

基于实时操作系统的吊舱管理系统

侯卫国, 林青, 张建秋, 王勇

(复旦大学电子工程系, 上海 200433)

摘要:介绍了基于嵌入式实时操作系统的飞机吊舱管理系统的设计方法。该方法采用模块化设计思想, 将整个系统功能分解为几个简单的易于实现的任务, 极大地提高了系统开发的效率。给出了系统设计的关键思想。实际结果验证了该设计方法的有效性。

关键词:吊舱; RTX51 TINY; 单片机

LOS Management System Based on RTOS

HOU Weiguo, LIN Qing, ZHANG Jianqiu, WANG Yong

(Dept. of Electronic Engineering, Fudan University, Shanghai 200433)

【Abstract】Based on real time operating system (RTOS), a new design method for the LOS management system is introduced. The whole system is decomposed into several simple subsystems so that the design efficiency is greatly improved. The key idea of the system design for the LOS management is also discussed. Experiment results confirm the effectiveness of the proposed design method.

【Key words】Line of sight (LOS); RTX51 TINY; MCU

飞机吊舱在飞机完成各项功能时起着关键的作用, 现代飞机吊舱功能越来越复杂, 对吊舱管理系统的设计提出了越来越高的要求。现代飞机吊舱管理系统需要管理吊舱系统的供配电、电源状态监测、与图像跟踪系统的通信、可见光和红外摄像机的命令控制、手动摇杆的命令解析以及吊舱和飞机系统的通信接口等任务。如何实现这些功能并实现实时性和安全性是现代飞机吊舱管理系统要解决的首要课题。

传统的吊舱管理系统设计采用单片机 (MCU) + 单任务循环结构的软件机制进行, 需要很长的开发周期。设计的系统复杂, 整个软件架构显得烦杂, 难于管理和维护。由于采用的是单任务顺序循环的软件结构, 系统的实时性无法得到保证。

为了解决传统吊舱管理系统设计方法的缺点, 本文引入 MCU + 实时操作系统 RTX51 TINY 的开发方法。采用实时多任务操作系统, 可以简化软件设计, 提高软件的模块化程度, 增加软件的可维护性, 显著地缩短项目周期。

1 RTX51 TINY 简介

RTX51 TINY 是德国 Keil 公司开发的一种应用于 MCS51 系列单片机功能强大的实时操作系统。RTX51 TINY 是一个很小的内核, 完全集成在 KEIL C51 编译器中。更重要的是, 它仅占用 900B 左右的程序存储空间, 可以在没有外放数据存储器的 8051 系统中运行。RTX51 TINY 采用循环任务切换, 并且支持信号传递, 还能并行地利用中断功能。

RTX51 TINY 采用 MCU 内部定时器 T0 来产生定时节拍, 各任务只在各自分配的定时节拍数(时间片)内执行。当时间片用完后, 切换至下一任务运行, 因此各任务是并发执行的。

中断处理采用的是 C51 的中断函数。当中断发生的时候, 程序就跳到了相应的中断函数, 它和正在运行的任务是相互独立的, 中断的处理是在 RTX51 TINY 系统之外, 和任务切换规则没有关联。

2 系统硬件结构

MCU 采用 Silicon 公司的混合信号单片机 C8051F021, 高速流水线结构的 8051 兼容的 CIP-51 内核 (可达 25MIPS), 片内全速非侵入式的在 JTAG 系统调试接口, 和丰富的片上外设。8 通道 12 位 ADC, 2 个 12 位 DAC, 硬件实现的 SPI、I2C 和 2 个 UART 串口, 5 个通用 16 位定时器, 片内看门狗定时器, VDD 监视器, 64KB 可在系统编程的 FLASH 存储器, 4MB 的片内 RAM。可以只包含一些简单的外设就可以实现本系统的要求。并且 JTAG 调试更加适合多任务系统的开发调试, 可以很方便地进行在系统调试, 缩短开发时间。

图 1 中阴影部分为吊舱管理系统硬件部分, 阴影外是与吊舱管理系统相连的其他分系统。

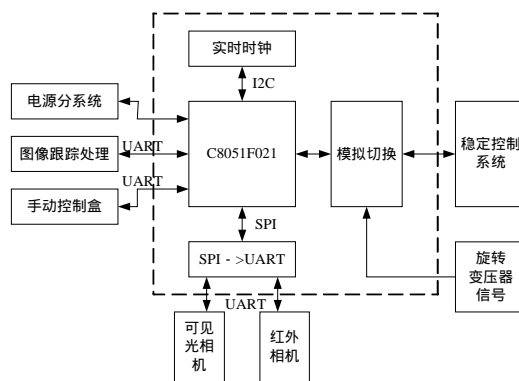


图 1 硬件系统框图

如图 1 外围电路有实时时钟电路、SPI 转 UART 电路和

作者简介:侯卫国(1979 -), 男, 硕士生, 主研方向: 多传感器数据融合; 林青, 博士生; 张建秋, 教授、博导; 王勇, 副教授

收稿日期: 2005-12-04 **E-mail:** hwguo@263.net

模拟切换电路。实时时钟模块提供吊舱系统所需的时间和日历信息,采用 I2C 总线与 MCU 相连。SPI 转 UART 模块负责扩展系统的 UART 接口满足与分系统通信需要,其用于和可见光摄像机以及红外摄像机进行通信,与 MCU 采用 SPI 接口。模拟切换模块负责切换手动摇杆控制信号、图像跟踪处理信号和旋转变压器信号的切换,以实现不同的控制模式。

3 系统功能分解

由于系统功能较复杂,要管理的功能模块较多,根据实时多任务操作系统软件模块化设计思想,分别为各个功能模块建立一个系统任务,每个任务都是一个死循环,是单独的循环顺序结构,完成简单的功能,便于编程实现。由实时操作系统进行各任务的分时调度,从而完成系统功能。

根据本系统功能的要求,划分为 5 个工作任务,分别完成不同的简单功能。具体分别是:初始化任务(0 号任务);实时时钟任务(1 号任务);手动摇杆响应任务(2 号任务);电源管理任务(3 号任务);图像跟踪响应任务(4 号任务)。

初始化任务为系统的入口点,在这里并不需要传统的主函数 main(),RTX51 TINY 直接进入初始化任务(0 号任务)。其完成系统的初始化、中断初始化及其他任务的创建。程序代码如下:

```
void initialall (void) _task_0
{
    init_periphery(); 完成系统的初始化设置
    os_create_task(1); 创建实时时钟任务
    os_create_task(2); 创建手动摇杆响应任务
    os_create_task(3); 创建电源管理任务
    os_create_task(4); 创建图像跟踪响应任务
    os_delete_task(0); 初始化任务已经完成了使命,删除初始化任务
}
```

当初始化任务创建完其他任务后,就可以将各个任务当作单独的功能来分别实现,每一个任务只处理和自己有关的功能,都是顺序死循环结构。以实时时钟任务程序代码示例如下:

```
void real_time_task (void) _task_1
{
    /* 处理实时时钟任务 */
    while(1){...}
}
```

图 2 是实时操作系统结构下设计多任务程序的结构图。

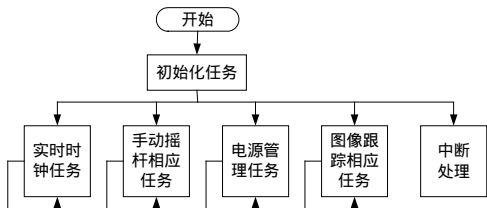


图 2 多任务下的程序结构

4 数据同步

为实现整个系统功能,必须定义一定全局变量,来实现任务间数据传递。这样各个任务都可以对这些全局变量进行访问,显然如果不加以限制的话,将引起数据冲突,甚至可能造成系统崩溃。

为解决这个问题,设定全局变量数据处理规则:每个全局变量只有一个源(即只有一个任务对其拥有修改权限),其他的任务只能对其进行读操作。这样只要在开始系统设计规

划时合理规划各个全局变量,就可以保证不会发生数据访问冲突的问题。

同时,全局变量在任务间也需要同步处理机制,以此来协调整个系统的数据处理。这里借鉴 Windows 消息驱动的思想,使部分不需要始终执行的任务处于等待信号状态,这样它们不会占用资源。只有当需要进行相应处理时通过需要它们运行的任务发送给它们 SIGNAL,它们才会进入 RUNNING 任务队列中,得到执行。通过这样的处理,可以使 MCU 在有限的资源下集中精力处理好耗时且重要的事情。例如在图像跟踪处理任务运行中,需要实时时钟任务一个时钟同步信号处理完成后才继续运行。程序代码示例如下:

图像跟踪处理任务等待实时时钟任务同步信号,任务阻塞。

```
void image_track_task (void) _task_4
{
    ...
    os_wait(K_SIG);
    ...
}
```

实时时钟任务传递同步信号给图像跟踪处理任务,激活图像跟踪处理任务。

```
void real_time_task (void) _task_1
{
    ...
    os_send_signal(4);
    ...
}
```

系统中有较多的外设,它们和 MCU 进行数据通信需要占用较多的资源,用查询方式无法实现实时响应。为此,对实时性要求高的数据比如图像跟踪响应数据、手动摇杆响应数据和实时时钟信号就要采用中断响应的方式。和任务间数据同步不同的是,在中断函数中应使用 isr_send_signal()函数向任务传递同步信号。以程序中串口 0 接收到图像跟踪分系统数据后向图像跟踪响应任务传递同步信号为例:

```
void UART0_I () interrupt 4 using 3
{
    ... 已接收到图像跟踪分系统数据
    isr_send_signal(4); 激活图像跟踪处理任务,处理相应数据。
    ...
}
```

5 系统性能

在吊舱管理系统设计中,笔者采用 RTX51 TINY 嵌入式实时操作系统,运用模块化的软件设计思想,在较短的时间内成功地研制了吊舱管理系统。该系统目前已成功地应用于实际项目中。在严格的系统测试和实际现场应用中,该系统工作性能优良,安全可靠。

6 结束语

在吊舱管理系统开发中嵌入操作系统大大提高了系统的稳定性。即使某个外设或任务出现故障,与之无关的任务一般也不会受到影响,不像传统的顺序循环机制的程序设计方法中如果某一环节出现问题会立刻造成整个系统的崩溃。

实践表明,在系统复杂、对实时性和精度要求比较严格,并且需要并行处理的任务较多的运动控制系统中,使用 RTX51 TINY 是一个较好的解决办法。此外,使用 RTX51 TINY 实时操作系统,会加快嵌入式控制系统的开发速度,降低软件编写的复杂度,提高产品的开发效率。维护和功能扩展都非常方便。

参考文献

- 1 Keil Software Inc.. RTX51 TINY User Guide[Z]. 2001.
- 2 Software Inc.. Cx51 Compiler, Keil[Z]. 2001.
- 3 Silicon Laboratories. C8051F02x Datasheet[Z]. 2003.