

航空领域嵌入式测控系统 自动生成平台*

Embedded Control System Automatic Generation Platform in Aviation

北京航空航天大学自动化科学与电气工程学院 袁海文 弭寒光 李晓光



袁海文

北京电力电子学会理事,中国航空学会仪表与测试分会主任委员,仪器仪表学会电子测量与仪器分会委员。现为北京航空航天大学检测技术与自动化装置学科责任教授,博士生导师。2002年获北京市科技新星计划资助,2003年获北京市优秀青年工程师标兵称号。累计负责完成各类横、纵项科技项目58项。2005年以来累计获得科技奖励9项(国家科技进步二等奖1项、省部级5项、校级3项)。

嵌入式测控系统在航空领域的应用越来越广泛,已经成为飞行器诸多子系统功能设计与实现的核心支

*航空基金资助项目(2008ZD51060)。

基于 ARM-Linux 的试验平台能够满足航电测控中绝大部分要求,并可以按照用户的实际需求进行软硬件功能的裁剪,增强了平台的适应性,同时也体现了嵌入式系统的快速定制性与特定功能的自由选择相结合的新型设计思想。

撑技术。

嵌入式系统(Embedded System)是指嵌入于各种设备及应用产品内部的计算机系统,它的体积小,结构紧凑,可作为一个部件埋藏于装置中,主要用于完成信号采集、传输和控制,包括提供用户接口、管理有关信息的输入输出、监控设备工作,使设备及应用系统有较高智能和性价比。

嵌入式系统功耗低,性价比高,便于携带,非常适用于测控领域;同时,随着嵌入式系统硬件复杂性的增加以及对系统功能、性能需求的提高,嵌入式测控系统本身的开发工作量骤增,占到全部产品开发工作量的70%以上。不同于通用的软件开发技术,嵌入式系统软件与硬件紧密相关。由于每个嵌入式系统硬件平台各不相同,导致每次应用软件开发都需要从硬件底层开始,并依赖于开

发人员熟悉底层硬件的程度,这种模式不仅导致了大量重复的编码工作,严重影响开发效率,而且系统的可靠性、稳定性、扩展性和可维护性也受到影响,使得嵌入式软件开发成为制约嵌入式系统应用的瓶颈。另一方面,由于不同行业开发人员在自己的领域有专长,考虑到成本以及系统开发周期等因素的影响,他们不太希望学习嵌入式系统的开发技术。因而如何将各种应用需求和硬件环境抽象出来,构建一个嵌入式开发平台已成为迫切需要解决的问题。

研究嵌入式测控系统自动生成技术,建立最终生成目标的体系结构尤为重要。作为系统设计与实现的抽象,体系结构明确了系统实现的约束条件,制约了系统的质量属性,同时为实现框架和构件的共享与重用、生成嵌入式测控软件提供了理论依

据,能够使生成的目标软件易于设计、开发、更新与维护。到目前为止,航空测控领域的嵌入式软件还没有普遍认同的体系结构模型。建立支撑嵌入式硬件开发的硬件体系结构是具有挑战意义的研究工作,它将为嵌入式测控系统的自动生成技术应用奠定基础。

一种嵌入式测控系统自动生成平台的研制

考虑到 ARM 和 Linux 分别在硬件和软件方面的优势,我们以此为基础,设计了一个通用的测控系统软硬件平台,它具有各种各样的通用嵌入式测控资源,资源开发出来后,及时采用 ATML 方式将其描述建模;然后,以该平台为基础,可以自动综合各种基于计算机实际测控系统,从而实现了一定程度的嵌入式测控系统自动生成,大大提高了设计效率,并减小了设计错误。下面介绍该支撑平台的软硬件构成及其开发方式。

1 ARM 硬件平台设计

(1) 硬件。

作为测控系统平台,既要求功能全面,能够进行扩充与缩减,同时又要求具有稳定的性能,满足一定的实时性要求。根据测控系统的需求特点,选择 AT91RM9200 处理器作为主控芯片,扩展了 32M 的 SDRAM、4M 的 NOR Flash 和 64M 的 NAND Flash,增加 SPI、RS485、RS232、CAN 总线、以太网、以及数字 I/O 等接口。另外,平台具有无线通信、键盘输入以及液晶显示功能,如图 1 所示。

AT91RM9200 内部自带 16k 的 SRAM,由于测控系统功能要求全面,在复杂的应用程序或是嵌入操作系统的情况下,16k 的空间是远远不够的,因此扩展 32M 外部 SDRAM 作为程序的运行空间。而内部 16kb 的 SRAM 被配置为堆栈区以提高系统性能。平台扩充的 NOR Flash

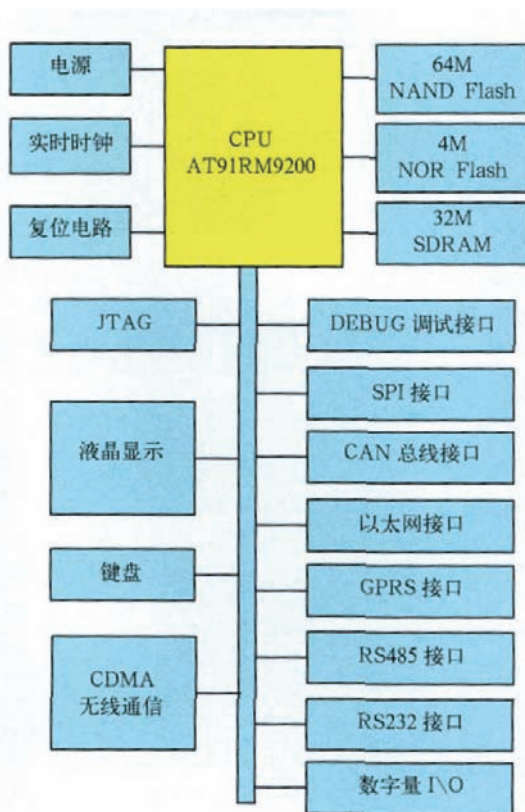


图1 硬件平台功能框图

内部存放启动代码、Linux 内核、用户程序等,这些资源要求掉电保存。测控系统有时需要存储记录大量的参数以及测量数据,因此,平台扩充 64M 的 NAND Flash 用于存储数据,并能为进一步建立嵌入式数据库提供硬件支持。

平台人机交互支持液晶显示与键盘操作的形式。使用的液晶显示屏为 240×128 点的蓝屏液晶,标准工作电压为 5V,与 ARM 通过 20 根线相连。除了电源、地线以及 8 根并行数据传输线以外,重要的控制线还有 LCD 驱动器的电源、LED 背光电源、数据/命令选择线、读/写信号选择线、片选线、忙等待信号线等。另外,采用电压转换器 74LS164245 匹配液晶工作电压与 AT91RM9200 的 I/O

O 口电压。键盘选用标准的 4×4 薄膜按键,也通过 74LS164245 隔离与 AT91RM9200 的 I/O 口相连,防止 AT91RM9200 的 I/O 口在外部的电压冲击下损坏。

(2) 抗干扰设计。

测控系统有时面临恶劣的电磁工作环境,而 ARM 主控制器模块电路的工作频率达到 180MHz,因此电路的抗干扰性是必须考虑的问题。干扰一般是以随机出现的脉冲形式进入系统的,主要由于电网上和系统内负载突变形成的交变磁场和电场耦合到系统内部,形成尖峰脉冲电势或电流叠加到某些芯片的数据线、地址线或控制线上,从而扰乱输入

信号,破坏某些器件的工作状态。

控制平台在开关量输入输出通道中都采用光电隔离措施,有效地防止干扰从过程通道进入处理器;在一些过程通道中使用了磁珠,用于抑制信号线、电源线上的高频噪声和尖峰干扰,吸收静电脉冲,在 RS485 接口处使用了瞬态电压抑制二极管 (TVS 管),抑制瞬态干扰。另外,在元件布局时,除基本的抗干扰措施外,配置了去耦电容,经过实际应用

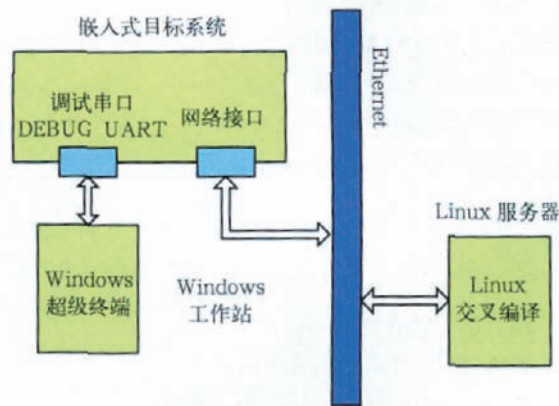


图2 软件开发平台结构图

证实,该平台具有较好的抗干扰能力。

2 建立嵌入式 Linux 开发平台

测控系统要求控制平台性能高效、稳定,功能强大,并且可定制性强。综合功能、性能与开发成本的考虑,选择嵌入式 Linux 作为控制平台的操作系统。嵌入式 Linux 具有明显的优点,在开发过程中,主要利用它的稳定性、源代码公开、支持多种文件系统以及网络通信机制的完善等性能。由于嵌入式系统硬件资源的限制,嵌入式 Linux 没有安装标准的桌面以及应用程序开发工具,所以控制平台硬件需要的相应驱动程序以及系统应用软件的开发采用在标准 Linux 桌面系统中进行编写、交叉编译,再把可执行文件下载到嵌入式 Linux 中执行的方法。整个软件开发平台结构见图 2。

(1) Linux 内核裁剪。

Linux 强调效能优先,是一种单核的操作系统,系统内核相对于嵌入式操作系统而言过于庞大,但是 Linux 提出了模块化思想,它允许用户根据自己的需求进行功能模块的裁剪。一个小型的嵌入式 Linux 系统只需要引导程序、Linux 微内核(由内存管理、进程管理和事务处理构成)、初始化进程 3 个基本元素。根据电力测控系统的要求,在嵌入式控制平台的 Linux 系统中,添加了 PPP 协议(支持用于 CDMA 无线通信)、DHCP(支持用于系统 IP 的自动获取)以及平台需要的 Linux 系统自带的硬件驱动等,删除了无用的服务,增强系统的性能以及安全性。经过裁剪的内核建立系统依存关系,编译内核之后下载到 ARM 中即可运行。

(2) 文件系统选择与加载。

由于测控系统需要记录大量的测量数据和系统设置参数,所以在扩充 Flash 的基础上,稳定、高效的文件系统必不可少。平台在 NOR

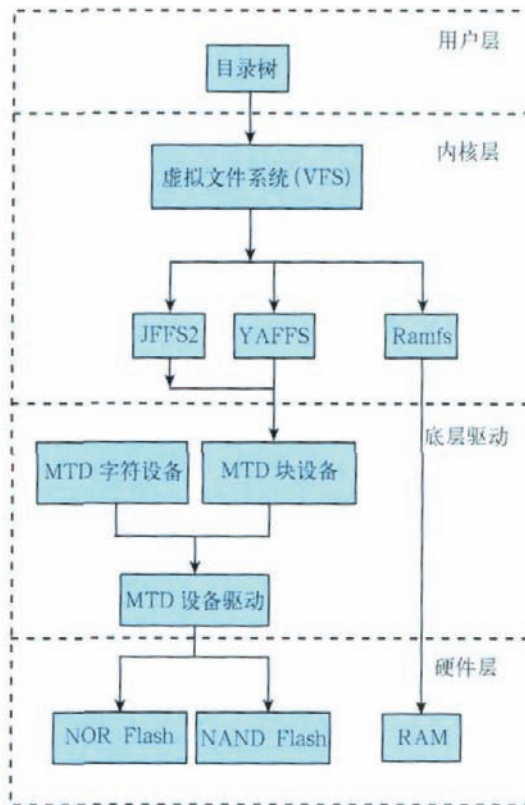


图3 Linux文件系统层次图

个抽象的接口,这个抽象接口包括 2 种: MTD 字符设备和 MTD 块设备。通过这 2 个接口可以像读写普通文件一样对 Flash 存储设备进行读写操作。Linux 下的文件系统层次结构见图 3。

控制平台有 4M NOR Flash,加载了 JFFS2 文件系统,用于存储用户应用软件。JFFS2 是一种日志文件系统,在改写文件时并不覆盖原节点的数据,而是将原节点废弃,在别处写入新的文件,并使用垃圾回收机制来收集那些废弃的节点,这种算法兼顾损耗平衡,减少了开销,提高了对 Flash 的利用率。另外,在 64M 的 NAND Flash 上加载了 YAFFS 文件系统,用于存储

Flash 和 NAND Flash 上都实现了文件系统支持。

在 Linux 中,文件系统通过 MTD (Memory Technology Device) 来隐藏对 Flash 存储器操作的细节,文件系统都运行在 MTD 之上。MTD 相关的源代码位于 /drivers/mtd 子目录下,其中 NOR Flash 存储器的驱动位于 chips 目录下,NAND Flash 存储器的驱动位于 nand 目录下。由于 MTD 驱动程序是专门为 Flash 存储器设计的,所以它们能更好地支持、管理基于扇区的擦除和读写操作的接口。MTD 在硬件和上层文件系统之间提供了一

存储测量数据和系统参数,并为扩充数据库提供文件系统支持。YAFFS 文件系统是专为 NAND Flash 设计的,也是一种日志型文件系统,与 JFFS2 文件系统相比,它最大程度地减小了 RAM 的负担和启动的时间,同时通过日志策略,也增强了文件系统的健壮性。并且 YAFFS 充分利用了 NAND Flash 每页 16 个字节的额外数据来判断坏块、作为数据的错误校验等,大大提高了效率。

3 Linux 下驱动程序开发

嵌入式系统开发与普通单片机开发相比,最明显的区别在于:嵌入式开发是在嵌入式操作系统平台上

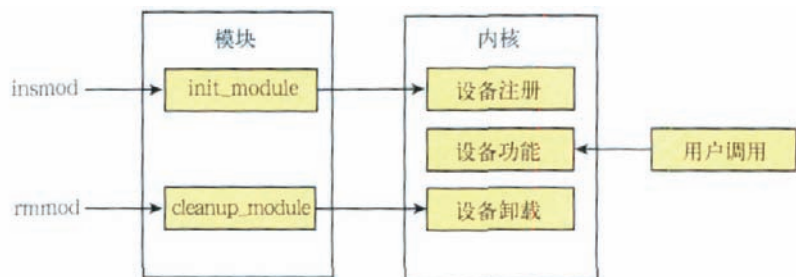


图4 字符型设备程序开发流程

进行应用软件的开发,而应用软件开发并不需要关心硬件资源。硬件资源的使用是由操作系统下驱动程序实现的,在Linux下,系统调用是操作系统内核和应用程序之间的接口,设备驱动程序是操作系统内核和机器硬件之间的接口。设备驱动程序为应用程序屏蔽了硬件的细节,这样,在应用程序看来,硬件设备只是一个设备文件,应用软件开发可以像操作普通文件一样对硬件设备进行操作,大大减轻了应用软件开发的工作量;并且基于模块化思想,便于进行整个系统的裁剪与功能的增加。驱动程序主要完成以下功能:

- (1) 对设备初始化和释放;
- (2) 把数据从内核传送到硬件和从硬件读取数据;
- (3) 读取应用程序传送给设备文件的数据和回送应用程序请求的数据;
- (4) 检测和处理设备出现的错误。

Linux下驱动程序的作用在于提供机制(Mechanism),而不是提供策略(Policy)。根据设备的不同分为3类主要的设备文件类型:字符设备、块设备和网络设备文件。设备驱动程序主要由以下部分组成:

- (1) 设备注册 register_chrdev;

- (2) 设备解除注册 unregister_chrdev;
- (3) 打开设备 open;
- (4) 释放设备 release;
- (5) 读写设备 read/write;
- (6) 获取内存 malloc。

由于驱动程序运行在内核空间,应用程序对驱动的读写操作在用户空间实现,不同运行空间不可见。读写设备 read/write 调用系统函数:

- copy_to_user(): 从内核空间拷贝数据到用户空间,设备读。
- copy_from_user(): 从用户空间拷贝数据到内存空间,设备写。

不同的设备类型开发流程不尽相同,常用的字符型设备驱动(如键盘驱动)开发流程如图4所示。

本试验系统适用于单处理器的单板系统,单板多处理芯片和多版配合情况暂未考虑,待后续工作实现扩充。

按照上述思路设计了一个多功能电力测控系统终端,结构如图6所示,

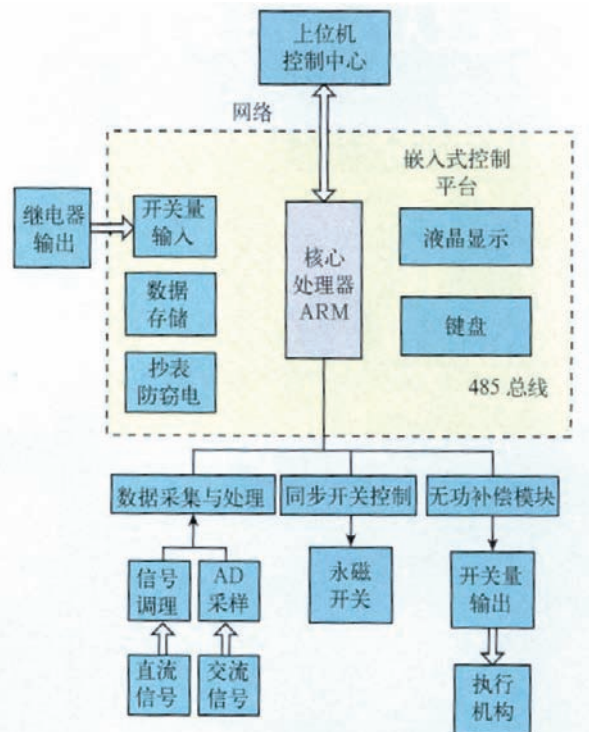


图6 电力测控装置整体结构图

航空电源嵌入式测控系统自动定制过程介绍

有了上述平台,通过将软硬件资源分层模型化,并将设计过程软件化,实现产品对象功能与嵌入式测控平台具体结构功能之间的映射,通过程序引导,就可以实现嵌入式测控系统设计方案的自动定制。具体对应过程如图5所示。

图5中,左侧“物理层”、“连接层”、“功能层”、“处理器层”、“虚拟层”、“构架层”、“代码层”对应嵌入式系统硬件部分;“数据库层”、“对象层”、“组件层”、“服务层”、“工作流层”、“语义层”、“功能层”对应位于集成开发平台、支持嵌入式测控系统产品设计实现方案自动生成的软件开发环境体系结构。

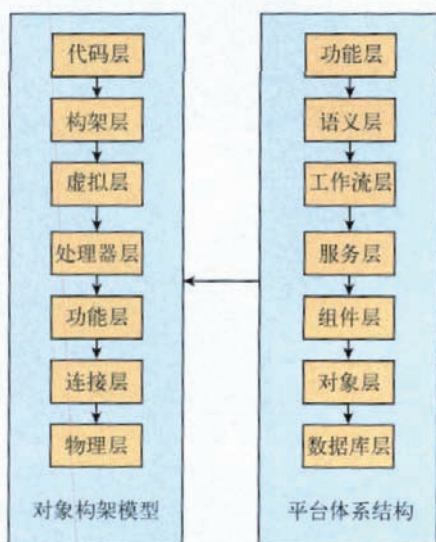


图5 嵌入式测控系统自动定制体系结构对应示意图

设计效率和目标系统的稳定性远高于传统的设计模式。

结束语

基于ARM-Linux的试验平台能够满足航电测控中绝大部分要求,并可以按照用户的实际需求进行软硬件功能的裁剪,增强了平台的适应性,同时也体现了嵌入式系统的快速定制性与特定功能的自由选择相结合的新型设计思想。通过适应具体的功能需求,此嵌入式测控平台可以在航空电源系统测控领域发挥重要作用,同时平台研制过程中的思想也体现了嵌入式系统的一个发展方向,即嵌入式系统功能与组件的快速定制,使之成为某个具体领域的基础平台,以适应不同的应用方向。

(责编 淡蓝)