# 单片机关键任务优先级的实现

卢永华
贵州梅岭电源有限公司，贵阳 遵义 563000

[摘要]与总体产品联调时，需要各个单机系统严格按照总体要求，进行数据输出，时间的偏差将出现系统异常，控制失败等不稳定情况产生，甚至影响到产品安全。因此必须确保某些关键任务的优先执行。单片机任务优先级一般有两种方式实现，基于单片机中断服务的中断函数进行实现和基于实时操作系统的任务调度实现。基于中断服务函数实现的任务优先级对单片机硬件资源有要求，而对于实时操作系统的任务调度方式，仅需一个定时器就可完成多任务多优先级的管理。
[关键词]关键任务；优先级；执行方法
DOI：10.33142/sca.v5i5.7325　　中图分类号：TP311　　文献标识码：A

# The Realization of the Priority of the Key Tasks of Single-chip Microcomputer

LU Yonghua
Guizhou Meiling Power Supply Co., Ltd., Zunyi, Guiyang, 563000, China

**Abstract:** During the joint debugging with the overall product, each stand-alone system needs to output data in strict accordance with the overall requirements, and the time deviation will lead to system exceptions, control failures and other instabilities, even affecting the product safety. It is therefore important to ensure the priority implementation of certain key tasks. There are generally two ways to realize the task priority of single-chip microcomputer, namely, the interrupt function based on single-chip microcomputer interrupt service and the task scheduling based on real-time operating system. The task priority based on interrupt service function requires the hardware resources of single-chip microcomputer, while for the task scheduling mode of real-time operating system, only one timer can complete the management of multi task and multi priority.
**Keywords:** key tasks; priority; execution methods

## 引言

参与某产品联调时，总体要求每间隔 5ms 向总控发送一次关键数据。当系统联调运行时，总控会产生超时报警，报警内容是通信超时。经过排查排除了硬件问题、电磁干扰问题、程序逻辑错误未正常发送数据等问题。通过报警时间比对，发现该报警出现时间没有规律性。通过示波器查看发现，其发送数据周期没有严格按照 5ms 间隔时间发送，发送时间落在 5ms 区间段内，任意时间点都可能会进行关键数据的传递，无法预测下次一次发送数据的准确时间，当系统在规定时间内未接收到数据时，产生系统报警。

经过对程序进行逻辑分析，出现问题原因是单片机运行任务是顺序执行，只有轮到发送数据任务执行时，才能发送数据，如果其他任务占用执行时间过长，将会导致发送任务不能在 5ms 时间内再次获得运行机会，因此也无法按时发送数据，造成数据超时问题。

## 1 关键任务优先执行方法

### 1.1 查找问题

下位机程序任务流程如下图 1 所示：

下位机程序按照项目功能需求，将不同功能划分为不同任务，根据每个任务特点，制定的间隔时间不一致，如对 RS485 等通信口监听时，其响应时间在 50ms 满足要求，自然环境下温度变化缓慢，因此温度采集 500ms 一次也满足要求。通过计时器进行技术，当 5ms 时置位 5ms 任务标志位，10ms 时置位 5ms 和 10ms 任务标志位，通过任务标志位定义了任务执行频率，优先级高的任务得到更多执行次数。该种任务执行方式称为任务协同方式，当一个任务执行时，必须等到该任务执行完成，才能执行下一个任务。当某一时刻，多个时间任务被置位时，其按照顺序结构运行程序，任务需要排队执行，实时性不高。

下位机程序使用任务协同方式进行运行，分别定义了 5ms，10ms，20ms，50ms，100ms，200ms，500ms 等任务。所有的任务基于顺序执行，其中 5ms 程序 critical_task 作为关键任务。某个时刻，如定时器在计数到 500ms 时，其上的 5ms，10ms，20ms，50ms，100ms，200ms 时间标志位被置位，任务均得到执行，导致 500ms 这一时刻需要顺序执行很多任务，如在 5ms 内不能执行完全部任务，那么下一次的关键任务程序 critical_task 将不能按时被执行，导致输出超时情况产生。 为解决超时问题，必须提升 critical_task 任务的优先级，提升任务优先级的方式较多，常用的方式有中断服务函数（前后台系统）、实时操作系统实现。

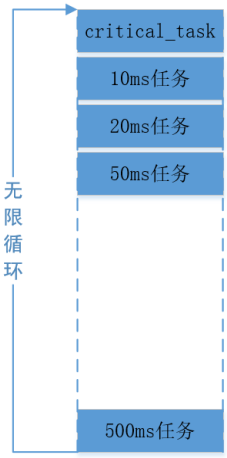


图 1　主程序逻辑

### 1.2 关键任务任务由前后台系统保证

该项目使用的是飞思卡尔系列单片机，该单片机是单核处理器，不能同时执行多个任务。

从主程序架构上看，该种顺序执行方式不能保证关键任务 critical_task 的优先执行。因此应该使用某种方式能够中断当前正在顺序执行的任务，转向执行优先级较高的任务。

单片机中断是指正在执行一项任务 A，然后突然停止任务 A 去执行任务 B，执行完任务 B 再回来继续执行任务 A 的过程。单片机中断有很多触发源，如定时器中断、外部按键中断、通信发送、接收数据中断，每个中断源都可以打断正在执行的任务，转向执行中断任务，中断任务执行完毕后，继续回到当前的任务进行未完成的操作，利用单片机的中断特性，能够保证某些代码的及时执行。将某些关键任务放入中断服务函数中，就能打断顺序执行任务而优先执行中断任务，使用该种方式提升了关键任务的优先级。

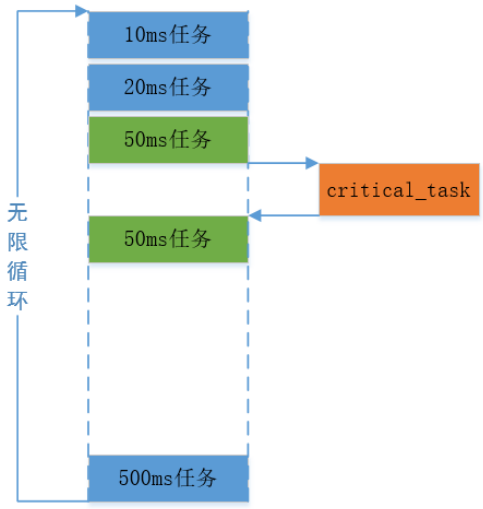


图 2　中断任务执行

单片机中断源的产生有很多方式，与产品联调问题是不能在 5ms 时准确的进行数据传输，因此需要在 5ms 时产生一次任务中断，以执行发送任务。为了满足上述要求，选择定时器中断可满足要求。

定时器中断是指单片机内部有一个从 0 开始向上(向下)计数的计数器，每一次计数时间均相同，设置一个计数目标值，当计数器计数到目标值时，会产生一个计数中断，中断后单片机可以打断当前正在执行的程序跳转执行中断服务程序。在中断服务程序中，清除中断服务向量，使得计数器归 0 重新计数，以此不断循环达到每隔一定时间就产生一次服务中断的工作模式。

如图 3 配置相关定时器参数，产生 5ms 中断计数。

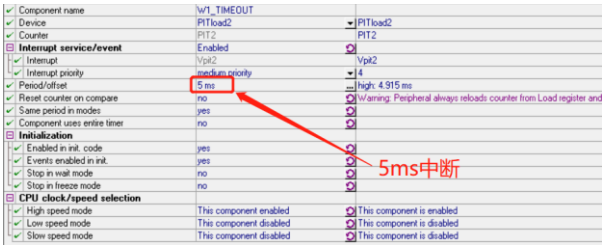


图 3　定时器配置

如图 4 在中断方法中添加关键任务 critical_task 的执行。

```
void W1_TIMEOUT_OnInterrupt(void)
{
  /* Write your code here ... */
  critical_task();
}
```

图 4　关键任务添加

我们将原程序中的 critical_task 任务从 5ms 任务重移除，添加到 Count5ms_OnInterrupt 中断服务程序中，通过前后台方式实现打断其他正在执行任务来保证 critical_task 的优先执行。程序更改后 critical_task 能够 5ms 一次准确输出，系统报警现象被消除。采用此种方法虽然简单，但是突出问题有几个：

（1）使用中断方式来保证优先级需要占用一个中断来完成，浪费资源。

（2）当有多个关键任务需要执行时，会出现中断嵌套，关键任务仍然会被打断执行。

（3）违背了中断中只执行不耗时简单操作的原则，仍然存在隐患。

（4）不能对不同任务进行不同权重的 CPU 使用权划分。

### 1.3 关键任务由 RTOS 系统保证

RTOS 全称为 RealTime OS，就是实时操作系统，强调的是：实时性。在实时操作系统中，我们可以把要实现的功能划分为多个任务，每个任务负责实现其中的一部分，

每个任务都是一个简单的程序，通常是一个死循环，RTOS 的核心内容在于：实时内核。

RTOS 内核负责管理所有任务，内核决定了运行哪个任务，何时停止当前任务切换到其他任务，这个是内核的多任务管理能力。可剥夺型内核可以剥夺其他任务的 CPU 使用权，它总是运行就绪任务中优先级最高的任务。

任务调度系统任务切换模式见图 5。系统开始运行在一个最低优先级空闲任务下。运行过程中达到任务调度点，任务调度系统查询当前就绪最高优先级任务，查询到图中绿色高优先级任务时，将打断低优先级任务，开始执行高优先级任务。中断服务程序具有最高优先级，中断服务程序发生执行中断服务。中断服务执行完毕，调度系统将查询系统中就绪最高优先级任务，此时查询到更高级优先任务，将执行更高优先级任务（图 5 中黄色任务）。更高优先级任务执行完毕后主动释放 CPU 占用权，调度系统查询就绪最优先级任务，此时查询到高优先级任务（图 5 中绿色任务），则从上次断点处继续执行，待任务执行完毕后主动释放 CPU 占用权，此时低优先级任务获得 CPU 使用权限，继续执行剩余任务。

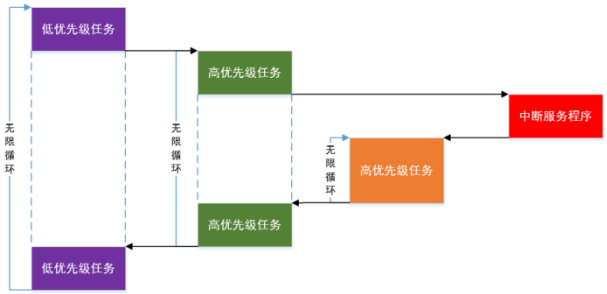


图 5 RTOS 系统任务切换模式

基于上述系统调度模型，我们将任务的优先级进行划分，critical_task 任务设置为 5ms 执行一次，优先级设置为 16（数值越大优先级越高），其余 10ms、20ms、50ms、100ms、200ms、500ms 优先级设置为 8。当调度器开启后，对所有任务进行优先级查找，优先级最高且就绪程序可以运行，因此 critical_task 任务被执行，执行完成后调用 RTOS 系统中的延时函数，延时函数能够使 critical_task 任务放弃单片机的使用权，其他低优先级的任务很够得到单片机的使用权，进而执行低优先级任务。当 critical_task 任务延时函数时间到，critical_task 任务转变为就绪态，调度系统将立即打断当前正在执行任务，转向执行 critical_task 任务，以此保证 critical_task 任务每经过一定延时后能够立刻得到单片机的使用权。critical_task 任务执行完毕，调度器将继续查找下一个就绪的优先级最高任务，若未找到，将返回之前被打断程序继续进行。

从 RTOS 的工作模式中可以得知，任务调度系统能保证高优先级任务获得 CPU 使用权，确保关键任务及时运行。与方式 2 中的前后台系统相比，RTOS 系统只占用一个计数器资源，就可以对任务进行不同优先级划分，实现对 CPU 使用权的划分。

程序更改后通过验证，具有自定义任务中最高优先级的 critical_task 任务能够保证 5ms 一次准确输出，系统报警现象被消除。这是一种典型的实时系统。该系统也有其缺陷：

（1）运行 RTOS 框架，会产生额外性能开销。

（2）需要移植 RTOS 框架，开发难度大，开发时间长，技能要求高。

## 2 结论

当系统中有多个任务执行，要保证关键任务能够及时运行，使用前后台系统及 RTOS 系统两种方式都能实现。当系统简单，逻辑不复杂，要保证的关键任务少，可以采用中断服务函数的方法，利用中断服务函数可以打断正在执行后台服务的特性，保证关键任务能够按照设定的时间间隔循环往复执行，提高关键任务的相应能力。当系统功能复杂，要保证不同任务的优先级执行，使用系统中断方式没有足够的硬件资源保证大量任务的优先执行，使用 RTOS 方式，仅使用一个定时器资源并配合任务调度方法就可以实现大量任务的优先级管理。根据实际情况选择合适的技术才是最优选择。

## [参考文献]

[1]朱迪.FreeRTOS 实时操作系统任务调度优化的研究与实现[D].南京:南京邮电大学,2015.

[2]左中凯.FreeRTOS 源码详解与应用开发---基于STM32[D].北京:北京航空航天大学出版社,2017.

[2]胡曙辉,陈健.几种嵌入式实时操作系统的分析与比较[J].单片机与嵌入式系统应用,2007(5):5-8.

[3]崔建华,孙红胜,王保进.硬件实时操作系统的设计与实现[J].电子技术与应用,2008,34(5):34-37.

作者简介：卢永华（1983.10-），男，毕业院校：哈尔滨工业大学，专业：工程力学。