基于MSP430和μC/OS-Ⅱ的智能电表的研制

常辉　翁东波

关键词：MSP430单片机　μC/OS-Ⅱ
嵌入式实时操作系统
智能电表

常辉先生，合肥工业大学电气学院研究生；
翁东波先生，淮南供电公司计量管理中心工程师。

一　前言

随着我国经济的高速发展，用电需求迅速增长，电力供需矛盾日渐突出，为了合理用电，科学调配，各地纷纷实行分时计费，错峰用电。智能电表因集电能计量、复费率控制、自动抄表等多种功能于一体而得到广泛使用。

一般智能电表软件设计多采用前台后台系统，应用程序是一个无限的循环，在循环中调用相应函数完成相应操作，这部分可看成后台行为。中断服务程序处理异步事件，这部分可看成前台行为。时间相关性很强的关键操作一定是靠中断服务来保证的。因为中断服务提供的信息一直要等到后台程序走到该处理这个信息这一步时才能得到处理，最坏情况下的任务级响应时间取决于整个循环的执行时间。因为循环的执行时间不是常数，程序经过某一特定部分的准确时间也是不能确定的。进而，如果程序修改了，循环的时序也会受到影响。

实时内核也称为实时操作系统或RTOS。它的使用使得实时应用程序的设计和扩展变得容易，不需要大的改动就可增加新功能。通过将应用程序分割成若干独立任务，RTOS使应用程序的设计过程大为简化。通过RTOS提供的服务，如信号量、邮箱、队列、延时、超时等，RTOS使单片机的资源得到更好的利用。绝大多数RTOS都是可剥夺型内核。最高优先级的任务一旦就绪，总能得到CPU的控制权。当一个运行着的任务使一个比它优先级高的任务进入了就绪态，当前任务的CPU使用权就被剥夺了，高优先级的任务立刻得到了CPU的控制权。实时操作系统的使用，能简化系统的应用开发，使用可剥夺型内核，最高优先级的任务什么时候可以执行、可以得到CPU的控制权是可知的。使用可剥夺型内核使得任务级响应时间得以最优化。

MSP430系列单片机是一种具有集成度高、功能丰富、功耗低等特点的16位单片机。它的集成调试环境Embedded Workbench 提供了良好的C语言开发平台。设计中考虑到程序的复杂性和程序容量大的要求选择了MSP430F449，这款单片机的FLASH存储器高达64kB，RAM多达2kB，串行通信可软件选择UART/SPI模式，同时具有160段液晶驱动能力，可满足智能电表的设计需要。

二　基于MSP430F449单片机的硬件设计

智能电表的硬件电路主要包括电源电路、计量电路、通信电路、LCD显示、实时时钟、EEPROM等模块。系统如框图所示，其中计量电路的设计十分重要，它是电能表具备计量功能与体现计量准确性的关键部分，MSP430F449单片机是智能电表系统的核心，实现系统中各个部件时间的协调，以及复费率控制、LCD显示、RS-485串行通信、红外通信、实时时钟等一系列重要的功能。

智能电表采用复费率计费方式，计量芯片采用美国模拟器件公司(ADI)的电力计量产品ADE7755。该芯片是一种高精度电能测量集成电路，其技术指标超过了IEC1036规定的准确度要求。电表接入线路后，通过分流器、分压电路分别对电流和电压信号进行采样，电流通道和电压信道的信号经放大器放大后，通过ADE7755内部的模数转换器转换为两路数字信号，然后经乘法、低通滤波、数字频率变换等电路的处理后，从ADE7755的CF端口输出与瞬时功率成正比的脉冲。脉冲信号经过光电隔离器后，送至MSP430的P1.1端口，P1.1端口被设置为上升沿触发中断模式，MSP430单片机通过中断方式，根据设定的不同时段累计脉冲数，并将脉冲数转化为电量进行存储并送到LCD上显示。

由于复费率电能表为分时计费方式，所以电量计算与时间有着密切联系。为保证时钟的精度，系统采用外置串行实时时钟芯片DS1307。为保证数据安全，系统添加片外独立的EEPROM芯片AT24C02，用以备份存储在单片机Flash中的本月及上月总用电量、各时段用电量及其他重要信息。智能电表同时具有红外通信接口和RS-485通信接口，用于参

数设置及抄表，两种接口在物理层上相互独立。

三　嵌入式实时操作系统μC/OS-Ⅱ及其在MSP430F449上的移植

μC/OS-Ⅱ是一种基于优先级的可剥夺型的实时内核，自从1992年发布以来，在世界各地获得了广泛的应用，它是一种专门为嵌入式系统设计的内核，目前已被移植到40多种不同结构的CPU上，运行在从8位到64位的各种系统上。尤其值得一提的是，该系统从2.51版本之后，就通过了美国FAA认证，可运行在诸如航天器等对安全要求极为苛刻的系统之上。鉴于μC/OS-Ⅱ可免费获得代码，对于RTOS而言，无疑是很好的选择。设计选用的μC/OS-Ⅱ版本为V2.76。

μC/OS-Ⅱ要运行在MSP430F449上首先要做的就是移植工作。μC/OS-Ⅱ的移植只需要修改与处理器有关的代码，具体内容有：

(1)os_cpu.h中设置堆栈的生长方向，因MSP430单片机堆栈从高地址向低地址生长，故有：

#define OS_STK_GROWTH 1 //堆栈从高地址向低地址方向生长。

(2)os_cpu.h中需要申明用于开关中断的宏，如下：

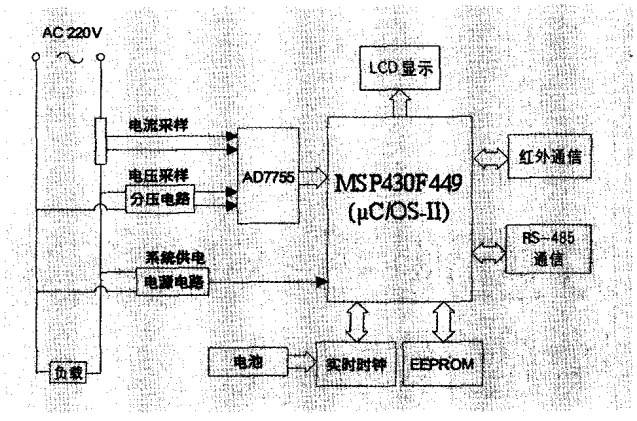
#define OS_CRITICAL_METHOD 3

#define OS_ENTER_CRITICAL() (cpu_sr = OSCPUSaveSR()) /*关中断*/

#define OS_EXIT_CRITICAL() (OSCPURestoreSR(cpu_sr)) /*开中断*/

#define OS_TASK_SW() OSCtxSw() /*用于任务切换*/

(3)os_cpu.h中需要设置10个数据类型，即：



系统框图

typedef unsigned char BOOLEAN

等。

os_cpu_a.s43中需要修改6个汇编语言的函数，这些函数都是与处理器相关的汇编程序。

os_cpu_c.c需要用C语言编写6个简单的函数。其中1个初始化任务堆栈函数和5个任务创建挂钩函数。

修改主头文件include.h，将上面的3个头文件和其他自己需要的头文件加入。

μC/OS-Ⅱ是一个可裁剪的操作系统，这意味着用户可以去掉不需要的服务。受单片机存储空间的限制，写好移植代码后，通过配置os_cfg_r.h选择需要的服务，可减少内核占用的存储空间。完成上述工作后μC/OS-Ⅱ即可运行在MSP430单片机上。

四　基于μC/OS-Ⅱ的软件设计

在μC/OS-Ⅱ环境下，各个不同的任务使用独立的堆栈空间，各个任务之间是相互独立的，实时内核通过任务创建、延时、挂起、恢复、删除等管理任务，各任务之间可通过信号量、邮箱、消息队列等操作系统的服务程序传递信息。μC/OS-Ⅱ可管理多达64个任务。内核占用的存储器容量与建立的任务数有关，考虑到系统的实际需求及单片机的片上资源有限，将最大任务数设为15，可节省一部分存储空间。系统的主要任务及优先级如表所示。其中任务优先级的值越小，任务的优先级越高。与传统的代码不同，在μC/OS-Ⅱ下，各任务代码一般为无限循环方式，μC/OS-Ⅱ总是运行进入就绪态任务中优先级最高的那个任务。在任务之间进行切换的工作是由调度器(Scheduler)完成的。

系统主要任务及优先级

任务描述	任务函数	任务优先级
电量计量	task_msr(void *pdata)	4
RS-485通信	task_rs485(void *pdata)	6
红外通信	task_infrd(void *pdata)	7
EEPROM存储	task_eeprom(void *pdata)	9
实时时钟	task_rtc(void *pdata)	10
LCD显示	task_lcd(void *pdata)	15

五　结束语

控制电路核心采用MSP430F449单片机，μC/OS-Ⅱ实时操作系统在MSP430单片机上的移植会带来额外的存储空间以及时间的消耗，实验表明这些额外的消耗处在可以接受的范围之内。基于MSP430单片机和μC/OS-Ⅱ的智能电表既可发挥单片机的功能特性，又能充分体现μC/OS-Ⅱ的多任务及实时性，完全能够满足系统设计需求。运用多任务编程方法设计智能电表，根据任务的关键性分配各任务的优先级，降低了程序设计的难度，缩短了开发周期，提高了系统的稳定性和可靠性。