(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102902556 A
(43)申请公布日 2013.01.30

(21)申请号 201210326854.X

(22)申请日 2012.09.06

(71)申请人 深圳市共进电子股份有限公司
地址 518067 广东省深圳市南山区南海大道1019号南山医疗器械产业园B411-413

(72)发明人 刘宏钧

(74)专利代理机构 深圳汇智容达专利商标事务所(普通合伙) 44238
代理人 赵蕊

(51)Int.Cl.
G06F 9/445(2006.01)

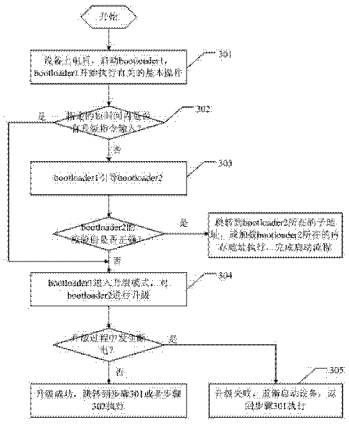
权利要求书 1 页 说明书 5 页 附图 2 页

(54)发明名称

一种嵌入式设备的多级引导加载方法

(57)摘要

本发明提供了一种嵌入式设备的多级引导加载方法，所述嵌入式设备的flash依次划分为bootloader2分区、image分区和data配置区，所述多级引导加载方法包括步骤：在存储区中添加bootloader1分区，该bootloader1分区具有升级bootloader2、引导bootloader2以及配置bootloader2的功能；设备上电后，先启动bootloader1；在指定时间内，循环检查是否有升级/配置bootloader2指令输入，若无，则bootloader1引导bootloader2启动，完成启动流程；若有，则bootloader1对bootloader2进行升级/配置，升级/配置之后跳转至bootloader2执行或者直接重启设备。本发明实施例中不仅支持image和bootloader的多样性，而且满足嵌入式升级时的安全性要求。



CN 102902556 A

1. 一种嵌入式设备的多级引导加载方法，所述嵌入式设备的 flash 依次划分为 bootloader2 分区、image 分区和 data 配置区，其特征在于，所述多级引导加载方法包括步骤：

在存储区中添加 bootloader1 分区，该 bootloader1 分区具有升级 bootloader2、引导 bootloader2 以及配置 bootloader2 的功能；

设备上电后，先启动 bootloader1；

在指定时间内，循环检查是否有升级 / 配置 bootloader2 指令输入，若无，则 bootloader1 引导 bootloader2 启动，完成启动流程；若有，则 bootloader1 对 bootloader2 进行升级 / 配置，升级 / 配置之后跳转至 bootloader2 执行或者直接重启设备。

2. 如权利要求 1 所述嵌入式设备的多级引导加载方法，其特征在于，所述 bootloader1 分区和 bootloader2 分区位于不同 flash 上；其中，bootloader1 分区设于 spi nor flash 上，bootloader2 分区设于 nand flash 上。

3. 如权利要求 1 所述嵌入式设备的多级引导加载方法，其特征在于，所述 bootloader1 分区和 bootloader2 分区位于同一 flash 上。

4. 如权利要求 1 所述嵌入式设备的多级引导加载方法，其特征在于，所述 bootloader1 分区占用 1 个 flash 的 1 个 block。

5. 如权利要求 1 所述嵌入式设备的多级引导加载方法，其特征在于，所述 bootloader1 分区设置有写保护。

6. 如权利要求 1 至 5 任一所述嵌入式设备的多级引导加载方法，其特征在于，该方法中，bootloader1 引导 bootloader2 启动的过程为：

bootloader1 从约定的地址找到对应的 block，再根据找到的地址读入 bootloader2 的信息，读取 bootloader2 的长度和预设校验值信息；

bootloader1 读取出 bootloader2 的全部内容并计算其校验值，将该校验值和预设校验值比较判断得出该校验值是否正确；如果校验值正确则跳转到 bootloader2 可执行代码所对应的地址，或跳转到 bootloader2 加载到内存并解压后对应的地址执行，完成启动流程；如果不正确，则 bootloader1 转换至升级模式，对 bootloader2 进行升级，升级后跳转至 bootloader2 执行或者直接重启设备。

7. 如权利要求 1 至 5 任一所述嵌入式设备的多级引导加载方法，其特征在于，该方法中，所述 bootloader1 对 bootloader2 进行升级的方式包括三种：串口升级方式、tftp 协议升级方式、http 或 ftp 协议升级方式。

一种嵌入式设备的多级引导加载方法

技术领域

[0001] 本发明涉及嵌入式技术领域，尤其涉及一种嵌入式设备的多级引导加载方法。

背景技术

[0002] 嵌入式设备具有两个显著的特点：一个是尽量降低成本，需要裁减掉多余的软硬件功能；一个是需要定制以满足多样化的需求，统一标准的方式难以满足所有的需求。

[0003] 作为程序的主要部分 image（程序镜像，通常包括 kernel（内核）和 rootfs（根文件系统））也是多样化的，随着技术的发展 image 本身的形式在不断发展，且有双 image，大小 image 等应用形式的出现，导致原有的 bootloader（启动加载程序）已经不能满足新的要求，不能兼容，因此 bootloader 本身也需要进行升级。有时，有关的关键参数如 MAC 地址，产测配置嵌入到 bootloader 本身，在参数扩充或修改时也需要对 bootloader 进行升级。

[0004] 嵌入式的 bootloader 通常是存储在 flash 上的，而传统的 bootloader 升级方法是：在升级 bootloader 时，需要把 bootloader 在 flash 上对应的 block 上的数据擦除掉，再从 ram 写入，这时可能因为断电和其他意外导致升级终止，而 bootloader 所在的 block 由于尚未写入数据或写入不完全导致损坏，上电重启就无法再从对应的 flash block 读出 bootloader 代码而无法运行，从而使嵌入式设备陷入瘫痪，俗称变成了砖头。这种情况维修和升级的成本都会很高，或者是永久性的损坏。

[0005] 因此必须找到一种方法来同时满足灵活性和安全性的要求。

发明内容

[0006] 本发明的目的在于提供一种嵌入式设备的多级引导加载方法，采用两个 bootloader 级联来引导 image，同时支持 image 和 bootloader 的多样性，又满足嵌入式升级时的安全性要求，在 bootloader 升级失败后仍然能够引导运行。

[0007] 本发明的目的是通过以下技术方案实现的。

[0008] 一种嵌入式设备的多级引导加载方法，所述嵌入式设备的 flash 依次划分为 bootloader2 分区、image 分区和 data 配置区，所述多级引导加载方法包括步骤：

在存储区中添加 bootloader1 分区，该 bootloader1 分区具有升级 bootloader2、引导 bootloader2 以及配置 bootloader2 的功能；

设备上电后，先启动 bootloader1；

在指定时间内，循环检查是否有升级 / 配置 bootloader2 指令输入，若无，则 bootloader1 引导 bootloader2 启动，完成启动流程；若有，则 bootloader1 对 bootloader2 进行升级 / 配置，升级 / 配置之后跳转至 bootloader2 执行或者直接重启设备。

[0009] 其中，所述 bootloader1 分区和 bootloader2 分区位于不同 flash 上；其中，bootloader1 分区设于 spi nor flash（串行外围接口或非门的闪存）上，bootloader2 分区设于 nandflash（与非门闪存）上。

[0010] 其中，所述 bootloader1 分区和 bootloader2 分区位于同一 flash 上。

[0011] 其中，所述 bootloader1 分区占用 1 个 flash 的 1 个 block。

[0012] 其中，所述 bootloader1 分区设置有写保护。

[0013] 其中，bootloader1 引导 bootloader2 启动的过程为：

bootloader1 从约定的地址找到对应的 block，再根据找到的地址读入 bootloader2 的信息，读取 bootloader2 的长度和预设校验值信息；

bootloader1 读取出 bootloader2 的全部内容并计算其校验值，将该校验值和预设校验值比较判断得出该校验值是否正确。如果校验值正确则跳转到 bootloader2 可执行代码所对应的地址，或跳转到 bootloader2 加载到内存并解压后对应的地址执行，完成启动流程；如果不正确，则 bootloader1 转换至升级模式，对 bootloader2 进行升级，升级后跳转至 bootloader2 执行或者直接重启设备。

[0014] 其中，所述 bootloader1 对 bootloader2 进行升级的方式包括三种：串口升级方式、tftp（tftp 简单文件传输协议）协议升级方式、http（超级文本传输协议）或 ftp（文件传输协议）协议升级方式。

[0015] 与现有技术相比，本发明实施例具有以下有益效果。

[0016] 本发明实施例中新增了一个 bootloader1 分区，具有引导、配置、升级原有的 bootloader2 分区的功能，在设备系统启动时 bootloader1 分区与 bootloader2 分区级联来进行引导加载，这样在 bootloader2 在损坏或者升级失败时，bootloader1 由于被写保护而不会被损坏，因而 bootloader1 可对 bootloader2 重新进行升级，不仅支持 image 和 bootloader 的多样性，而且满足嵌入式升级时的安全性要求。

附图说明

[0017] 图 1 是现有的 flash 分布结构示意图。

[0018] 图 2 是本发明实施例提供的 flash 分布结构示意图。

[0019] 图 3 是本发明实施例提供的嵌入式设备的多级引导加载方法流程图。

[0020] 图 4 是本发明实施例提供的升级方法流程图。

具体实施方式

[0021] 为了使本发明的目的、技术方案及优点更加清楚明白，以下结合附图及实施例，对本发明进行进一步详细说明。应当理解，此处所描述的具体实施例仅仅用以解释本发明，并不用于限定本发明。

[0022] 请参阅图 1，现有的 flash 按顺序依次划分为一个 bootloader 分区、一个 image 分区（实际也可为多个 image 分区）和一个 data 配置区。其中 bootloader 分区用于存储 bootloader 程序本身和 bootloader 所需要的配置参数；image 分区通常包括 kernel 和 rootfs 两部分；data 配置区为可选，包括系统运行需要的配置参数和运行的记录信息，有时可能包括在 rootfs 里。

[0023] 与现有的 flash 分布结构不同，本实施例中在现有的 bootloader 分区之前添加一个新的 bootloader 分区，新的 flash 分布结构如图 2 所示，包括两个 bootloader 分区（分别称为 bootloader1 分区和 bootloader2 分区）、一个 image 分区（实际也可为多个 image 分区）和一个 data 配置区。下面将对 bootloader1 分区和 bootloader2 分区分别进行描述。

[0024] bootloder1 分区：具有引导 bootloader2、升级 bootloader2 以及配置 bootloader2 的有关参数的功能；其中的升级功能可以包含最基本的串口升级方式，也可以包含网口升级方式等更高级的方式，详见下文。该 bootloder1 不支持直接引导或升级 image 和修改 data 配置区。bootloder1 的大小受到限制，通常为 1 个 flash 的 1 个 block 大小，有时可能包括一个以上，所占 flash 空间比 bootloder2 要小很多。

[0025] bootloader2 分区：保持现有单个 bootloader 的有关特性基本不变，其作用就是做一些基本的硬件、运行环境的初始化，解压 kenel 部分的代码，并把 cpu 的控制权交给 kenel，从而实现对引导操作系统的 kenrnel 部分的引导，也包含升级自身和 kernel，或 bootloader1 除外的整个 flash 的功能。

[0026] 在实际应用中，bootloder1 分区和 bootloader2 分区可以不在同一个 flash 上，比如：bootloader1 分区设于 spi nor flash，而 bootloader2 分区设于 nandflash 上。

[0027] 通常，bootloder1 是写保护的，采用 flash 支持的写保护方式（可以是软件方式和硬件方式）。bootloader2 没有权限或不能修改 bootloader1 的内容，该限制可以是通过代码中规定并在 bootloader1 设置写保护来实现。bootloader2 如果修改 flash 的写保护方式必须保证 bootloader1 相关的 block 的写保护状态不改变。而 bootloader2 通常也不需要对 bootloder1 进行读取。

[0028] 本实施例中，添加设置 bootloader1 分区的目的就是：在嵌入式设备上电后先跳转到 bootloder1 执行，完成对 bootloader2 的升级或者配置后引导 bootloader2。bootloader1 引导 bootloader2 时通过固定的起始地址来加载 bootloader2，并在前面指定地址读取 bootloader2 的长度和校验值。该长度和校验值会根据实际情况变化而变化。如果采用 nandflash，不论 bootloader2 是否处于文件系统中，bootloader1 如果遇到坏块进行跳过在前面若干个指定的 block 范围内查找，直到根据特征值找到 bootloader2 位置。

[0029] 请参阅图 3，本实施例中，基于上述具有两个级联的 bootloader 分区的 flash 分区结构，嵌入式设备的引导加载方法包括以下步骤。

[0030] 301、设备上电后，从指定地址启动 bootloader1，bootloader1 开始执行有关的 bootloader 基本操作，如初始化内存、定时器，关中断，初始化串口、按键。

[0031] 302、在指定的短时间内（如 1 秒钟），循环检查是否有升级指令输入（升级指令可通过指定按键输入或者串口按键输入；按键输入是指：板上有真实的按键，按下去会有电平变化报告给 cpu，并有对应驱动处理；串口按键指：板上有串口，通过 RS232 转接 pc 机，pc 机的键盘有输入时会把输入的字符或命令信息通过串口传到嵌入式板上），如果没有升级指令输入则进入步骤 303，如果有升级指令输入则进入步骤 304。

[0032] 配置命令一般来自串口按键输入，本例是实现的通常举例说明，实际上如果有需求，可使用按键来完成特定的配置动作，如恢复默认配置的情况也可能存在。

[0033] 303、bootloader1 引导 bootloader2，该引导过程具体如下。

[0034] 从约定的地址（norflash 情况）或根据指定的规则找到对应的 block（如 nandflash），再根据找到的地址读入 bootloader2 的信息，读取出长度和预设校验值信息，之后读取出全部的 bootloader2 内容并计算校验值，将该校验值与预设校验值比较判断该校验值是否正确（预设校验值在编译时生成，在烧录 flash 时已写入到 flash 上，校验值是根据不包括预设校验值的 flash 上的内容在启动时由 cpu 计算得出的，校验值和预设校验

值相同则表明校验值正确)。如果校验值正确则跳转到 bootloader2 可执行代码所对应的地址(位于 flash 上),或跳转到 bootloader2 加载到内存后并解压(如果有需要)后对应的地址(位于内存 ram 上)执行,完成启动流程。如果不正确则跳转到步骤 304 执行。

[0035] 304、bootloader1 进入升级模式,在串口打印出命令提示,bootloader1 升级功能必须支持下面三种升级方式之一,或全部支持,可以根据实际的体积和功能需要取舍,升级过程如图 4 所示。这些升级方式的获取文件所采用的协议方式和通常在正常有 bootloader2 所用的协议实现是一致的,对具体实现方式不具体描述。

[0036] 升级方式①:采取串口升级,需要在串口输入命令,采用的协议为 x-moden, y-modem, z-moderm 系列协议,下载文件到内存,然后升级改写 flash,升级完成后可以手动执行命令或跳转到 bootloader2 或自动重启设备。该方法实现简单,代码体积小,为推荐的方式和一般的方式。

[0037] 升级方式②:使用 tftp 升级,需要在串口输入命令采用的协议为 tftp 协议,bootloader 需要网络驱动和网络协议支持,较复杂,tftp 下载到内存后升级改写 flash,升级完成后可以手动执行命令或跳转到 bootloader2(即跳转到步骤 303 执行)或自动重启设备(即跳转到步骤 301 执行)。

[0038] 升级方式③:使用 http 或 ftp 协议支持,不需要直接在串口操作,需要网卡驱动和 http 服务,在浏览器中打开预定的 http 地址,选择升级文件下载后即会自动升级,跳转到 bootloader2(即跳转到步骤 303 执行)或自动重启设备(即跳转到步骤 301 执行)。

[0039] 不管采用何种升级方式,bootloader1 在对 bootloader2 进行升级时不影响和改变 bootloader1 本身,且不直接升级 image(kernel 和 rootfs) 和 data 配置区,只支持升级 bootloader2 和 bootloader2 有关的参数。如果 bootloader2 存在写保护,则先去掉对应 block 的写保护,写完后对除 bootloader2 之外的写保护状态保持原有状态。

[0040] 305、当使用 bootloader2 对自身和 kernel 和 rootfs 等其他部分进行升级时,如果发生断电等意外的情况时 bootloader2 会损坏而无法启动,更谈不上正确引导 kernel。发生这种情况时,重新启动设备,返回步骤 301 执行。

[0041] 在该步骤中,当重新上电 bootloader1 起来后由于会对 bootloader2 进行完整性校验,检验时会发现 bootloader2 损坏,会进入升级模式,并调用 led 灯位显示 bootloader2 损坏,提醒用户手动对 bootloader2 进行升级,bootloader2 升级后再重启后启动 bootloader2 对 kernel 部分进行升级。这时即使 bootloader2 即使再次升级失败,由于 bootloader1 完好,下次依然同样可以启动 bootloader1,再次进行升级。

[0042] 在本实施例中,bootloader1 可以不需要支持对自身的升级。这时候的情况可以把 bootloader1 视作嵌入到 cpu 内部的 otp flash (一次性编程闪存),因此是不需要有升级功能的。如果确有需要,修改引导 bootloader2 有关的参数也可以实现对 bootloader1 的升级。一般地,这些参数是不需要更改的,就是默认值可以能够引导 bootloader2,如果需要调整有关参数,则需要把参数单独占用一个 block,该 block 的位置在 bootloader1 的 block 之后,在 bootloader2 的 block 之前。该 block 只由 bootloader1 来修改,不允许 bootloader2 来修改。能够通过修改参数的尽量通过修改参数来解决。对于修改参数仍不能解决的情况,需要对 bootloader1 自身所在的 block 进行升级,升级时需确保一定不会出现断电的情况,如果发生断电仍然会导致 bootloader1 自身损坏,升级前需要对该 block 去

写保护，升级功能为减少复杂性，仅对 bootloader1 进行升级写完后再对该 block 加上写保护。该升级功能原则上禁止使用，使用可以进行限制，比如需要特权密码限制，进入该模式时需要同时按几个键，并需要多几个步骤，输入几次指令进行确认，以保证安全性。

[0043] 以上所述仅为本发明的较佳实施例而已，并不用以限制本发明，凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等，均应包含在本发明的保护范围之内。

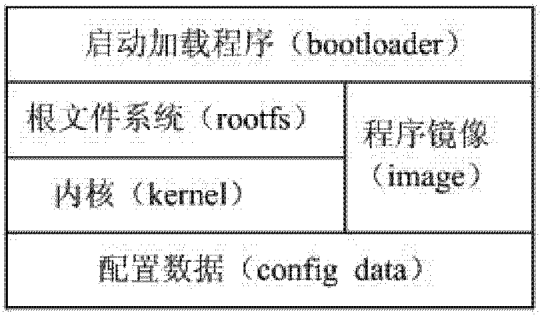

启动加载程序（bootloader）
根文件系统（rootfs）
内核（kernel）
程序镜像（image）
配置数据（config data）

图 1

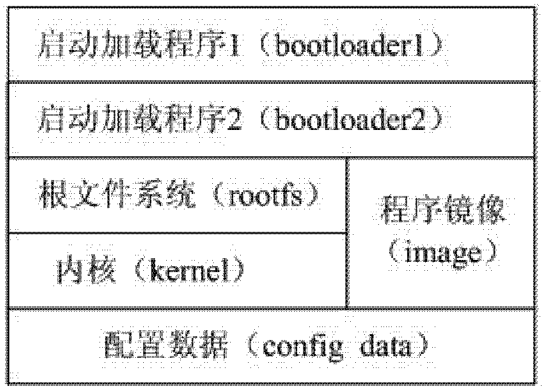

启动加载程序1（bootloader1）
启动加载程序2（bootloader2）
根文件系统（rootfs）
内核（kernel）
程序镜像（image）
配置数据（config data）

图 2

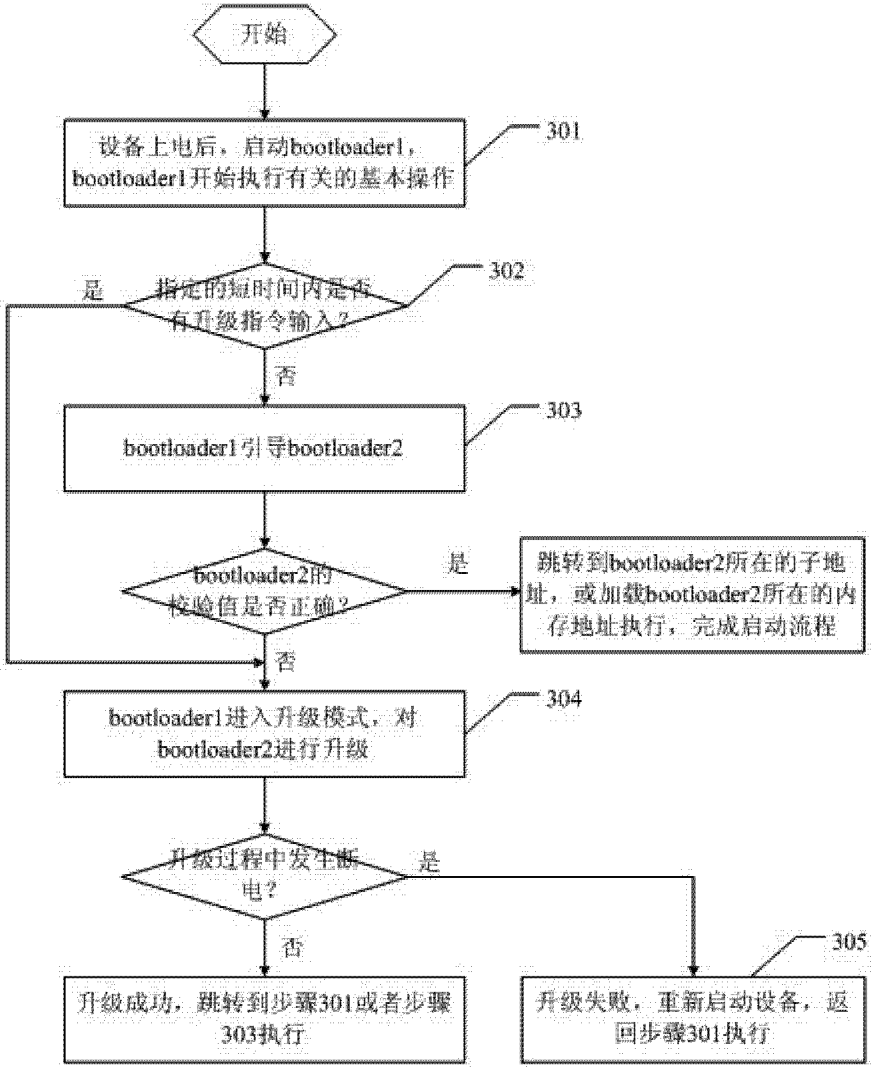

开始
设备上电后，启动bootloader1，bootloader1开始执行有关的基本操作
301
指定的短时间内是否有升级指令输入？
302
是
否
bootloader1引导bootloader2
303
bootloader2的校验值是否正确？
是
跳转到bootloader2所在的子地址，或加载bootloader2所在的内存地址执行，完成启动流程
否
bootloader1进入升级模式，对bootloader2进行升级
304
升级过程中发生断电？
是
否
升级成功，跳转到步骤301或者步骤303执行
升级失败，重新启动设备，返回步骤301执行
305

图 3

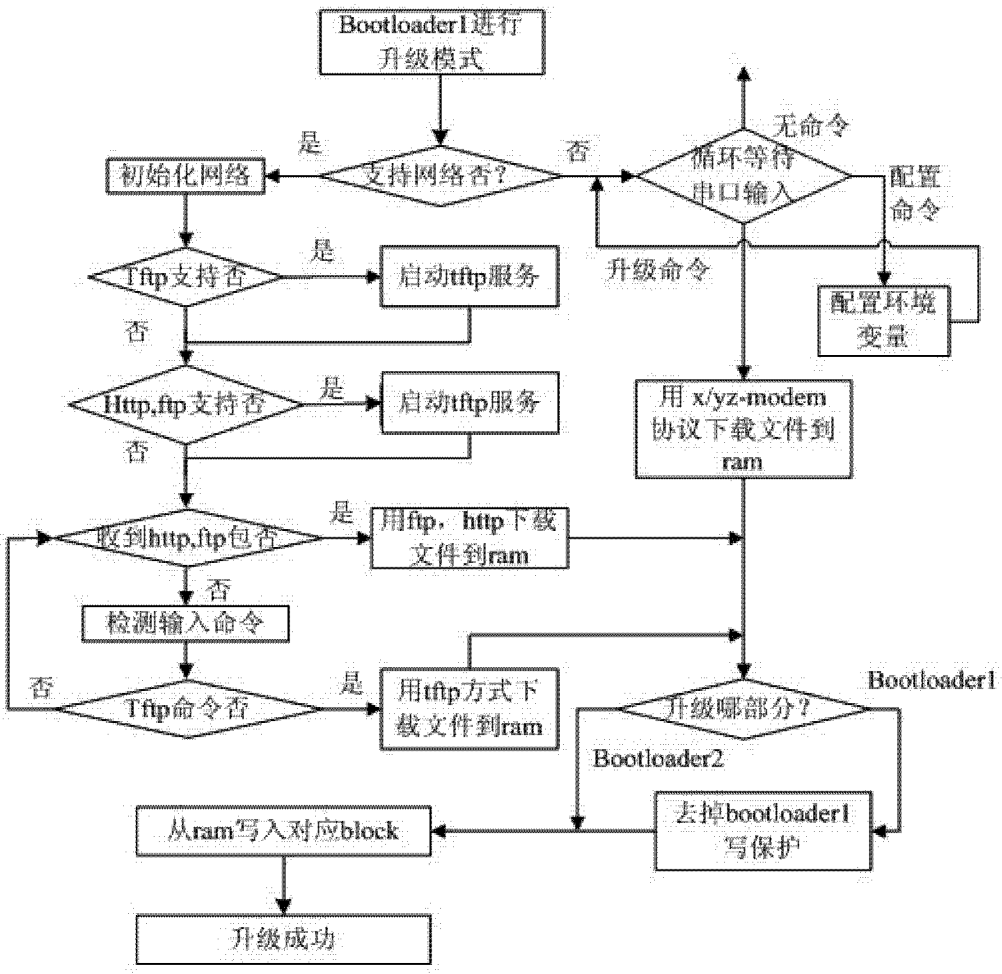

Bootloader1进行升级模式
支持网络否?
是
否
初始化网络
循环等待串口输入
无命令
配置命令
升级命令
Tftp支持否
是
启动tftp服务
否
配置环境变量
Http,ftp支持否
是
启动tftp服务
否
用 x/yz-modem 协议下载文件到ram
收到http,ftp包否
是
用ftp，http下载文件到ram
否
检测输入命令
否
Tftp命令否
是
用tftp方式下载文件到ram
升级哪部分?
Bootloader1
Bootloader2
从ram写入对应block
去掉bootloader1写保护
升级成功

图 4