实时操作系统 RTX51Tiny 在教学中的应用

卢勇威
（广西职业技术学院　电子机械系 197#　广西 南宁　530226）

【摘　要】：本文探讨了嵌入式实时操作系统 RTX51Tiny 在教学中的应用。嵌入式实时操作系统在实际的应用中越来越重要，有必要在教学中引入相关的知识。作者选取 RTX51Tiny 作为嵌入式实时操作系统的代表，详细的谈论了它的内核结构和工作原理。同时还介绍了把 RTX51Tiny 移植到目标系统的步骤，指出移植过程中容易出现的错误和排错方法以及程序调试的方法。

【关键词】：实时操作系统　RTX51Tiny 移植

1. 引言

随着科技的不断进步与普及，嵌入式实时操作系统已经不仅仅在国防工业或者是航空航天工业的设备中使用，而在其他各行各业中也得到了广泛的应用。比如消费类电子、医疗设备、汽车、工业控制等行业中都有嵌入式系统的应用。在嵌入式实时操作系统的基础上编写的各种应用程序与通常的前后台系统程序相比，具有以下优势：功能更强大、设计周期更短、程序更容易移植与扩展。因此，嵌入式实时操作系统在程序设计中的应用已经是非常重要的技术。

由于各种原因，我系并没有开设嵌入式实时操作系统的相关课程。为了让学生掌握嵌入式操作系统的程序设计方法和思维模式，笔者决定在教学中引入操作系统的相关知识。经过反复的研究与实践，笔者发现 RTX51Tiny 与 uClinux、WinCE、uC/OS-II 等系统相比，具有结构简洁、移植过程简单、容易被初学者掌握等特点。因此笔者在教学中增加了嵌入式实时操作系统 RTX51Tiny 的相关内容。

2. RTX51Tiny 内核分析

2.1 RTX51Tiny 特性

RTX51 是一个为 8051 系列处理器量身定做的多任务实时操作系统(RTOS)。它由 KELL 公司开发，完全集成在 KEILC51 编译器中，包括 RTX51FULL 和 RTX51TINY 两个版本。

RTX51Tiny 是 RTX51FULL 的子集，内核很小(900K 左右)，可以很容易地在没有任何外部存储器的单片 8051 系统上运行。它支持 16 个任务，任务间按照时间片循环调度、支持任务间信号传递、支持以下等待操作：等待超时、等待另一个任务或等待中断的信号。在建立系统任务的同时，它还可以使用 8051 单片机中除定时器 0 之外的其它中断。

下面分别从调度器(scheduler)、对象(object)、服务(service)几个方面对 RTX51Tiny 的内核进行分析。

2.2 调度器(scheduler)

2.2.1 调度算法

调度器是内核的核心，它提供决定何时必须执行哪个任务的算法。RTX51Tiny 支持时间片轮转调度。时间片轮转调度为每个任务提供等份额的 CPU 时间。RTX51Tiny 使用 8051 单片机内部定时器 T0 来产生定时节拍(clock tick)，任务只在各自分配的节拍数(时间片)内执行。即每一个任务允许执行一个预定的时间，然后内核切换到另一个准备运行的任务并且允许这个任务执行片刻。由于时间片非常短，通常为几个毫秒，因此各任务宏观上看来是同时执行的。

RTX51Tiny 调度规则如下：

如果出现以下情况，当前运行任务中断：

① 任务调用 os_wait 函数并且指定事件没有发生。

② 任务运行时间超过定义的时间片轮转超时时间。

如果出现以下情况，则开始另一任务：

① 它任务正在运行。

② 将要开始的任务处于"READY"或"TIMEOUT"状态。

2.2.2 上下文切换

当调度器从一个任务切换到另一个任务时，发生上下文切换。每一个任务都有自己的上下文，它是每次被调度运行时所要求 CPU 寄存器的状态。

每次新任务被创立时，内核也创立一个和维护认为相关的任务控制块 TCB(Task Control Block)。TCB 主要包括三项内容：

TRY[task_id]: task_id 任务的代码入口地址，位于 CODE 空间，2 字节为一个单位。

P[taskid]: taskid 任务所使用堆栈栈底位置，位于 IDATA 空间，1 字节为一个单位。

TE[taskid].time 和 STATE[tasked].state: 前者表示任务的定时节折计数器，在每一次定时节拍中断后都自减一次；后者表示任务状态寄存器，用其各个位来表示任务所处的状态。位于 IDATA 空间，以 2 字节为一单位。

当任务运行时，其上下文是高度动态的。任务不运行时，在 TCB 中冻结上下文，以确保作为下一次运行时使用。

2.3 对象(object)

内核对象是特殊的构件，是嵌入式系统的应用开发的建造模块。RTX51Tiny 的对象包括：任务(task)、信号(semaphore)。

2.3.1 任务

用户任务是用户建立的能够完成某中特定功能的独立程序段。RTX51Tiny 的用户任务具有以下几个状态。

状态 说明

状态	说明
RUNNING	任务处于运行中。同一时间只有一个任务可以处于“RUNNING”状态。
READY	任务正在等待运行，在当前运行的任务时间片完成之后，RTX51TINY 运行下一个处于“READY”状态的任务。
WAITING	任务等待一个事件，如果所等待的事件发生的话，任务进入“READY”状态。
DELETED	任务不处于执行队列。
TIMEOUT	任务由于时间片用完而处于“TIMEOUT”状态，并等待再次运行。该状态与“READY”状态相似，但由于是内部操作过程使一个循环任务被切换而被观以标记。

表 1 RTX51Tiny 任务的状态

RTX51Tiny 最多支持 16 个任务，每个任务都是一个无限的循环，每个任务某个时刻只能处在以上 5 个状态中的某一个状态。

2.3.2 信号

RTX51Tiny 中用户可以使用 os_wait 功能暂停一个任务并等待从另一个任务发出的信号。这主要用来协调两个或更多的任务。

SIGNAL：用于任务之间通信的位，可以用系统函数置位或清除。如果一个任务调用 os_wait 函数等待 SIGNAL 而 SIGNAL 标志是 '0'，则该任务将一直处于挂起状态。如果标志是 '1'，当任务查询信号时，信号标志会被清除，并继续执行任务。

RTX51Tiny 内核除了使用 SIGNAL 之外，还使用用以下事件进行任务间的通信和同步。

基金项目：广西教育厅科研项目资助[2006]4 号

TIMEOUT: 由 os_wait 函数开始的时间延迟。其持续时间可由定时节拍数确定。使用 TIMEOUT 值调用 os_wait 函数的任务将被挂起，直到延时结束时才返回到 READY 状态并可被再次执行。

INTERVAL: 由 os_wait 函数开始的间隔延迟，其间隔时间为指定的时钟报时信号。与 TIMEOUT 的区别是，任务的节拍计数器没有复位。

2.4 服务(service)

RTX51Tiny 内核为用户提供堆栈管理的服务。内核把对存储器的管理简化为堆栈管理，它为每个任务都保留一个独立的堆栈区，全部堆栈管理都在 IDATA 空间进行。为了让当前正在运行的任务获得尽可能大的堆栈空间，其他未运行的所有任务的堆栈空间将被移走。如下图所示：

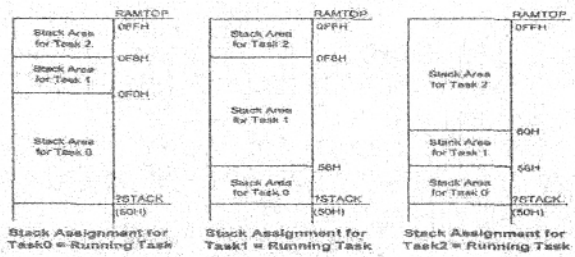

图 1、RTX51Tiny 堆栈的管理

由图可知，RTX51Tiny 总是将全部的空闲内存分配为正在运行的任务的堆栈区。当堆栈自由空间小于 FREESTACK(默认为 20)字节时，就会调用宏 STACK_ERROR，进行堆栈出错处理。

3. RTX51Tiny 的移植

RTX51Tiny 可以在没有任何外部数据存储器的 8051 单片机系统上运行。要把 RTX51Tiny 移植到目标系统上，按照以下步骤操作就可以了。

3.1 了解 RTX51Tiny 提供给用户的函数极其功能

只有了解 RTX51Tiny 提供给用户的每一个函数的功能，才能够正确的使用这些函数来编写用户程序。RTX51Tiny 提供给用户的函数如下图示：

系统函数名称	功能说明
os_create_task(task_id)	添加一任务到执行队列
os_delete_task(task_id)	删除一个从执行队列中的任务
os_send_signal(task_id)	发送信号给一个任务
os_clear_signal(task_id)	删除一个发送的信号
isr_send_signal(task_id)	在中断中发送信号给一个任务
os_wait()	等待
os_wait1()	等待
os_wait2()	等待

表 2 RTX51Tiny 的用户函数

3.2 修改配置文件

根据目标系统的具体情况，修改 RTX51Tiny 的配置文件(conf_tny.a51)中相应的参数。这个配置文件定义了许多可以修改为适合具体的应用程序的常数，见下表：

变量	说明
INT_REGBANK	指示哪些寄存器组将用于 RTX51Tiny 的系统中断
INT_CLOCLK	定义系统时间间隔。定义数字确定了每一系统中断的 CPU 周期数。
TIMESHARING	定义时间片轮转任务切换的超时时间。如果为 0
RAMTOP	定义 8051 的派生产品内存储单元的最大值。8051 为 7FH，8052 为 0FFH。
FREE_STACK	按字节定义自由堆栈区的大小。当切换任务时，RTX51Tiny 检验堆栈区指定数量的有效字节。如果堆栈太小，RTX51Tiny 将激活 STACK_ERROR 宏。
STACK_ERROR	RTX51Tiny 检测到一个堆栈出错时运行的宏。

表 3 RTX51Tiny 的配置文件参数

3.3 建立源程序

打开 Keil uVision2 建立工程和源文件，并在源文件中包含 RTX51Tiny 的头文件 rtx51tny.h。然后编写用户程序，并编译连接。这时编译器会提示出错：

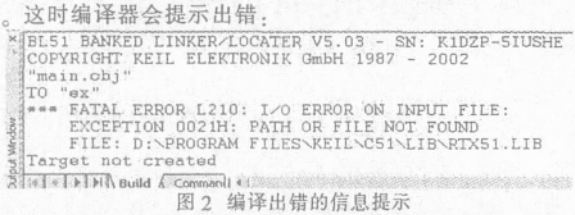

图 2 编译出错的信息提示

避免这样的错误的方法：打开目标属性窗口如下图，在操作系统选项中选择"RTX-51 Tiny"。再次编译就可以通过了。

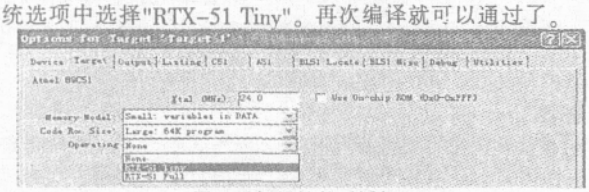

图 3 目标属性的对话框

3.4 调试程序

源程序编译通过后，就可以进行调试了。使用了 RTX51Tiny 后调试的方法与前后台系统程序的调试方法不同。在进入调试状态后，点击"Peripherals"菜单下的"Rtx-tiny Tasklist"命令可以调出下面窗口：

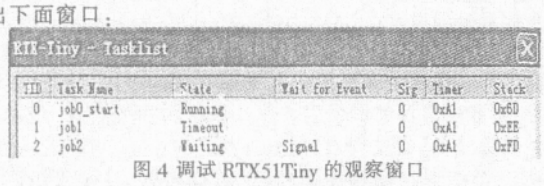

图 4 调试 RTX51Tiny 的观察窗口

从上图可以得到以下信息："TID"：任务号；"Task Name"：任务的名称；"State"：指明任务函数的状态；"Wait for Event"：指明任务正在正在等待哪些事件；"Sig"：指明信号标志位的状态；"Timer"：指明 timeout 所需的报时信号数量；"Stack"：指明内部存储器中各任务的堆栈起始地址；

利用这些信息就很容易了解到每个任务所处的状态以及它等待的事件，给程序的调试带来很大的方便。

4. 结语

实践表明，引入 RTX51Tiny 后，程序设计更加简单高效。由于 RTX51Tiny 比较简单，能够在较短的时间内被学生理解与掌握。通过对 RTX51Tiny 的学习，学生理解了嵌入式操作系统的程序设计理念和思维方法，为他们进一步学习更复杂的嵌入式操作系统知识打下基础。因此在教学中把 RTX51Tiny 作为嵌入式操作系统的入门课程是正确的选择。

参考文献：

1. [美]Qing Li.王安生译.嵌入式系统的实时概念[M].北京:北京航空航天大学出版社,2004.47-57.
2. Keil Corp.RTX-51 user's guide.http:www.keil.com
3. 马忠梅,刘滨等.单片机 C 语言 Windows 环境编程宝典[M].北京:北京航空航天大学出版社,2003.534-553.
4. 张迎新.单片机原理与接口技术[M].北京:机械工业出版社,2003.

(上接第 63 页)

系统观；进一步促进了学习观念的转变、为信息人才的培养、信息环境的创设提供信息环境支持，促进我国教育信息化的深化发展。

参考文献：

1. 余胜泉，《当代教育资源建设的问题及其对策》，中国计算机报 2004.4.1
2. 庄秀丽 孙波，《面向服务分布式资源网管理平台(ERNet)系统解决方案》 电化教育资讯，2002.11
3. 吴小红 张剑平，《中国的远程教育网站及其分析》，开放教育研究，2001.2
4. 李 芃，《E-Learning 应用中的网络创新技术》，中国电化教育，2003.7
5. 庄秀丽 何克抗，《基础教育网络资源建设管理研究》，中国电化教育，2002.10
6. Wikipedia 电子百科全书，http://eu.wikipedia.org./wiki/Main
7. 网络新潮文化之"维客"不完全手册，http://tech.tom.com