

基于 S3C2440 与 WinCE 平台上的 FIQ 驱动程序的实现

开元海¹, 汪超², 郑勇³

(1. 贝加莱工业自动化西安分公司, 陕西 西安 710075;

2. 台州职业技术学院 机电工程学院, 浙江 台州 318000;

3. 安徽省滁州市第一职业高级中学, 安徽 滁州 239002)

摘要: 在简述 WinCE 系统中普通 IRQ 中断处理过程的基础上, 针对采用普通 IRQ 中断的驱动程序实时性能不高的问题, 提出使用 ARM 处理器的 FIQ 中断在实时性能要求较高的场合中的应用方法。系统硬件部分使用 ARM S3C2440 处理器作为主控部分, 软件部分开发出 WinCE5.0 操作系统下使用 FIQ 中断针对步进电机驱动器的驱动程序。实践证明, 驱动程序使用 FIQ 中断是稳定的且能获得比 IRQ 中断更好的实时性能。

关键词: WinCE; S3C2440; FIQ

中图分类号: TP316.2

文献标识码: A

文章编号: 1674-7720(2012)24-0089-03

Realization of driver using FIQ based on S3C2440 and WinCE embedded system

Kai Yuanhai¹, Wang Chao², Zheng Yong³

(1. Xi'an Branch of Bernecker+Rainer Industry-Electronic GmbH, Xi'an 710075, China;

2. Taizhou Vocational & Technical College, Taizhou 318000, China;

3. Chuzhou NO.1 Vocational Senior School, Chuzhou 239002, China)

Abstract: Based on the brief introduction of general IRQ interrupt processing in WinCE system and the less real-time performance of the driver used IRQ interrupt, this paper put forward the method of using FIQ interrupt of ARM processor to achieve higher performance requirements in real time applications. The system uses ARM S3C2440 processor as the main control unit. The driver use FIQ interrupt for stepping motor driver was developed under the WinCE5.0 operating system. Practice has proved that the driver using the FIQ interrupt is stable and can obtain better performance than using IRQ interrupt.

Key words: WinCE; S3C2440; FIQ

Microsoft Windows CE 是一个 32 位、紧凑、高效、可扩展的操作系统, 适用于各种嵌入式系统和产品。它拥有多线程、多任务、确定性的实时、完全抢占式优先级的操作系统环境, 专门面向只有有限资源的硬件系统。同时, 它的模块化设计方式使得系统开发人员和应用开发人员能够为多种多样的产品来定制它, 可以选择、组合和配置 Windows CE 的模块和组件来创建用户版的操作系统^[1]。良好的用户界面使其在导航仪等消费类电子中得到了广泛的应用。在工业控制方面, 有少数厂家以它作为内核开发工业控制器, 也有一些厂家只是作为终端显示设备。在外围没有 CPLD、FPGA 或 DSP 协同工作的

情况下, 对于实时性较高的应用场合一般不会选择 WinCE。

本文以步进电机驱动程序为例, 针对普遍使用的 IRQ 中断, 提出用 ARM 处理器的 FIQ 中断在实时性能要求高的场合中的应用方法, 可以大大提高控制系统实时处理能力, 进而可以替代某些场合中使用的专用芯片或 PLD 芯片来满足性能要求, 以节约成本, 降低功耗。

1 IRQ 中断过程简述

对于一个硬件中断, 系统内核在捕获之后, 会交给 OEMInterruptHandler 函数处理, 这个函数就是实现中断处

理的中心函数。函数可以从 CPU 的寄存器里获得中断的信息,这些信息可以指出是哪个中断源触发了中断。

在得到触发中断源信息后,最简单的中断处理办法就是在 OEMInterruptHandler 中直接对中断源进行判断,然后调用服务程序。

另一种中断处理办法是应用程序建立中断服务线程(IST),然后申请一个系统逻辑中断号(SYSINTR),创建一个事件(Event),使用 InterruptInitialize 函数将 Event 与 SYSINTR 绑定,随后 IST 阻塞在等待 Event 上面。当硬件中断发生后,OEMInterruptHandler 只给系统返回一个 SYSINTR,用来标记需要哪个程序来服务中断,系统根据绑定关系激活相应的 Event,使得随后的 IST 得以运行。在中断处理完成之后需要调用 InterruptDone,参数为该中断的 SYSINTR,用来通知系统中断处理完成,系统重新使能该中断。这种处理方法可以使中断被关闭、系统被锁定的时间最短。此外,还需要实现 OEMInterruptEnable 和 OEMInterruptDisable 函数来改变处理器中断屏蔽寄存器^[2]。处理过程如图 1 所示。



图 1 WinCE 中断处理流程图

当使用上文所述 IRQ 中断服务程序结构时,为了实时性更好,采用直接在 OEMInterruptHandler 中对中断源进行处理的方式对步进电机驱动器 Q2HB44MA 和步进电机 57BYGHM403 进行控制。由于中断优先级和响应速度的原因,步进电机在旋转过程中产生剧烈振动,根本无法在实际应用场合中使用。

2 FIQ 中断驱动程序在 WinCE 系统中的实现

2.1 ARM 处理器快速中断 FIQ

FIQ 和 IRQ 是 ARM 处理器中两种不同类型的中断。FIQ 比 IRQ 有更高优先级,如果 FIQ 和 IRQ 同时产生,那么先处理 FIQ。当 CPU 处于 FIQ 模式处理 FIQ 中断的过程中,预取指令异常,未定义指令异常,软件中断全被禁止,所有的中断被屏蔽。所以 FIQ 会很快被执行,不会被其他异常或者中断打断。而 IRQ 不一样,当 ARM 处于

IRQ 模式处理 IRQ 中断时,如果来了一个 FIQ 中断请求,那么正在执行的 IRQ 中断处理程序会被抢断,ARM 切换到 FIQ 模式去执行该 FIQ 中断^[3]。

2.2 步进电机加减速实现

步进电机在启动或停止时若步进脉冲变化太快,转子由于惯性而跟不上电信号的变化,会产生失步或超步现象。一般来说对于电机起始转速小于 2 r/s 时,根据负载情况可能不会出现失步或超步的现象。但在转速较高、负载较大的情况下,为了避免失步或超步现象,通常采用加减速控制。加减速过程需要设定步进电机的起始频率、稳定频率和加速时间,但是在控制中则需要计算出加速过程所需要的脉冲个数。对于加速过程有以下关系式:

$$\sum_{i=0}^n \frac{1}{f_a + f_{\Delta} \times i} = t \quad (1)$$

$$f_b = f_a + f_{\Delta} \times n$$

经过化简计算得:

$$n = \left\lceil \left[(f_b - f_a) \times \frac{t - \frac{1}{2} \left(\frac{1}{f_b} + \frac{1}{f_a} \right)}{\ln f_b - \ln f_a} + 0.5 \right] \right\rceil \quad (2)$$

其中: f_a 为起始频率, f_{Δ} 为每步增长的频率, f_b 为稳定频率, t 为加速时间, n 为加速过程脉冲数量。

2.3 添加步进电机结构和 IOCTL 接口

WinCE 操作系统设计借鉴了 Windows 2000/XP 操作系统的设计,从体系结构上,它具有分层结构的特点,从上到下为应用程序层、操作系统层、OEM 层和硬件层。OEM 层是逻辑上位于硬件和操作系统之间的一层硬件相关代码。它的主要作用是对具体的硬件进行抽象,抽象出统一的接口,然后 WinCE 内核可以使用这些接口与硬件进行通信,这样在移植 WinCE 到新的硬件平台时可以减少对操作系统的修改,通俗地说就是为 WinCE 操作系统抹平 MCU 的差异,使其能方便地移植到其他 MCU 上运行。OEM 抽象层(OAL)是整个 OEM 层的主体,它包含了高度硬件相关的代码,OAL 主要负责 WinCE 内核与硬件交互。

为了能够用 WinCE 的流接口驱动程序操作 ARM 处理器的 FIQ 中断,需要修改 WinCE 系统 OEM 层中的 OAL 部分。在 s2440.h 文件中定义步进电机数据结构,包括起始频率、稳定频率、上升脉冲数、剩余脉冲数、当前上升/下降沿状态等运行参数和状态变量。

在 oemioctl.c 文件中定义步进电机结构的全局变量。为了使驱动程序可以访问到此变量,使用 WinCE 系统的 IOCTL 接口。在 WinCE5.0 中,应用程序和驱动程序可以通过调用 KernelIoControl() 函数来访问 WinCE 内核,导致调用 OEMIoControl 函数,这样应用程序和驱动程序就可以访问到 OAL 中的资源了,可以说 OEMIoControl 是一个很有用的 WinCE 内核输入/输出函数。在 OEMIoControl 函数中添加 case 语句使得驱动程序传来的运行参数和

应用奇葩

Example of Application

运行/停止命令可以得到记录和响应,在更新参数命令处理中把运行的参数赋值给步进电机结构变量;在开始运行命令处理中打开定时器 FIQ 中断使能、设置定时器周期、开始定时器运行;在停止命令中记录停止信息使得在当前脉冲循环完成后停止脉冲的输出。

2.4 添加中断处理过程代码

由于在 IOCTL 的开始运行命令处理中打开了定时器的 FIQ 中断使能,当定时器时间达到时,中断服务程序不再是 OEMInterruptHandler,而是 FIQHandler。由 FIQHandler 调用 OEMInterruptHandlerFIQ 函数进行处理。

在 armtrap.s 汇编文件中的 FIQHandler 入口处,调用 OEMInterruptHandlerFIQ 函数的前后应该使用 msr 汇编指令添加禁止/使能 FIQ 中断的操作。

在 arminit.c 文件的 OEMInterruptHandlerFIQ 函数中添加更新 GPIO 口输出电平、脉冲计数、加减速/恒速状态切换、定时器周期之更新等操作,如果当前发送脉冲为最后一个脉冲的下降沿,则需要关闭 FIQ 的中断使能,这样当脉冲发送完成后会产生普通 IRQ 中断来触发 OEMInterruptHandler 函数的调用,让系统通知驱动程序当前运动任务的完成。

另外,还需要实现 OEMInterruptEnable 和 OEMInterruptDisable 函数来改变处理器中断屏蔽寄存器。

2.5 驱动程序添加

驱动程序采用 WinCE 的普通流接口驱动程序,通过 KernelIoControl 接口对步进电机进行参数设置和启/停控制,在设置参数之前应该使用式(2)计算加/减速脉冲个数。

综合以上过程,采用 FIQ 中断的处理流程图如图 2 所示。

3 试验测试

文章采用 Windows CE5.0 嵌入式操作系统,结合

ARM 处理器的 FIQ 中断机制,完成了步进电机驱动程序的实现,在这种机制下,很好地克服了采用普通 IRQ 中断的驱动程序实时性能不高的问题,可以在实时性能要求较高的应用场合中应用。在试验过程中可以看到,使用本文提出的采用 FIQ 中断的驱动程序,可以平稳地输出 120 kHz 的脉冲,远远超过普通步进电机驱动器最高输入脉冲频率的 20 kHz。

本文针对采用普通 IRQ 中断的驱动程序实时性能不高的问题,实现了使用 ARM 处理器的 FIQ 中断来开发驱动程序。使用普通 IRQ 中断的驱动程序来发送脉冲时,由于中断优先级和响应时间的问题,步进电机产生剧烈振动,导致根本不能在实际应用中使用。实践证明,驱动程序使用 FIQ 中断是稳定的且能获得比 IRQ 中断更好的实时性能,合理采用可以替代某些场合中的专用芯片或 PLD 芯片,以节约成本,降低功耗。

参考文献

- [1] 卢爱臣,王剑宇,郭伟,等.基于 WinCE 自动控制台的设计与实现[J].微计算机信息,2011(2):87-88.
- [2] 微软公司.Windows CE 设备驱动程序开发指南[M].北京:北京希望电子出版社,1999.
- [3] 杜春蕾.ARM 体系结构与编程[M].北京:清华大学出版社,2003.
- [4] 何宗键.Windows CE 嵌入式系统[M].北京:北京航空航天大学出版社,2006.

(收稿日期:2012-08-20)

作者简介:

开元海,男,1979年生,中级工程师,硕士研究生,主要研究方向:嵌入式系统。

汪超,女,1977年生,讲师,硕士研究生,主要研究方向:机电一体化。

郑勇,男,1976年生,讲师,大学本科,主要研究方向:机电一体化。

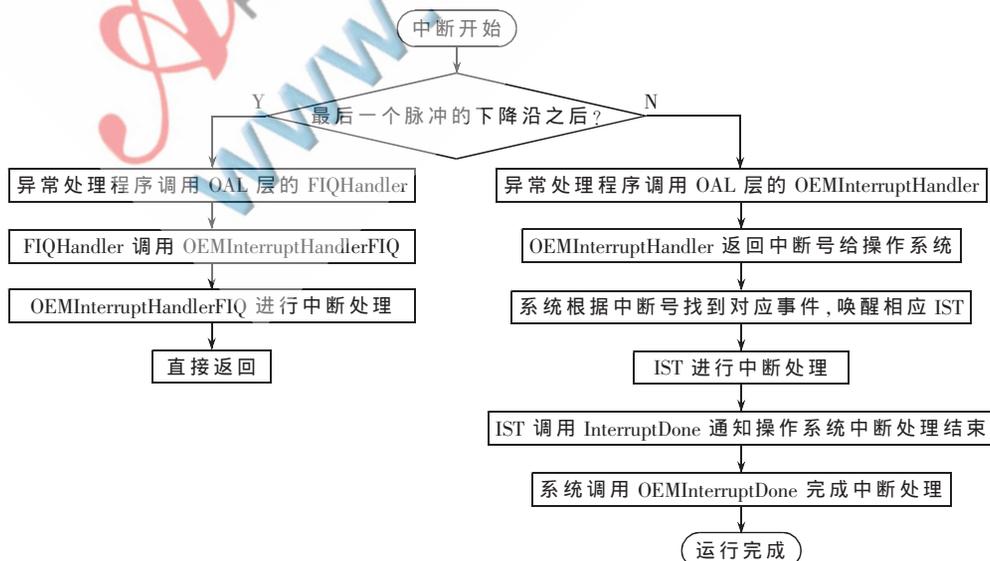


图 2 采用 FIQ 中断的处理流程图