

## 适用于行星巡视探测的集中式载荷控制方法\*

王连国<sup>1,2</sup>, 朱岩<sup>1</sup>, 马苗<sup>1</sup>, 饶家宁<sup>1</sup>, 梁耀明<sup>1</sup>, 王蔚<sup>1</sup>

(1. 中国科学院国家空间科学中心复杂航天系统电子信息技术重点实验室, 北京 100190;

2. 中国科学院大学, 北京 100049)

**摘要:**以行星巡视探测任务为背景, 针对资源紧张条件下自主控制载荷进行科学探测的需求, 提出一种集中式载荷控制方法。建立新型的载荷电子学和载荷数管集成一体化硬件架构; 软件采用基于事件表和工作模式表的控制方式自主控制载荷工作; 采取多项健康管理措施, 依据预设的规则监控载荷的状态, 自主对故障载荷采取恢复和隔离措施。实现的公用控制部分仅重3 kg, 功耗6 W, 软件支持9个工作模式表同时运行, 指令执行时刻误差优于40 ms。该方法应用到我国首次火星探测任务中, 地面测试结果表明其简捷、高效、可靠, 适合在资源苛刻的行星巡视探测任务中, 对各载荷进行控制。

**关键词:**行星巡视探测; 集成一体化; 自主控制; 健康管理

**中图分类号:** V446 **文献标志码:** A **文章编号:** 1001-2486(2019)02-008-09

## Centralized payload control method for planetary rover exploration

WANG Lianguo<sup>1,2</sup>, ZHU Yan<sup>1</sup>, MA Miao<sup>1</sup>, RAO Jianing<sup>1</sup>, LIANG Yaoming<sup>1</sup>, WANG Wei<sup>1</sup>

(1. Key Laboratory of Electronics and Information Technology for Space Systems, National Space Science Center,

Chinese Academy of Sciences, Beijing 100190, China;

2. University of Chinese Academy of Sciences, Beijing 100049, China)

**Abstract:** To satisfy the requirement for autonomous scientific detection in resource constrained planetary rover exploration mission, a centralized payload control method was proposed. A novel highly integrated hardware architecture was constructed by integrating payload electronics units and traditional payload OBDH (onboard data handling) unit. The software adopted the control method based on the event table and work mode table to control the payload autonomously. Several health management measures were taken to make fault recovery and isolation according to pre-defined rules. The common control part weighs only 3 kg, and consumes 6 W power. The software supports 9 work mode tables running at the same time. Instruction executing time accuracy is better than 40 ms. The method has been applied to China first Mars exploration. Ground test results show that the method is simple, efficient and reliable, and suitable for payload control in planetary rover exploration missions with harsh resource constraints.

**Keywords:** planetary rover exploration; integration; autonomous control; health management

围绕行星科学领域进行太阳系内的地外天体探测活动, 尤其是通过巡视器(rover)开展行星表面巡视探测, 具有重大科学意义, 国际上的主要航天组织规划了一系列行星巡视探测计划<sup>[1]</sup>。

行星巡视探测器与地球轨道探测器有许多不同之处。首先是资源约束极其苛刻, 重量、体积、能源、数据量都有严格要求。其次是探测器与地面的通信机会有限且时间间隔较长, 必须在长时间没有监控的情况下运行和进行科学探测。因此, 除了需要自主导航, 还应对探测器的工作状态进行自主监测, 能够在一定范围内自主决定要进行的任务, 并能对出现的一些故障进行在轨自主

定位、诊断和修复或重构<sup>[2]</sup>。

对于有效载荷, 一是要实现轻小型化, 低功耗, 提高有效数据比重, 降低数据冗余; 二是要具备一定的自主任务执行能力, 减少对地面控制的依赖, 高效地完成科学探测任务; 三是要具有健康管理能力, 对出现异常的设备, 采取修复或隔离措施。

美国国家航空航天局(National Aeronautics and Space Administration, NASA)和欧洲航天局(European Space Agency, ESA)的深空探测研究起步早, 技术先进, 探测器可提供较多能源, 具有高性能的计算能力<sup>[3-5]</sup>。在自主探测方面, 进行了深入的研究和应用, 具有自主任务规划、智能目

\* 收稿日期: 2018-01-27

基金项目: 国家中长期科技发展规划重大专项资助项目

作者简介: 王连国(1979—), 男, 山东高密人, 研究员, 硕士, E-mail: wanglg@nssc.ac.cn

标选取、科学现象在轨发现能力<sup>[6-8]</sup>。

国内方面,受技术基础和工程约束限制,探测器可提供的资源有限,故在“嫦娥三号”探测器中,采用一种与 NASA 及 ESA 的有效载荷控制方式不同的方案,首次实现载荷数管与多载荷电子学紧耦合轻小型一体化设计。“嫦娥五号”探测器的有效载荷也是采用这种电子学一体化设计。仅使用一台载荷数管设备(载荷电控箱)完成有效载荷的管理,最大限度地实现了系统资源共享和信息综合处理。存在的问题是:各电子学单元直接挂接到 CPU 的数据总线、地址总线、控制线上,这种方式在载荷电控箱集成前,各载荷单独调试和试验时,需要公用部分 CPU 的配合,需要的地检资源多。同时由于所有有效载荷电子学共用 CPU 总线,也会产生相互影响和时序竞争。另外,在载荷运行控制方面,主要依靠地面上注立即指令和延时指令,自主能力较弱。

本文针对行星巡视探测需求和存在的问题,采取了新型有效载荷集中式控制方法。一是采用新型的载荷电子学和载荷数管集成一体化架构,将集成在机箱内的各电子学单元之间接口串行化,目的是在各电子学板功能集成的前提下,降低接口耦合度。二是采用立即指令、事件表和工作模式表等控制方式,可自主控制各载荷工作。三是采取基于预设规则的监控措施和故障影响最小化的隔离方法,能够依据预设的规则监控载荷的工作状态,自主对故障载荷采取恢复和隔离措施而不影响整个有效载荷系统自主探测过程。

本文的集中式控制方法在我国首次火星探测任务中进行了应用,用于对火星车有效载荷进行控制。本方法有以下优点:①实现了有效载荷功能和资源复用,既提高了系统功能密度,解决了系统重量、体积和功耗严重不足的问题,又优化了各电子学间的接口;②具备较强的自主控制载荷进行科学探测的能力,能够适应深空探测通信延迟大、无实时测控的特点。

### 1 具体实现

#### 1.1 集成一体化架构

通过对行星巡视探测任务进行需求分析,距离遥远的探测任务,并不需要高性能的处理能力,但对重量、功耗、数据量等约束苛刻。故本文提出了中等处理性能、低功耗、低成本的载荷电子学和载荷数管一体化集成化方案,系统架构如图 1 所示。公用部分,即载荷数管,是高度集成的计算机与数据处理单元、电源单元。计算机与数据处理单元的原理框图见图 2, CPU 采用 AT697F,运行频率 64 MHz,现场可编程门阵列(Field Programmable Gate Array, FPGA)集成了 1553B 控制器、NAND FLASH 存储控制、科学数据处理、命令/状态处理、OC 指令发送、模拟量采集等功能。各载荷电子学完成各载荷探测仪器的控制,公用部分集中完成整个有效载荷系统的数据管理,包括转发各载荷命令,组织各载荷遥测参数,采集各载荷的科学数据,对载荷数据进行压缩、组包、存储等。

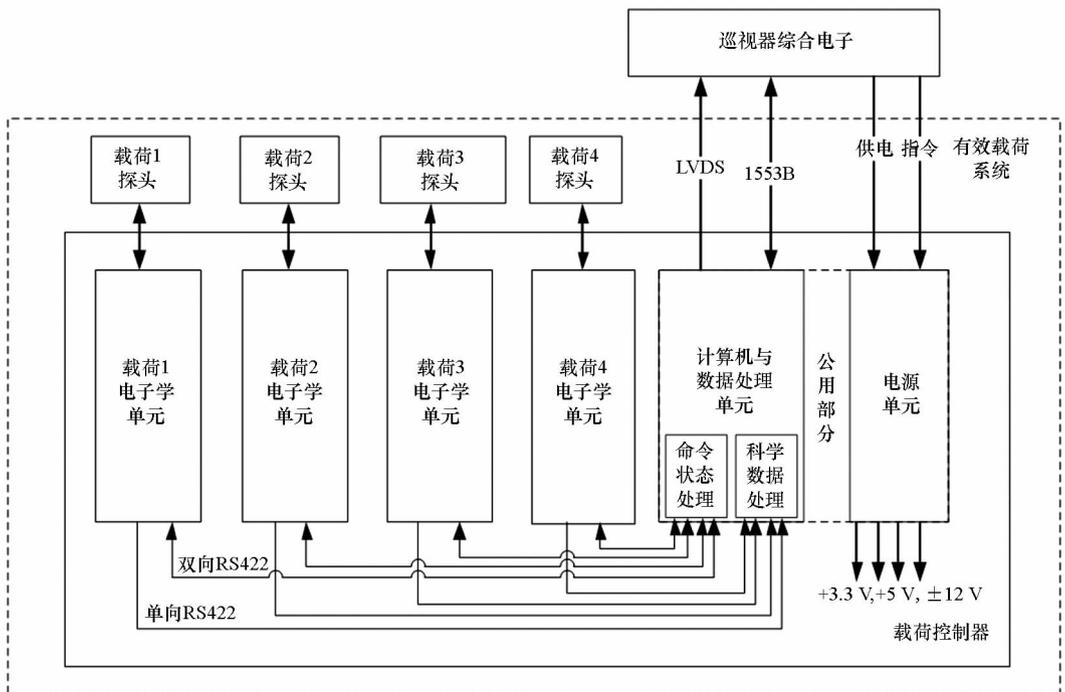


图 1 新型集成一体化载荷数管体系结构

Fig. 1 Novel integrated architecture of payload OBDH

公用部分与载荷电子学之间的控制接口使用双向 RS422 方式,数据接口使用单向 RS422 方式,都是点对点连接,采用标准通用异步收发传输器 (Universal Asynchronous Receiver/Transmitter, UART) 协议,控制接口比特率 115 200 bit/s,数据接口比特率 1 Mbit/s。通过分析目前和未来一些深空探测任务的需求,控制接口比特率 115 200 bit/s 可满足需求,受对地通信限制,载荷产生的数据量不能太大,故载荷的数据率不高,1 Mbit/s 可满足载荷数据传输需求。

该体系结构有以下创新点:

1) 内部控制接口和数据接口都使用点对点串口。该接口方式简单,占用资源少,成本低,方便各电子学单元独立调试和测试,容易进行故障检测、隔离。

2) 设计了适合深空探测应用的控制接口。

该控制接口用于向载荷电子学发送串行数据指令,获取载荷的工作状态参数。采用 RS422 主从式半双工接口方式,为减少接口信号数量,使用同一对差分线,分时进行发送和接收。采用主从式通信方式,公用部分为主机,各载荷电子学为从机,采用主动应答式的握手协议。为保证通信可靠性,制定了完备的通信协议,包括响应时间、超时时间、字节间距、重传机制、错误状态等。

针对深空探测对地通信能力弱和延迟大的特点,优化设计了数据指令,缩短指令长度,统一各载荷的指令格式,对指令应答码和应答遥测数据格式也进行了统一,如图 3 所示。地面注入时,载荷单条指令的有效部分仅为 4 B,即载荷编号、命令类型、命令参数高低字节。其优点是:便于在轨自主按工作模式表执行指令,便于进行在轨健康管理;大大减少数据注入指令长度。

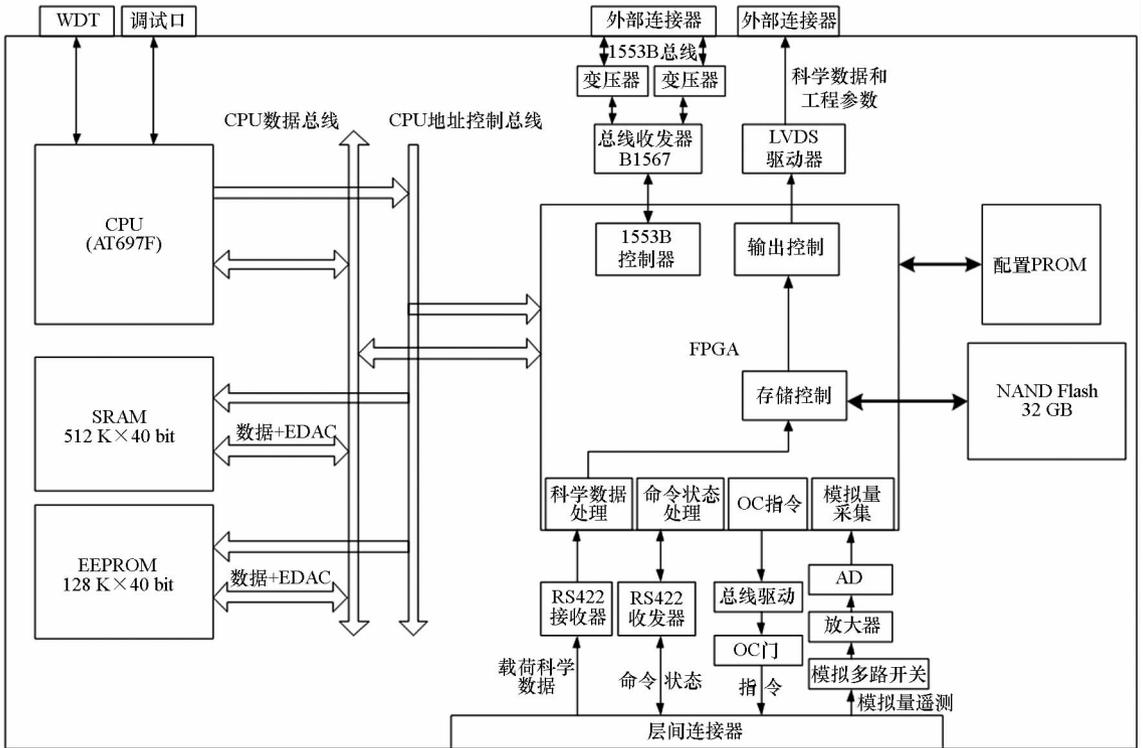


图 2 计算机与数据处理单元原理框图

Fig. 2 Functional block diagram of computer and data process unit

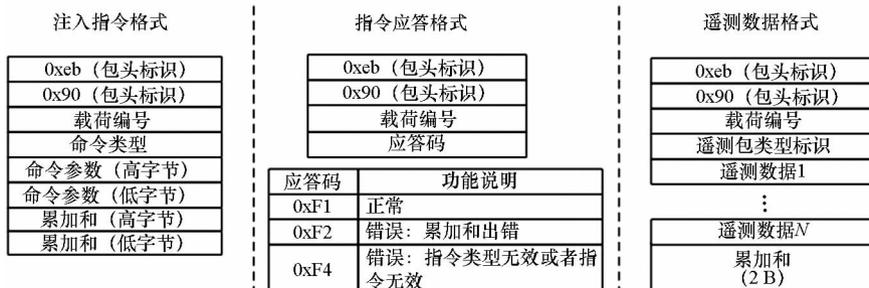


图 3 控制接口数据格式

Fig. 3 Data format of control interface

### 1.2 载荷控制方式

#### 1.2.1 软件控制模型

本文借鉴了ESA的Rosetta探测器软件<sup>[9]</sup>和NASA的火星探测漫游者(Mars Exploration Rover, MER)软件<sup>[10]</sup>,在面向对象模型分析、分层次自主运行、正常运行任务与异常处理配合等方面的设计思想。软件基于实时嵌入式操作系统VxWorks设计,NASA的MER和火星科学实验室(Mars Science Laboratory, MSL)任务也都使用VxWorks操作系统。

软件的主要任务是进行总线通信管理、载荷遥测遥控管理、载荷运行管理、载荷健康管理、数据采集及存储管理等。设计的软件控制模型见图4,由五个模块组成:数据注入模块、决策模块、执行模块、在线监控模块、报告模块。

数据注入模块负责解析地面应用系统上行的数据注入。数据注入分为三种,见图5。第一种是没有时间标签的立即指令,软件收到后立即执

行;第二种是由带有绝对时间标签的指令组成的事件表,软件根据各指令的时间标签执行;第三种是由指令和等待时间组成的工作模式表,软件执行指令,并等待相应的等待时间后再执行下一条指令。工作模式表由立即指令或事件表中的指令启动。

决策模块负责查询事件表和立即指令表,判断是否有事件需要执行,对于需要执行的事件交由执行模块来处理,并删除异常事件和已执行事件。对于立即指令,则立即执行。

执行模块负责具体的事件执行任务,对于低级指令直接执行,对于高级指令则依据工作模式表拆分为多个低级指令来执行。

在线监控模块负责监视各载荷的运行情况,当出现异常情况时采取一定的措施对载荷进行控制。

报告模块负责汇总整理有效载荷系统的运行情况。

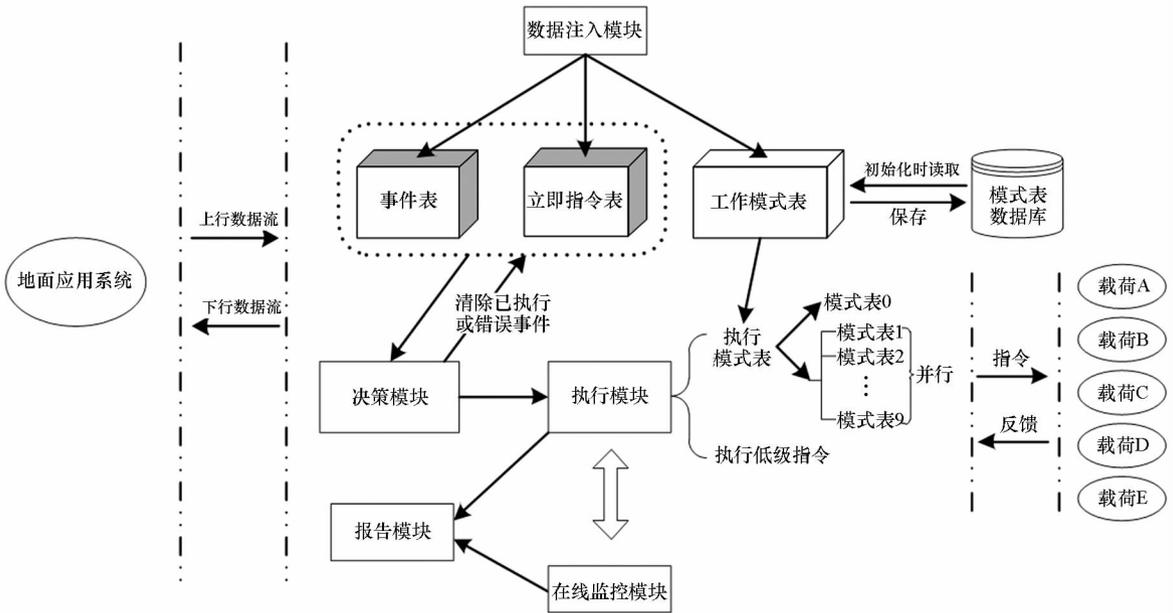


图4 软件控制模型

Fig. 4 Software control model



图5 数据指令类型

Fig. 5 Data instruction type

#### 1.2.2 基于事件表的控制方式

事件表由带有绝对时间标签的指令组成,每一条指令称为一个事件。事件表用于使用绝对时间控制指令执行的情况。软件的决策模块对事件表中的所有事件判断是否执行的依据为时间标签。

定义  $T_s$  为系统当前时间,  $T_e$  为事件指定的执行时间码:若  $T_e > T_s + 2s$ ,表明时间未到达,不对该事件做任何操作,继续保留在事件表中;若

$T_s - 2s \leq T_e \leq T_s + 2s$ , 表明到达事件的执行时间, 交由执行模块处理, 并将其从事务表中删除; 若  $T_e < T_s - 2s$ , 表明事件已超时, 将其从事务表中删除。

传统的做法是将事件表设计为一个无序的数组, 即不按事件所指定的时间码来排序, 这样经过多次添加删除事件后, 数组中已使用的元素位置并不连续。

记数组容量为  $N$ , 添加一个事件的时间复杂度为  $O(N)$ , 删除一个事件的时间复杂度为  $O(1)$ 。由于数组中的事件无序, 决策算法需要依次对所有事件进行查询判断, 查询事件表的时间复杂度则为  $O(N)$ 。

本文使用基于优先级队列的堆排序 (heap sort), 将事件表设计为最小堆的数据结构, 最小堆是一棵完全二叉树, 特点为每个节点的值都小于它的子节点的值, 因此根节点的值即为整个堆的最小值。由于嵌入式软件内存资源有限, 不适合动态分配空间, 所以使用数组而非链表来实现最小堆结构。为了保持最小堆的特性, 插入任一节点、删除根节点的时候都要对堆的其他节点进行调整。对于有  $N$  个节点的最小堆来说, 树的高度为  $\lg N$ , 因此插入一个节点、删除根节点的复杂度为  $O(\lg N)$ , 获取最小节点 (即根节点) 的复杂度为  $O(1)$ 。

决策算法只需获取事件表中时间码最小的事件 (即根节点), 因此, 查询事件表的时间复杂度为  $O(1)$ 。对事件表的删除操作都是删除距当前事件表中时间码最近的事件, 即最小堆的根节点, 因此, 删除一个事件的时间复杂度为  $O(1)$ 。

两种数据结构的性能对比如表 1 所示。

表 1 两种数据结构性能比较

Tab. 1 Performance comparison of the two data structure

数据结构	插入 新事件	删除 最近事件	查询 最近事件
无序数组	$O(N)$	$O(1)$	$O(N)$
最小堆	$O(\lg N)$	$O(\lg N)$	$O(1)$

对本文设计的软件, 查询最近事件的频率远远高于插入新事件以及删除最近事件的频率, 因此采用最小堆算法效率更高。

1.2.3 基于工作模式表的控制方式

对于深空探测, 通信能力有限, 如地球与火星的环回通信时间约 8 ~ 42 min, 每天只有几次通信机会, 地面规划人员需规划下一个火星日的工作

计划并上注, 探测器自主按工作计划运行, 而无地面人员干预<sup>[5]</sup>。

本文的工作模式表即工作计划, 工作模式表中指令执行时刻是相对时间的, 如果工作计划不变, 可重复调用之前的工作模式表。

1) 工作模式表设计

软件支持 10 个模式表 (模式表 0, 模式表 1, ..., 模式表 9), 每个模式表的容量为 128 个指令。将模式表设置为两种优先级, 其中模式表 0 为高优先级, 模式表 1 ~ 9 为低优先级, 同一优先级的模式表可同时执行。

每个模式表中可以包含多个载荷的指令, 也可以只包含单个载荷的指令。对于等待延时指令, 其动作内容为 3 B 的等待时间, 以 100 ms 为单位。

2) 模式表控制过程

对模式表的控制作为一个单独的任务, 100 ms 调度一次。图 6 是模式表调度任务的控制流程, 图 7 是单个模式表的处理流程。

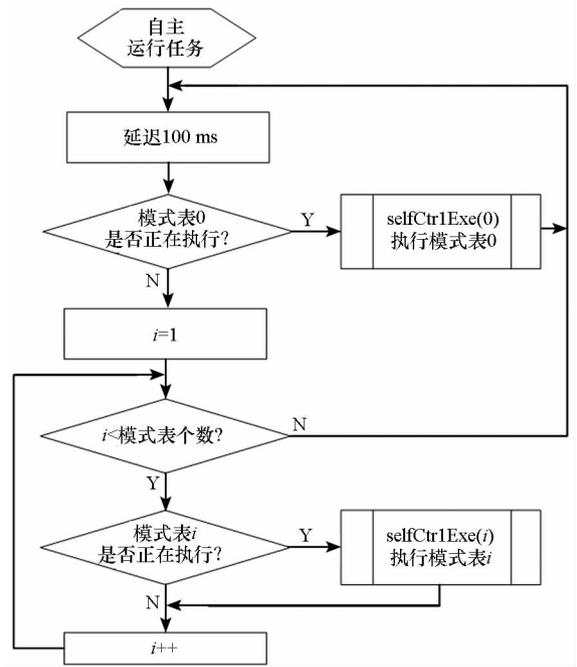


图 6 自主运行任务控制流程

Fig. 6 Flowchart of autonomous control task

3) 执行时间分析

模式表任务每次调度最多执行模式表中的一条动作, 在不考虑模式表 1 ~ 9 运行过程中被模式表 0 打断的情况下:

①若两条指令之间没有等待动作, 则它们之间的时间间隔为 100 ms;

②若两条指令间有 1 条等待动作, 则它们的时间间隔为该等待动作所指定的时间;

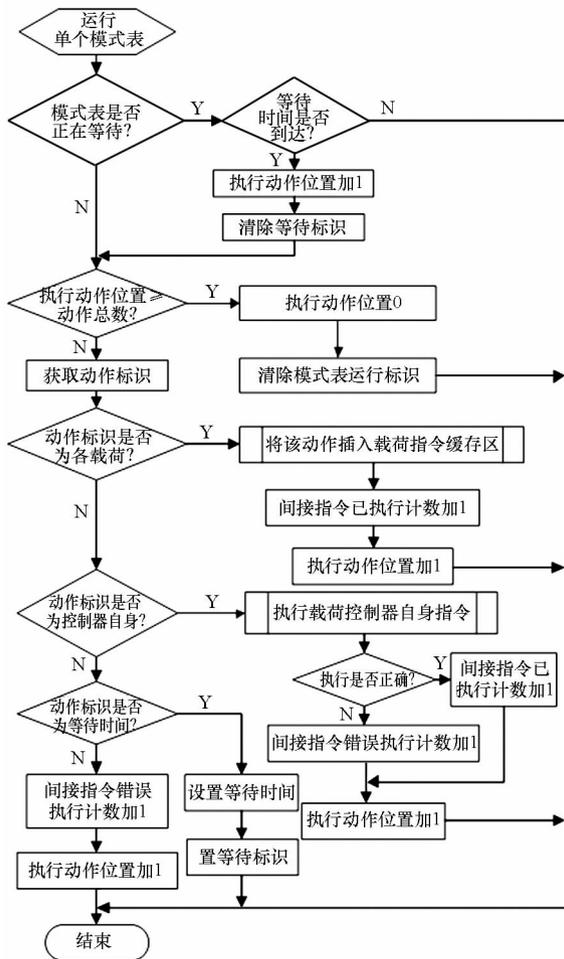


图7 工作模式表运行流程

Fig. 7 Flowchart of work mode table

③若两条指令间有连续  $N$  条等待动作,则它们之间的时间间隔为  $T_1 + T_2 + \dots + T_N - 0.1 \times (N - 1)$  s,  $T_i$  为等待动作指定的时间;

④模式表存储。模式表存储在 NAND FLASH 中的载荷工作模式表存储区,每个模式表单独占用 FLASH 中的 1 块数据空间(约为 2 MB),每个模式表存储三份。FLASH 存储器采用多种保护措施,包括使用(12,8)汉明码进行存储器使用信息保护,使用RS(256,252)编译码对存储数据进行保护<sup>[11]</sup>。

系统上电时,从 FLASH 中读取模式表,放入 SRAM 模式表运行区。模式表在 SRAM 中也存储三份,采取三取二冗余设计。

系统复位时,首先校验 SRAM 模式表运行区是否有效,若无效,则从 FLASH 中读取该模式表;若有效,则不必再从 FLASH 中读取。

可通过数据注入指令修改在 SRAM 运行的模式表区内容。可以每个模式表单独注入,也可以整个模式表一起修改,还可以修改模式表中的单个动作。需要将更新的模式表存储时,则存储到 FLASH 中。

### 1.3 健康管理

健康管理的原则为:①监测系统设备状态,而不是事件状态,恢复的是系统设备状态。②监测点离故障点越近越好,即在软件的最底层监测并执行动作。这也是 NASA 的火星科学实验室(Mars Science Laboratory, MSL)探测器软件采取的原则<sup>[12]</sup>。另外,对于所有监控,启动一次异常处理之后,自动禁止监控。各项监控可通过注入指令使能、禁止。监控阈值可通过注入指令修改。采取的健康管理策略见表2。

表2 健康管理策略

Tab. 2 Health management strategy

监测点	规则	采取动作
各载荷开机温度	执行载荷加电指令时,判断载荷温度是否达到允许开机温度	满足条件则开机,否则不开机
各载荷工作温度	载荷工作时,判断载荷温度是否满足工作温度范围	连续2次异常,则关机
工作电流	根据载荷工作模式判断整机工作电流	连续3次超出阈值,则给处于加电状态的载荷断电
载荷1工作模式设置	判断工作模式与注入指令是否一致	如果不一致,则再次注入“工作模式”指令,并判断返回的工作状态。若连续3次指令均无法改变当前工作模式,则给该载荷断电,间隔3s后再次上电,并恢复其工作状态
载荷1工作状态	载荷处于工作状态时,判断其是否开始采集数据	连续3次轮询,数据计数都不变,则先注入“待机”指令,等待3s后再次注入“工作”指令
载荷2工作电源	判断3.3V供电状态	若连续3次异常,则给该载荷断电
载荷2目标指向	发送指向命令后,判断指向是否成功	若成功,则发送下一条指令,否则继续遥测,最长等待2min,若2min后仍然不成功,则给该载荷断电

表 2(续)

监测点	规则	采取动作
载荷 2 调焦	发送调焦命令后,判断调焦是否成功	若成功,则发送下一条指令,否则继续遥测,最长等待 4 min,若 4 min 后仍然不成功,则给该载荷断电
载荷 3 工作状态	判断载荷工作是否正常	若连续 3 次异常,则给该载荷断电,间隔 5 ~ 10 s 再开机。连续 2 次重加电后仍出现异常状态,则第 2 次重加电后不再进行监控,保持加电状态,继续工作
载荷 4 温控	对有特殊温控的部位,判断其温控是否满足要求	若满足,则发送启动工作指令,否则等待下次遥测轮询时再次判断;若仍不满足,则继续判断,最长判断 5 min,之后发送启动工作指令

## 2 实现结果及分析

本文提出的集成一体化架构,将载荷电子学和公用控制部分集成到一起,用高集成度、高性能的公用控制部分(载荷数管)实现载荷集中控制。各电路板为 6U 尺寸,公用部分的电源单元在一块电路板上实现主备份冗余,计算机与数据处理单元有 2 块板,组成主备份冗余。通过采取轻小型化措施,公用部分仅重 3 kg,功耗 6 W。

控制软件基于 VxWorks 操作系统,软件任务设计见表 3。

表 3 软件任务设计

Tab. 3 Software task design

序号	任务名称	优先级	分配堆栈	实际堆栈
1	1553B 总线通信	150	2048	368
2	RS422 通信	155	2048	384
3	自主运行	160	2048	368
4	数据压缩	170	2048	528
5	主任务	175	2048	632
6	FLASH 管理	180	2048	280

其中,1553B 总线通信任务、RS422 通信任务、自主运行任务和主任务是周期性调度任务。使用操作系统的 `getUserTime()` 函数(获取当前系统时间码,时间码分辨率为 1 ms)记录各个任务每次调度的时间,得到各个任务的实际调度周期。图 8 给出了周期性任务的设计调度周期和实际调度周期最小、最大值。从图中可以看出,优先级高的 1553B 总线通信任务、RS422 通信任务、自主运行任务,实际调度周期和设计调度周期相差很小。图中的任务实际调度周期是软件多次运行的统计结果,1553B 总线通信任务调度 23 670 次,RS422 通信任务调度 47 349 次,自主运行任务调度 9469 次,主任务调度 47 314 次。

各任务运行时间的测试方法是使用 FPGA 内

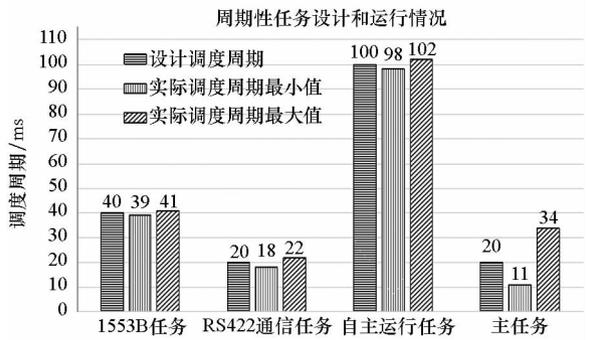


图 8 周期性任务设计和运行情况

Fig. 8 Periodic task design and running result

的时间码标签寄存器,分辨率为 20  $\mu$ s。各任务执行时间如图 9 所示,除数据压缩任务外,其他有实时性要求的任务运行时间很短。FLASH 管理任务与读、写、擦除动作相关,运行时间不一,此处未列出。

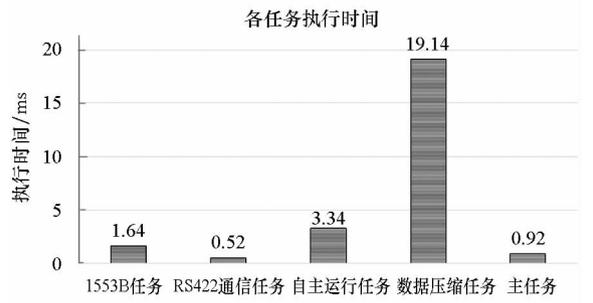


图 9 任务运行时间

Fig. 9 Running time of task

使用 `getUserTime()` 函数记录事件表和工作模式表中各指令的执行时刻,得出各指令执行时刻的抖动情况。图 10 为事件表中指令执行时刻抖动情况,该结果为由 64 条指令组成的事件表在不同负荷下运行结果,最大偏差小于 40 ms,优于误差 2 s 的指标要求。图 11 为工作模式表中指令执行时刻抖动情况,该结果为由 9 个模式表(模式表 1 ~ 9)并行运行时的结果,最大偏差为 3 ms,远优于误差 2 s 的指标要求。

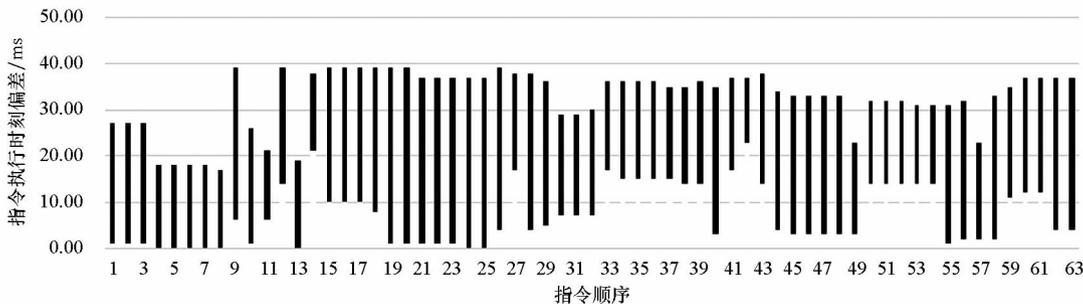


图 10 事件表指令执行时刻抖动情况

Fig. 10 Jitter of event table instruction executing time

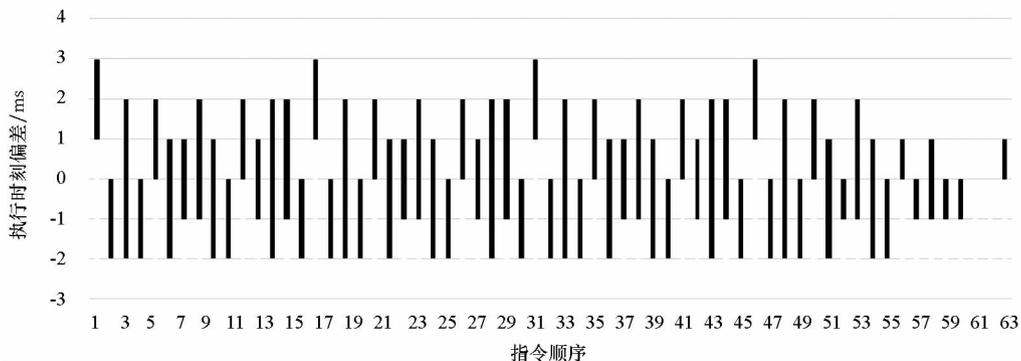


图 11 工作模式表指令执行时刻抖动情况

Fig. 11 Jitter of work mode table instruction executing time

### 3 应用情况

我国首次火星探测任务已进入研制阶段,火星车巡视探测科学任务着眼于火星局部地区,开展高精度科学探测。本文的集中式载荷控制方法在火星车有效载荷系统中进行了应用。将各载荷专用的电子学和公用的载荷数管集成到载荷控制器机箱中,由公用部分统一提供二次电源,统一进行各载荷的运行控制和数据获取,统一提供与探测器平台接口。

地面应用系统科学家将规划好的下一火星日工作计划(即载荷工作模式表)上传至火星探测器平台,平台数管将载荷工作模式表转发给载荷数管。当日探测时,载荷数管根据载荷工作模式表自主控制各载荷进行科学探测,并存储科学数据。在探测工作结束后,将科学数据发送至平台数管存储,平台数管在通信弧段将数据下行地面。

该集中式载荷控制方法经过了初样阶段有效载荷系统联试和探测器整器测试,整个有效载荷系统运行良好,验证了该方法的有效性。该方法具有简捷、高效、可靠、功耗低、成本低的优点,适合行星巡视探测类任务对载荷进行管理。

### 4 结论

本文根据行星巡视探测任务对载荷轻量化和自主探测能力的需求,提出一种集中式载荷控制方法。硬件平台采用新型的载荷电子学与载荷数管集成一体化设计,在结构、功能集成的基础上,尽可能降低接口的耦合度,以方便各单机调试和试验。软件采用基于事件表和工作模式表的控制方式自主控制载荷工作,可在一个工作模式表中规划多个载荷的探测动作,也可在不同的模式表中分别规划载荷的探测动作,多个模式表可同时运行,具有较高的灵活性。同时采取了一系列健康管理措施,能够在长时间无人干预的情况下,保证载荷设备安全,并隔离故障载荷或恢复故障载荷的工作流程。该方法经过了实际工程验证,可满足实际工程任务需要,具有很好的工程借鉴意义。

### 参考文献(References)

- [1] Sanguino M T J. 50 years of rovers for planetary exploration: a retrospective review for future directions [J]. *Robotics and Autonomous Systems*, 2017, 94: 172 - 185.
- [2] 孙泽洲, 孟林智. 中国深空探测现状及持续发展趋势 [J]. *南京航空航天大学学报*, 2015, 47 (6): 785 - 791.

- SUN Zezhou, MENG Linzhi. Current situation and sustainable development trend of deep space exploration in China[J]. Journal of Nanjing University of Aeronautics and Astronautics, 2015, 47(6): 785–791. (in Chinese)
- [3] Vladimirova T, Fayyaz M, Sweeting M N, et al. A novel autonomous low-cost on-board data handling architecture for a pin-point planetary lander [J]. Acta Astronautica, 2011, 68(7/8): 811–829.
- [4] Ellery A. Planetary rovers: robotic exploration of the solar system [M]. Germany: Springer, 2015.
- [5] Bajracharya M, Helmick D, Maimone W, et al. Autonomy for Mars rovers: past, present, and future [J]. Computer, 2008, 41(12): 44–50.
- [6] Castano A, Fukunaga A, Biesiadecki J, et al. Automatic detection of dust devils and clouds on Mars [J]. Machine Vision and Applications, 2007, 19(5/6): 467–482.
- [7] Estlin T, Bornstein B J, Gaines D M, et al. AEGIS automated science targeting for the MER opportunity rover[J]. ACM Transactions on Intelligent Systems and Technology, 2012, 3(3): 50.
- [8] Francis R, Estlin T, Doran G, et al. AEGIS autonomous targeting for ChemCam on Mars science laboratory: deployment and results of initial science team use [J]. Science Robotics, 2017, 2(7): eaan4582.
- [9] Baksa A, Balázs A, Pálós Z, et al. Software on a comet: the philae lander's central onboard computer [J]. IEEE Software, 2016, 33(2): 81–84.
- [10] Reeves G E, Snyder J F. An overview of the Mars exploration rovers' flight software[C]//Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2005.
- [11] 宋琪, 邹业楠, 李姗, 等. 卫星固态存储器数据容错设计与机制 [J]. 国防科技大学学报, 2016, 38(1): 101–106.
- SONG Qi, ZOU Yenan, LI Shan, et al. Design and mechanism of the fault tolerance of solid state recorder[J]. Journal of National University of Defense Technology, 2016, 38(1): 101–106. (in Chinese)
- [12] NTRS. The Mars science laboratory flight software—a platform for science and mobility[R]. California Institute of Technology, 2012.