

小卫星设计中软件重注入的关键技术研究

周 健 段登平 王建宇

(中国科学院上海技术物理研究所 上海 200083)

摘 要

针对小卫星的一体化设计思想,采用 CCSDS 测控体制,星上采用 CAN 总线的分布网络,对小卫星软件重注入的课题做了深入的研究。

关键词 软件重注—分布式系统—分包遥测和遥控—CAN 总线

1 引言

小卫星是航天高技术发展的产物,是采用现代高新技术研制的具有高功能密度的现代小卫星。现代小卫星研制过程与传统的大卫星的不同之处在于一体化设计思想的引进。由于体积、功耗等的限制,它无法像大卫星一样通过增加硬件备份来提高可靠性,而是从整个系统设计角度出发,通过功能上的备份达到可靠性的要求。当某一模块失效时,通过故障诊断措施加以隔离,代之以不同的模块恢复系统的运行或降级运行,这就是系统重构的概念。在重构的过程中,不可避免地伴随着执行软件的改变,如何实现星上软件的重注入也就成了必要的课题。据报道, NASA 和欧洲空间局研制的小卫星多数具有软件重注入功能。英国的 SURREY 公司将“在轨注入软件,完成系统加载”作为其商业化小卫星公用平台的主要特色。但对具体的实现方法却少有提及。

小卫星的一体化设计思想带来的是系统的模块化和标准化,包括测控体制和星上测控网络的标准化,这些已成为现代小卫星的发展趋势。本文将以此为出发点,结合某型号的小卫星,采用 CCSDS 测控体制对软件重注入的研究课题做较深入的探讨。

2 系统描述

本文研究的小卫星采用分布式管理的方案,各分系统间利用 CAN 总线实现卫星内部的信息交换。CAN 网络包括星务主机和若干 CAN 网络节点。星务主机负责收集遥测数据和分发控制指令及遥控指令,实现总体的功能调度。星务主机和通信系统间还有一条用于传送测控数据的专线。

程序注入的过程是:首先,地面站通过星地通信链路(一般是测控信道)在卫星飞抵区域上传注入程序,卫星上的通信系统接收后,利用专线传给星务主机。星务主机将信息帧解包后识别出是程序数据,根据其目的地址远程控制指定的 CAN 节点完成程序加载。

2001-08-31 收到原稿, 2002-05-12 收到修定稿

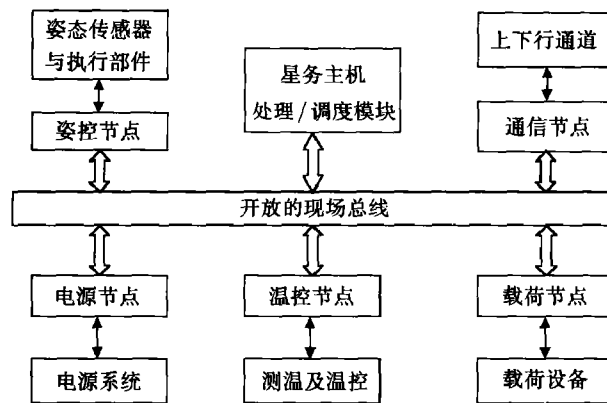


图 1 小卫星分布式管理系统组成框图

Fig.1 On-board data handling system with distributed architecture

程序注入要求非常高的可靠性,为了保证两段传输过程的无差错,测控采用的分包遥测遥控体制和星上系统互联通信的 CAN 总线各自有其独立的差错控制协议。同时,我们还增加了两段过程的应答措施,这样就实现了 3 层的差错控制,以避免传输过程中程序数据的损失。

3 分包测控体制解决信道复用和实时性的问题

当前,遥测遥控的观念已发生变革,简化接口并使之标准化是未来测控体制的方向。国际空间数据系统咨询委员会 (CCSDS) 已为此制定了相应的分包遥测和遥控标准^[1],英国 SURREY 公司已在 UoSAT 系列小卫星上逐步采用分包技术,在国内,“实践五号”小卫星上也已成功运用了这一技术。

分包遥测的基本思想是,每个被测参数按一个测量过程来测量处理和包装,加上少数字节的包头包尾,利用逻辑上的虚拟信道共享同一物理信道。所谓虚拟信道,是一种多信源多用户共享同一物理信道的传输机制,属于时分复用,在一段时间片内仅有一个虚拟信道的数据被物理信道所传送,这是现代分包遥测的关键。虚拟信道与遥测数据流程图如图 2 所示。

遥控任务涉及全星,除常规遥控指令(称之为直接遥控指令)外,还设置大量的间接遥控指令。分包遥控采用分包遥测的思想,将间接指令按包头和多组应用过程指令进行组合实施遥控。这样,复合型的遥控指令系统利用同一物理信道,可以完成常规遥控功能,也可完成分包遥控功能。分包遥控从逻辑上分为 7 层:物理层、信道编码层、传送层、分段层、分包层、系统管理层和应用过程层^[2]。

分包遥控的遥控信源之间是独立的,也与信道独立无关,遥控信源包和信道传输帧之间通过分段层来完成匹配。在分段层中,利用多达 64 个分段过程,将遥控包分段,段长最大为 250 B,用时分复用 (TDM) 合成一个虚拟信道。在传输层中每 250 B 包装成一个传送帧格式。在传输层中,考虑到遥控信息的传输可靠性比遥测高,为保证无差错传输,采

用自动请求重发 (ARQ) 技术. 为进一步提高遥控数据传输的可靠性, 在编码层中将收到的传输帧分解成 $N = 32/40/48/56$ 位的节, 加上 (63,56) BCH 奇偶检错码后称为码块. 编码层中将所有码块集合起来, 加上帧头和帧尾, 构成命令链路传输单元 (CLTU)^[2] 送到物理层对载波进行调制. CCSDS 的分包遥控不但能够保证无码位差错传输, 而且能保证特定顺序, 遥控数据无遗漏、无重复地传输.

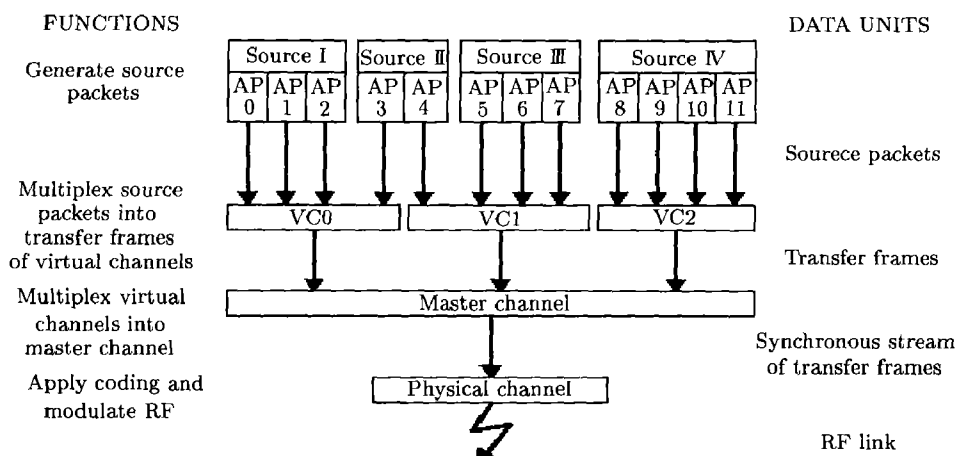


图 2 遥测数据流与虚拟信道的关系图

Fig.2 Telemetry data flow

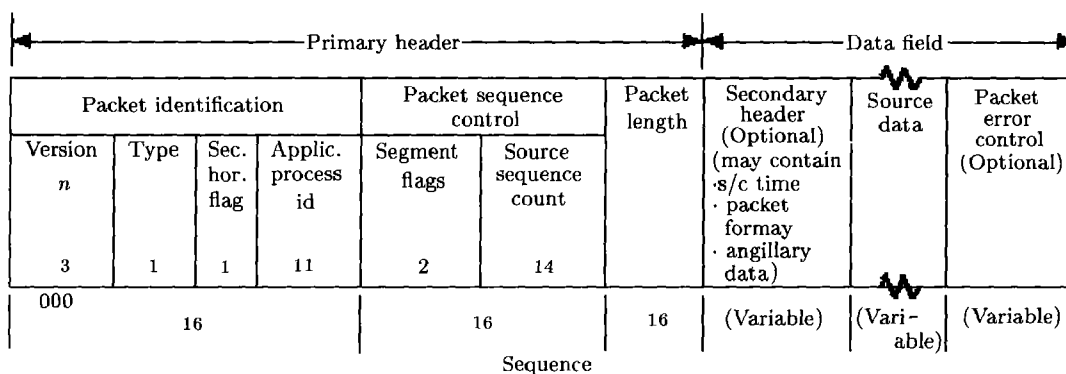


图 3 遥控包标准数据格式

Fig.3 Source packet format

在应用中, 遥控指令和注入程序做为独立的遥控信源包在分段层中和信道传输帧完成匹配. 利用分包遥控, 能够实现对星上过程和设备的高可靠控制, 简化用户设备大量的遥控指令物理接口, 变为统一的数据包接口, 实现多任务的远程透明操作控制.

分包遥控系统必须能够处理多个应用过程的实时任务请求; 在注入程序的同时能保证实时遥控指令的执行. 如果这些实时任务请求在时间上发生冲突, 就需要一个调度方案来解决. CCSDS 分包遥控标准中, 有两个信息处理层 (分段层和传输层) 可以支持多任务

调度, 分段层允许设置 64 个接收目标地址指针 (MAP)^[3], 传输层允许设置 64 个虚拟信道 (VC). 如果每个应用过程对应于一个 MAP, 那么分包遥控最多可支持 4096 个应用过程. 如何解决如此之多的实时任务调度是分包遥控的关键, 也是确保卫星在程序注入阶段能够稳定运行的关键. 可以根据各用户信源的模型和实时性要求, 按优先级排队进行发送. 优先级的分配是动态进行的, 在运行中动态统计资源的利用率、数据发送延时等, 要求既要满足各用户的传送实时性要求, 又要使资源最节省、传送最可靠, 依此预测性能, 动态改进调度策略.

4 CAN 总线节点软件加载

星务主机在分发遥控指令的同时, 把注入程序发送到指定的 CAN 节点. 为了区分节点接收到的数据种类, 需要为数据帧重新加上标识. 采用 CAN 总线 2.0B 协议的扩展帧格式* 在实现目的寻址的同时, 对节点内部的数据种类加以区分.

接收节点采用 89C52 单片机 (核与 8051 相兼容) 控制, 通过 Philips 公司的 CAN 总线控制器 SJA1000 实现总线通信^[4]. SJA1000 从总线接收到数据后, 以中断方式通知 89C52 予以处理. 中断程序将完成数据类型的判别和程序数据的接收. 由于中断程序的入口地址是固定的, 这样就保证了程序注入的可靠性和可控性.

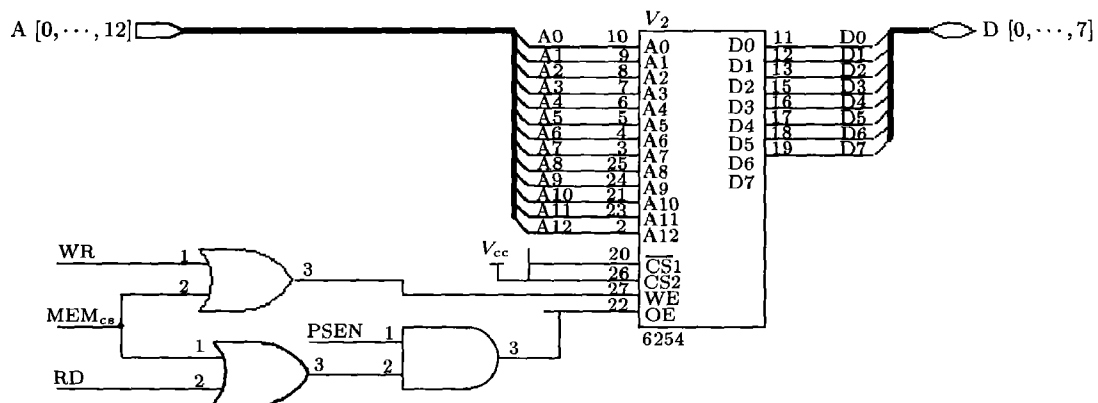


图 4 部分数据空间与程序空间的合并

Fig.4 Integration of the data memory and program memory

89C52 内部有 8KB 程序存储器, 将 EA 管脚接 V_{cc} (+5V), 则程序计数器 PC 的值在 0—1FFFH 时, CPU 取指令访问内部的程序存储器, PC 值大于 1FFFH 时, 访问外部的程序存储器. 此时, 引脚 PSEN 输出外部程序存储器的读选通信号^[5].

根据这一特点, 在设计中用一块 RAM (或 E²PROM) 作为接收下来的程序数据的专用存储器, 程序接收完毕后就在此空间运行. 因而, 需要设计逻辑控制使其成为 89C52 的部分外部数据存储空间和部分外部程序存储空间的合并体. 具体做法是把 89C52 的 WR (写

*Robert Bosch GmbH. CAN Specification. Version 2.0. Stuttgart: BOSCH Company, 1991. 46—48

外部数据存储器) 信号接 RAM(或 E²PROM) 的 WE(写选通输入信号), 把 RD(读外部数据存储器) 信号和 PSEN(读外部程序存储器) 信号逻辑“与”后接 RAM(或 E²PROM) 的 OE(读选通输入信号). 为了避免和其他地址空间冲突, 用译码后的 RAM(或 E²PROM) 片选信号 MEMcs 进行控制. 下面是此逻辑连接的原理图.

在传送程序数据的过程中, 我们规定星务主机先发送待注入的程序长度, 然后发送程序数据. 发送过程分为若干次 8 B 数据传送和余数字节的传送. 接收节点的工作主要在中断程序中, 主程序与此相关的内容是为了防止程序加载执行后跑飞, 对合并存储器初始化为零, 相当于填入 NOP 语句, 并在存储器底部放入长跳转指令, 以便可以跳回加载程序的首部开始执行. 以下是接受方中断程序的流程图.

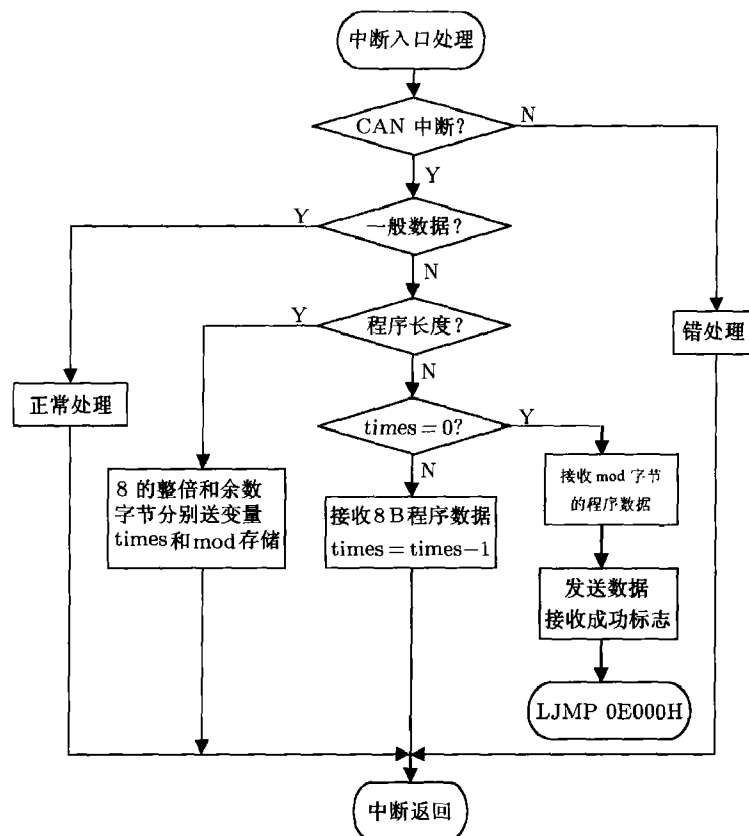


图 5 接收节点中断流程图

Fig.5 Interrupt service routine for data receiving process in remote node

在中断程序中用绝对地址的长跳转来达到改写 PC 值, 实现程序加载的目的, 但此时中断程序还没有返回. 为了能够响应其他中断请求, 在加载程序的开始对堆栈指针初始化, 将需执行的指令地址压栈, 后面的 RETI 指令将使中断返回并从需要处顺序执行. 堆栈的初始化是为了防止程序跑飞时堆栈出现混乱的情况, 为保险起见, 在加载程序的开始还有其他的一些初始化过程.

为了协调发收双方的信息联络,使星务主机确切知道软件是否加载成功,接收节点必须回馈信息.在加载程序中包括一段代码,是向总线发送特定的加载成功标志.直至发送方接收到此信息才认为加载成功.否则,星务主机会重复整个发送过程.3次后仍不成功,认为远程节点无法加载程序.不论成功与否,星务主机都要向地面站汇报,由地面站做出进一步举措.

5 结束语

小卫星一体化设计的要求,使软件重注成为今后小卫星领域必然的发展方向.本文针对软件重注的应用要求提出了具体实施方案,此方案从测控体制到星上的通信平台都考虑了今后的发展主流,具有一定的先进性.本文还提出了此方案中着重需要解决的几个问题,针对其中的一些技术细节作了讨论,并通过实物形式予以验证.

本文提出的软件重注方案应用于实际,将改善小卫星在轨运行的应变能力,提高小卫星的功能密度.在星上部件出现故障时,能够以此实现系统重构,保证系统的良好运行,增强抗突发事件的能力.此外,对于一些试验性的执行功能,也可通过软件重注的方法在卫星运行晚期达到实验目的.总之,软件重注的课题研究有着非常好的应用前景,同时由于它刚刚起步,缺乏有效的地面检验措施和上天经验,因此急需做进一步的工作.

参 考 文 献

- [1] Packet Telemetry. CCSDS 102.0-B-5, Issue 5, Blue Book. Washington, DC: Consultative Committee for Space Data Systems, 2000. 9—11
- [2] Telecommand Part1: Channel Service. CCSDS 201.0-B-3, Issue 3, Blue Book. Washington, DC: Consultative Committee for Space Data Systems, 2000. 13—14
- [3] Huang Wei, Zhang Jisheng. Introduction of the CCSDS AOS suggestion. *J. Aeroc. Meas. Cont.*, 2000, 19(3):37—46. in Chinese. (黄薇, 张纪生. CCSDS AOS 建议介绍. 飞行器测控学报, 2000, 19(3):37—46)
- [4] Wu Kuanming. CAN Field Bus Protocol and System Design. Beijing: BUAA (Beijing University of Aeronautics & Astronautics) Publishing Company, 1996. 33—60. in Chinese (邹宽明. CAN 总线原理和应用系统设计. 北京:北京航空航天大学出版社, 1996. 35—60)
- [5] Zhang Youde. The Mechanism. Application and Experiment of Microcontroller, Shanghai: Fudan University Publishing Company, 1992. 59—60. in Chinese (张友德. 单片微型机原理、应用与实验. 上海:复旦大学出版社, 1992. 59—60)

RESEARCH ON THE PROGRAM RELOADING IN THE MICRO-SATELLITE DESIGN

ZHOU Jian DUAN Dengping WANG Jianyu

(Shanghai Institute of Technical Physics, The Chinese Academy of Sciences, Shanghai 200083)

Abstract

The difference of the development of modern micro-satellite from traditional large satellites is the introduction of the integration conception. The integrated design conception of the micro-satellite make the program reloading on-board form the trend in the development of small satellite. Based on the conception of modularization and standardization, a method for program reloading in the micro-satellite is proposed and discussed in detail to meet the requirement of the micro-satellite design principle in this paper. The discussion is based on the CCSDS telemetry and telecommand scheme and distributed system on board connected with CAN bus. Some extension of the CAN protocol is involved to distinguish the normal data and program data when the node is receiving bytes from the bus. And some additional processes ensure the program data's security and precision. This method can be developed for system bootloading or just for error handling and system re-construction on-board. It will have wide application in the future.

Key words Program reloading, Distributed system, Packet telemetry, Packet telecommand, CAN field bus