

基于 RS485 总线的起落架控制系统 实时通信算法实现

邓志云, 路红伟

(中航飞机起落架有限责任公司, 长沙 410200)

摘要: 以 AG600 水陆两栖飞机的起落架系统综合试验平台为基础, 针对多路传感器信息采集与传输、主备单元之间信息重构、逻辑解算和驱动控制的信息共享需求, 组建了一种 RS485 主从式通信网络, 实现了以帧为单位的数据解析算法, 并在此基础上进行了以帧元为单位的算法优化设计和实现; 试验验证表明, 该算法具备较好的实时性, 满足起落架控制系统的实时通信控制需求。

关键词: 起落架控制; 实时通信; RS485 总线; 帧元; 数据解析算法

Algorithm Implemented of Realtime Communication for Landing Gear Control System Based on RS485 Bus

Deng Zhiyun, Lu Hongwei

(AVIC Landing Gear Advanced Manufacturing Corp., Changsha 410200, China)

Abstract: Based on the platform of AG600 amphibious landing gear system, a kind of RS485 master-slave network was set up to meet the information sharing needs of information acquisition and transmission of multiple sensors, information reconfiguration between master and slave units, logic solution and drive control. From the communication network, the data analysis algorithm based on the frame is realized, and on this basis, the algorithm optimization design and implementation based on the frame element are carried out. Experimental results show that the algorithm has good real-time performance and meets the real-time communication control requirements of the landing gear control system.

Keywords: landing gear control; realtime communication; RS485 bus; frame element; data analysis algorithm

0 引言

随着全新的多电/全电飞机技术的深入研究及其在多种机型中的验证与实际应用, 推动了飞机起落架系统的收放装置、前轮转弯以及机轮刹车等重要功能部件从传统的液压机式控制向数字电子控制的方式转变。数字电子控制方式有效的实现了飞机重量减负, 同时也提升了起落架控制系统的响应速率和灵活性。研究实现具有高可靠、高安全性的机载起落架控制系统成为迫切需要解决的问题。

数字电子控制系统由于其电子器件固有的失效概率, 在实际系统的设计中主要采用功能部件的冗余部分及有效管理来弥补其不足。系统工作时, 冗余部件分别正常的独立工作, 任意时刻仅有一个部件完成系统输出控制。冗余部件间能够对彼此间的重要信息进行共享、监控及功能故障时重构, 必然要求有高效、可靠的数据通信作为基础支撑。

RS485 作为一种串行通信接口, 具有传输距离较长、速度较高、低成本、可靠度高等特点, 已在智能管理^[1]、在

线控制^[2]等众多领域中都有着广泛的应用^[3]。RS485 提供并保障了硬件层面 Bit 位数据传输的可靠性, 应用系统则可以在利用 RS485 总线组建主从式或总线式通信网络的基础上, 再进行应用逻辑层面的通信协议/算法的设计和实现, 可以有效提高应用系统的灵活性, 同时也满足了实际应用需求的便捷性。

本文以 AG600 水陆两栖飞机的起落架系统综合试验平台为基础, 针对多路传感器信息采集与传输、主备单元之间信息重构、逻辑解算和驱动控制的信息共享需求, 组建了一种 RS485 主从式通信网络, 实现了以帧为单位的数据解析算法, 并在此基础上进行了以帧元为单位的算法优化设计和实现。试验验证表明, 该算法具备较好的实时性, 满足起落架控制系统的实时通信控制需求。

1 控制系统总体架构

起落架控制系统总体架构设计如图 1 所示。

控制系统主要由主控单元(主控 1 路、备控 1 路)、驱动单元(前起驱动主驱 1 路、前起驱动备驱 1 路、左主起驱动主驱 1 路、左主起驱动备驱 1 路、右主起驱动主驱 1 路、右主起驱动备驱 1 路)共 8 路组成。

1.1 主控单元

作为控制系统的核心调度部件, 主控单元实现的主要

收稿日期:2018-03-02; 修回日期:2018-04-04。

作者简介:邓志云(1975-),男,高级工程师,硕士,主要从事嵌入式系统控制,飞机起落架的收放与转弯控制方向的研究。

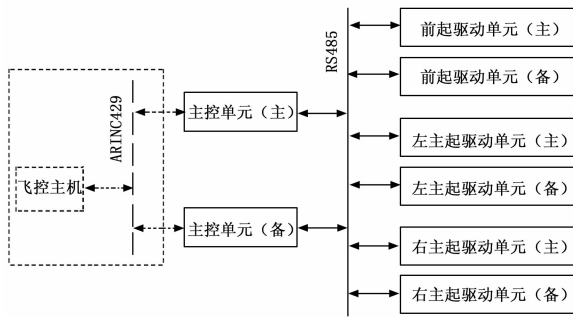


图 1 起落架控制系统总体架构

功能及工作原理：

- 1) 负责接收和响应飞控主机下达的收放控制及系统工作状态监控指令；
- 2) 将接收的控制指令转化为实际需要执行的驱动指令，下达给各相关驱动单元；
- 3) 接收各驱动单元反馈的逻辑解算及状态信息；
- 4) 对各驱动单元的反馈信息进行综合，执行起落架系统级控制逻辑解算；
- 5) 将飞控主机的指令执行及控制系统的健康状态通过 ARINC429 总线反馈给飞控主机；
- 6) 默认状态下，主控单元正常工作输出，备控单元正常工作但不输出控制信号，仅在主控单元出现故障时才接管输出控制。

1.2 驱动单元

作为控制系统的执行驱动部件，驱动单元实现的主要功能及工作原理：

- 1) 负责接收主控单元下达的驱动指令和单元工作状态监控指令；
- 2) 将接收的驱动指令转化为实际需要执行的驱动指令；
- 3) 负责完成所连接的位置传感器信息采集及传感器校准，连接的传感器包括前起驱动主/备单元各 7 路，左主起驱动主/备单元各 6 路、右主起驱动主/备单元各 6 路，共 38 路 ($7 \times 2 + 6 \times 2 + 6 \times 2$)；
- 4) 进行驱动单元级逻辑解算，驱动动作部件完成驱动功能；
- 5) 将主控单元下达指令的执行情况及驱动单元自身的健康状态通过 RS485 总线反馈给主控单元；
- 6) 默认状态下，主驱单元正常工作输出，备驱单元正常工作但不输出控制信号，仅在主驱单元出现故障时才接管输出控制。

1.3 数据通讯与共享

- 1) 主控单元与飞控主机之间通过 ARINC429 总线（采用总线及通信协议由飞控主机方确定）进行通信，具体的 ARINC429 总线信息可参考文献 [4] 做进一步的详细了解；
- 2) 主控单元与各驱动单元通过 RS485 总线进行控制命令（包括起落架收指令、起落架放指令、应急放指令）及

健康状态（包括传感器工作状态、系统上电自检 PBIT、周期/连续自检 CBIT 等）以及系统告警指示等信息的传输；

3) 主控单元和驱动单元各自的主、备单元之间通过共享 RAM 方式进行信息重构且互为备份；

4) 系统设置主位置（硬件板块插接位置）、单元工作有效两个状态标识位用以识别主/备单元，当系统识别出两个标识位都有效时才确定其为主单元，否则识别为备份单元，具体的有效性由相应的 CPLD 硬件仲裁确定。

2 通信算法设计

对于应用系统而言，RS485 串行通讯标准定义了电压、阻抗等内容，但不对软件协议给予定义，它通过对起始、停止电平的检测与转换，加上奇偶校验以及硬件握手信号等，实现并保障了通信的可靠性，其提供服务的仅仅是面向 Bit 位的基础通信，要实现应用系统的业务逻辑，特别是在主从式、总线型通信架构中多设备之间进行通信时，必然需要通过自定义的通信协议来实现，即本文所指的面向应用系统业务逻辑的通信算法。

2.1 帧结构定义

针对起落架控制系统的通信需求，每次通信的数据长度为 15 个字节（包括数据的头、尾标识信息），本文在参考文献 [5-7] 中帧格式设计的基础上，简化设计了如图 2 所示固定长度的帧结构。

帧头	目的 ID	源 ID	传输数据	控制位	帧尾
----	-------	------	------	-----	----

图 2 帧结构定义

帧结构主要由 6 部分组成：

- 1) 帧头：表示一帧数据的开始，用于同步控制，系统默认设定为 0xAA，长度为 1 字节；
- 2) 目的 ID：表示各主备单元分配的 ID 号，表明该帧数据的接收者，取值范围 (0~7, 255)，其中 255 为广播 ID，仅由主控单元发起单元 BIT、故障复位等特殊指令时使用，长度为 1 字节；
- 3) 源 ID：表示各主备单元分配的 ID 号，表明该帧数据的发送者，取值范围 (0~7)，长度为 1 字节；
- 4) 传输数据：内容为要传输的数据，长度固定为 10 个字节；
- 5) 控制位：表示该源 ID 发送的数据是控制输出/备份输出的标识，长度为 1 字节；
- 6) 帧尾：表示一帧数据的结尾，系统默认设定为 0x55，长度为 1 字节。

2.2 单帧解析算法

所谓单帧解析算法是指在待解析数据中同时存在数据头、数据尾且数据长度符合帧长度要求时，即为解析出完整的数据帧。

针对系统中每帧数据长度为 15 个字节，采用循环缓冲区结构（数据缓冲区大小 64 字节，最多同时缓存四帧数

据)实现接收数据的存储。设计使用的数据指针包括: 新数据指针 NIdx、数据访问指针 AIdx, 帧有效标识 Fflag (初始时为假)。

1) 数据接收采用中断方式, 流程如图 3 所示, 基本思想如下。

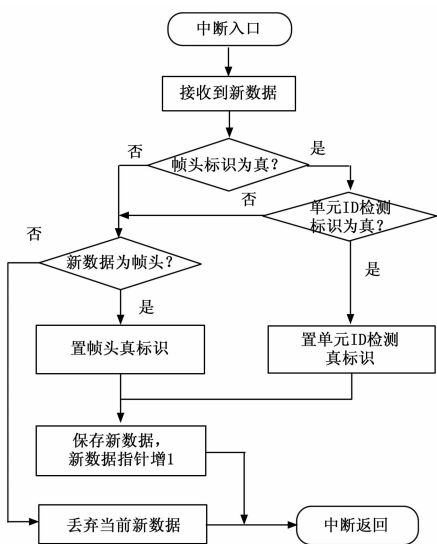


图 3 中断数据接收流程

a) 接收到新数据, 判断帧头标识, 为假转到 b), 为真转到 c);

b) 判断新数据是否为帧头, 是则置帧头标识为真, 转到 d), 否则转到 e);

c) 判断单元 ID 检测标识, 为假则判断该数据是否与当前单元 ID 或为 255, 是则单元 ID 检测标识为真, 转到 d), 否则转到 b);

d) 将新数据添加到数据缓冲区 NIdx 位置, NIdx 增 1, 中断返回;

e) 丢弃当前新数据, 中断返回。

2) 单帧解码算法采用定时器方式周期处理, 流程如图 4 所示, 基本思想是:

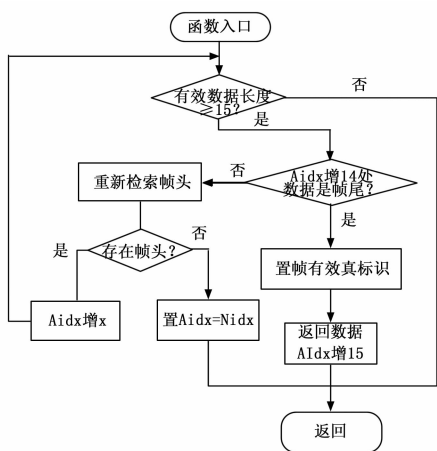


图 4 单帧解析流程

a) 判断缓冲区的有效数据长度 ≥ 15 (帧长)? 是则转

到 b), 否则转到 e);

b) 判断数据缓冲区 AIdx 增 14 处数据是否为帧尾标识, 是则转到 c), 否则转到 d);

c) 置 Fflag 为真, 返回实际数据, AIdx 增 15, 转到 e);

d) 置 Fflag 为假, 在数据缓冲区中检索帧头, 存在则置 AIdx 增 x (x 为帧头位置计数), 转到 a), 否则置 AIdx 为 NIdx, 转到 e);

e) 函数退出。

注: 算法中增 x 操作是指数据加 x 后取模 64 的操作。

2.3 通讯策略

起落架控制系统中采用主控单元接收飞控机的指令 (指令频度 10 Hz), 再下达给驱动单元执行的模式工作 (控制频度 20 Hz)。控制系统设定任意一次通讯均由主控单元 (主) 发起, 驱动单元 (主、备) 应答反馈的时间片轮转方式, 主要包括:

1) 主控单元 (主) 向驱动单元 (主/备) 发起通信指令;

2) 驱动单元 (主/备) 接收到主控单元 (主) 的通信指令后, 进行信息打包并反馈。

采用时间片轮转的主从应答方式, 避免了通信总线的访问冲突。

3 算法优化设计

3.1 单帧解析算法分析

采用时间片轮转的主从应答方式, 带来了系统一定效率的损失。对主控单元而言, 单次完整的通讯涉及驱动单元的主/备各 3 个共 6 次通信, 按控制频度 20 Hz (50 ms)、通信 3 ms/时间片来计算, 用于通信的 CPU 实际占用率为 $6 \times 3 / 50 \times 100\% = 36\%$, 对于系统本身的实时性要求而言, 能够满足实际应用要求。

由单帧解析算法可知, 算法处理的是定长数据, 把整个通信数据的处理分成两个环节: 串口中断数据接收和目的 ID 验证、帧内传输数据的验证提取。

改进思路是: 将数据验证提取归并到串口数据接收的同时进行, 即数据接收的同时进行帧元数据的验证提取。此外, 为了提高算法的灵活性, 将定长数据变更为变长数据, 同时加强了帧数据的有效性校验方式。

3.2 变长帧结构定义

在图 2 中帧结构定义的基础上, 设计的变长帧结构定义如图 5 所示, 由 7 部分组成。

帧头	目的 ID	源 ID	数据长度	传输数据	控制位	帧尾
----	-------	------	------	------	-----	----

图 5 变长帧结构定义

1) 数据长度: 表示该帧传输数据的字节数, 长度为 1 字节;

2) 帧尾: 将系统默认设定的 0x55 变更为传输数据的

校验和 (累加, 取反, 加 1, 取低 7 位; 或为数据字节的异或操作), 长度为 1 字节。

3) 其余各字段与图 2 中所述的含义相同。

3.3 帧元解析算法

帧元解析算法采用串口中断函数实现, 在串口数据接收的同时进行帧元数据的校验和提取, 算法用到的数据结构与单帧解析算法相同, 当前验证标识 Vtag 初始化为帧头, 流程如图 6 所示。

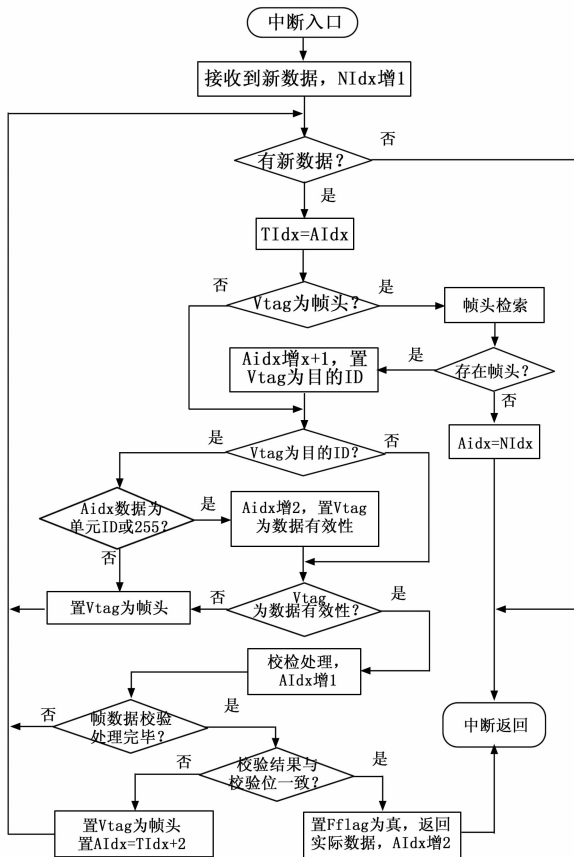


图 6 帧元解析算法流程

基本思想是:

- 接收到新数据, 将其添加到数据缓冲区 NIdx 位置, NIdx 增 1;
- 判断数据缓冲区是否有新数据 ($AIdx \neq NIdx$), 有则转到 c), 否则转到 d);
- 临时存储 AIdx 到 TIdx;
- 判断 Vtag 是否为帧头, 是则转到 e), 否则转到 f);
- 在数据缓冲区中检索帧头, 存在则置 AIdx 增 $x+1$ (x 为帧头位置计数), 置 Vtag 为目的 ID, 转到 f), 否则置 AIdx 为 NIdx, 转到 d);
- 判断 Vtag 是否为目的 ID, 是则转到 g), 否则转到 h);
- 判断数据缓冲区 AIdx 位置数据是否为本单元 ID 或 255, 是则置 Vtag 为数据有效性, AIdx 增 2, 转到 h), 否则置 Vtag 为帧头, 转到 b);

h) 判断 Vtag 是否为数据有效性, 是则转到 i), 否则置 Vtag 为帧头, 转到 b);

i) 按照设定的校验方式对当前数据进行校验处理, AIdx 增 1, 判断是否完成该帧中指定数据长度的数据, 是则转到 j), 否则转到 b);

j) 判断实际校验结果与校验位是否一致, 是则转到 k), 否则置 Vtag 为帧头, 置 AIdx = TIdx + 2, 转到 b);

k) 置 Fflag 为真, 返回实际数据, AIdx 增 2;

l) 中断返回。

3.4 试验验证

试验验证以 AG600 水陆两栖飞机起落架系统的综合试验平台为基础, 由主控、驱动共 8 个单元的冗余控制结构组成, 组建了 RS485 主从式通信网络, 采集处理的信号包括传感器 38 路、电磁阀 3 路驱动控制、前起落架和左右主起落架共 3 路电机驱动控制信号以及各主控和驱动单元的电压、CPU 工作状态、流程关键环节的监测及系统周期 BIT 等。

对主控单元而言, 单次完整的通讯涉及驱动单元的主/备各 3 个共 6 次通信。

1) 系统控制频度设定为 20 Hz (50 ms 周期) 时:

a) 单帧解析算法采用通信 3 ms/时间片轮转, 用于通信的 CPU 实际占用率为 $6 \times 3 / 50 \times 100\% = 36\%$, 消耗时间实际为固定时间片轮转所需间隔周期;

b) 帧元解析算法用于通信的时间主要为中断响应和处理时间, 其消耗的时间为指令取指、译码及存贮等指令执行周期, 无时间片轮转等待周期, 实时性自然好得多。

经在实际的硬件平台中加以验证比较后, 两种算法均满足实时性要求。

2) 系统控制频度设定为 50 Hz (20 ms 周期) 时:

a) 单帧解析算法采用通信改为 2 ms/时间片轮转, 用于通信的 CPU 实际占用率为 $6 \times 2 / 20 \times 100\% = 60\%$, 严重影响了传感器数据采集、电机驱动控制的周期执行, 系统实时性无法保证;

b) 帧元解析算法保持性能不变, 满足实时性要求。

经在实际的硬件平台中加以验证比较后, 帧元解析算法出现数据包丢失, 系统控制响应慢等现象, 帧元解析算法保持了较好的实时性。

显然, 从两种算法的实现思想进行分析比较, 帧元解析算法采用固定时间片、固定数据长度, 且数据通信和数据提取处理相互单独, 顺序进行, 系统效率过多的消耗在时间片轮转等待中, 而帧元解析算法则采用在数据接收的同时完成了数据的验证提取, 通讯时间基本上是物理链路所需的信息传输时间, 且两次通信间无等待间隔, 对传感器数据采集、电机驱动控制的周期执行无影响, 系统实时性较好, 满足系统裕度要求。

4 结论

本文通过以 AG600 水陆两栖飞机起落架系统的综合试

(下转第 83 页)