

手持式终端设备中电源监测技术研究与实现

满梦华, 原 亮, 丁国良, 巨政权

(军械工程学院 计算机工程系, 河北 石家庄 050003)

摘 要:根据嵌入式系统内置电源维护方式简化、降低制造和使用成本的具体要求, 详细介绍了基于串连式可充电电池组的电源监测技术。在完成监测系统原理分析及电路设计的基础上, 进行了具体的电路搭制、调试和 Linux 环境下驱动程序编写, 并结合相应的显示需求而进一步实现了 Qt/Embedded 应用程序编写和植入 Qtopia 的方法。实现了手持式嵌入式系统中的电源的精确可视化监测。

关键词:嵌入式系统; Linux; 电源监测; 驱动程序; Qt/Embedded; Qtopia

中图分类号: TP303.3

文献标识码: A

文章编号: 1673-629X(2009)01-0237-04

Study and Implementation of Power - Monitoring Technology for Hand - Hold Terminal

MAN Meng-hua, YUAN Liang, DING Guo-liang, JU Zheng-quan

(Department of Computer, Ordnance Engineering College, Shijiazhuang 050003, China)

Abstract: Based on the requirement of simplifying the maintenance of battery - power system inside the embedded system and reducing the relative expenses on it, introduces the power monitoring method of series - connected rechargeable battery pack in detail. The principle of monitoring system is carefully studied, and the corresponding circuits are implemented as well. Then, the relative Linux drivers are discussed, and written according to the demand of displaying the windows on LCD, which shown the working status of batteries, by using the Qt/Embedded application program and mounted it into Qtopia. Implemented the power - monitoring technology for hand - hold embedded system.

Key words: embedded system; Linux; power - monitoring; driver; Qt/Embedded; qtopia

0 引 言

随着数字信息技术和网络技术的高速发展, 各种嵌入式系统已成为了市场的新焦点。鉴于嵌入式系统是对功能、可靠性、成本、体积和功耗等有严格要求的专用计算机系统, 因此降低其系统功耗、提高内置电源持续工作的能力就成为一项重要的研究内容。

与笔记本电脑类似, 嵌入式系统通常配备一个由多只电池串联组成的整体式电池包^[1]。当经过反复充电使用而其参数不再满足整个系统要求时, 必须更换整个电池包。一般说来, 由于电池包内部各个电池之间的电气参数略有差异, 很难使得每个单体都得到平衡、充分的放电, 进而导致在电池组充电的过程中, 电压过高的电池芯可能提早触发电池组过充电保

护, 而在放电过程中电压过低的电池芯又可能首先导致电池组的过放电保护, 从而使电池组的整体容量明显下降, 使得整个电池组的实际容量常为电池组中性能最差的电池所限。

针对上述问题, 目前常用两种解决方法:

①研究或选用新型电池, 尽量减小个体差异并从整体上提高电池的额定容量, 例如由最初的镍镉、镍氢电池发展到现在主流的锂电池;

②提高电池芯的利用率, 即利用监测技术探索每个电池芯的工作状况。若有部分电池芯过早地放电结束或者出现问题时, 能够自动检测并予以提示, 此时只需进行个别更换即可同样保证系统正常工作。

对于普通用户而言, 难以直接涉足前者所属的相关领域, 而后者所述方式能够较为方便、容易地得到实现。由此一来, 可望较好地解决前述问题, 同时避免了电池组整体更换时其中的正常电池也连带报废的现象, 进而提高电池利用率。此外, 尚可解决专用电池成本较高、购买不易的问题, 以便有效降低系统的维护成本。

收稿日期: 2008-05-25

基金项目: 国家 863 计划 (2007AA01Z454); 国家自然科学基金项目 (60471022)

作者简介: 满梦华 (1984-), 男, 硕士研究生, 研究方向为智能检测与诊断; 原 亮, 教授, 研究方向为智能检测与诊断。

1 监测系统工作原理与电路设计

结合一个具体的手持/嵌入式超级终端系统^[2],文中对其内置电源的监测模式进行了较为详尽的研究。因其使用通用的五号可充电电池,所以可按第二方法考虑解决电池的使用和更换问题,进而基于 Linux 操作系统较好地实现了相应的“电池管理与更换提示”功能。

1.1 监测系统工作原理

本系统采用 12 节额定电压为 1.2V 的镍氢电池串联供电,并通过专门设定的 12 个监测点逐一监测。监测点上的模拟电压值通过多路开关和辅助电路输入 ARM 芯片 S3C2410x 自带的 AD 转换器,再经 Linux 驱动程序和 GPIO 端口^[3]读取相应监测点的电压值,传递给数据处理和图形显示等应用程序,从而实现了电源的监测。具体方法如图 1 所示。

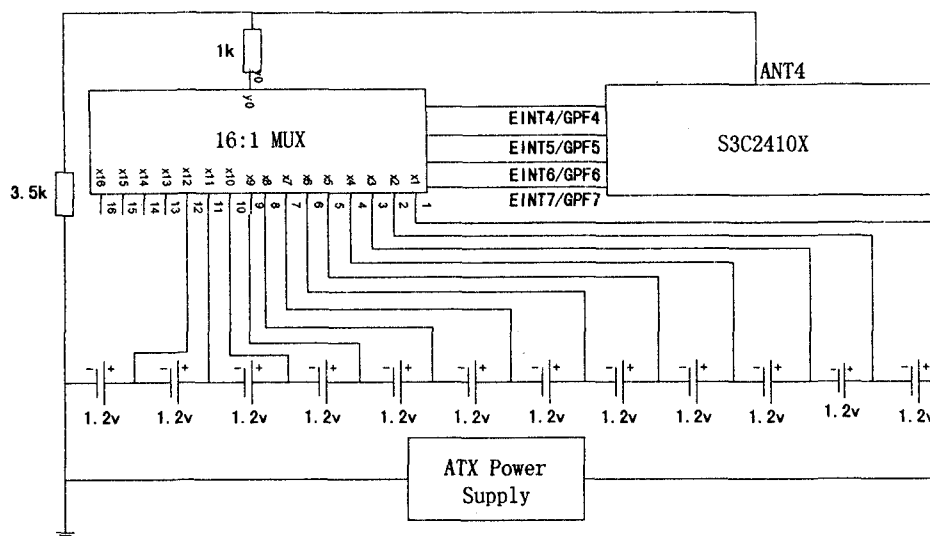


图 1 电源监测系统电路图

1.2 AD 转换与辅助电路

监测系统利用 S3C2410x 自带的 AD 转换器将模拟电压输入转换为应用程序所需要的数据。芯片自带 AD 转换器为 8 路模拟输入、10bit 数字输出,最大转换率为 500ksps,转换时钟为 2.5MHz,模拟输入范围为 0~3.3V。本系统所需要监测的电压范围为 0~14.4V,针对这种情况,本系统设计了分压电路,在应用程序中只需要利用比例公式就能计算出各个监测点的实际电压值。并且,考虑到监测精确度和电路功耗的因素,把分压电路的总电阻定为 4.5kΩ。这样一来,对应的模拟输入电压值范围为 0.26~3.2V,以便充分利用的 AD 转换器的输入带宽。此时电路的耗损功率仅为 0.04608W。

1.3 控制电路

利用 S3C2410x 的 GPIO 端口控制选通多路门开

关电路,监测系统可以随机地提取不同监测点的电压值。选通逻辑通过译码实现,如表 1 所示。

表 1 选通逻辑

GPF7:GPF6:GPF5:GPF4	功能选择
0:0:0:1	监测点 1 输入
0:0:1:0	监测点 2 输入
0:0:1:1	监测点 3 输入
0:1:0:0	监测点 4 输入
0:1:0:1	监测点 5 输入
0:1:1:0	监测点 6 输入
0:1:1:1	监测点 7 输入
1:0:0:0	监测点 8 输入
1:0:0:1	监测点 9 输入
1:0:1:0	监测点 10 输入
1:0:1:1	监测点 11 输入
1:1:0:0	监测点 12 输入
其它情况	省电模式

2 电源监测系统驱动程序设计

鉴于文中所述的超级终端系在 Linux 环境下运行,各个检测环节的驱动程序和相关的实施策略、编写方法亦多有特点,现详细描述如下。

2.1 Linux 驱动程序工作原理

在 Linux 内核中,设备驱动程序是作为文件系统的一个模块存在的^[4],它对下负责与硬件设备的

交互,对上通过一个通用的接口挂接到文件系统中,从而和系统的内核等联系起来。它是软件概念和硬件设备间的一个抽象层。系统调用是内核和应用程序之间的接口,而设备驱动程序则是内核和外设之间的接口。设备驱动程序为应用程序屏蔽了外设硬件的细节,从应用程序角度来看,对于外设的操作方法与普通文件一样。设备驱动程序与系统外设关系如图 2 所示。

2.2 字符设备

字符设备是 Linux 系统中最简单的设备,可以像文件一样访问。当字符设备初始化的时候,其驱动程序向 Linux 核心登记,在 chrdevs 向量表中增加一个 device_struct 数据结构条目。这个设备的主设备标识符用作这个向量表的索引。一个设备的主设备标识符是固定的。chrdevs 向量表中的 device_struct 数据结构包括一个登记设备驱动程序名称的指针和一个指向一

组文件操作的指针。这组文件操作本身位于这个设备的字符设备驱动程序中,并处理一些特定任务。本电池监测系统在Linux下的驱动程序就把外设视为字符设备,在驱动程序里实现了打开、读、写和关闭等操作。

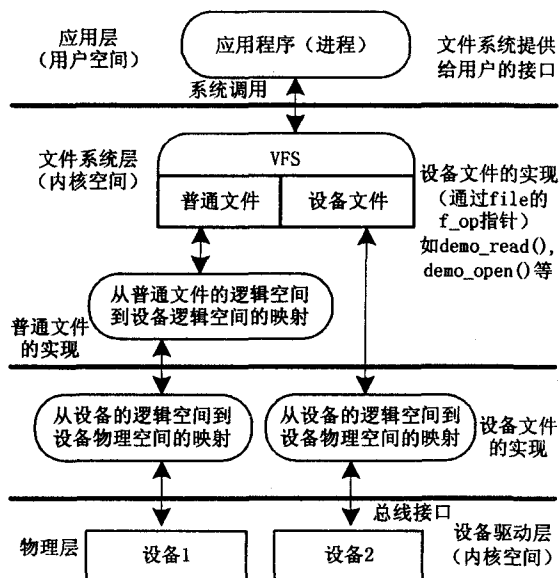


图2 Linux 驱动程序与系统外设关系图

2.3 电池监测驱动实现策略

系统内部,I/O设备的存取使通过一组固定的入口点来进行的。这组入口点由特定设备的设备驱动程序提供,在数据结构file_operations()中定义。该程序定义了一个数据结构为file_operations()的变量adc_fops。

```
static struct file_operations adc_fops = {
    owner: THIS_MODULE,
    open: adc_open,
    read: adc_read,
    ioctl: adc_ioctl,
    close: adc_close,
}
```

2.3.1 adc_open 函数

此函数打开并初始化设备以待I/O操作。adc_open()子程序必须对其做好必要的准备,如果设备是独占的,adc_open()子程序必须设置一些标志以表示设备处于忙状态。

2.3.2 adc_read 函数

对AD转换器设备文件进行读操作时调用adc_read()子程序。电池监测系统要求实时读取外设监测点的电压值。因此,在adc_read()函数中实时触发AD转换器开始工作,并且利用程序轮询方式读取转换后的数据。adc_read()函数内容如下:

```
static int adc_read (struct file * file, char * buffer, size_t count,
    loff_t * ppos)
```

```
{
    ADCCON = PRESCALE_ENDIS | PRSCVL (40) | ADC_INPUT (AIN5); //初始化 AD 转换控制寄存器
    ADCCON |= ADC_START; //触发 AD 转换器开始转换
    while (! (ADCCON & 0x8000)); //程序轮询,确保转换结束
    key = (ADCDAT0 & 0x03ff); //转换结束后,读取数据寄存器的数据
    copy_to_user (buffer, &key, sizeof key); //把数据传到用户空间,返回给应用程序
    Return 1;
}
```

2.3.3 adc_ioctl 函数

控制外围的辅助电路时调用adc_ioctl()函数。对照监测点逻辑控制表,从S3C2410x的GPIO端口发出选择指令,接着就可以进行读操作以读取相应的数值。在不需要监测时,控制监测系统进入省电模式。

```
Static int adc_ioctl (struct inode * inode, struct file * file, unsigned int chanel, unsigned long arg)
```

```
{
    if ((chanel >= 0) && (chanel < 16)) //chanel 是由应用程序传递给驱动的参数,要求选通的监测点
    {
        for (i = 0; i < 4; i++)
        {
            set_gpio_ctrl(adc_table[i] | GPIO_PULLUP_EN | GPIO_MODE_OUT); //初始化 GPIO 端口
            write_gpio_bit(adc_table[i], chanel & i); //连通所需要的监测点的电路开关
        }
        return 1;
    }
    else { return -EINVAL; }
}
```

2.3.4 adc_close() 函数

adc_close()函数用来关闭外设。当监测结束时,应用程序会调用此函数关闭设备文件。同adc_open()函数一样,也可以为NULL。

3 设计 Qt/Embedded 应用程序和植入 Qtopia 的方法

QT/Embedded为嵌入式Linux提供了一种友好交互接口,是完整的自包含C++ GUI和基于Linux的嵌入式平台开发工具^[5],专为高端嵌入式图形领域应用而设并得以普及。鉴于文中所使用的系统需要按照手动点击检查和欠压自动报警的方式工作,所以,必须遵循相应的规则和方法,进行高效、美观的图形界面设计。

3.1 编写 Qt/Embedded 应用程序

在宿主机上编译 Qt for X11 时生成的 Qt Designer 软件是一个非常流行的快速应用程序开发工具,用来设计界面和编制代码。在 Qt Designer 中,可以通过拖拉或点击的方式,在一张空白表单的适当位置上添加一些输入框和按钮等窗口组件。这时 Designer 工具会自动编写和维护代码。

使用 Qt Designer 进行 C++ 程序编制的基本步骤是:首先建立窗体,并根据应用的需要在窗体中添加控件。Qt 会将建立的窗体保存为 .ui 文件,使用 Qt 提供的 uic 工具将文件转换为 .h 和 .cpp 文件,对于控制动作是需要手动添加不同的操作函数。之后使用 progen 工具为该应用程序建立 .pro 工程文件,并通过 tmake 工具为该工程建立 Makefile 文件。最后,只需要运行 make 即可生成可执行文件。

在编写操作函数时,本系统通过计算相邻监测点的观测值确定每个电池芯的电压值。通过查询选用电池芯的放电结束电压查找表^[6],得出每个电池芯的剩余电量。当电池组电量降至一定阈值或某个电池芯的储电量低于 3% 时^[7],利用应用程序报警,通知用户予以充电或按照指示的电池编号及时进行个别更换。这样,既可保证电源的正常工作指标,又能够提高所有电池的利用率。

3.2 将应用程序植入 Qtopia

在 Qtopia 平台上发布自己的应用程序,需要三个文件:一个执行文件、一个启动器文件、一个图标文件。执行文件就是前面讲到编写并编译生成的可执行文件,需要将该可执行文件保存在 qtopia/bin 目录下。图标文件就是为应用程序制作 48 * 48 大小的 PNG 格式的小图像,它一般存放在 qtopia/pics 目录下。同时还需要建立应用启动器(.desktop)文件,把它保存在

qtopia/apps/Applications 目录下。将上述文件分别复制好以后,重新运行 Qtopia,就可以看到添加的应用图标,点击此图标便可运行该应用程序了。

4 结束语

降低系统能耗需要嵌入式系统硬件环节提供可靠支持,同时也离不开嵌入式操作系统和应用程序对硬件资源的合理管理^[8]。现有的 Linux 提供了电源管理的部分功能,但由于具体进行嵌入式系统设计时的灵活性,一般需要根据特定需要而定制应用程序对电源电量进行精确的监测与提示。通过对手持式终端设备中电源监测技术的研究和实现,将有助于其它类型嵌入式系统的电源监测、管理工作的完成,亦有望对类似系统的建立具有一定的参考和借鉴价值。

参考文献:

- [1] 朱玉军,王香凤. 笔记本电脑电池简介[J]. 化学教学, 2006(4):30-32.
- [2] 原亮,王盼卿,陈立云,等. 装备信息化工作中嵌入式系统的研究实现[J]. 机械工程学院学报, 2007, 8(1):79-81.
- [3] 潘巨龙,黄宁. Arm 嵌入式 Linux 系统构建与应用[M]. 北京:北京航空航天大学出版社, 2006.
- [4] Bovet D P, Cesati M. 深入理解 Linux 内核[M]. 北京:中国电力出版社, 2004.
- [5] 倪继利. Qt 及 Linux 操作系统窗口设计[M]. 北京:电子工业出版社, 2006.
- [6] 陈金舟. 储氢合金测试系统及镍氢电池管理系统的设计[D]. 北京:中国科学院研究生院, 2004.
- [7] 华宏懿. 镍氢蓄电池的数学建模及其电池管理系统实现[D]. 北京:北方工业大学, 2006.
- [8] 阳富民,梁晶,张杰,等. 嵌入式 Linux 电源管理技术的研究与实现[J]. 计算机工程与科学, 2004, 26(12):92-93.
- [9] 侯文国,傅秀芬,谢翠萍. 网格的数据挖掘[J]. 计算机应用研究, 2004(10):241-247.
- [10] HAN Jiawei, Kamber M. Data Mining Concept and Techniques [M]. [s.l.]:Morgan Kaufmann Publishers Inc, 2001.
- [11] 顾宁,刘家茂,柴晓路. Web Services 原理与研发实践[M]. 北京:机械工业出版社, 2006.
- [12] 邱莉榕,史忠植,林芬,等. 基于主体语义 Web 服务自动组合研究[J]. 计算机研究与发展, 2007, 44(4):643-650.
- [13] 邹力鹏. 数据挖掘方法本体研究[J]. 计算机科学, 2005, 32(3):197-199.
- [14] 张佩云,孙亚民. 动态 Web 服务组合研究[J]. 计算机科学, 2007, 34(5):4-7.
- [15] 袁琴,杨小虎. 基于本体分类的 Web 服务合成的研究及应用[J]. 计算机工程, 2007, 33(2):79-81.
- [16] 李曼,王大治,杜小勇,等. 基于领域本体的 Web 服务动态组合[J]. 计算机学报, 2005, 28(4):644-650.
- [17] 蔡刚. 基于网格的分布式数据挖掘体系结构研究[D]. 重庆:重庆大学, 2007.7.
- [18] Box D. Simple object Access Protocol(SOAP) 1.1[EB/OL]. 2001. <http://www.w-3.org/TR/SOAP/>.
- [19] 侯敬军,曾致远,向凌. 一种基于 Web 服务的分布式数据挖掘体系结构[J]. 微机发展, 2004, 14(6):48-51.
- [20] 吴小竹. 利用 Web service 技术构建开放式数据挖掘系统[J]. 计算机工程与设计, 2007, 28(15):3563-3565.

(上接第 236 页)

buted System Integration[J]. IEEE Computer, 2002, 35(6): 37-46.