中图分类号: TM351 TM341 文献标识码: A 文章编号: 1001-6848(2007)05-0052-03

基于 μC/OS-Ⅱ的永磁同步电动机伺服系统软件设计

陈 涛, 彭侠夫, 叶瑰昀
(厦门大学 信息科学与技术学院自动化系, 厦门 361005)

摘 要: 对以 TMS320LF2407为基础, 如何在 μC/OS-Ⅱ实时多任务内核的环境下建立永磁同步电动机伺服系统的位置环、速度环和电流环进行了阐述; 通过运用 μC/OS-Ⅱ来调度任务, 提高了系统的稳定性、可靠性和安全性, 缩短了软件设计的周期和难度, 提高了程序的可读性和可维护性。

关键词: μC/OS-Ⅱ; 伺服系统; 软件; 永磁同步电动机; 数字信号处理器

Software Design of the PMSM Servo System Based on the μC/OS-Ⅱ

CHEN Tao, PENG Xia-fu, YE Gui-yun
(Institute of Computer and Information Engineering, Xiamen University, Xiamen 361005, China)

ABSTRACT: Using the TMS320LF2407A as the microprocessor, the paper presents how to design the software the three kinds of position, speed and current closed loop control models in the PMSM servo system which using μC/OS-Ⅱ real-time multitask kernel. Using μC/OS-Ⅱ to attemper the task, can improved stability, reliability and security, shorten the period and difficulty of the software designing, improved the readability and maintainability of procedure.

KEY WORDS: μC/OS-Ⅱ; Servo system; Software; PMSM; DSP

0 引 言

通常伺服系统软件采用前后台系统 (Foreground/Background System), 应用程序是一个无限的大循环, 事件都是顺序执行, 实时性很强的任务靠中断服务程序来执行。这种编程方法的缺点是系统的稳定性和实时性差, 编程难度大; 伺服系统对系统的响应时间要求很严, 它要同时完成采样、计算、控制、通讯等任务, 没有一个实时多任务操作系统是不行的。μC/OS-Ⅱ是一种源码公开的占先式实时多任务操作系统。μC/OS-Ⅱ除了提供最核心的占先方式的任务调度功能外, 还具有任务管理、时间管理、任务间的通讯和同步、内存管理等功能。因此把 μC/OS-Ⅱ作为伺服系统软件, 通过 μC/OS-Ⅱ来调度系统任务, 各模块之间相互独立, 杜绝了由于某个任务陷入死循环而死机。系统可以调用其他的任务来弥补进入死循环的任务。因此使得位置伺服系统的软件设计都变得非常简单, 能极大限度地降低开发难度和缩短软件开发周期。

收稿日期: 2006-05-30

1 硬件电路

本功率驱动电路采用智能功率模块 IPM, 工作频率可以达到 20 kHz, 具有过流保护和短路保护、过热和欠压锁定保护功能, 系统的可靠性和集成性提高。另外还有逻辑控制电路, 电流、速度和电压检测, 键盘和显示等部分。转子位置采样采用混合式光码盘, 其输出的 A、B信号直接进入编码器接口 QEP1和 QEP2引脚, 采用中断的方式获知转子的位置。利用时间管理器模块 (EV) 产生 PWM信号, 实现对电机的控制。在 DSP内部实现位置环、速度环和电流环的控制算法。

2 系统软件设计

2.1 任务建立

想让 μC/OS-Ⅱ管理用户的任务, 必须先建立任务。可以通过传递任务地址和其它参数给 OSTaskCreate ()。任务可以在多任务调度开始前建立,

也可以在其它任务的执行过程中被建立。在开始多任务调度（即调用 OSStart（））前，用户必须建立至少一个任务。任务不能由中断服务程序（ISR）来建立。OSTaskCreate（）需要 4个参数：task是任务代码的指针，pdata是当任务开始执行时传递给任务的参数的指针，ptos是分配给任务的堆栈的栈顶指针，prio是分配给任务的优先级。在伺服系统中，中断的优先级最高，保证中断的实时性，通用定时器 T1的周期中断启动 A/D转换。

2.2　任务调用

μC/OS-Ⅱ可以管理多达 64个任务；从中保留了 4个最高优先级和 4个最低优先级的任务供自己使用，所以用户可以使用的只有 56个任务。任务的优先级必须不同。μC/OS-Ⅱ采用基于优先级的调度法，即 CPU总是让处在就绪态的优先级最高的任务先运行。优先级的确定主要依据任务的重要性来确定。本伺服系统主要任务优先级顺序：位置环→速度环→电流环。

在 μC/OS-Ⅱ系统中，每个任务都是一个无限循环的函数。当正在运行的任务进入延时程序或在等待消息时会自动挂起，CPU运行其它的就绪任务。当某任务运行条件满足时，就进入就绪状态，操作系统依据其优先级进行调度。如果它比当前的任务优先级更高，就进行任务的切换，系统保证永远运行优先级最高的任务。在 μC/OS-Ⅱ中，每个任务的工作状态通过单独的任务控制模块（OS_TCB）来记录，包括任务地址指针、优先级、运行状态、堆栈信息等。系统中任务之间通过信号量、邮箱、消息队列来实现任务之间的通讯。μC/OS-Ⅱ中需要一个时钟资源来实现时间的延时和期满的功能，没有它就没有嵌入式系统的任务调度。在 TMS320LF2407A中用计数器的周期中断为系统提供时钟节拍；OSTimeTick（）完成任务的延时和挂起。

3　多任务伺服软件结构

软件部分是控制的核心，主要由位置捕捉、转速估算、电流和电压测量、三环控制算法实现、SVPWM算法、故障检测、通讯和显示等组成。下面主要介绍三环算法结构以及整体软件结构。

3.1　位置环实现

通过 DSP内部的编码信号处理电路对输入信号 4倍频，利用通用定时器根据转向计数，每个 PWM周期对编码盘采样一次，每两次 PWM周期的采样脉冲之差即脉冲增量，也就是转子的机械转角增量，对增量累计就可以得到转子的绝对位置。本系统取 10个速度环周期对位置采样计算一次。位置环程序结构如下。

```
Void PositionTask (void *pdata) //
{
  pdata = pdata;
  For (;;)
{
  OSSemPend (SpeedLoop, 0, err);
  发 AD转换信号，采样给定位置信号;
  读取转子脉冲增量并计算绝对机械位置;
  由位置环控制算法计算给定参考速度大小;
  }
}
OSTaskCreate (PositionTask, (void *) PositionP, (void *) PositionStk[0], 0); //创建位置环任务。
```

3.2　速度环实现

由于惯性较大，机械时间常数远大于电时间常数，因此速度采样不是在每个 PWM周期内进行，取 20个 PWM周期对速度采样计算一次。通过对 20个 PWM周期内脉冲的增量乘以 Kspeed计算得到，给定转速与转速反馈量的偏差经过速度 PI调节器，输出用于转矩控制的电流 q轴分量。速度环程序结构如下。

```
Void SpeedTask (void *pdata) //速度环任务
{
  pdata = pdata;
  For (;;)
{
  OSSemPend (CurrentLoop, 0, err);
  获取脉冲增量并计算实际转速;
  由速度环控制算法计算给定电流参考量;
  If (速度环次数 + + >10)
  OSSemPost (PositionLoop);
  }
}
OSTaskCreate (SpeedTask, (void *) SpeedP, (void *) SpeedStk[0], 1); //创建速度环任务。
```

3.3 电流环实现

电流环的调节周期一般很短，这里选择 PWM 波形的开关频率为 15 kHz，即定时器的周期为 66.67 μs。通用定时器 T1的周期中断产生时进行电流环计算。电流参考值与电流反馈量的偏差经过电流 PI调节器分别输出旋转电压矢量，利用 SVPWM技术产生 PWM 信号控制逆变器，改变定时器的比较寄存器的值，从而改变 PWM 的占空比。电流环程序结构如下。

```
Void CurrentTask (void * Pdata) //电流环任务
{
  Pdata = Pdata;
  For (;;)
{
  OSSemPend (CurrentLoop, 0, err);
  启动 AD采样，采样实际电流信号;
  由电流环控制算法计算占空比的大小;
  If (电流环次数 + + >20)
  OSSemPost (SpeedLoop);
}
}
OSTaskCreate (CurrentTask, (void *) CurrentP, (void *) CurrentStk [0], 2); //创建电流环任务。
```

3.4 程序中断结构

这里的中断形式和前后台系统一样，仅仅是在原来用户的中断程序里面加上函数 OSIntEnter () 和 OSIntExit ().

```
ISR
{
保存处理器寄存器的值;
OSIntEnter ();
执行中断函数;
OSIntExit ();
恢复处理器寄存器的值;
中断返回;
}
```

3.5 系统 main函数结构框架

在主函数中用 OSTaskCreate () 函数建立初始化任务 TaskInit ()。初始化任务中建立一系列的信号量和邮箱：唤醒 AD任务，唤醒液晶显示任务，唤醒键盘任务，唤醒时钟中断任务。建立位置环任务、速度环任务、电流环任务等。在 μC/OS-II实时内核下整个程序的结构框架如下。

```
Void main (void) //主函数
{
硬件初始化;
OSInit ();          //内核的初始化
调用 OSTaskCreate () 创建初始化任务 TaskInit ();
OSStart ();         //开始多任务的调度
}
Void TaskInit (void * data)
{
硬件时钟初始化;
创建用户任务;
定时检查系统的状态;
定时复位看门狗;
}
```

4 结 论

在 μC/OS-II的基础上可以实现 A/D 转换、RS-232通信、数据处理、键盘和显示等任务；只需把要完成的工作分成单个任务，编写独立的任务块然后向系统添加即可，需要处理的任务越多，μC/OS-II的优越性越显著。利用 μC/OS-II可以提高系统的可靠性和实时性，缩短软件设计的周期和难度，程序的可读性和可维护性提高。采用 μC/OS-II的伺服控制系统较传统伺服控制系统在程序设计的模块化和快速性等方面都具有明显的优势，取得了更好的控制效果和更好的系统综合性能。

参考文献

[1] JEAN J. LABROSSE. 邵贝贝译. μC/OS-II源码公开的实时嵌入式操作系统 [M]. 北京：中国电力出版社，2001.

[2] 刘卫国. μC/OS-II在无刷直流电机位置伺服系统中的应用，微电机，2004，38 (5)：57-59.

[3] 张天水. 基于嵌入式操作系统 μC/OS-II和 dsp的伺服电机多任务控制 [J]. 冶金自动化，2004 (增刊).

作者简介：陈 涛 (1982 -)，男，河南人，硕士研究生，研究方向为控制理论与控制工程。