

弹内高速 1553B 总线 RT 端软件设计与应用

贾金艳, 陈海峰, 丁炳源, 胡 纯

(北京机电工程研究所, 北京 100074)

摘要: 随着导弹功能性能复杂性的不断提高, 弹载设备间的信息交换量日益增加, 对 1553B 总线系统数据传输速率的要求越来越高; 1 Mbps 1553B 总线已经无法满足现代航空、航天和地面车辆系统应用中各项数据交换的基本需求; 目前, 高速 1553B 总线产品应运而生; 通过对高速 1553B 总线通讯特性进行研究, 结合弹内高速 1553B 总线不同远程终端 (RT) 的通讯需求, 给出了弹内高速 1553B 总线 RT 端软件的中断与查询两种设计方案; 这两种设计方案已成功应用于某系列导弹弹内的各个 RT 端设备, 通过了该导弹武器的研制与定型阶段的各项工程试验验证, 结果表明中断与查询两种设计方案满足该导弹系统的不同 RT 端的应用需求, 具有一定的推广价值。

关键词: 导弹; 高速 1553B 总线; 远程终端; 软件设计

Design and Application of 1553b Bus RT Soft on Missile

Jia Jinyan, Chen Haifeng, Ding Bingyuan, Hu Chun

(Beijing Electro-Mechanical Engineering Institute, Beijing 100074, China)

Abstract: As the complexity of missile increases continually, the communication between the devices in the missile becomes more and more frequent, which requires faster and faster transfer speed of 1553B bus. The traditional 1 M1553B bus is not fast enough to transfer information in modern aeronautics, astronautics and ground vehicle system. High speed 1553B bus appears at present. This paper researched the communication character of high speed 1553B, and proposed two design approaches of interrupt and query for Remote Terminal (RT) software based on the demand of different RTs in the missile. The two approaches were successfully utilized in the RTs of one series missile, and passed various tests. during the development stage, the experimental results demonstrated that the interrupt and query methods met the requirements of the different RTs, which could be used in other missiles.

Keywords: missile; high-speed-1553B bus; RT; soft design

0 引言

1553B 总线是飞机内部时分制指令响应式多路传输数据总线。由其传输速率高, 设备之间连接简单灵活, 噪声容限高, 通信效率高而且可靠。从推出到现在, 经过二三十年的发展, 1553B 总线技术也在不断的改进。目前 1553B 总线已经成为在航空航天领域占统治地位的总线标准^[1-4], 预计在未来十年 1553B 总线还将得到广泛的应用。

现代战争中, 导弹武器已经成为体系化防务对抗系统的一部分。随着导弹功能性能复杂性的不断提高, 弹载设备间的信息交换量日益增加, 对 1553B 总线系统数据传输速率的要求越来越高。目前, 1Mbps 的传输速率已经无法满足现代航空、航天和地面车辆系统应用中各项数据交换的基本需求^[5]。近年来, 国内多家高速 1553B 总线产品应运而生。本文给出弹内高速 1553B 总线 RT 端通讯软件的两种设计方案, 已在某系列导弹武器上应用, 具有一定的推广价值。

1 高速 1553B 总线

1.1 系统组成简介

导弹采用高速 1553B 总线作为弹内骨干通讯网络, 完成导弹发控、测试、飞控等阶段的控制数据和遥测数据通讯功能。

制导计算机作为总线控制器 (BC), 是启动总线数据传输

任务的唯一终端, 负责发送指令字, 参与数据传输, 接收状态响应和监控系统状态。测控系统兼作远程终端 (RT) /总线监视器 (MT), MT 用于监听和收集来自总线上的所有消息, 完成总线上数据信息的记录和分析, 但不参与任何总线数据传输。其它设备作为 RT, 对来自总线控制器的指令做出响应, 按指令要求完成数据传输和其他操作。每个 RT 被指定一个专用总线地址, 从 1 到 30 均可采用, 不采用地址 0 作为 RT 的专用地址, RT 地址 31 为所有远程终端的公用地址。弹内高速 1553B 总线通讯拓扑结构如图 1 所示。

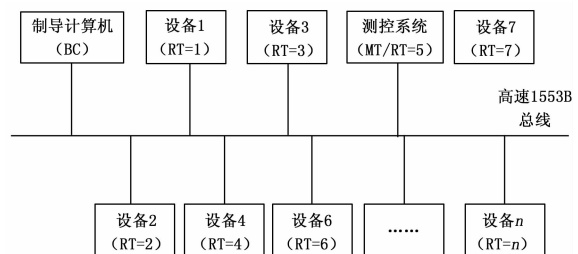


图 1 弹上高速 1553B 总线通讯拓扑结构

1.2 总线消息特性分析

1553B 标准定义了十种消息格式: BC->RT Transfer、RT->BC Transfer、RT->RT Transfer、Mode Code No Data、Tx Mode Code With Data、Rx Mode Code With Data、Broadcast、RT-to-RTs (Broadcast) Transfer、Broadcast Mode Code No Data、Broadcast Mode Code With Data, 实现 BC 到 RT、RT 到 BC、RT 到 RT 以及广播方式和系统控制方

收稿日期: 2015-07-16; 修回日期: 2015-09-11。

作者简介: 贾金艳 (1974-), 女, 辽宁北票人, 高级工程师, 主要从事嵌入式软件设计方向的研究。

式。高速 1553B 总线控制器发出的最小消息间隔为 $1.0\ \mu\text{s}$ (4 M 1553B) / $0.8\ \mu\text{s}$ (5 M 1553B) / $0.7\ \mu\text{s}$ (6 M 1553B)。远程终端响应有效指令字的间隔时间为 $1.0 \sim 3.0\ \mu\text{s}$ (4 M) / $0.8 \sim 2.4\ \mu\text{s}$ (5 M) / $0.7 \sim 2.0\ \mu\text{s}$ (6 M)。总线控制器在一路总线上启动传输后, $6.0\ \mu\text{s}$ 内未收到 RT 端状态字, 则认为远程终端无响应超时。上述最小消息间隔、响应时间、无响应超时 3 个性能指标不同于 1 M1553B, 在 RT 端软件设计中要注意最小消息间隔, 避免处理不当造成消息丢失。

2 传输通讯协议设计

1553B 标准定义的十种消息格式中 BC $\rightarrow$ RT Transfer、Mode Code No Data、Tx Mode Code With Data、Rx Mode Code With Data、Broadcast、Broadcast Mode Code No Data、Broadcast Mode Code With Data 按照协议约定 BC 主动进行, RT $\rightarrow$ BC Transfer、RT $\rightarrow$ RT Transfer、RT $\rightarrow$ RTs (Broadcast) Transfer 通过矢量字方式握手机制。以矢量字握手方式下的 RT $\rightarrow$ BC 通讯传输示例如下:

1) 当 RT 某子地址数据就绪后, 设置矢量字的相应位, 其中矢量字 D0 位 (低) 为 1, 表示子地址 1 数据就绪。D1 位为 1, 表示子地址 2 数据就绪, 依次类推, D15 位为子地址 16 数据就绪。

2) BC 向 RT 发送“方式指令字 (发送)”, RT 返回状态字和一个矢量字。

3) 如果 BC 检查矢量字的相应位已置“1”, 则完成 RT $\rightarrow$ BC 的相应子地址数据传输。

4) BC 取走某一子地址数据后, 由 RT 自动清除矢量字的相应位, 避免 BC 端重复读取该子地址数据。

RT $\rightarrow$ RT Transfer、RT $\rightarrow$ RTs (Broadcast) Transfer 通讯协议同 RT $\rightarrow$ BC Transfer。

3 高速 1553B 远程终端软件设计

3.1 HT-61843GB/L 简介

HT-61843GB/L 高速 1553B 总线控制模块支持 BC/RT/MT3 种工作模式, 软件操作兼容 DDC 公司 BU-61580。HT-61843GB/L 为主机提供了一个内存映射软件接口, 地址空间是 $4\text{ K} \times 16\text{ bit}$ 的内部 RAM, 同时还包括 17 个通用操作寄存器。寄存器定义、RT 存储映射和管理都与 BU-61580 兼容。其强大功能简化了 1553B 总线接口终端复杂协议逻辑的软件设计过程, 其软件设计重点为对总线控制系统初始化配置, 以及应用层消息处理, 而不需要过多关注 1553B 总线的底层传输的设计。

3.2 RT 端软件设计

3.2.1 寄存器初始化

寄存器初始化步骤如下:

1) 设置中断屏蔽寄存器为 0x01, 复位控制器。

Bit00: RESET。

2) 设置配置寄存器 3 为 0x8000, 增强模式。

Bit15: ENHANCED MODE ENABLE。

3) 初始化中断屏蔽寄存器为 0x11。

bit04: RT SUBADDRESS CONTROL WORD EOM。

bit00: END OF MESSAGE。

4) 设置 ConfigReg2 为 0x3816。

bit01: Enhanced RT Memory Management enabled。

bit02: Clear Service Request。

bit04: interrupt status Auto Clear。

bit11: Overwrite Invalid Data。

bit12: Rx SA Double Buffer Enabled。

bit13: Busy Lookup Table Enabled。

5) 设置 ConfigReg3 为 0x801D。

Bit00: ENHANCED MODE CODE HANDLING。

bit02: RT FAIL Flag WRAP Enable。

bit03: Busy Transfer Disable。

bit04: Illegal Rx Transfer Disable。

bit15: Set Enhanced Mode。

6) ConfigReg4 = 0x2000。

bit13: Mode Command Override BUSY。

7) 设置 ConfigReg5 为 0x0900。

bit8: GAP Check Enabled。

bit11: Expanded Crossing Enabled。

3.2.2 内存初始化

内存初始化步骤如下:

1) 初始化内存, 写 0 到整个内存区。

2) 初始化接收子地址查询指针表 140 H $\sim$ 15 FH; 双缓冲, 每个子地址指针递增 40 H。

3) 初始化发送子地址查询指针表 160 H $\sim$ 17 FH; 单缓冲, 每个子地址指针递增 20 H。

4) 初始化子地址控制字表 1A0H $\sim$ 1BFH 为 C200H。15 (MSB) 接收: 双缓冲使能; 14 发送: 消息结束中断; 9 接收: 消息结束中断。

内存初始完成后设置 ConfigReg1 为 0x8f80;

Set RT Mode, bit13 为 0; Current Area is A。

3.2.3 RT 端处理消息程序设计

1) RT 端采用中断方式处理消息:

该方式下总线控制器模块的 EOM 接到 RT 端处理器的外部中断管脚。当消息传输完毕 (EOM; End Of Message) 会触发 RT 端 DSP 外部中断, 通过设计合理的中断服务程序来处理消息: 无论 RT 接收或发送消息, 均需要及时地提取当前堆栈指针指向对应的消息块状态字的相应信息 (发送/接收指令、子地址), 若是 RT 发送消息, 还需要清除对应子地址的矢量字相应位, 避免 BC 端重复读取消息; 若是 RT 接收, 则还需要提取接收字长、接收数据块指针。

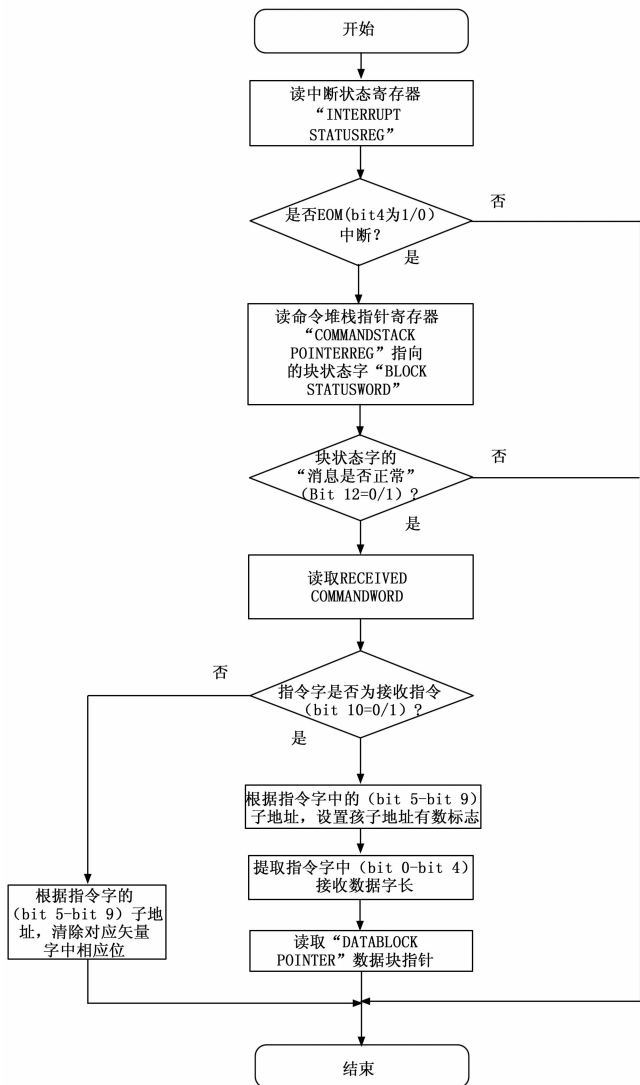
中断服务程序设计如图 2 所示。

2) RT 端采用查询方式处理消息:

高速总线消息最小间隔为 $1.0\ \mu\text{s}$ (4 M 1553B) / $0.8\ \mu\text{s}$ (5 M 1553B) / $0.7\ \mu\text{s}$ (6 M 1553B), 对于消息密集的 RT 设备, 若还采用中断处理方式时, 当第 1 个消息的 EOM 触发进入中断服务程序开始处理第 1 个消息, 而第 2 个消息的命令堆栈寄存器的内容是在第 2 个消息 SOM 序列尾部更新, 这就意味着第 1 个消息的中断服务程序需要在第二个消息堆栈指针更新前取走第 1 个消息的堆栈指针, 否则第 1 个消息的中断服务程序直接取到第 2 个消息的堆栈指针, 即丢失了第 1 个消息。为防止丢失消息, 需采用查询的方式来处理消息。查询方式下 RT 端消息处理程序设计如图 3 所示。

查询方式的设计要点如下:

1) 通过 RT 存储映射的 Stack Pointer A (B) 0x100 (104) 和命令堆栈指针寄存器“COMMAND STACK POINTER



```

graph TD
    Start([开始]) --> Init[自定义全局变量已处理过的消息指针  
RT_STACKPT (初始为0)]
    Init --> ReadPT[读取指向消息描述块的指针的值赋给  
RT_MessagePT]
    ReadPT --> Decision1{RT_MessagePT  
!=RT_STACKPT  
?}
    Decision1 -- 否 --> End([结束])
    Decision1 -- 是 --> ReadBlock[读取指针RT_STACKPT的所指内容 (块状态字BLOCK  
STATUS WORD)]
    ReadBlock --> Decision2{块状态  
字的EOM是否产生  
(Bit15=1/0)}
    Decision2 -- 否 --> Decision3{RT_MessagePT=  
((RT_STACKPT+4)  
&0fff)?}
    Decision3 -- 是 --> Note1[新的消息已启动了, RT_STACKPT  
消息EOM还没有产生, 则丢弃  
RT_STACKPT的消息;  
RT_STACKPT=(RT_STACKPT+4)  
&0xFF continue]
    Decision3 -- 否 --> Note2[Break;说明此消息正在  
传输中, 下一查询周期  
再进行处理]
    Note1 --> End
    Note2 --> End
    Decision2 -- 是 --> Decision4{块状态字  
的消息是否正常  
(Bit12=0/1)?}
    Decision4 -- 否 --> End
    Decision4 -- 是 --> ReadCmd[读取指令字RECEIVED  
COMMAND WORD]
    ReadCmd --> Decision5{指令字是否为接收指令  
字 (bit10=0/1)?}
    Decision5 -- 否 --> Note3[根据指令字的  
(bit5-bit9)子地址, 消除对应矢  
量字中相应位]
    Decision5 -- 是 --> Note4[根据指令字中的  
(bit5-bit9)子地址,  
设置孩子子地址有数标志]
    Note3 --> Decision1
    Note4 --> Extract[提取指令字中 (bit0-  
bit4)接收数据字长]
    Extract --> ReadPtr[读取“DATA BLOCK  
POINTER”数据块指针]
    ReadPtr --> UpdatePT[RT_STACKPT=(RT_STACKPT+  
4)&0xFF]
    UpdatePT --> Decision1
  
```

(下转第 167 页)

在 Qt 应用程序中提供多线程是非常简单的: 只需要子类化 QThread 并且重新实现其 run () 函数就可以了, 这样, 只需要在 run 函数中实现数据的解算。但实际应用中, 试图在子线程中调用 GUI 类是不可能的, 也就是说解算出来的数据不能直接显示在界面上。Qt 特有的跨线程的信号和槽机制打开了子线程和 GUI 线程通讯的方便之门^[4], 通常情况下, 信号和槽机制可以同步操作, 这就意味着在子线程解算完数据后, 采用信号与槽, 发射信号, GUI 界面上就可以显示结果了。

获得一个完整的全帧后, 就进入解析线程, 结合飞机的格式格栅信息, 解算数据, 具体的解算流程图如图 7 所示。

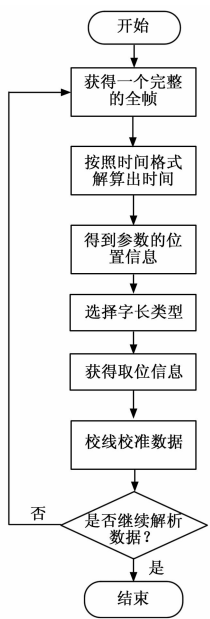


图 7 实时解算流程图

在获得实时的飞行数据后, 就可以在主界面上显示数据, 同时保存数据, 把部分关键参数数据发送给客户端子系统用于监控, 整个软件的设计流程图如图 8 所示。

3 软件应用

本软件集成了实时数据采集、显示、存储、网络发送、图形显示、数据回放等功能。既满足了遥测数据处理的实时性要求, 又能查看部分参数值和时间历程曲线。

4 结论

本文提出了一种基于 RTR 接收机的网络遥测数据实时处理的方法, 经过大量飞行试验证明, 本软件运行稳定和数据完整可靠, 不仅完全满足普通科目实时监控的要求, 而且经过改进满足颤振等特殊科目参数高采样的监控需要; 另外, 与现有实时数据处理方式相比, 具有配置简单, 使用灵活, 可以安装在局域网中的任意一台电脑上, 就可以构建一套遥测实时数据

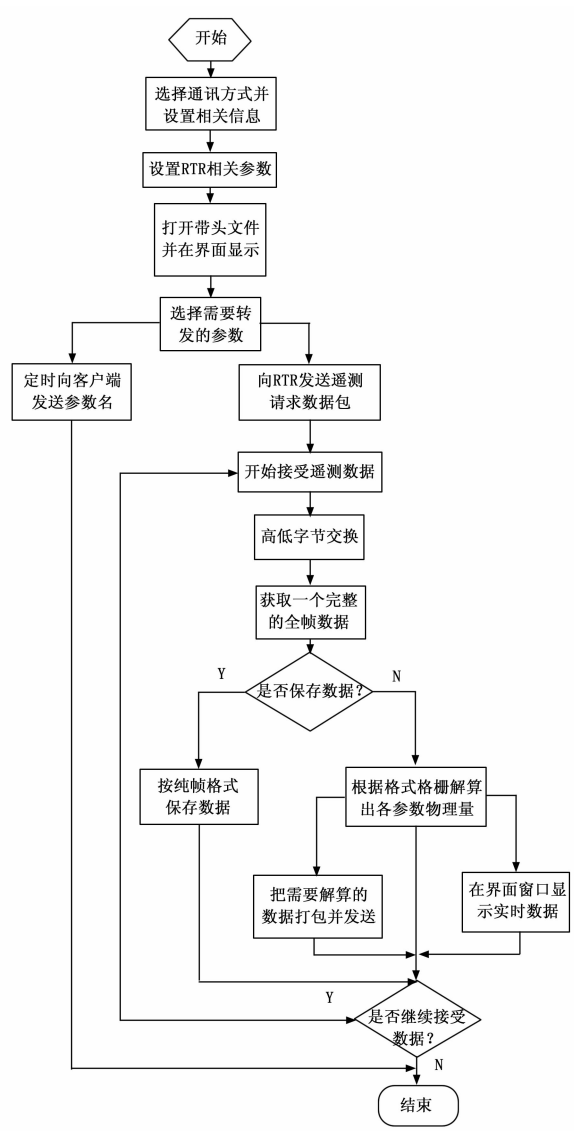


图 8 基于 RTR 的遥测数据实时处理软件逻辑流程图

处理系统, 更加符合监控资源网络化、分布式的发展趋势。

参考文献:

[1] 高陪先, 张俊芳, 乔东峰. 遥测实时系统深层次设计缺陷分析与测试 [J]. 电子测量技术, 2007 (2): 182-183.

[2] IN-SNEC, STI 100013_RTR [Z]. 2009.

[3] 闫锋欣, 曾泉人, 张志强. C++ GUI Qt 4 编程 (第二版) [M]. 北京: 电子工业出版社, 2011.

[4] 蔡志明, 卢传富, 李立夏. 精通 Qt4 编程 (第 2 版) [M]. 北京: 电子工业出版社, 2011.

[3] 刘士全, 等. 1553B 总线应用发展研究 [J]. 电子与封装, 2013, 13 (12): 12-15.

[4] 陈 亮, 等. 基于 1553B 协议的导弹控制系统信息流设计探讨 [J]. 计算机测量与控制, 2012, 20 (8): 2180-2182.

[5] 杨 兵, 等. 一种 10 Mb/s1553B 总线接口的设计 [J]. 微电子学, 2012, 42 (6): 783-787.

(上接第 164 页)

参考文献:

[1] 陈春燕, 等. 基于 1553 总线的飞行器电气综合系统管理设计 [J]. 计算机测量与控制, 2014, 22 (6): 1865-1867.

[2] 徐贵贤. 1553B 总线简介及其实现 [J]. 通信技术, 2011, 44 (5): 166-168.