

## 嵌入式系统的掉电保护方案设计

在嵌入式系统设计过程中,系统的掉电保护越来越受到重视。本文介绍的方法是在用 ARM7 系列芯片 S3C4510B 和  $\mu$  Clinux 构建的嵌入式平台上实现的。整个掉电保护实现的基本思路是:产生掉电信号,捕捉掉电信号和处理掉电信号。重点介绍这个过程的具体实现。

系统防掉电设计的目的是:采用一种机制,使得系统在意外失去供电的情况下,可以保证系统运行状态的确定性以及记录数据的完整性;当系统供电恢复后,现场数据可以及时恢复,避免应用系统产生混乱。我们知道,在嵌入式系统设计与开发中越来越多地应用嵌入式操作系统。由于操作系统的引入,数据的读写往往是通过文件的方式完成,而不是直接对存储单元地址操作。用文件读写方式操作数据,在程序的运行过程中往往将数据暂存在易失性的存储空间,如 SDRAM,一旦系统意外失电,这些数据往往被丢失。因此,当系统意外失电时必须采取一定的措施进行系统的掉电保护,以避免系统产生混乱。总的说来,防掉电程序的主要思路就是:产生掉电信号,捕捉掉电信号,处理掉电信号和数据以及现场状态的恢复。

如果不引入操作系统,直接对存储单元进行数据操作,每次操作的数据量小,可以利用中断服务的方式进行掉电保护;而用文件的方式进行数据操作,数据量一般比较大,因此基于中断服务的方式进行掉电保护已经不再可靠。本文研究的对象是基于操作系统的较为复杂的嵌入式系统设计过程中的掉电保护。

### 1 掉电保护方案实现的系统基础

掉电保护是在由 ARM 体系的硬件平台和  $\mu$  Clinux 嵌入式操作系统的基础上实现的。

ARM7 系列的微处理器支持八种类型的中断处理。外部中断请求会在外部中断引脚有效(一般是低电平),并且程序状态寄存器相关位(即 CPSR 的 I 控制位)设置为允许时得到处理器响应。响应后处理器进入中断工作模式,PC 被装入中断向量 0x00000018。在这个地址单元存放中断服务程序入口地址,中断服务程序就可以被执行。在掉电保护方案中,中断服务程序很简单,就是将表示掉电的全局变量置位即可。这样可以缩短程序执行时间。

Flash 存储器是一种可在系统(in system)进行电擦写,电后信息不丢失的存储器。它具有低功耗、大容量、可整片或分扇区在系统编程(烧写)、擦除等特点,并且可由内部嵌入的算法完成对芯片的操作,因而在各种嵌入式系统中得到了广泛的应用。作为一种非易失性存储器,Flash 在系统中通常用于存放程序代码、常量表以及一些在系统掉电后需要保存的用户数据等。常用的 Flash 为 8 位或 16 位的数据宽度,编程电压为单 3.3V。

与Flash存储器相比较,SDRAM不具有掉电保持数据的特性,但其存取速度大大高于Flash存储器,且具有读/写的属性,因此,SDRAM在系统中主要用作程序的运行空间、数据及堆栈区。当系统启动时,CPU首先从复位地址0x0处读取启动代码,在完成系统的初始化后,程序代码一般应调入SDRAM中运行,以提高系统的运行速度,同时,系统及用户堆栈、运行数据也都放在SDRAM中。SDRAM的存储单元可以理解为一个电容,总是倾向于放电,为避免数据丢失,必须定时刷新(充电)。因此,要在系统中使用SDRAM,就要求微处理器具有刷新控制逻辑,或在系统中另外加入刷新控制逻辑电路,特别的情况是在系统失电后,要采取一种有效的机制确保将sDRAM中的数据写入Flash中。

## 2 基于掉电保护方案的硬件设计

图1是一种典型的嵌入式系统硬件设计方案。系统的微处理器采用S3c4510B,是基于ARM7体系结构的。SDRAM是一种易失性存储器作为程序的运行空间,类似于PC机的内存;Flash作为程序存储空间是非易失性的。程序运行过程中的数据往往缓存在sDRAM中,在系统失电时必须写往Flash。

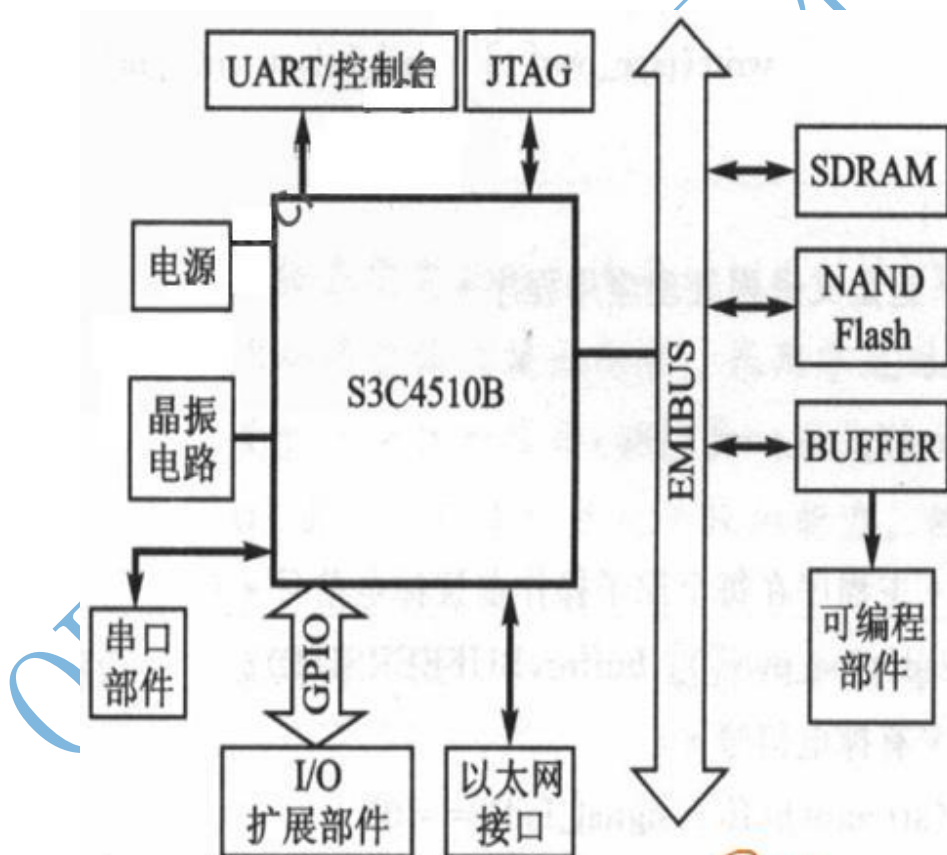


图1 典型的嵌入式系统硬件设计方案

在系统中,需要使用5V和3.3V的直流稳压电源。其中,S3C4510B及部分外围器件需3.3V电源,另外部分器件需5V电源。为简化系统电源电路的设计,要求整个系统的输入电压为高质量的5V的直流稳压电源。有别于一般的电源回路

设计,本系统的电源回路设计过程中增加了有关掉电保护的设计。包含这个设计的系统电源电路如图 2 所示。

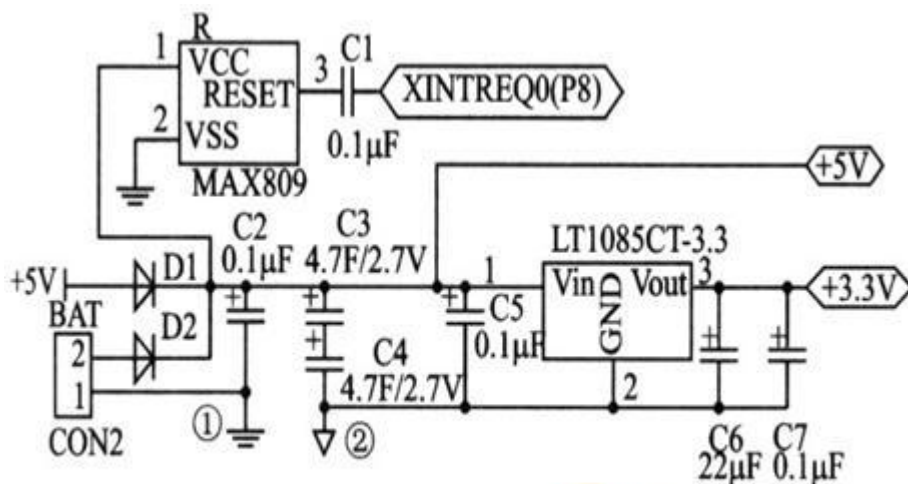


图 2 电源电路原电子工程网

这个电源回路除了可以提供 5v 和 3.3v 的电源以外,还为系统掉电保护提供了延时及预警功能,通过软件的配合可以实现系统的掉电保护机制。正常情况下,由供电回路 1 给整个系统供电。当系统由于意外原因掉电时,由于输入的比较电压降低,这样 MAX809 模块输出电压产生翻转为系统提供掉电中断预警信号,中断请求通过外部中断引脚 XREQ0 产生;同时供电回路 2 开始启用。通过大电容 C3、c4 放电,继续为系统提供一段供电电压,支持掉电中断服务程序完成。供电回路 2 只给最小系统供电,并不给耗电量的外围部件供电。这样,给最小系统的供电时间足够长,可以完成敏感数据的保护操作。

通过软件测算,电容放电可供最小系统工作时间在 0.5~4.5S 之间。这种测算方法很简单.编写一个掉电中断服务子程序,这个程序只是不断进行时间刷新操作。同样,可以通过软件测定在这段时间里向 Flash 擦写 2~3MB。可见,在采用这种硬件体制的情况下,系统掉电保护能够得到可靠的保证。

### 3 掉电信号处理软件方法的实现

在  $\mu$  Clinux 系统下,掉电信号的捕捉有两种方式可以进行。一种是运用系统调用,即采用 `void(*signal(intsig, void(*func)(int)))(int)`。这个函数可以为特定的中断信号安排制订的执行函数,用参数 `func` 传递。在  $\mu$  Clinux 中,共有 31 个系统中断信号,其中掉电信号为 SIGPWR。假设掉电中断服务处理程序为 `void interrupt_service(int)`,则中断服务与信号关联的方式为:`signal(SIGPWR, interrupt_service)`。这种方式充分利用系统调用,实现简单。在掉电保护方案设计初期也是采用这种机制。但事实证明这种机制并不可靠,其原因是 Linux 内核产生和管理信号的机制并不完善,有可能存在信号丢失。查阅有关 Unix 或 Linux 的相关资料,可以发现这种状况也普遍存在于某些其他版本的 Linux 和 Unix 中。

另一种方式是采用守候进程的方式，开通一个进程，此进程专门等待中断信号。主程序根据数据操作对象的不同，将自己的流程方案划分成若干原子操作，所谓原子操作即划定的程序块要么完全执行，要么不执行。每个操作对应惟一状态标志。在每个原子操作前，主进程都将会通过管道通信的方式阅读中断信号。如果中断信号产生，主进程首先保存状态标志，然后将相关数据写往 Flash 后退出，电源恢复后，主进程首先根据标志字确定系统恢复方案。图 3 用流程图的方式实现这一过程。

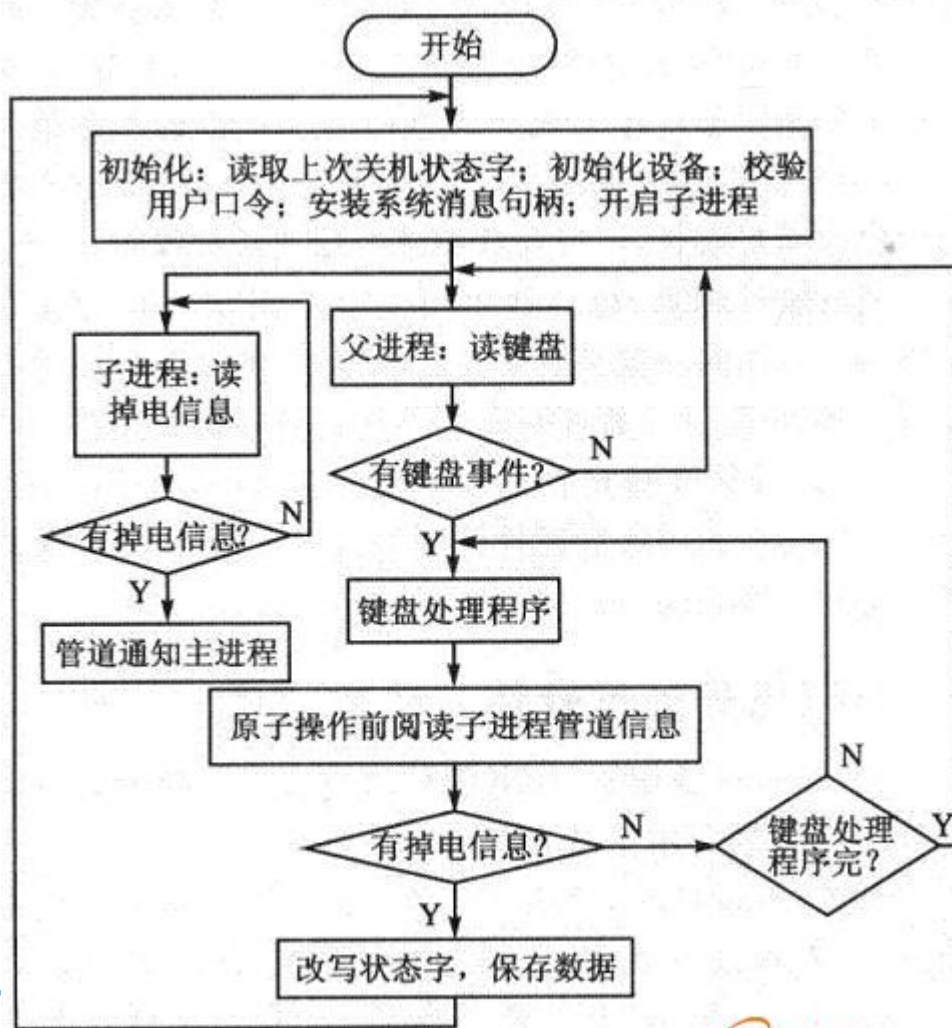


图 3 利用守候进程方式捕捉掉电

### 结语

基于该方案设计的税控收款机在实际运行过程中，掉电保护功能完备。此掉电保护设计方法应用对象基于 ARM 和  $\mu$  Clinux 构建的嵌入式系统，在 32 位嵌入式系统开发中具有典型代表意义。因此在嵌入式系统设计中具有推广价值。