高可靠性单片机固件程序远程升级方法

刘京威，李惠琼，武新，田富俊
重庆电子科技职业大学 电子与物联网学院，重庆 401331

DOI: 10.61369/SSSD.2025100018

摘　　要： 本文针对大容量闪存单片机提出一种固件程序远程升级的高可靠性方法，采取多种措施规避远程升级过程中的各种错误导致系统无法运行的风险，能有效提升远程升级的可靠性。本方案将单片机内部闪存划分为4个区域，分别为保底固件区、升级固件1区，升级固件2区、参数保存区。在升级固件运行不正常的情况下可以自动或通过人工干预的方式回到保底固件继续运行待重新尝试升级，避免升级不成功反而遭受不可恢复型破坏的风险。升级固件运行时仍然可以再次升级，且采取双区交替写入的方式进行升级。上述方法在小华半导体有限公司产品 HC32F460KE 单片机上实施并运行成功，本文详细介绍了各部分设计概要并给出具体实施案例。

关 键 词： 单片机；微控制器；固件程序；远程升级；HC32F460

High Reliability Method for Remote Upgrade of MCU Firmware

Liu Jingwei, Li Huiqiong, Wu Xin, Tian Fujun
Chongqing Polytechnic University of Electronic Technology, Chongqing 401331

Abstract: This paper proposes a high reliability method for remote upgrade of firmware program for large-capacity flash MCU. It can effectively improve the reliability of remote upgrade by taking various measures to avoid the risk of system failure caused by various errors in the process of remote upgrade. In this scheme, the internal flash memory of the MCU is divided into four areas, namely, the guaranteed firmware area, the upgrade firmware area 1, the upgrade firmware area 2, and the parameter storage area. In case of abnormal operation of the upgraded firmware, you can automatically or manually intervene to return to the guaranteed firmware and continue to run until you try to upgrade again to avoid the risk of irreversible damage caused by unsuccessful upgrade. The upgrade firmware can still be upgraded again when running, and the upgrade is carried out by means of alternate writing in two zones. The above methods have been implemented and run successfully on HC32F460KE MCU of Xiaohua Semiconductor. This paper introduces the design outline of each part in detail and gives the specific implementation case.

Keywords: one-chip computer; MCU; firmware; remote upgrade; HC32F460

引言

随着物联网技术的发展，越来越多的单片机系统具备了连接互联网的能力，因此通过网络远程升级单片机固件程序成为趋势，可极大节省现场升级程序所消耗的人力物力[1-4]。与现场升级程序[5-7]相比，远程升级存在一定风险，比如数据传输错误、数据传输中断、升级固件本身有错误等问题。这些错误不仅可能导致升级失败，甚至可能导致原本能够运行的程序不能再运行，俗称“变砖”。传统基于引导加载程序（Bootloader）的远程升级方式[8]也无法避免这个问题。

本文提出一种高可靠性远程升级方法，采取保底固件加交替升级的方式避免升级过程中可能出现的问题，在确保接收到的固件完整性的情况下才进行跳转。针对升级固件本身有错误的情形，本文也设计了自动或人工干预的方式回到保底固件运行待重新升级。上述措施能够显著提升固件程序远程升级的可靠性。

一、概述

将单片机的闪存区域划分为4个区域，分别为保底固件区、升级固件1区，升级固件2区，参数保存区。设备出厂时将保底固件烧写到保底固件区。保底固件不同于一般意义上的引导加载程序（Bootloader），它是一个具备所有功能的完整程序，除正常的业务应用程序之外，还包括通过网络获取升级固件的功能、检验并跳转到升级固件的功能、监测升级固件是否正常运行的功能。因

项目信息：本文受重庆市教育委员会科学技术研究计划青年项目（KJQN202303102）资助。
作者简介：刘京威（1981年—），男，高级工程师/副教授/硕士；主要从事物联网应用、传感技术、嵌入式系统设计等方向科研及教学工作。

为保底固件功能具有完整性，因此一般情况下设备运行保底固件即可，在非升级不可时才通过网络进行固件升级。

设备上电或复位之后，始终首先运行保底固件启动阶段程序。启动阶段程序根据单片机自身复位原因、是否有用户现场干预、参数保存区内的升级固件参数信息这三个因素判断是否跳转到升级固件运行。一旦有条件不满足，则继续运行保底固件，保证设备基本功能运行，最大限度避免数据传输错误、数据传输中断、数据写入错误等原因造成设备运行故障。

保底固件正常运行时，一旦从网络接收到固件升级请求，首先判断目前是否满足可升级的条件，如果满足则从网络接收升级固件的二进制码，写入到约定的升级固件区（1区或者2区）。二进制码接收完毕之后，保底固件程序检验数据接收的完整性，如果接收完整无差错，则向参数保存区写入相应的标志（包括有效的升级固件在哪个区，固件二进制码完整性校验值等信息），然后复位。复位之后仍然首先运行保底固件启动阶段程序，综合判断跳转条件，如果满足则跳转到升级固件运行。

升级固件正常运行时，如果再次从网络接收到固件升级请求，首先判断目前是否满足可升级的条件，如果满足则从网络接收升级固件的二进制码，写入到另一个升级固件区（如果目前1区固件正在运行，则升级固件写入到2区，因为固件不能在运行的时候同时修改自己）。除此之外，升级的其余流程与保底固件一样。总体来说，可升级固件采取交替升级方式，能够有效避免应用程序修改闪存带来的风险。

二、保底固件

保底固件烧写在单片机上电启动区，上电即运行。对于大多数单片机而言，一般就是从闪存地址0开始的区域。保底固件运行大致分为两个阶段，第一阶段判断条件信息，检查是否具备跳转到升级固件的条件，如果具备条件则跳转到升级固件继续运行，否则继续运行保底固件第二阶段的程序，即业务应用程序。

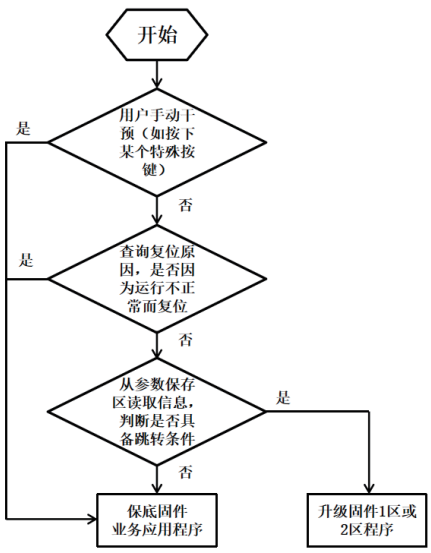

图1 保底固件跳转判断流程

保底固件流程如图1所示。在第一阶段首先判断是否存在用户手动干预，例如按下某个特殊的按键。这项设计给用户或维护人员一个手动干预的机会，在远程升级不成功的情况下，用户可就近手动干预，继续运行保底应用程序。保底应用程序具备通过网络获取升级固件的功能，可以再次尝试远程升级。

随后检查单片机的复位原因，如果是正常情况下的复位，则继续往下判断是否具备跳转条件。如果是非正常复位，例如看门狗导致的复位，表明升级固件的运行有问题，因此也应该继续运行保底应用程序。这项设计在其它检验全部通过，但是升级固件程序本身不能正常运行时，不用手动干预也能自动恢复到保底固件运行。笔者在使用小华半导体（原华大半导体）有限公司产品 HC32F460KE 单片机[9]时，利用其内部的硬件看门狗实现了此功能。

在判断是否具备跳转条件时，首先在参数保存区读取“区域参数”决定跳转到升级固件1区还是2区，然后读取1区或2区的“文件长度参数”以及“校验值”。接下来从1区或2区首地址读取二进制数据计算其校验值，与参数保存区内事先保存的校验值对比看是否一致。如果校验值一致，证明升级固件区内的数据是完整的，可以跳转到升级固件尝试运行。

三、升级固件

升级固件在由程序维护人员编译发布。程序员根据下次升级的区域（1区或2区）将程序链接到不同的地址，编译形成二进制文件之后提供给网络服务器管理人员。远程的单片机系统随后利用网络传输适时地下载更新。

除开业务功能更新之外，升级固件与保底固件应该有三个不同之处。一是程序的链接地址不一样，链接地址在编译器的图形界面或者编译脚本里设置。保底固件链接在单片机的启动地址，而升级固件根据实际情况链接在1区或2区的首地址。二是固件程序的版本标识不一样，通过本地或远程进行查看固件的版本标识可以判断是否升级成功。三是升级固件直接连续运行，不需要在启动阶段进行跳转判断，避免进入跳转死循环。

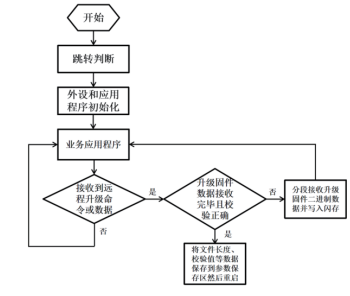

图2（a）保底固件程序流程

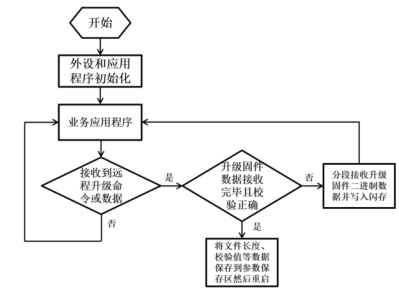

图2（b）升级固件程序流程

升级固件与保底固件主要程序流程如图2所示，保底固件多一个启动阶段的跳转判断过程，其余流程一致，这意味着升级固件在运行期间仍然具备再次升级的能力，而不必回到保底固件执行升级。但是升级固件再次升级时要将接收到的数据写入另一个固件区，因为一般而言单片机程序不能在运行时修改自身代码，这也是升级固件设置为两个区交替升级的由来。当然，如果将闪存内的程序复制到内存（RAM）中运行，则可避免运行时不能修改自身的问题，但是这种方式处理起来程序更为复杂，更容易出现程序缺陷（bug）。

保底固件和升级固件在运行业务应用程序期间如果接收到网络服务器发来的固件升级命令，则分配一定的时间处理接收到的数据，并写入到单片机内部闪存之中。所有数据接收完毕，则读取所有保存的数据进行本地校验，将此校验值与通过升级命令发送而来的校验值进行对比，如果一致则表明接收的所有数据完整无误，将文件长度和校验值记录到参数保存区之后，即可重启尝试运行升级固件；如果不一致，则表明数据接收有误，放弃本次升级，不影响程序继续运行。

上述关于数据传输的完整性校验，只能保证从网络服务器接收到的数据是完整正确的，但它是否能够正确运行是另一个层面的问题，因此需要监测升级固件的运行情况。如前文所述，一种可能的实现方法是启用硬件看门狗，在保底固件启动阶段设置并使能硬件看门狗，跳转到升级固件运行之后，如果升级固件不能正确喂狗，表明升级固件不能正常运行，看门狗超时重启之后回退到保底固件运行。前文提到小华半导体 HC32F460KE 单片机内具有硬件看门狗外设，可以方便地实现这个功能。内部没有硬件看门狗的单片机，也可以外接看门狗芯片实现此功能。

四、具体实现方式

本文提出升级方法具有较强的通用性，而具体工程项目单片机系统和网络服务器的设计千差万别，笔者在总结实际项目的基础上给出一种具体的实现方式，其硬件连接如图3所示。

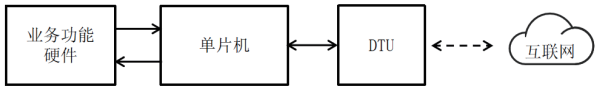

图3 硬件连接示意图

单片机系统除业务功能所需硬件之外，通过串口（UART）连接数据传输单元（DTU），DTU 具有通过有线或无线方式连接互联网的能力。目前市面上这类 DTU 产品众多，具备的功能多种多样，但实现本文所述方法只需 DTU 具备二进制数据透明传输功能即可，市面上所有 DTU 都具备此功能 [10]。

单片机端和网络服务器端分别编写程序实现固件数据交互传输功能。单片机固件程序二进制文件大小一般在几十 KB 到几百 KB 之间，考虑到 DTU 和单片机数据缓存和处理能力，适宜采用分块传输、确认应答的方式完成传输。一种类似简单文件传输协议（TFTP）[11] 的实现方式如图4所示。

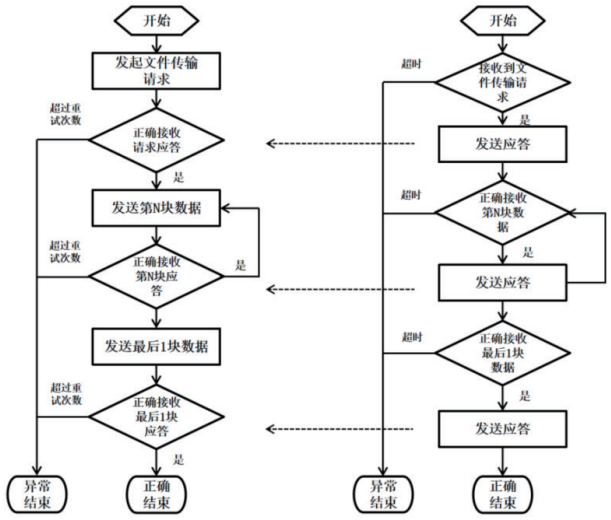

图4 网络服务器与远程单片机端传输固件程序文件流程

在需要升级固件时，网络服务器向远程单片机终端发起文件传输请求，请求帧一般携带文件名、版本号（可用于区分存放在哪个区）、文件校验值等信息。单片机终端目前如果满足升级条件，则暂存文件校验值信息用于接收完成之后的校验值比对，然后发送应答帧。服务器收到正确应答之后接下来循环发送分块数据，直到最后一帧发送文件按整块分割完后的剩余数据。在发送过程中，应设置应答超时时间，以及最大重试次数。超过超时时间未接收到正确应答则再次发送未收到正确应答的那一帧，超过最大重试次数则认为本次文件传输失败。笔者在实践过程中发现重发机制非常重要，如果没有重发机制，传输失败的概率非常高。

服务器端在接收到最后一块的正确应答之后结束传输流程，等待远程单片机终端重启后跳转到新的固件运行。远程单片机端在发送最后一块数据的应答之后，计算所有接收数据的校验值，与文件传输请求中携带的校验值比对，如果一致则将区域信息、文件长度、校验值等数据写入闪存的参数保存区然后重启运行。

上述流程是按照网络服务器发起文件传输请求，单片机终端应答请求的方式编写的。具体实现时，也可以设计为单片机终端以间歇性查询的方式发起文件传输请求，服务器应答的方式实现。两种方式除请求发起者不同外，文件分块传输的流程是一致的。

五、结语

值得注意的是本方法也有一定的局限性。本文所述保底固件和2个区的升级固件都具有完全业务功能，因此对单片机的闪存容量有要求，即完全功能的固件大小应小于单片机闪存容量的三分之一。笔者最早在小华半导体 HC32F460KE 单片机实现本方法，这块单片机内置闪存高达512KB，全功能固件程序大约120KB，因此在具体实现时将其划分为3个160KB 固件区，1个32KB 参数保存区。

本文所述方法是实际工程项目的总结，具有较强通用性，很

容易在类似架构的系统中实现。现代单片机基本都具有运行时修改自身闪存数据的功能，因此只需使用一路 UART 配合具有透明传输功能的 DTU 即可实现本文方法。文件传输协议方面，利用类似“花生壳”等内网穿透软件，网络服务器和单片机端均可自行测试文件传输功能，再进行联合调试，有助于提升开发效率。

参考文献

[1] 周远举 . 基于云平台的 STM32 固件远程升级设计 [J]. 软件 , 2024, 45 (07): 181-183.

[2] 焦金涛 , 黄灿坤 , 张佶 , 等 . 基于 STM32 微控制器的物联网设备安全更新平台设计 [J]. 电脑编程技巧与维护 , 2024, (09): 173-176. DOI:10.16184/j.cnki.comprg.2024.09.030.

[3] 刘瑞鹃 . 地铁自动售检票系统读写器控制器固件远程更新机制研究 [J]. 电脑编程技巧与维护 , 2025, (01): 108-111. DOI:10.16184/j.cnki.comprg.2025.01.015.

[4] 陈峰 , 刘鹏飞 , 徐明阳 , 等 . 基于 485 总线的 STM32 远程固件更新与实现 [J]. 计算机测量与控制 , 2022, 30 (11): 147-152. DOI:10.16526/j.cnki.11-4762/tp.2022.11.022.

[5] 唐鹏程 , 汪旭明 , 胡力 . 用 IAP 技术在线升级 STM32 单片机固件 [J]. 吉首大学学报 (自然科学版), 2019, 40 (01): 21-26. DOI:10.13438/j.cnki.jdzk.2019.01.006.

[6] 陈景郁 , 朱洪雷 . 基于 ST-Link 的 STM32 单片机多路固件烧录方法研究 [J]. 机电信息 , 2020, (09): 76-77. DOI:10.19514/j.cnki.cn32-1628/tm.2020.09.039.

[7] 金杭 , 徐京生 , 朱毓 , 等 . 单片机通过 USB 升级固件的方法 [J]. 电工技术 , 2025, (03): 157-158+163. DOI:10.19768/j.cnki.dgjs.2025.03.038.

[8] 李瑞 , 江学焕 . 基于 Ymodem 的 GD32 固件升级 Bootloader 设计 [J]. 湖北汽车工业学院学报 ,2023,37(03):53-57+62.

[9] 小华半导体有限公司 . HC32F460_F45x_A460 系列 32 位 ARM® Cortex®-M4 微控制器参考手册 [EB/OL]. https://www.xhsc.com.cn/product/1246.html

[10] 卢瑞祥 , 王文丹 , 张晓庆 . 基于远程数据透明传输的医院用呼吸机巡查监测系统 [J]. 医疗装备 , 2021, 34 (22): 4-7.

[11]RFC1350:THE TFTP PROTOCOL (REVISION 2).https://www.rfc-editor.org/rfc/rfc1350.

