试用例执行过程和测试进度。

(4)测试配置表管理功能

用于对端系统的配置表进行配置和修改，包括测试端配置表和被测试端配置表。

(5)测试报告生成功能

用于在端系统测试完成后生成测试报告，供用户确认端系统状态。

3.3.2 非功能性需求

软件的非功能性需求影响着软件是否能够持续稳定并高效的提供服务，AFDX 端系统的非功能需求如下：

(1)可靠性要求

软件的可靠性对于用户使用软件的体验感受和软件的使用效果至关重要，对于 AFDX 协议符合性软件来说在软件运行过程中不能出现严重漏洞导致软件程序崩溃，一旦出现问题，软件能保证数据完整性和快速的恢复能力。

(2)易用性要求

对于端系统测试软件，用户能够通过简洁的操作和清晰的信息显示完成端系统的测试准备工作，为后续的自动化测试的执行打下良好的基础。直观的界面设计能够符合用户的操作习惯，并且能够节约系统资源。

(3)扩展性要求

为了兼容不同的测试环境和测试需求，软件应该通过预留一些接口，方便软件升级扩展，以适应未来的业务变化和环境变化。

3.4 自动化测试软件设计

3.4.1 协议符合性测试方案

协议符合性测试是对端系统所实现的功能是否满足协议要求而进行的测试。为了实现 AFDX 端系统的协议符合性的自动化测试，我们需要将板卡硬件控制、测试配置表输入、测试参数输入、测试用例执行、测试结果验证等操作自动化。根据 TR-TE 测试策略，AFDX 端系统协议符合性测试方案如图 3-1 所示。

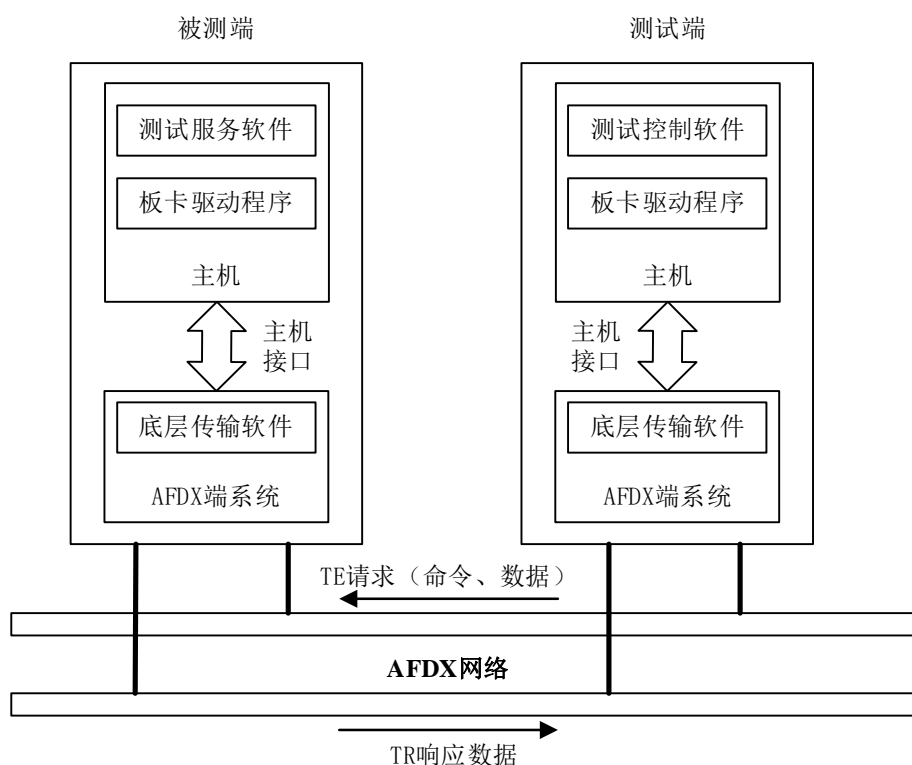


图 3-1 AFDX 端系统协议符合性测试方案

测试端包括测试控制软件、板卡驱动程序和测试控制端系统，被测端包括测试服务软件、板卡驱动程序和被测端系统。两个端系统的两个物理网络接口使用 AFDX 链路连接。端系统板卡通过 PCI 接口与主机相连。

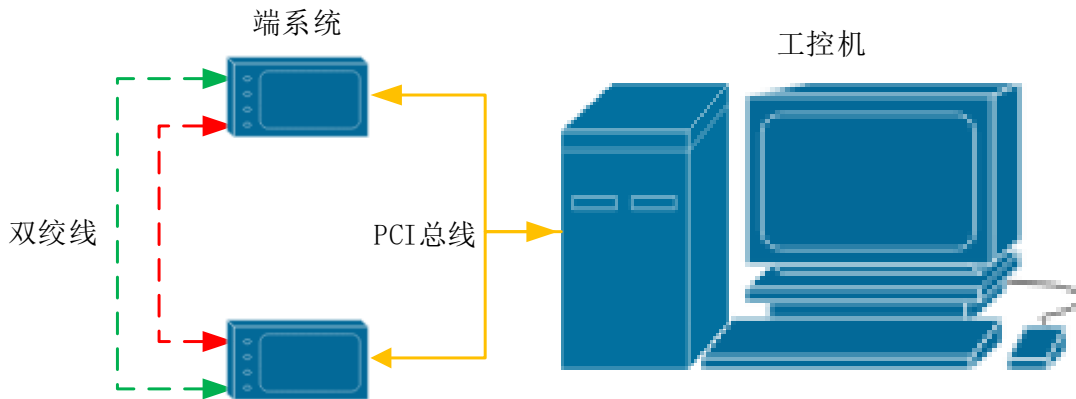
端系统仿真卡的驱动程序已由项目组其他同学实现，这里仅调用驱动程序 API。AFDX 端系统自动化测试软件包括测试控制软件和测试服务软件。测试控制软件和测试服务软件介绍如下：

测试控制软件是整个自动化测试软件的核心，负责对整个测试工作的自动执行与控制。测试控制软件根据 AFDX 端系统协议符合性自动化测试流程运行。首先，控制软件引导用户配置测试环境，提示用户检查板卡连接和安装驱动，随后测试控制软件进行端系统测试用例的管理并按照设计好的测试用例控制测试端发送测试指令，接收测试指令的响应，收集测试过程中产生的数据并记录到日志中。测试完成后对比验证测试结果，

最后自动生成测试报告供用户查看。

测试服务软件受到测试控制软件的控制，辅助测试控制软件完成测试工作。测试服务软件用于接收和解析测试控制软件发送的测试指令，根据指令控制被测端 AFDX 端系统执行对应功能并将执行结果反馈给测试控制软件。

根据测试方案搭建的端系统自动化测试硬件环境结构如图 3-2 所示，主要包括一台含多 PCI 接口的高性能工控机、测试 AFDX 端系统、被测 AFDX 端系统和百兆交叉线。



3-2 测试硬件环境示意图

3.4.2 自动化测试软件架构设计

根据自动化测试方法和自动化测试执行方案以及软件需求分析，自动化测试软件的总体层次结构如图 3-3 所示。

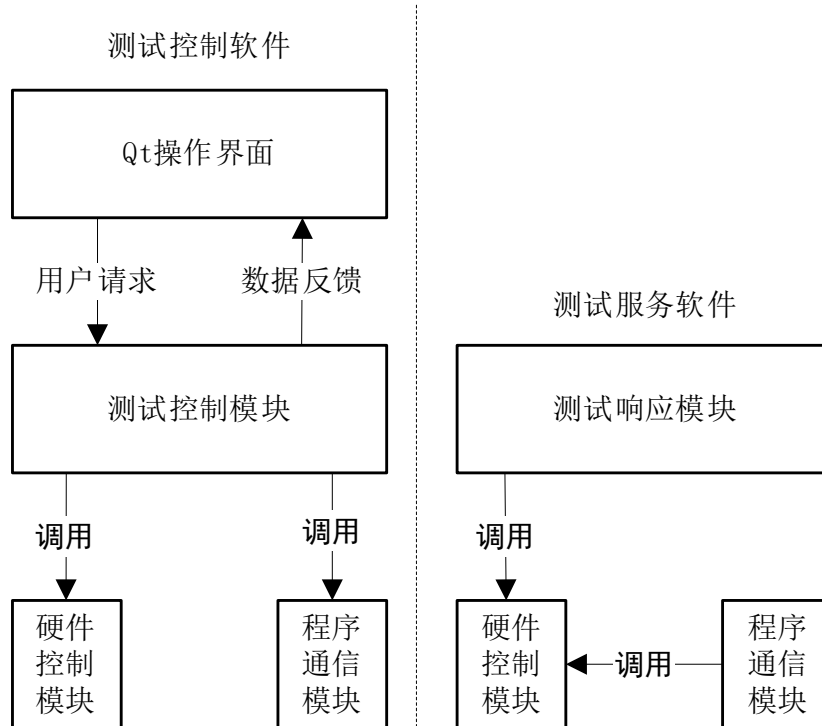


图 3-3 软件总体层次结构

测试控制软件运行在测试端，用于进行测试前的配置和测试过程中的控制和测试后测试报告的管理，一共分为三层。上层为用户层，简洁的用户界面方便测试人员的操作使用。中间层为测试控制模块，是整个测试软件的关键部分，负责按照用户选择好的测试用例和测试顺序执行自动化测试，以及实现端系统自动化测试流程的控制。下层为硬件控制模块和程序通信模块，被测试控制模块调用，硬件控制模块控制测试硬件发送测试命令和接收测试服务软件的反馈信息，测试控制软件的程序通信模块用于与测试服务软件通信，用于获取测试服务端的部分信息。

测试服务软件在 TR 端运行，用于测试过程中被测试端的管理，主要分为两层。上层为测试响应模块，通过调用硬件控制模块接收控制软件传输来的不同指令，测试响应模块执行不同的操作，并将结果返回给测试控制软件。测试响应模块的执行离不开下层硬件控制模块，硬件控制模块与 TE 端硬件控制模块功能基本相同，其用于控制被测试端的 AFDX 端系统。程序通信模块用于与测试控制软件通信，根据测试端请求调用硬件控制模块获取被测端系统的信息并返回给测试端。测试服务软件不需要程序界面，可以节约主机系统资源，提高测试效率。

3.5 本章小结

本章首先对测试对象和测试操作流程进行了介绍，然后介绍了 AFDX 端系统协议符合性测试内容，随后对端系统协议符合性自动化测试软件的需求分析进行说明，最后对协议符合性测试方案和自动化测试软件的架构设计进行了说明，为第四章端系统自动化测试软件的实现奠定基础。

第四章 端系统自动化测试软件实现

在第三章对端系统协议符合性自动化测试软件需求分析和架构设计的基础上，本章对从各个模块的角度端系统自动化测试软件的实现进行详细说明。

4.1 软件开发环境

AFDX 端系统自动化测试软件使用 C/S 结构，软件开发环境如表 4-1 所示。

表 4-1 软件开发环境

软件环境	开发语言	C、C++
	开发环境	MinGW 7.3.0 64-bit
	开发工具	Qt Creator 4.11.1、Qt Designer 5.14.2
	操作系统	Windows 10 IoT 企业版 LTSC
硬件环境	CPU	Intel(R) Core (TM) i7-8700
	RAM	16G
	GPU	GeForce GTX 1060 6G DDR5
	硬盘	500G

4.2 测试控制软件实现

根据软件的需求分析和设计，测试控制软件主要包括以下几个功能模块：板卡管理模块，测试管理模块，配置表管理模块，测试数据显示模块，测试报告管理模块，硬件控制模块和程序通信模块，测试控制软件模块划分如图 4-1 所示。

板卡管理负责发现主机上的板卡，用户选择端系统板卡的角色，可以配置为测试控制端（TE）或者被测试端（TR）。

测试管理用于管理端系统协议符合性测试用例和控制自动化测试执行过程。

配置表管理用于编辑和解析测试过程中端系统所加载的二进制 BIN 格式配置表文件，包括 TE 端配置表和 TR 端配置表。

测试数据显示主要是进行测试过程中的日志，包括 TR 的测试日志和 TE 的测试日志，显示测试过程的总进度。

测试报告管理主要是根据测试过程产生的数据生成相应的测试报告。

硬件控制模块是实现测试过程的基础，通过与端系统仿真卡驱动程序交互，进行 AFDX 端系统板卡的控制。

端系统通信模块用于获取 TR 端的其他部分信息，包括测试过程日志、被测试板卡的状态。

除此之外，针对 TE 和 TR 之间的请求和响应，设计用于传输数据或命令的数据结构。最后，按照用户交互友好原则，设计一套简洁大方、易于操作的用户界面。

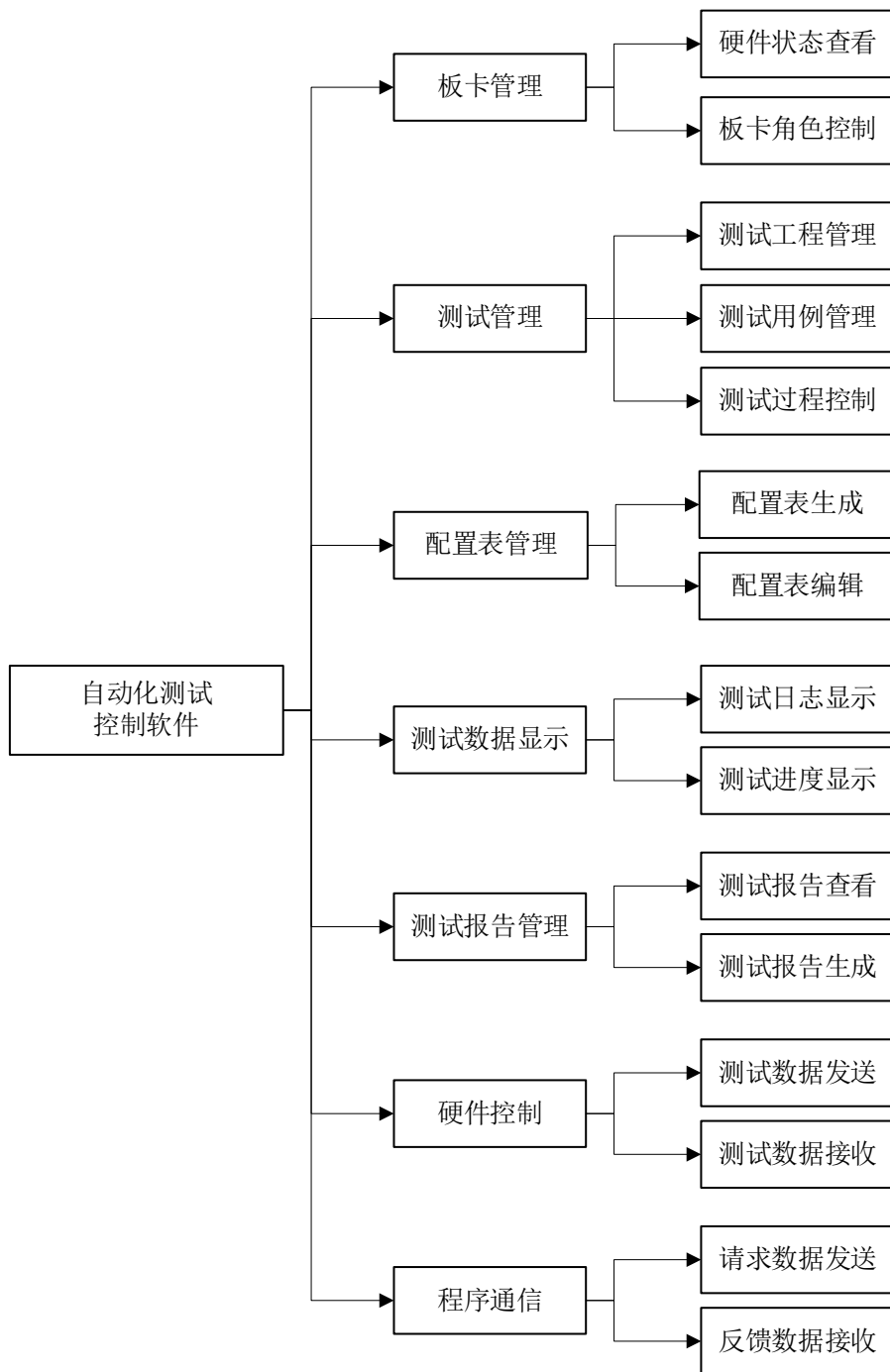


图 4-1 控制软件功能模块设计

4.2.1 端系统硬件控制模块

硬件控制模块是实现端系统测试的基础。根据 AFDX 端系统数据手册和硬件手册可知，AFDX 在端系统的硬件的 PCI 接口中定义了四个空间，寄存器类型的 BAR0 空间，FIFO 类型的 BAR1 空间，DMA 通道空间和 PCI 接口配置空间。AFDX 端系统硬件控制

模块通过调用 AFDX 端系统 PCI 板卡驱动程序接口对 PCI 接口的 BAR 空间读写交互方式实现。

端系统硬件控制模块按功能划分，分为设备管理、数据处理以及通用接口功能。设备管理功能包括：设备查找、设备启动、设备初始化、配置表加载、端口配置等。设备数据处理功能包括：采样端口数据发送和接收、队列端口数据发送和接收、SAP 端口数据发送和接收。通用接口模块主要通过 BAR 空间的读写实现板卡状态的获取和控制。硬件控制模块的一般执行流程如图 4-2 所示。

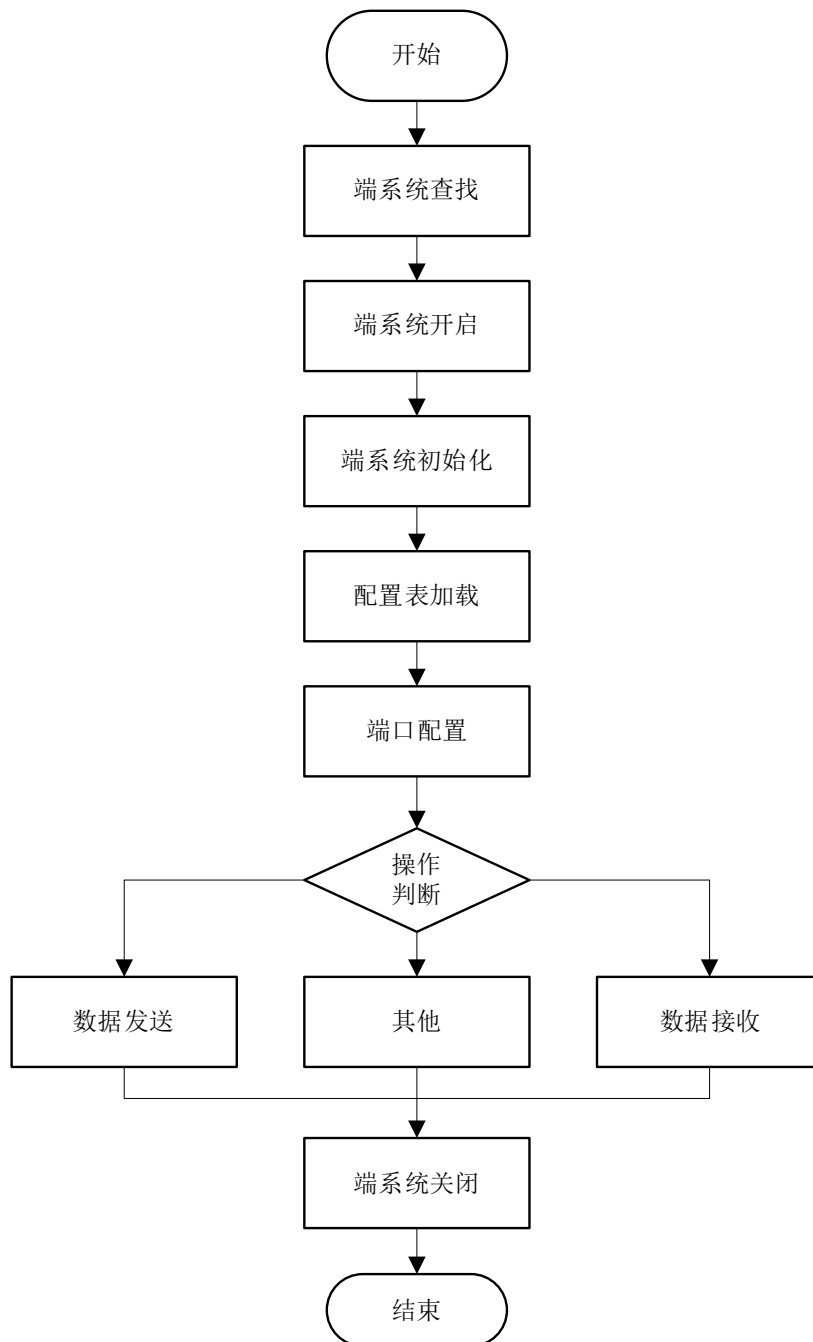


图 4-2 硬件控制模块执行流程

硬件控制模块中定义了设备接口函数，每个函数接口都有一个返回值，通过返回值可以获得函数执行的状态，部分函数的返回值预定义如表 4-2。

表 4-2 部分函数的返回值预定义

定义	返回码
AFDX_OPEN_DEVICE_SUCCESSFUL	0 打开设备成功
AFDX_LOADER_SUCCESSFUL	0 加载配置表成功
AFDX_SUCCESSFUL	0 成功
AFDX_ESID_FAIL	1 ES 设备 ID 错误
AFDX_OPEN_DEVICE_FAIL	2 打开设备失败
AFDX_LOADER_FAIL	3 加载配置表失败
AFDX_NO_FIND_PORT	4 查询端口失败
AFDX_NO_CREATE_PORT	5 无创建的端口
AFDX_CONFIG_FAIL	6 创建失败
AFDX_MODE_FAIL	7 模式错误
AFDX_SEND_DATA_TOO_LARGE	8 发送数据超长
AFDX_SEND_NO_DATA	9 无数据发送
AFDX_TXBUF_OVERFLOW	10 发送缓冲区溢出
AFDX_NO_DATA_RECEIVED	11 无数据接收
AFDX_RECEIVE_DATA_TOO_LARGE	12 接收数据超长
AFDX_ALLOC_MEMORY_FAIL	13 申请空间失败
AFDX_CREATE_PORT_FAIL	14 创建端口失败
AFDX_PCI_INIT_FAIL	15 初始化 PCI 失败
AFDX_CLOSE_DEVICE_FAIL	16 关闭设备失败
AFDX_PORT_NUM_OVERFLOW	17 端口数目溢出
AFDX_CFG_FILELEN_ERROR	18 配置文件长度错误

下面介绍硬件控制模块的各个功能的函数接口。

(1) 设备管理

a. 设备查找功能

测试应用程序需要给不同的端系统分配不同的角色，所以需要查找主机中存在多少设备。首先通过调用一系列 windows 库函数，获得设备句柄。SetupDiGetClassDevs 函数获取设备连接信息，其返回一个包含本机设备上所有被请求的设备信息的设备信息集句柄。SetupDiEnumDeviceInterfaces 枚举信息集中设备接口，返回不同设备接口的 SP_DEVICE_INTERFACE_DATA 结构。SetupDiGetDeviceInterfaceDetail 获取设备接口相关详细信息。获得设备句柄后获取设备的 ID，最后返回设备数目并删除信息集，释

放关联内存。函数定义如表 4-3 所示。

表 4-3 设备查找函数

函数名称:	AFDX_DEVICE_HANDLE AfdxFindDevices(int *cardcnt)
函数说明:	查找设备
输入参数:	无
输出参数:	设备句柄
返回值:	设备数目

b. 打开设备

当 Windows 系统启动时, BIOS 会枚举 PCI 设备并为每个设备分配地址, 并写入 PCI 设备配置空间。内核驱动程序完成设备识别, 获取设备配置空间信息, 注册中断函数, 完成硬件基本初始化, 将 BAR 空间映射到测试控制软件的内存。测试控制模块调用一系列库函数获得设备信息, 调用 CreatFile 库函数打开设备并获得设备句柄, 将设备打开标志位至 1。随后根据驱动 IRP 定义, 调用 DeviceIoControlAPI 函数获得板卡信息。函数定义如表 4-4 所示。

表 4-4 打开设备函数

函数名称:	U32 AfdxOpenDevice (U32 esid)
函数说明:	打开指定的 ES 设备。
输入参数:	esid:设备号。
输出参数:	无。
返回值:	0:成功; 非 0:失败。

c. 配置表加载

要实现端系统的收发消息需要给端系统加载配置表。首先, 需要对二进制 Bin 文件进行解析。使用 fopen_s 函数打开文件, 再调用 fread 函数读取文件, 其次进行大小端转换(端系统内为小端存储, Windows 系统为大端存储), 随后根据配置表的大小和偏移, 将配置表解析到配置表结构体中。最后, 将配置表拷贝至 PCI 接口中, 发送中断, 板卡加载配置表, 并修改配置表加载标识。加载的配置表包括: 端系统配置表、发送端口配置表、接收端口配置表、发送 VL 配置表、接收 VL 配置、分区配置表等。函数定义如表 4-5 所示。

表 4-5 配置表加载函数

函数名称:	U32 AfdxLoadConfig (AFDX_CFG *pconfig, U32 esid)
函数说明:	端系统配置表加载。
输入参数:	esid:设备号 Pconfig:配置表结构体指针

输出参数：无。
返回值： 0:成功； 非 0:失败。

d.端口配置

当配置表配置成功后，端系统为每个发送端口和接收端口在 BAR 空间中开辟一段独立缓冲空间用于测试程序和硬件板卡间发送和接收端口的相关数据的交互。首先根据配置表枚举不同类型的端口，然后按类型分别读取端口的是否创建标志，当创建标志为否时，根据配置表中定义的消息长度和消息深度分配不同的缓冲空间长度。端口配置包括采样端口配置、队列端口配置、SAP 端口配置。函数定义如表 4-6 所示。

表 4-6 创建端口函数

函数名称：U32 AfdxCreatePort(U32 esid)
函数说明：枚举指定端系统的端口并配置端口。
输入参数：esid:设备号。
输出参数：无。
返回值： 0:成功； 非 0:失败。

e.设备关闭

当测试完成，关闭 AFDX 端系统。硬件控制模块首先判断设备是否占用，如在占用，先停止设备。未占用则调用函数关闭句柄。设备打开标志位置 0，设备不可访问。函数定义如表 4-7 所示。

表 4-7 关闭设备函数

函数名称：U32 AfdxCloseDevice(U32 esid)
函数说明：选择指定设备并关闭
输入参数：esid :设备号。
输出参数：无。
返回值： 0:成功； 非 0:失败。

(2)数据处理

数据处理包括采样端口的数据发送和接收、队列端口的数据的发送与接收、SAP 端口数据的发送与接收。

消息发送时，测试控制程序按照测试用例生成相应的消息数据，调用硬件控制模块发送数据接口发送消息，函数依次检查设备是否打开，配置表是否已加载，端口号是否正确，消息大小是否正确，端口是否创建，端口缓冲区是否有空闲，所有检查通过后，将输入拷贝到配置端口时专门开辟的缓存中，最后修改数据标识。底层软件通过数据标

识检查到数据，底层传输软件对消息进行协议栈处理和检查，再将消息通过 MAC 发送。

消息接收时，硬件先收到消息数据，硬件和底层软件配合，对消息进行初步检查，包括 SN 号检查、CRC 校验、协议栈检查等等，检查通过后数据放入配置的缓存中。测试控制程序调用硬件控制模块接口接收消息，函数检查设备是否打开，配置表是否加载，端口号是否正确，数据是否正确，消息标识是否正确等。如果所有检查无误，将消息传递给测试控制程序，最后修改数据处理标识。以队列消息处理为例，其发送和接收消息函数定义如表 4-8 所示。

表 4-8 发送和接收消息函数

函数名称: U32 AfdxQueueSend (U32 esid, U32 portid, char *data, U32 datalen)
函数说明: 通过队列端口发送消息
输入参数: esid: 设备 id; portid: 端口序号; *data: 发送数据指针; datalen: 发送数据长度;
输出参数: 无
返回值: 0 发送成功; 非 0 发送失败。
函数名称: U32 AfdxQueueRecv(U32 esid, U32 portid, char *data,U32 *datalen)
函数说明: 通过队列端口接收消息
输入参数: esid: 设备号; portMapId: 端口序号;
输出参数: *data: 接收数据指针; datalen: 接收数据长度; newdata: 新旧数据标识;
返回值: 0 接收成功; 非 0 接收成功。

4.1.3 通用接口

驱动程序将硬件的寄存器的基地址进行内存映射，映射为测试程序的一个虚拟地址，硬件控制模块通过访问虚拟地址，实现对硬件的寄存器的读写，实现通用接口功能。

通用接口包括获取 AFDX 端系统状态，读写 RTC 时钟等通用功能。端系统硬件手册中定义了不同的配置信息在 PCI 接口的定义的地址偏移，通用接口通过驱动程序按手册中定义的地址对 PCI 接口 BAR 空间的不同位置读写控制信息实现。读写 Bar 空间的函数定义如表 4-9 所示。

表 4-9 读写 Bar 空间函数

void AfdxPciBar0Write(unsigned int offset, unsigned int data,int width)	写 Bar0 空间
void AfdxPciBar0Read(unsigned int offset, ,int width)	读 Bar0 空间
void AfdxPciBar1Write(unsigned int offset, unsigned int data,int width)	写 Bar1 空间
void AfdxPciBar1Read(unsigned int offset, ,int width)	读 Bar1 空间

以获取端系统的配置表版本为例，首先判断设备号是否存在，设备是否打开，当判断正确时，读取 Bar0 空间的配置表版本寄存器。其函数定义如表 4-10 所示。

表 4-10 获取端系统的配置表版本

函数名称: U16 AfdxGetConfigVer(U32 esid, char *escfgver)
--

函数说明：获取配置表版本号
输入参数：esid：设备号
输出参数：escfgver：配置表版本号
返回值：0:成功，非0:失败

还有其他通用接口函数如下。

ReadRtcTime：获取 RTC 时钟

WriteRtcTime：写入 RTC 时钟

ESGetOnline：获得设备的在线状态

GetBitResult：获取设备自检结果

RestDevice：设备复位

4.2.2 端系统板卡管理模块

端系统管理模块用于显示端系统的状态信息，以及选择端系统的角色。端系统设备管理处理流程如图 4-3 所示。

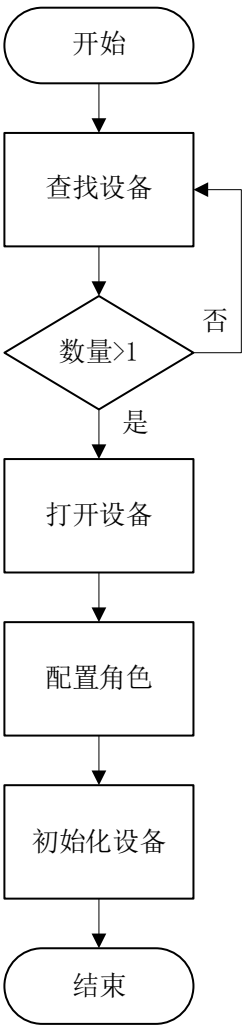


图 4-3 板卡管理流程

(1)状态信息显示

当板卡插入工控机的 PCI 槽中，启动测试控制软件。首先软件调用硬件控制模块接口进行板卡的查询，获得工控机上已插入端系统仿真卡的数量。用户使用测试控制软件时点击端系统管理按钮，测试软件调用接口获取端系统的状态包括端系统软件版本、端系统的驱动版本、端系统的自检结果等信息，获得的端系统的状态信息通过用户界面展示给用户。

(2)端系统角色选择

当测试软件检测到所在主机上有两张即以上端系统仿真卡时，端系统可以对两张端系统的角色模式进行配置，配置为 TR 或者 TE，一旦其中一张选择为 TE 或 TR，那么必须再选择一张 TR 或 TE。当检测到主机上只有一张端系统仿真卡时，则不能为其分配角色。

4.2.3 测试管理模块

测试管理模块为 AFDX 协议符合性自动化测试系统提供整个测试用例和测试执行控制的管理，是 AFDX 端系统协议符合性自动化测试的核心。测试管理流程如图 4-4 所示。

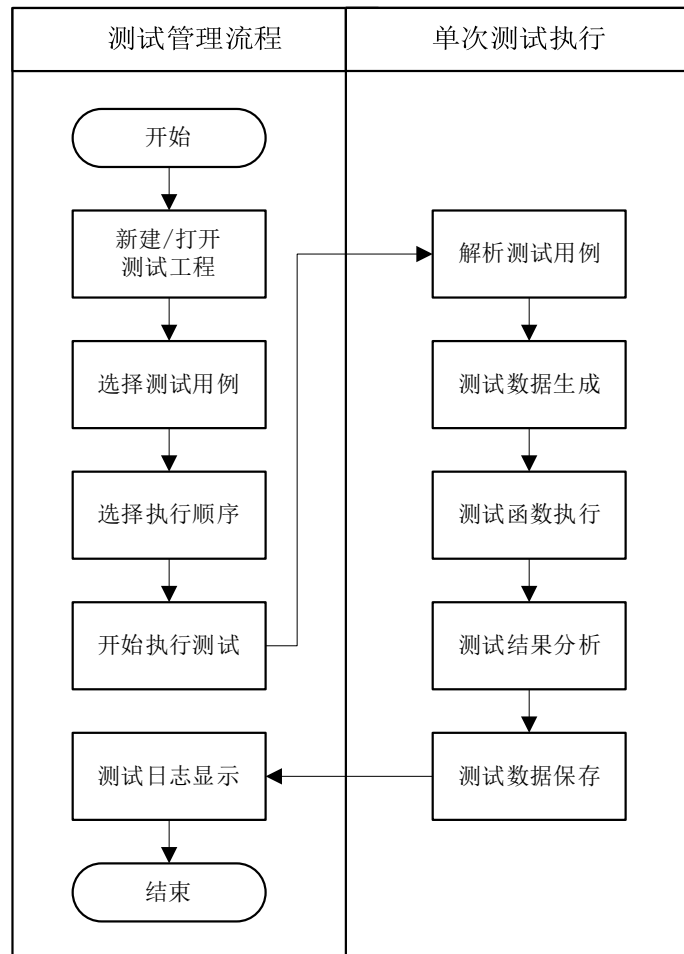


图 4-4 测试管理流程图

软件启动，用户配置完成端系统后，使用交叉线连接测试端系统和被测端系统。用户点击新建测试工程按钮，创建新的测试工程，为新创建的测试工程的命名并选择测试工程所在目录路径。完成新建后用户在测试管理中用户勾选需要执行的测试用例，并且通过上移和下移，用户可以调整测试用例的执行顺序，完成后保存。随后用户点击开始测试，软件下发测试执行命令，软件按照用户指定的测试用例和用例顺序执行。

自动化测试执行的过程中，软件界面为用户提供测试用例执行的输出信息，并显示测试过程执行进度。测试过程中，如果用户需要取消测试，可以点击停止测试按钮，自动化测试软件将等待当前测试用例完成后停止测试。在单个测试用例的自动化测试过程中，一般包括了测试用例解析，测试数据生成、硬件执行测试、测试结果分析、测试数据保存等。

测试控制软件在执行测试时，需要向 TR 发送测试指令，控制 TR 执行测试并返回响应，实现测试自动化。本文在数据帧的 payload 部分自定义一个通信数据结构，结构如图 4-5 所示。

目的地址	源地址	类型	IP头	UDP头	AFDX Payload				
					Cmd	Len	Port	Count	Payload

图 4-5 自定义通信数据结构

其中，Cmd 为指令段为 4 字节，Len 为长度段为 2 字节，标识发送消息长度，Port 为端口段为 2 字节，标识发送消息端口号，Count 为计数段为 2 字节，标识发送消息次数，总长度固定为 10 字节。测试过程中，控制软件将根据测试用例，将测试命令填入自定义结构中。如果需要传递其他数据信息，则将数据填入 Payload 段中。最后，本文进行了命令定义，部分定义如表 4-11 所示。

表 4-11 自定义指令

指令头 高字节数据 (2 字节)	指令头 低字节数据 (2 字节)	指令名称	指令说明	返回值
0x0001 发送 AFDX 消息	0x0001	SampTx	发送采样消息	发送确认
	0x0002	QueueTx	发送队列消息	发送确认
	0x0003	SAPTx	发送 SAP 消息	发送确认
0x0002 配置指令	0x0001	NetA	配置 A 网	无
	0x0002	NetB	配置 B 网	无
	0x0003	NetAB	配置 AB 网	无

	0x0004	Reset	软复位	复位完成
	0x0005	CleanCount	清空计数	清除完成
	0x0006	SetRTC	配置 RTC	无
0x0003 获取状态指令	0x0001	GetSoftVer	获取软件版本	版本号
	0x0002	GetCfgVer	获取配置表版本	版本号
	0x0003	GetStatus	获取运行状态	运行状态
	0x0004	GetTime	获取运行时间	运行时间
	0x0005	GetTxIPCount	获取发送 IP 包计数	发送 IP 包 总计数
	0x0006	GetTxUDPCount	获取发送 UDP 包计数	发送 UDP 包 总计数
	0x0007	GetRxIPCount	获取接收 IP 包计数	接收 IP 包 总计数
	0x0008	GetRxUDPCount	获取接收 UDP 包计数	接收 UDP 包 总计数
	0x0009	GetARxCount	获取 A 端口接收计数	A 端口接收 数据包计数
	0x000a	GetACRCErr	获取 A 端口 CRC 错误计数	A 端口 CRC 错误计数
	0x000b	GetAVLErr	获取 A 端口 VLid 错误计数	A 端口 VLid 错误计数
	0x000c	GetBRxCount	获取 B 端口接收计数	B 端口接收 数据包计数
	0x000d	GetBCRCErr	获取 B 端口 CRC 错误计数	B 端口 CRC 错误计数
	0x000e	GetBVLErr	获取 B 端口 VLid 错误计数	B 端口 VLid 错误计数

4.2.4 配置表管理模块

配置表是 AFDX 端系统发送确定性消息的必要条件，只有给端系统加载配置表完成后，才能进行后续测试阶段测试消息数据的发送和接收。AFDX 的配置表有固定的格式定义，配置表管理模块可以对端系统的配置表二进制文件进行解析并显示在用户图形界面，用户通过配置表管理模块方便的修改配置表中的字段并保存。配置表管理模块处理流程如图 4-6 所示。

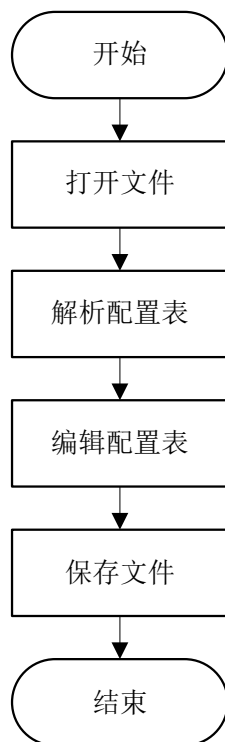


图 4-6 配置表管理模块流程

解析配置表是配置表管理模块的关键，Qt 提供 `QDataStream` 类，通过 `readRawData` 函数可以很方便的按指定字节长度将文件解析。例如读取配置表的版本代码如下。

```
QFile bin(QDir::currentPath()+"/TE.bin");
bin.open(QIODevice::ReadOnly);           //打开文件
QDataStream out(&bin);                   //文件序列化
out.readRawData(cfgver,16);              //以源格式读取 16 字节
```

4.2.5 测试数据显示模块

数据显示模块用于在用户界面中显示测试执行过程产生的数据，主要是测试过程产生的日志，包括 TR 的日志和 TE 部分的日志。这里采用第三方库 `Log4Qt` 实现日志的记录和输出。

`Log4Qt` 是一个用于记录应用程序日志到各种输出目标（例如文件、控制台和 `syslog`）的 C++ 库，是流行的 Java 日志库 `log4j` 的一个移植版本，提供类似的 API 和配置文件格式。

`Log4Qt` 主要由三个组件组成，记录器(logger)、附加器(appender)、布局(layout)。记录器用于在代码中采集输出消息，并可以配置将消息记录到特定的附加器。附加器负责将日志消息写入各种输出目标，如控制台，文件等。布局指定日志消息的格式，包括时间、日期、线程等。`Log4Qt` 支持多源输出，且线程安全，通过统一的配置文件实现日志的简单定制。`Log4Qt` 部分配置文件代码如图 4-7 所示。

```

1 # 定义日志输出的格式
2 log4j.appender.console.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} [%-5p] %c{1}:%L - %m%
3 |
4 # 将日志输出到控制台
5 log4j.appender.console=org.apache.log4j.ConsoleAppender
6 log4j.appender.console.target=System.out
7 log4j.appender.console.layout=org.apache.log4j.PatternLayout
8 log4j.appender.console.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} [%-5p] %c{1}:%L - %m%
9
10 # 将日志输出到文件
11 log4j.appender.file=org.apache.log4j.RollingFileAppender
12 log4j.appender.file.File=logs/application.log
13 log4j.appender.file.MaxFileSize=10MB
14 log4j.appender.file.MaxBackupIndex=10
15 log4j.appender.file.layout=org.apache.log4j.PatternLayout
16 log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} [%-5p] %c{1}:%L - %m%n
17
18 # 设置日志级别
19 log4j.rootLogger=DEBUG, console, file

```

图 4-7 Log4Qt 配置文件

这里配置将日志信息输出到文件和控制台，这里再通过 QFile 和定时器读取不断刷新的文件，将内容实时显示到 QTextBrowser 组件。

测试数据显示模块还包括测试进度的显示功能，通过 QProgressBar 类生成测试进度条并将其显示在测试控制程序界面。

4.2.6 测试报告管理模块

测试报告是为记录测试结果而撰写的文档，提供简明的测试结果，方便用户了解端系统的测试状态和质量状况，为系统开发人员提供参考。测试报告管理模块在测试过程结束后，通过分析测试过程中产生的数据，生成用户测试报告。测试报告中主要包含以下内容。

(1)引言。提供测试环境的主要信息。包括测试主机硬件信息、主机系统版本、硬件板卡 id、端系统底层软件版本、驱动版本号、配置表版本号等。

(2)测试概述。提供对测试执行的总体描述。包括测试的持续总时间、测试项执行时间等基本信息。

(3)测试详细结果。提供每个测试用例的详细结果，包括测试用例的名称和编号、测试通过的用例数、失败的用例数、测试通过率、测试失败的位置、以及失败原因概要等。

(4)测试结论。为测试人员提供测试结果的总结，说明端系统测试的功能是否符合协议要求。

测试报告管理模块提供测试报告生成以及提供测试报告的打印功能，通过 Qt 中 QPrinter、QPrintDialog、QFile 和 QFileDialog 模块，用户通过点击用户界面上的打印按钮，选择报告文件，Qt 模块调用 Windows 系统的打印接口，完成测试报告文件的打印。

4.2.7 程序通信模块

程序通信模块作为AFDX自动化测试的辅助程序，TE使用程序通信模块获取TR端测试开始前的状态信息，当AFDX链路失效，测试控制软件通过程序通信模块与测试服务程序通信。程序间通信采用客户端-服务端模式（Client-Server Model），客户端向服务器发出请求，并等待服务器响应。服务器在接收到请求后，执行相应的操作，并将结果返回给客户端。这里TE作为客户端，TR作为服务端。程序间通信有多种方式，例如管道、消息队列、共享内存等方式，本文采用Socket实现，Socket通信实现快捷方便，适用多种平台，若自动化测试环境变化，例如将TR和TE分别放置在两台主机上，则可通过简单修改配置，实现自动化测试软件对环境的适配。

Socket是一组网络程序通信接口，对TCP和UDP通信进行了封装。Socket通信分为了两种，一种是流式Socket（TCP Socket），使用TCP协议，面向连接的数据传输。另一种是数据报Socket（UDP Socket），使用UDP协议进行通信，数据传输是无连接的，不需要建立连接，数据包也不需要按顺序到达。

本文的测试环境中TR和TE处于同一主机，所以使用本地流式Socket，即采用TCP协议实现。因为测试过程中需要传输的数据量不大，通过三次握手建立连接后，数据的传输是可靠的，TCP协议保证了数据传输的有序性，数据包会按照发送的顺序到达接收端，流式Socket交互流程如图4-8所示。

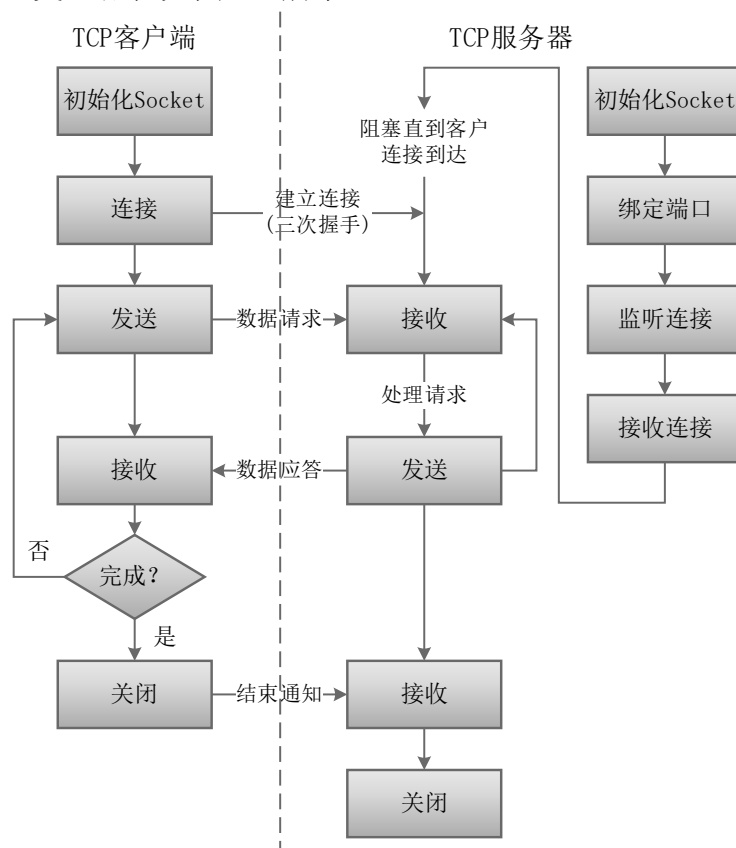


图 4-8 流式 Socket 交互流程

4.2.8 软件图形界面

测试控制软件的界面设计如图 4-9 所示，包括了测试菜单区、功能菜单区、测试状态栏、测试项管理区、测试日志显示区。

功能菜单区主要是板卡管理、测试管理、配置管理、报告管理、关于和帮助的按钮区。

测试菜单区主要是新建、打开、保存测试工程的管理的按钮区以及下方测试过程控制包括开始测试、停止测试、测试报告的按钮。

测试状态栏主要是用于显示软件欢迎词、以及显示测试进行进度的进度条。

测试项目管理区主要用于选择单次测试过程中测试项的选择。

测试日志显示区用于测试过程中 TR 和 TE 产生的测试日志信息。



图 4-9 软件图形界面

4.3 测试服务软件实现

测试服务软件作为测试系统重要组成部分，用于被测端系统的控制和管理，测试服务程序受到测试控制软件的控制，根据测试控制软件发送的不同指令，测试服务软件执行不同的操作，并将操作结果返回给测试控制软件。根据软件总体设计，测试服务软件包括测试响应模块、硬件控制模块和程序通信模块。

测试响应模块对从 AFDX 链路收到 TE 发过来的二进制格式的指令消息进行处理，解析收到的指令，其次根据不同的命令类型调用硬件控制模块执行相应的操作，最后再将执行的结果通过硬件控制模块返回测试控制软件。服务软件执行流程如图 4-10 所示。

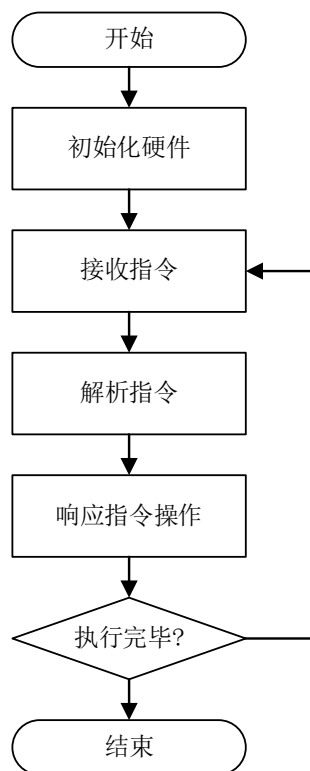


图 4-10 测试服务软件执行流程

测试服务软件的硬件控制模块与测试控制软件的功能相同，通过驱动程序接口，实现对被测端系统的板卡硬件控制。

测试服务软件的程序通信模块与测试控制软件不同，实现为服务端，根据客户端的请求，调用硬件控制模块获取被测端的信息并返回给测试端。测试服务软件程序通信模块用 QT 实现 TCP 服务端的主要步骤如下：

- (1)开启服务器端口服务，创建 `QTcpServer` 对象，使用 `listen()`方法监听指定的端口。
- (2)服务端开启监听后，客户端可通过本地 IP 和端口连接到该服务端，服务端通过 `newConnection` 信号来判断是否有客户端请求连接。
- (3)当客户端与服务端建立连接，服务端将会产生一个套接字对象用于与客户端通信，使用 `nextPendingConnection` 函数的返回值获取这个 `QTcpSocket` 对象。
- (4)最后通过监听 `readyRead` 信号，通过 `read` 函数读取客户端传来的指令，再通过 `write` 函数，将指令响应的结果返回给客户端。
- (5)当通信结束，客户端发出 `disconnected` 信号，通过 `disconnectFromHost` 函数关闭 Socket 连接。

4.4 本章小结

本章在第三章的软件需求分析与设计的基础上，对端系统协议符合性自动化测试软件进行了实现。分别介绍了测试控制软件和测试服务软件，着重介绍了测试控制软件中端系统硬件控制模块、板卡管理模块、测试管理模块、用户图形界面。

第五章 软件测试与分析

软件测试是软件质量保证的一个重要部分，本章节针对软件的各个模块功能的正确性进行了验证，并将自动化测试与人工测试方式进行比较。

5.1 测试环境搭建

AFDX 端系统符合性自动化测试软件的功能验证的硬件和软件组成如表 5-1 所示。

表 5-1 端系统测试环境

名称	说明
IPC-610L	工控机
XT-ES-PCI	端系统板卡两张
480904	双绞线两根
Jlink	芯片仿真器
Wireshark	网络抓包工具
ARM Developer Suite	ARM 开发软件包
Qt5.14.2	软件开发环境
Mingw	软件编译环境
AfdxTransportSoft	端系统底层传输软件
AfdxEmulatorDriver	端系统驱动软件
Ultra Editor	二进制文本编辑器

在进行端系统测试软件启动前，需要完成相关的测试准备工作。

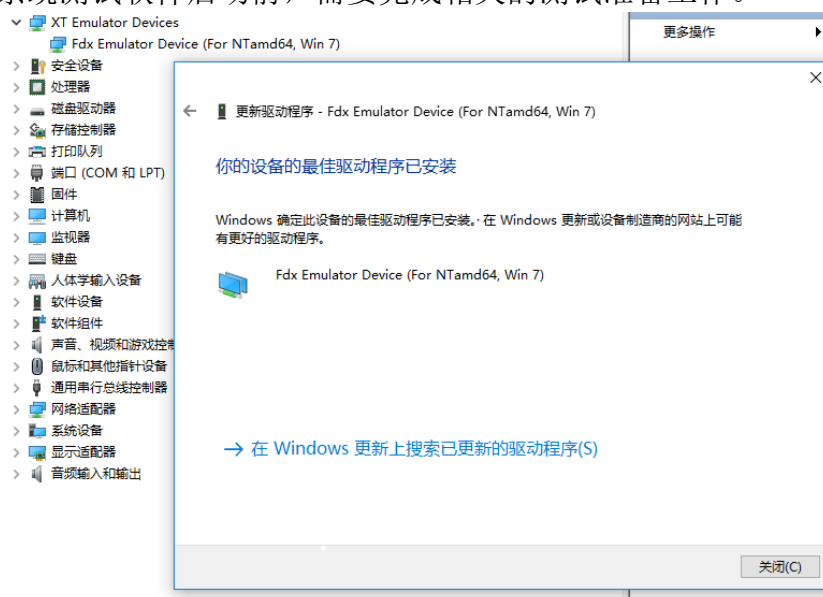


图 5-1 驱动安装示意图

首先在断电状态将板卡插入工控主机，接着上电，使用烧录软件和仿真器将端系统底层传输软件烧录进端系统板卡烧录完成后重启电脑。随后通过设备管理器检查电脑是否正确识别端系统板卡，当系统识别到未知设备，点击更新设备驱动程序，选择对应的inf文件依次安装两个端系统的驱动，驱动安装结果如图 5-1 所示。最后再次打开系统设备管理器查看设备信息，当信息正确且无感叹号表明设备正常，然后使用交叉双绞线连接端系统的网口。

5.2 软件功能测试

5.2.1 硬件控制模块

硬件控制模块的测试通过编写测试代码进行，测试硬件控制模块中的函数对板卡的控制是否生效。首先将 Jlink 仿真器连接端系统板卡，随后打开 ADS 软件并配置好软件环境。其次单步运行测试代码执行相应的函数，然后通过 Jlink 和 ADS 软件读取板卡内存空间，最后根据读取的数据判断结果是否与预期结果一致。硬件控制模块测试记录如表 5-2 所示。

表 5-2 硬件控制模块测试记录表

测试内容	预期结果	测试结果
当主机插入 2 张固化过软件的端系统板卡且安装驱动完成，AfdxFindDevices 函数正确返回板卡数。	函数返回板卡数量为 2。	通过
当主机查找到两张板卡，通过调用函数 AfdxOpenDevice 获得设备句柄。	函数返回正确设备句柄。	通过
找到设备后使用 AfdxLoadConfig 函数将二进制配置表加载到端系统。	配置表文件成功加载，且板卡内部为小端存储。	通过
加载配置表完成，AfdxCreatePort 函数创建端口。	创建端口成功，板卡内部端口信息区更新成功。	通过
Bar 空间读写测试，通过函数读取和写入 Bar 空间数据。	函数读取结果与通过仿真器获取结果一致。	通过
数据发送接收测试，通过自环连接端系统 AB 网口，通过发送函数，选择不同端口发送指定数据，再通过接收函数从指定端口接收数据。	端系统成功发送数据和接收数据，且数据一致。	通过
测试停止，通过 AfdxCloseDevice 关闭设备。	设备句柄关闭。	通过

5.2.2 板卡管理模块

当端系统自动化测试环境准备完成后，即安装好端系统板卡和驱动后，即可打开自

动化测试软件，点击顶部功能菜单区的板卡管理按钮，即可进入板卡管理窗口。板卡管理界面如图 5-2 所示。



图 5-2 板卡管理界面

在板卡管理窗口，左侧板卡列表显示当前主机中已插入板卡，选择其中一个板卡，即可在右侧显示板卡信息。右侧信息面板中显示了板卡的简要信息，通过角色选项，配置端系统作为 TE 测试端或 TR 被测试端。板卡的信息显示正确且板卡角色配置正确，板卡管理模块测试通过。

5.2.3 配置表管理模块

打开自动化测试软件，点击顶部功能菜单区的配置管理，即可进入配置表管理窗口。随后点击 OpenConfigFile 按钮，进入文件选择对话框，选择配置表文件界面如图 5-3 所示。

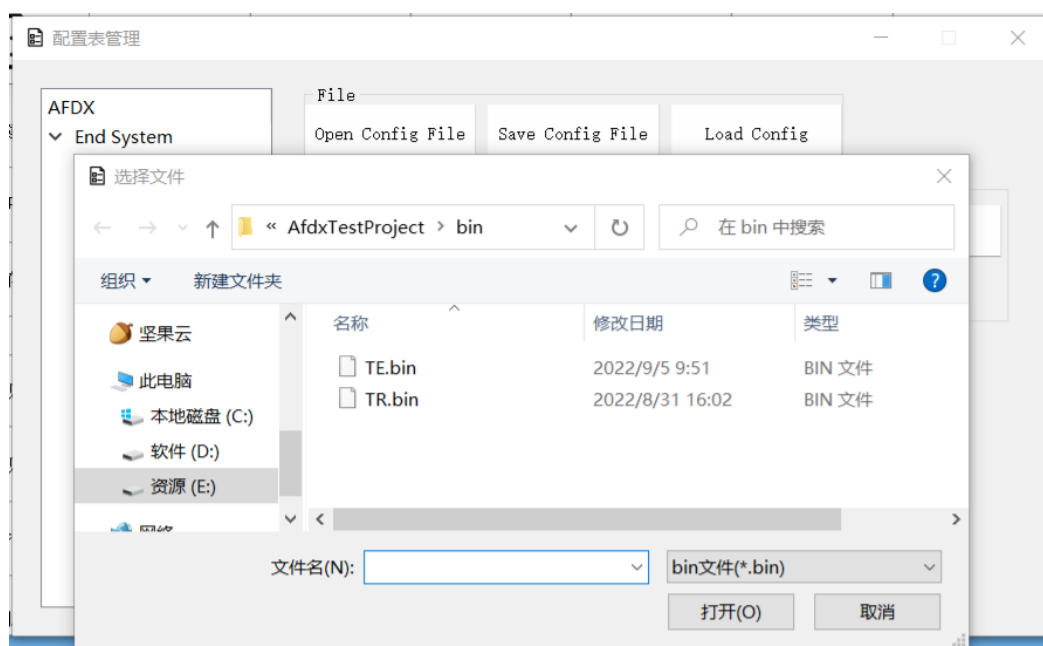


图 5-3 选择文件界面

打开的测试工程目录包含默认配置表，选择 TE 或 TR 的配置表，点击打开按钮。配置表管理模块解析二进制文件到图形界面，在图形界面根据需求修改配置表中的配置项。配置表解析结果以及配置表编辑界面如图 5-4 所示。

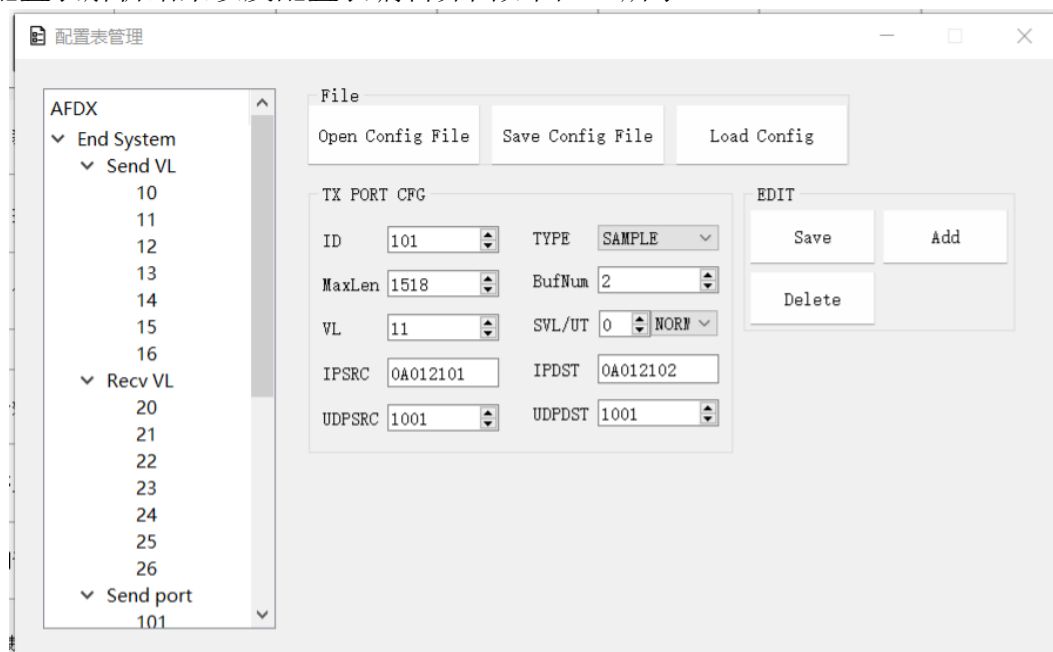


图 5-4 编辑配置表文件界面

用户编辑完成端系统配置表，点击 Save Config File 按钮，配置文件保存后用户点击 Load Config 按钮，调用硬件控制模块的配置表加载和创建端口。最后通过 ADS 和 Jlink 工具读取端系统的配置表内存地址，内存中数据与配置表文件数据对比一致，软件的配置表管理模块测试通过。

5.2.4 测试管理模块

打开测试软件点击软件菜单中测试管理按钮，进入测试用例管理界面，对测试用例进行管理，测试管理界面如图 5-5 所示。

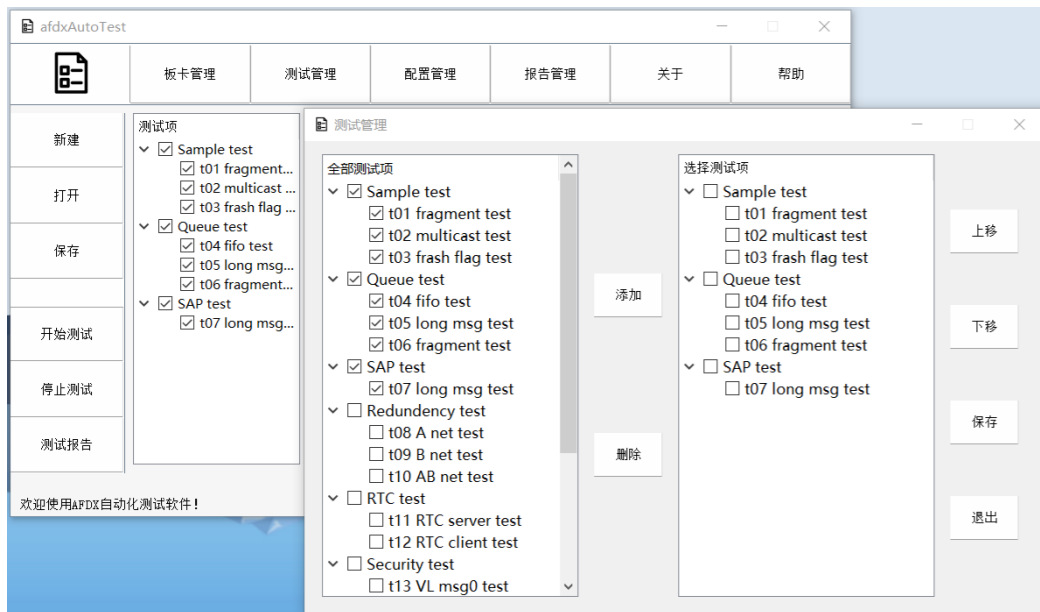


图 5-5 测试管理界面

界面的左侧为系统默认的全部测试用例树形列表，通过复选框勾选测试项，点击添加即可将测试项加入需要测试的列表。勾选右侧列表中的测试用例，点击删除按钮，选择删除的测试项从左侧待测试列表中删除。通过勾选和上移下移按钮，将测试项顺序进行修改。如果测试项选择完成，点击保存按钮，测试项保存，返回主界面，测试项将显示在测试控制软件主界面的测试项目管理区。

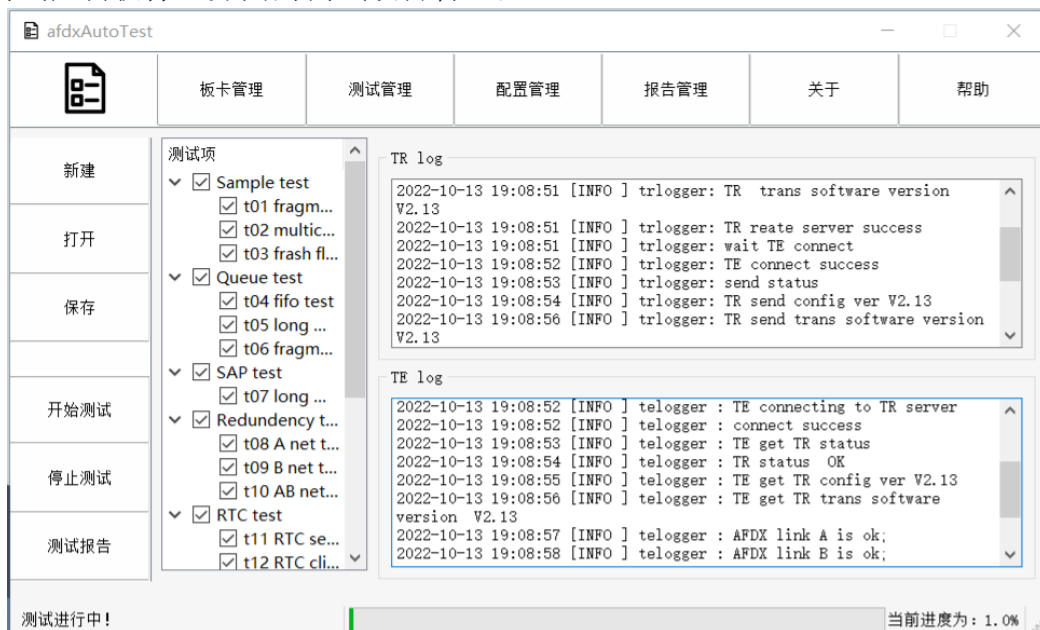


图 5-6 测试过程界面

最后，点击开始测试按钮，测试管理模块开始执行自动化测试，测试日志打印在测试控制软件的主窗口上，包括 TR 和 TE 的运行日志信息，进度条显示当前测试进度。测试过程界面如图 5-6 所示。

点击停止测试，自动化测试过程即停止，日志文件保存在测试工程目录下。测试管理模块测试通过。

5.2.5 报告管理模块

测试完成后点击测试报告按钮，测试控制软件的测试报告管理模块自动生成本次测试报告，测试报告以文件形式存储在测试工程目录下。点击报告管理按钮，进入文件选择，选择目录中单次测试报告文件后点击打开按钮，浏览测试报告，测试报告管理界面如图 5-7 所示。最后，点击打印按钮，进入打印对话框，选择打印机并输入相关打印信息后完成测试报告打印功能。测试报告管理模块测试通过。

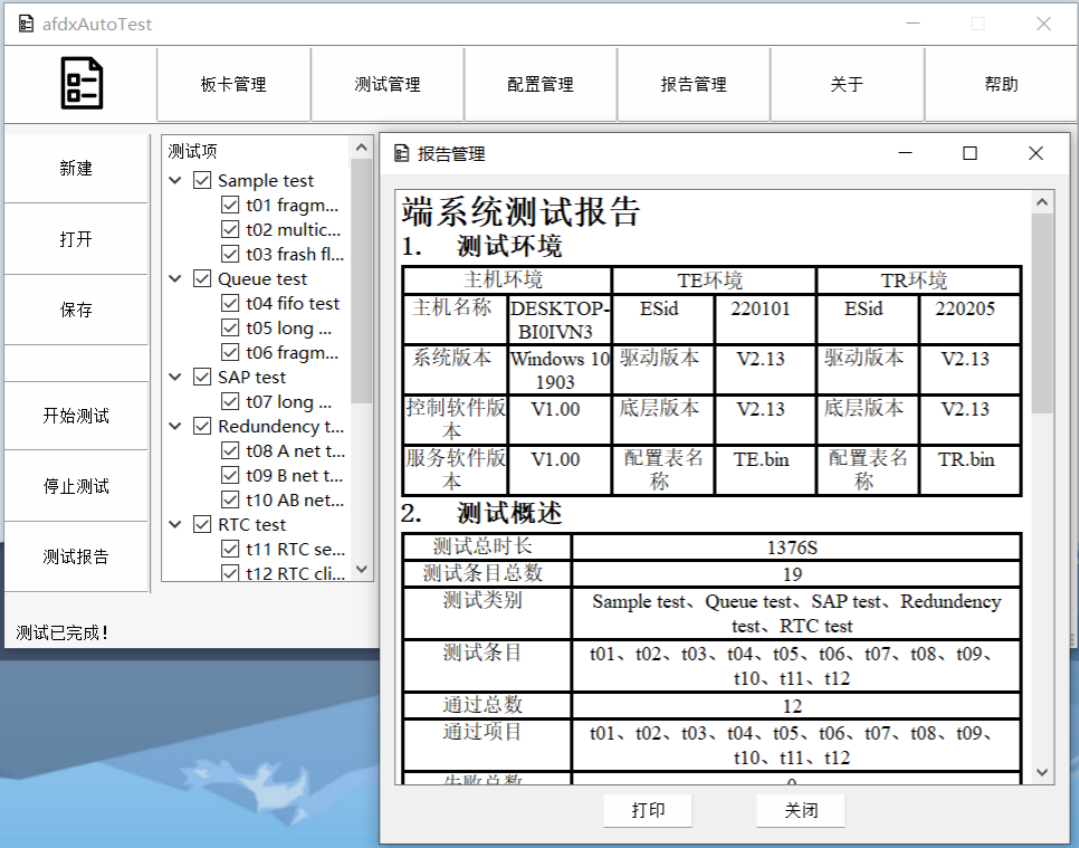


图 5-7 测试报告管理管理界面

5.3 自动化测试与人工测试方式对比

为了对比人工与自动化测试的好坏，这里选择除了端系统可靠性测试之外的剩余所有测试内容，分别使用人工方式和自动化测试软件方式对端系统执行相同测试项各五次。实际人工和自动化测试对比统计如表 5-3 所示。

表 5-3 人工和自动化测试对比

人工方式		自动化方式	
测试过程	平均单次人工 操作用时	测试过程	平均单次自动化 执行用时
硬件环境搭建	10 min	硬件环境搭建	10 min
软件环境搭建	20 min	软件环境搭建	15 min
测试用例执行	60 min	自动化测试执行	50 min
测试过程记录	10 min		
测试结果验证	20 min		
测试报告编写	30 min		
总用时	150 min	总用时	75 min
人工方式通过率	95%	自动化执行通过率	100%

由表格看出，人工操作测试中用时最多的是测试执行、测试验证以及测试报告的编写。采用人工方式，针对不同的测试项，测试人员需要使用不同的测试工具和测试软件，一项测试完成需要重新配置不同的软件环境，测试过程中测试人员需要专注记录测试过程，随后再进行对比完成结果验证，这些都导致了端系统测试效率低下，而且测试人员在同一时间只能专注于一张端系统测试。使用自动化测试软件则实现测试执行、测试验证以及测试报告的编写的自动化，端系统测试用时大大缩短，花费时间只有人工测试的一半。自动化测试过程用时随着测试项的增多，自动化测试用时与人工测试的比值会越来越小，并且一个测试人员可以同时兼顾多台设备的测试。因此，使用自动化测试软件执行端系统测试与人工测试的方式相比较用时更短，效率更高。

除去可靠性测试项目，在人工测试过程中，发生了少数的操作失误。而使用自动化测试的执行中没有发生人为操作失误的可能。因此相对于人工方式，自动化测试的准确度更高。

综上所述，AFDX 端系统的协议符合性测试工作中，本文设计的自动化测试软件优势更大，可以作为测试工作的一个更好选择。

5.4 本章小结

本章的主要内容为对端系统自动化测试软件功能按模块进行测试。首先对测试环境进行了介绍，随后针对硬件控制模块，板卡管理模块等进行了测试，软件各个功能模块正常运行无误。最后通过自动化测试与人工测试对比，体现了自动化测试的高效性和准确性。

第六章 总结与展望

6.1 论文工作总结

AFDX 作为一种可靠性高、带宽大的航空电子设备通信的网络协议，广泛应用于飞机上的数据通信系统中。在航空电子设备的设计制造过程中，AFDX 端系统的协议符合性测试是非常重要的。传统的 AFDX 端系统测试由测试工程师手动进行的，测试流程繁琐，效率低下，而且容易出现误差。在此背景下，设计实现一款 AFDX 端系统协议符合性自动化测试软件可以大大提高测试效率和准确性，同时也可以减少测试成本和人力资源的浪费。

首先，本文介绍了 AFDX 端系统测试的研究背景与意义，介绍了 AFDX 端系统的测试工作的重要性以及引入自动化概念测试的重要意义，研究分析了 AFDX 目前在国内外的应用现状和国内外 AFDX 测试相关的研究现状，对比了国内外差距以及介绍了现有工具的局限性和不足之处。

接下来，为了解决以上问题，基于对端系统自动化测试相关理论和对 AFDX 协议内容的研究总结，包括自动化测试周期方法学，端系统协议栈和消息收发处理机制等，在此基础上分析并总结了 AFDX 端系统协议符合性自动化测试内容。根据测试内容和测试方法，本文设计了端系统协议符合性自动化测试软件的总体方案，搭建了实际测试环境，使用 TR-TE 测试策略，将端系统自动化测试软件划分为测试控制软件和测试服务软件，测试控制软件在 TE 端，测试服务软件在 TR 端。同时以测试方法为基础，针对每个测试项，设计了详细的测试步骤。

随后，按照总体测试方案，对端系统自动化测试软件的需求进行分析，分别对测试控制软件和测试服务软件划分了软件模块，主要包括硬件控制模块、板卡管理模块、测试管理模块、配置表管理模块、测试数据显示模块、测试报告管理模块以及程序通信模块。然后介绍了各个模块的详细设计进，并设计了易于操作的用户图形界面。

最后对基于协议符合性的 AFDX 端系统自动化测试软件进行了功能测试验证，通过端系统的自动化测试和人工测试对比，端系统自动化测试相对于人工测试方式有着巨大优势。

总之，本文设计实现了一个针对 AFDX 端系统协议符合性的自动化测试软件，具有一定的创新性和实用性。该方案可以为端系统测试提供有效的支持和保障，为航空电子系统的安全性和可靠性的提高做出贡献。

6.2 未来工作展望

本文设计的测试软件基本上实现了端系统的协议符合性的自动化测试功能，但是由

于 AFDX 协议本身的复杂性以及实际硬件环境条件的制约，本测试软件还有一定的不足：

首先，测试软件的测试对象是端系统仿真卡，测试环境包括基础硬件、相应驱动软件和操作系统都是在仿真环境，而对应的测试软件模块的设计与实现都是针对上述的环境设计实现的，且不能同时对更多数量的端系统进行测试。因此，本文设计的测试软件的通用性略显不足。接下来的工作中，可以提升软件的通用性和跨平台性，使其适用于更多的测试环境。

其次，测试项目和自动化测试方案是在结合自身测试工作经验的总结而来，由于 AFDX 协议内容复杂，且目前硬件环境有限，目前的测试项并不能完全覆盖端系统的协议实现的全部功能。未来可以向提升测试覆盖率的目标上前进，丰富自动化测试项目，进一步完善自动化测试功能。

最后，对于后续的研究，可以向更加自动化的方面考虑，更大可能减少人工操作的步骤，推动 AFDX 自动化测试技术的发展。

参考文献

- [1] 熊华钢, 周贵荣, 李峭. 机载总线网络及其发展[J]. 航空学报, 2006(6): 1135-1144.
- [2] 周强, 熊华钢. 新一代民机航空电子互连技术发展[J]. 电光与控制, 2009, 16(4): 1-6.
- [3] 刘建平. AFDX 端系统协议栈虚拟链路层的应用研究[D]. 哈尔滨工业大学, 2012.
- [4] 曾北溟. 自动化测试框架的研究与实现[D]. 武汉大学, 2004.
- [5] 杨健. 基于开源系统的自动测试系统软件组件设计[D]. 电子科技大学, 2020.
- [6] 刘旭阳. 航空航天与国防电子新形势下自动化测试系统的应用[J]. 国外电子测量技术, 2014, 33(7): 4-5.
- [7] 张颖. AFDX 网络数字化仿真平台技术研究[D]. 西安电子科技大学, 2015.
- [8] 王传霸. ARINC664 终端测试系统软件的设计与开发[D]. 北京交通大学, 2018.
- [9] 汪宁. AFDX 航空通信协议及其核心技术研究[J]. 工业控制计算机, 2019, 32(11): 4-6+9.
- [10] 杨峰, 洪元佳, 夏杰等. AFDX 网络技术综述[J]. 电子技术应用, 2016, 42(4): 4-6+10.
- [11] 贾世伟, 何向栋, 张志平. 一种千兆 AFDX 端系统的设计与实现[J]. 信息通信, 2020(2): 108-110.
- [12] 支超有, 李振水, 李育. 大飞机先进机载数据总线 AFDX[C]//大型飞机关键技术高层论坛暨中国航空学会 2007 年年会. 2007.
- [13] 朱晓飞. AFDX 总线传输特性测试要求标准指标体系的确定[J]. 航空电子技术, 2021, 52(1): 61-66.
- [14] 李晓庆, 张志平. 端系统冗余度功能测试系统设计与实现[J]. 电脑编程技巧与维护, 2021(5): 15-16.
- [15] 侯溪溪, 周卓, 李天赋等. 基于 TestCenter 的 AFDX 网络性能测试技术研究[C]//第十六届中国航空测控技术年会论文集. 航空工业测控技术发展中心、中国航空学会测试技术分会、状态监测特种传感技术航空科技重点实验室, 2019: 191-194.
- [16] 王磊, 陈琳, 李勇. AFDX 网络测试技术研究与实现[J]. 计算机测量与控制, 2017, 25(3): 40-42+47.
- [17] 宁毅. ARINC664 标准符合性验证基本方法研究[J]. 科技创新导报, 2016, 13(14): 7-8.
- [18] 付军立. AFDX 总线 10M 接口电气参数极限适应性测试方法[J]. 测控技术, 2015, 34(5): 45-48.
- [19] Villegas J, Fortes S, Escano V, et al. Verification and Validation Framework for AFDX Avionics Networks[J]. IEEE Access, 2022, 10: 66743-66756.
- [20] Peng W, Makki S. The CSMA/DM LAN protocol[J]. Int. J. Commun. Syst., 2006, 19(8): 897-914.
- [21] 杜宏伟, 马捷中. 航空电子全双工交换式以太网及其关键技术研究[J]. 测控技术,

- 2008, 27(12): 65-67.
- [22] Fuchs C. The Evolution of Avionics Networks From ARINC 429 to AFDX[J]. *Innov. Internet Technol. Mob. Commun. Aerosp. Netw. IITM AN*, 2012, 65.
- [23] 刘晓胜, 吴瑾, 何晔等. 基于 AFDX 的航空数据网络终端设备设计[J]. *测控技术*, 2011, 30(2): 78-82.
- [24] 索高华. 基于 HK664SW 的 AFDX 网络交换机软件设计与实现[D]. 西安电子科技大学, 2019.
- [25] 罗帅. AFDX 网络端系统的设计[D]. 西安工程大学, 2017.
- [26] Schuster T, Verma D. Networking concepts comparison for avionics architecture[C]//2008 IEEE/AIAA 27th Digital Avionics Systems Conference. 2008: 1.D.1-1-1.D.1-11.
- [27] Erdiñç E. Soft afdx (avionics full duplex switched ethernet) end system implementation with standard pc and ethernet card[D]. Middle East Technical University, 2010.
- [28] 张琦, 王勇, 褚文奎等. AFDX 端系统虚链路触发返回技术研究[J]. *激光杂志*, 2015, 36(11): 131-135.
- [29] 张涵中. AFDX 航空通信协议及其核心技术分析[J]. *电子世界*, 2022(1): 9-10.
- [30] Al Sheikh A, Brun O, Chéramy M, et al. Optimal design of virtual links in AFDX networks[J]. *Real-Time Syst.*, 2013, 49(3): 308-336.
- [31] An D, Kim K H, Kim K I. Optimal Configuration of Virtual Links for Avionics Network Systems[J]. *Int. J. Aerosp. Eng.*, 2015, 2015: e329820.
- [32] Tian Y, Ma Z, Zhou S. Analysis of AFDX Network Delay Based on NS2[J]. *J. Phys. Conf. Ser.*, 2021, 2026(1): 012010.
- [33] 田泽, 姜丽云, 陈伟等. AFDX 网络关键协议分析与研究[J]. *电子技术应用*, 2016, 42(4): 7-10+14.
- [34] Li M, Zhu G, Savaria Y, et al. Reliability Enhancement of Redundancy Management in AFDX Networks[J]. *IEEE Trans. Ind. Inform.*, 2017, 13(5): 2118-2129.
- [35] 李硕, 王学望, 康锐. 面向完整性要求的航空电子全双工交换式以太网可靠性评价参数研究[J]. *西安交通大学学报*, 2013, 47(3): 126-131.
- [36] 徐双平. AFDX 网络时钟同步设计[D]. 中国民航大学, 2020.
- [37] 应杭. 软件自动化测试技术及应用研究[D]. 浙江大学, 2006.
- [38] Garousi V, Elberzhager F. Test Automation: Not Just for Test Execution[J]. *IEEE Softw.*, 2017, 34(2): 90-96.
- [39] 冷先刚. 软件测试模型与方法研究[D]. 武汉理工大学, 2009.
- [40] Dustin E. The automated testing life-cycle methodology (atlm)[C]//STAR EAST 2000 Conference, Orlando, Florida. 2000.
- [41] 赵永库, 唐来胜, 李贞. AFDX 网络测试技术研究[J]. *计算机测量与控制*, 2012, 20(4):

945-947.

- [42] 周卓, 张玉杰, 关乃轩等. 基于 TRTE 测试策略的 AFDX 网络端系统测试[C]//2014 航空试验测试技术学术交流会论文集. 航空工业测控技术发展中心、中国航空学会测试技术专业委员会、《测控技术》杂志社, 2014: 170-172.
- [43] 郭闯. 协议一致性测试原理及其应用[D]. 西华大学, 2008.
- [44] 周逊. IPv6——下一代互联网的核心[M]. Di 1 ban. Bei jing: 电子工业出版社, 2003.
- [45] 施颖, 丁日佳, 信春华等. 标准符合性测试初探[J]. 科技管理研究, 2013, 33(6): 135-137+208.
- [46] 董丽, 赵琪, 周健. 嵌入式软件安全性分析和测试方法研究[J]. 信息系统工程, 2021(5): 70-71.
- [47] 陶幸辉, 宋志刚. 软件系统测试类型及测试用例设计[J]. 科技经济市场, 2011(6): 3.
- [48] 宋孟华, 刘雪梅, 鲍成伟. 基于 HKS664ES 的 AFDX 端系统设计[J]. 自动化技术与应用, 2019, 38(7): 130-133.
- [49] 王绮卉. AFDX 核心交换技术的研究与实现[D]. 西安石油大学, 2010.
- [50] 孟军, 赵平均, 余奎等. 基于 ARINC664 PART7 的航电系统[Z]. 2017.
- [51] 张志平, 李晓庆, 周耿. 端系统冗余度管理功能设计和实现[J]. 电子测试, 2020(7): 3.