

# 基于 LabWindows/CVI 的 ARINC429 总线数据采集系统设计

## Design of ARINC429 Data Acquisition System based on LabWindows/CVI

徐 州 徐 真 张 勇 魏 博 / Xu Zhou Xu Zhen Zhang Yong Wei Bo

(上海飞机设计研究院,上海 201210)

(Shanghai Aircraft Design and Research Institute, Shanghai 201210, China)

### 摘 要:

首先设计了一套基于 LabWindows/CVI 的 ARINC429 总线数据采集系统,该系统采用服务器端和客户端结合的方式,利用线程池的次线程采集数据,并利用异步时钟把数据打包发送给客户端。其次,讨论了两种采集方式对数采系统的影响,并且对 ARINC429 总线某个 Label 号的数据失效进行了处理,为以后的 ARINC429 的数采操作提供了借鉴。

**关键词:** ARINC429; LabWindows/CVI; 数据采集; 数据失效

[Abstract] Firstly, this article designed a data acquisition system based on LabWindows/CVI. The system combined service and client terminal. It used secondary thread of thread pool to acquire data and packed and transmitted data to client terminal in asynchronous timer. Secondly, this article discussed the influence of two different ways of data acquisition and dealt with invalidation of some Labels, which can provide reference for further using LabWindows/CVI to acquire data of ARINC429.

[Key words] ARINC429; LabWindows/CVI; Data Acquisition; Data Invalidation

## 0 引言

ARINC429 数据总线是专为航空电子系统通信规定的航空工业标准,其全称为数字信息传输系统(Digital Information Transfer System,简称 DITS)<sup>[1]</sup>,它详细规划了航空电子系统中各个电子设备之间及电子设备和系统之间的通讯方式,定义了电气特性、传输数据特性和通讯协议,忽略了不同厂家航空电子系统接口的复杂性,为系统互联提供了统一的平台,因此广泛应用于航空电子产品<sup>[1-2]</sup>。目前,空客与波音系列产品都用到了 ARINC429 总线,包括 A320、A340、B757、B767、B777 等,中国自主研发的某型飞机也采用了 ARINC429 总线作为主要的航电信号传输总线。

## 1 ARINC429 总线简介

ARINC429(以下简称 A429)总线的接线布局基

于一条 78ohm 的屏蔽双绞线,每条总线有一个发送器和多个接收器(可以多达 20 个),是一种单一传输广播式数据总线。实际的发送速度可有两种选择:低速为 12.5kHz,高速为 100kHz<sup>[3]</sup>。

一个典型的 A429 字长度为 32 位,由五个部分组成:标签(Label)、源/目标标识位(SDI)、数据区(Data)、符号/状态位(SSM)、奇偶校验位(P)。其具体传输字格式如图 1 所示。

P	SSM	Most Significant Data				DATA - 19 bits										Least Significant Data				SDI	8-Bit Octal Label										
32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
MSB		32-BIT ARINC 429 Word																			LSB										

图 1 ARINC429 总线传输字格式

ARINC 字中的 1 由  $10 \pm 1V$  产生,而 0 由  $-10 \pm 1V$  电压产生。ARINC 的拓扑结构一般都采用星形拓扑结构。

## 2 ARINC429 操作编程

每块 429 卡都集成了综合航空函数库,该函数

库能够支持多种航空总线的编程。A429 编程函数是综合航空函数库的子集。综合航空函数库主要包含两类对 A429 总线操作的函数,分别是以“LL\_”开头的底层低级别函数和以“a429\_”及“sbs\_”开头的高级别函数。他们能方便地被各种编程语言调用,比如 LabWindows/CVI。这些函数能够实现对 A429 总线进行设备管理、数据接收、数据发送、中断处理以及总线监控等功能。主要对 A429 总线进行设备管理及数据接收和采集操作。利用综合航空函数库对 A429 卡进行操作的基本步骤如下<sup>[3]</sup>:

(1) 初始化 A429 设备,使用 sbs\_init\_device (device\_number, queue\_length, bsm\_length, firmware\_source) 函数,这个函数能打开 A429 设备、启动固件、初始化内存、中断序列缓冲区和 BM 序列缓冲区;

(2) 定义传输、接收和总线监控数据结构;

(3) 用 sbs\_start\_io(device\_number) 启动 A429 总线 I/O 处理;

(4) 用 A429\_get\_current\_value(device\_number, channel\_number, label, values, force\_swap) 函数进行数据的接收与采集;

(5) 用 A429\_stop\_reciever(device\_number, channel\_number) 停止采集数据;

(6) 用 sbs\_stop\_io(device\_number) 函数停止 A429 总线 I/O 处理;

(7) 关闭 A429 设备,用 sbs\_close\_device(device\_number) 函数。

### 3 A429 总线数据采集系统设计

本文设计的 A429 数据采集系统如图 2 所示。

该系统由数采服务器采集 A429 总线数据,并打包发送至数采客户端。采集和发送数据的过程通过多线程来完成。由客户端来控制采集程序的开始与终止,在异步时钟的回调函数中进行响应。数采服务器向客户端发送的数据按需发送,即客户端选择某些需要的 Label 号,服务器端通过数据需求数据库表来读取客户端所需要的数据,对这些数据进行解析,并将这些数据在采集后打包发送。

数据的解析主要有两类,一类是离散量的解析,一类是模拟量的解析。离散量解析相对简单,从一个 A429 总线数据的第 11 位开始取出,每一位对应一个变量进行赋值即可。而模拟量的解析则需要确定该模拟量对应的数据区的起始位和终止

位,并知道这些数据位的最小分辨率,结合这两个信息能解析得到相应的变量的值。

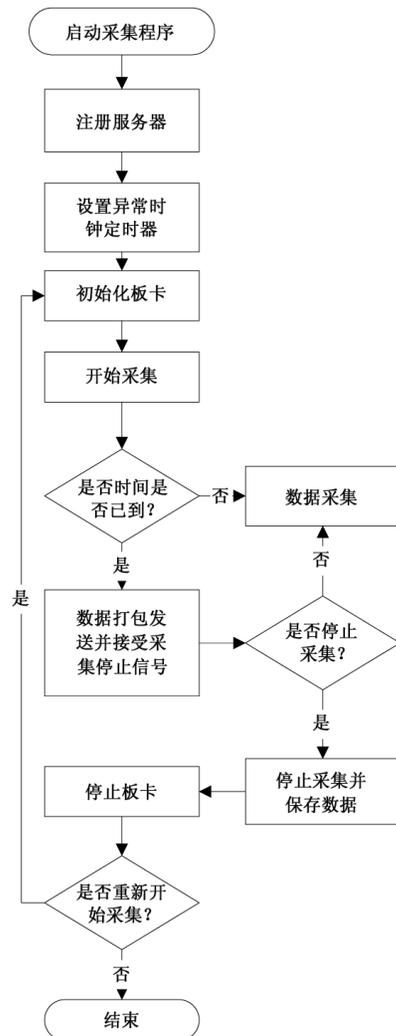


图 2 A429 数据采集系统流程图

## 4 数据采集系统的几个关键问题

### 4.1 采集方式的选择

在本文设计的 A429 数据采集系统中,由于客户端需要的数据实时变化,因此,在数采服务器端进行采集的方式主要可以有两种。一种是根据客户端需要的 Label 号进行数据的采集,也就是说,服务器端读取客户端的需求列表,然后一一对需求列表进行判断,只采集列表中列出的那些 Label 数据再进行打包发送;另一种是循环全部采集,也就是说数采服务器端对所有的 Label 号的数据都进行采集,然后在打包发送给客户端时根据客户需求列表进行打包发送。

这两种采集方式各有优劣。对于第一种采集

方式来讲,当总的 Label 数较少时,在一个设定的时钟内采集的数据个数较多,能够满足较高的采集频率的要求,不过当客户的需求变化时,一个时钟内采集每个 Label 号的数据个数变化很大。比如当客户需要的 Label 号只有 5 个时,在一个时钟内,每个 Label 号采集的数据个数能达到 100 多个,而需要的 Label 数变为 50 个时,每个时钟采集的每个 Label 号的数据个数就只有 10 多个,因为增加了很多判断及解析的过程。对于第二种方式由于采集时对所有 Label 号都进行采集和解析,因此在一个时钟周期内,每个 Label 号的数据个数较少,只能适用于采集频率较低的系统。不过这种采集方式比较稳定,采样的点数不随客户的需求而变化。另外,由于本系统采用的是服务器端采集后打包发送的形式,在每次打包的过程中,一个包中包含的数据个数是根据客户端的需求来的,比如说本系统中的异步时钟设为 0.3s,如果一个客户端需要 Label 的采样频率为 20Hz,则一个包里包含该 Label 的数据个数为 6 个。当采集的频率太高时,打包发送的数据会集中在某一个时间点附近,导致数据采集不均匀。

综合上述理由,由于本系统要求的采样频率不高,故采用全部循环采集的方式。

#### 4.2 基于多线程的数据采集

多线程是指操作系统支持一个进程中执行多个线程的能力,它能够更好地开发并行性任务和提高系统性能,在避免阻塞、减少运行时间和最大程度利用多处理器的性能方面具有明显的优点<sup>[4]</sup>。

在本系统中,采用 LabWindows/CVI 多线程编程,使用主线程来操作 A429 卡的启动、配置、开始采集、关闭设备以及异步时钟的创建等功能,使用异步定时器来进行数据的定时打包发送与接收客户端的终止程序的请求,使用次线程池来执行数据的采集与解析,这样有利于给数据采集与解析分配专门的线程,提高效率。

异步时钟由 NewAsyncTimer(double doubleInterval, int count, int status, AsyncTimerCall backPtrcallbackFunc, void \* callbackData) 进行设置,而数据的打包与发送及接收客户端的终止程序请求在异步时钟里的回调函数中执行。

数据的采集与解析利用线程池中的次线程来执行,线程池的次线程可以调用 Utility Library 库中的 CmtScheduleThreadPoolFunction 函数,把想要在次线程中执行的函数名传递给它,线程池就会调度这

个函数在它的一个线程中运行。CmtScheduleThreadPoolFunction 函数的原型为:

```
int CmtScheduleThreadPoolFunction ( int pool -
Handle, ThreadFunctionPtr threadFunction, void *
thread - FunctionData, int * threadFunctionID );
```

#### 4.3 A429 数据失效的处理

在 A429 数据采集系统中,往往需要进行一些故障状态的采集,比如失去某一个 Label 的 A429 数据来检查整个系统对失去该 Label 数据的响应,因此需要判断在某一个时刻该 Label 的数据是否失效。而当某个 Label 的数据失效时,利用 A429\_get\_current\_value 采集到的数据会保持失效前一个时刻的值或者值为 0,如果保持前一个时刻的值,则无法分辨数据失效的那个时间;如果采集到的数据为 0,也无法判断该 0 值是由于该 Label 数据失效造成的还是该 Label 本身的值即为 0,容易产生混淆。因此本文利用 A429\_get\_current\_value 采集到的 A429 数据自带的时间戳进行了数据失效时刻的判断。

在 A429\_get\_current\_value ( device\_number, channel\_number, label, values, force\_swap) 函数采集到的 values 可以分成五个部分: values[0] ~ values[4]。其中, values[0] 和 values[1] 为数据, values[2] ~ values[4] 为该数据的时间戳 ( values[2] 为时间戳的低位, values[4] 为时间戳的高位)。通过比较某个 Label 失效前后的时间戳可以发现,当该 Label 未失效时,时间戳的数值是一直在增大的;而当该 Label 失效时,第奇数个数据的时间戳相同,第偶数个数据的时间戳也相同。利用这个规律,可以设计一个简单的算法来进行某个 Label 号的数据是否失效。本文采用的方法如下:

$$\text{Time\_stamp1} = \text{values}[0][2] + \text{values}[0][3] * 10^5 + \text{values}[1][2] + \text{values}[1][2] * 10^5 + \text{values}[18][2] + \text{values}[18][3] * 10^5 + \text{values}[19][2] + \text{values}[19][3] * 10^5;$$

Time\_stamp1 是前一个包中的数据的时间戳的加权之和,参与加权的是每个包的第 0、1、18 和 19 个数据的时间戳的前两个 values 值 (时间戳有 3 个 values,其中 values[4] 始终保持不变)。每个 values[3] 乘上  $10^5$ ,是为了体现 values[3] 时间权重。如果前一个包的时间戳权重之和与后一个包的时间戳权重之和相同,则认为该 Label 的数据已经失效,可以把该 Label 的数据设置为一个标志失效的值,比如说本系统设置为 999。

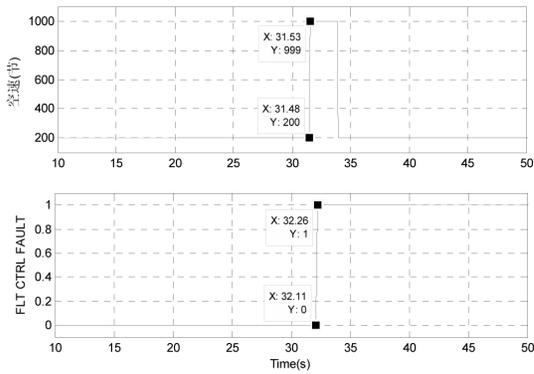


图3 空速失效触发故障信息测试图

经过失效的处理结果如图3所示。从图中可以看出,在31.53s时,空速由200节变为无效,在32.26s时触发“FLT CTRL FAULT”故障信号。经过这样处理的信号很清楚地知道失效的发生时刻。

## 5 结论

基于 LabWindows/CVI 的特点设计了一套

ARINC429 总线的数据采集系统。该系统由服务器端和客户端组成,在客户端进行数据采集的开始与结束的控制,并利用线程池进行数据采集,异步时钟进行数据的打包发送。本文也讨论了两种采集方式的优劣性,根据实际系统的需求,选择了合适的方式;并对 ARINC429 总线中当某个 Label 数据失效时进行了处理,为以后的 ARINC429 数据采集系统设计提供了借鉴。

### 参考文献:

- [1]张学毅,于惠钧,梁建华.基于ARM与Windows CE的LCD显示器设计[J].电子测量技术,2008,31(7):76-79.
- [2]郭跃云,杨明涛,郑宾.基于DSP的ARINC429总线收发系统设计[J].仪表技术与传感器,2010(5):56-58.
- [3]SBS Technologies Inc. ARINC User's Manual[Z]. 2005.
- [4]杨东升,王高峰.多线程技术在虚拟仪器开发软件LabWindows/CVI中的实现[J].电测与仪表,2005,42(3):39-41.

(上接第70页)

上讲,模型和输入变量不一定是始终不变的,需要始终保持对航空市场的关注,对各种影响因素进行

分析。在有外部原因变化使得某些因素对航空业的发展有重大影响时,可以引入新的参数以使得模型更加准确,如表1所示。

表1 影响航空需求的相关因素

人口增长	超出飞机的承载量	飞机制造商定价	竞争和调整架构(欧洲/非欧洲)	欧洲“单一天空”计划	人力制度	行业金融和收益率	旅行体验
人口统计学的趋势	非欧洲航空公司的战略	GDS 战略/成本	MRO 成本	团体旅行趋势	航空公司联合战略	人力流动趋势	工业集中水平
GDP 趋势	非欧洲政府的战略	旅行分布市场	枢纽/轮辐战略	恐怖行动	用于航空运输的政府资金	新商业模式	新基础设施的成本
空中旅行的价格	商务/休闲趋势	网络渗透	货机增长	安全成本和运行影响	通讯业务发展	培训	环境成本/政策
欧洲国家的数量	机场的容量和位置	安全因素成本	航空公司成本结构	团体旅行定价	燃油价格	收入分配	机场收费和资金
空中管制效率	人力成本	法规和销售	健康/疾病问题	改变旅行模式	低成本运营商增加	高速铁路的竞争	新技术

## 7 结论

总的来讲,对民用飞机市场需求预测方法的研究对市场研究能力起着决定性作用,需求预测可采用短期或长期预测法,飞机产量预测适合采用长期预测方法。如何处理数据和建模是预测工作所面临的挑战,要重视实际数据中的不连续点和转折点,以及预测工具可方便获取及使用成本;并运用市场知识对数据的合理性加以验证,在此基础上得

出的预测结果更为可靠。

### 参考文献:

- [1]Mariya A. Ishutkina\_ and R. John Hansman Analysis of Interaction between Air Transportation and Economic Activity .
- [2]Marco Kouwenhoven, Eric Kroes ,Jan Veldhuis Welfare Effects of Capacity Constraints at Schiphol Airport - A New Model to Forecast Air Demand.