

# 计算机运行全过程日志记录系统设计

徐立颖, 许松伟, 冯笑, 刘正尧, 周亚光

(中国电子科技集团公司 第15研究所, 北京 100083)

**摘要:** 为了解决当前国产特种计算机(如星载计算机, 机载计算机等)出现偶发故障无日志记录, 无法准确获取软硬件和外部环境在故障出现时的具体状态, 造成后续难以进行故障分析, 无法做到及时归零处理的难题; 深入分析了计算机运行的全过程以及影响计算机运行过程的内外因素, 创新性地提出一种计算机运行全过程日志记录技术; 该技术采用嵌入式硬件平台进行设计, 运行嵌入式软件, 采集计算机运行过程状态和外部环境信息并进行存储, 解决了国产计算机从通电至操作系统启动完成这一阶段无日志记录的难题, 同时还实时记录计算机运行过程的环境信息; 完成了软硬件设计, 并进行试验验证, 设计故障注入, 验证了所设计的系统能够准确反映计算机运行全过程状态信息, 表明该技术有助于故障定位、机理分析、故障复现、采取措施及验证, 最终完成故障归零处理。

**关键词:** 全过程; 日志记录; 特种计算机; 故障分析; 故障归零

## Design of Log Recording System in the Whole Process of Computer Operation

XU Liying, XU Songwei, FENG Xiao, LIU Zhengyao, ZHOU Yaguang

(No. 15 Research Institute of China Electronics Technology Group Corporation, Beijing 100083, China)

**Abstract:** In order to solve the problem that domestic special computers (such as on-board computer, airborne computer, etc.) have occasional failures without log records, and it cannot accurately obtain the specific status of the software, hardware and external environment when failures occur, which makes it difficult to carry out failure analysis and cannot complete the fault closed loop in time. After the deep analysis of the whole computer operation process and the internal and external factors affecting computer operation, a log recording technology for the whole computer operation process is innovatively proposed. The technology adopts the embedded hardware platform design, runs the embedded software, collects and stores the computer operating process status information and external environment information. The problem of no log records is solved for domestic computer during the period from power-on to start-up of the operating system, the environmental information of the computer running process is recorded in real time, the hardware and software design is completed, and the test verification is carried out. Fault injections are designed to verify that the designed system can accurately reflect the state information of the whole computer operation process, indicating that this technology is helpful for the fault location, mechanism analysis, fault reentry, taking measures and verification, finally the fault closed loop is completed.

**Keywords:** whole process; log records; special computer; fault analysis; fault closed loop

## 0 引言

计算机在运行过程中不可避免会产生故障<sup>[1]</sup>, 包括硬件故障和软件故障。特种计算机应用于需要自主可控的特定领域, 要求更高的可靠性<sup>[2]</sup>, 但其工作环境一般为高温、低温、高湿、震动、复杂电磁等恶劣环境中, 其故障的发生可能与计算机软件或硬件本身相关, 也可能与外部恶劣环境相关。特种计算机运行过程中不可避免会出现故障, 某些故障偶发难以复现甚至无法复现, 为特种计算机安全可靠运行留下了隐患。如何进行故障定位、机理分析, 提出解决措施, 进而重建外界环境, 复现故障, 再证明故障定位准确、机理清晰、解决措施有效并举一反三, 最终实现对故障的归零处理, 对于故障处理尤为重要<sup>[3-4]</sup>。这种场

景下日志记录成为故障分析的重要基础<sup>[5]</sup>。文献[1]、[6]对日志收集、分析软件进行了设计, 文献[7]对操作系统故障管理软件进行设计, 文献[8]对日志系统提出了高性能低时延分析方法, 以上的研究对象为操作系统日志或应用程序日志, 依赖于操作系统的正常运行, 缺少对与底层硬件相关的故障日志记录及分析, 如电源故障、板卡局部过温以甚至是其他环境参数等。针对此问题, 本文提出并设计实现了一种全过程日志记录系统, 实现了操作系统外计算机运行全过程日志记录, 与操作系统及应用日志互补, 实现对计算机运行全过程日志记录。

## 1 计算机运行全过程分析

计算机运行全过程包括主板上电、BIOS启动、操作系

收稿日期: 2023-02-23; 修回日期: 2023-04-07。

基金项目: 国家科技部重点专项(2022YFB2902703)。

作者简介: 徐立颖(1972-), 硕士研究生, 研究员。

引用格式: 徐立颖, 许松伟, 冯笑, 等. 计算机运行全过程日志记录系统设计[J]. 计算机测量与控制, 2024, 32(1): 172-178.

统启动及运行三大过程。

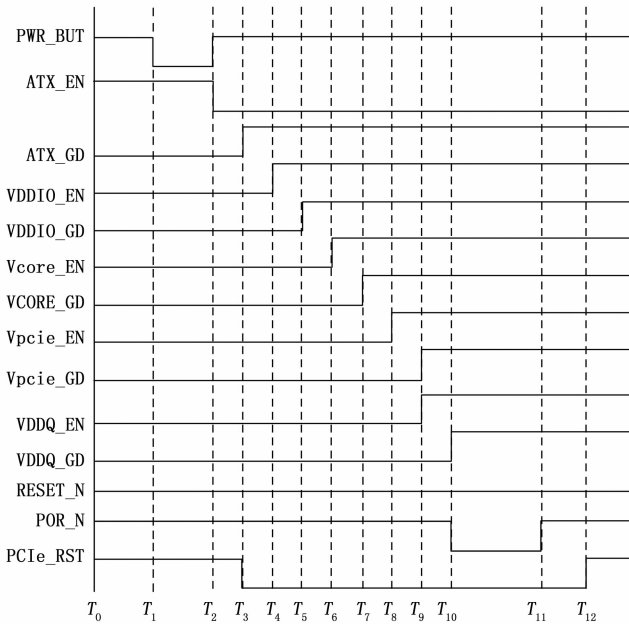


图 1 飞腾某款 CPU 上电时序图

主板上电过程涉及底层硬件执行初始化逻辑, 不同硬件平台启动过程大致相同, 细节略有差异。以国产飞腾某款 CPU 为例, 其上电过程由板载 CPLD (X86 平台采用板载 EC) 严格控制上电的每个环节, 包括供电电源状态检测、电源时序控制、电源电压异常监测控制等。图 1 为飞腾某款 CPU 上电时序图, 在外部供电电源正常的条件下, 从 T<sub>0</sub> 开始, 上电按钮按下后 T<sub>1</sub>, CPLD 检测到按钮脉冲信号 T<sub>2</sub> 后, 开始按照时序进行上电直至 T<sub>12</sub><sup>[9]</sup>, 在此过程中, 涉及多种电压规格的电源, 任意一种电源异常 (电压异常、带载异常) 都将造成上电异常, 最终导致无法进入到下一启动环节。

当最后一个电源上电完成后, CPU 进行上电复位, 复位完成后 CPU 开始按照设计好的进程运行, 并进入到 BIOS 启动环节, 该环节共有 7 个阶段, 其中第 5 阶段是 BIOS 和操作系统过渡的阶段, 第 6 阶段是操作系统启动并进入就绪状态, 具体如图 2 所示<sup>[10-11]</sup>。

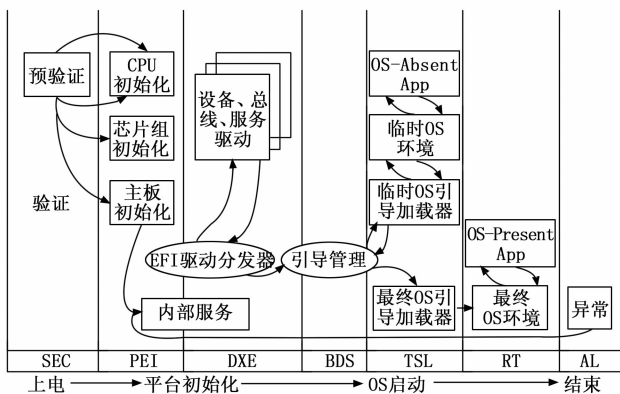


图 2 BIOS 启动流程图

第 1 阶段: SEC (Security Phase), 该阶段主要是完成验证工作, 将内存信息放置在缓存里, 为下一阶段做好准备。

第 2 阶段: PEI (Pre-EFI Initialization Phase), 该阶段主要进行 CPU、内存、芯片组的初始化工作, 完成硬件的初步初始化工作, 形成内存和资源表, 为下一阶段做好准备;

第 3 阶段: DXE (Driver Execution Environment), 该阶段对第 2 阶段的硬件进行进一步初始化, 并完成相关驱动的调用安装, 为引导操作系统提供设备操作环境。

第 4 阶段: BDS (Boot Device Selection), 启动设备选择阶段, 该阶段会扫描所有可启动设备, 并形成启动设备列表, 操作系统引导程序根据用户设置的启动优先级顺序在下一阶段引导操作系统启动。

第 5 阶段: TSL (Transient System Load Phase), 该阶段是 BIOS 和操作系统启动的过渡阶段, 由 BIOS 通过操作系统引导程序引导操作系统启动, 操作系统开始接管 BIOS 的资源和设备, 对相关资源和设备进行管理。

第 6 阶段: RT (Run Time Phase), 该阶段操作系统完成启动, 并进入运行状态。

第 7 阶段: AL (After Life), 该阶段的作用主要用于错误处理和灾难恢复机制, 并不是每一次启动过程必须经历的阶段, 但作为计算机运行的全过程, 该阶段也是必不可少的阶段。

## 2 全过程日志记录元素分析

当前特种计算机未进行全过程日志记录的设计, 从上电至操作系统完成启动、操作系统及应用程序日志记录服务启动前这一阶段的软硬件故障日志记录未引起足够的重视, 未建立起相应的日志记录系统, 日志记录依赖于操作系统正常启动后的系统日志、应用日志等日志记录<sup>[12]</sup>, 造成从主板上电至操作系统完成启动、操作系统及应用程序日志记录服务启动前这一阶段没有相关日志记录。该阶段影响主板正常启动的因素有主板所处的环境因素、电源上电及主板设备初始化。因此, 全过程日志记录主要包括环境参数日志记录、电源上电过程日志记录、主板设备初始化过程日志记录、操作系统及应用日志记录。下面逐一分析全过程日志记录需要记录的元素。

普通商用或家用计算机其运行场所一般比较固定, 外部物理环境也比较优良, 外部环境对于普通商用或家用计算机的正常工作影响可以不做深入考虑。与之相比, 特种计算机由于其外部环境一般为恶劣环境, 外部环境的因素对于计算机的正常工作影响大。例如, 应用于车载平台的某型特种计算机, 其工作环境指标设计指标为 -40℃ ~ +65℃, 93% @ 35℃, 普通商用或家用计算机不会在如此恶劣的环境下工作。一般而言, 特种计算机外部环境包括温度、湿度、震动、霉菌、盐雾、低气压 (海拔高度) 等, 其中温度、湿度直接影响元器件的电气性能, 震动、霉菌、

盐雾、低气压等因素则是通过间接方式影响元器件正常工作,例如震动不会直接影响电气性能,而是由于震动造成接触不良(焊点脱焊,连接器异位等)导致计算机运行异常。此外,温度、湿度影响电气性能后期不易通过外观识别确定因素,而震动、霉菌、盐雾、低气压等因素影响电气性能,后期可以通过外观识别,例如震动造成的焊盘脱焊、连接器异位,霉菌、盐雾造成外观腐蚀,低气压造成对气压敏感的元器件外观形变等<sup>[13]</sup>。综上,针对环境因素的日志记录元素选取温度和湿度两个因素。

计算机主板包含多种电源,CPU对于每种电源上电的先后顺序均有明确的要求,因此电源上电过程日志记录需要记录电源的电压、电源状态以及先后顺序。以国产飞腾某款主板为例,电源规格超过10种,上电时序通过CPLD进行严格控制,只有当当前电源上电完成,并输出POWERGOOD状态信号后方可进行下一个电源使能上电。针对此类主板设计,虽然可以通过CPLD输出电源上电过程的日志记录简单而快捷的获得日志记录。但对于采用专用EC进行上电时序设计的主板,则无法输出电源上电过程日志记录,为了使全过程日志记录系统更具通用性,在硬件电路上设计电源参数采集电路,实现全过程日志记录。

电源上电过程中,CPU会根据上电状态及时初始化相关设备,尤其是与CPU运行相关的参数,例如时钟、主频、缓存、内存、功能接口、外部设备等<sup>[14]</sup>。当前国产计算机设备在初始化过程中都能通过通讯接口输出初始化过程及状态信息,此部分信息是启动过程状态信息的重要组成部分,也是全过程日志记录元素的重要组成部分。

操作系统及应用日志记录。操作系统日志又称为主机日志,由操作系统自动生成,与操作系统行为密切相关。它记录了系统发生的各种事件,此类日志文件一般是纯文本的文件,通常存放在“var/log”目录下<sup>[15]</sup>。应用软件日志由其运行在服务器或终端微机的应用数据处理软件产生,记录软件运行的状态信息和异常信息<sup>[1]</sup>。全过程日志记录对操作系统及应用软件日志记录依赖于操作系统及应用软件本身。

### 3 方案设计

全过程日志记录从记录阶段进行分析,可以分为操作系统及应用程序日志记录服务执行前和后两个阶段,不妨将操作系统及应用程序日志记录服务执行前的日志记录信息称之为系统外日志记录,将操作系统及应用程序日志记录服务执行后的日志信息称之为系统内日志记录。操作系统外日志记录包含环境参数、上电过程及状态、主板硬件初始化过程状态信息。从软硬件角度而言,全过程日志记录包括硬件和软件两大部分。硬件上设计独立于主板功能电路的嵌入式处理模块,用于环境参数采集、电气参数采集、主板硬件初始化状态信息接收、记录存储等。软件则是运行在嵌入式处理模块上的嵌入式应用软件,包括各个嵌入式外设的驱动、数据处理、记录存储等。

#### 3.1 硬件电路设计

为了保证日志记录的完整、全面,嵌入式处理模块设计上电快速启动,并记录环境参数,然后记录上电过程信息并接收来自硬件初始化的信息。按照功能单元对嵌入式处理模块硬件电路进行划分,包括国产微处理器单元、电气参数采集单元、存储单元、环境参数采集单元、UART通讯接口单元、“PS\_ON”控制上电、独立供电单元,具体如图3所示。其中虚线部分与嵌入式处理模块存在通讯,但不属于嵌入式处理模块的组成部分,包括外部计算机和板卡功能电路。

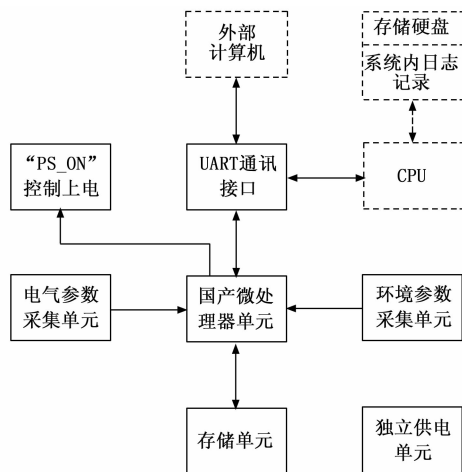


图3 嵌入式硬件处理模块硬件设计方案

##### 3.1.1 国产微处理器单元

国产微处理器单元选用兆易创新微处理器GD32F407系列,该处理器是基于ARM Cortex-M4 RISC内核的32位通用微处理器,片上集成3072KB Flash、192KB SRAM、3个16通道12位ADC控制器、3路I2C接口、2路UART接口、3路SPI接口、82个GPIO接口等众多设备,内部高速总线工作频率高达168 MHz,具有高性能、低功耗等众多特性<sup>[16]</sup>,可以满足计算机运行全过程日志记录系统对硬件资源的需求。

国产微处理器单元是嵌入式硬件处理模块的核心,通过UART通讯接口模块与本地CPU、外部计算机设备进行通讯,实现CPU上电初始化和操作系统启动过程状态信息的获取以及日志记录对外发送;通过I2C接口与环境参数采集单元进行通讯,实现环境参数获取;通过SPI接口与存储单元进行通讯,实现日志记录的存储与读取;通过ADC控制器实现电气参数的采集输入;通过GPIO接口实现对主板的“PS\_ON”上电控制。

##### 3.1.2 电气参数采集单元

电气参数采集单元用于采集板卡电气参数,包括电压采集和电流采集。电压采集直接连接至ADC输入电路实现;电流采集由霍尔传感器实现,通过霍尔传感器将电流信息转换成电压信息,再输入ADC输入电路。需要说明的是,嵌入式微处理器IO口的输入电平是5 V Tolerant,对

输入 ADC 电压超过 5 V 的信号, 需要通过精密电阻分压进行处理后方能输入至 ADC, 在嵌入式软件读取 ADC 采样值时, 需要按照分压比例进行换算。

### 3.1.3 存储单元

存储单元采用 Flash 存储颗粒进行设计, 用于存储操作系统外日志记录。Flash 存储颗粒设计预存不少于最近 20 次上电启动的系统外日志记录。

### 3.1.4 环境参数采集单元

环境参数采集单元用于实现对环境参数的采集, 包括温度和湿度两个元素。采用国产中科银河芯 GXHT35 系列高精度集成温湿度传感器, 该系列传感器在硅基 CMOS 晶圆上集成高灵敏度 MEMS 湿敏元件可以有效减少多芯片信号传输的干扰, 具有包括全温湿度范围校准和温度补偿数字输出、16 位高精度 ADC 转换电路、2.2~5.0 V 宽电源电压适应性等多项优良性能, 温湿度采集精度高达  $\pm 1.8\%$  RH 和  $\pm 0.1\text{ }^{\circ}\text{C}$ 。GXHT35 系列高精度集成温湿度传感器、信号整形、滤波、换算、ADC 转换等功能, 采用 I2C 接口进行通讯, 用于读取温湿度相关数据<sup>[17]</sup>。设计时将温湿度传感器挂载至国产微处理器单元。

### 3.1.5 UART 通讯接口单元

UART 通讯接口单元包括 2 路 UART 通讯串口, 其中 1 路直接连接至主板 CPU 调试串口, 另 1 路通过电平转换电路转换成 RS232 输出, 用作外置通讯串口。

连接至主板 CPU 调试串口的 UART 用于接收主板上电过程中 CPU 初始化、BIOS 运行、操作系统启动过程输出的日志信息, 并及时存储至存储单元中, 实现上电启动过程日志记录。

外置通讯串口用于主板启动异常, 无法正常读取到存储单元内部的日志记录的场景。外部计算机通过外置通讯串口向嵌入式微处理模块发送日志读取指令, 国产微处理器单元将存储单元的日志记录信息通过 UART 通讯接口发送至外部计算机, 实现日志记录的输出。

### 3.1.6 “PS\_ON”控制上电

当主板按下上电开关后, 独立供电单元随即开始工作, 电源稳定输出后嵌入式处理单元开始进入工作状态。嵌入式处理单元通过环境传感器单元采集环境参数, 通过电气参数采集单元采集输入电压, 以上信息做为 CPU 上电前的环境参数, 在上电的第一时间进行采集并记录存储在存储单元。CPU 上电前的环境参数满足主板上电要求时, 嵌入式处理单元输出“PS\_ON”控制上电信号, 控制主板上电。

主板上电后, 电气参数采集单元会实时采集各路电源输出电压, 同时监控电源芯片输出的 POWERGOOD 状态信息, 将电源电压及状态信息实时保存至存储单元中。

### 3.1.7 独立供电单元

特种计算机板级产品一般采用单电源供电, 如 12 V、24 V、28 V 等。为了保证嵌入式处理模块不受主板功能电路影响, 设计有独立供电单元, 专用于嵌入式处理单元的

供电, 同时为了保证嵌入式处理单元在 CPU 开始上电时已经处于就绪状态 (随时可以记录上电过程记录信息), 由嵌入式处理单元通过“PS\_ON”的方式控制主板功能电路的上电。

## 3.2 软件设计

计算机运行全过程日志记录系统软件包括嵌入式端记录软件和外部计算机上位机软件。嵌入式端记录软件以嵌入式处理模块为硬件平台, 实现数据采集、日志记录获取及存储功能, 具体框架如图 4 所示; 上位机软件采用通用的串口调试助手, 实现外部计算机与计算机运行全过程日志记录系统的通讯, 读取存储在存储单元的日志记录信息。

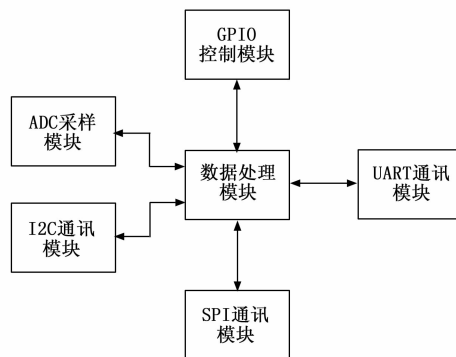


图 4 嵌入式端记录软件框架

嵌入式端记录软件运行在国产微处理器硬件平台上, 直接与硬件系统相关, 属于底层软件, 采用模块化设计思想, 实现模块的高效复用<sup>[18]</sup>, 包括数据处理模块、ADC 采样模块、I2C 通讯模块、SPI 通讯模块、UART 通讯模块、GPIO 控制模块等六大功能模块。其设计流程图如图 5 所示。

设备上电后, 各个功能模块进行初始化; 初始化完毕后采集环境参数并记录环境参数, 然后控制模块输出上电使能信号, 主板开始上电。在上电过程中, ADC 采样模块依次对各个电压进行采样并存储。若采样各个数据均为正常, 表明主板具备启动状态, 则进入串口数据监视状态, 随时接收串口数据。串口数据若为 CPU 发送过来的初始化及操作系统启动过程日志, 则将串口数据进行保存。若串口数据为输出日志指令, 则通过 SPI 通讯模块读取 Flash 的日志记录输出。UART 通讯完毕后, 表明主板已进入稳定状态, 则每隔 10 分钟进行一次环境参数和电气参数的采集、存储。若环境参数或电气参数出现异常, 则每隔 1 s 进行采样, 并记录异常次数, 连续 5 次异常后, 则结束软件流程。

### 3.2.1 数据处理模块

数据处理模块是嵌入式端记录软件的核心模块, 实现各个功能模块的组织、调用, 是各个模块数据流的中心<sup>[19]</sup>。数据处理模块包通过接收来自 ADC 采样模块的电气参数、环境参数数据, 并进行相应的数据处理; 通过 UART 串口接收来自 CPU 的初始化日志记录、操作系统启动过程日志

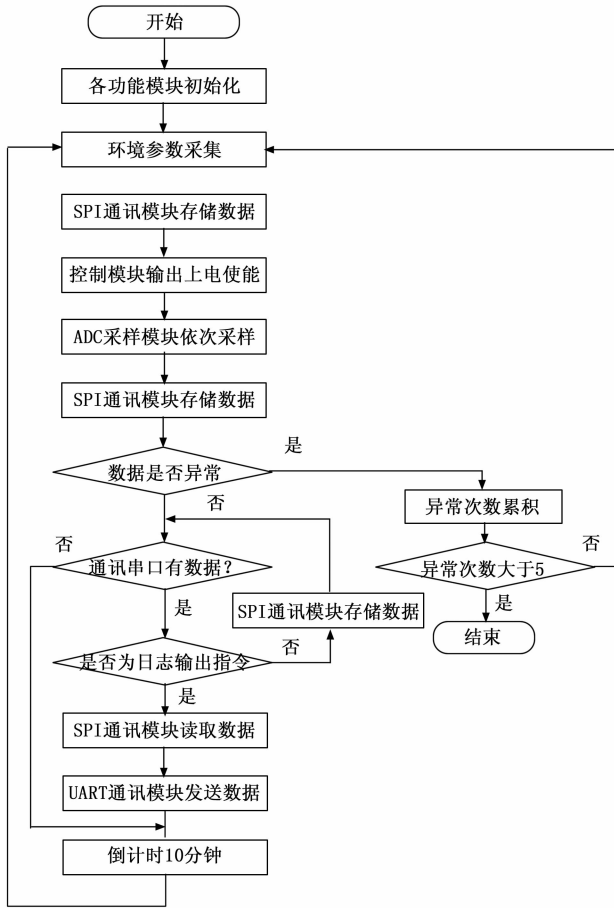


图 5 嵌入式软件流程图

记录；通过 UART 串口发送已经存储的日志记录；将日志记录信息通过 SPI 通讯模块进行数据存储。

### 3.2.2 ADC 采样模块

ADC 采样模块用于实现电气参数的采集输入。通过嵌入式微处理器 ADC 设备实现电气参数 AD 转换，为实现相应的功能软件设计时建立了 3 个相应的函数，分别为 ADC 控制器初始化函数 void Adc\_Init (void)、ADC 数据采集函数 u16 Get\_Adc (u8 ch) 以及 ADC 数据处理函数 u16 Get\_Adc\_Average (u8 ch, u8 times)。

初始化函数 void Adc\_Init (void) 用于实现对 ADC 控制器的初始化功能，包括时钟设置、引脚使能、通道选择、采样模式等。

ADC 数据采集函数 u16 Get\_Adc (u8 ch) 用于 ADC 数据的采集获取，具体如下：

```

u16 Get_Adc(u8 ch)// 返回值即为转换结果;ch 为指定通道
{
    ADC-RegularChannelConfig(ADC1, ch, 1, ADC_SampleTime_480Cycles);
    //设置指定 ADC 的规则组通道,一个序列,采样时间
    ADC_SoftwareStartConv(ADC1);
    //使能指定的 ADC1 的软件转换启动功能
    while(! ADC_GetFlagStatus(ADC1, ADC_FLAG_EOC));

```

//等待转换结束

```

return ADC_GetConversionValue(ADC1);
//返回最近一次 ADC1 规则组的转换结果
}

```

ADC 数据处理函数 u16 Get\_Adc\_Average (u8 ch, u8 times) 用于对采样的数据进行处理，具体为通过形参灵活设置平均参数，对采样的结果进行平均值计算，用以提高数据的准确性。

### 3.2.3 I2C 通讯模块

I2C 通讯模块用于与外部 I2C 设备通讯，实现环境参数的采集。I2C 总线在传送数据过程中共有 3 种类型信号，分别是：开始信号、结束信号和应答信号。嵌入式硬件处理模块环境参数的获取通过操作 I2C 设备进行相应的操作，具体函数包括 I2C 接口初始化函数 void IIC\_Init (void)、I2C 发送数据函数 void IIC\_Send\_Byte (u8 txd)、I2C 读取数据函数 u8 IIC\_Receive\_Byte (unsigned char ack)。

初始化函数 void IIC\_Init (void) 用于 IIC 控制器的初始化，包括时钟设置、设备地址、引脚功能设置、通讯模式设置等。

I2C 发送数据函数 void IIC\_Send\_Byte (u8 dataAddr, u8 byte) 用于向 I2C 设备发送控制指令，包括 start 信号、写操作头、温湿度转换指令、stop 指令等。

I2C 读取数据函数 u8 IIC\_Receive\_Byte (u8 dataAddr) 用于温湿度数据的获取。

### 3.2.4 SPI 通讯单元

日志记录通过 SPI 接口存储至 Flash 存储芯片中，因此系统日志存储与读取通过 SPI 通讯单元来实现。SPI 主机和从机都有一个串行移位寄存器，主机通过向它的 SPI 串行寄存器写入一个字节来发起一次传输。寄存器通过 MOSI 信号线将字节传送给从机，从机也将自己的移位寄存器中的内容通过 MISO 信号线返回给主机。这样，两个移位寄存器中的内容就被交换，实现读写操作。SPI 通讯单元包括初始化函数 void SPI\_Init (void)、数据传输函数 u8 SPI\_ReadWriteByte (u8 TxData)。

初始化函数 void SPI\_Init (void) 用于实现 SPI 控制器初始化，包括时钟设置、使能、工作模式设置等。

```

u8 SPI_ReadWriteByte(u8 TxData)
{
    while (SPI_I2S_GetFlagStatus(SPI1, SPI_I2S_FLAG_TXE)
    RESET){}
    //等待发送区空
    SPI_I2S_SendData(SPI1, TxData);
    //通过外设 SPIx 发送一个 byte 数据
    while (SPI_I2S_GetFlagStatus(SPI1, SPI_I2S_FLAG_RXNE)
    RESET){}
    //等待接收完
    return SPI_I2S_ReceiveData(SPI1);
    //返回通过 SPIx 最近接收的数据
}

```

### 3.2.5 UART 通讯模块

UART 通讯模块包括初始化函数 void uart\_init (u32 bound)、发送函数 void USART\_SendData (USARTx, Data); 接收函数 u16 USART\_ReceiveData (USARTx)。

初始化函数 void uart\_init (u32 bound) 用于对 UART 控制器进行初始化, 包括时钟设置、引脚使能、波特率设置等。

发送函数 void USART\_SendData (USARTx, Data) 用于向外部计算机设备发送数据, 通过 SPI 功能模块读取 Flash 存储单元的数据, 然后由 USART 发送函数实现数据传输。

接收函数 u16 USART\_ReceiveData (USARTx) 用于接收 CPU 串口发送过来的数据信息, 通过 SPI 功能模块将数据存储至 Flash 存储单元, 实现数据存储。

### 3.2.6 GPIO 控制模块

GPIO 控制模块用于实现嵌入式处理模块“PS\_ON”控制上电的功能。GPIO 控制模块通过控制引脚的高低电平实现控制功能, 包含初始化函数 void GPIO\_Init (void); 置位函数 GPIO\_SetBits (GPIOx, GPIO\_Pin\_x); 复位函数 GPIO\_ResetBits (GPIOx, GPIO\_Pin\_x)。

初始化函数 void GPIO\_Init (void) 用于对 GPIO 接口的初始化, 包括时钟、引脚功能、输入输出模式等。

置位函数 GPIO\_SetBits (GPIOx, GPIO\_Pin\_x) 用于将对应的 GPIO 接口进行置位, 实现高电平输出;

复位函数 GPIO\_ResetBits (GPIOx, GPIO\_Pin\_x) 用于将对应的 GPIO 接口进行复位, 实现低电平输出。

## 4 试验设计及结果验证

### 4.1 试验设计

根据全过程日志记录系统的设计要求, 试验时需要完整记录主板加电至操作系统完成启动整个过程的全部信息。试验时选用国产化 FT1500 平台搭载嵌入式硬件处理模块, 通过记录国产化平台启动过程日志记录验证计算机运行全过程日志记录系统的功能。试验包括国产化平台正常启动全过程日志记录和国产化平台启动异常两类场景, 用于全面验证计算机运行全过程日志记录系统的功能。

### 4.2 试验过程

试验选用国产飞腾 FT1500 平台进行测试验证。试验验证环境如图 6 所示。FT1500 主板上电后嵌入式硬件处理模块将及时采集环境参数信息, 并存储至 Flash 中, 然后 FT1500 主板开始正常上电启动。FT1500 主板上电启动时由 CPLD 控制, 嵌入式硬件处理模块将对每个电源电压进行采样, 并进行存储。FT1500 上电完毕后, CPU、内存及其他硬件开始进行初始化, 引导 BIOS、操作系统, 直至启动完成。启动完成后通过外部陪试计算机读取日志记录信息, 查看嵌入式硬件处理模块是否完整记录全过程日志记录信息。

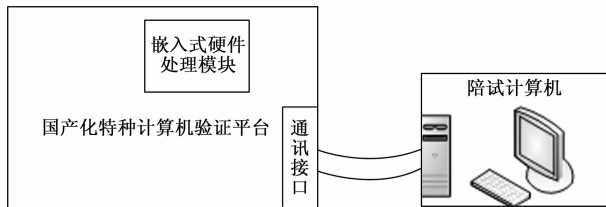


图 6 计算机运行全过程日志记录系统试验框图

为了全面验证系统的功能, 在试验过程中进行故障注入, 模拟 FT1500 主板启动过程中出现异常时系统运行情况。故障模拟时, 将 FT1500 主板 1.8 V 电源使能信号与电源芯片使能引脚断开, 造成 1.8 V 电源供电异常的故障。FT1500 主板上电过程中, 如果某个电源异常, 将造成主板无法正常启动, 处在上电过程的中间状态。故障注入完成后, 给 FT1500 主板上电, 最后查看系统记录的全过程日志记录是否与实际相符。

### 4.3 结果验证

利用串口调试软件接收正常启动时的日志记录, 查看日志记录是否完整, 记录的参数是否与实际参数一致。如图 7 所示, 为一次完整启动的日志记录, 记录参数与现场实际参数一致, 表明日志记录系统正常工作, 满足功能需求。

```

[14:25:13]power on
[14:25:23]temp1 24.6
[14:25:23]temp2 24.6
[14:25:23]temp3 24.5
[14:25:23]RH1 46.4
[14:25:23]RH2 46.5
[14:25:23]RH3 46.8
[14:25:23]power on
[14:25:23]12V 12.95V OK
[14:25:23]3.3V 3.33V OK
[14:25:24]5V 5.24V OK
[14:25:24]1.8V 0V ERR
[14:25:25]1.8V 0V ERR
[14:25:26]1.8V 0V ERR
[14:25:27]1.8V 0V ERR
[14:25:28]1.8V 0V ERR
  
```

(a) 各电压正常启动 (b) CPU开始初始化 (c) 引导操作系统启动

图 7 主板正常启动, 记录的日志信息

利用串口调试软件接收启动异常时的日志记录, 查看日志记录是否与注入的故障记录导致主板启动异常的日志记录一致, 查看记录的参数是否与实际参数一致。如图 8 所示, 为故障注入 1.8 V 电源异常时的日志记录, 与日志记录系统记录的信息一致, 表明日志记录系统正常工作。

```

[14:25:13]power on
[14:25:23]temp1 26.2
[14:25:23]temp2 26.0
[14:25:23]temp3 26.0
[14:25:23]RH1 46.6
[14:25:23]RH2 46.5
[14:25:23]RH3 46.8
[14:25:23]power on
[14:25:23]12V 12.95V OK
[14:25:23]3.3V 3.33V OK
[14:25:24]5V 5.24V OK
[14:25:24]1.8V 0V ERR
[14:25:25]1.8V 0V ERR
[14:25:26]1.8V 0V ERR
[14:25:27]1.8V 0V ERR
[14:25:28]1.8V 0V ERR
  
```

图 8 故障注入, 启动日志记录

## 5 结束语

特种计算机设备应用于特定的场合, 可靠性要求高, 每一次启动过程记录需要准确记录, 为可能面临的故障提供日志记录, 以便后续分析。计算机运行全过程日志记录系统记录了设备所处环境的物理参数, 同时记录硬件设备上电启动至操作系统完成启动整个过程的所有状态信息, 对及时掌握特种计算机运行状态、出现故障后进行的故障分析、故障隔离、故障定位、故障复现、归零处理措施具有重要意义。尤其是, 针对当前国产化替代过程中经常出现的偶发故障、单发故障的分析、处理、优化, 是一项不可或缺的工具。随着现代大数据和人工智能技术的发展, 将全过程日志记录与深度学习相结合, 通过故障模型训练、故障特征识别、故障演化规律获取, 最终进行有效的故障预测将成为可能<sup>[20-21]</sup>。

### 参考文献:

- [1] 魏江涛, 冯建峰, 姜美雷, 等. 船载系统日志自动分析软件的设计与实现 [J]. 计算机测量与控制, 2021, 29 (6): 142 - 146.
- [2] 石钊铭, 胡哲琨, 陈敬东. 国产特种计算机自动测试控制系统设计 [J]. 舰船电子工程, 2021, 41 (6): 164 - 168.
- [3] 中国国军标发行部. GJB9001C-2017 质量管理体系要求 [S]. 北京: 中国国军标发行部, 2017.
- [4] 全国宇航技术及其应用标准化技术委员会. GB/T 29076-2021 航天产品质量问题归零实施要求 [S]. 北京: 中国标准出版社, 2021.
- [5] 王国林, 介阳阳, 叶君好, 等. 基于日志挖掘的装备健康管理系统设计及实现 [J]. 计算机测量与控制, 2018, 26 (4): 112 - 115.
- [6] 张 骁, 应 时, 张 韬. 应用软件运行日志的收集与服务处理框架 [J]. 计算机工程与应用, 2018, 54 (10): 81 - 89.
- [7] 许 斌, 杨 漫. 机载嵌入式计算机通用操作系统故障管理软件设计 [J]. 航空电子技术, 2020, 51 (4): 38 - 41.

- [8] 徐永新. 高性能分布式日志系统研究与设计 [J]. 工业控制计算机, 2020, 33 (12): 44 - 46.
- [9] 天津飞腾信息技术有限公司. FT-1500A/16 高性能通用微处理器数据手册 (V2.7) [EB/OL]. (2018-06) [2022-07-31]. [https://www.phytium.com.cn/ftdownloads/document/download/datasheet/FT1500\\_datasheet\\_V2.7.pdf](https://www.phytium.com.cn/ftdownloads/document/download/datasheet/FT1500_datasheet_V2.7.pdf).
- [10] UEFI Forum, UEFI Platform Initialization Specification [S]. 2016.
- [11] 王 凯. 基于 UEFI 的计算机远程可信启动研究与实现 [D]. 北京: 北京工业大学, 2020.
- [12] 李雪源, 韩 康, 马晓晨. 机载信息系统中日志记录系统的设计与实现 [J]. 航空计算技术, 2022, 52 (1): 83 - 86.
- [13] 张世莉, 陈 亮, 陈亚兰. 归零工作对提升产品可靠性的效用研究 [J]. 微电子学, 2020, 50 (3): 450 - 454.
- [14] David A. Patterson, John L. Hennessy. Computer Organization and Design: The Hardware/Software Interface [M]. Fourth Edition. Singapore: Elsevier, 2012: 83 - 85.
- [15] 鸟 哥. 鸟哥的 Linux 私房菜 (3 版) [M]. 北京: 北京邮电出版社, 2010.
- [16] 兆易创新科技集团股份有限公司. DS-GD32F407xx-Datasheet-Rev2.6 [EB/OL]. [2023-03-26]. <https://www.gigadevice.com.cn/Public/Uploads/uploadfile/files/GD32F407xx-Datasheet-Rev2.6.pdf>.
- [17] 北京中科银河芯科技有限公司. 温湿度传感器 GXTH3X-DIS [EB/OL]. (2022-10) [2023-03-26]. <https://www.jfd-ic.com/Documents/GXHT>.
- [18] 陈诗军, 王慧强, 吕宏武, 等. 以构造为中心的底层软件复用方法 [J]. 计算机工程与设计, 2019, 40 (5): 1370 - 1375.
- [19] 怀 特. 嵌入式系统设计与实现 [M]. 余水清译. 北京: 机械工业出版社, 2013.
- [20] 吴立金, 夏 冉, 詹红燕, 等. 基于深度学习的故障预测技术研究 [J]. 计算机测量与控制, 2018, 26 (2): 9 - 12.
- [21] 刘洪歧, 陈远平, 马建化. 系统日志模板提取研究 [J]. 计算机系统运用, 2019, 28 (10): 239 - 244.

(上接第 171 页)

- [13] 王 宏, 柴天佑, 丁进良, 等. 数据驱动故障诊断与容错控制: 进展与可能的方向 [J]. 自动化学报, 2009, 35 (6): 739 - 747.
- [14] 王 承, 陈 光, 谢永乐, 等. 基于径向基函数神经网络的模拟/混合电路故障诊断 [J]. 电路与系统学报, 2007, 12 (2): 66 - 68.
- [15] 宁梓呈, 郑玉航, 王爱亮, 等. 基于 PCA-GA-BP 神经网络的状态评估算法 [J]. 兵工自动化, 2014, 33 (9): 27 - 30.
- [16] 李仿华, 王爱平, 姚丽娜, 等. 基于遗传优化的 RBF-BP 网络的实时故障检测 [J]. 微型机与应用, 2012, 31 (8): 90 - 92.
- [17] 王秋彦, 鞠建波, 宋振宇, 等. 故障诊断技术研究现状及发展趋势 [J]. 电子测量技术, 2009, 32 (4): 5 - 8.

- [18] 刘学勇. 详解 MATLAB/Simulink 通信系统仿真与建模 [M]. 北京: 电子工业出版社, 2001: 192 - 280.
- [19] 才 华, 刘广文, 陈广秋, 等. 基于神经网络的 MIMO-OFDM 信道估计 [J]. 计算机工程与应用, 2011, 47 (34): 114 - 116.
- [20] FARSAIDN, GOLDSMITH A. Neural network detection of data sequences in communication systems [J]. IEEE Transactions on Signal Processing, 2008, 56 (21): 5663 - 5678.
- [21] HE H, WEN C, JIN S, et al. Deep learning based channel estimation for beamspace mmwave massive mimo systems [J]. IEEE Wireless Communications Letters 2017 (5): 852 - 855.
- [22] 廖 勇, 花远肖, 姚海梅, 等. 高速移动环境下基于深度学习的信道估计方法 [1]. 电子学报, 2019, 47 (8): 1701 - 1707.