

文章编号: 1671- 251X( 2010) 07- 0022- 04

# 基于 TFTP 协议的 ARM 软件远程更新系统

李 俊, 王 金 海

(天津工业大学信息与通信工程学院, 天津 300160)

**摘要:** 针对嵌入式产品维护和无人值守区嵌入式产品升级困难的问题, 提出了一种基于 TFTP 协议的 ARM 软件远程更新系统的设计方案。该系统主要由 LPC2290 微控制器、DM9000E 网卡芯片以及 SST39VF1601 型 NOR FLASH、MT45W4MW16 型 RAM 组成, 采用 bootloader 技术及以太网 TFTP 协议来远程更新无操作系统运行环境下 ARM 微控制器的应用软件。测试结果表明, 该系统能够保证数据的确传输和存储, 大大降低了 ARM 嵌入式系统软件的升级维护难度。

**关键词:** 嵌入式微控制器; ARM; 远程更新; 以太网; TFTP; bootloader; NOR FLASH 固化

**中图分类号:** TD679 **文献标识码:** B

## Remote Update System of ARM Software Based on TFTP

LI Jun, WANG Jin-hai

(College of Information and Communication Engineering of Tianjin Polytechnic University,  
Tianjin 300160, China)

**Abstract:** In order to resolve problems of maintenance and upgrade of embedded products in unattended area, a design scheme of remote update system of ARM software based on TFTP protocol was proposed. The system is mainly composed of LPC2290 micro-controller, DM9000E network card chip, SST39VF1601 NOR FLASH and MT45W4MW16 RAM. It can remotely update application software of ARM micro-controller without operating system by use of bootloader technology and TFTP protocol of Ethernet. The testing result showed that the system can ensure correct transmission and storage of data, and reduce difficulty of upgrade and maintenance of ARM embedded system software greatly.

**Key words:** embedded microcontroller, ARM, remote update, Ethernet, TFTP, bootloader, NOR FLASH programming

## 0 引言

以太网以其高速、高效、高性能的优点, 已经在金融、商业和各种工矿企业得到了广泛应用。现在, 越来越多的微控制系统都设计了以太网通信接口, 这给软件的远程更新提供了有利条件。传统的 ARM 微控制器软件更新方式是通过现场对芯片编程或者将其回收编程来实现的<sup>[1]</sup>。在产品数量较多、产品分布区域较广的情况下, 采用该方式实现对

软件的更新将耗费大量的人力和物力。因此, 实现远程更新就显得尤为重要。能进行远程更新的系统可以大大节省维护升级成本, 同时也能增强产品的竞争力<sup>[2]</sup>。

在实现远程更新便利性的同时, 也要保证微处理器的运行软件在完成更新之后能够正常启动并可靠运转。本文提出利用 bootloader 技术<sup>[3]</sup>对无操作系统运行环境下的微处理器的运行软件实现远程更新, 即开发一个 bootloader 程序来实现微控制系统开机启动后的远程通信和程序下载固化, 并启动所下载的新应用程序, 从而实现嵌入式系统软件的远程更新。Bootloader 程序基于 TFTP 通信协议, 这样能充分提高对大文件的下载速度<sup>[4]</sup>, 并有效地提高远程更新的可靠性。

收稿日期: 2010- 03- 05

基金项目: 天津市科委自然科学基金项目( 08JCYBJC14700)

作者简介: 李 俊( 1984- ), 男, 湖南益阳人, 天津工业大学在读硕士研究生, 主要研究方向为信号与信息处理。E-mail: ljdtj@163.com

## 1 系统结构

ARM 软件远程更新系统主要由以太网接口模块、微处理器模块以及存储器模块 3 个部分组成, 如图 1 所示。

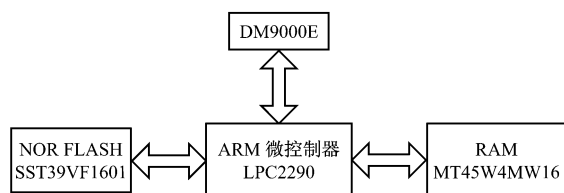


图1 ARM 软件远程更新系统组成

微控制器选用具有 ARM7 内核的 LPC2290。LPC2290 没有片内 FLASH, 其外部存储器模块分为 NOR FLASH 和 RAM 两部分。NOR FLASH 选用 2 MB SST39VF1601, RAM 选用 8 MB MT45W4MW16。以太网接口芯片选用的是 DM9000E 10/100 M 以太网控制器。另外, 将 LPC2290 的一个 GPIO 口连接一个按键, 作为升级请求按键; 再选择一个 GPIO 口连接 LED 指示灯, 作为升级开始/完成指示灯。

## 2 系统工作原理

将设计好的 bootloader 程序固化在从 NOR FLASH 起始地址开始的低地址区域, 剩余地址空间用于存放用户应用程序。系统通电启动后, 将从 NOR FLASH 引导运行。此时, bootloader 程序将完成对 LPC2290 和 RAM 的初始化, 并将 bootloader 代码的剩余执行部分复制到 RAM 中运行; 然后完成 DM9000E 的初始化, 接着打开 LPC2290 内部的定时器来检测升级请求按键是否被按下。如果在定时时间内检测到升级请求按键被按下, 则使指示灯闪烁一次, 表示升级开始。此时 bootloader 程序将通过 TFTP 协议向上位机服务器请求下载新的应用程序。接收到的 TFTP 数据包首先缓存在 RAM 中, 随后 bootloader 程序调用 NOR FLASH 擦除程序擦除用户应用程序地址段一个 block 的存储区, 然后将数据包的内容以字编程的方式固化在该存储区。将所有接收到的数据包固化到 NOR FLASH 之后, 程序退出与上位机服务器的 TFTP 通信, 然后点亮升级完成指示灯并跳转到用户应用程序的首地址, 开始执行新的应用程序。此时, 原来的应用程序已经被擦除, 并被新的应用程序所覆盖。若升级请求按键没有在定时时间内被按下, 则程序直接跳转到用户应用程序的首地址开始

执行。

## 3 系统硬件设计

对于系统微控制器 LPC2290, BOOT[1:0] 脚的状态控制着引导方式。这里选择将 P2.27 脚接下拉电阻, 使得复位时 LPC2290 的 BOOT[1:0] 状态值为 01, 表示从 Bank0 引导启动。LPC2290 含有外部存储器控制器 EMC。EMC 是一个 AMBA AHB 总线上的从模块, 它为 AMBA AHB 系统总线和外部存储器器件提供了一个接口。该模块可同时支持多达 4 个单独配置的存储器组, 每个存储器组都支持 SRAM、ROM、FLASH 存储器或一些外部 I/O 器件<sup>[5]</sup>。

### 3.1 以太网接口电路设计

LPC2290 使用 16 位总线方式对 DM9000E 进行控制, 并使其工作在 100 M 全双工模式下。以太网接口电路如图 2 所示。LPC2290 的数据总线 D0~D15 与 DM9000E 的 SD0~SD15 连接, 地址线也相应连接。LPC2290 的片选线 CS3 与 DM9000E 芯片的 AEN 端连接, LPC2290 的地址线 A2 与 DM9000E 芯片的命令/数据使能端 CMD 连接。这样, 结合片选线得到的 DM9000E 的地址端口的 32 位地址为 0x83000300, 其数据端口的 32 位地址为 0x83000304。

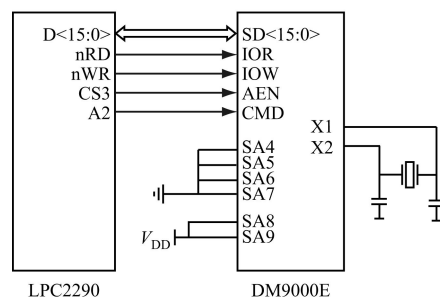


图2 以太网接口电路

### 3.2 存储器接口电路设计

对于 LPC2290 与 SST39VF1601 的接口电路, LPC2290 的片选线 CS0 与 SST39VF1601 的 CE 端连接, 这样 SST39VF1601 属于 Bank0 组。当 bootloader 代码固化到 SST39VF1601 时, 系统启动后即可运行 bootloader 程序。访问地址为 0x80000000~0x801FFFFFFF。存储器组配置成 16 位宽度, LPC2290 的地址总线 A1~A23 分别与 SST39VF1601 的 A0~A22 连接。

对于 LPC2290 与 MT45W4MW16 的接口电路, 由于 MT45W4MW16 也是采用 16 位总线接口,

所以使用 LPC2290 的地址总线 A1~A23 与其连接。LPC2290 的片选线 CS1 与 MT45W4M W16 的  $\overline{\text{CE}}$  端连接。该存储器属于 Bank1 组, 所以其访问地址为 0x81000000~0x817FFFFF。

#### 4 系统软件设计

所设计的 bootloader 程序主要完成系统初始化、开始 TFTP 通信、对 FLASH 进行擦写操作和执行程序的跳转等功能。整个软件的工作流程图 3 所示。

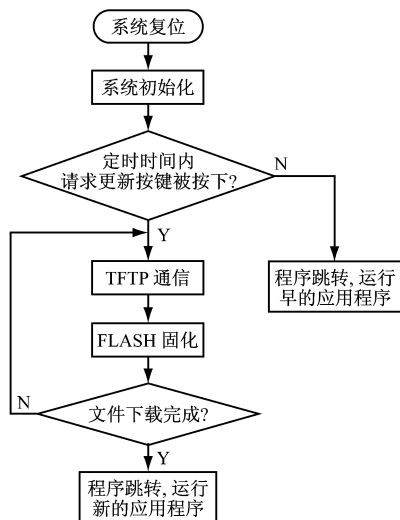


图 3 软件工作流程

##### 4.1 系统初始化程序

系统初始化程序依次完成对外部总线控制器的初始化、对堆栈的初始化、对定时器配置和对 DM9000E 网卡芯片的初始化。实现对 DM9000E 网卡芯片的初始化必须通过设置 DM9000E 网卡芯片内部的寄存器才能完成。通过如下几个读端口与写端口的宏定义:

```

# define DM9000_outb(d, r) ( * ( volatile unsigned char * )
r = d)
# define DM9000_outw(d, r) ( * ( volatile unsigned short * )
r = d)
# define DM9000_outl(d, r) ( * ( volatile unsigned long * )
r = d)
# define DM9000_inb(r) ( * ( volatile unsigned char * ) r)
# define DM9000_inw(r) ( * ( volatile unsigned short * ) r)
# define DM9000_inl(r) ( * ( volatile unsigned long * ) r)
  
```

分别实现了 DM9000 内部寄存器读函数 DM9000\_regr() 和 DM9000E 内部寄存器写函数 DM9000\_regw(), 并由 eth\_init() 函数完成对 DM9000E 网卡芯片的初始化。DM9000E 网卡芯片的初始化流程: 首先复位 DM9000E 网卡芯片, 延时 1 ms 后激活 DM9000E 内部 PHY 的功能, 然后设置 PHY 的工

作模式; 清除状态标志位, 清除中断标志位; 设置 DM9000E 的 MAC 地址; 使能 DM9000E 的接收功能; 中断使能; 最后延时等待初始化完成。

系统初始化完成之后程序进入定时检测按键状态阶段。在所设定的 5 s 定时时间内, 系统循环检测与升级请求按键相连的 GPIO 口的输入电平状态。如果检测到低电平, 程序将进入 TFTP 通信模块。

##### 4.2 TFTP 通信模块程序

TFTP 即简单文件传输协议, 是一种基于 UDP 的传输协议, 比 FTP 协议简单和短小, 因此, 更有利于在存储资源相对较少的嵌入式系统中使用。TFTP 协议同样分为客户端和服务端, 将嵌入式目标平台作为客户端, 远程 PC 作为服务器端。除了需要实现应用层的 TFTP 协议之外, 还需要实现数据链路层的 ARP 协议, 用以获得远程 PC 的 MAC 地址, 以及网络层的 IP 协议和传输层的 UDP 协议。

TFTP 共有 5 种类型的报文: RRQ、WRQ、DATA、ACK 和 ERROR<sup>[3]</sup>。每个 TFTP 数据报的前面添加 8 B 的 UDP 首部形成 UDP 报文, UDP 报文的前面添加 20 B 的 IP 首部形成 IP 数据报。将通过 TFTP 协议下载新的应用程序的过程设计成客户端从服务器端读文件的过程, 其大致流程: 首先, bootloader 程序在进入 TFTP 通信模块后, 将向远处 PC 服务器发送读请求 RRQ 的 TFTP 数据报。该数据报包含需要读取的二进制文件名和文件模式, 这里设置文件模式为 octet。如果这个文件能被这个客户端读取, PC 服务器端将返回块编号为 1 的数据分组 DATA。客户端将发送块编号为 1 的 ACK, 确认收到数据分组。服务器端随后发送块编号为 2 的数据, 客户端发回块编号为 2 的 ACK。重复该过程直到这个文件传送完毕。除了最后一个数据分组可能含有不足 512 B 的数据外, 其它每个数据分组均含有 512 B 的数据。当客户端收到一个不足 512 B 的数据分组时, 说明文件读取完毕。随后, 程序将退出 TFTP 通信模块, 进入跳转子程序。跳转子程序由汇编语言编写, 直接对 ARM 的程序指针 PC 进行操作, 完成程序的跳转, 执行新的应用程序。

由于 TFTP 基于不可靠的 UDP 协议, 因此, 必须在程序中添加能够保证数据有效传输的机制来及时处理分组丢失的情况。在程序中设置了定时机制来达到超时重传的目的, 这样能有效提高远程更新

的可靠性。为了能够同时完成对多个客户端的软件升级,程序中采用了随机端口机制,即每个客户端采用随机端口号。

#### 4.3 NOR FLASH 操作模块程序

在TFTP通信模块程序中,每收到一个DATA数据分组,程序将调用NOR FLASH操作函数实现对数据包的FLASH固化。由于所设计的bootloader程序需要固化在NOR FLASH的起始地址,且因为其大小小于64 KB,因此,对NOR FLASH的存储空间作了如图4所示的布局。通过TFTP接收到的数据包将被固化在从0x80010000地址开始的地方,新的应用程序将覆盖旧的应用程序。在跳转子程序中就是将程序指针PC指向该地址,从而开始执行新的应用程序。

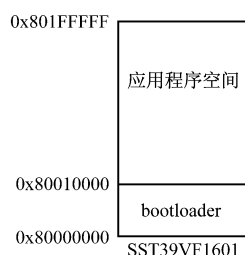


图4 NOR FLASH 存储空间布局

NOR FLASH 操作模块程序中实现了FLASH擦除函数flash\_erase()和FLASH编程函数write\_word()。在对NOR FLASH编程之前,首先要擦除NOR FLASH。NOR FLASH擦除分为扇区擦除、块擦除和整片擦除。程序中选用块擦除方式。对SST39VF1601的擦除步骤:首先写2个解锁周期,即在地址0x5555写入0x00AA和在地址0x2AAA写入0x0055;紧接一个设置周期,即在地址0x5555写入0x0080;接着还是写2个解锁周期;最后一个总线周期向要擦除的块地址写入

0x0050,然后等待擦除的完成。对SST39VF1601的编程采用字编程写入方式,分4个总线周期完成,具体步骤:首先写2个解锁周期,即在地址0x5555写入0x00AA和在地址0x2AAA写入0x0055;紧接一个设置周期,即在地址0x5555写入0x00A0;第四个周期向要写入的地址写入数据,然后等待编程的完成。

#### 5 结语

提出了一种基于TFTP协议的ARM软件远程更新方案,设计并实现了远程更新系统。该系统充分利用了以太网的传输能力和其便捷性。通过实验测试,基于TFTP协议的ARM软件远程更新系统能够保证数据的正确传输和存储。该系统的实现使得对嵌入式系统软件的升级维护难度大大降低,具有较好的应用前景。

#### 参考文献:

- [1] 栗欣,周东辉,孙晓苗,等.单片机程序远程升级的设计[J].微计算机信息,2006,11(2):75-77.
- [2] 姜小梅,李祥和,任朝荣,等.基于ARM的IAP在线及远程升级技术[J].计算机应用,2008,28(2):123-125.
- [3] 单承刚,戴学丰,刘树东,等.基于ARM的嵌入式Bootloader设计与启动过程[J].微计算机信息,2006,11(2):139-142.
- [4] 汪小燕,连晓平,董燕,等.基于TFTP协议的嵌入式系统开发方法设计与实现[J].华中科技大学学报:自然科学版,2006,34(12):56-58.
- [5] 周立功.ARM嵌入式系统基础教程[M].北京:北京航空航天大学出版社,2005.
- [6] STEVENS R W.TCP/IP详解卷1:协议[M].范建华,译.北京:机械工业出版社,2000.

## 泰山建能机械集团被国家工业和信息化部列为 首批矿采机械再制造试点单位

日前,山东泰山建能机械集团大族激光再制造技术有限公司被国家工业和信息化部列为第一批矿采机械再制造试点单位。

泰山建能机械集团针对煤矿大量废旧矿山机械核心零部件(如减速器、电动机、齿轮等),按照循环经济“减量化、再利用、资源化”的原则要求进行再制造,缓解资源短缺与资源浪费的矛盾,实现了变废为宝,避免了重新冶炼加工带来的巨大能源浪费,符合国家资源节约综合利用的产业政策。

该项目采用纳米电刷镀、高速电弧喷涂、纳米自修复添加剂、激光表面熔覆等工艺,对机械核心零部件进行再制造,技术达到国际先进水平,再制造后的产品各项指标均达到新机标准。

该项目已成年再制造10万台套隔爆电动机及配件和5万台套矿用减速器及配件的生产能力,产值为3亿多元,综合节能率为60%,节材70%。

(周峰 王明章)