



(12) 发明专利

(10) 授权公告号 CN 101477471 B

(45) 授权公告日 2014. 04. 30

(21) 申请号 200910300092. 4

CN 1928820 A, 2007. 03. 14, 全文.

(22) 申请日 2009. 01. 07

审查员 孙韬敏

(73) 专利权人 杭州海康威视数字技术股份有限公司

地址 310012 浙江省杭州市西湖区马腾路 36 号(高新区)

(72) 发明人 赵先林 孙彬 栾焕志

(74) 专利代理机构 杭州杭诚专利事务所有限公司 33109

代理人 王鑫康

(51) Int. Cl.

G06F 9/445(2006. 01)

(56) 对比文件

US 6615404 B1, 2003. 09. 02, 全文.

JP 特开 2004-157789 A, 2004. 06. 03, 全文.

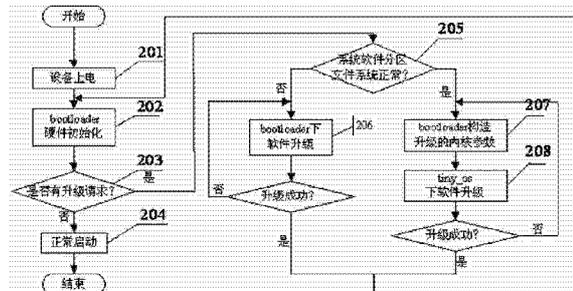
权利要求书2页 说明书6页 附图2页

(54) 发明名称

一种嵌入式系统固件在线升级方法

(57) 摘要

本发明公开了一种嵌入式系统固件在线升级方法,将非失性存储器分为两个分区,将 bootloader 和 tiny_os 所在的分区保护起来,在 tiny_os 下运行升级脚本实现升级,而相应的下载命令由开源的 busybox 提供,文件系统以及 tcp/ip 协议栈等的支持由 tiny_os 提供,而相应的下载命令采用动态链接技术并与系统软件共享,通过 bootloader 向内核传递不同的参数来区分是升级操作还是正常操作,整个升级系统只需增加大约 400K 的存储空间就可支持多种下载协议和文件系统。另外还提供了一种备用的升级方法,当系统软件分区遭到意外的破坏时,仍然可以对系统进行升级,保证升级的可靠性和安全性。



1. 一种嵌入式系统固件在线升级方法,其特征在于,包括以下步骤:
 - (1) 嵌入式系统固件包含一非易失性存储器;
 - (2) 非易失性存储器分为分区 A 和分区 B 两个分区;分区 A 代码采用二进制的文件格式,按地址进行访问;分区 B 采用 linux 支持的、nor flash 上应用较多的 jffs2 文件格式;
 - (3) 分区 A 用于存放 bootloder 和简化操作系统 tiny_os 软件程序;
 - a、所述 bootloder 包括初始化加载模块和升级模块;
 - b、所述简化操作系统 tiny_os 提供 tcp/ip 协议栈以及文件系统的读写支持;
 - (4) 分区 B 用于存放根文件系统和系统软件,其中
 - a、根文件系统包括动态链接库、升级脚本,升级脚本至少包括正常启动代码和升级代码 B;
 - b、系统软件包括嵌入式操作系统和应用软件;
 - (5) 初始化设置:
 - a、分区 A 设置为保护分区,对该分区置于硬件保护和软件保护;
 - b、分区 B 设置为只读分区,该分区中的嵌入式操作系统设置为只读文件;
 - (6) 实现在线升级:
 - a、嵌入式系统上电,运行分区 A 中的 bootloder,完成硬件的初始化;
 - b、检测是否有升级请求,若有,则进入步骤 c,如无升级请求进入步骤 f;
 - c、检测根文件系统代码是否正确,若正确则进入步骤 d,否则进入步骤 e;
 - d、运行分区 A 中 bootloder 的构造内核参数代码来构造升级的内核参数并加载 tiny_os;运行分区 B 中的根文件系统的升级脚本,并根据内核参数运行软件升级代码 B,接受用户输入的升级参数,与升级服务器进行通信,并通过指定协议下载升级文件到 flash,升级完毕回到步骤 a;
 - e、启用备用的升级方法:运行分区 A 中的软件升级代码 A,接受用户输入的升级参数,与升级服务器进行通信并通过指定的下载协议下载升级文件,运行分区 A 中的文件系统格式转化代码将升级文件转换为系统软件所需的文件系统格式后写入 flash;如果 jffs2 分区内容遭到意外的破坏,则继续在分区 A 运行 u-boot 中提供的的备用升级代码,使用 tftp 协议将升级包下载至内存,并通过文件系统转换代码转换成 jffs2 格式,再写入 flash,完成升级;升级完毕后返回到步骤 a;
 - f、启动系统软件的嵌入式操作系统,进行正常工作模式。
2. 根据权利要求 1 所述的一种嵌入式系统固件在线升级方法,其特征在于,所述分区 A 中的 bootloder 的初始化加载模块进一步包括:
 - a、系统初始化代码;
 - b、升级检测代码;
 - c、构造内核参数代码;分区 A 中的 bootloder 的升级模块进一步包括:
 - a、软件升级代码 A;
 - b、文件系统格式转化代码。
3. 根据权利要求 2 所述的一种嵌入式系统固件在线升级方法,其特征在于,所述升级检测代码包括检测设备终端是否有升级请求的代码和检测分区 B 中根文件系统是否是正

确的代码,并且支持分区 B 所采用的文件系统的读操作。

4. 根据权利要求 2 所述的一种嵌入式系统固件在线升级方法,其特征在于,所述文件系统格式转化代码是将下载的升级文件的格式转换成分区 B 所支持的文件系统格式。

5. 根据权利要求 1 或 2 所述的一种嵌入式系统固件在线升级方法,其特征在于,所述构造内核参数代码根据是否有升级请求构造不同的内核参数,并加载 tiny_os,以便根文件系统的升级脚本能够根据所需选择运行相应的软件升级代码。

6. 根据权利要求 1 所述的一种嵌入式系统固件在线升级方法,其特征在于,所述根文件系统为 tiny_os 和系统软件所共用:

a、根文件系统中升级脚本的软件升级代码 B,为使用 tiny_os 进行升级所需的程序;

b、根文件系统中的动态链接库,为 tiny_os 和系统软件所需的程序;

c、根文件系统中的升级脚本的启动代码,为系统软件所需的程序。

7. 根据权利要求 1 或 6 所述的一种嵌入式系统固件在线升级方法,其特征在于,所述的软件升级代码 B 为采用脚本编写的程序,当需要采用新的网络传输协议进行升级时,只需修改该软件升级代码 B 的部分内容和增加相应的下载命令,即可实现升级。

8. 根据权利要求 1 或 2 所述的一种嵌入式系统固件在线升级方法,其特征在于,所述分区 A 和分区 B 的设置是通过以下方法实现:

分区 A 设置为保护分区,是在 bootloader 的初始化代码部分将该分区设置为硬件保护和软件保护的分区;

分区 B 设置为只读分区,是在系统软件的嵌入式操作系统中将该分区设置为只读分区。

9. 根据权利要求 1 所述的一种嵌入式系统固件在线升级方法,其特征在于,所述的备用的升级方法的启用是按照下述判断而确定:

如果分区 A 中的升级检测代码检测到分区 B 中的根文件系统正确,则启动 tiny_os,选择升级脚本中的升级代码 B 进行升级,否则启用备用的升级方法,选择分区 A 中的软件升级代码 A 进行升级。

一种嵌入式系统固件在线升级方法

技术领域

[0001] 本发明涉及嵌入式软件技术领域,更确切的说是一种嵌入式系统固件在线升级方法。

背景技术

[0002] 嵌入式设备的功能主要通过软件来实现和扩展,随着嵌入式设备功能的不断完善,应用的不断增加以及 bug 的不断修正,需要对原有系统中的软件进行不断地升级。升级就是将新的软件写入嵌入式设备的非易失存储器,替代以前有缺陷或者功能不完善的软件。另外,当软件系统遭意外的破坏而无法正常运行,也需要对其进行升级。因此,嵌入式设备中的升级程序一定要安全可靠。

[0003] 嵌入式设备通常是由生产设备的厂商在生产时通过 JTAG 调试器或者专用的 flash 烧写器将 bootloader 烧写至 flash。bootloader 在传统的嵌入式系统中在初始化硬件,加载操作系统的同时,也完成相应的升级任务,但是由于 bootloader 提供的升级功能较弱不支持 tcp/ip 以及主流的 flash 文件系统的写功能,因此无法使升级程序支持丰富的下载协议以及文件系统,因此为了支持更多的下载协议通常都采用裁剪后的操作系统为升级软件提供 tcp/ip 协议栈以及文件系统访问的支持,但是采用操作系统后相应的升级软件也变的更加复杂,并且必须依靠相应的库函数的支持,这样的话必然会导致升级软件占用的非易失性存储器的尺寸增大,增加系统成本,增加开发难度,并且当需要增加对新的协议的支持时必须更新相应的升级软件,从而增加了升级软件被破坏的几率。

[0004] 中国专利文献 CN 1928820A 公开了一种嵌入式系统升级方法,在不扩充 bootloader 的情况下,让系统可以通过 Http、tftp 协议下载软件实现系统升级,通过在 FLASH 增加一个经操作系统裁减后的软件升级模块,来下载升级软件,不需要在 bootloader 中再写驱动程序和网络协议,即直接利用裁减后的操作系统中已有的驱动程序和网络协议来下载升级软件,而且在主文件系统损坏时还可提供系统恢复功能。

[0005] 其不足之处:虽然支持了更多的下载协议,但是没有想到下载的升级文件和系统文件的格式不同后怎么处理,没有对系统文件实施有效的保护方法,也没有很好的提出文件系统意外破坏后的升级方案,文件系统意外破坏后的升级和修复一直以来是本领域技术人员难以攻破的难题,而该发明虽然提到能修复系统继续升级,但对于具体的方案没有充分的公开,能与不能还存在着疑问。

发明内容

[0006] 本发明的目的在不增加系统成本的情况下,提供一种占用存储空间少,开发难度低,开发周期短,安全可靠,并且支持丰富的下载协议和文件系统,还能在系统遭到意外破坏后启用备份升级的一种嵌入式系统固件在线升级方法。

[0007] 为了解决上述技术问题,本发明是通过以下技术方案实现:一种嵌入式系统固件在线升级方法,其特征在于,包括以下步骤:

- [0008] (1) 嵌入式系统固件包含一非易失性存储器；
- [0009] (2) 非易失性存储器分为分区 A 和分区 B 两个分区；
- [0010] (3) 分区 A 用于存放 bootloader 和简化操作系统 tiny_os 软件程序；
- [0011] a、所述 bootloader 包括初始化加载模块和升级模块；
- [0012] b、所述简化操作系统 tiny_os 提供 tcp/ip 协议栈以及文件系统的读写等支持；
- [0013] (4) 分区 B 用于存放根文件系统和系统软件，其中
- [0014] a、根文件系统包括动态链接库、升级脚本，升级脚本至少包括正常启动代码和升级代码 B；
- [0015] b、系统软件包括嵌入式操作系统和应用软件；
- [0016] (5) 初始化设置：
- [0017] a、分区 A 设置为保护分区，对该分区置于硬件保护和软件保护；
- [0018] b、分区 B 设置为只读分区，该分区中的嵌入式操作系统设置为只读文件；
- [0019] (6) 实现在线升级：
- [0020] a、嵌入式系统上电，运行分区 A 中的 bootloader，完成硬件的初始化；
- [0021] b、检测是否有升级请求，若有，则进入步骤 c，如无升级请求进入步骤 f；
- [0022] c、检测根文件系统代码是否正确，若正常则进入步骤 d，否则进入步骤 e；
- [0023] d、运行分区 A 中 bootloader 的构造内核参数代码来构造升级的内核参数并加载 tiny_os；运行分区 B 中的根文件系统的升级脚本，并根据内核参数运行软件升级代码 B，接受用户输入的升级参数，与升级服务器进行通信，并通过指定协议下载升级文件到 flash，升级完毕回到步骤 a；
- [0024] e、启用备用的升级方法：运行分区 A 中的软件升级代码 A，接受用户输入的升级参数，与升级服务器进行通信并通过指定的下载协议下载升级文件，运行分区 A 中的文件系统格式化代码将升级文件转换为系统软件所需的文件系统格式后写入 flash；升级完毕后返回到步骤 a；
- [0025] f、启动系统软件的嵌入式操作系统，进行正常工作模式。
- [0026] 将 bootloader 和经简化后的操作系统（下称 tiny_os）编译链接在一起存放到 flash 上的一个分区，tiny_os 为升级软件提供了 tcp/ip 协议栈以及文件系统的读写等支持。升级程序采用 script 脚本进行编写，在脚本中直接调用 busybox 提供的各种下载协议的程序从服务器下载所需升级的文件，这样可以大大地降低升级程序的编写难度，也大大缩短了开发周期。同时将 busybox 等采用动态链接技术，升级程序所需的根文件系统以及动态链接库，busybox 等与系统软件共用，存放在系统软件的分区中，这样采用该技术后系统只需额外提供几百 KB（大约 400Kbytes）的存储空间即可支持多种下载协议和多种文件系统。同时由于将升级脚本和下载程序与 bootloader 和 tiny_os 分开存放，可以保证当需要修改升级脚本或者增加新的下载协议时无需修改 bootloader 和 tiny_os 所在分区内容，因此可以将该分区设置为硬件保护，在系统的操作系统中也将该分区设置为只读属性，这样就保证了该分区不会被破坏。另外，为了防止系统软件所在分区的文件系统意外地遭到破坏，在 bootloader 中同时提供备用的升级方案，保证做到可靠的升级。
- [0027] 作为优选，所述分区 A 中的 bootloader 的初始化加载模块进一步包括：
- [0028] a、系统初始化代码；

[0029] b、升级检测代码；

[0030] c、构造内核参数代码；

[0031] 分区 A 中的 bootloader 的升级模块进一步包括：

[0032] a、软件升级代码 A；

[0033] b、文件系统格式转化代码。

[0034] 系统引导程序 bootloader 是嵌入式系统上电启动运行的第一段软件代码，是操作系统内核或用户应用程序运行之前，运行的一段子程序。通过这段程序，初始化硬件设备、建立内存空间的映射图，以便为最终调用操作系统内核或用户应用程序准备好正确的环境。即在嵌入式系统中系统引导模块和操作系统相连，引导操作系统运行。

[0035] 作为优选，所述的升级检测代码用于：检测设备终端是否有升级请求的代码和检测分区 B 中根文件系统是否是正确的代码，并且支持分区 B 所采用的文件系统的读操作。

[0036] 升级请求检测由分区 A 中的升级检测代码完成，判断设备终端是否有升级请求输入。如果无升级请求，则系统正常启动；如果有升级请求，则进入升级模式。

[0037] 作为优选，所述文件系统格式转化代码是将下载的升级文件的格式转换成分区 B 所支持的文件系统格式。

[0038] bootloader 一般不支持文件系统的写操作，所以通过在 bootloader 下增加文件系统格式转化代码以支持某种格式文件系统的写操作。

[0039] 作为优选，所述构造内核参数代码根据是否有升级请求构造不同的内核参数，并加载 tiny_os，以便根文件系统的升级脚本能够根据所需选择运行相应的软件升级代码。

[0040] 通过 bootloader 向内核传递不同的参数来区分是升级操作还是正常操作。

[0041] 作为优选，所述根文件系统为 tiny_os 和系统软件所共用；

[0042] a、根文件系统中升级脚本的软件升级代码 B，为使用 tiny_os 进行升级所需的程序；

[0043] b、根文件系统中的动态链接库，为 tiny_os 和系统软件所需的程序；

[0044] c、根文件系统中的升级脚本的启动代码，为系统软件所需的程序。

[0045] 作为优选，所述的软件升级代码 B 为采用脚本编写的程序，当需要采用新的网络传输协议进行升级时，只需修改该软件升级代码 B 的部分内容和增加相应的下载命令，即可实现升级。

[0046] 升级程序采用 script 脚本进行编写，在脚本中直接调用 busybox 提供的各种下载协议的程序从服务器下载所需升级的文件，这样可以大大地降低升级程序的编写难度，也大大缩短了开发周期。同时将 busybox 等采用动态链接技术，升级程序所需的根文件系统以及动态链接库，busybox 等与系统软件共用，存放在系统软件的分区中，这样采用该技术后系统只需额外提供几百 KB(大约 400Kbytes) 的存储空间即可支持多种下载协议和多种文件系统。

[0047] 作为优选，所述的分区 A 和分区 B 的文件格式为：

[0048] 分区 A 的文件采用二进制格式，按地址进行访问；

[0049] 分区 B 采用嵌入式操作系统支持的文件系统格式。

[0050] 分区 A 中的代码是固定代码，不在软件升级的范围之内，软件升级所指为对分区 B 的软件升级。

[0051] 作为优选,所述分区 A 和分区 B 的设置是通过以下方法实现:

[0052] 分区 A 设置为保护分区,是在 bootloader 的初始化代码部分将该分区设置为硬件保护和软件保护的分区;

[0053] 分区 B 设置为只读分区,是在系统软件的嵌入式操作系统中将该分区设置为只读分区。

[0054] 为防止系统软件所在分区的文件系统意外地遭到破坏,在 bootloader 中同时提供备用的升级方案,保证做到可靠的升级,所以对分区 A 进行保护是必要的。

[0055] 作为优选,所述的备用的升级方法的启用是按照下述判断而确定:

[0056] 如果分区 A 中的升级检测代码检测到分区 B 中的根文件系统正确,则启动 tiny_os,选择升级脚本中的升级代码 B 进行升级,否则启用备用的升级方法,选择分区 A 中的软件升级代码 A 进行升级。

[0057] 升级方式分为 tiny_os 下升级和 bootloader 下升级两种。分区 B 的升级代码 B 负责 tiny_os 下的升级;分区 A 的升级代码 A 负责 bootloader 下的升级。分区 B 中的根文件系统完好无损是实现在 tiny_os 下升级的前提,因此先执行分区 B 的代码检测,如果分区 B 中的根文件系统完好,则执行 tiny_os 下升级完成升级,否则执行 bootloader 下升级完成升级。tiny_os 下的升级,采用 tiny_os 支持的一种数据传输协议和文件系统格式,升级软件以文件名的形式写入 flash 存储器,flash 对用户来说是透明的,用户无需关注对 flash 的写操作;bootloader 下的升级,升级代码 A 在 bootloader 下运行,数据传输必须采用 bootloader 支持的一种传输协议,bootloader 一般不支持文件系统的写操作,所以通过在 bootloader 下增加文件系统格式转化代码以支持某种格式文件系统的写操作,最后升级代码 A 将转换后的文件写入 flash 的分区 B。

[0058] 本发明有益的效果是:将非失性存储器分为两个分区,将 bootloader 和 tiny_os 所在的分区保护起来,保证升级失败后的安全性;在 tiny_os 下运行升级脚本实现软件升级,而相应的下载命令由开源的 busybox 提供,文件系统以及 tcp/ip 协议栈等的支持由 tiny_os 提供支持,因此只需编写相应的升级脚本即可完成,开发难度和开发周期大大降低。而相应的下载命令采用动态链接技术并与系统软件共享,通过 bootloader 向内核传递不同的参数来区分是升级操作还是正常操作,所以整个升级系统只需增加大约 400K 的存储空间就可支持多种下载协议和文件系统。另外还提供了一种备用的升级方法,当系统软件分区遭到意外的破坏时,仍然可以对系统进行升级,保证升级的可靠性和安全性。

附图说明

[0059] 图 1 为本发明非易失性存储器 nor flash 的空间分区图;

[0060] 图 2 为本发明整个系统的软件升级模块图;

[0061] 图 3 为本发明软件升级的流程图。

具体实施方式

[0062] 下面以 ARM9 开发板的软件升级为例,结合附图对本发明作进一步说明。

[0063] 如图 1 所示,Flash 的分区图:开发板的软件系统存储在 nor flash 芯片上,根据存储的内容不同,将 flash 空间分为两个分区,分区 A 11 为只读分区,存储不需更新的程

序,包括 bootloder 和 tiny_os,此分区的代码采用二进制的文件格式,按地址进行访问。分区 B 12 存储简化后的系统软件,包括嵌入式操作系统,根文件系统以及应用程序等,在 tiny_os 下升级程序所需的升级脚本以及相应的下载命令程序都在根文件系统内,此分区采用 linux 支持的、nor flash 上应用较多的 jffs2 文件格式。

[0064] 如图 2 所示,整个升级模块之间的关系,其中两个大的实框模块为分区 A11、分区 B12,三个虚线框内的模块为 bootloder 模块 13,根文件系统模块 15 以及 tiny_os 模块 14。嵌入式 OS16,根文件系统 15 以及 tiny_os 14 都是由 bootloder 的初始化加载模块 21 加载到内存的,初始化加载模块 21 包括系统初始化代码、升级检测代码 27、构造内核参数代码 28,如果无需升级则启动系统软件的嵌入式 OS16,进行正常工作模式。当使用 tiny_os 14 进行升级时,加载模块将根文件系统 15 和 tiny_os14 加载到内存并运行 tiny_os14, tiny_os 14 执行根文件系统 15 中的升级脚本 22 的软件升级代码 B26,软件升级代码 B26 通过 tiny_os 14 的 tcp/ip 协议栈 18 和网络驱动 24 将软件升级包 19 下载下来,然后通过 tiny_os 14 提供的文件系统 29 将升级文件通过 flash 驱动 25 写入 flash 20。当使用备用的 bootloder 的升级模块 23 进行升级时,升级模块 23 包括升级代码 A30 和文件系统格式转换模块 17,升级模块 23 用软件升级代码 A 30 调用 bootloder 提供的网络驱动 24 下载软件升级包 19,然后通过文件系统格式转换模块 17 将升级包 19 转换为系统软件所在分区的格式后通过 bootloder 中提供的 flash 驱动 25 将文件写入 flash 20。详细过程如下所示:

[0065] 首先,开发板上移植了用于硬件初始化的 u-boot,由于 u-boot 本身支持多种开发板及多种硬件接口,因此稍微修改 u-boot,使其支持板载串口、nor flash 和网络接口驱动。由于 u-boot 本身支持 TFTP 网络协议,因此 u-boot 下可通过 TFTP 下载升级包并写入 flash 进行升级。为了在 u-boot 中能够支持对 jffs2 的写操作,需在 u-boot 中增加 jffs2 文件系统转换代码将从服务器上下载的文件转换成 jffs2 格式后写入 flash。

[0066] tiny_os 和嵌入式操作系统采用都是 linux,针对此开发板移植了 linux,系统正常的 linux 内核要支持板载的网卡,串口,支持 TCP/IP 协议以及 jffs2 等文件系统外还需根据需求选择系统所需的所有其它功能模块。而对于 tiny_os 只需将该 linux 内核中与网络和 jffs2,串口以及 flash 驱动等无关的代码裁剪掉即可。

[0067] 根文件系统采用 busybox 和 uclibc 进行编译,并且在 busybox 中选择所需的下载协议对应的命令,如 tftp, ftpget 等,修改编译生成根文件系统的启动脚本,增加检查 u-boot 是否传递了升级请求参数的代码,如果有升级请求则调用升级脚本进行软件升级,否则运行正常工作的启动脚本。

[0068] 如图 3 所示,嵌入式系统固件在线升级方法步骤如下:

[0069] 步骤 201:嵌入式系统上电。

[0070] 步骤 202:运行分区 A 中的 bootloder 中的初始化代码,完成硬件的初始化。

[0071] 步骤 203:运行 bootloder 中的升级检测代码检测是否有升级请求,若有,则进入步骤 205,如无升级请求进入步骤 204。

[0072] 步骤 204:启动系统软件的嵌入式操作系统,进行正常工作模式。

[0073] 步骤 205:运行 bootloder 中的升级检测代码检测根文件系统是否正确,若正常则进入步骤 207,否则进入步骤 206。

[0074] 步骤 206 :运行分区 A 中的升级代码 A,接受用户输入的升级参数,与升级服务器进行通信并通过指定的下载协议下载升级文件,运行分区 A 中的文件系统格式转化代码将升级文件转换为系统软件所需的文件系统格式后写入 flash,升级完毕后回到步骤 202。

[0075] 步骤 207 :运行分区 A 中 boot loader 的的构造内核参数代码来构造升级的内核参数。

[0076] 步骤 208 :加载 tiny_os,运行分区 B 中的根文件系统的升级脚本,并根据内核参数运行软件升级代码 B,接受用户输入的升级参数,与升级服务器进行通信,并通过指定协议下载升级文件到 flash,升级完毕回到步骤 202。

[0077] 步骤中分区 A 是受保护分区,在 boot loader 的初始化代码部分将该分区设置为硬件保护和软件保护,在系统软件的嵌入式操作系统中也将该分区设置为只读分区。分区 A 文件采用二进制格式,所述的分区 B 采用嵌入式操作系统支持的文件系统格式。分区 B 中的软件升级代码 B,采用脚本编写,当需要新的网络传输协议进行升级时,只需修改该部分内容和增加相应的下载命令即可实现。

[0078] 实例中开发板升级的流程如下 :升级前,将主机与开发板建立网络连接,在主机上建立 tftp 或 ftp 服务器,并将升级程序包放在服务器目录下。将主机与开发板通过串口连接查看打印信息。板上电复位后,首先运行 u-boot,完成硬件的初始化并加载相关的硬件驱动程序。当 u-boot 初始化结束后,通过串口终端检测是否有键盘上的升级请求的按键按下,若有,则进入软件升级流程,首先检测 jffs2 分区中的文件是否完好,如果是,则构造升级的内核参数并启动 tiny_os, tiny_os 加载后运行根文件系统中的升级脚本,升级脚本检查到升级的内核参数后运行软件升级代码 B,在升级代码 B 中通过调用 busybox 提供的 tftp 或者 ftpget 命令通过文件系统将升级文件从服务器下载至 flash,完成升级。如果 jffs2 分区内容遭到意外的破坏,则继续在分区 A 运行 u-boot 中提供的的备用升级代码,使用 tftp 协议将升级包下载至内存,并通过文件系统转换代码转换成 jffs2 格式,再写入 flash,完成升级。如果未检测到升级按键按下,则启动分区 B 中 linux 和应用程序,系统正常运行。

[0079] 以上所述的仅是本发明的优选实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明核心技术特征的前提下,还可以做出若干改进和润饰,这些改进和润饰也应视为本发明的保护范围。

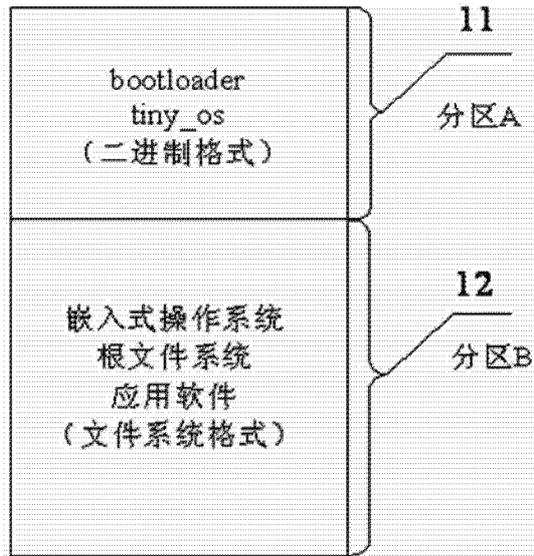


图 1

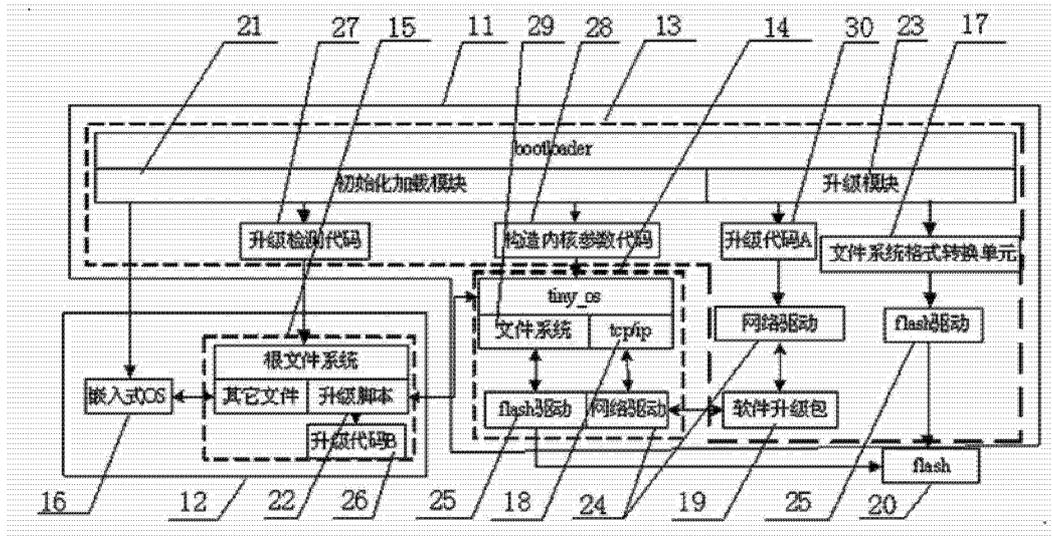


图 2

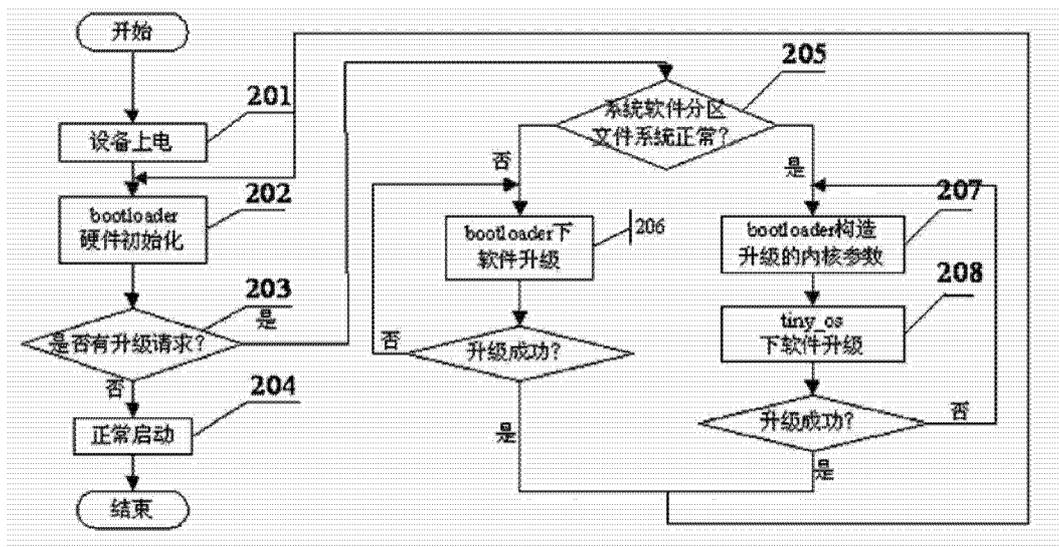


图 3