

# 引导程序中 USB 下载功能的设计与实现

柴东岩<sup>1,2</sup>, 侯紫峰<sup>1,3</sup>

(1. 中国科学院计算技术研究所, 北京 100080; 2. 中国科学院研究生院, 北京 100049; 3. 联想研究院, 北京 100085)

**摘要:**在新型智能手机、移动终端、PDA 等便携设备的开发过程中,通过串口或以太网进行文件下载的传统方式受到下载速度或设备尺寸等因素的限制,无法满足研发需求,因此,需要一种更具可行性和适用性的替代方案。该文基于引导程序开发了一套 USB 文件下载系统,在不依赖于任何上层系统软件的情况下,能够方便高效地下载任意文件到设备端内存中,在开发调试阶段用于下载内核和驱动,在产品化阶段用于系统软件的更新和升级。

**关键词:**引导程序; U-Boot 引导程序; 通用串行总线; 嵌入式系统

## Design and Implementation of USB Download Function in Bootloader

CHAI Dong-yan<sup>1,2</sup>, HOU Zi-feng<sup>1,3</sup>

(1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080;

2. Graduate University of Chinese Academy of Sciences, Beijing 100049; 3. Lenovo Corporate Research, Beijing 100085)

**【Abstract】**When developing new generation portable devices such as advanced mobile terminal and personal digital assistant, the traditional serial port and Ethernet interface are not suitable for downloading larger files. A novel method using USB to download files in Bootloader is proposed. For TI H2 evaluation board and the self-designed mobile terminal, this paper develops the USB download function, which can be used in developing and production phases for system software download and upgrade.

**【Key words】**Bootloader; U-Boot; USB; embedded system

在传统的嵌入式系统的开发中,经常使用串口进行调试和下载文件。但是,串口数据传输速率有限,当需要下载的数据量较大时,通过以太网连接并借助 TFTP 协议来下载文件是更好的选择。智能手机、移动终端、PDA 等便携设备的日渐流行给开发人员提出了新的挑战。一方面,此类设备的功能丰富,因此,在开发调试过程中需要下载的系统软件规模较大,可达几兆甚至是十几兆字节;另一方面,受设备重量、尺寸等限制,此类设备不会配备以太网络接口。为了提高数据下载速率,基于USB实现下载功能是一个很好的选择,这是因为:(1)USB理论上提供 12 Mb/s或 480 Mb/s的数据传输速率,满足了性能需求;(2)USB接口是个人计算机的标准配置,且很多便携设备都使用USB接口进行数据同步或充电,不用增加额外的硬件设备和生产成本,具有可行性和广泛的适用性。本文描述了基于Bootloader(移植自U-Boot<sup>[1]</sup>)的USB下载功能的设计和实现方法。文中所涉及的硬件细节均基于TI的Helen2 评估板(简称H2)和OMAP1610<sup>[2]</sup>平台。按照USB规范<sup>[3]</sup>的定义,本文称PC为主机(host)端,称h2 或移动终端为设备(device)端。

### 1 设计方案

结合常见的 USB 设备的应用方式,为了在 Bootloader 中实现 USB 下载功能,可以考虑采用如下几种方案:(1)USB Net 方式:在主机端通过驱动程序将 USB 设备模拟成网络设备,当作以太网卡使用,双方可以使用标准的网络协议和服务进行通信。(2)USB Serial 方式:与 USB Net 方式的思路类似,但是将 USB 设备模拟成串口设备,通过串口终端程序(如 minicom)使用 kermi, xmodem 或 zmodem 等传输协议与设备端通信。(3)USB Storage 方式:将 USB 设备当作 U 盘使用,

在主机端使用文件拷贝的方式完成文件下载。(4)自定义方式:在实现 USB 数据传输的基础上,自行定义主机端和设备端的通信协议,从而完成文件下载功能。

将主要的开发任务分解为图 1 所示的几个部分。无论采用何种方案,其中设备端的各组件都需自行开发。USB 总线驱动和设备驱动部分的开发内容相同,但功能驱动和命令行接口不同;主机端的总线驱动已由操作系统提供支持,根据方案的不同,其他组件所需工作量和开发复杂度均有所不同。



图 1 USB 下载功能开发任务的分解

在方式(1)中,通过使用开源社区提供的内核补丁<sup>[4]</sup>,主机端只需要进行简单的修改和配置就可以使用标准的TCP/IP 协议栈和各种网络服务;设备端需要完成以太网组帧和上层协议的解析工作,并开发相应的命令行接口。方式(2)同样可以使用内核补丁<sup>[4]</sup>提供的USB驱动,使用操作系统提供的串口终端程序作为客户端工具,在设备端要实现串口通信协议和命令行接口。方式 3 本质上是开发一个U盘,在主机端不

**作者简介:**柴东岩(1978 -),男,博士研究生,主研方向:无线网络,移动计算,嵌入式系统;侯紫峰,研究员、博士生导师

**收稿日期:**2007-05-10 **E-mail:** dychai@yahoo.com.cn

需要进行任何工作，但在设备端需要组织文件系统和用户视图映射，开发的工作量大于前 2 种实现方式。在方式(4)中可以自定义一种格式简单传输协议，简化设备端的实现；但采用了非标准协议，必须自行开发主机端专用的驱动和客户端工具。方式(4)所需的实际工作量整体上是最大的。综合考虑实现的复杂度和使用的方便性，本文选择USB NET方式作为USB下载功能的实现方案。为实现下载功能，不需要在设备端开发完整的网络协议栈，只要支持一种简单的文件传输协议TFTP。使用时，用户通过命令行接口下载文件，不需要在主机端操作任何客户端软件，操作简便、易用性好。

## 2 USB 下载功能的实现

### 2.1 硬件初始化

TI的H2 评估板使用Philips的ISP1301<sup>[5]</sup>作为USB外部收发器，ISP1301 通过I<sup>2</sup>C总线与OMAP连接，其总线地址是0x2D。硬件初始化的过程为：(1)Bootloader通过I<sup>2</sup>C总线完成ISP1301 的硬件配置。(2)要进行OMAP的管脚设置。OMAP提供了丰富的外设接口，包括 3 组USB接口，需要由开发人员根据具体设计指定相应芯片管脚的功能。H2 通过OMAP的第 2 组USB接口与ISP1301 通信，分别使用管脚W14, W16, W13, V15, R13, W4 作为USB.TXD, USB.TXEN, USB.SE0, USB.RCV, USB.SPEED, USB.PUEN信号。(3)初始化OMAP的USB设备控制器和OTG控制器，以便与主机端正确通信，定义USB描述符信息如下：

配置数	1
接口数	1
数据传输端点数	2
测试用厂商 ID	0xCCAA
测试用产品 ID	0xCCDD
数据传输输出端点号	1
输出端点包大小	64 B
数据传输输入端点号	2
输入端点包大小	64 B
设备类型	通信设备类
设备子类型	自定义
协议类型	自定义

### 2.2 “中断”处理流程

在完成硬件初始化后，USB 的配置和数据传输都是通过响应 USB 设备控制器(UDC)的各种中断请求来实现的。由于Bootloader 中不完全支持中断，因此采用轮询方式，通过循环读取OMAP的USB设备控制器的中断源寄存器(IRQ\_SRC)来判断是否有相应的事件产生，如果有，则转入与之对应的“中断”处理例程中处理，处理流程伪码如下：

```

if (有中断事件产生，即寄存器 IRQ_SRC 的值为非 0) {
    if (IRQ_SRC_DS_CHG 置位) {
        //设备状态转移中断处理函数
        USB Device State Changed Interrupt Handler;
    } else if (IRQ_SRC_EP0_RX 置位) {
        //控制端点接收中断处理函数
        Endpoint 0 RX Interrupt Handler;
    } else if (IRQ_SRC_EP0_TX 置位) {
        //控制端点发送中断处理函数
        Endpoint 0 TX Interrupt Handler;
    } else if (IRQ_SRC_SETUP 置位) {
        //配置中断处理函数
        Setup Interrupt Handler;
    }
}

```

```

} else if (IRQ_SRC_EPn_RX 置位) {
    //数据接收端点中断处理函数
    Non-Isynchronous Non-Control Endpoint \
    Receive Interrupt Handler;
} else if (IRQ_SRC_EPn_TX 置位) {
    //数据发送端点中断处理函数
    Non-Isynchronous Non-Control Endpoint \
    Transmit Interrupt Handler;
}
}

```

### 2.3 组帧和 TFTP 协议解析

Bootloader 中定义 USB 数据传输端点的类型为 BULK 方式，一次传输的最大数据包大小为 64 B，而主机端以太网的 MTU 定义为 1 500 B，两者不一致。向主机发送 TFTP 读请求和 TFTP 确认等数据报文时，将其封装成以太网数据帧，添加 CRC 数据，之后将封装好的数据帧分割成多个 USB 请求块(URB)发送；反之，从主机接收数据时须进行多次 USB 数据接收操作以得到一个完整的数据帧，之后再对其进行校验和上层协议解析，得到 TFTP 数据包。模拟以太网数据封装操作是实现 USB 文件下载功能的核心步骤，必须严格按照相关协议的协议数据单元(PDU)格式进行。

本系统采用TFTP<sup>[6]</sup>作为文件传输协议。TFTP传输起自一个读取或写入文件的请求，这个请求也是连接请求。如果TFTP服务器批准此请求，则建立连接。创建连接时，通信双方各选择一个TID，用于在UDP通信时选择端口。第 1 次请求时客户端会将请求发到TID 69，即服务器的 69 端口上。应答时，服务器使用一个选择好的TID作为源TID，并在接收包中提取客户端TID作为目的TID进行发送。这 2 个TID在本次文件传输中会一直被使用。除最后一个数据包外，双方以 512 B 定长传输数据。发送方发出下一个数据包前必须得到接收方对上一个数据包的确认，如果数据包在传输过程中丢失，发送方会在超时后重新传输最后一个未被确认的数据包。一个有效的TFTP数据帧包含MAC头、IP头、UDP头、TFTP头、TFTP数据以及CRC校验值，其PDU格式如图 2 所示(括号中的数字表示占用字节数)。其中，TFTP头中 2 B的操作码指出了 5 种包类型之一：读请求(RRQ,1)，写请求(WRQ,2)，数据包(DATA,3)，确认包(ACK,4)和差错包(ERROR,5)。如果只实现USB下载功能，不会用到写请求包。



图 2 TFTP 协议数据单元格式

### 2.4 命令行接口

在 Bootloader 中添加“usbload”命令用于支持文件下载。该命令的使用格式为“usbload loadAddress loadFilename”，其中，loadAddress 为内存地址，loadFilename 为主机端 TFTP 服务器上存放的文件名称。接收到 usbload 命令后，Bootloader

执行如下操作：(1)根据 usbload 命令行中传递的文件名，发出 TFTP 读请求包；(2)以轮询方式响应 UDC 的 EPn\_RX 中断，接收 USB 数据，并将接收的数据重组成以太网的数据帧；

(3)对以太网数据帧进行 CRC 校验，如果通过，则将其作为一个有效的 TFTP 数据包接收，并发送 TFTP 确认包；(4)重复(2)、(3)，直到收到一个实际数据小于 512 B 的 TFTP 数据包，表明已经接收到最后一个数据包，文件下载结束；(5)根据文件头进行数据完整性校验(可选)。

## 2.5 主机端 USB 驱动

主机端的 USB 驱动来自开源社区的 USBDNET 内核补丁。该驱动模块初始化过程中向系统内核注册 USB 设备，通过 id\_table 数据结构识别要管理的 USB 设备。只须保证 USBDNET 中的 id\_table 和 Bootloader 中提供的描述符信息一致，主机端就可以正确检测到 H2 并与之通信。在源文件 usbdnet.c 中修改代码如下：

```
#define H2_VENDOR_ID 0xCCAA //厂商 ID
#define H2_PRODUCT_ID 0xCCDD //产品 ID
#define CDC_DEVICE_CLASS 0x02 //通信设备类
#define H2_DEVICE_CLASS CDC_DEVICE_CLASS
#define H2_INTERFACE_CLASS 0xff //自定义接口类
#define H2_INTERFACE_SUBCLASS 0x02 //接口子类
static __devinitdata struct usb_device_id id_table[] = {
//厂商 ID, 产品 ID, 设备类, 接口类, 接口子类
{MY_USB_DEVICE(H2_VENDOR_ID,
H2_PRODUCT_ID,
H2_DEVICE_CLASS,
H2_INTERFACE_CLASS,
H2_INTERFACE_SUBCLASS)}
//termination entry
```

(上接第 267 页)

在“studyManager” bean 中覆盖从 txProxyTemplate 模版中继承来的事务策略。

### 2.2.4 领域模型

领域模型是实际需求的业务对象集合，能在不同的层之间移动。它实现粗粒度的传递方式，为表示层提供表现所需要的数据源，为持久层提供被持久的对象。

系统中把重用度高的并且与 domain object 相关的业务逻辑封装领域对象模型中，而那些重用度低的、与 domain object 没有密切关联并且对 StudyDAO 接口有显式依赖的置于 service 层中，从而避免领域对象与其他的应用组件相耦合的情况出现，使它尽可能地得到重用，保证各层间单向依赖，即 Martin Fowler 所指的 rich domain object。同时，在 Study-Manager 中也做了简单封装，以保证实体类中的业务逻辑进行事务管理和持久化触发。

领域模型对象包括 VO 和 PO。VO 表示业务层的数据，PO 表示持久层的数据。表示层中 View 的 Form Bean 数据，在 Action 中由 VO 转化得到。由于使用 Hibernate 作为持久化框架，可以动态生成 PO，使 PO 状态管理可以脱离 Session，这样 PO 就可以完全取代 VO，在业务层和持久层间流动。

## 3 结束语

本文系统基于轻量级 J2EE 框架，采用松耦合编程体系结构，分离了业务逻辑层、数据持久层和表现层，在实际应

{});

完成上述修改后，重新编译得到 USBDNET 驱动模块。当 H2 通过 USB 线与主机连接时，主机自动加载驱动并完成网络配置，在 Bootloader 中通过 usbload 命令即可从主机上下载文件到指定内存区域中。

## 3 结束语

本文在 Bootloader 中设计了一套基于 USB NET 方式的文件下载系统，分别在 TI 的 H2 评估板和自主开发的移动终端上开发实现，在实际使用中功能稳定，有效下载速率分别达到了 1.3 Mb/s 和 1.7 Mb/s。为了方便普通用户使用，将增加文件上传功能用于系统备份，并在 Microsoft Windows 平台上实现了 USB 网络设备驱动。同时为了提高安全性，将增加签名验证功能，以防止恶意用户或因误操作对系统造成破坏。

### 参考文献

- [1] Wolfgang Denk. The DENX U-Boot and Linux Guide (DULG) for TQM8xxL[EB/OL]. (2004-11-18). <http://www.denx.de/wiki/bin/view/DULG/Manual>.
- [2] OMAP1610 Multimedia Processor Technical Reference Manual, SWPU062C[R]. Texas Instruments, 2003.
- [3] Compaq, Intel, Microsoft, NEC. Universal Serial Bus Specification Reversion 1.1[S]. USB1.1, 1998.
- [4] Lynne S, Rushworth T. Linux USBDNET Kernel Patch [EB/OL]. (2004-11-19). <http://www.ruault.com/Zaurus/patches/sbdnet-2.4.20-patch.gz>.
- [5] ISP1301 Universal Serial Bus On-the-go Transceiver, Rev.01[R]. Philips Electronics, 2004.
- [6] Sollins K. The TFTP Protocol (Revision 2)[S]. RFC 1350, 1992.

用中有以下优势：(1)操作系统无关性。由于 J2EE 的跨平台性，可在不同操作系统间移植。(2)数据库无关性。Hibernate 能很好地支持跨数据库的可移植性，本系统已在商业数据库 DB2, oracle, derby、开源数据库 MySQL 及国产达梦数据库上测试过，很好地实现了在各数据库间进行数据迁移。(3)低成本。由于本系统基于轻量级 J2EE 框架，因此无须使用高端服务器或商用数据库，且它使用的是国产基础软硬件 Linux + NC，即节省了成本又有利于推动国产软硬件的发展。(4)易维护及升级。使用 3 层架构，使层间松散耦合，易于维护，具有良好的扩展性和伸缩性。

本系统已应用在北京、新疆、四川、宁夏、陕西等地的 12 所示范学校，并且为海量教育资源管理系统的集成提供了远程资源访问接口，还须在高级功能实现和个性化方面不断完善。

### 参考文献

- [1] Johnson R, Hoeller J. Expert One-on-one J2EE Development Without EJB[M]. American: Wrox, 2004.
- [2] Davism. Struts, an Open-source MVC Implementation[EB/OL]. (2003-10-11). <http://www-106.ibm.com/developerworks/library/j-struts/?dw-zone=java>, 2003.
- [3] Harrop R. Spring 专业开发指南[M]. 成都: 电子工业出版社, 2006.
- [4] Alur D, Crupi J, Malks D. Core J2EE Patterns: Best Practices and Design Strategies[M]. American: Prentice Hall 2003.