

使用 Android 智能手机的 ARINC429 总线系统设计

刘连生, 梁林林

(中国民航大学基础实验中心,天津 300300)

摘要: 随着航空电子的发展,传统的 ARINC429 总线系统因体积大、灵活性差、集成度不高等缺点,已满足不了航空测试、诊断等用途。提出了一种基于 Android 智能手机的 ARINC429 总线系统,手机终端以 Android 为平台开发应用程序,S3C2440 作为硬件控制模块,HI-8583 芯片发送和接收 ARINC429 数据。以无线网络作为传输介质,完成了手机端与硬件控制端之间的控制和数据信息的交互及 S3C2440 对 HI-8583 的驱动控制,允许多用户同时自由接入和退出系统。通过 TcpDump 和 Wireshark 2 个软件的结合利用,验证了网络通信的正确性,最后对系统进行整体验证,保证了系统各个方面设计的可靠性。

关键词: 航空电子系统; ARINC429; Android; WiFi; 总线系统设计

中图分类号: V19 **文献标志码:** A **文章编号:** 1671-637X(2016)05-0104-04

Design of ARINC429 Bus System Using Android Smartphone

LIU Lian-sheng, LIANG Lin-lin

(Civil Aviation University of China, Tianjin 300300, China)

Abstract: Traditional ARINC429 bus system has the drawbacks of large volume, poor flexibility, and low integration level, which can not meet the requirements of aviation testing and diagnostics with the development of avionic system. An ARINC429 bus system based on Android smartphones is proposed, in which Android is used as a platform for mobile client to develop application program, S3C2440 is used as hardware control module, and HI-8583 chip is used for sending and receiving ARINC429 data. With wireless network as the transmission medium, the data information interaction between mobile terminal and hardware control terminal, and the drive control of S3C2440 to HI-8583, are implemented. Multiple users are allowed to access and exit system at the same time. Through the combination of TcpDump and Wireshark, the correctness of the network communication is verified. Finally, the reliability of the system is proved through overall system verification.

Key words: avionic system; ARINC429; Android; WiFi; bus system design

0 引言

ARINC429 是一种航空电子总线,它将飞机的各系统间或系统与设备间通过双绞线互连起来,是数字信息传输的主要路径,也是飞机的神经网络^[1]。传统的 ARINC429 总线系统的设计多为 FPGA 或专用芯片加上中央控制单元和 PC 机,通过 USB 或 PCI 通信来完成^[2],但其硬件体积过大,浪费计算机资源,通信速率不高。随着电子科技的发展,目前 Android 系统的智能手机用户的数量已超过了 PC 机,本文使用 Android

智能手机设计了 ARINC429 总线系统,具有较高的实时性和可靠性,同时支持多用户同时使用该系统。

1 总体设计思路

针对系统本文进行了软件和硬件两方面的设计,如图 1 所示。以嵌入式微处理器 S3C2440 为下位机的控制核心^[3],并通过 USB 接口接入无线网卡与 Android 智能手机进行通信,接收手机终端发来的控制信息并根据控制信息发送或者接收 ARINC429 总线数据。手机作为控制下位机的管理器,提供良好的交互界面,实时与微处理器进行通信。

本系统为 C/S 结构,嵌入式微处理器作为服务器端,监听客户端的连接。手机终端作为客户端,可以随时接入服务器。系统支持多客户端的情况,允许多

收稿日期:2015-05-19

修回日期:2016-01-09

作者简介:刘连生(1960—),男,河南郑州人,教授,硕导,研究方向为航空电子工程。

个 Android 用户同时控制下位机的微处理器而不会互相干扰。用户携带安装有本系统手机端应用程序的 Android 手机进入无线网络范围时,手机客户端会自动搜索网络,识别嵌入式微处理器所在的局域网,并请求连接,成功后,Android 手机与 S3C2440 控制模块就可以通过 Socket 进行通信。

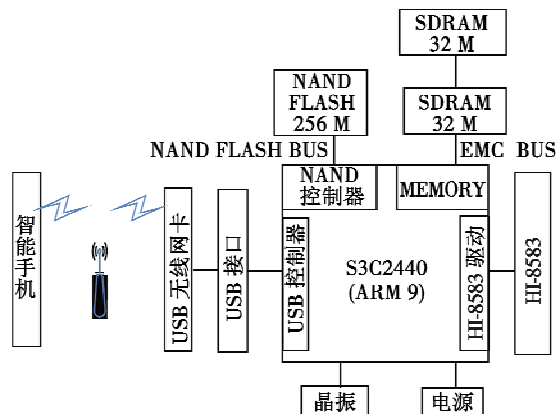


图1 系统总体设计框架图

Fig.1 The diagram of system design

数据存储 (NAND FLASH) 使用的是 K9F2G08, SDRAM 部分使用 2 片 16 位总线宽度的存储芯片组成 32 位总线宽度^[2]。无线网卡是终端网络设备,采用中兴 TL-WN322G+, 通信速率为 100 Mbit/s, 传输速率和有效距离都满足系统要求。这样就构成了一个实时性强、可靠性高、通用性好、操作方便、具有良好的可扩展性的嵌入式 ARINC429 总线系统。

2 Android 智能手机控制终端设计

2.1 用户登录

Android 是 Google 公司开发的基于 Linux 平台开源的手机操作系统^[4]。通过 Socket 编程建立客户端与 WiFi 通道的信息交互,因此在登录界面需要用户选择服务器所属的无线局域网,设置服务器的 IP 地址和端口号。手机终端根据 IP 地址,向服务器提出连接请求,监听到客户端的服务器接受请求后,双方即可以进行通信。当用户打开终端应用程序时,程序用 Wifi-manager 提供的 API 接口来查询 WiFi 的连接及使用情况,如果 WiFi 未打开,会提醒用户打开;当服务器和客户端不在同一个局域网内时,会提醒用户重新进行无线网络的连接。

2.2 发送、接收界面

发送和接收界面主要用来设置 ARINC429 数据的控制信息,如是否连续发送,传输速率为 12.5 kbit/s 或 100 kbit/s,奇偶校验等信息。然后等待用户将要发送的 ARINC429 数据写入,通过 WiFi 通信将数据发送

到微处理器,并且开启一个线程等待处理接收的 ARINC429 数据。

首先设置选项,写入要发送的数据,点击确认发送按钮,程序会首先将控制信息发送出去,初始化 HI-8583,而后将 ARINC429 数据发送给微处理器。接收端一直在等待 ARINC429 数据的到来,图中,先将 ARINC429 数据 0x6FF8AE23 发送出去,使 HI-8583 的发送和接收引脚互连后,应用程序会将接收到的数据显示出来。

服务器与客户端是通过 TCP 协议进行网络通信的,在通信时,必须先由客户端去连接服务器端。用 Socket() 方法创建完套接字,而后通过 connect() 方法与服务器端建立连接并可使用收发函数进行控制命令的数据传输。使用 close() 方法可以断开连接^[5]。客户端应用程序流程如图 2 所示。

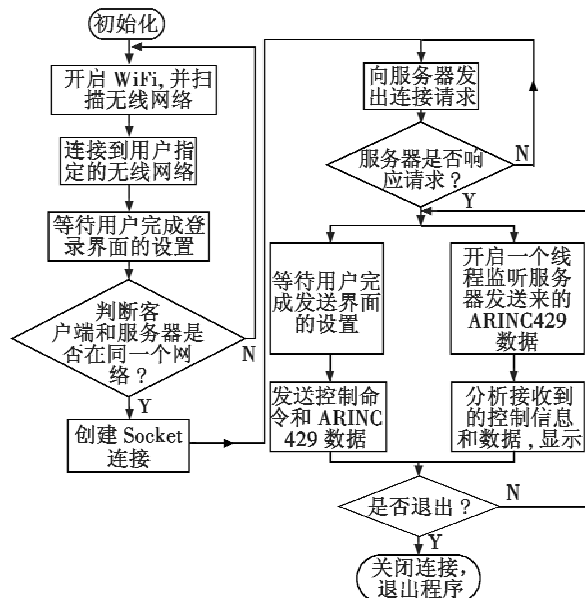


图2 手机客户端程序流程图

Fig.2 The flow chart of mobile client program

3 嵌入式控制模块设计

3.1 嵌入式系统的搭建

系统硬件平台是以 S3C2440 微处理器为核心搭建的,外围接必要的内存和外存^[6],并采用 802.11 标准的中兴 TL-WN322G+ 无线网卡实现 WiFi 网络通信。本系统目的在于实现 ARINC429 总线系统,所以采用“友善之臂”的 bootloader: Superboot2440. bin,内核镜像 zImage_A70,以及文件系统 rootfs_qtopia_qt4. img 搭建 Linux 开发平台,系统中包含了各种常见设备的驱动,支持 USB、串口、SD 卡等。

3.2 HI-8583 驱动程序设计

HI-8583 芯片具有可以自动完成电平转换、自动产

生字与字之间的间隔,具有独立的双向接收和发送接口、串行数据速率可设置、自动完成校验位等功能^[7]。S3C2440 与 HI-8583 硬件连接: HF1, HF2 连接到 S3C2440 的具有中断复用的 GPIO 引脚上 (GPF0, GPF1), 数据总线 BD0-BD15 连接到 S3C2440 的 LDA-TA0-LDATA15, 其他引脚连接到 S3C2440 的普通 GPIO 引脚即可。HI-8583 驱动程序的编写实际是在配置 HI-8583 的时序, 程序中用 S3C2440 的定时器资源 (5 个 16 位的定时器), 以保证时序的准确性, 同时在驱动程序中设置有 2 个缓存区, 发送缓存用来将多个用户想要发送的 ARINC429 数据排成队列, 进行 FIFO 发送。接收缓存用来保存 HI-8583 接收到的 ARINC429 数据。

首先对 HI-8583 进行初始化, 设置其为发送模式, 即向寄存器中写入 31e1, 且在这期间 CWSTR 必须保持至少 80 ns 的低电平。本系统保持了 100 ns, 符合其时序的要求。ARINC429 数据是 32 位, 但是 HI-8583 只有 16 位的数据总线, 因此一次只可以写 16 位, 数据需要分两次写入, 先写入高位。首先 PL1 保持至少 80 ns 的低电平时写入 ARINC429 数据的高 16 位, 而后置 PL1 为高电平。PL2 保持至少 80 ns 的低电平时送入 ARINC429 数据的低 16 位, 而后置 PL2 为高电平。PL1 和 PL2 的低电平时刻至少要间隔 85 ns, 这时 ARINC429 数据已经写入到 HI-8583 中, 最后置 ENTX 为低电平启动 HI-8583 发送数据使能。

驱动程序中首先监视中断信号 HF, 当 HF 出现低电平时, 系统进入中断模式, 开始从 HI-8583 的 FIFO 中取走 ARINC429 数据, HI-8583 只有 16 位数据线, 所以要取两次, 先取高位。当中断出现后置 SEL 为低电平, 而后置 EN 为低电平, 间隔至少是 10 ns, 之后置 EN 为高电平, 读走 ARINC429 数据的高 16 位, EN 高电平至少要保持 60 ns, 数据才可以正确接收。而后再置 SEL 为高电平, EN 为低电平, 读走 ARINC429 数据的低 16 位, EN 的低电平至少要保持 60 ns。

3.3 嵌入式端应用程序设计

服务器端的通信程序是通过套接字编程的方法实现的, 按照用户定义的网络类型, 协议类型和具体的协议标号等参数来定义 Socket() 函数, 建立套接字, 设置可以连接客户端的个数, 本系统设置为 255 个, 通过 bind() 方法将套接字与应用程序端口绑定, 本系统端口号为 8080, 之后定义 lister() 函数监听客户端的请求, 当有客户端发来连接请求时, 调用 accept() 函数来接收请求, 这样服务器就可以和客户端进行通信^[8]。

系统用线程来监听客户端的连接, 为每一个连接上的客户端创建一个 Socket 流。接收 ARINC429 数据程序用线程来等待中断、处理中断, 并将接收到的数据存于缓存中, 图 3 所示为服务器程序流程。

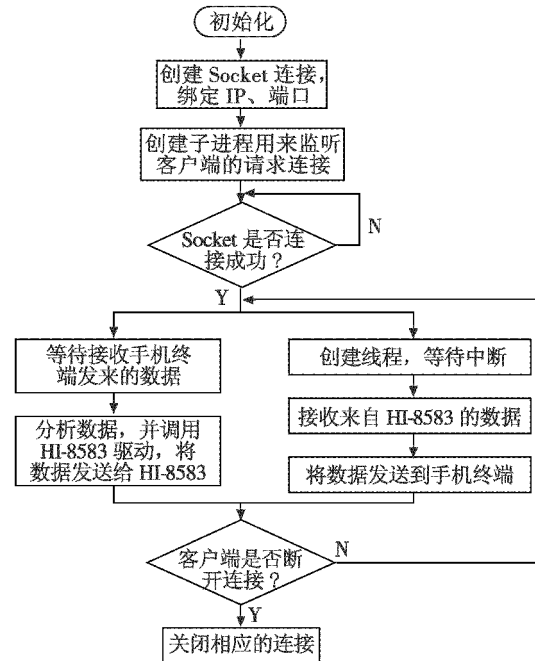


图 3 服务器端应用程序流程图

Fig. 3 The flow chart of server-side application

4 系统功能验证

4.1 网络通信可靠性验证

系统运行时, 允许多个客户端同时接入, 为了验证各个客户端与服务器之间的网络通信无丢包等问题, 数据的采集和分析是不可少的。TcpDump 是 Linux 中强大的网络数据采集分析工具之一。通常是先使用带 -w 参数的 TcpDump 截获数据并保存到文件中, 然后再使用其他程序进行解码分析。这里使用的是 Windows 平台下的 Wireshark 软件来分形 TcpDump 捕捉到的网络包。

移植 TcpDump 完成后在目标板终端执行 ./TcpDump-p-vv-s 0-w ARINC429.pcap & 命令, 让软件运行于后台, 捕捉到的网络包保存在 ARINC429.pcap 文件中。运行服务器端、客户端程序, 用 Wireshark 来分析 ARINC429.pcap 文件。

网络包分析如图 4 所示, 从图中可以看出系统中有 192.168.191.2 与 192.168.191.3 两个用户, 分别向服务器发送了 0x6FF8AE23 与 0x6FF8AE25 两个 ARINC429 数据, 没有丢包等网络问题, 网络各方面性能正常。

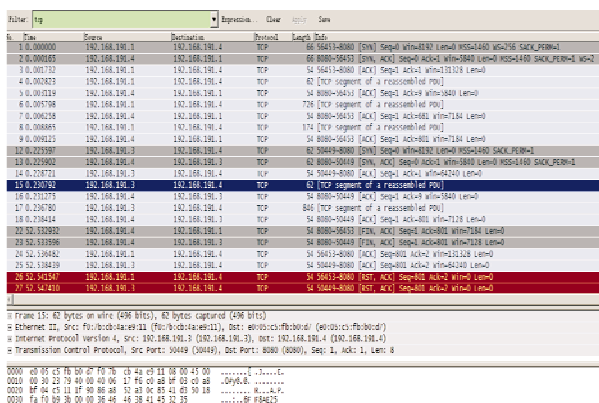


图4 网络包分析图

Fig. 4 The analysis chart of network packet

4.2 系统发送、接收可靠性验证

验证系统网络通信正确后,还需对整个系统进行验证,在手机终端程序中写入要发送的 ARINC429 数据 0x6FF8AE23,点击发送键,分别设置高、低速,结果表明其电压幅值、高速和低速时的频率以及电平上升、下降时间均符合 ARINC429 总线协议规范,如图 5、图 6 所示。

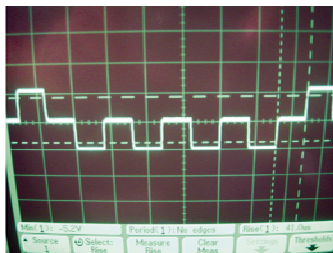


图5 低速时检测到的 ARINC429 数据
Fig. 5 ARINC429 data detected at low speed



图6 高速时检测到的 ARINC429 数据
Fig. 6 ARINC429 data detected at high speed

从图中可以看出系统发送端是正确的。将 HI-8583 的输出和输入引脚互连,观察手机客户端接收界面上接收到的 ARINC429 数据,经测试,接收到的数据为 0x6FF8AE23,由此证实了系统接收端的正确性。

5 结束语

系统首先搭建硬件平台,利用智能手机体积小、便于携带、成本低、系统开放、功能完备等特点代替 PC 机

作为上位机和下位机进行交互通信,用 S3C2440 作为嵌入式平台驱动 USB 无线网卡和 HI-8583 芯片,完成系统的各个部分的功能。软件平台则通过网络通信的 Socket 流来建立,组成了以 S3C2440 为服务器、智能手机为客户端的软件系统,并通过了网络数据分析软件的验证。将智能手机的无线通信技术与控制系统相结合应用于航空领域,性能可靠、携带方便、成本低,同时支持多用户同时在线,为航空电子设备的发展提供了一种新颖的思路。

参考文献

- [1] 支超有. 机载数据总线技术及其应用 [M]. 北京:国防工业出版社,2009. (ZHANG Y. Avionics data bus technology and its applications [M]. Beijing: National Defense Industry Press, 2009.)
- [2] 李寰宇,王勇,刘安. 基于 PCI 的多通道 ARINC429 总线接口卡设计 [J]. 电光与控制,2009,16(2):72-75,83. (LI H Y, WANG Y, LIU A. Design of a multi-channel ARINC429 bus interface board based on PCI [J]. Electronics Optics & Control, 2009, 16(2):72-75, 83.)
- [3] 杨水清,张剑,施云飞,等. ARM 嵌入式 Linux 系统开发技术详解 [M]. 北京:电子工业出版社,2008. (YANG S Q, ZHANG J, SHI Y F, et al. Embedded Linux system development for ARM [M]. Beijing: Publishing House of Electronics Industry, 2008.)
- [4] 韩超,梁泉. Android 系统原理及开发要点详解 [M]. 北京:电子工业出版社,2010. (HAN C, LIANG Q. Android principle and development points comments [M]. Beijing: Publishing House of Electronics Industry, 2010.)
- [5] 王雷,蓝箭,陈雪娟,等. 基于 Android 平台的无线 WiFi 控制方法 [J]. 微型电脑应用,2012,28(7):58-61. (WANG L, LAN J, CHEN X J, et al. Control method based on WiFi in Android [J]. Microcomputer Applications, 2012, 28(7):58-61.)
- [6] 孙弋,张晓刚. 基于嵌入式 Linux 的车载终端的设计与实现 [J]. 微计算机信息,2009(17):86-88. (SUN Y, ZHANG X G. Design and realization of a carrying terminal based on the embedded Linux vehicles monitoring management system [J]. Microcomputer Information, 2009(17):86-88.)
- [7] Holt integrated circuits HI-8583, ARINC429 system on a chip [Z]. 2015.
- [8] 刘平. Android 手机访问服务器的一种数据交互方法 [J]. 电子设计工程,2010,18(9):96-98,102. (LIU P. Effective data interactive method from Android cell phone accessing Web app server [J]. Electronic Design Engineering, 2010, 18(9):96-98, 102.)