

# 基于 ARM 与 $\mu$ C/OS-II 的嵌入式信息交互终端设计<sup>①</sup>

李红艳

(湖北经济学院 计算机学院, 武汉 430205)

**摘要:** 为了满足终端系统对处理速度、实时性和可扩展性等技术指标方面的更高需求, 以高性能的 32 位嵌入式 ARM 芯片为处理器, 采用嵌入式实时操作系统  $\mu$  C/OS-II, 设计了一种软硬件易移植、易裁减, 结构优化的电子交易信息终端。根据资源需求, 给出了终端系统设计的硬件结构。在硬件目标平台上移植了  $\mu$  C/OS-II 操作系统, 构建了嵌入式软件开发平台, 提出了新颖的带中间件模块设计的系统软件结构, 确定了系统任务划分及优先级设定, 并给出了主程序执行流程, 同时阐述了系统实现的一些关键技术。通过中间件模块设计方案增强了系统软件的可移植性、可扩展性和开发效率, 由此设计的终端稳定可靠。

**关键词:** 嵌入式系统; ARM;  $\mu$  C/OS-II; 信息终端; 中间件

## Design of Embedded E-Trade Information Terminal Based on ARM and $\mu$ C/OS-II

LI Hong-Yan

(School of Computer, Hubei University of Economics, Wuhan 430205, China)

**Abstract:** An e-trade information terminal is designed to meet the higher requirements for processing speed, real time capability and expansibility based on the 32 bit ARM chips of high performance as processor and combining with real time operation system  $\mu$  C/OS-II. According to the resource requirements, the hardware structure of terminal is proposed. How to form an embedded development platform based on  $\mu$  C/OS-II. In addition, the transplantation of  $\mu$  C/OS-II in ARM microprocessor is discussed. A novel middleware module architecture is put forward, which increases reuse, transplant performance and expansibility. The division of task, assignment of software priority and the execution path of main program are determined. At last, some key technologies for implementing the information terminal are expatiated. The designed terminal runs reliably.

**Keywords:** embedded system; ARM;  $\mu$  C/OS-II; Information terminal; middleware

随着信息技术与网络技术的飞速发展, 诸如网络支付、电子交易等信息交互业务给人类生活带来便利, 信息交互终端表现出巨大的市场潜力。信息交互终端主要是通过自身输入界面与平台间的通信实现将用户相关交易请求信息与平台进行交互, 完成远程交易, 同时根据业务的不同为用户打印相关凭证功能。目前, 信息交互终端大多采用 8 位单片机为核心控制器, 软件设计采用传统的主循环控制, 通过判断不同的标志调用相应功能函数的模式。为了保证终端与远端电子

交易平台之间正常通信, 往往需要将通信部分设计成独立部件, 并通过串口与终端业务进行通信。这种采用多 CPU 结构设计方式主要缺点是结构复杂、调试困难、处理速度慢、系统可扩展性差、通信可靠性差、开发时间长、调试困难和实时性差。而随着计算机和微电子科学与技术的发展, 微处理性能的提高, 以及嵌入式系统可靠性、低成本、体积小等优点, 嵌入式系统越来越多地用到终端系统设计领域<sup>[1-4]</sup>。 $\mu$ C/OS-II 是个源码公开、易移植、易固化、易裁减、占先式、

① 基金项目:湖北省教育厅科技处中青年项目(Q20102202);湖北省教育科学“十一五”规划课题(2009B422);湖北经济学院校级科研项目(2008011)

收稿时间:2010-06-21;收到修改稿时间:2010-08-04

多任务实时嵌入式操作系统, 被广泛用于工业控制和其他嵌入式产品中<sup>[5-7]</sup>。本设计以提高系统的处理速度、实时性和可扩展性为目标, 采用嵌入式系统的设计思想, 硬件方面使用低成本、低功耗、高性能的 32 位高速 ARM 处理器; 软件方面引入实时多任务嵌入式操作系统  $\mu\text{C}/\text{OS}-\text{II}$ , 进行多任务调度, 各任务模块可以独立开发, 通过消息传递进行通信, 降低开发难度, 缩短开发时间, 提高系统的实时性和扩展性。

## 1 硬件设计

图 1 所示为终端系统硬件设计平台, ARM 选用 PHILIPS 公司的 LPC2220, LPC2220 微控制器是基于一个支持实时仿真和嵌入跟踪的 32 位 ARM7TDMI-S CPU, 它基于 ARM 通用体系结构的 RISC 微处理器, 具有标准 32 位 ARM 指令集和 16 位 THUMB 指令集, 具有低成本、低功耗、高性能的优点, 是系统的核心部件, 负责整个终端系统的运行。ARM 通过片选信号、总线复用分别实现对 LCD 显示驱动芯片、CPLD、FLASH 和 TDA8007 进行控制。CPLD 芯片选用的是 Altera 公司的 MAX7000S 系列器件 EPM7128SLC84, 此芯片由 128 个逻辑宏单元和 2500 个可用逻辑门组成, 可以通过 JTAG 在线编程, 采用 CPLD 扩展 I/O 接口主要提供了 SAM 卡、磁卡、大容量卡的时钟、复位信号、数据输入信号、电源信号等控制接口, 以及键盘的扫描、打印机电流、相序、出针控制等接口。CMX865 芯片主要负责 DTMF 信号、FSK 信号的调制与解调, ARM 通过 C/BUS 总线对 CMX865 芯片进行控制、操作; PHILIPS 公司的 IC 卡接口芯片 TDA8007 被用来实现 SAM 卡、大容量卡以及磁卡的驱动, ARM 通过总线复用对 TDA8007 的寄存器进行控制。

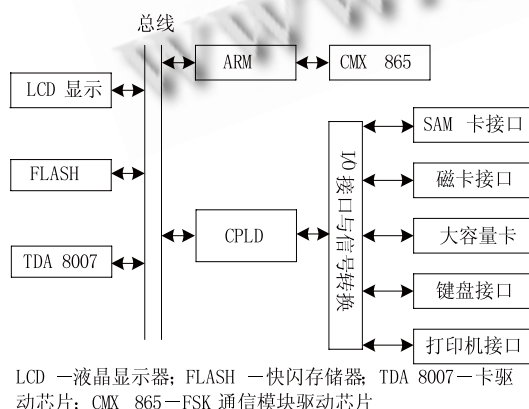


图 1 终端硬件结构

整个硬件目标平台以 LPC2220 处理器为核心, 扩展了 2M 字节 Flash, 用来下载程序、参数文件、字库文件等, Flash 映射在 LPC2220 的 Bank0 上, 系统上电后处理器从 Flash 的 0x8000000 地址处取指令开始运行; 64K 字节 SDRAM, 由于需要对 Flash 进行写操作时不能读 Flash, 否则会发生毁灭性事故, 所以将对 Flash 操作方面的函数和中断等程序搬到 SDRAM 运行, SDRAM 还需开辟出部分空间提供 LCD 显示缓冲区、打印缓冲区, 以及用来存储临时数据、堆栈等, SDRAM 从地址 0x40000000 开始运行。

## 2 软件设计方案

由于本系统中应用程序较多、实时性要求较高, 在软件设计上引入多任务实时操作系统  $\mu\text{C}/\text{OS}-\text{II}$  非常必要。软件设计采用模块化设计方案, 减少终端开发周期。

### 2.1 $\mu\text{C}/\text{OS}-\text{II}$ 在 LPC2220 上移植

$\mu\text{C}/\text{OS}-\text{II}$  在 LPC2220 上移植工作主要集中在操作系统程序启动运行、操作系统功能裁减上及增加 10ms 粒度的系统软件定时器三方面。移植增加了启动代码, 终端的汇编接口代码以及初始化向量中断控制器部分代码, 并以操作系统的时钟为基础增加了软件定时器功能函数, 改写了与处理器相关的文件<sup>[8,9]</sup>。

(1) OS\_CPU.H 文件。修改头文件参数定义几种与编译器相关的数据类型、定义栈的增长方向、定义使能、禁止中断宏等。

(2) OS\_CPU\_C.C 文件。包含与移植相关的 6 个函数: OSTaskCreatHook()、OSTaskDelHook()、OSTaskshook()、OSTaskstatHook()、OSTimeTickHook()、OSTaskstklinit(), 其中前 5 个函数主要用于扩展  $\mu\text{C}/\text{OS}-\text{II}$  功能, 必须被声明但不必包含任何代码。真正需要修改的只有 OSTaskstklinit() 函数。

### 2.2 软件结构

终端系统软件结构如图 2 所示, 它是个开放的体系, 具有可置换、可裁减的功能, 只要模块的接口相同, 功能相同的模块可以相互置换, 同时它可以根据系统的需要增减相应的功能模块。本操作系统内核没有文件系统、图形系统和内存管理, 只有中断管理和任务管理。系统结构最底层是硬件物理层, 再上一层是操作系统内核和 BSP 驱动层, 中间包含软件中间件

层，最上面才是终端应用程序层。由于操作系统内核和 BSP 存在函数互调用关系，将其归纳为同一结构层次。其中将主控、打印、通信、显示等模块划分为中间件层，这些模块通过调用底层标准函数为业务模块提供标准化对外接口函数，可以很容易地移植到别的硬件和软件平台上，中间件层中打印模块和通信模块是系统模块化的基础，诸如打印模块仅提供启动打印设备资源、打印操作、释放打印设备资源三个标准函数接口，允许相同接口函数任意模块置换。采用这种中间件软件设计结构突出优势是可以提供一个应用嵌入式软件开发平台，借助于此平台，根据不同应用需求，可以缩短终端开发周期。目前，通过软件中间件层将消息传递、业务任务创建等功能均封装成标准化函数接口，应用业务开发者可以在不了解操作系统基础上，完成应用程序开发，提高程序开发效率。

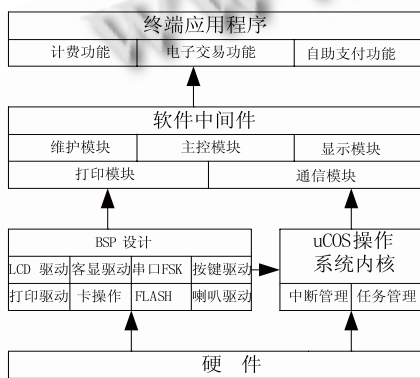


图2 系统软件结构

### 2.3 任务及优先级设计

按终端所要求实现的功能，将整个系统划分为几个并行存在的任务，并根据任务重要程度及被允许抢占概率确定其优先级。主控和显示共用一个任务，设置其优先级 10；打印模块单独占用一个任务，确定优先级 14；通信模块存在 FSK 通信和串口通信两个模块，为防止等待通信数据导致任务阻塞，影响数据正常通信，通信模块需要创建两个任务，优先级分别是 18 和 22；三个业务模块独立存在，优先级设置分别是 25、28、31；维护模块任务优先级最低确定为 34，BSP 模块不需要创建任务，BSP 模块事件一般通过按键触发、中断或函数调用实现，仅存在发送消息函数。优先级间隔设置目的主要是方便模块增加任务，而不用重新分配整个系统优先级。

当终端上电复位后，首先进行嵌入式处理器及外设初始化；接着进行嵌入式操作系统  $\mu\text{C}/\text{OS}-\text{II}$  和用户界面初始化，同时液晶显示主界面；完成上面的工作后，建立任务间通信的信号量、消息队列；上面准备工作做好后，开始创建任务并且分配任务优先级，所有新建任务被置为就绪态；启动任务调度，系统程序从优先级最高的任务开始执行。终端系统的主程序执行流程图如图 3 所示。平时  $\mu\text{C}/\text{OS}-\text{II}$  操作系统处于空闲状态，当有消息到来时任务调度机制根据消息类别调度相应任务使其进入运行状态，由于操作系统内核是剥夺型内核，当有优先级更高的任务就绪时，操作系统将任务切换先执行高优先级任务，等到高优先级任务处理完毕后进入休眠态，操作系统再切换任务到被中断的那个任务继续运行，所有任务处理完毕后操作系统再回到空闲状态或调度下一个低优先级的任务。

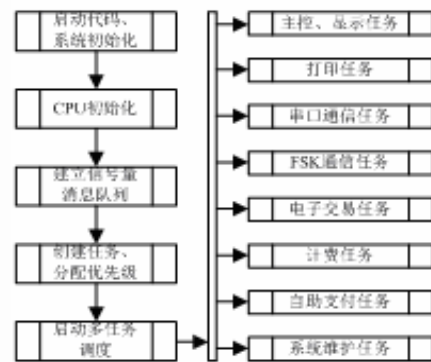


图3 终端系统主程序执行流程图

在终端系统中，每个任务都是一个无限循环，由  $\mu\text{C}/\text{OS}-\text{II}$  内核来进行调度。每个任务可能处于 4 种不同的状态：休眠态、就绪态、挂起态和运行态。在运行状态下，还可能由于发生中断而转向中断服务子程序。任务管理函数的调用将引起各任务状态间的转换，它们的关系如图 4 所示：

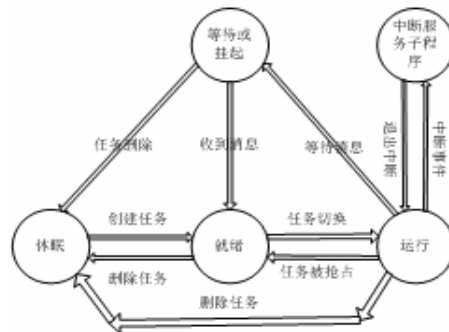


图4 任务状态转换关系图

### 3 关键技术

#### 3.1 SPI 资源的复用

终端系统中 SPI 中断源有射频卡操作、打印送数据操作及 CMX865 通信操作, 实际使用过程中任意两种, 甚至三种操作都有可能同时操作 SPI 端口。比如 SPI 正在向打印机芯寄存器缓存送点行打印数据, 此时线路上有 FSK 信号, CMX865 芯片产生中断信号引起 SPI 中断, 接收的 FSK 信号将发送到机芯打印缓存, 导致打印操作异常; 而打印模块任务优先级高, 通信模块也没有机会获取实际 SPI 接收的 FSK 通信信号, 导致通信失败。射频卡与其他两种操作复用 SPI 资源同样也会出现故障问题, 这里仅以打印和 FSK 操作为例解析处理方法。为了解决此问题, 对 FSK 资源加保护信号是必要的, 但是 SPI 与机芯寄存器之间数据传输是没有片选信号的, 简单说就是只要 SPI 上有信号就会自动覆盖存入打印机芯寄存器。根据理论分析, 机芯打印点行数据时间间隔约 4 毫秒, FSK 通信 1 字节接收时间约 9 毫秒, 向机芯发送点行数据时间约 2.5ms。设计 200 微秒粒度循环定时器, 以打印点行数据 4 毫秒周期为基准, 每一个 4 毫秒周期内, 前 5 次进入定时器中断服务函数内, 允许判断是否有 CMX865 芯片请求 SPI 资源使用的标志, 如果有则执行 CMX865 通信操作相应中断服务函数。后面 15 次进入定时器中断服务函数 SPI 资源被打印独占, 以确保打印机芯寄存器数据不被其他 SPI 操作修改。这种设计方案可以确保每一个 9 毫秒周期内, CMX865 芯片能够至少获得一次 SPI 资源实现 CMX865 通信操作, 又能够保证打印在打印周期内向机芯寄存器发送完毕点行数据而且打印数据不会被修改。

#### 3.2 系统软件多任务设计

为了满足系统多任务之间的同步以及通信等要求, 必须通过消息的发送来实现其通信, 考虑到应用可能同时接收到多个模块发送来的消息, 统一采用消息队列机制接收任务消息。防止多任务使用公共资源的冲突, 需要对共用资源加信号量进行保护。 $\mu\text{C}/\text{OS}-\text{II}$  操作系统采用基于优先级的抢占式调度策略, 没有时间片的概念, 处理任务阻塞问题是系统设计的重要环节。由于每一个业务模块只分配一个任务, 业务模块执行过程中是不允许阻塞等待其他任务消息返回

结果的, 这样必须将可能要求阻塞等待处理结果事件放在中间件允许阻塞任务内执行。比如业务模块等待接收数据, 此时业务模块向通信模块发送等待接收数据消息, 通信任务只能阻塞等待接收相应业务需求数据, 当然等待过程中, 通信任务实时让出 CPU 资源供其他优先级低任务有机会运行。正因为通信任务有阻塞可能, 系统通信任务根据通信类型不同创建了两个任务。

### 4 结语

基于当前终端系统对处理速度和功能可扩展性更高要求, 设计了一种基于 ARM 和  $\mu\text{C}/\text{OS}-\text{II}$  的嵌入式多功能信息交互终端。在硬件方面, 以 PHILIPS 公司的 LPC2220 芯片为处理器, 通过总线复用对外围驱动芯片进行控制, 并给出了终端系统设计的硬件结构; 在软件方面, 提出开放式软件设计结构, 便于软件模块移植和裁减, 提高了软件开发效率。最后介绍了终端系统设计的一些关键技术。

#### 参考文献

- 1 田家林, 陈利学, 寇向辉. 基于 ARM 与 FPGA 的嵌入式数控系统设计. 机床与液压, 2007, 35(2): 93-95.
- 2 彭国盛. 基于 ARM 和以太网的振动信号采集器设计. 电力自动化设备, 2007, 27(1): 87-89.
- 3 聂伟, 李云瑞. 基于 ARM 和  $\mu\text{C}/\text{OS}-\text{II}$  的车载移动终端设计. 北京化工大学学报, 2006, 33(5): 71-74.
- 4 王田苗, 陈友东. 基于  $\mu\text{C}/\text{OS}-\text{II}$  嵌入式数控系统研制. 北京航空航天大学学报, 2006, 32(4): 471-475.
- 5 王福刚, 曾兵. 基于嵌入式系统的以太网语音通信系统设计. 电力自动化设备, 2006, 26(11): 98-100.
- 6 刘志雄, 李浙昆. 基于 ARM 和  $\mu\text{C}/\text{OS}-\text{II}$  的嵌入式移动机器人平台设计. 机电产品开发与创新, 2007, 20(1): 14-15.
- 7 张利锋, 程明霄. 基于 ARM 和  $\mu\text{C}/\text{OS}-\text{II}$  的高性能分析仪平台设计. 微处理机, 2007, (1): 95-98.
- 8 陆强, 戴文. 基于 ARM 和  $\mu\text{C}/\text{OS}$  嵌入式系统移植的研究. 山东科技大学学报, 2006, 25(4): 83-86.
- 9 尚宇, 鄧琦.  $\mu\text{C}/\text{OS}-\text{II}$  在 LPC2210 上的移植研究. 计算机技术与发展, 2007, 17(2): 103-106.