

doi: 10.3969/j.issn.1006-1576.2010.02.029

SM502 移动多媒体协处理器在嵌入式系统中的应用

宋方伟^{1,2}, 刘勇¹, 聂诗良², 王能¹

(1. 中国兵器工业第五八研究所 军电部, 四川 绵阳 621000; 2. 西南科技大学 信息工程学院, 四川 绵阳 621010)

摘要:为解决大多数嵌入式处理器不能支持高分辨率显示或在高分辨率下显示效果不好的问题,将 SM502 应用于嵌入式系统中。简单介绍该系统的功能特性和工作原理,分析其硬、软件设计方法。SM502 与处理器的接口设计采用 Configuration C 配置方式。而 SM502 的 linux 驱动设计包括 SM502 设备驱动结构、设备驱动的数据结构。该芯片已成功应用于某单兵系统,大大增强了系统的多媒体处理能力,提高整个系统的工作效率。

关键词: SM502; 嵌入式系统; linux 设备驱动

中图分类号: TP311.52 **文献标识码:** A

Application of SM502 Mobile Multimedia Coprocessor on Embedded System

SONG Fang-wei^{1,2}, LIU Yong¹, NIE Shi-liang², WANG Neng¹

(1. Dept. of Military Electronic, No. 58 Research Institute of China Ordnance Industries, Mianyang 621000, China;

2. School of Information Technology, Southwest University of Science & Technology, Mianyang 621010, China)

Abstract: Aiming at the problem that most embedded processor can't support high-resolution display or can't work efficiently in high-resolution display, apply SM502 to embedded system. Briefly introduces the characteristic of the system and work principle, and analyzes the hardware and software design method. The interface of SM502 to processor uses Configuration C. And the design of SM502 driver includes SM502 structure of device driver, the structure device driver data. The chip has been successfully applied to a certain type individual soldier system, greatly improves the capacity of multimedia processing of the system, and achieves high efficiency.

Keywords: SM502; Embedded system; Linux device driver

0 引言

目前,很多嵌入式处理器内部都集成了 LCD 控制器,可方便地驱动 LCD 液晶显示。处理器将 LCD 显示器的数据存放于 SDRAM 中的 LCD 帧缓冲区内,由处理器中的 LCD 控制器提取 LCD 帧缓冲区中的数据并输送给 LCD 显示器。以 Samsung 公司的 S3C2410 为例,在 TFT LCD 模式下,典型的实际支持的最大分辨率为 640×480。另外,受到处理器 LCD 帧缓冲区大小的影响,在达到最大分辨率时,LCD 数据的位宽度降低,使显示效果变差。

SM502 是便携式多媒体协处理器芯片,为嵌入式工业提供视频高级功能,它具有视频和 2D 能力,并支持多种输入/输出接口,同时还带有 64 个 GPIO 引脚,便于与外部器件连接。故将 SM502 应用于嵌入式系统中,以解决嵌入式处理器不能支持高分辨率显示或在高分辨率下显示效果不好的问题。

1 SM502 硬件系统设计

1.1 SM502 与处理器的接口设计

SM502 支持 3 种不同的系统配置,其中 2 种模式是 Unified Memory Architecture (UMA) 模式,即系统内存和帧缓存共享存储器资源。由于 Silicon

Motion 公司可提供内嵌 8 M 的 SDRAM 的 SM502 的芯片,可简化采用局部内存的硬件设计线路。将这 8 M 的 SDRAM 作为帧缓存,只需配置相应的内存映射即可实现对这 8 M 内嵌 SDRAM 的访问,使系统性能大大提高。采用 Configuration C 配置方式,其接线简单,是 UMA 方式中较好的一种,该模式下局部 SDRAM 包含系统内存和帧缓存,SM502 作为从设备与主机连接,当局部 SDRAM 在 150 MHz 时钟下运行时,Configuration C 的带宽高于 Configuration B 的带宽。以 Intel 的 PXA 系列 cpu 为例,采用 Configuration C 方式配置局部内存,原理设计如图 1,其中,SM502 以总线方式接入 PXA 处理器,PXA 处理器的 GPIO[8:2]中的一个管脚作为 SM502 的中断输入;由于 SM502 可能要驱动系统的 SDRAM,所以连接到 PXA 处理器的 SDCLK1。

PXA270 与 SM502 连接模式为 PXA270 作为主机有独立的 SDRAM,SM502 除了可以访问主机内存外,还拥有 8 MB 的本地内存作为帧缓存。PXA 处理器的片选决定了 SM502 的片内 SDRAM 和 I/O 空间的地址,SM502 的帧缓存与 I/O 寄存器端口共占有 64 M 地址空间。其中 0~62 M 属于帧缓存,高 2 M 地址空间为 SM502 的 I/O 寄存器端口空间,

收稿日期: 2009-09-03; 修回日期: 2009-12-17

作者简介: 宋方伟(1983-),男,四川人,在读硕士,从事音视频处理、嵌入式系统设计研究。

这 2M 的 I/O 端口空间中每 64 KB 属于一组寄存器。

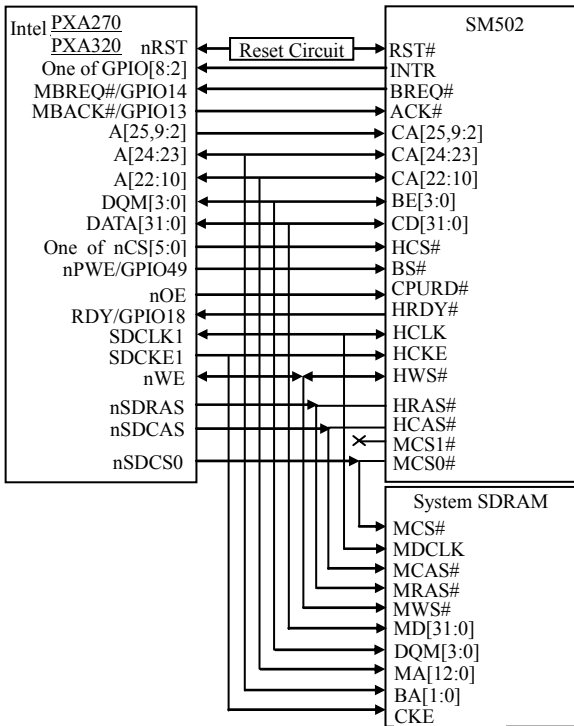


图 1 SM502 在 PXA 系统中的原理图

1.2 SM502 的 GPIO 配置管脚

SM502 的 GPIO 配置管脚 GPIO[7:0], GPIO[15:12], GPIO31, GPIO29 必须正常配置, 否则不能正常工作。其中 GPIO0 配置 SM502 以总线方式或者 PCI 方式连接处理器; GPIO[2:1]管脚配置 CPU 的架构 (Xscale, MIPS 等); GPIO[6:5]引脚配置局部内存行地址大小; GPIO[15:13]引脚决定局部内存的大小 (最小 2MB, 最大 64MB); GPIO31 和 GPIO29 决定时钟输入源。

2 SM502 的 linux 驱动设计

2.1 Sm502 设备驱动结构

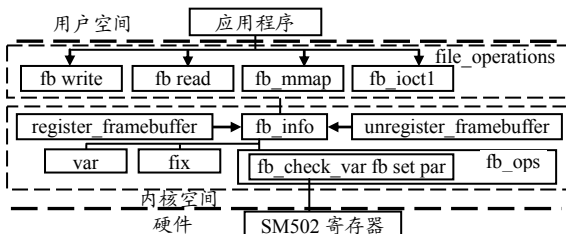


图 2 linux 下 SM502 设备驱动结构

SM502 为字符设备, 故设计成为文件层一驱动层的接口形式。图 2 所示为 SM502 驱动的主要结构。作为字符设备, SM502 帧缓冲设备的文件操作结构体 file_operations 由 /Linux/driver/video/fbmem.c 中的 file_operations 提供, 其成员函数已经实现。用户空间通过 file_operations 结构体来实现对 SM502 的

操作。特定帧缓冲设备 fb_info 结构体的注册、注销以及其中成员的维护, 尤其是 fb_ops 中成员函数的实现则由 sm502fb.c 提供且由驱动设计人员编写, 其中 fb_ops 中的成员函数最终会操作 SM502 的相关寄存器。用户空间可通过 file_operations 结构成员函数操作 fb_info 结构体来完成对 SM502 的操作。

2.2 设备驱动的数据结构

帧缓存 (Frame Buffer) 是 Linux 为图形设备提供的抽象接口, 它允许上层应用程序在图形模式下对显示缓冲区读写操作。这种操作是抽象、统一的。应用程序不必关心物理显存的位置、换页机制等细节。在 /linux/include/fb.h 中定义了帧缓冲设备的驱动接口 fb_info 结构体, 及当前显卡的工作状态。

```

struct fb_info {
    int node;
    int flags;
    struct mutex lock; /*互斥锁 */
    struct fb_var_screeninfo var; /*可设定参数 */
    struct fb_fix_screeninfo fix; /* 固定参数 */
    struct fb_monspecs monspecs; /* 显示器标准 */
    struct work_struct queue; /* 帧缓冲事件队列 */
    struct fb_pixmap pixmap; /* 图像硬件 mapper */
    struct fb_pixmap sprite; /* 光标硬件 mapper */
    struct fb_cmap cmap; /* 当前颜色表 */
    struct list_head modelist; /* mode list */
    struct fb_videomode *mode; /* 当前模式 */
    #ifdef CONFIG_FB_BACKLIGHT
    struct mutex bl_curve_mutex;
    struct backlight_device *bl_dev; /* 背光设备 */
    u8 bl_curve[FB_BACKLIGHT_LEVELS]; /*背光调节 */
    #endif
    struct fb_ops *fbops;
    struct device *device; /* This is the parent */
    struct device *dev; /* This is this fb device */
    int class_flag; /* private sysfs flags */
    #ifdef CONFIG_FB_TILEBLITTING
    struct fb_tile_ops *tileops; /* Tile Blitting */
    #endif
    char __iomem *screen_base; /* 虚拟基地址 */
    unsigned long screen_size; /* 虚拟内存大小 */
    void *pseudo_palette; /* 伪 16 色颜色表 */
    #define FBINFO_STATE_RUNNING 0
    #define FBINFO_STATE_SUSPENDED 1
    u32 state; /* 硬件状态, 如挂起 */
    void *fbcon_par;
    void *par;
};

```

该结构体中记录了帧缓冲设备的全部信息, 包括设备参数, 状态以及操作函数指针。fb_ops 为指向底层操作的函数指针, 需要驱动工程师编写。对于 SM502, 可在 fb_info 为基础定义 SM502 设备的数据结构 sm502fb_info:

(下转第 96 页)

控制部分循环执行工作：1) 调度程序从调度序列中选择一个活动来执行，如果是 STOP 知识源，则停止；2) 若选择执行知识源条件部分且条件成立，则将刺激框架、响应框架及激活的知识源振动部分指针送入调度序列；3) 若选择执行知识源动作部分，则黑板将被修改，黑板监督程序将对应于该种黑板变化类型知识源条件部分指针送入调度序列。

由于调用一个知识源可引起黑板上的多种变化，而每种变化又会激活多个知识源，故在系统运行的任一时刻，调度队列中可能会有许多等待执行的活动。对于调度队列中的每个活动，调度程序根据其刺激框架、响应框架以及黑板上的其他有关信息计算它的优先级，让优先级最高的活动首先执行。

4.4 输入/输出模块

输入/输出模块负责同飞机总线、PVI 系统以及其它辅助子系统之间的信息交换，完成数据的译码、编码。采集飞机总线上的数据作为输入，根据需要，将处理结果输出给任务规划模块、战术规划模块以及 PVI 系统，驾驶员可通过 PVI 来监控飞机的工作状态，并对一些处理结果做出最后的决策。

黑板结构具有很好的拓展性，其它几个模块也

(上接第 92 页)

```
struct sm502fb_info {
    struct device *dev;
    struct fb_info *fb;
    struct resource *fbmem_res; /*帧缓冲资源*/
    struct resource *regs_res; /*寄存器资源*/
    struct sm502_platdata_fb *pdata; /*板卡数据*/
    unsigned long pm_crt_ctrl;
    int irq; /*中断*/
    void __iomem *regs; /*重映射寄存器*/
    void __iomem *fbmem; /*重映射帧缓冲区*/
    size_t fbmem_len; /*重映射长度*/
};
```

其中，fbmem_res,regs_res,pdata,可以在设备初始化时由平台结构体 platform_device sm502_plat_dev 获得。

2.3 SM502 驱动的初始化

Linux 内核中通过 module_init(sm502_init)宏来注册设备驱动初始化函数。由于 SM502 作为一块独立的 SoC 硬件模块存在。因此，在设备初始化函数中，只注册 SM502 设备平台驱动。如下：

```
platform_driver_unregister(&sm501_plat_drv);
```

而初始化 sm502_fb 结构体的固定和可变参数、硬件初始化、申请内存空间，子平台创建等，移交给平台驱动探测函数完成。平台驱动结构体如下：

```
static struct platform_driver sm502_plat_drv = {
```

可应用黑板结构，形成几个并列的黑板结构，再采用一块总控黑板进行各个模块之间的协调和任务调度。这样，IPAS 系统的复杂任务就转化成一个个层次分明的黑板树的求解问题。

5 结束语

研究了智能驾驶员辅助决策系统的总体结构和基于黑板模型的知识组织和调度方法，并就其中的系统状态管理模块进行了深入分析。

参考文献：

- [1] Shella B. Bank and Carl S.Lizza, Pilot's Associate: A Coopertive Knowledge-based System Application[J]. IEEE Expert, 1991, 6(3):18-29.
- [2] I. Snyder, K. T.; Sokoloff, S.; Bearden, M.: The cockpit associate: A "co-pilot in a box" for general aviation[J]. Digital Avionics Systems Conference, 2003.
- [3] C.A. Miller and M. D. Hannen, The Rotorcraft Pilot's Associate: Design and Evaluation of an Intelligent User Interface for Cockpit Information Management[J]. Elsevier, 1999: 443-455.
- [4] Friedlander C. The Pilot's Associate: A Forum for the Integration of Knowledge based Systems and Avionics[C]//. Aerospace and Electronics Conference, 1988.
- [5] 邢传鼎, 杨家明, 等. 人工智能原理及应用[M]. 上海: 东华大学出版社, 2005: 2.

```
.driver = {
    .name = "sm502",
    .owner = THIS_MODULE,
},
.probe = sm502_plat_probe,
};
```

sm502_plat_dev 描述设备的硬件资源，将其作为参数传递给平台驱动探测函数。内核启动便会调用 sm502_plat_probe 完成 SM502 的初始化。

另外，除了作为帧缓冲设备，SM502 还有 2 个串口、1 个 AC97、1 个主 USB 等控制器。故如果要使用这些控制器，可将其作为 SM502 的子平台，采用子平台用 platform_device 表示，子平台驱动用 platform_driver 注册，并在 SM502 的平台探测函数中，创建子设备平台，最基本的子平台就是帧缓冲子平台。

3 结束语

SM502 已成功应用于某单兵系统，系统的多媒体处理能力增强，提高了整个系统的工作效率。

参考文献：

- [1] Silicon Motion®, Inc. SM502 Mobile Multimedia Companion Chip Data book[S]. Taiwan: Silicon Motion®,Inc, 2006(12).
- [2] 宋宝华. Linux 设备驱动详解[M]. 北京: 清华大学出版社, 2008: 478-485.