

基于国产平台 VPX 架构的 IPMC 控制器设计

张睿, 黄泳铭, 吴悠, 陈相宇

(华北计算技术研究所, 北京 100083)

摘要: 针对 VPX 架构体系中由于 IPMC 未能完全按照规范要求设计而引发的一系列兼容性问题, 提出了一种基于国产 MCU 控制芯片的通用 IPMC 控制器的设计与实现; 该 IPMC 控制器应用于国产平台通信单元机箱内功能板卡端, 设计满足 VITA46.11 标准和 IPMI 协议, 作为管理系统的核心组件智能平台管理控制器, IPMC 设计实现功能板卡与机箱控制器 ChMC 之间的标准 IPMI 管理以及各种故障信息的诊断与处理; 该 IPMC 控制器管理接口设计灵活, 可拓展性强, 便于后续的移植与维护, 经联调测试验证了该设计的灵活性与实用性。

关键词: 智能平台管理接口; 现场可替换单元; VPX 标准; IPMB 总线

Design of the VPX Architecture IPMC Controller Based on Domestic Platform

Zhang Rui, Huang Yongming, Wu You, Chen Xiangyu

(North China Institute of Computing Technology, Beijing 100083, China)

Abstract: A design and realizer of universal IPMC controller base on homemade MCU controller chips is raised for resolving a series compatibility problem of nonstandard design of IPMC in VPX system. The IPMC controller applies on the function board of homemade communication unit, design of it is the VITA46.11 standard and IPMI protocol. As the core groupware of the intelligent platform, IPMC design and implementation of the standard management between the function modules and the chassis controller and the diagnosis of various fault information. The IPMC controller design of the management interface is flexible, expansibility and convenient for subsequent transplantation and maintenance. The flexibility and practicability of the design are verified by combined test.

Keywords: IPMI; FRU; VPX standard; IPMB bus

0 引言

军用环境的复杂多变对军事化设备的维护和管理提出很高的要求, 如何保证军用设备运行的稳定性并进一步降低日常的管理维护成本已经成为各个相关领域研究的热点之一。广泛应用于对高可靠性有极大需求的服务器制造业、电信业、工业控制以及军事等领域的 IPMI 管理技术, 为提高系统的整体智能化管理、自适用管理和高可靠性提供一种开放的、低成本的解决方案。

本设计适用于军用国产平台功能板卡上, 满足 VPX VITA46.11 标准和 IPMI2.0 协议, 设计目的是实现系统管理员能够在单一的控制台上实现对不同软、硬件平台的功能板卡模块进行监视和管理, 同时可以对各个功能板卡(主要包括计算功能模块、制冷模块、供电模块、交换模块)电源、电流、CPU、制冷单元等各个部件的工作状态进行有效的实时监控。

作为 IPMI 管理的重要组成部分, 合理的 IPMC 控制器设计, 特别是控制器管理软件如何设计出满足协议标准又能实现接口灵活, 现场维护能及时有效地避免各种事件给系统造成宕机带来的损失, 从而提高系统的可维护性和稳定性是实现上述系统管理的关键^[1]。

1 机箱管理系统架构

VPX 机箱管理系统分为三个子系统: 系统管理子系统、机箱管理子系统和板卡管理子系统^[2]。其中, 系统管理子系统运行于机箱外, 通过 SNMP 协议接口对机箱进行系统级别管理; 机箱管理子系统运行于机箱内的主控板卡上, 对上通过 SNMP 协议处理系统管理软件对机箱的配置请求, ChMC(板级管理控制器)通过 IPMB 总线收发 IPMI 消息实现对机箱内 FRU 模块的管理配置; 板卡管理子系统运行于各个功能板以及电源模块、制冷模块的 IPMC 控制器上, 其控制器 IPMC 通过 IPMB 总线收发 IPMI 消息, 处理对本地板卡的配置管理请求以及将各类事件进行上报^[3]。

VPX 机箱管理系统整体架构^[4]如图 1 所示。

2 IPMC 控制器设计与实现

2.1 板卡管理子系统功能

作为机箱管理系统的重要组成部分, 板卡管理子系统承担机箱内多个不同类型功能模块的 IPMI 管理控制任务^[5]。为机箱管理提供访问本地 FRU 的管理接口。板卡管理子系统主要实现下列管理功能:

- 1) 实现功能板卡的插入检测、通断电、自检、冷热启动等相关控制;
- 2) 响应主控板卡通过 IPMB 总线发送的命令及错误处理;
- 3) 实现温度、电压传感器的实时监测;
- 4) 实现机箱管理访问控制 FRU 和 SDR 信息;
- 5) 向机箱管理软件上报各类信息, 包括负载电源自检、

收稿日期: 2018-03-17; 修回日期: 2018-04-18。

作者简介: 张睿(1983-), 山东人, 工程师, 主要从事嵌入式系统研发方向的研究。

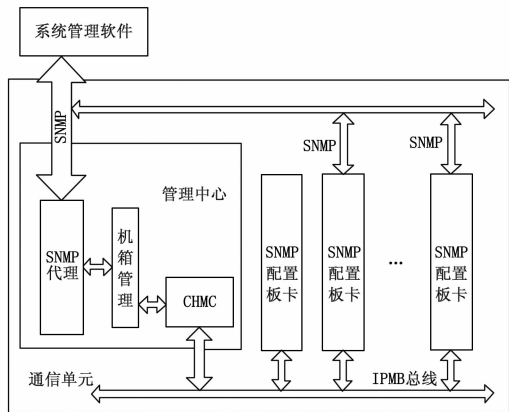


图 1 VPX 机箱管理系统架构图

现场可替换单元状态切换、各类传感器告警事件的实时上报。

机箱管理子系统通过 IPMB 总线向功能板卡发送 IPMI 管理命令，IPMC 控制器内的命令解析模块实现收发消息包的解析组包功能。命令处理模块根据 IPMI 协议标准将解析后的数据进行处理，包括 SDR 和 FRU 相关信息的存储读取；命令处理模块在处理命令时调用底层硬件访问接口实现上层主控板卡需求数据的采集计算。此外，IPMC 控制器内集成了 CONSOLE 命令解释器，通过串口实现命令行管理。板卡 IPMC 控制器管理的各模块组成见图 2。

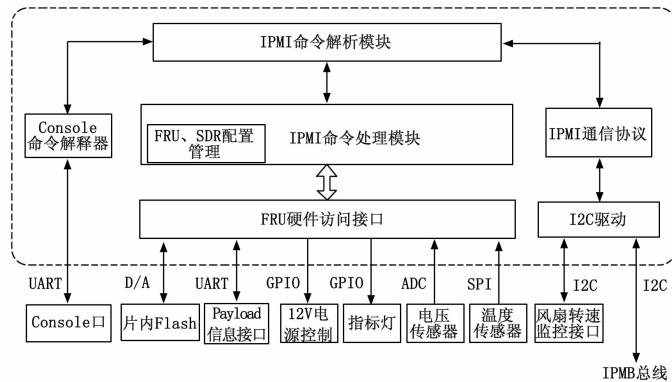


图 2 管理功能模块示意图

板卡管理软件自上而下分为三层，分别是：业务层、中间件和外设层。业务层实现主要板卡管理功能；中间件提供通信、控制和硬件读写接口；外设层为底层硬件驱动。各层详细情况见图 3。



图 3 管理功能层次划分

2.2 板卡状态管理

按 VITA46.11 标准规定^[6]，将 VPX 机箱内板卡的工作状

态分为：M0、M1、M4、M6 和 M7。工作状态定义及描述见表 1。

表 1 板卡工作状态定义

状态	状态描述	PL 电源	IPMC 电源
M0	板卡不存在	未知	未知
M1	IPMC 在线 板卡未使能	禁止	使能
M4	IPMC 在线 板卡使能	使能	使能
M6	IPMC 在线 板卡断电	禁止	未知
M7	IPMC 通信丢失	未知	未知

2.2.1 工作状态转换

按照 VITA46.11 标准，将功能板卡总体管理流程分为四个主要阶段，各个阶段模块的工作状态之间的转换关系见图 4。

- 1) IPMC 上电初始化阶段：工作状态 M0→M1；
- 2) 功能板卡负载电源加载阶段：工作状态 M1→M4；
- 3) 功能板卡负载电源掉电阶段：工作状态：M4→M6→M1；
- 4) 功能板卡信息丢失阶段，对应工作状态 M7。

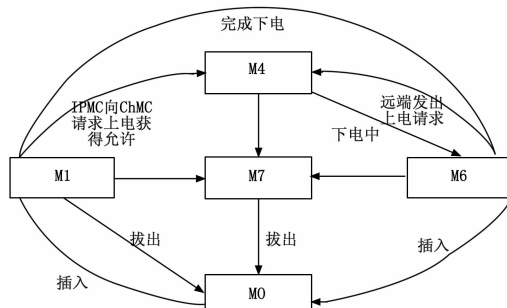


图 4 工作状态转换过程示意图

2.2.2 上电初始化阶段管理流程

板卡管理的关键过程在于控制其上电、断电系列流程，当板卡插入到插槽中，备用电源驱动 IPMC 工作，首先进行硬件地址的自检，此时板卡工作状态为 M1，IPMC 与 ChMC 之间进行一系列的消息握手，包括设备 ID 的获取、设备出厂信息的获取、IPMB 总线冗余设置、事件接收地址设置、VSO 版本获取等信息。上电初始化阶段 IPMC 与主控板卡 ChMC 之间的信息交互见图 5。

2.2.3 上电阶段管理流程

上电初始化阶段命令响应完毕后，功能板卡进入加载负载电阶段。板卡在这一阶段需要完成负载电源自检并将自检结果以事件信息的方式上报给主控板卡；此外，IPMC 将工作状态转换事件上报给主控板卡，收到主控板卡的相应信息后，加载负载电，功能板卡进入到正常运行状态 M4；在 M4 阶段，IPMC 周期性对本板各个传感器进行轮询，同时读取 EEPROM 中存入的 SDR 信息进行解析，将轮询到的传感器实时数值与对应的 SDR 阈值进行比较判断，如果数值超限，IPMC 以告警事件信息方式上报给主控板卡。上电阶段的信息交互见图 6。

2.2.4 负载电卸载阶段管理流程

正常运行于 M4 状态的功能板卡接收到主控板卡卸载负载

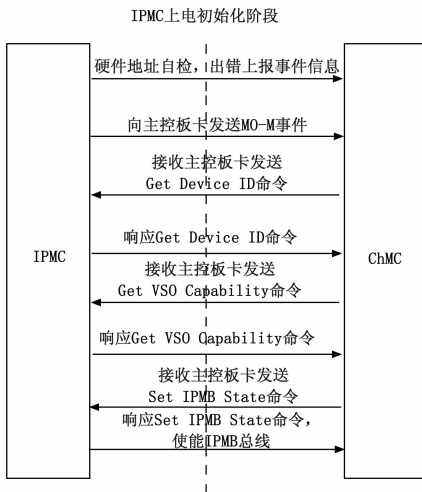


图 5 上电初始化阶段信息交互

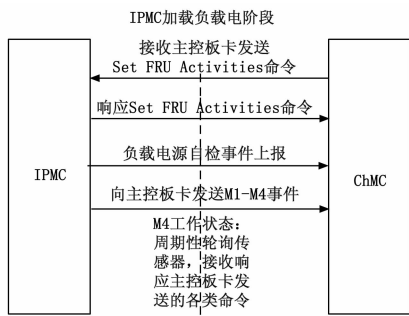


图 6 加载/卸载电阶段信息交互

电源命令, 关闭负载电源并响应回复主控板卡; 负载电关闭后, IPMC 将板卡工作状态转换信息 (M4-M6) 以事件上报给主控板卡; IPMC 在 M6 状态接收主控板卡周期性发送的轮询设备 ID 命令并响应上报板卡是否在线信息。负载电卸载过程见图 7。

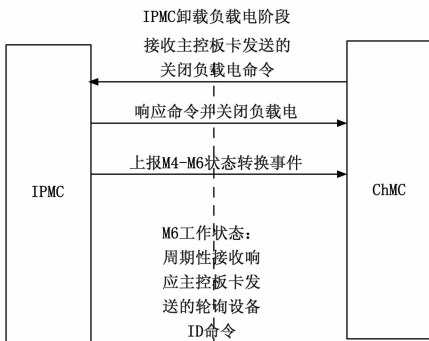


图 7 卸载负载电阶段信息交互

2.3 平台事件处理

2.3.1 平台事件信息分类

事件信息是指在管理软件系统探测到重要或者危急系统管理事件时由管理控制器发送的一组专用消息。事件具体到计算机系统中包括超越温度阈值、超过电压/电流阈值、电源报错等。VPX 机箱的事件信息包括内部事件和外部事件, 外部事件主要由 BIOS 事件、SMS 系统管理软件事件、IPMB 事件和

总线事件组成^[7-8]。不同的事件信息通过不同接口上报给主控板卡事件接收器。机箱管理系统的事件分类及产生来源请求路径示意图见图 8。

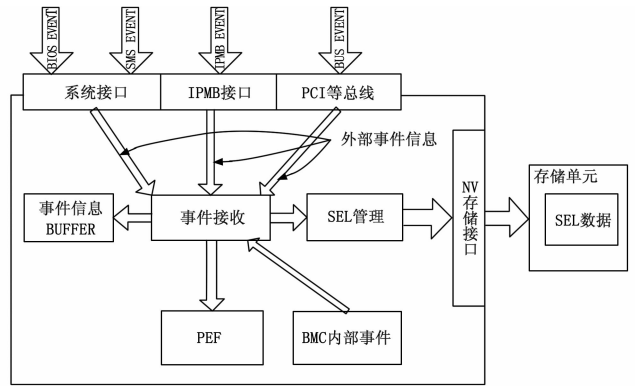


图 8 事件产生请求示意图

板上上报事件属于 IPMB 事件, 即 IPMC 通过 IPMB 总线向主控板卡上报的一系列消息。根据板卡的具体业务和控制流程, 此类消息共分为三类: 负载电源自检结果事件、板卡工作状态转换事件、各类板卡传感器告警事件。

2.3.2 IPMB 事件消息格式

IPMC 在进行事件请求前需要先响应主控板卡的事件接收地址设置命令, 按照 IPMI2.0 的协议标准^[9], 事件接收方的地址即 IPMB 总线的从地址, 标准设置为 20 h。事件信息格式遵循 IPMI 协议中 IPMB 事件格式, 事件信息封装 IPMI 和 IPMB 两层协议包头, 其中 IPMI 协议包头与上报数据部分共同进行校验。事件信息由 IPMC 主动上报给主控板卡的 ChMC, 事件信息使用 02h 作为统一的命令码, 详细的信息格式如图 9 所示。



图 9 事件消息协议格式

2.3.3 事件处理策略

VPX 机箱系统中会存在各种复杂的情况, 如果当各种事件同时发生的时候, 事件处理核心如何去设定一个执行策略是事件告警处理设计中的核心问题。策略设计的是否高效率, 处理结果是否合理会直接影响到系统整体的工作效率。针对系统内的三类事件, 事件上报采取不同的策略。

传感器告警事件上报分两种情况, 当板卡模块运行在 M4 状态时, IPMC 会周期性地对其所在模块的各个传感器状态进行轮询, 并将轮询到的数值与其阈值进行比较, 如果产生告警事件, 将事件上报给主控板卡; 当主控板卡下发获取传感器实时数值命令时, IPMC 将采集到的传感器数值阈值比对后, 将数值和状态信息以响应命令的形式进行回复; 同时判断状态位的值, 若不是正常值, 则调用事件信息发送, 将请求告警信息发给事件接收方。

工作状态类型转换是第二类需要请求上报的事件, 此处的工作状态指上述 M0-M7 的状态转换。此类事件主要产生于

系统控制板上、下电过程中。当功能板卡完成每一个工作状态切换时，需要将事件转换情况和原因进行上报。

第三类需要及时上报的事件是负载电源的自检结果事件。为提高板卡及系统的稳定性，硬件平台为 IPMC 提供了一系列负载电源状态自检的方法，能够确保负载电源运行结果完全受控。IPMC 对负载电源进行周期性检测，并将自检结果以事件信息形式进行上报。

事件处理设置重传机制，针对不同类型事件的紧急程度设置重传次数和重传周期。建立重传队列，将需要重传的信息放入重传队列。针对传感器数据需要倒摄比对的情况，事件告警系统为传感器告警信息预留单条记录。由于信息刷新频率较高，为避免主控板卡过于密集地受到相同类型的告警事件。传感器事件告警设置滞回值，以百分比的形式存入板卡的 SDR 信息中。并设置正向参数和负向参数，告警策略设计将本次读取数值与上次数值进行比对，并结合参数和滞回值进行判断是否触发事件上报^[10]。

阈值滞回策略见图 10，当读取数据超上线时，如果参数是正向的，则进行超上线持续告警；如果参数是负向的，则结合滞回值进行判断，当数据低于滞回值则取消上报。当读取数值超下限时，当参数为负向时，则触发负向持续告警；若参数为负向时，则结合滞回值来判断，高于滞回值范围，不再进行事件告警。

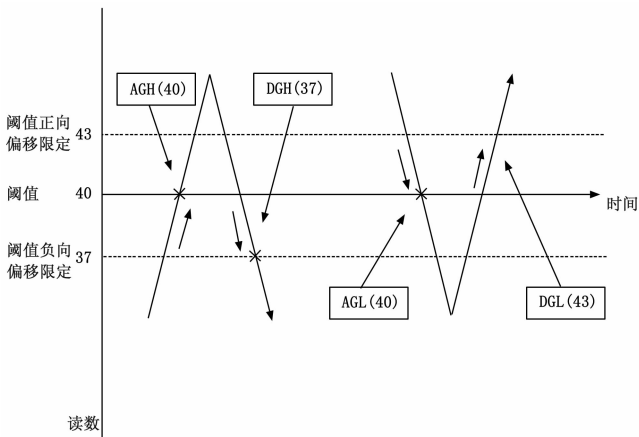


图 10 阈值滞回策略示意

2.4 参数设计及阈值判断策略

2.4.1 传感器参数设计

IPMC 管理内容中的核心部分是各类板级传感器数据的实时采集和超限告警事件上报。根据 IPMI 标准协议的规定，IPMC 需要将底层各个接口读取的原始数据 (Raw value) 上报给主控板卡，主控板卡读取 SDR 信息中的参数表结合原始寄存器数值进行计算得到最终值。这样做的原因是由于标准的 IPMI 协议中命令或者事件上报的值均为一个字节长度，无法满足浮点型数据上报的实际需求，系统通过参数设计解决上述问题以适应实际数据的精度要求。

经过对 IPMI 标准及 VITA46.11 规范的解析，得出读取值与实际值得转换数学公式：

$$y = L[(Mx + (B * 10^{k1})) * 10^{k2}] \text{ uints}$$

其中：x 为原始数据值，L [] 为线性类型函数，M 和 B 为有符号整型系数，k1 和 k2 为有符号整型指数。这四个参数

对于不同类型传感器设定不同，设计中进行一一计算比对，得到相对准确的参数填入 SDR 数据中，SDR 数据根据不同的 sensor_number 进行分类逐条存储，当主控板卡读取到原始数据后结合上述参数机型计算取值。

2.4.2 阈值判断策略

IPMI 标准中事件触发类型，事件可分为事件产生、事件消除、正常状态三类。根据所监控的传感器特点，将其引发事件主动上报的阈值设置为上限下限各三条，分别为：UNR、UC、UNC 和 LNC、LC、LNR。即不可恢复超上限、一般超上限、轻微超上限，与其对应的是不可恢复超下限、一般超下限、轻微超下限。并在正常数值两侧加入 2.3 节中的滞回值设计。

阈值判断策略具体流程是：首先建立全局变量表，将各个 SDR 信息填入表中，表中的 val 值作为上次数值，条用底层接口函数读取的数值作为此次值，根据此次数值所在的区间（6 条阈值界定界限划分）并结合与上次的数值比对，确定传感器状态和变化方向。当状态与上次不同并且没有回到滞回值范围内，则触发事件发生标志位；如果此次数值回到滞回值区内，则触发事件消除标志位；此次与上次状态一致时不予上报事件。

此阈值判断策略经过实际应用验证，将数值设置在不同区间和状态下，能够覆盖到全部可能值的判断，精确地上报超限事件。

2.5 FRU 与 SDR 设计

FRU 与 SDR 数据存储和解析设计作为 IPMI 管理的重要组成部分。VPX 架构整机中全部的 FRU 信息，概括起来可分为两大类：机箱 FRU 信息和板卡 FRU 信息。具体到板卡模块中，板卡 FRU 信息被固定存储在各个板卡的 EEPROM 中，按照板卡模块分类，FRU 信息分为三类：功能板卡 FRU 信息、CU（制冷单元）FRU 信息和 PM（电源模块）FRU 信息。

各个模块的 FRU 通用信息包括：FRU 模块产品信息、地址信息、设备 ID、热交换信息、最大电流值等信息。上述 FRU 信息被各自存储在各自模块的 EEPROM 中。

2.5.1 FRU 数据设计

FRU 数据区域划分为三部分：Chassis Area、Board Area、Product Area。Chassis Area 区域存储机箱相关的信息数据，包括机箱类型、机箱生产相关信息等；Board Area 区域存储板卡模块的信息数据，包括设备 ID 号码、板卡生产日期、板卡序列号等；Product Area 区域的重要信息是厂家初始化信息。上述信息需要在响应主控板卡获取设备号的命令时读取并解析，将解析后的数据发送给 ChMC 处理。

为了方便解析 FRU 数据，设计 32 字节 FRU 数据包头，将包头分为区域标记和偏移量标记部分。前一个标记用于指示具体存储数据的区域，偏移量指示相关数据存储的具体地址。管理软件设计了较为灵活的接口函数用于初始化、生成、存储、读取和解析 FRU 信息，便于后期进行设备信息的添删改操作。

2.5.2 SDR 数据设计

SDR 信息主要存储设备传感器相关监控信息，包括模块上各个关键指标的监控，包括各个关键参数的阈值设置。SDR 存储数据包括条目号、关键电压、电流、温度、风扇转速、负载电源自检结果传感器、工作状态转换传感器等相关值，传感

器 ID、传感器类型、传感器数量、传感器接收地址、实体 ID、传感器阈值设定、滞回值设定、正负参数设定等。

按照 IPMI 协议规定, 一条 SDR 信息容量为固定 64 字节。每一条传感器条目对应存储一个传感器信息, 由于同样类型的传感器根据不用的位置和作用, 阈值设定不尽相同, 设计时在预留字符串加注标识用以区分同一类型不同实体的传感器条目。

3 设计实现与应用

本文设计的 IPMC 控制器已经成功应用于某自主可控国产服务器机箱内的十块不同类型包括电源模块、制冷模块、交换模块、存储模块和计算模块的功能板卡上。应用实践表明, 该 IPMC 控制器的设计能够满足服务器内各类板卡的集成数据通信管理, 管理接口设计灵活, 能够方便地在各个板卡进行移植。图 11 是主控板卡用一款通用的 IPMI 管理图形界面监控的界面截图, 图中是计算板卡上 3.3 V 电压的监控信息, 左侧视窗列表中显示机箱内各个板卡信息, 展开信息为该计算模块上所有传感器的信息; 右侧视窗里显示传感器编号为 0x15 的传感器监控信息, 六种不同颜色的线是传感器上下六条阈值界限, 黄色的线是实际检测到的电压。

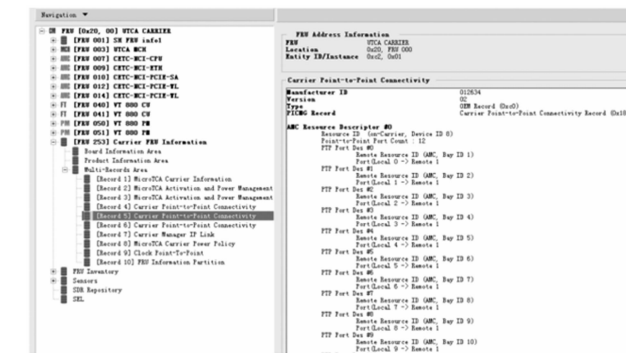


图 11 板卡管理界面演示

利用本文提出的板卡 IPMC 控制器的设计和实现方法, 可实现 VPX 架构服务器平台的标准 IPMI 管理, 利用通用的 IPMI 监控图形界面可实现运行高效、界面美观的服务器机箱管

(上接第 98 页)

可扩展性和安全性, 最为主要的是能够解决高并发的通信问题。

本系统适应未来智慧城市的发展方向, 对其他类似的运用物联网技术的应用设计有着很大的参考价值。

参考文献:

[1] 廖建尚, 等. 物联网开发与应用 [M]. 第 1 版, 北京: 电子工业出版社, 2017.

[2] 刘少强, 张靖, 等. 现代传感器技术: 面向物联网应用 [M]. 第 2 版, 北京: 电子工业出版社, 2016.

[3] 刘海燕, 杨帮华, 丁丽娜, 等. 基于 STM32 的红外火灾探测系统设计 [J]. 计算机测量与控制, 2013: 1114 - 1116.

[4] 侯琛, 赵千川, 冯浩然, 等. 一种物联网智能数据采集系统的研究与实现 [J]. 电子测量技术, 2014 (5): 11 - 12.

[5] 易飞, 余刚, 何凌. GPRS 网络信令实例详解 [M]. 第 1 版. 北京: 人民邮电出版社, 2013.

[6] 刘波文, 孙岩. 嵌入式实时操作系统 μ COS-2 经典实例: 基于 STM32 处理器 (第 2 版) [M]. 北京: 北京航空航天大学出版

理系统。

4 结束语

本文设计并实现了一个基于 VPX 架构和 IPMI 管理协议的 IPMC 控制器。本文在分析 IPMI 智能管理系统结构、板卡管理子系统功能、板卡工作状态管理、事件信息上报策略设计、设备信息与传感器信息存储和解析等方面对 IPMC 控制器进行设计, 并验证了该方法的可行性, 为其他系统 IPMI 智能管理的开发提供非常好的技术借鉴。

本设计与之前设计的提升在于各类传感器数值处理策略设计。IPMC 板级管理软件对于读取的传感器数据设置合理的阈值判断标准, 对于不同精度类型的数据, 管理软件建立不同的列表用相应的容错值和滞回值配合上下限阈值进行精确地判断, 触发事件上发并设计重传机制, 以完成传感器数据处理。

参考文献:

[1] 梅建超. IPMI 在 ATCA 系统中的应用与架构设计 [J]. 数字技术, 2011: 56 - 58.

[2] 朱红育. IPMI 在 VPX 系统中的应用与设计 [J]. 火控雷达技术, 2013: 66 - 69.

[3] 孙博文. 基于 ARM9 的通用 IPMC 载板设计 [J]. 计算机测量与控制, 2014 (6): 1926 - 1928, 1931.

[4] 洪艳. 基于 VPX 高速综合信息处理平台设计 [J]. 导弹与航天运载技术, 2011: 58 - 61.

[5] 裴晓骞. 智能管理平台接口研究及实现 [J]. 计算机技术与发展, 2006.

[6] 46.11, VITA 46.11 System Management on VPX [S]. 全本, 2013. 11. 18.

[7] 黄泳铭. 基于 VPX 总线的车载计算平台设计 [J]. 测控技术, 2011 (4): 94 - 96, 103.

[8] 郑东卫. VPX 总线的技术规范及应用 [J]. 火控雷达技术, 2009: 73 - 77.

[9] V2.0. IPMI Intelligent Platform Management Interface Specification Second Generation [S]. 全本. 2013. 9. 1.

[10] 王波. 一种数学形态学的哈特曼-夏克传感器阈值选取方法 [J]. 光电子技术, 2008: 227 - 231.

[7] 左忠凯, 刘军, 张洋. FreeRTOS 源码详解与应用开发: 基于 STM32 [M]. 第 1 版. 北京: 北京航空航天大学出版社, 2017.

[8] 勒加雷. 嵌入式协议栈 μ C \ TCP-IP: 基于 STM32 微控制器 [M]. 第 1 版, 邝坚, 等译. 北京: 北京航空航天大学出版社, 2013.

[9] Carrell J L, Chappell L A. TCP \ IP 协议原理与应用 (第 4 版) [M]. 第 1 版, 金名, 译. 北京: 清华大学出版社, 2014.

[10] Stevens W R. TCP/IP 详解卷 1: 协议 [M]. 第 1 版, 北京: 机械工业出版社, 2000.

[11] 李林锋, Netty 权威指南 [M]. 第 2 版, 北京: 电子工业出版社, 2015.

[12] 魏莹. 基于 Netty 框架的智能终端与服务器通信的研究 [D]. 西安: 西安电子科技大学, 2015.

[13] Maurer N. Netty in Action [M]. Manning Publications, 2014.

[14] Matam S, Jain J. Pro Apache JMeter: Web application performance testing [M]. Apress; 1st ed, 2017.

[15] Erinle B. Performance testing with JMeter 2.9 [M]. Packt Publishing, 2013.