

应用笔记

Application Note

文档编号: **AN1117**

增强微控制器电磁兼容性(**EMC**)的软件技术

版本: **V1.1**

1 引言

在现代电子设备中，电磁兼容性（**EMC**）已成为设计和开发微控制器系统时不可忽视的重要因素。随着技术的不断进步，微控制器被广泛应用于各种领域，包括工业自动化、消费电子、医疗设备等。然而，随着设备功能的复杂化和集成度的提高，**EMC** 问题也日益突出，可能导致设备间的干扰、性能下降甚至失效。因此，提高微控制器的 **EMC** 性能不仅是满足法规要求的必要条件，也是确保系统可靠性和用户体验的关键。

EMC 干扰可能引发一系列问题，例如通信信号丢失、数据错误、系统崩溃等。这些问题在关键应用场合尤为显著，比如医疗设备、航空航天系统和工业自动化控制等领域。若未能有效应对 **EMC** 干扰，可能导致软件故障，进而影响设备的安全性和可靠性。

为了提升微控制器的 **EMC** 性能，软件层面的技术手段显得尤为重要。本文将重点介绍两种主要的软件技术：预防技术和自动恢复技术。

- 预防技术：旨在通过优化软件设计和算法，降低系统受 **EMC** 干扰的风险。
- 自动恢复技术：侧重于在遭遇 **EMC** 干扰后，系统能够快速且自动地恢复到正常工作状态。这包括实现异常检测和处理机制，当系统检测到异常时，可自动重启或切换到安全模式。这种技术能够显著降低因干扰导致的停机时间，提高系统的可用性和用户体验。

目录

1 引言	1
2 自动恢复技术	3
2.1 复位源识别与标记	3
2.2 非易失性存储数据管理	4
2.3 本地看门狗监测与控制	5
2.4 使用 RAM 进行数据存储	6
3 预防技术	8
3.1 中断向量闲置优化	8
3.2 看门狗动态监控与恢复机制	8
3.3 筛选滤除非法和关键字节	10
3.4 闲置存储区安全防护	11
3.5 输入信号筛选与比较	11
3.6 冗余数据存储管理和交换	12
3.7 寄存器重配置与状态监测	12
3.8 模数转换结果均值化处理	13
3.9 数字模块预防方案	13
4 可实现的成果与效益	15
5 修订历史	16

2 自动恢复技术

在微控制器(MCU)的电磁兼容性(EMC)性能提升中,自动恢复技术是一种重要的机制,用于处理由于电磁干扰(EMI)引起的故障,如意外复位、程序计数器跳转和寄生中断等。

- **故障检测:** 自动恢复技术首先需要实时监测系统状态,以检测任何异常行为或故障。例如,可以通过监控程序计数器或特定的状态寄存器来判断是否发生了错误。
- **自我修复:** 一旦检测到故障,MCU 可以执行预定义的恢复策略。这可能包括重启系统、重新加载程序、或者从安全的备份存储(如 EEPROM)中恢复关键参数。
- **数据保护:** 在设计时,可以利用 RAM 和 EEPROM 的特性来保护数据。尽管 RAM 在掉电时会丢失数据,但 EEPROM 能够在掉电后保留数据,从而提供一种持久的存储解决方案。自动恢复技术可以在系统正常工作时定期将重要数据从 RAM 备份到 EEPROM,以减小因掉电导致的数据丢失风险。

2.1 复位源识别与标记

在微控制器(MCU)的电磁兼容性(EMC)自动恢复技术中,使用复位标签来辨识复位源是一种有效的策略。这种方法可以帮助系统在复位后快速恢复到适当的状态,而无需执行复杂的初始化程序。

2.1.1 复位源的类型

- **低电压监测器(LVD):** 当电压低于设定阈值时,MCU 会自动复位。
- **看门狗复位:** 如果软件未能在预定时间内刷新看门狗,MCU 会复位。
- **上电复位(POR):** 在 MCU 上电时,系统会进行复位以确保正常启动。
- **热复位:** 当复位引脚处于低电平状态时,可能会导致寄生或外部复位。

2.1.2 复位寄存器的作用

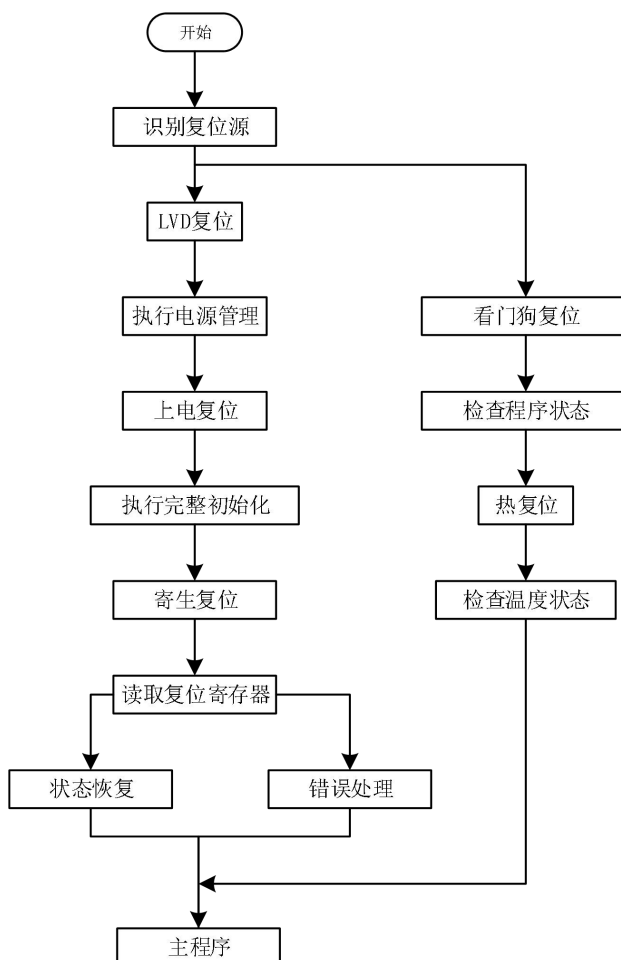
MCU 中有一个专门的复位寄存器,用于标记复位源。只要 MCU 处于上电状态,该信息就会一直保存。

2.1.3 程序中的应用

- **测试复位寄存器:** 在程序开始时,MCU 可以测试复位寄存器的值,以确定复位的来源。这一过程可以帮助系统决定是否需要执行特定的恢复程序。
- **分支到上下文恢复程序:** 根据检测到的复位源,系统可以选择性的分支到相应的上下文恢复程序,而不是重新启动通常复杂且耗时的“上电复位”初始化程序。这种方法提高了系统的响应速度和效率。
- **管理寄生复位:** 检测和管理寄生复位是非常重要的,因为它们是微控制器 EMC 故障最常见的原因。通过识别复位源,系统可以采取适当的措施来处理这些故障,确保稳定性。

使用复位标签辨识复位源的技术, 通过在复位寄存器中保存复位信息, 使得 MCU 能够在复位后快速恢复到适当的状态。

图 1 复位源识别



2.2 非易失性存储数据管理

在微控制器(MCU)兼容性(EMC)自动恢复技术中, 数据存储在非易失性存储器(如数据 EEPROM)中是一个重要的概念。

2.2.1 非易失性存储器的特点

- 存储时间: 数据存储在非易失性存储器(如 EEPROM)所需的时间明显长于存储到随机存取存储器(RAM)的时间。这意味着在进行数据写入时, 系统可能会面临更长的延迟。
- EMC 干扰的风险: 在编程期间, EMC 干扰会降低系统稳定性, 导致系统出现异常复位并终止程序运行, 甚至还会导致数据损坏, 尤其是在写入过程中。

2.2.2 数据保护策略

为了防止数据损坏，通常采取以下策略。

- 冗余存储：数据应该存储在冗余存储器中。这意味着在多个存储位置保存相同的数据副本，以防止单点故障。
- 有效标记：通过使用有效标记(如校验和或时间戳)来保持数据的一致性。这些标记可以帮助系统在读取数据时验证其有效性，确保数据未被损坏。

在每个系统启动之后，实际使用数据之前，需要进行检查数据存储器中内容的有效性，以确保数据在上次写入后未被损坏，这可以通过读取有效标记并进行比较来实现。

2.3 本地看门狗监测与控制

在微控制器(MCU)的电磁兼容性(EMC)自动恢复技术中，使用看门狗定时器(WDT)是一种有效的策略，用于监控软件的执行状态并确保系统的可靠性。

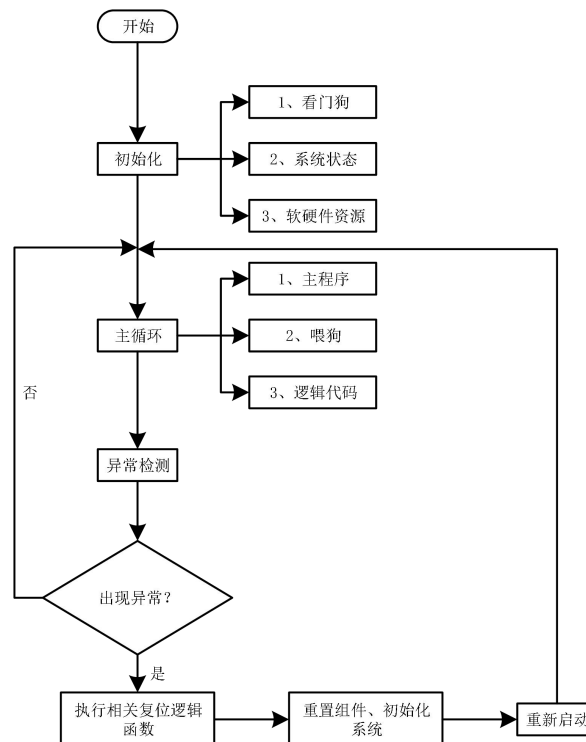
看门狗定时器的作用：

- 监控软件执行：看门狗定时器用于监控 MCU 中软件程序的执行。如果程序在预定时间内未能正常运行(例如，进入死循环或发生异常)，看门狗定时器会超时并触发复位或执行恢复程序。
- 避免“定时炸弹”式的使用：编程人员常常将看门狗定时器视为一个“定时炸弹”，仅仅将其刷新为最大值，以获得尽可能宽的裕量。这种做法是不理想的，因为它与程序的实际执行时间没有直接关系，可能导致系统在异常情况下无法及时响应。

更好的使用方法：

- 定期检查执行时间：更好的方法是使用看门狗定时器寄存器来检查单独软件程序的执行时间。通过设定合理的时间间隔，程序可以在每个关键步骤后刷新看门狗定时器。
- 快速响应异常：如果在看门狗计数结束之前未能刷新定时器，系统应迅速做出反应，执行立即复位或转到软件恢复程序。这种机制确保了在软件出现异常时，系统能够及时恢复，避免长时间的无响应状态。

图 2 本地看门狗监测与控制



2.4 使用 RAM 进行数据存储

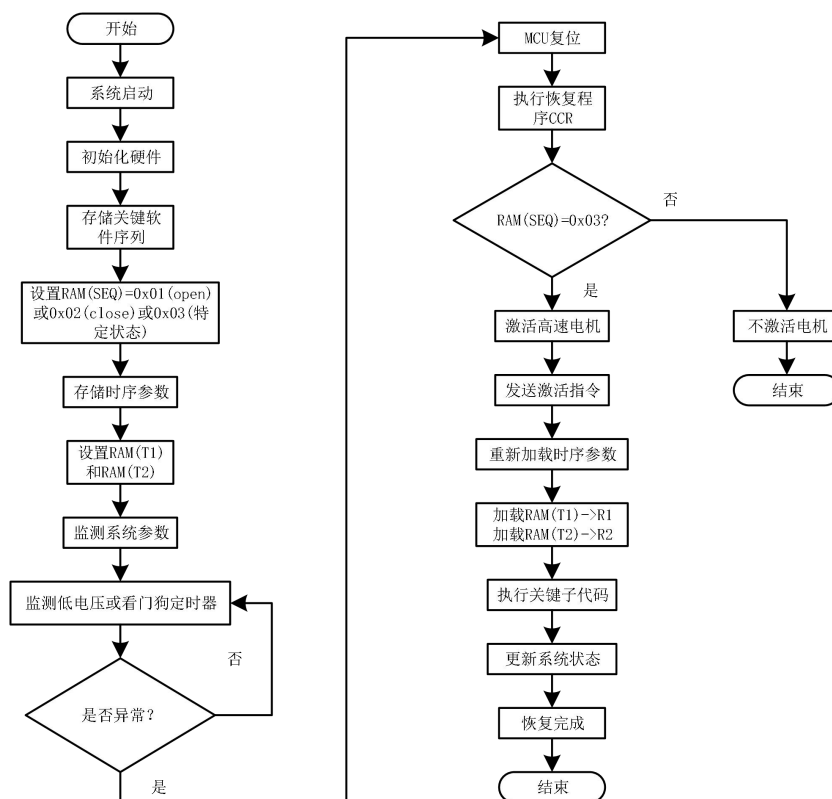
在微控制器(MCU)的电磁兼容性(EMC)自动恢复技术中,将关键内容保存在 RAM 中是一种有效的策略。这种方法可以在发生 EMC 事件导致 MCU 复位时,快速恢复系统的正常运行。如图 3 就是一个示例。

- **关键软件序列存储:** 关键的控制指令(如门的打开或关闭指令、高速电机控制指令)被存储在 RAM 的特定字节中(例如, RAM(SEQ))。这种存储方式允许在 MCU 复位后快速访问和恢复这些指令。
- **复位后的恢复机制:** 当 EMC 事件导致 MCU 复位时,系统可以通过检查 RAM(SEQ)的值来判断是否需要执行特定的操作。例如,只有当 RAM(SEQ) = 03 时,高速电机才会被激活。这种条件检查确保了系统在恢复时的安全性和可靠性。
- **应用参数的存储:** 应用参数(如时序值 T1 和 T2)也存储在 RAM 中。这意味着在软件失控或 MCU 复位的情况下,系统能够恢复到最后的状态,而无需外部干预。
- **恢复程序(CRR):** 恢复程序负责在 MCU 复位后重新加载最后的门命令和时序参数。该程序会自动执行,无需用户干预,从而提高了系统的自动化和可靠性。
- **低电压监测和看门狗:** 通过低电压监测或看门狗定时器,系统能够检测到潜在的故障并触发复位。这种机制确保了在不稳定的电源条件下,MCU 能够安全地恢复到已知的良好状态。

在 RAM 区中保存关键内容的技术,通过快速恢复机制和自动化的恢复程序,显著提高了 MCU 在面对 EMC 事件时的可靠性和稳健性。这种方法确保了系统能够在复位后迅速恢复到正常工作状态,

减少了对外部干预的依赖。

图 3 自动恢复软件示例



注意：并非所有微控制器（MCU）都具备此功能。一般情况下，RAM 区域的数据在 MCU 掉电或软复位后会被清零，无法保留先前的状态，只有具备备份 RAM 区域的 MCU 才能实现数据的保存。备份 RAM 的设计目的是在掉电或复位时仍能保持数据，与普通 RAM 不同，备份 RAM 通常会通过电池或其他电源在掉电时维持供电，从而防止数据丢失。例如，APM32F4 系列 MCU 就具备备份 RAM 区域，其地址为 0x40024000。

表格 1 自动恢复技术总结

技术方式	优点	缺点	实现方法
使用 RAM 进行数据存储	读写速度快、适合频繁更新的数据	易失性存储，断电后数据丢失，仅适用于具有备份 RAM 区域的微控制器(MCU)	程序中定义备份 RAM 的使用区域，通过特定的地址进行读写操作
本地看门狗监测与控制	自动监测系统状态、提高系统可靠性	无	配置看门狗定时器，定期重置以防止系统挂起
复位源识别与标记	精确识别复位原因、有助于故障排查	需要额外的存储空间	在复位时记录复位源信息，使用特定寄存器或内存位置
非易失性存储数据管理	数据在断电后仍然保留、适合长期存储	写入速度慢且写入次数有限	使用 EEPROM 或 Flash 存储，定期更新数据

3 预防技术

在微控制器(MCU)的电磁兼容性(EMC)性能提升中,预防技术是一种重要的策略,旨在增强产品的稳健性和抗干扰能力。

3.1 中断向量闲置优化

在电磁兼容性(EMC)预防技术中,中断向量闲置优化技术是指在微控制器系统中有效管理和处理中断向量,以减少不必要的中断触发和系统资源的浪费。这种管理方式可以帮助提高系统的稳定性和抗干扰能力,尤其是在存在电磁干扰的环境中。

闲置中断是指在某些情况下,微控制器接收到的中断信号并不需要立即处理,或者该中断的处理并不影响系统的实时性和功能。管理这些闲置中断可以避免系统频繁切换上下文,从而提高效率。

中断向量管理涉及到对中断服务例程(ISR)的有效调度和管理。通过合理配置中断优先级和处理逻辑,可以确保系统在接收到中断时能够快速响应,同时避免不必要的中断处理。

在 APM32 中,启动文件为每个中断向量分配了一个默认的中断服务函数入口地址,称为 **Default Handler**。在 NVIC 中,每个中断向量都有一个标志位,用于指示该中断是否开启。实际的中断产生仅会发生在那些已开启的中断向量上。

3.2 看门狗动态监控与恢复机制

看门狗定时器(Watchdog Timer, WDT)是一种用于监控系统运行状态的硬件或软件机制,主要用于确保系统在出现故障或异常时能够自动重启或恢复正常工作。

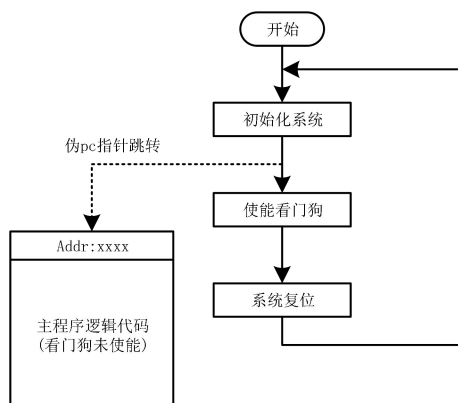
3.2.1 工作原理

- 初始化: 在系统启动时,初始化看门狗定时器,设定其超时时间(即“喂狗”的时间间隔)。
- 喂狗: 在正常运行过程中,系统定期向看门狗发送信号(通常称为“喂狗”),以重置定时器的计时器。
- 超时检测: 如果系统在设定的时间内没有喂狗,看门狗定时器将超时并触发复位信号,导致系统重启。
- 复位机制: 看门狗可以通过硬件复位或软件复位的方式来恢复系统,具体取决于设计。这种方式最简单,但也会出现个别不当的情况。

3.2.2 看门狗配置不当: 启动后使能延迟问题

这种情况是指在系统启动过程中,看门狗定时器的使能(或激活)延迟到达预定的时间点,导致在系统启动的初始阶段,可能无法及时监测系统的运行状态。具体可见图 4。

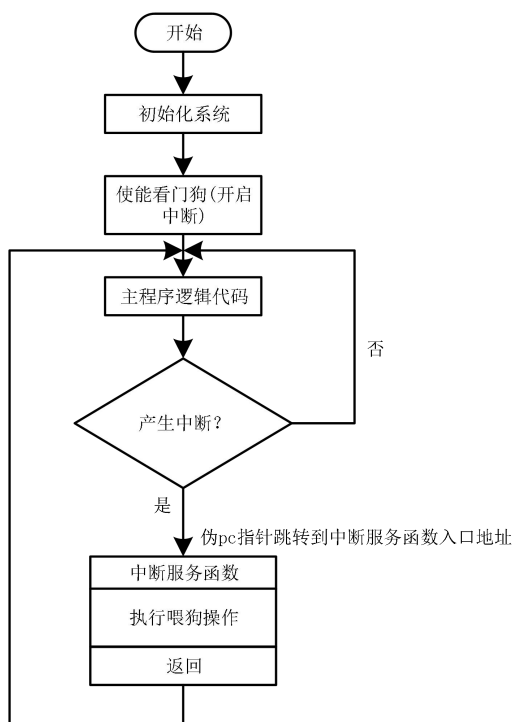
图 4 看门狗启用后使能延迟



3.2.3 看门狗配置不当：在中断服务例程中进行喂狗操作

这种情况是指在中断服务函数中执行喂狗操作，即在中断处理过程中定期重置看门狗的定时器，以防止其超时并触发系统复位。具体可见图 5。

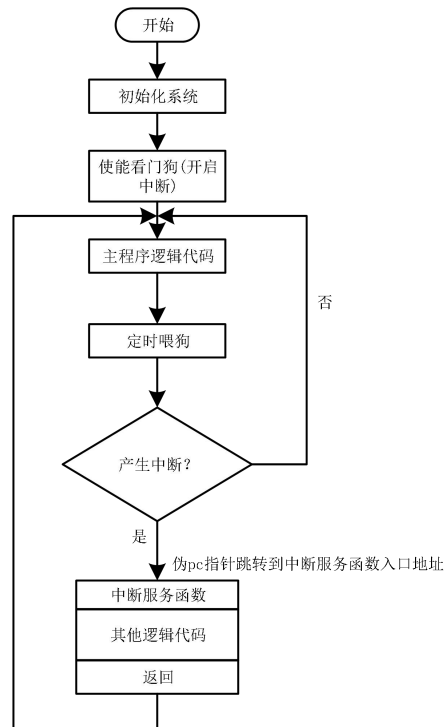
图 5 在看门狗中断服务函数中进行喂狗操作



3.2.4 看门狗的正确配置

一般来说，避免以上两种情况，就是看门狗的正常使用。流程如下。

图 6 看门狗正确配置流程图



3.3 筛选滤除非法和关键字节

3.3.1 关键字节

关键字节是指特定的指令或数据字节，这些字节用于唤醒微控制器，使其从低功耗或休眠状态恢复到正常工作状态。关键字节的存在可能导致系统在意外情况下进入不稳定状态，尤其是在程序计数器(PC)损坏时。

去除方法：

- 检查程序的 .list 文件：编译后生成的 .list 文件包含了汇编代码和机器码，可以用来识别关键字节的出现。
- 插入 NOP 指令：在关键字字节之前插入一个或多个 NOP(无操作)指令，可以改变后续指令的地址，从而移位关键字节的值，避免其被错误解码。

3.3.2 非法字节

非法字节是指在 MCU 的指令集架构(Instruction Set Architecture, ISA)中未定义的字节序列或指令。当执行这样的字节时，MCU 无法识别并解释为有效指令。

在某些架构中，非法字节可能被解释为“无操作(NOP, No Operation)”指令，即 MCU 会跳过该指令的执行，继续执行下一条指令。这通常不会引发错误，但也无法产生实际的计算结果。

在一些微控制器上，遇到非法字节可能会触发复位，导致微控制器重新启动。这是为了保护系统，避免在执行未知或错误的指令状态下继续运行，确保系统的稳定性和安全性。

去除方法:

- 识别非法字节：通过分析程序代码，识别出不符合指令集的字节值。
- 使用关键字去除技术：采用与去除关键字相似的方法，插入 **NOP** 指令或其他有效指令，以避免非法字节的出现。

3.4 闲置存储区安全防护

在预防技术中，“闲置存储区安全防护”是指对微控制器或嵌入式系统中未使用的存储区域进行管理和保护，以防止这些区域在遭遇电磁干扰(EMC)时被意外修改或损坏。

针对电磁干扰(EMC)，可以采取以下操作来保护闲置程序的存储区域。

(1) 读写保护

- 读保护：在微控制器中启用读保护功能，防止外部设备读取和修改程序存储区域。
- 将未使用的存储区域设置为只读，防止在运行时被意外写入。

(2) 冗余存储

- 备份程序：在不同的存储区域中保留程序的冗余副本，以便在主存储区域受到干扰时能够快速恢复。
- 校验和：在程序存储区域中使用校验和或哈希值，定期检查存储内容的完整性，及时发现并纠正错误。

(3) 故障检测与恢复

- 自检机制：在系统启动时进行自检，验证程序存储区域的完整性，确保系统能够正常运行。
- 故障异常捕获：利用微控制器的故障异常处理机制来增强系统在遭遇电磁干扰(EMC)的可靠性和安全性。**APM32** 微控制器能够捕获非法的存储访问及非法编程行为，并使用未定义的指令填充闲置存储区域，这样可以在程序计数器为空时提供一种保护机制，通过提高故障异常的使用率，系统能够在发生错误时迅速恢复到安全状态。这种方式可以防止系统执行错误的指令，并确保了系统的稳定性和可靠性，尤其是在高干扰环境中。

3.5 输入信号筛选与比较

在电磁干扰(EMC)预防技术中，输入信号筛选与比较是一种用于减少电磁干扰对电子设备影响的技术。它通过在设备的输入端添加滤波器，来抑制不必要的高频噪声和干扰信号，从而提高是设备的抗干扰能力。

在继续编程执行和验证状态之前，最佳做法是通过程序对输入引脚的稳定性进行多次检查，以避

免外部噪声对输入电路造成不必要的尖峰响应。

这种方法易于实现，并且不需要额外的成本。

3.6 冗余数据存储管理和交换

3.6.1 冗余数据存储

冗余数据存储是指在不同的存储区域中保存相同或互补的数据，以防止由于电磁干扰(EMI)或其他极端条件导致的数据损坏。这种方法可以提高数据的可靠性，确保在发生故障时能够恢复数据。

实施方法:

- **双倍互补值存储:** 将数据的双倍互补值存储在非临近的存储区域中。例如，如果在一个存储单元中存储数据，则在另一个物理位置存储该数据的互补值。
- **使用奇偶校验位:** 在数据存储时附加奇偶校验位，以便在读取数据时检查其完整性。奇偶校验位可以帮助检测单个比特的错误。
- **错误校正码(ECC):** 使用 ECC 技术来存储数据，这种方法不仅可以检测错误，还可以自动修正某些类型的错误。ECC 通常用于内存模块中，以提高数据的可靠性。

3.6.2 数据交换

数据交换是指在不同的存储区域或设备之间传输数据，以确保在一个存储区域发生故障时，能够从另一个存储区域恢复数据。这种方法可以提高系统的容错能力。

实施方法:

- **定期备份数据:** 定期将数据从主存储区域备份到备用存储区域，以防止数据丢失。
- **使用冗余存储设备:** 在系统中使用多个存储设备(如 SD 卡、闪存等)，并在这些设备之间进行数据同步，以确保数据的冗余存储。

3.7 寄存器重配置与状态监测

3.7.1 寄存器重配置

寄存器重配置是指定期或在特定条件下重新设置或更新微控制器中的寄存器内容，以防止由于电磁干扰(EMI)导致的寄存器内容意外改变。寄存器通常包括时钟控制寄存器、I/O 配置寄存器和数据寄存器，这些寄存器与芯片的输出焊盘紧密相关，容易受到干扰。

实施方法:

- **定期刷新寄存器:** 通过定时器或主循环定期读取和写入寄存器的值，以确保其内容保持在预期状态。

- 条件刷新: 在特定事件或条件下(如外部中断或状态变化)触发寄存器的重编程, 以应对潜在的干扰。

3.7.2 寄存器状态监测

寄存器状态监测是指定期验证寄存器的内容是否符合预期值, 以检测是否发生了意外的变化。这种检查可以帮助及时发现由于 EMC 干扰导致的寄存器内容异常。

实施方法:

- 读取和比较寄存器值: 定期读取寄存器的当前值, 并与预设的安全值进行比较。如果发现不一致, 则可以采取相应的纠正措施。
- 集成到主循环或定时任务中: 将常规检查集成到主循环或定时任务中, 以确保系统持续监控寄存器状态。

3.8 模数转换结果均值化处理

模数转换结果均值化处理技术是通过对多次采样的输入信号进行平均处理, 得到的一个更稳定的输出值。这种方法可以有效减少由于噪声引起的误差, 提高测量的准确性。

结果分析:

- 噪声抑制: 通过多次采样并取平均值, 可以显著降低随机噪声的影响, 因为噪声通常是随机的, 而真实信号则是相对稳定的。
- 数据稳定性: 平均值提供了一个更可靠的信号表示, 尤其是在存在电磁干扰(EMI)或其他噪声源的环境中。
- 误差减少: 在进行 A/D 转换时, 噪声可能导致单次转换结果的不准确。通过多次转换并取平均, 可以减少这些误差, 提高系统的整体性能。

使用方法:

- A/D 转换操作: 在进行模拟信号的数字化时, 使用 ADC(模数转换器)进行多次转换。例如, 可以设置 ADC 进行 10 次转换。
- 结果存储: 将每次转换的结果存储在 RAM 中。可以使用数组或缓冲区来保存这些值, 以便后续处理。
- 计算平均值: 计算存储结果的平均值, 以获得更准确的信号表示。

3.9 数字模块预防方案

对于 PMU 模块受 EMC 冲击有可能使 LDO 电压上漂超过 1.7V 从而导致数字模块跑飞的风险。

系统解决方案:

软件代码在系统上电初始化时对复位寄存器中的所有数字模块复位一次;

如以下 G32M3101 SDK 例程里对数字模块复位的代码:

```
static void SystemReset(void)
{
    // ...此处省略其他代码

    /* Reset all digital module */
    SCU->APBRST = 0x8477;
    SCU->APBRST = 0x00000000U;
    SCU->AHBRST = 0x73;
    SCU->AHBRST = 0x00000000U;

    // ...此处省略其他代码
}
```

表格 2 预防技术总结

预防技术	优点	缺点	实现方法
看门狗动态监控与恢复机制	自动重启系统以防止死锁并、提高系统可靠性	可能导致意外重启、需要合理配置时间	设置看门狗定时器, 定期重置看门狗计数器
闲置存储区安全防护	防止意外写入和数据损坏、提高安全性	可能浪费存储空间、需要额外的管理	将闲置区域标记为只读或使用保护机制
输入信号筛选与比较	提高系统对噪声的抵抗力、增强数据完整性	可能增加延迟、需要复杂的算法	实现输入信号的滤波和阈值比较
中断向量闲置优化	减少中断冲突、提高系统稳定性	需要额外的管理和配置	设计中断向量表, 确保闲置中断不被触发
筛选滤除非法和关键字节	提高代码安全性、减少潜在漏洞	可能影响代码的灵活性	使用静态分析工具检查和清理代码
模数转换结果均值化处理	提高测量精度、减少噪声影响	计算复杂度增加、可能增加延迟	对多次采样结果进行平均处理
寄存器重配置和状态监测	提高系统的自我检测能力、及时发现故障	可能增加系统负担、需要额外的资源	定期检查寄存器状态, 重编程以恢复正常状态
冗余数据存储管理和交换	提高数据可靠性、防止数据丢失	增加存储需求、实现较复杂	使用双倍互补值存储、奇偶校验或 ECC 技术

4 可实现的成果与效益

(1) 增强的抗干扰能力

- 结果: 经过优化的软件可以提高系统对电磁干扰(EMI)和静电放电(ESD)的抵抗力。
- 影响: 系统在高 EMC 电平下仍能正常工作, 减少因外部干扰导致的故障。

(2) 自动故障检测与恢复

- 结果: 软件可以实现对 EMC 干扰引起的故障进行实时监测和检测。
- 影响: 一旦检测到异常, 系统能够自动启动恢复程序, 进行复位或重新初始化, 确保系统快速恢复到正常状态。

(3) 提高系统可靠性

- 结果: 通过软件技术的提升, 系统能够在面对潜在的 EMC 问题时, 保持更高的可靠性。
- 影响: 减少了因 EMC 干扰导致的系统崩溃或功能失效的风险, 提升了用户的信任度和满意度。

(4) 优化的资源管理

- 结果: 软件可以优化资源的使用, 确保在 EMC 干扰情况下, 系统仍能高效运行。
- 影响: 提高了系统的整体性能, 降低了能耗, 延长了设备的使用寿命。

(5) 故障分析与记录

- 结果: 系统能够记录 EMC 干扰事件及其影响, 提供故障分析数据。
- 影响: 有助于后续的故障排查和系统优化, 提升了设计和维护的效率。

(6) 用户体验的提升

- 系统在面对 EMC 干扰时的稳定性和可靠性提升, 用户体验得到改善。
- 用户在使用过程中遇到的问题减少, 系统的可用性和响应速度提高。

通过软件技术的提升, 微控制器在 EMC 方面的表现可以显著改善。这不仅提高了系统的抗干扰能力和可靠性, 还增强了故障检测与恢复能力, 最终提升了用户体验和系统的整体性能。

5 修订历史

表格 3 文件版本历史

日期	版本	变更历史
2024.10	1.0	新建
2025.10	1.1	增加 3.9 数字模块预防方案 章节

声明

本手册由珠海极海半导体有限公司（以下简称“极海”）制订并发布，所列内容均受商标、著作权、软件著作权相关法律法规保护，极海保留随时更正、修改本手册的权利。使用极海产品前请仔细阅读本手册，一旦使用产品则表明您（以下称“用户”）已知悉并接受本手册的所有内容。用户必须按照相关法律法规和本手册的要求使用极海产品。

1、权利所有

本手册仅应当被用于与极海所提供的对应型号的芯片产品、软件产品搭配使用，未经极海许可，任何单位或个人均不得以任何理由或方式对本手册的全部或部分内容进行复制、抄录、修改、编辑或传播。

本手册中所列带有“®”或“™”的“极海”或“Geehy”字样或图形均为极海的商标，其他在极海产品上显示的产品或服务名称均为其各自所有者的财产。

2、无知识产权许可

极海拥有本手册所涉及的全部权利、所有权及知识产权。

极海不应因销售、分发极海产品及本手册而被视为将任何知识产权的许可或权利明示或默示地授予用户。

如果本手册中涉及任何第三方的产品、服务或知识产权，不应被视为极海授权用户使用前述第三方产品、服务或知识产权，也不应被视为极海对第三方产品、服务或知识产权提供任何形式的保证，包括但不限于任何第三方知识产权的非侵权保证，除非极海在销售订单或销售合同中另有约定。

3、版本更新

用户在下单购买极海产品时可获取相应产品的最新版的手册。

如果本手册中所述的内容与极海产品不一致的，应以极海销售订单或销售合同中的约定为准。

4、信息可靠性

本手册相关数据经极海实验室或合作的第三方测试机构批量测试获得，但本手册相关数据难免会出现校正笔误或因测试环境差异所导致的误差，因此用户应当理解，极海对本手册中可能出现的该等错误无需承担任何责任。本手册相关数据仅用于指导用户作为性能参数参照，不构成极海对任何产品性能方面的保证。

用户应根据自身需求选择合适的极海产品，并对极海产品的应用适用性进行有效验证和测试，以确认极海产品满足用户自身的需求、相应标准、安全或其它可靠性要求；若因用户未充分对极海产品进行有效验证和测试而致使用户损失的，极海不承担任何责任。

5、合规要求

用户在使用本手册及所搭配的极海产品时，应遵守当地所适用的所有法律法规。用户应了解产品可能受到产品供应商、极海、极海经销商及用户所在地等各国有关出口、再出口或其它法律的限制，用户（代表其本身、子公司及关联企业）应同意并保证遵守所有关于取得极海产品及/技术与直接产品的出口和再出口适用法律与法规。

6、免责声明

本手册由极海“按原样”（as is）提供，在适用法律所允许的范围内，极海不提供任何形式的明示或暗示担保，包括但不限于对产品适销性和特定用途适用性的担保。

极海产品并非设计、授权或担保适合用于军事、生命保障系统、污染控制或有害物质管理系统中的关键部件，亦非设计、授权或担保适合用于在产品失效或故障时可导致人员受伤、死亡、财产或环境损害的应用。

如果产品未标明“汽车级”，则表示不适用于汽车应用。如果用户对产品的应用超出极海提供的规格、应用领域、规范，极海不承担任何责任。

用户应该确保对产品的应用符合相应标准以及功能安全、信息安全、环境标准等要求。用户对极海产品的选择和使用负全部的责任。对于用户后续在针对极海产品进行设计、使用的过程中所引起的任何纠纷，极海概不承担责任。

7、责任限制

在任何情况下，除非适用法律要求或书面同意，否则极海和/或以“按原样”形式提供本手册及产品的任何第三方均不承担损害赔偿责任，包括任何一般、特殊因使用或无法使用本手册及产品而产生的直接、间接或附带损害（包括但不限于数据丢失或数据不准确，或用户或第三方遭受的损失），这涵盖了可能导致的人身安全、财产或环境损害等情况，对于这些损害极海概不承担责任。

8、适用范围

本手册的信息用以取代本手册所有早期版本所提供的信息。

©2025 珠海极海半导体有限公司 – 保留所有权利