http://www.avionicstech.com.cn avionicstech@avic.com

DOI:10.12175/j.issn.1006-141X.2021.04.06

一种机载航电设备软件故障监控技术的设计与实现

杨 漫，许 斌，吉沛琦

（中国航空无线电电子研究所，上海 200233）

[摘 要] 机载航电设备软件故障监控技术是通过调用操作系统钩子函数、板级支持包中插装代码的方式实现的，可以为设备提供更加全面的故障监控能力，为故障快速定位提供重要依据，提升产品的可维护性。该技术可兼容多种处理器和多种操作系统，具有一定通用性。

[关键词] 故障监控；BSP；VxWorks

[中图分类号]TP311 [文献标识码]A [文章编号]1006-141X(2021)04-0033-05

The Design and Implementation of Software Fault Monitoring Technology for Airborne Avionics Equipment

YANG Man，XU-Bin，JI Pei-qi

（China National Aeronautical Radio Electronics Research Institute, Shanghai 200233, China）

Abstract: The software fault monitoring technology of airborne avionics described is implemented by calling hook function of operating system and inserting code in BSP. The implementation of this technology can provide more comprehensive fault monitoring capability for equipment. It provides an important basis for rapid fault location, improving the maintainability of products. The technology is implemented in software, compatible with multiple processors and multiple operating systems with universality.

Key words:fault monitoring; BSP; VxWorks

随着对机载航电设备功能、性能要求的不断提高，软、硬件设计复杂度日益提升。软件的规范化、流程化管理可以很大程度上减少软件类问题的产生，但并不能杜绝。机载航电类软件问题所引起的故障可能会导致飞机某些重要功能失效，并且由于机上环境与地面模拟环境存在不同，导致部分故障只在机上偶发，给故障排查带来了很大不便，给设备继续使用带来了隐患，甚至影响飞机任务的执行。在这种情况下，对软件故障的监控能力就显得尤为重要，可以为故障快速定位提供重要依据，提升产品软件层面的可维护性。

以某型项目显控处理机故障为例，某年外场频繁报机上画面冻结故障，严重影响飞行任务，由于该故障只在机上发生，实验室无法复现，且产品本身除部分机内测试（BIT：Build-in-Test）外，缺乏必要的故障监控手段，导致排故过程异常艰难。后经多次临时增加故障监控手段，通过分析故障日志找到最终问题。

上述问题反映了当前机载产品中故障监控能力的两点不足：

收稿日期：2021-01-20

引用格式：杨漫，许斌，吉沛琦 . 一种机载航电设备软件故障监控技术的设计与实现 [J]. 航空电子技术 ,2021,52(4)： 33-37.

（1）出厂阶段缺乏完整的故障监控手段，这种手段除包括传统的硬件 BIT，也应包括软件层面的异常捕获、中断监控、重启监控等；

（2）缺乏通用化故障监控设计方法，导致项目中故障监控五花八门，覆盖内容不全、可靠性差。

本文针对上述问题，提出了一种软件故障监控方法，通过调用操作系统钩子函数及板级支持包（BSP：Board Support Package）插装的形式实现，在与机载应用软件解耦的前提下，实现对常见软件故障的监控。

1 设计方案

本文提供的故障监控方法以软件的形式实现，功能上主要包括：中断监控模块，处理器异常（exception）监控模块，任务（进程）死循环监控模块，软重启监控模块，及 ARINC 653 操作系统下 HM 监控模块、系统调用（system call）监控模块等。

上述功能模块分布在 BSP 或应用软件中被调用，最终与 BSP、应用软件、操作系统集成使用，如图 1 所示。其中故障监控软件功能模块第一部分包括：任务（进程）死循环监控模块和 ARINC 653 操作系统下 HM 监控模块中的应用层实现部分。与应用程序中的原有功能模块在一个层面运行，通过监控原功能模块的全局数据及函数调用的方式实现功能。故障监控软件功能模块第二部分包括：中断监控模块、处理器异常（exception）监控模块、软重启监控模块、ARINC 653 操作系统下 HM 监控模块中的其他部分、系统调用（system call）监控模块，在 BSP 中通过函数调用的方式实现功能。

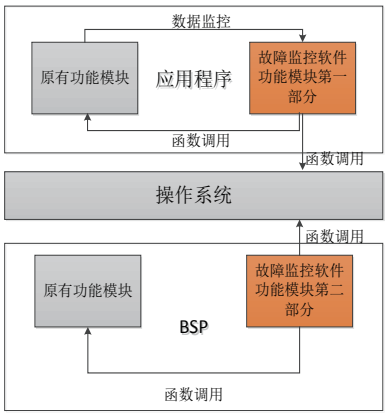

图 1 故障监控软件实现框图

1.1 中断监控模块

导致中断问题原因主要有以下两方面：

（1）中断服务程序处于“死循环”或“长时间阻塞”状态；

（2）中断由于非正常原因被频繁触发，导致 CPU 频繁处理中断。

这些问题会导致中断服务程序一致占用 CPU 资源，应用任务（进程）无法正常执行，从而导致产品功能不正常。

根据上述对中断常见原因的分析，通过在 BSP 中增加进出中断服务程序 “探针”的方式实现对是否有“死循环”或“长时间阻塞”情况的识别，如图 2 所示。

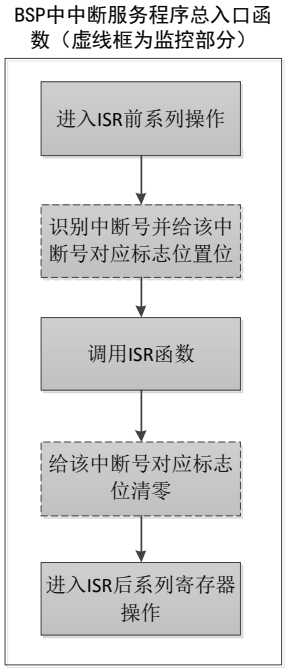

图 2 中断管理功能中识别“死循环”方法

BSP 中有针对处理器所有中断的总入口，如 PowerPC 中 0x500 异常向量对应的处理程序即为总入口，中断服务程序的实现是在进入总入口后，根据中断向量号，进入对应的中断服务程序。本文提供的方法是在中断服务程序总入口函数中，在进入对应中断向量的服务程序之前，识别本次中断的向量号，并调用通用函数接口完成该向量号对应非易失性存储器（NVM：Non-Volatile Memory）的标志位置位。中断服务程序随即正常执行，退出时再次调用提供的函数接口，完成对应 NVM 区域的标志位清位。这种情况下，本次上电结束后或故障发生时，可结合故障现象，通过读取 NVM 相关区域是否有未清除的中断向量，来识别哪个中断向量发生过中断复位程序“死循环”或“长时间阻塞”故障，

可对该服务程序进行针对性排查。

此外，本文通过在总入口中，进入中断服务程序之前，增加中断计数的方式，实现了对中断发生“不正常频繁触发”情况的监控。即在进入中断服务程序之前，获取中断向量号，并调用提供的函数接口完成以下工作：读取中断向量号对应的 NVM 区域中记录的发生次数，进行累加。同时，时钟中断将会在指定周期（在本文实现中该周期以宏定义形式存在，由用户定义）内完成对应 NVM 区域的清除，这种情况下可以计算在指定周期（同上）内进入某种中断的次数，并结合实际情况设定阈值，超出阈值的中断则认为发生“不正常频繁触发”的情况。需要注意的是，这种中断管理方式由于会延长中断处理时间，建议只在故障排查时或中断故障监控时开启。

1.2 处理器异常监控模块

常见的机载嵌入式操作系统中都具有针对处理器异常（exception）的处理入口，以钩子函数的形式提供给用户使用，该函数会在操作系统内核执行异常默认操作之前被调用。本文针对处理器异常类软件问题的监控就是利用操作系统钩子函数完成的。在 BSP 中实现钩子函数与监控函数的挂接，当有异常事件发生时，故障管理函数会在记录异常事件类型的同时，记录当前任务的堆栈信息、CPU 寄存器信息，以及时间信息、及其他 BIT 检测信息等内容。

1.3 任务死循环监控模块

任务死循环监控实现了对任务中由于代码编写问题导致的部分指令无限循环执行问题的监控，实现方法如图 3 所示，在被监控任务（图中 Task_A、Task_B）末尾增加执行计数（CNT_A、CNT_B），同时建立高优先级的监控任务，该任务周期被触发，执行对 CNT_A、CNT_B 的判断，如果设定周期内未变化，则认为该任务发生了死循环，调用操作系统接口函数，完成对该任务当前上下文的记录，包括当前任务的堆栈信息、CPU 寄存器信息，以及时间信息、及其他 BIT 检测信息等内容。

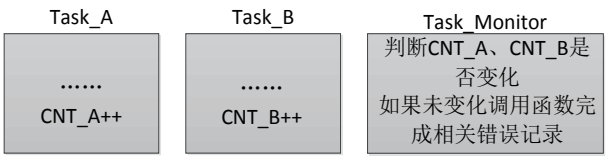

图 3 任务（进程）死循环监控方法

1.4 软重启监控模块

在操作系统检测到某些关键故障（Fatal error）的情况下，会调用 reboot 函数完成操作系统重启，此外，应用软件主动调用 reboot 函数也会造成软重启的发生。

常见机载嵌入式操作系统 BSP 中，通常会有操作系统调用 reboot 的入口函数，该函数完成中断保护、CPU 寄存器保护、Cache 关闭等操作，并将 PC 指针指向初始地址，完成软重启。

本文软重启监控功能就是基于该函数实现的，通过采用 BSP 插装代码的形式，在进入 reboot 入口函数时，将上下文信息（堆栈信息、CPU 寄存器信息、时间信息）记录至 NVM，完成软重启事件信息的保存。

1.5 调用监控模块

在 ARINC 653 操作系统中，分区应用会大量使用系统调用的方式，完成核心层（CoreOS）硬件驱动的调用，实现总线收发、数据存取等功能。通过系统调用执行的驱动函数中如果存在死循环或长时间阻塞的情况，会影响分区调度，严重的会导致所有分区应用停止执行。

本文通过在 BSP 或 coreOS 的系统调用总入口函数处插装代码的方式实现对系统调用故障的检测。实现方法如图 4 所示，在进入系统调用入口函数时，识别本次系统调用的向量号，并将标志位记录至 NVM，当入口函数正常退出时，清除 NVM 中的标志位。同时，在 CoreOS 中启动周期任务，如果在用户定义的周期范围内，识别到某个系统调用号存在标志位未清除的情况，则表明该系统调用处于异常状态，此时记录时间、系统调用号和系统调用参数等信息，供后续排查使用。

需要注意的是，这种通过周期任务实现的分区管理方式会在一定程度上降低系统的时间确定性，所以建议只在故障排查时或特别状态监控时开启，并且周期任务的周期应远大于分区调度的周期（建议 1000 倍以上），任务执行时间应远小于分区调度周期（建议小于 1 ms）。

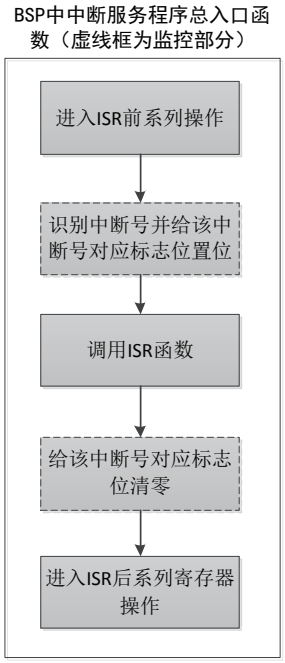

图 4 ARINC 653 操作系统下系统调用监控方法

1.6 健康管理监控模块

健康管理（HM：Health Monitor）在 ARINC 653-1 协议第 2.4 章节中定义，分为应用层级、分区层级和模块层级，用于应用软件、操作系统软件及硬件的故障检测。应用层级的故障监控可以通过操作系统钩子函数挂接的形式实现，分区层级和模块层级的故障监控钩子处理方式在 ARINC 653-1 中并未定义，而在 ARINC 653-5 协议中建议。分区层级和模块层级的故障处理挂接均在 HM 配置表中定义。

本文提出的 HM 监控方法中，针对应用层级故障的监控，基于操作系统提供的钩子函数实现。针对分区层级和模块层级故障，采用增加 HM 配置表中定义函数，并重新修改配置表的形式实现。可在故障发生时，对当前任务堆栈信息、时间信息、寄存器信息完成记录。

2 测试验证情况

2.1 软件测试情况

目前该方案软件已完成自测试及三方测试。作为机载软件的一部分已完成鉴定。代码及测试情况如表 1 所示。

表 1 代码及测试情况

功能描述	代码行数	测试用例个数	自测试发现问题个数	三方测试发现问题个数
中断管理	846	16	8	0
异常管理	2111	19	2	0
重启管理	662	16	4	2
运行日志管理	772	15	7	0
任务管理	1375	13	13	2
分区管理	903	22	3	0
HM 管理	647	48	6	0
解析管理	637	26	N/A	3
配置管理	1028		2	0
通用化适配	729		3	1
总数	9710	175	51	8

表 2 通用故障管理软件适配平台

	VxWorks653 2.x	VxWorks 5.x	VxWorks6.x	KV1	KV2	KV3
MPC7448	✓	✓	✓	✓	✓	N/A
MPC 8640	✓	✓	✓	✓	✓	N/A
MPC 8270	✓	✓	✓	✓	✓	N/A
P2020	✓	✓	✓	✓	✓	N/A
T1022	✓	✓	✓	✓	✓	N/A
IMX6	N/A	N/A	✓	N/A	N/A	N/A
Zynq7000	N/A	N/A	✓	N/A	N/A	N/A
FT2000A-2(-4)	N/A	N/A	N/A	✓	✓	✓

该软件应用之前，各项目中故障监控软件均为设计师自行设计，不具备通用的需求和设计方法，容易造成设计不完整，代码可靠性差等问题。

该软件涵盖了全部常见故障的监控功能，具有极强的通用性，支持 ARM、PowerPC 等常用处理器及 VxWorks、天脉等常用操作系统。使用该软件前后，项目中故障监控软件适配过程的情况对比如表 3 所示。

2.2 项目应用情况

目前该软件支持的处理器及操作系统环境如表 2 所示。

表 3 使用该软件前后情况对比表

	使用该软件之前适配故障监控情况	使用该软件之后适配故障监控情况
适配周期	7 天	0.5~2 天
功能覆盖	不够全面	目前总结的全部监控功能
软件质量	未完全测试	经过同行评审和软件测试

2.3 典型应用案例

该软件随产品使用过程中，在多次排故中发挥了关键作用，给故障排查提供了大量重要信息，大大缩短了排故周期，举例如下：

某项目外场使用过程中报故，该软件记录的故障日志中发现 HM 监控及异常监控中记录了显控模块应用分区内发生过 0x300 异常，且通过对 PC、LR 等 CPU 寄存器的反汇编分析，定位到了对应应用程序的代码位置，记录的故障发生时间与机上真实故障发生时间一致。后经实验室内针对该故障代码位置对应功能的烤机验证，故障可稳定复现。经排查发现，应用代码处存在指针越界访问，导致更改了存放数据地址的内存区域内容，造成问题发生。从外场报故到故障解决只用了 3 天时间。

某项目系统联试过程中，偶发 1553 总线无法正常上线问题，通过对该软件记录的故障日志分析发现，中断管理功能中记录了 1553 总线上线中断多达 5000 多次，后经进一步分析发现，故障是由于 1553 中断挂接错误导致，1553 驱动在中断挂接时使用了边沿触发，导致存在干扰的情况下会误触发中断，改为使用电平触发后故障消失。从实验室排故到问题最终定位只用了半天时间。

某项目外场使用过程中报故，该软件的记录日志信息中，异常管理功能中记录了产品内部多个 SRU 出现 0x200、0x300、0x700 异常，记录时间与故障发生时间一致；运行日志中记录了视频压缩芯片温度一直上升无法达到热平衡，最高时温度超过 110 度。经上述日志分析，可初步判断为产品散热能力存在问题，后经观察发现，产品与托架之间，负责给风扇供电的连接器存在变形，导致风扇工作不正常。

某项目外场使用过程中报故，该软件的记录日志信息中，异常管理功能中记录了多次通过 PCIE 总线访问视频压缩芯片过程发生 0x200 异常，该信息证明视频压缩芯片存在故障，返厂后对该芯片进行压力测试，可复现并定位。

上述通过个别典型案例，列举了通过该软件能够直接找到问题原因的故障情况，此外，该软件还在一定程度上起到了故障原因排除的作用，可让排故过程更加聚焦、更有说服力。

3　结束语

本文针对基于嵌入式操作系统的机载航电设备设计了一种软件故障监控技术，该技术以软件形式实现，以函数接口的形式被 BSP 软件、应用软件调用，具有一定通用性，适用于可支持 PowerPC、ARM、MIPS、X86 系列的机载航电常用处理器，及 VxWorks、天脉等常用机载操作系统。可在与机载应用软件解耦的前提下，实现常见软件故障的监控，能有效提高软件故障的定位能力，提高软件后期维护效率。

参 考 文 献

[1]STENFAN B, OLIVER H. Towards automatic reconfiguration of aviation software system[J], 35th IEEE Annual Computer Software and Applications Conference Workshops, 2011: 57-60.

[2] 周启平 , 等 . VxWorks 下设备驱动程序及 BSP 开发指南 [M]. 中国电力出版社 , 2004: 34-38.

[3] 朱剑锋 , 缪万胜 , 康介祥 . 基于堆栈回溯的异常处理 [J]. 计算机工程与设计 , 2014, 12: 4176-4180.

[4] 杨晓宁 , 曹原 . 嵌入式实时分区操作系统中健康监控机制的设计与实现 [J]. 电子设计工程 , 2013, (13): 101-103.

[5] 李运喜 , 叶宏 . 满足 ARINC653 要求机载操作系统的模型研究 [J]. 测控技术 , 2013, (32): 152-156.

[6] 康介祥 . 分布式安全杭电软件系统架构 [J]. 航空电子技术 , 2007, (4): 46-51.