

# 一种嵌入式系统驱动架构的分析及实现

杜 旭 顿新平 黄 建

(华中科技大学电子与信息工程系,武汉 430074)

E-mail: dxunxinping@163.com

**摘 要** 建立了设备驱动与嵌入式系统的一种关系模型,分析和抽象出了一种基于该模型的设备驱动架构设计方案,并给出了一例实现。重点叙述了各模块的功能和该驱动架构的关键技术。

**关键词** 驱动架构 嵌入式系统 关键技术

文章编号 1002-8331-(2004)25-0116-04 文献标识码 A 中图分类号 TP311

## Analysis and Realization of a Device Driver Frame Based on Embedded System

Du Xu Dun Xinping Huang Jian

(Electronics and Information Engineering Department, Huazhong University of Science and Technology, Wuhan 430074)

**Abstract:** A relationship model of device driver and embedded system is established in this paper a design for frame of device driver in embedded system based on this model is presented and analyzed and then a realization example is given. Each module function and the key technology of the design are described respectively.

**Keywords:** device driver frame embedded system key technology

### 1 引言

嵌入式系统是以应用为中心,以计算机技术为基础,能够满足应用对功能、性能、体积、成本、功耗等方面要求的专用系统<sup>[1]</sup>。在开发一个完整的、能实际运行的系统时,特别是开发没有 ROM BIOS 支持的嵌入式系统时,开发设计和实现驱动工作量要占整个系统的三分之一甚至还多,驱动实现的好坏影响到系统硬件的性能的发挥,在整个操作系统中占有很重要的作用。

然而基于嵌入式系统设备驱动的开发和桌面计算机系统设备驱动的开发不一样,后者的驱动开发有现存的模型,比如有 Windows 驱动程序模型(WDM),O'Reilly&Associates 公司出版的《LINUX DEVICE DRIVER》也提出 LINUX 下驱动设计的模式等。对于前者由于面向不同应用,系统结构也各不一样,外加嵌入式系统一般不公开源代码,嵌入式设备驱动方面的资料很少,所以很难有统一的设备驱动开发模式。但是嵌入式系统中设备驱动的设计和实现无章可循,势必会延长嵌入式系统的开发时间和增大维护和调试系统的难度。通过对多个嵌入式系统设备驱动的设计及其实现的归纳总结,该文抽象出并分析了一种比较通用的嵌入式设备驱动的架构,给出了一例应用此架构开发的嵌入式系统。应用此架构开发嵌入式系统,明显地缩短了系统的开发时间和方便了驱动和维护。

### 2 设备驱动与嵌入式系统的关系模型

对于嵌入式系统来说,由于应用的不同其硬件构成和软件的体系结构千变万化,相应的设备驱动与嵌入式系统的关系也

不一样。该文从复杂的嵌入式系统应用中总结设备驱动与嵌入式系统的关系,建立了一种简单抽象的关系模型,如图 1 所示。

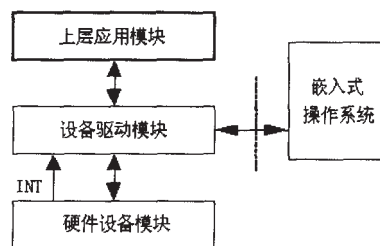


图 1 设备驱动模块与嵌入式系统的关系模型

从分层次模块化的角度来看,设备驱动模块与嵌入式系统中其它模块的关系如图 1 所示,最下层是硬件设备,在硬件设备之上的是设备驱动和操作系统,应用模块位于最上层。

设备驱动模块与嵌入式系统中各模块的交互,可以总结为:(1)向嵌入式操作系统申请资源,比如中断,内存, I/O 等;(2)为上层应用提供服务,比如设备功能子模块供上层调用;(3)对硬件设备进行操作,比如读写设备寄存器;(4)为设备提供中断服务。

该模型有广泛的代表性,比如可以广泛地应用在以 Vx-works、pSOS、Nucleus 为代表的嵌入式系统中。但是对于嵌入式 LINUX<sup>[2]</sup>来说,由于设备驱动作为操作系统的一部分不满足以上模型,不在该文讨论的范围之内。

### 3 嵌入式系统设备驱动架构

作者简介:杜旭,男,副教授,硕士生导师,主要研究方向:宽带网络通信,数字集成电路设计与验证。顿新平,男,硕士研究生,主要研究方向:宽带网络通信技术,嵌入式操作系统。黄建,男,博士研究生,主要研究方向:宽带网络通信技术。

嵌入式系统设备驱动架构的设计建立在图 1 所示的关系模型之上。明确设备驱动与嵌入式系统的关系,也就是确定了设备驱动与外界的抽象接口,基于以上抽象接口的嵌入式系统设备驱动架构如图 2 所示。

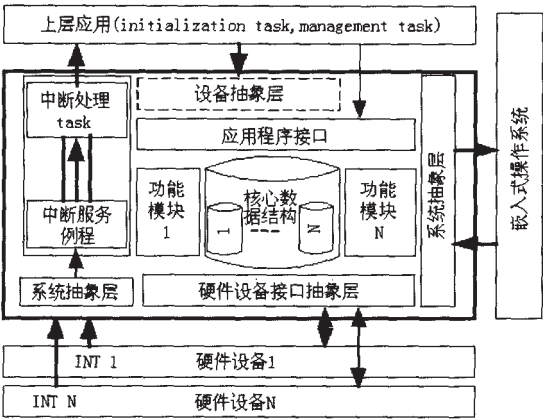


图 2 嵌入式系统设备驱动框架

### 3.1 模块及功能介绍

**硬件设备接口抽象层:**它实现与硬件设备交互,有间接和直接读写硬件设备两种方式。硬件设备中有各种寄存器,比如说状态、计数、中断、配置寄存器等等。硬件设备接口抽象层通过宏实现其抽象层的功能。用宏方式实现读写操作,当对设备读写操作方式改变时,相应地只需要改变宏定义,从而实现了抽象层功能。另外,读写操作简单,用宏实现可提高程序的执行效率和系统的稳定性。

**系统抽象层:**它实现设备驱动与嵌入式操作系统交互的封装,一般而言,交互的内容为:请求操作系统为设备分配和释放必要的内存空间;操作系统的时钟管理;设备驱动对设备操作可能要用到定时器;系统的中断管理功能用于开关中断和中断请求;进程通讯和同步等其他系统功能。系统抽象层封装了操作系统与设备驱动的交互,使设备驱动适用于不同的操作系统。

**设备抽象层:**封装设备驱动的应用程序接口。对于比较复杂的嵌入式系统来说,设备众多、管理非常复杂。管理每个设备需要调用很多应用接口函数,对多设备的管理势必会造成混乱,设备抽象层封装了设备管理要调用的应用接口函数,管理设备一般只需调用设备抽象层的一个抽象接口。

**应用程序接口:**提供对上层应用的支持。应用程序接口包含了设备具有的功能接口,能充分支持各种应用。

**驱动功能子模块:**仅与设备相关。不同的设备所提供的功能不一样,相同设备这些功能模块可重用。

**核心数据结构模块:**核心数据结构分为模块层和设备层。每个设备单元都用一个设备数据结构和配置数据结构表示,这两个核心数据结构处于模块层。模块层数据结构可以由多个设备层数据结构组成,模块核心数据结构处于模块层。数据分层有利于系统管理。

**中断处理模块:**中断处理模块分为两个子模块,中断服务例程与中断处理任务子模块,两个子模块可有多种通讯方式,随操作系统而定,比如有消息队列、共享内存、事件等多种方式。中断服务程序子模块只进行简单的操作,主要是读取设备的状态,将设备的相关数据拷贝至中断处理数据结构,然后将相应

的中断处理数据发送到队列。中断处理任务,在系统的调度下运行。中断处理任务在队列里取中断处理数据然后调用相应的中断处理函数,中断处理完成后释放数据空间。

### 3.2 驱动架构的关键技术

#### 3.2.1 模块和设备两级核心数据结构

引入模块和设备两级数据结构是驱动架构的一大亮点。在嵌入式系统中,设备驱动与系统相对独立,造成驱动设计很灵活,但是对大系统来说,设备驱动的管理将会很复杂,包括管理申请系统资源,驱动程序的调试和维护等。采用分层次的数据结构很好地解决了大系统设备管理的问题。用树来描述,图 3(a)部分树根结点  $M$  表示模块级数据结构,树的叶子  $\{DU[i], CU[i]\}$  表示  $M$  的第  $i$  个设备单元,每个设备单元包括一个或多个功能结构相似的设备。 $DU$  表示设备单元数据结构,  $CU$  表示设备单元配置数据结构,当配置数据相同只需一份配置数据结构。这种树结构使得驱动具有很好的扩展性:如果要在某模块中加一个设备单元,只需在设备单元数据结构数组中增加一个元素,然后将设备配置数据加到相应的设备配置数据结构中;卸载设备操作相反。

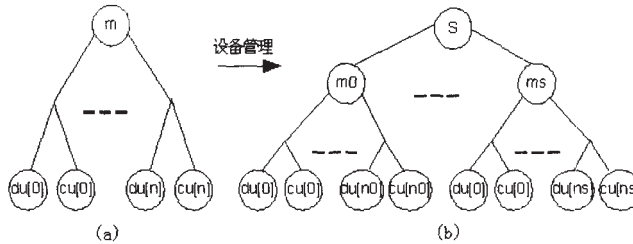


图 3 两级核心数据结构树和系统管理树

在设计设备单元数据结构时有两条原则:(1)相同设备共用数据结构,采用数组表示。(2)功能结构相似的设备组合成一个设备单元数据结构。依据以上原则设计的系统如图 3(b)所示,系统用  $S+1$  个模块管理  $N0+N1+...+Ns$  个设备单元,大大简化了系统的管理。下面定量作一下比较:

	无分层数据结构	分层数据结构
驱动程序数量	大于 $S+1$	$S+1$
系统增减设备	复杂	简单
代码量	较大	较小
代码调试	比较复杂,无规律	简单,有统一架构
程序维护	复杂,出错不易处理	简单,易处理

对应以上设备核心数据结构树,相应的数据结构抽象模型如下:

```
Module{
    ...
    DeviceUnit* ;           //设备单元数据结构指针
    ConfigurationUnit* ;    //设备单元配置数据结构指针
}

DeviceUnit{
    DeviceUnit_ Property ;  //设备单元属性
    DeviceUnit_ Method ;    //设备单元的操作函数指针
}

ConfigurationUnit{
    ...
    Configure_DeviceUnit ;  //配置设备单元数据结构
```

### 3.2.2 初始化和中断处理函数由上层应用处理

初始化和中断处理函数放在应用上层由用户管理,使得驱动程序具有良好的适应性。嵌入式系统有一个显著的特点,它是以应用为中心的。不同的应用,其体系结构和系统构成千差万别,即使同一结构,在应用场合同时,也要根据其用途来决定系统中各设备的处理过程和设备的初始化配置。把设备单元的初始化函数、设备单元配置数据和中断处理函数放在应用程序中时,可以很容易地根据具体的应用环境相应地改变设备单元的配置和处理函数,而不用改变驱动程序的其他部分。为了让系统运行时初始化设备,把设备单元初始化设置成系统的任务。对于复杂的系统,在应用层内加设备管理模块,同样把设备单元管理模块设置成系统的任务,设备任务负责调用设备抽象层对设备进行操作,一般而言系统的人机界面是通过设备管理模块实现的。

中断处理函数放在上层是设备单元的数据结构面向对象的结果。每个设备单元数据结构有其自身的函数指针指向中断处理函数。

对于初始化从模块级的角度来看,模块中每个设备单元采用循环方式初始化,抽象模型如下:

```
Start
Module open //模块打开状态,申请系统资源等
Module start //模块开始状态 可以把设备配置数据放入申请的内存中
While( DeviceUnit valid )
Do{
    DeviceUnit open //设备单元打开,给设备数据结构分配内存,基址,中断等
    DeviceUnit init //初始化设备单元,用设备配制数据配置设备数据结构等
    DeviceUnit start //开始设备单元,开中断等
    NextDeviceUnit
}
Stop
```

### 3.2.3 灵活的中断处理方式

采用了中断服务程序和中断处理任务子模块,中断处理方式非常灵活。

嵌入式系统中,嵌入式微处理器具有以下特点<sup>[3]</sup>:(1)具有较强的实时多任务处理能力。嵌入式系统大多是实时多任务系统,因此,嵌入式微处理器必须具有实时处理能力和多任务处理能力,以保证系统工作的正确性和高效率。(2)具有较强的中断处理能力。嵌入式系统的构成复杂,中断源种类繁多,中断的性质差异很大,这就要求微处理器应该具有较强的中断管理能力和较快的响应速度。

基于嵌入式系统的较强的多任务处理能力和中断管理能力,驱动架构首选了中断处理方式。中断服务例程在关中断状态下做了少量的事情,然后把处理的结果发送到队列。为了充分利用嵌入式系统的多任务处理能力,中断处理中引入了中断处理任务模块,用户可以根据具体的情况来确定中断处理任务的优先等级。架构的中断处理方式类似于 LINUX 的中断 top half 和 bottom half 处理。

中断处理的灵活性主要体现在,用户可以修改中断处理任务优先级,中断子模块有多种通信方式,依操作系统而定。采用了这种结构后减少了中断禁止时间,提高了中断处理的能力。特别是对于高速设备来说,中断服务占用处理器最少的时间接

收处理,发送到队列后,开中断后由中断任务处理中断,保证了处理器的中断处理能力和中断响应速度。

### 3.2.4 丰富的抽象层

可移植性是架构设计的基本要求<sup>[4]</sup>。为了获得好的可移植性,在设计的系统中采用了系统抽象层,设备抽象层,硬件设备接口抽象层。采用抽象层是现代设备驱动的趋势。利用抽象层,设备驱动有了好的可移植性,改变系统抽象层里的封装,就可使设备驱动适用于不同的操作系统平台。改变设备抽象层的封装就可适用于不同的上层应用,方便了用户对设备的管理。同样地,改变硬件设备接口抽象层的宏定义即可适用于不同设备。

## 4 基于该架构驱动的实现

根据以上的讨论,下面给出基于该架构的嵌入式系统中设备驱动实现的例子。嵌入式系统结构如图 4 所示,嵌入式微处理为 TOSHIBA TX3927,嵌入式操作系统为 NECLEUS OS,ASIC 网络处理系统由某研究中心与美国 COMBRIO NETWORKS 公司合作开发的 3 块网络芯片和 INTEL IXF1104<sup>[5]</sup>千兆以太网媒体接入控制器组成。INTEL IXF1104 与 ASIC 芯片通过 SPI3 接口标准相接,挂接在与处理器的 PCI Local BUS 上。微处理负责配置 IXF1104 和做出错处理。通过调用设备的功能子模块函数可以将设备的运行状态和对设备的配制显示在人机界面上。

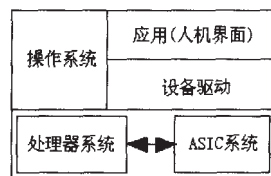


图 4 嵌入式系统

INTEL IXF1104 在系统中起分流作用,将高速的数据包分 4 路发给下游设备。系统中用到了三块 IXF1104,每块可以处理 4Gbps 的带宽。IXF1104 采用此驱动架构,下面简要介绍 IXF1104 的驱动实现过程:

(1)定义各模块接口和抽象层。

(2)定义数据结构,完全采用了两极核心数据结构。

依据双层数据结构模型,很容易写出设备的数据结构。分析核心设备单元数据结构如下:

```
typedef struct _sIXF1104_device
{
    UINT2    configNum ;
    void    *baseAddr //I/O 基地址
    CFG_GLOBAL cfgGlobal //设备全局配置
    CFG_CHAN cfgChan [IXF1104_CHAN] //单个通道的配置,设备有 4 个通道
```

```
Device_Int Interrupt //设备中断处理数据结构,该结构包含很多中断处理函数指针
```

```
}sIXF1104_DEVICE ;
```

与设备结构模型对比,可以看出设备单元数据结构 sIXF1104\_DEVICE 的前四项为设备单元属性 DeviceUnit\_Property,而设备中断处理数据结构对应设备单元的方法 DeviceUnit\_Method。其他数据结构依次类推。

(3)编码各接口,抽象层和功能模块。



(4)初始化和中断处理,通过应用程序对 IXF1104 初始化和中断处理。

在 IXF1104 初始化的过程中,采用了如图 5 所示的流程,首先是模块级的初始化,模块级的初始化在 ModuleOpen 主要是给 Module、Device 和 Configuration 结构静态申请内存,给中断处理分配内存,由于 NECLEUS 采用内存池的内存管理方式,所以内存的使用很固定,采用静态分配易于管理。在 ModuleStart 阶段将用户配置的 Configuration 数据以数组方式存入分配的内存中。

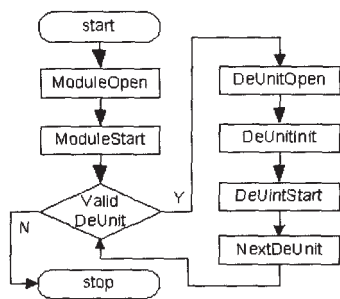


图5 初始化流程

其次,进入设备的初始化,DeviceUnitOpen 阶段在申请的静态内存中为当前设备分配内存空间,申请中断。DeviceUnitInit 阶段把配置数组相应数据赋给当前设备数据结构。DeviceUnitStart 阶段开中断等。

中断处理过程相对简单。中断服务例程做两件事情(1)来中断后,判断中断源并设置相应的数据结构。(2)发送中断事件。采用了中断处理任务模块,中断任务模块在接受事件,然后调用应用程序来处理。

(5)在 NECLEUS<sup>[6]</sup>中设置 TASK。

在 NECLEUS 中设置初始化 TASK、中断处理 TASK,另外在 人机界面 TASK 中加入管理 IXF1104 的界面。系统运行时,将调用这些 TASK 对系统进行初始化,处理中断和人机交互。

```
struct IN_TASK_DEFINITION_STRUCT IN_System_Tasks[] =
{
    {SystemInitializationTask ...},
    {IsrTask ...},
    {ConsoleTask ...},
    ...
    {END_OF_LIST}
};
```

## 5 结束

基于嵌入式系统的设备驱动架构的设计,尤其在较大的系统中,采用此架构设计的驱动可以明显地减少驱动程序的数量,减少程序调试和维护的难度,使得系统具有良好的扩展性和代码具有良好的可移植性。对嵌入式系统驱动架构的设计是对嵌入式系统开发的一次很好的尝试,随着嵌入式系统的广泛使用,驱动架构将会发挥更大的作用。(收稿日期 2003 年 12 月)

## 参考文献

- 1.魏忠等.嵌入式开发详解[M].北京:电子工业出版社,2003-01
- 2.ALESSANDRO RUBIN.LINUX DEVICE DRIVER[M].O'Reilly&Associates Inc,1998
- 3.李林功等.嵌入式系统的构成与特点[J].电测与仪表,2003(5)
- 4.高晓峰等.框架技术在嵌入式系统中的应用[J].计算机工程与应用,2002,38(22):124-127
- 5.Intel IXF1104 Quad-Port 10/100/1000 Mbps Ethernet Media Access Controller Preliminary Datasheet.Revision Number 005
- 6.The source codes of NUCLEUS OS

(上接 106 页)

设计师所要做的只是通过软件界面对映射方式进行控制,这使设计师的工作方式更接近于传统的设计方式。图 6 为人体尺寸驱动设计的界面。待驱动设计的产品尺寸从场景中捕捉或从尺寸列表中选择;驱动人体尺寸从人体数据库中选择,映射驱动方式在驱动设计面板中进行控制和参数输入。目前“中设”软件所提供的驱动映射方式暂时只有比例驱动一种,更复杂的驱动函数将根据对 人机设计具体分类任务的研究逐步设立添加。

作业分析是对所设定的工作姿态或产品结构所要求的工作姿态进行人体生理力学分析,评价其安全性与健康性。该部分功能尚在研制中。

人机部件库为开放式模型库,除系统提供的标准人机部件外,用户还可将自定义部件添加入库中。人机部件库中的部件尺寸全部与人体数据库关联,可以方便地对部件尺寸参数进行人机驱动设定。

## 4 结论

“中设 R.0”软件人机设计模块的开发实践对 人机工程 CAD 的研究推进有如下意义:

- (1)实现了系统全面的标准人体数据的查询;
- (2)建立了产品设计过程中的实时人体模型视觉化辅助,

并以人体模型为人体数据载体,简化了设计师对人体数据的查询工作;

(3)实现了产品尺寸的人体数据驱动设计,简化了设计过程,使设计师的工作更接近自然的设计模式,有利于把设计师的工作重点集中在概念的处理和设计实施而不是复杂的软件操作上。

“中设”软件人机设计模块进一步的工作重点是人体数据驱动工具的开发。目前仅实现了尺寸的驱动,形态和概念方案的驱动设计将是今后的主要研究目标。(收稿日期 2004 年 5 月)

## 参考文献

- 1.刘肖健等.基于遗传算法的产品人机 CAD 研究[J].计算机工程与应用,2003,39(33):35-37
- 2.宋保华等.计算机辅助工业设计系统研究与中设系统 R1.0 简介[J].计算机辅助设计与图形学学报,2000,12(11)
- 3.李桂琴等.产品的多感知设计过程原型系统[J].计算机工程与应用,2003,39(27):41-43
- 4.国家技术监督局.GB 10000-88:中国成年人人体尺寸[S].1987
- 5.Henry Dreyfuss Associates.Human Scale 1/2/3 4/5/6 7/8/9[M].MIT Press,2001
- 6.Henry Dreyfuss Associates.The Measurements of Man and Woman.2002