

嵌入式操作系统 $\mu\text{C}/\text{OS-II}$ 在数字化电源控制卡上的应用研究

许升辉 许瑞年 柯新花 沈天健

(中国科学院上海应用物理研究所 上海 201800)

摘要 本文介绍了上海光源数字化电源控制卡 DPSC-100 的性能,对上海光源数字化电源控制卡软件进行了分析,提出了一种基于嵌入式操作系统的软件开发思路。然后阐述了移植嵌入式操作系统 $\mu\text{C}/\text{OS-II}$ 的关键环节。最后分析了 DPSC-100 的软件结构,给出了其软件的状态转移图,并在此基础上设计了基于 $\mu\text{C}/\text{OS-II}$ 的数字化电源控制卡控制程序。

关键词 上海光源, 数字化电源控制卡, $\mu\text{C}/\text{OS-II}$, 嵌入式操作系统

中图分类号 TL503.5

上海光源数字化电源控制卡 DPSC-100 由本组与成都大博有限公司联合开发,是我国第一套投入使用的高精度数字化电源控制卡产品。DPSC-100 由 DSP 卡、ADC 卡和背卡组成,长期稳定度达 100 ppm,提供 4 路 PWM 输出,可工作在斩波器、H 桥、半桥、全桥及全桥 ZVS 方式。DSP 卡以 TMS320F2812 为核心,实现控制算法及 PWM 信号生成、监控、保护^[1]。控制卡软件流程见图 1。

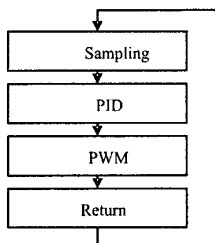


图 1 上海光源数字化电源控制卡 DPSC-100 软件流程图
Fig.1 Flow chart of the DPSC-100 code

DSP 软件开发是数字化电源控制卡的重要内容。上海光源数字化电源控制卡 DPSC-100 的 DSP 软件开发是基于全过程的顺序编程,但若程序庞大,编写、调试、扩展和维护都将十分困难,故引入嵌入式操作系统,简称为 RTOS,以使控制程序的设计和扩展变得容易,无需大改动就可增加新功能。与传统基于过程的软件开发不同,嵌入式操作系统是基于任务和消息处理编程。RTOS 通过将控制程序分割成独立的任务,使控制程序设计大为简化。实际上,给控制系统增加一些低优先级的任务,几

乎不影响其对高优先级任务的响应时间。

本文将嵌入式操作系统 $\mu\text{C}/\text{OS-II}$ 移植到上海光源数字化电源控制卡 DPSC-100 的 DSP 芯片 TMS320F2812 中,并设计了基于 $\mu\text{C}/\text{OS-II}$ 的控制程序。

1 $\mu\text{C}/\text{OS-II}$ 的移植

$\mu\text{C}/\text{OS-II}$ 问世于 1992 年,是基于优先级的可抢先的硬实时内核,是专门为嵌入式设备设计的内核,具有可移植、可固化、可裁减等特点,已移植到 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。

$\mu\text{C}/\text{OS-II}$ 源代码可分为与处理器无关代码、处理器相关代码、与应用有关代码,为不同处理器编写处理器相关代码,是 $\mu\text{C}/\text{OS-II}$ 移植的关键所在^[2]。

内核将应用系统和底层硬件有机地结合成实时系统,要使同一内核能适用于不同的硬件体系,内核和硬件间就须有一中间层,这就是与处理器相关的代码。处理器不同,这部分代码也不同。在 $\mu\text{C}/\text{OS-II}$ 中,这部分代码分成 OS-CPU.H, OS-CPU-A.ASM 与 OS-CPU-C.C 等三个文件。

OS-CPU.H 包括用#define 定义的与处理器相关的常量、宏和类型定义,有系统数据类型定义、栈增长方向定义、关中断和开中断定义、系统软中断定义等。

OS-CPU-A.ASM 对处理器的寄存器进行操作,须用汇编语言编写 OSStartHighRdy()、OSCtxSw()、

OSIntCtxSw()与 OSTickISR()等四个子函数^[3]。

OSStartHighRdy()在多任务系统启动函数 OSStart()中调用,完成的功能是:设置系统运行标志位 OSRunning=TRUE,将就绪表中最高优先级任务的栈指针装载到 SP 中,并强制中断返回。这样就绪的最高优先级任务就如同从中断里返回到运行态一样,使整个系统得以运转。以下为该函数部分代码:

```
MOVW    DP, #_OSTCBHighRdy;
        (取 OSTCBHighRdy 的段地址)
MOVL    XAR4, @_OSTCBHighRdy;
        (将高优先级任务堆栈的首地址赋给 XAR4)
MOVW    AL, *+XAR4[0];
        (当前辅助寄存器地址 XAR4 加上 0 后地址赋给 AL)
MOVW    @SP, AL
POP RPC
IRET[4]; (中断返回)
```

OSCtxSw()在任务级切换函数中调用。任务级切换通过 TRAP #31 人为制造的中断来实现。ISR 的向量地址须指向 OSCtxSw()。这一中断完成的功能是:保存任务的环境变量(主要是寄存器的值,通过入栈来实现),将当前 SP 存入任务 TCB 中,载入就绪最高优先级任务的 SP,恢复就绪最高优先级任务的环境变量,中断返回。这样,就完成了任务级的切换。

OSIntCtxSw()在退出中断服务函数 OSIntExit()中调用,实现中断级任务切换。具体完成的任务是:调整栈指针(因为调用函数会使任务栈结构与系统任务切换时堆栈标准结构不一致),保存当前任务 SP^[5]。载入就绪最高优先级任务的 SP,恢复就绪最高优先级任务的环境变量,中断返回。这样就完成了中断级任务切换。以下为该函数部分代码:

```
SUBB    SP, #4; 调整堆栈指针
MOVW    DP,
        #_OSTCBCur; OSTCBCur->OSTCBStkPtr = SP
MOVL    XAR4, @_OSTCBCur
MOV     AL, SP
MOV     *+XAR4[0], AL
```

OSTickISR()是系统时钟节拍中断服务函数,这是一个周期性中断,为内核提供时钟节拍,频率越高系统负荷越重。其周期决定了内核所能给应用系统提供的最小时间间隔服务,一般只限于 ms 级(与 CPU 有关)。该函数具体内容包括:保存寄存器(2812 发生中断时,硬件只自动保存部分寄存器值),调用 OSIntEnter(),调用 OSTimeTick(),调用 OSIntExit(),恢复寄存器,中断返回。

μC/OS-II 在 OS-CPU-C.C 文件中定义了六个函数:

```
OSTaskStkInit()
OSTaskCreateHook()
OSTaskDelHook()
OSTaskSwHook()
OSTaskStatHook()
OSTimeTickHook()
```

其中最重要的是 OSTaskStkInit(),其它都是对系统内核扩展时用的,若无特殊要求,只需简单地将它们实现为空函数即可。

OSTaskStkInit()是在用户建立任务时系统内部自己调用的,对用户任务的堆栈进行初始化,使建立好的进入就绪态任务的堆栈与系统发生中断并且与环境变量保存完毕时的堆栈结构一致。建立好的堆栈结构为:

Pdata (Low Address)

```
pdata
ST0
T
AL
AH
PL
PH
AR0
AR1
ST1
DP
IER
DBGSTAT
task
task
AR1H
AR0H
AR4
AR4H
AR5
AR5H
AR6
AR6H
AR7
AR7H
TL
T
RPCL
```

RPCH (High Address)

结合 DSP 芯片自带的定时器，我们对μC/OS-II 做了一些实时性能测试：其任务切换时间为 1.70 μs，中断响应时间为 0.193 μs，性能指标良好，可满足大部分应用程序要求。

2 基于μC/OS-II 的控制程序设计

2.1 控制程序分析

上海光源数字化电源控制卡 DPSC-100 控制主程序基于状态转移设计，在主程序中设置初始状态，进入状态机循环。系统共分为 7 个状态，包括启动(Startup)、待机(Standby)、上电(PowerOn)、运行(Run)、关机(PowerOff)、关机延时(PowerOffDelay)和出错(Fatal)。状态转移条件由程序运行结果或外部控制命令所决定。具体如下：

(1) 启动(Startup)：初始化系统工作环境，设置系统初始状态，初始化电源故障。开启与 EEPROM 的 SPI 通信(存储参数)，开启本地异步串口通信 SCIO，开启远控光纤异步串口通信 SCI1，开启与 ADC 卡 FPGA 的 McBSP 通信，获取 PWM 信息参数，获取 PID 控制参数，更改系统状态。如控制卡启动顺利，则跳转到待机状态；否则跳转至出错状态，进行出错处理。

(2) 待机(Standby)：初始化 ADC 同步采样，关

闭 PWM，设置输出数字量，控制指示灯，通过 McBSP 巡检 ADC 进行采样，巡检本地和远控的接收与发送，共享同一命令解释 doLnkMsgRecv()函数进行参数获取与发送，巡检数字输入量。待机过程中根据程序运行结果或外部控制命令跳转到相应状态。

(3) 上电(PowerOn)：完成上电过程联锁控制，上电过程中根据程序运行结果或外部控制命令跳转到相应状态。

(4) 运行(Run)：巡检 ADC 采样 PollingDualAdc，巡检 PID 控制 PollingPid，巡检本地和远控的接收与发送 PollingUartAll，巡检 PollingDigitIn 数字输入量。巡检 PID 控制 pollingPid()中调用 PidProcess()函数进行算法处理。运行过程中根据程序运行结果或外部控制命令跳转到相应状态。

(5) 关机(PowerOff)：完成关机过程控制。关机过程中根据设置的保护参数——重启反应时间和最大通风判断时间的大小跳转到相应状态。

(6) 关机延时(PowerOffDelay)：完成延时机过程控制。按设定的重启反应时间延时关机，延时期满后，跳转至待机状态。

(7) 出错(Fatal)：完成电源故障出错处理。

系统状态机如图 2 所示：

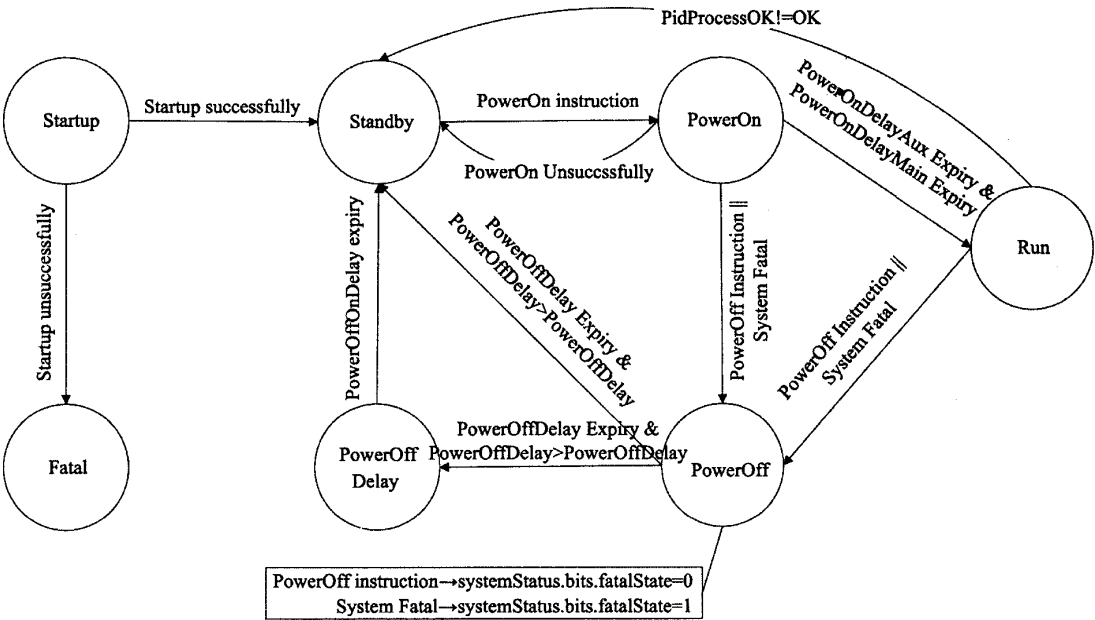


图 2 上海光源数字化电源控制卡 DPSC-100 状态迁移图
Fig.2 DPSC-100 state transfer

2.2 基于 $\mu\text{C}/\text{OS-II}$ 的控制程序设计

$\mu\text{C}/\text{OS-II}$ 程序设计基于任务, 而上海光源数字化电源控制卡 DPSC-100 的程序则基于状态, 两者的程序设计思想不同, 如何将两者结合, 是设计基于 $\mu\text{C}/\text{OS-II}$ 的控制卡程序的最大问题^[6,7]。

不难看出, 控制卡主控程序虽有七个系统状态, 但实现控制卡控制功能的还是在“运行(Run)”状态中, 控制卡大部分时间也是运行在此状态中, “运行(Run)”状态是控制卡最重要的状态。

鉴此, 我们将系统分为五个任务: TaskStart、TaskDualAdc、TaskPid、TaskUartAll 与 TaskDigitIn, 其任务优先级依次为 0、1、2、3、4, 任务之间通过 $\mu\text{C}/\text{OS-II}$ 提供的系统邮箱进行通信。

TaskStart 为主控程序, 实现启动(Startup)、待机(Standby)、上电(PowerOn)、关机(PowerOff)、关机延时(PowerOffDelay)和出错(Fatal)状态所完成的功能, 并负责各状态间的转移。

TaskDualAdc、TaskPid、TaskUartAll 与 TaskDigitIn 共同实现原程序在“运行(Run)”状态下

所完成的控制功能。TaskDualAdc 负责 AD 卡与 DSP 卡之间的通信, 传送 ADC 转换后的数据至 DSP 芯片中; TaskPid 负责调用 PID 算法, 控制 PWM 的输出; TaskUartAll 负责实现 DPSC-100 的本地与远程通信功能; TaskDigitIn 负责数字输入量的检测与显示。

基于此设计思想, 程序运行流程如下: 初始化操作系统 $\mu\text{C}/\text{OS-II}$, 系统启动, 运行最高优先级任务, 任务 TaskStart 被激活, 设置系统初始状态, 进入状态机循环, 程序运行至“运行(Run)”状态, TaskStart 挂起, 按任务优先级依次激活 TaskDualAdc、TaskPid、TaskUartAll、TaskDigitIn, 实现控制卡电源控制功能。DPSC-100 状态机由系统的五个任务根据状态转移条件实现。任务 TaskStart、TaskDualAdc、TaskPid、TaskUartAll 及 TaskDigitIn 由嵌入式操作系统 $\mu\text{C}/\text{OS-II}$ 依据系统邮箱 Mailbox 和任务优先级进行统一调度, 实现任务的并行运行。图 3 为基于 $\mu\text{C}/\text{OS-II}$ 的 DPSC-100 系统调度图。

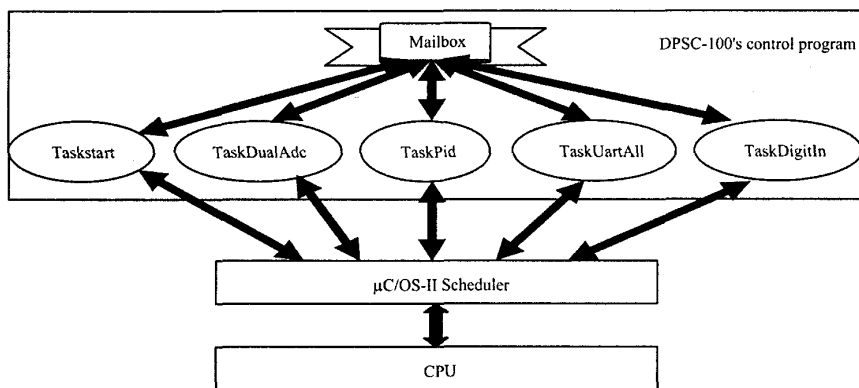


图 3 基于 $\mu\text{C}/\text{OS-II}$ 的 DPSC-100 系统调度图
Fig.3 DPSC-100's scheduler based on $\mu\text{C}/\text{OS-II}$

经实验验证, 基于此设计方法, 可较好地嵌入操作系统 $\mu\text{C}/\text{OS-II}$ 应用到上海光源数字化电源控制卡 DPSC-100 中, 实现了控制卡电源控制功能设计, 简化了控制卡的软件设计, 提高了控制卡软件的继承性。

3 总结

$\mu\text{C}/\text{OS-II}$ 实时性强、代码量小, 可被广泛移植到 x86、ARM、DSP、MCS51 等许多处理器上。国外象 APC、ROTEK 等企业都在产品中成功使用了 $\mu\text{C}/\text{OS-II}$ 内核, 从而证明了 $\mu\text{C}/\text{OS-II}$ 是一个稳定可靠的内核。本文成功地嵌入操作系统 $\mu\text{C}/\text{OS-II}$

移植到上海光源数字化电源控制卡 DPSC-100 上, 并基于 $\mu\text{C}/\text{OS-II}$ 和原控制程序设计了电源控制卡控制功能, 为上海光源数字化电源控制卡 DPSC-100 的软件升级和维护提供了一种思路。

参考文献

- 1 上海应用物理所, 成都大博有限公司. 数字电源控制系统 DPSC-100 使用手册. 4-6
Shanghai Institute of Applied Physics, Double Co.Ltd.
Digital Power Supply Controller DPSC-100 Manual, 4-6
- 2 刘海峰, 刘百芬. 华东交通大学学报, 2006,23(1): 112-114
LIU Haifeng, LIU Baifen. Journal of East China

- Jiaotong University, 2006, **23**(1): 112–114
- 3 Labrose J J 著, 邵贝贝等译. 嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ (第 2 版), 北京: 北京航空航天大学出版社, 2007. 283–317
- Labrose J J. $\mu\text{C}/\text{OS-II}$ The Real-Time Kernel (Second Edition). Beijing: Beihang University Press, 2007. 283–317
- 4 TMS320F28x DSP CPU and Instruction Set Reference Guide (Rev. C), TEXAS INSTRUMENTS, Chapter 6, 114
- 5 TMS320F28x Control and Interrupts Reference Guide (Rev. A), TEXAS INSTRUMENTS, Chapter 3, 2–23
- 6 邵克存, 毛六平, 戴瑜兴. 电工技术, 2005, **3**: 35–37
GAO Kecun, MAO Liuping, DAI Yuxing. Electric Engineering, 2005, **3**: 35–37
- 7 金雪丰, 王华敏, 董宏宪, 等. 基于 $\mu\text{C}/\text{OS-II}$ 的分布式应急电源控制系统设计. <http://www.gd-emb.org/detail/id-50104.html>
- JIN Xuefeng, WANG Huamin, DONG Hongxian, *et al.* Design Distributed Power Supply Control System Based On $\mu\text{C}/\text{OS-II}$. <http://www.gd-emb.org/detail/id-50104.html>

Application of embedded system $\mu\text{C}/\text{OS-II}$ on digital power supply controller

XU Shenghui XU Ruinian KE Xinhua SHEN Tianjian

(Shanghai Institute of Applied Physics, Chinese Academy of Sciences, Shanghai 201800)

Abstract To upgrade the DPSC-100, a controller for digital power supplies of Shanghai Synchrotron Radiation Facility, an embedded system using $\mu\text{C}/\text{OS-II}$ was proposed. Key points to transplanting $\mu\text{C}/\text{OS-II}$ to TMS320F2812 are discussed in this paper. The DPSC-100's state transfer and design of the control program based on $\mu\text{C}/\text{OS-II}$ are given.

Key words SSRF, Digital power supply controller, $\mu\text{C}/\text{OS-II}$, Embedded system

CLC TL503.5