

航天测试系统通用数据处理软件的设计

马志刚^{1,2}, 刘文怡², 张文栋²

(1 山西农业大学 信息科学与工程学院, 山西 太谷 030801;

2 中北大学 仪器科学与动态测试教育部重点实验室, 电子测试技术重点实验室, 山西 太原 030051)

摘 要: 为了设计一种具有较高通用性和灵活性的数据处理软件, 分析了航天测试数据的帧结构, 讨论了 C、C++、MFC 中文件操作的一般方法. 利用 Win32 API 相关函数, 研究了测试数据的解包、分析等处理方法. 结合对数据帧结构和混编包结构的分析, 利用 Visual C++ 6.0 编写了通用数据处理软件. 经过实际测试验证, 提出的方法和设计的软件具备一定的通用性和灵活性, 可以有效地对各类测试数据进行处理, 且已成功应用于多项航天测试任务, 为相关测试和试验提供了可靠依据.

关键词: 通用; 航天测试; 帧结构; 数据处理; 应用程序接口

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2014)11-0098-04

Design of Universal Data Processing Software in Aerospace Test System

MA Zhi-gang^{1,2}, LIU Wen-yi², ZHANG Wen-dong²

(1 College of Information Science and Engineering, Shanxi Agricultural University,

Taigu 030801, China; 2 Key Laboratory of Instrumentation Science & Dynamic

Measurement, North University of China, Ministry of Education, Science and Technology on

Electronic Test & Measurement Laboratory, Taiyuan 030051, China)

Abstract: In order to design a kind of software with high universality and flexibility for data processing, the frame structure of aerospace test data was analyzed, and the general methods of file operation in C, C++ and MFC were discussed. Data unpacking, analysis and other treatment methods were studied combined with relational functions in Win32 API. According to the analysis of data frame and mixed-framed packet structure, the universal data processing software was programmed in Visual C++ 6.0. By means of practical testing verification, the proposed methods and software had been proved having some universality and flexibility to carry out several types of test data analysis effectively. Meanwhile, the proposed software had been successfully applied to a number of aerospace test tasks, which provided the reliable basis for relational tests and experiments.

Key words: universality; aerospace test; frame structure; data processing; Application Program Interface (API)

1 引言

航天测试领域常见的各种数字量和模拟量(缓变、振动、冲击、噪声、高速图像等)测试数据被传输至计算机并形成数据文件后,通常需要进行完整性判读、数据分路、模拟量分析(物理量导出、曲线描

绘、平滑滤波、谱分析、统计学分析)等处理. 上述分析和处理都要通过数据处理软件完成.

在航天测试系统中,需要处理的各类数据通常具有不同的数据组织方式,但同时又具备一定的共性或相似性. 如果每一类甚至每一组测试数据都分别编制数据处理软件,显然不利于程序移植和维护,

收稿日期: 2014-01-06; **修回日期:** 2014-02-28

基金项目: 国家自然科学基金项目(51275491)

且同时会造成较大浪费.因此根据测试数据的数据组织(编帧)方式,利用计算机程序设计的特点,进行通用数据处理软件的研究,对提高程序执行效率、减少代码编写量、避免过多重复劳动、增强软件的可移植性和通用性具有重要意义.本文讨论的数据处理算法是基于C++的,软件开发工具采用 Visual C++ 6.0.

2 测试数据的数据组织方式

一般而言,虽然各种测试数据有不同的数据组织方式,但有规律可循.工程上一般将这种数据组织方式称为“编帧”,编帧得到的一帧数据称为“帧结构”^[1],测试数据就是由一系列“帧结构”组成的.

2.1 简单数据帧结构

数据的帧结构有的比较简单,有的则相对复杂.表1所示的是一类比较简单的数据帧结构.

表1 简单的数据帧结构				
名称	帧头 (FrmHead)	数据 (FrmData)	帧计数 (FrmCnts)	帧尾 (FrmTail)
实例	0X"EB 90"	0X"XX ... XX"Cnt1 Cnt2... Cntn		0X"14 6F"
数据长度	LenHead	LenData	LenCnts	LenTail
		FrmLen		

表1所示的数据帧结构可分成四部分:

(1) 帧头(FrmHead)用于指明一帧数据的开始位置.工程实践中常用 0X"EB 90"两个字节作为帧头,一般认为"EB 90"不易在传输中被干扰.

(2) 数据(FrmData)是测试数据的核心部分,也是测试信息所在.一帧数据中的“数据”部分可长可短,其长度设置一般需考虑的因素有:采样路数、采样率、时序等.

(3) 帧计数(FrmCnts)用于指定当前帧的序号.其作用在于判断两个相邻帧是否连续,正常情况下,相邻两帧的计数差值为1.如果当前帧的计数达到最大值,则下一帧的计数会清零.以三个字节帧计数为例,若当前帧的计数为 0X"FF FF FF",则下一帧的计数应为 0X"00 00 00".

(4) 帧尾"FrmTail"指明该帧结束,一般用两个字节表示.实际应用时,在有帧头的情况下,帧尾部分可以省略.

2.2 复合数据帧结构

复合帧结构是工程中另一类较为常用的帧结构.例如,某类 PCM 数据常用以下帧结构,如表2所示.

表2 某类复合数据帧结构				
列	共 M 列			
行				
第 1 行	数据(ColData)	行计数(ColCnt)	行标志(ColSgn)	
第 2 行	数据(ColData)	行计数(ColCnt)	行标志(ColSgn)	
共 N 行	...	...	...	...
第 N-1 行	数据(ColData)	行计数(ColCnt)	行标志(ColSgn)	
第 N 行	数据(ColData)	帧计数(FrmCnts)	帧尾(FrmTail)	

在表2所示的数据帧结构中,一帧数据包含 M×N 个字节,共分 N 行,每行 M 个字节.其中:前(N-1)行每行包括(M-3)个字节数据、1 字节行计数和 2 字节行标志(常用 0X"EB 90");最后一行(第 N 行)包括(M-5)个字节数据、3 字节帧计数和 2 字节帧尾(常用 0X"14 6F").这类复合帧结构构造方式多用于多类、多通道参数的编帧,常用的有帧长为 24 576(192 列×128 行)字节、24 000(250 列×96 行)字节等帧结构.

正常情况下,测试数据中相邻两帧数据的帧计数应该是连续的.如果在数据传输、存储等过程中存在干扰或异常,就会产生数据误码或丢失.若误码出现在帧计数、帧标志上,或者丢数都会导致相邻两帧数据的帧计数不连续.若测试数据中只有少量错误,单凭肉眼观察原始数据是很难发现的,因此需要借助专门的数据处理软件来判读.

3 测试数据文件操作

文件一般分为文本文件和二进制文件^[1].本文讨论的测试数据文件属于二进制文件.通常,程序中的数据在程序运行结束之后,就会从内存中清除,而在数据分析时不可避免地会遇到数据永久保存的问题,这时就涉及到文件操作.

3.1 文件的常规操作方法

在 C 语言中,可先用 fopen() 函数创建并打开一个文件,同时将获得一个指向 FILE 结构体的指针,在读写文件时需分别调用 fread() 和 fwrite() 函数^[2].在 C++ 中,可利用 ofstream 类调用 read()、write() 等函数进行文件读、写^[3].

MFC 中提供了支持文件操作的 CFile 类,该类提供了很多方法,读取和写入数据可分别调用 Read() 和 Write() 函数.而通过 GetLength() 函数可以获得文件的大小,该函数的返回值为 DWORD 型(32 位无符号整数).这意味着,该函数只能用于处理大小不超过 4 GB 的文件.如果文件大小超过 4 GB,则需考虑采用别的方法来获取其大小.

3.2 利用 Win32 API 函数操作文件

Win32 应用程序接口(Application Program Interface, API)提供了大量与文件操作有关的函数,包括: CreateFile()、ReadFile()、WriteFile() 和 FindFirstFile()等. CreateFile()用于创建一个新文件或打开一个已有文件;ReadFile()可以从指定的文件中读取数据;WriteFile()可以向指定文件中写入数据;通过调用 FindFirstFile()函数,可获得指定文件的详细信息. FindFirstFile()函数的语法格式如下:

```
HANDLE Find First File
(
    LPCTSTR lpFileName;
    LPWIN32_FIND_DATA lp Find File Data
);
```

该函数执行成功后, lpFindFileData 将获得一个 WIN32_FIND_DATA 型结构体,其具体内容如下:

```
typedef struct _WIN32_FIND_DATA
{
    DWORD dwFileAttributes;
    FILETIMEft CreationTime;
    FILETIMEft LastAccessTime;
    FILETIMEft LastWriteTime;
    DWORDn FileSizeHigh;
    DWORDn FileSizeLow;
    DWORD dwReserved0;
    DWORD dwReserved1;
    TCHAR cFileName[ MAX_PATH];
    TCHAR cAlternateFileName[14];
}
```

该结构体包含的信息较多,其中的 nFileSize-High 和 nFileSizeLow 分别表示文件大小(以字节为单位)的高 32 位和低 32 位. 可见,使用该函数可以处理大小不超过 16 EB(16 777 216 TB)的文件.

4 测试数据分析

通常,对测试数据的分析包括:分离并分析通道数据、数据完整性分析等. 在有的测试系统中,还需对连续变化的模拟量参数进行曲线显示或物理量导出. 另外,一般还需要生成分析报告,该报告用于保存数据分析结果,包括测试时长、有效帧数、是否存在丢帧情况等信息. 数据处理的一般过程如图 1 所示.

4.1 通道数据分离和分析

在实际工程应用中,各类测试数据往往较多,其采样时间、采样率、参数类型等也各有不同,因而获

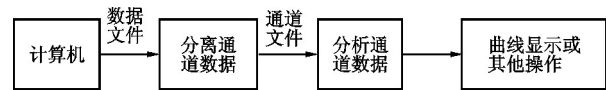


图 1 数据分析的一般过程

得的测试数据大都由各类数据混合编帧(混编)而成. 因此,在数据分析之前,必须先将各类(或各通道)数据进行分离(分路),并生成单独的通道数据文件.

表 3 所示的是某混编数据的包结构示意图,这是一种比较简单的情況. 该包结构共 103 字节,包含两类数据,分别为数字量数据(包尾 0X "AA+AA")和模拟量数据(包尾 0X "E7+E7"),它们的包头均为 0X "55+55". 为了判断数据包的完整性(是否丢包),包结构中设计了包计数(3 字节),采用两类数据包分别计数的方式.

表 3 某混编数据的包结构示意图

包头(2 字节)	有效数据	包计数	包尾(2 字节)
0X "55+55"	(96 字节)	(3 字节)	0X "AA+AA"或 0X "E7+E7"

数据分离时,需先将测试数据从文件中读出. 当数据量不大时(例如不超过 200 MB),可直接将文件中的所有数据一次全部读出并存入动态数组;当数据量较大时,则应采用“按块读取”、“逐块分析”的方式. 即先读出一块数据(例如 1 MB)进行分析,之后将该块数据中剩余的未分析部分缓存至临时空间;再读出一块数据,将上一块剩余数据与该块数据拼接后再进行分析. 以此类推,直至分析完毕^[1].

一般而言,每类通道数据均有其特定的帧结构,但在帧结构设计上有一定规律性,因此在编写数据分析算法时需充分考虑程序的灵活性、通用性、可移植性. 为了适应不同数据的分析,相关帧参数(帧头、数据长度、帧计数个数和帧尾等)应当是可设置的.

4.2 曲线显示和其他操作

曲线是对测试数据最直观的描述方式. 因此在很多情况下,都会要求对连续变化的模拟量信号(电压、温度、液位、压力等参数)进行曲线显示.

微软基本类库(Microsoft Foundation Classes, MFC)中提供了大量封装好的类,其中包含丰富的绘图工具和方法. 绘图工具类封装了位图、画笔、画刷等,包括: CGdiObject、CPen、CBrush、CBitmap 等. 应用时需要哪种绘图工具,就用对应的绘图工具类生成相应的对象,然后调用 SelectObject() 函数将其选入设备描述表,使其生效^[1]. 在绘制曲线时,可以使用 LineTo() 函数依次连接两个数据点,最终

形成一条曲线.为了提高程序执行效率,也可以直接调用 PolyLine() 函数完成曲线绘制.

此外,有的测试系统还要求对测试数据进行:预处理(野点修正、插补漏值、趋势项消除等),时域分析(统计特性分析、平滑、滤波等),频域分析(频谱分析、功率谱分析、冲击谱分析)等.

5 测试数据通用处理软件设计

为实现数据处理软件的通用性,软件中必须将通道数据的帧结构、包结构设计为有选项、可设置模式.对于数据解包,需要设置参数:包内通道数据数目、包头字节数、包头内容、包内数据长度、包计数字节数、包尾字节数和包尾内容.其中,包内通道数据数目即包内共有几种不同类型通道数据,每种通道数据都需要分别设置相关参数.采用 Visual C++ 6.0 编写的软件界面如图 2 所示.对于通道数据的分析,需要设置参数:帧头字节数、帧头内容、帧内数据长度、帧计数字节数、帧尾字节数和帧尾内容,其界面效果与图 2 所示类似,此处不再赘述.



图 2 数据解包模块软件界面

6 结束语

本文以航天测试数据处理为背景,从软件的通用性问题出发,结合 C++、MFC、Win32 API 等对文件操作的支持,研究了测试数据处理的一般方法,并在 Visual C++ 6.0 环境下编写了通用数据处理软件.通过对自检数据和实际测试数据的分析处理,证明本文讨论的通用数据处理方法和软件具有较好的通用性,能够完成多类测试数据的处理,经过简单修改或改进可以运用到其他测试领域.此外,还有如下问题值得考虑:

(1) 实时处理问题.在许多复杂测试系统中,都需要对数据进行实时处理,以便系统在任何一个子

系统或模块出现故障时能立即得到正确处理;而在另外一类测试领域,实时数据处理意味着能及时地对测试目标的行为做出反应(控制).但在通常情况下,这种实时处理是比较困难的^[4].

(2) 软件的执行效率问题.待分析处理的数据量越大、帧(包)结构越复杂,耗时就会越长,因此程序执行效率必须考虑.程序执行的效率与数据的存储结构、程序的控制结构、所处理的数据量等因素有关.

(3) 如何对大容量(例如超过 4 GB)甚至海量数据^[5-6]进行处理,需借助更多技术和算法,例如:内存映射技术、多线程技术^[7]、多核技术^[8]、并行技术^[9]、大数据处理技术^[10]等.

参考文献:

- [1] 马志刚,刘文怡. 航天测试系统大容量数据曲线绘制方法的研究[J]. 图学学报, 2014, 35(4): 623-629.
- [2] 谭浩强. C 语言程序设计[M]. 4 版. 北京: 清华大学出版社, 2010: 331-337.
- [3] 陈维兴,林小茶. C++ 面向对象程序设计教程[M]. 3 版. 北京: 清华大学出版社, 2009: 314-321.
- [4] 刘文怡,甄国涌. 高速测控系统中实时数据处理软件技术研究[J]. 华北工学院学报, 2004, 25(3): 191-193.
- [5] 王桂强. 海量数据分析处理方法的研究[D]. 上海: 上海交通大学, 2010.
- [6] 汤劲. 海量数据处理系统框架关键技术研究[D]. 南京: 南京航空航天大学, 2006.
- [7] 唐轶轩,吴俊敏,陈国良,等. 面向多线程程序基于效用的 Cache 优化策略[J]. 计算机研究与发展, 2013, 50(1): 170-180.
- [8] 刘虎球,赵鹏. 一种多核间内存公平调度模型[J]. 计算机学报, 2013, 36(11): 2191-2199.
- [9] 魏海涛,于俊清,余华飞,等. 一种面向数据流程序的软件流水并行化方法[J]. 计算机学报, 2011, 34(5): 889-898.
- [10] 李建中,刘显敏. 大数据的一个重要方面:数据可用性[J]. 计算机研究与发展, 2013, 50(6): 1147-1162.

作者简介:

马志刚 男,(1982-),博士研究生,讲师.研究方向为惯性感知与测控技术、航天智能测量系统数据处理等.

刘文怡 男,(1970-),博士,教授,博士生导师.研究方向为航天智能测量系统、无线传感网络等.