

RTX51tiny实时操作系统在单片机 STC12C5A60S2上的移植

余向阳

(530005 南宁市第一职业技术学校 广西 南宁)

【摘要】 RTX51tiny实时操作系统是Keil C51单片机开发集成环境自带的一个实时操作系统，此操作系统功能强大，简单易用，本文讲解把它移植到单片机STC12C5A60S2上的方法。

【关键词】 RTX51tiny 实时操作系统 单片机 STC12C5A60S2

RTX51tiny实时操作系统是Keil C51单片机集成开发环境自带的一个实时操作系统，它功能强大、简单易用。它可以应用在所有8051衍生的单片机系列中。STC12C5A60S2单片机是深圳宏晶科技有限公司旗下的功能强大的1T单片机。此单片机是新一代增强型8051单片机，运行的速度是传统8051单片机的8~12倍，这是相当快的速度。它内部资源丰富，兼容性强，并且内部ROM可达60KB，对于一般的应用已经足够。

在一些比较复杂的应用中，如果使用超级循环进行编程，则在实时性、逻辑性、资源的共享等方面存在较为复杂的关联性，这对编程来说是不利的。另外在增加与删除功能上也会增加编程的复杂性。能不能把RTX51tiny操作系统移植到STC12C5A60S2单片机上呢？如果可以则在保证实时性的基础上，编程的复杂性会大幅降低，程序可维护性也会大幅提升。增加或者删除功能会变得相当容易。

答案是肯定的。RTX51tiny操作系统完全可以移植到STC12C5A60S2单片机上。

先来看看RTX51tiny操作系统介绍。

一、RTX51tiny操作系统的特性

RTX51tiny操作系统是集成在Keil C51开发环境中的实时系统。它的主要特点如下：

(一) 采用时间片轮转调度策略进行任务切换，不支持抢占式任务调度策略。

(二) 允许最大16个任务循环切换。

(三) 支持信号传递，不支持消息处理、队列处理、邮箱等功能。

(四) 内核很小，最大仅占用900字节。

(五) 系统函数仅有13个，非常容易使用。

(六) 支持硬件中断功能。

根据它的特点，我们可以得到这样的结论：RTX51tiny操作系统非常适合应用于8051系列单片机，可以实现多任务协同工作。在实时性上，可以采用硬件中断响应，速度与没有使用操作系统一样快捷。

二、RTX51tiny工作原理

RTX51 Tiny 用标准8051的定时器0(模式1)生产周期性的中断。该中断就是RTX51 Tiny的定时节拍(Timer Tick)。库函数中的超时和时间间隔就是基于该定时节拍来测量。

RTX51tiny通过时间片轮询的办法对任务进行切换。时间片通常设置很短只有几个到几十个毫秒。每一个任务都有自己的时间片，当自己的时间片用完或者调用了os_wait()函数与os_switch_task()函数则任务进行切换。只要这个时间片足够短，则多个任务之间在宏观上可以看成“并行”执行，而微观上则是轮流执行。

多个任务可以“并行”执行，这在单片机处理多个事件时，有很多的好处，因为我们在使用单片机时经常会遇到需要同时处理的事件。

三、STC12C5A60S2单片机介绍

STC12C5A60S2功能强大，它的主要特点如下：

(一) 具有较大的ROM，可以达到60KB。

(二) 不仅与传统8051单片机一样拥有256字节的内部RAM，而且还在内部扩展了1024字节的RAM。

(三) 通过串口下载程序，程序下载次数可以达10万次以上。省去了仿真器，程序调试相当方便。

(四) 兼容传统8051指令集。

(五) 运行速度是传统8051的8~12倍。

(六) 片内拥有1K字节的EEPROM，可以在程序运行时读写，如须保存一些永久数据则可以使用它。

(七) 具有两个串口。

以上是此单片机的主要特点。这些特点显示它是一款功能十分强大的单片机，在较为复杂的应用中也能够胜任。

四、RTX51tiny针对单片机STC12C5A60S2的设置

在Keil C51这个软件的安装目录：Keil/C51/RtxTiny2/SourceCode/下有一个文件Conf_tny.A51，此文件是RTX51tiny操作系统的设置文件。可以通过记事本打开它，也可以通过Keil C51软件打开它。它里面的设置是用汇编指令编写的。

RTX51tiny需要设置的内容如下：

(一) 指定定时节拍中断寄存器组

默认状态：INT_REGBANK EQU 1

寄存器组默认设置为0。对STC12C5A60S2单片机来说，它有四组通用寄存器，在没有特殊要求的情况下，使用哪一组都行，使用默认值即可。

(二) 指定定时节拍长度

默认状态：INT_CLOCK EQU 10000

RTX51tiny为了产生系统需要的定时节拍，固定使用传统8051单片机的定时中断T0，用它产生定时节拍。默认状态下，一个定时节拍的长度为10000个机器周期。对于传统8051单片机来说，如果采用12MHz晶振，则一个机器周期是1μs，10000个机器周期则是10ms，所以定时节拍时长为10ms。STC12C5A60S2单片机它比传统的8051单片机快8~12倍。在定时时长方面则是12倍(AUXR特殊功能寄存器的T0x12位置1)。由此它的机器周期是1/12μs，则定时节拍为833.3μs。为了得到整数，我们可以设定INT_CLOCK为12000、24000、36000、48000、60000等，对应的定时节拍时长为：1ms、2ms、3ms、4ms、5ms。这里的定时节拍可以作为os_wait()函数等待的一种事件。如定时节拍数是整数，则使用os_wait()函数来延时更方便计算。

(三) 指定任务运行时间片长短

默认状态：TIMESHARING EQU 5

时间片是任务不间断运行的最长时间，默认为5，即为5个定时节拍。如果设定定时节拍为1ms，则时间片为5ms，以些类推。此默认值可不改变。为了使任务“并行”执行，时间片不能太短，太短会使单片机花费在任务切换上的时间变多，而实际任务执行的时间变短。时间片不能太长，太长则会使任务切换不够快，并行执行的效果差，严重时会使快速

的信号得不到处理。这时我们应该根据任务的数量来确定此值,比如任务的数量比较少,可以把此值设得大一些,任务数量多,可以把此值设定得小一些。总之使时间片处在几个到几十个毫秒之间即可。

(四) 是否执行长中断

默认状态: LONG_USR_INTR EQU 0

不执行长中断。如果为1则执行长中断。执行长中断是什么意思呢?

RTX51tiny,在系统执行过程中,有两种中断:一种是单片机的硬件中断。一种是任务切换造成的中断。此处指硬件中断。在RTX51tiny运行时,希望硬件中断发生时,执行中断服务函数非常快,不影响到任务切换,一旦影响到任务切换就会影响到程序的实时性。但有时,中断服务函数要处理的件事需花费较长的时间,此时就会影响到任务的切换。为了让系统支持长中断,可以设置此参数为1。一般情况下,不使用长中断,即使有件事要花较长的时间处理,也不使用长中断。可以把中断服务作为一个任务来处理。发生中断时,可以发信号给此任务,让它变成就绪态,然后运行。这样就可以保证其他任务的实时性。

(五) 在定时节拍中断中,加入自己的代码

```
默认状态: HW_TIMER_CODE MACRO
; Empty Macro by default
RETI
ENDM
```

默认状态是不加任何代码。一般情况下,我们可以不加代码,如果我们想得到一个与定时节拍相关的信号则可以在空处加入如下代码:

```
CPL P10 ;此指令功能是对P1.0取反。
```

此时可以从P1.0引脚输出一个频率是定时节拍频率两倍的信号。

为了实现这个功能,我们还得在Conf_tny.A51文件中,加入P10的定义。可以在地址定义处加入以下两行汇编代码:

```
P1 DATA 0B0H
P10 BIT 0B7H
```

(六) 是否支持代码分页

默认状态: CODE_BANKING EQU 0

不支持代码分页。如果为1,则支持代码分页。什么是代码分页呢?

对于8位的单片机,它的寻址范围只有64KB(2¹⁶),如果单片机的程序编译以后得到的16进制文件超过了64KB,则需要通过分页的方式进行访问,否则超过的部分无法访问。这就是代码分页。而我们使用的STC12C5A60S2单片机的ROM只有60KB,没有达到8位单片机寻址的极限,不需要代码分页。这个参数保持默认值。

(七) 设置内部RAM的最大值

默认状态: RAMTOP EQU 0FFH

默认RAM的最大值为255。这与传统的单片机内部的RAM结构一致。而STC12C5A60S2单片机内部扩展了1024字节的内部扩展RAM。它的实际内部RAM地址为0x000~0x3ff。那么在这里我们应该把这个参数设置为什么值呢?

在STC12C5A60S2单片机的使用手册中,有这样的表达:

当把寄存器AUXR中的EXTRAM设置为0时,就可以通过MOVX @DPTR和MOVX @Ri两条语句来访问内部扩展的RAM。MOVX @DPTR可以访问0x00~0x3ff地址的内部扩展RAM,而超过这个范围的则都是访问外部RAM。MOVX @Ri语句只能访问0x00~0xff的地址。与传统8051单片机一样。

根据这样的表述我们是不是就可以把这个参数设置为3FFH呢?

我们再来看看RTX51tiny中的设置文件Conf_tny.A51中对RAMTOP这个参数的操作,实际上都是把RAMTOP当成一个字节

来操作,如果我们把RAMTOP设置为3FFH,超出了一个字节的范围,在操作时,还是当成一个字节。由此我们得出结论,RAMTOP这个参数还是使用默认值。那么STC12C5A60S2单片机内部扩展的1024字节内部RAM如何使用呢?在使用手册中,有明确的表示:在变量前加上xdata关键字即可。

(八) 设置发生任务切换时,最小的堆栈需求

默认状态: FREE_STACK EQU 20

在默认状态下,这个参数设置为20,即最小需要20个字节的RAM存储空间。在这里设置为默认值。

(九) 在发生堆栈错误时运行的宏

在任务切换时,单片机内部RAM没有达到最小20字节的要求,则发生堆栈错误。这是一个循环。代码如下:

```
STACK_ERROR MACRO
CLR EA ; disable interrupts
SJMP $ ; endless loop if stack space
is exhausted
ENDM
```

首先把总中断允许位清零,不再允许任务切换,然后进入一个死循环,不再执行任务操作。

10. 设置是否允许单片机进入空闲模式

```
CPU_IDLE_CODE EQU 0 ; 0 CPU_IDLE MACRO
is not inserted
; 1 CPU_IDLE MACRO
is executed
```

默认状态: 不插入CPU进入空闲模式的宏。

有很多的单片机有这个功能,STC12C5A60S2也有这个功能,我们可以使能这段代码。这样在没有其他任务运行时,单片机会进入到空闲模式。在空闲模式下,只有CPU不工作,其他部分仍然在工作。这主要是从节能方面考虑,如没有这项考虑可以不插入这段宏。

```
CPU_IDLE MACRO
ORL PCON,#1; set 8051 CPU to IDLE
ENDM
```

以上是可以插入的宏。

到此,针对STC12C5A60S2单片机的设置已经完成。

五、使用RTX51tiny的步骤

(一) 在C语言文件的开头,包含头文件“RTX51TNY.H”。如下: #include <rtx51tny.h>

(二) 把Conf_tny.A51文件拷贝到自己的工程中,并添加到工程中。然后进行相应的设置。

(三) 编写自己的程序。

六、使用RTX51tiny注意事项

(一) 确认把头文件RTX51TNY.H包含到文件中来。

(二) 不需要编写main函数,RTX51tiny有自己的main函数。

(三) 至少建立一个任务函数。这个任务就是任务0。

(四) 至少调用一个RTX51tiny系统函数。否则,连接器不会包含RTX51tiny库。

(五) 在任务0中建立其他任务。

(六) 任务函数不能有形参与返回值。并且任务内要采用死循环的方法把要执行的程序包起来。

经过以上的设置后,RTX51tiny可以很方便的移植到STC12C5A60S2上。

七、结束语

本文论述了把RTX51tiny操作系统移植到单片机STC12C5A60S2上的方法。这种方法具有广泛的应用性。但限于作者的水平,文中难免有错误的地方,希望得到谅解。

参考文献:

- [1] 《RTX51tiny帮助文件》2004年1月CHM格式的初始版本。
- [2] 《STC12C5A60S2.PDF》2009年11月电子版。