



1. 一种面向嵌入式系统低功耗实时任务调度的简化方法，其特征在于：

1) 实时任务的时间片调度：

当实时任务队列中，存在多个实时任务时，系统将通过时间片的方式对系统中的任务进行调度，以满足各个实时任务的需要；

时间片调度的方式是将处理器的运行时间划分等分的时间片，当任务在处理器上运行时，只能运行一个时间片长度，当时间片用完后，被强制停止执行，换下一个任务执行；

2) 新实时任务的频率比计算：

在实时系统中，一个实时任务它有以下几个参数是确定：任务的开始时间、实时任务的最终期限、以及任务的运行时间；

频率比就是将上面的三个参数进行融合所产生的，频率比的计算方法如下面的公式所示：

$$pf=p/(e-b)$$

其中

pf 是实时任务的频率比，

p 是实时任务实际需要运行的时间，

e 是实时任务的最终期限，

b 是实时任务的开始时间；

3) 总实时任务的频率比计算：

当新的实时任务加入到任务运行队列，系统将计算新实时任务的频率比计算，然后计算总任务的频率比；

总实时任务频率比的计算方法如下面的公式所示：

$$pfsum=pfsum+T.pf$$

其中 pfsum 是总实时任务的频率比，它的初始值为 0，

“=” 表示为赋值，即 pfsum 的新值等于 pfsum 原来得值加上 T.pf，

T 是新的实时任务的控制字，

pf 是新实时任务的频率比，

T.pf 表示实时任务 T 的频率比；

4) 处理器频率设置：

调度系统根据总实时任务的频率比计算出一个新的频率值，当新的频率值大于现在的频率时，将处理器的频率设置成新计算出来的频率，否则不变。

## 一种面向嵌入式系统低功耗实时任务调度的简化方法

### 技术领域

本发明涉及基于嵌入式系统软件节能技术领域，特别是涉及一种面向嵌入式系统低功耗实时任务调度的简化方法。

### 背景技术

在便携式嵌入式设备电源管理领域，目前的困难在于既要满足便携式终端对电源供电的要求，又要做到占用空间小、重量轻和供电时间更长。下一代消费类电子产品的电源解决方案重点应该集中在硬件和软件两方面技术，包括：(1) 在小巧外形尺寸下，如何实现所需电源性能的工艺和技术，涉及热管理、降噪、电池管理和功能整合等技术；(2) 动态功率管理技术，它取决于 CPU 性能、软件、中间件以及用户对更换电池的时间间隔等要求；(3) 动态功率管理技术对操作系统内核和驱动器，以及应用编程接口(API)对驱动器、中间件和应用本身的影响。

现在嵌入式设备的功能变得越来越强大，功能也越来越丰富。随着嵌入式设备功能越来越多，用户对嵌入式设备电池的能量需求也越来越高，现有的锂离子电池已经越来越难以满足消费者对正常使用时间的要求。对此，业界主要采取两种方法，一是开发具备更高能量密度的新型电池技术，如燃料电池，在可以预见的 5 年内，电池技术不可能有很大的突破；二是在电池的能量转换效率和节能方面下功夫。在目前新的高能电池技术(如燃料电池)仍不成熟的情况下，下一代手持设备的电源管理只能从提高电源利用率和降低功耗这二个方面着手。

如何延长电池的使用寿命，以及尽量减少电池能量的消耗已经成为嵌入式领域的一个研究热点。现在主要集中在硬件设计和软件优化两方面。其中软件优化方面现在主要包括系统软件和应用软件两方面。系统软件主要集中在编译器和操作系统内核两块。

在操作系统领域，现在主要的电源管理方法是利用操作系统内核，动态的调整系统处理器和总线的频率，降低系统的整体能耗。而且系统可以通过动态频率指令改变系统状态，是系统处于低功耗状态，以达到节能的目的。在编译器方面，现在主要通过编译器在编译应用程序阶段，对代码进行优化，使代码尽可能的紧凑以及访问设备尽量集中，以达到节能的目的。

上面的方法中，实现起来都需要比较繁琐的过程，而且没有考虑实时性，在现在嵌入式系统领域的应用存在一定的限制。

## 发明内容

本发明的目的在于提供一种面向嵌入式系统低功耗实时任务调度的简化方法。

本发明解决其技术问题所采用的技术方案是：

### 1) 实时任务的时间片调度：

当实时任务队列中，存在多个实时任务时，系统将通过时间片的方式对系统中的任务进行调度，以满足各个实时任务的需要；

时间片调度的方式是将处理器的运行时间划分等分的时间片，当任务在处理器上运行时，只能运行一个时间片长度，当时间片用完后，被强制停止执行，换下一个任务执行；

### 2) 新实时任务的频率比计算：

在实时系统中，一个实时任务它有以下几个参数是确定：任务的开始时间、实时任务的最终期限、以及任务的运行时间；

频率比就是将上面的三个参数进行融合所产生的，频率比的计算方法如下面的公式所示：

$$pf=p/(e-b)$$

其中

pf 是实时任务的频率比，

p 是实时任务实际需要运行的时间，

e 是实时任务的最终期限，

b 是实时任务的开始时间；

### 3) 总实时任务的频率比计算：

当新的实时任务加入到任务运行队列，系统将计算新实时任务的频率比计算，然后计算总任务的频率比；

总实时任务频率比的计算方法如下面的公式所示：

$$pfsum=pfsum+T.pf$$

其中 pfsum 是总实时任务的频率比，它的初始值为 0，

“=” 表示为赋值，即 pfsum 的新值等于 pfsum 原来得值加上 T.pf，

T 是新的实时任务的控制字，

pf 是新实时任务的频率比，

$T_{pf}$  表示实时任务 T 的频率比；

#### 4) 处理器频率设置：

调度系统根据总实时任务的频率比计算出一个新的频率值，当新的频率值大于现在的频率时，将处理器的频率设置成新计算出来的频率，否则不变。

本发明与背景技术相比，具有的有益的效果是：

本发明将操作系统的实时性和低功耗调度工作相结合，利用现有操作系统的实时性来保证任务的实时性要求。本发明通过将低功耗调度方法融入到实时的任务调度中，在保证认识实时性的同时，达到低功耗的目的，延长系统电池的使用时间。

(1)实时性：应用程序在系统中运行时，调度算法按照时间片轮换的方法进行调度，保证了系统的实时性。

(2)稳定性：操作系统将动态调整系统状态的权利掌握在自己手中，而不是下放给应用程序，这样系统就能在兼顾全局的情况下动态调整系统的状态，保证系统的稳定。

(3)实用性：利用低功耗实时任务调度的简化方法，可以尽量少的修改系统的代码，达到节能目的。在经过反复的验证，这个方法可以很好的减少程序员的工作量，十分实用。

### 附图说明

图 1 是 pf 值原理图；

图 2 是整个调度系统工作的流程图。

### 具体实施方式

在实施嵌入式系统低功耗实时任务调度的简化方法时，操作系统在调度过程中，考虑了实时和节能两个约束。

嵌入式系统低功耗实时任务调度的简化方法具体实现流程如下。

#### 1) 实时任务的时间片调度：

当实时任务队列中，存在多个实时任务时，系统将通过时间片的方式对系统中的任务进行调度，以满足各个实时任务的需要。

时间片调度的方式是将处理器的运行时间划分等分的时间片，当任务在处理器上运行时，只能运行一个时间片长度，当时间片用完后，被强制停止执行，换下一个任务执行。

当处理器上的任务被调度出去后，调度器选择一个新的任务，并分配时间片给这个新的任务。在这个时间片时间内，这个任务将占有处理器的资源，直

到时间片用完或者任务完成。当任务的时间片用完后，任务将被暂时剥夺使用处理器的权利，这个任务将被放到运行队列的末尾，等待下一次被调度到处理器上运行。同时，另外一个任务将被调度到处理器上执行。时间片的轮换保证了所有任务都有运行的机会。

### 2) 新实时任务的频率比计算：

在实时系统中，一个实时任务它有以下几个参数是确定：任务的开始时间、实时任务的最终期限、以及任务的运行时间；

频率比就是将上面的三个参数进行融合所产生的，频率比的计算方法如下面的公式所示：

$$pf = p / (e - b)$$

其中

pf 是实时任务的频率比，

p 是实时任务实际需要运行的时间，

e 是实时任务的最终期限，

b 是实时任务的开始时间；

要保证任务的实时性，系统必须得保证任务在实时任务的最终期限前完成所有工作，通过确保系统在实时任务的最终期限前完成任务的执行来保证系统的实时性。任务实际需要运行的时间说明完成任务需要执行的时间数。

计算 pf 的值，可以帮助系统在确保实时性的前提下，减少工作的计算量和计算复杂度。pf 融合了这三个参数，并且 pf 也是有实际意义的，一个进程只要声称好自己的 pf 参数，就可以说明自己的实时性的要求。比如 pf 为 0.1，那么就说明在如果从 b 到 e 这个进程一直在运行，处理器的频率只要设置在最高频率的 1/10 之一，这个任务就可以在期限前完成。

之所以可以用 pf 的值来确定处理器的频率，是因为 pf 表示一个任务从开始到结束如果一直占用处理器，处理器所需要的频率，那么如果是多个任务把它们的 pf 的值加起来，就是这些任务所需要的处理器频率，虽然系统以分时间片的方式来分配时间，但是总的效果是一样的，如图 1 所示，其中图 1a 是实时任务 T1 的频率比，图 1b 是实时任务 T2 的频率比，图 1c 是实时任务 T1+T2 的总频率比。

### 3) 总实时任务的频率比计算：

计算单个任务 pf 值的主要目的是为计算整个系统的总实时任务频率比。当新的实时任务加入到任务运行队列，系统将计算新实时任务的频率比计算，然

后计算总任务的频率比。

总实时任务频率比的计算方法如下面的公式所示：

$$pfsum = pfsum + T.pf$$

其中  $pfsum$  是总实时任务的频率比，它的初始值为 0，

“=” 表示为赋值，即  $pfsum$  的新值等于  $pfsum$  原来得值加上  $T.pf$ ，

$T$  是新的实时任务的控制字(控制字是操作系统用来描述任务的数据结构)，

$pf$  是新实时任务的频率比，

$T.pf$  表示实时任务  $T$  的频率比；

总实时任务频率比的计算是节能调度的基础，通过计算总实时任务的频率比可以计算出一个合理的处理器运行频率。

#### 4) 处理器频率设置：

调度系统根据总实时任务的频率比计算出一个新的频率值，当新的频率值大于现在的频率时，将处理器的频率设置成新计算出来的频率，否则不变。

当系统没有实时任务时，就把处理器的频率调到最低。系统维持两个参数，一个是正在运行队列里的任务的  $pf$  和  $pfsum$ ，另一个是系统当前频率。当系统有新的任务加到运行队列的时候，计算新任务的  $pf$  值。把它的  $pf$  值加到原先的  $pfsum$  里面，然后再根据新的  $pfsum$  的值确定处理器的频率，如果这个频率比现在的频率要高，那么调高处理器的频率。当系统没有实时任务的时候，就把  $pfsum$  置 0，处理器的频率调到最低。新处理器频率的计算公式为：

$$f = pfsum \times fmax$$

其中，

$f$  为新的处理器频率，

$pfsum$  为正在运行队列里的任务的  $pf$  和，

$fmax$  为处理器的最大运行频率。

调度系统的整体流程图如图 2 所示。

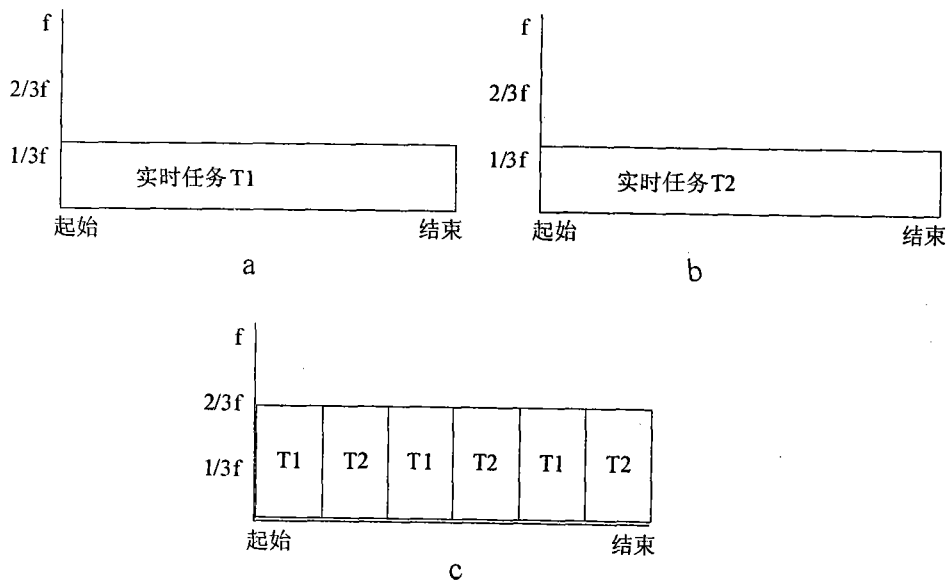


图 1



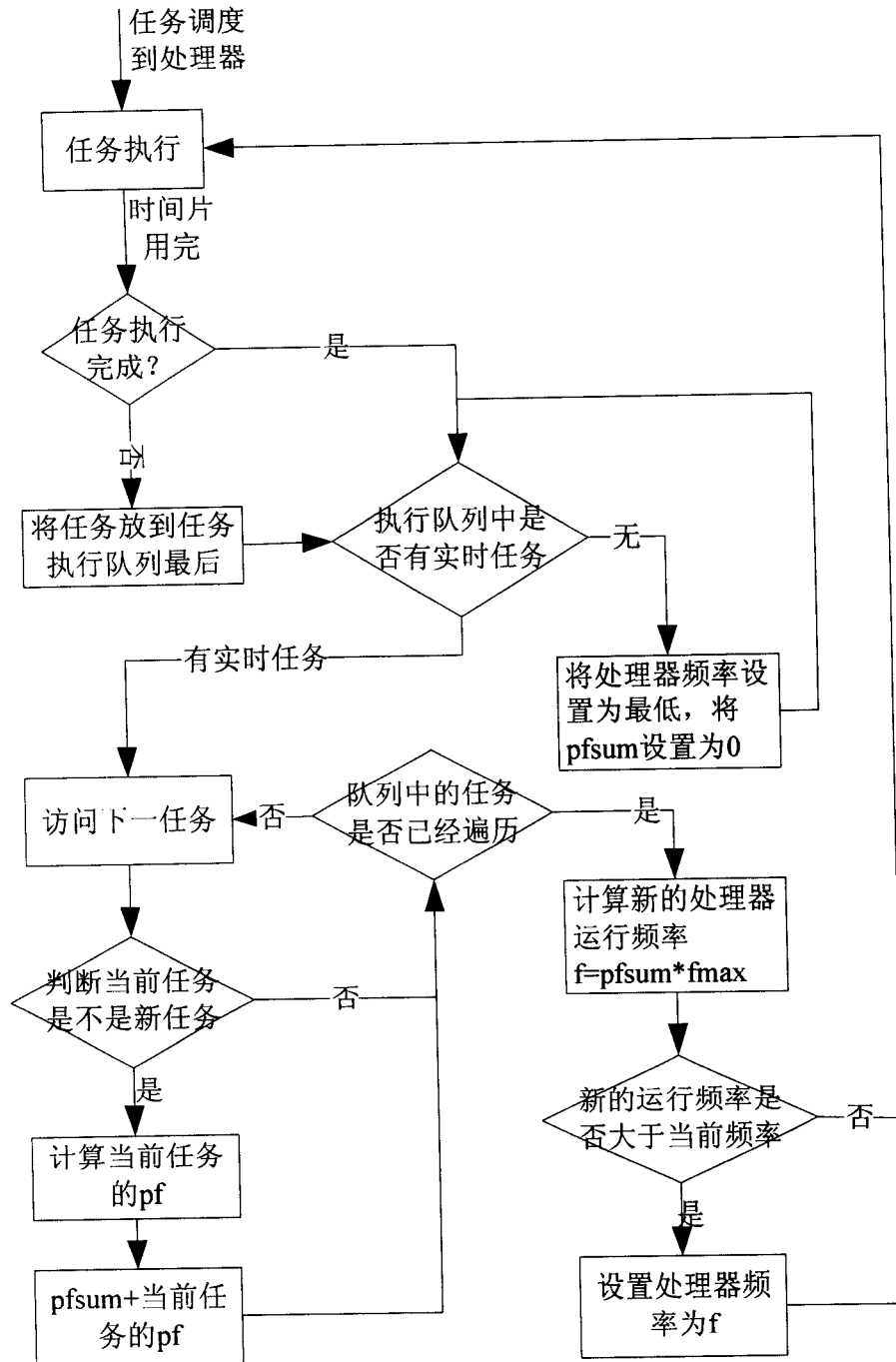


图 2