基于STM32芯片的固件搬运功能研究

杜 翔[1] 罗 婷[2通讯作者]

（1. 中国电子科技集团公司第二十九研究所，四川 成都 610000；
2. 中国电子科技集团公司第二十四研究所，重庆 400030）

摘 要：随着单片机技术的应用越来越广泛，单片机的固件升级技术研究对于电子产品的功能优化具有非常大的促进意义。本为主要以STM32单片机为例，研究单片机固件搬运技术，实现芯片flash中固件搬运。根据单片机的FLASH大小将单片机分为三个区域，分别为BOOTLODER程序存储区、固件1存储区以及固件2存储区。单片机上电后BOOTLODER存储区开始运行BOOTLODER程序，没有触发固件搬运功能时BOOTLODER实现程序跳转功能，使单片机程序运行固件1存储区域的程序功能，当通过按键触发后系统初始化重新跳转至BOOTLODER区域执行BOOTLODER程序并且触发了单片机固件搬运功能，通过读取固件2存储的内容并将数据写入固件1区域，写入完成后将固件1区域数据进行标准CRC16校验与固件2区域整体数据进行CRC16校验，确保固件搬运后的数据正确。

关键词：单片机 FLASH 固件 BOOTLODER

中图分类号：G647.38 **文献标识码：**A **文章编号：**1003-9082（2023）04-0004-03

随着半导体技术的快速发展，嵌入式系统已逐渐告别了以电子通信领域应用为主的单片机时代，开始踏入电子、计算机、生物、医学、材料等多学科交叉的嵌入式时代。从人性化的穿戴设备到让人身临其境的虚拟现实产品，嵌入式设备正不断改善着人们的生活水平。作为嵌入式系统的核心，嵌入式软件又称嵌入式固件，协调管理系统内部资源的同时又负责系统内部与外部之间的信息交互。在嵌入式开发流程中，开发者需要通过JTAG等接口将编译好的软件下载到芯片内。当程序被烧写入芯片后，只需要上电复位系统就可以开始正常运转。以往，如果系统遇到软件故障，维修人员必须携带专用的烧写工具赶赴现场，然后重复上述过程，将优化后的程序装载到系统内。上述本地手动升级方式不但周期漫长，而且耗时耗力[1]。特别是一些高空作业的嵌入式设备无法通过该方式完成系统修复。嵌入式软件远程升级技术正是在这样的背景下产生的。身处异地的设备生产商利用该技术可以在很短的时间内，远程地对成千上万的设备进行静默升级[2]。它不仅可以修复系统漏洞，提升系统稳定性，还能使设备增加许多新的功能，提高设备的性能。嵌入式软件远程升级技术的这些优点使它具有广阔的应用前景和巨大的经济价值[3]。

本论文深入研究了现有的嵌入式设备中的固件升级过程的缺陷和缺点，通过研究固件搬运的方法，实现嵌入式系统产品固件的升级，提高了嵌入式产品固件更新的效率。很多消费电子产品都可以通过后台升级程序，在产品芯片内部有专用的程序升级管理，利用产品的物联网通信技术，实现产品的后台程序升级功能，这种技术不但保障了产品功能的实时更新，同时对于厂家也降低了产品维护成本，提升了产品的竞争力。

一、固件搬运方案设计

1.固件搬运总体方案

本文主要研究单片机的固件升级原理，实现单芯片内固件搬运的功能。本为主要以STM32单片机为例，研究单片机固件搬运技术，实现芯片flash中固件搬运。根据单片机的FLASH大小将单片机分为三个区域，分别为BOOTLODER程序存储区、固件1存储区以及固件2存储区[4]。单片机上电后BOOTLODER存储区开始运行BOOTLODER程序，没有触发固件搬运功能时BOOTLODER实现程序跳转功能，使单片机程序运行固件1存储区域的程序功能，当通过按键触发后系统初始化重新跳转至BOOTLODER区域执行BOOTLODER程序并且触发了单片机固件搬运功能，通过读取固件2存储的内容并将数据写入固件1区域，写入完成后将固件1区域数据进行标准CRC16校验与固件2区域整体数据进行CRC16校验，确保固件搬运后的数据正确。如表1所示。

表1 系统方案简化图

BOOTLODER	0X08000000
APP1	0X08008000
APP2	0X08010000

2. STM32芯片FLASH分区

STM32的FLASH主要由主存储器、系统存储器、OPT区域和选项字节4部分构成。主存储器，该部分用来

存放代码和数据常数。分为12个扇区，前4个为16KB大小，扇区5～11是128K大小，不同容量的STM32拥有的扇区数不一样，从上图可以看出主存储器的起始地址就是0X08000000，B0、B1都接GND（地）的时候，就是从0X08000000开始运行代码的。系统存储器：用来存放STM32的BOOTLODER代码，也就是出厂代码，专门来给主存储器下载代码的，系统B0接V3.3，B1接GND的时候，从该存储器启动，（即进入串口下载模式）。OPT区域即一次性可编程区域，共528字节，被分成两个部分，前面512字节（32字节为1块，分成16块），可以用来存储一些用户数据（一次性的，写完一次，永远不可以擦除!），后面16字节，用于锁定对应块[5]。选项字节，用于配置读保护、BOR级别、软件/硬件看门狗以及器件处于待机或停止模式下的复位。本文以STM32F103C8T6为设计对象，48引脚，FLASH大小128K。

表2　STM32F103C8T6芯片FLASH存储表

模块	名称	地址	大小（字节）
主存储块	页0	0x0800 0000 – 0x0800 03FF	1K
	页1	0x0800 0400 – 0x0800 07FF	1K
	页2	0x0800 0800 – 0x0800 0BFF	1K
	页3	0x0800 0C00 – 0x0800 0FFF	1K
	页4	0x0800 1000 – 0x0800 03FF	1K
	……	……	……
	页127	0x0801 FC00 – 0x0801 FFFF	1K
信息块	系统存储器	0x1FFF F000 – 0x1FFF F7FF	2K
	选择字节	0x1FFF F800 – 0x1FFF F80F	16
闪存存储器接口寄存器	FALSH_ACR	0x4002 2000 – 0x4002 2003	4
	FALSH_KEYR	0x4002 2004 – 0x4002 2007	4
	FALSH_OPTKEYR	0x4002 2008 – 0x4002 200B	4
	FALSH_SR	0x4002 200C – 0x4002 200F	4
	FALSH_CR	0x4002 2010 – 0x4002 2013	4
	FALSH_AR	0x4002 2014 – 0x4002 2017	4
	保留	0x4002 2018 – 0x4002 201B	4
	FALSH_OBR	0x4002 201C – 0x4002 201F	4
	FALSH_WRPR	0x4002 2020 – 0x4002 2023	4

如表2所示，STM32芯片FLASH可用带下为128K，按照系统方案将FLASH大小设为3个区域，BOOTLODER区域大小20K，地址为0X8000000至0X8007FFF，APP1区域大小20K，地址为0X8008000至0x800FFFF，APP1区域大小20K，地址为0X8010000至0x8017FFF。因此，在建立三个软件工程时设置存储的区间大小。

3.数据校验方案

单片机完成数据搬运后，为了保证数据写入正确，需要设计数据校验方案进行校验，确保写入的数据正确。单片机在搬运过程中，将数据从固件2的地址读出，每次读取固定长度，每次读取完成后，将读取的数据按照固件1的起始地址开始写入，每写入完成一次，将数据读取出，然后根据校验算法校验数据，与固件2读取出的数据校验值比较，判断二者是否一致，一致则写入正确，否则，重新写入[6]。

CRC可以高比例的纠正信息传输过程中的错误，可以在极短的时间内完成数据校验码的计算，并迅速完成纠错过程，通过数据包自动重发的方式使得计算机的通信速度大幅提高，对通信效率和安全提供了保障。由于CRC算法检验的检错能力极强，且检测成本较低，因此在对于编码器和电路的检测中使用较为广泛[7]。从检错的正确率与速度、成本等方面，都比奇偶校验等校验方式具有优势。因而，CRC成为计算机信息通信领域最为普遍的校验方式。CRC校验应用广泛，并且计算得到的数据具有唯一性，相比和校验更准确，因此本设计校验方式选用CRC校验。

二、程序方案设计

1.bootloder程序设计

主要由三个软件工程，BOOTLODER工程，APP1以及APP2工程，BOOTLODER软件工程主要完成软件的固件搬运以及跳转功能，因此设计BOOTLODER工程软件流程图。系统上电初始化，主要初始化系统时钟以及GPIO，程序循环执行是否有固件搬运标记，如果有，则执行固件搬运功能，如果没有，则执行程序跳转功能，跳至执行APP1程序。如图1所示。

2.固件搬运程序设计

固件搬运功能主要将APP2存储的FLASH数据读取后，写入APP1的FLASH区域。为了保证数据写入的完整性，每次读取写入都需要进行数据校验。

BOOTLODER程序中主要实现固件搬运功能，系统通过按键触发，开始固件搬运，程序开始从APP2的FLASH起始地址读取数据，每一次读取512个数据，并计算得到CRC校验值，将读取的数据写入APP1 FLASH地址，为了每次写入数据的正确性，写入后通过读取APP1 FLASH写入地址的数据进行CRC校验，将读取的APP2 FLASH数据的CRC与

APP1 FLASH地址读取的数据校验值比较，如果两个校验值相同则写入正确，不同则写入失败，需要重新写入数据。系统通过地址识别读取的APP2 FLASH数据是否是最后一帧数据，如果是最后一帧，则固件搬运完成，否则继续读取下一帧数据，继续进行固件搬运[8]。如图2所示。

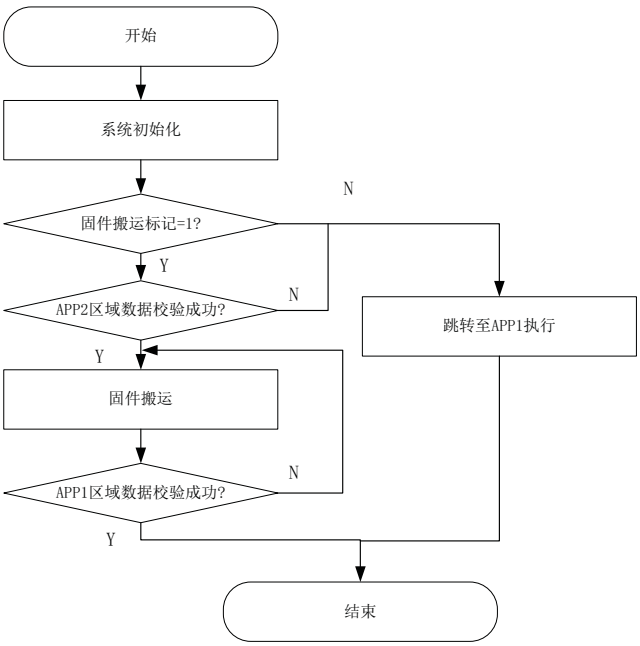

图1　BOOTLODER软件流程图

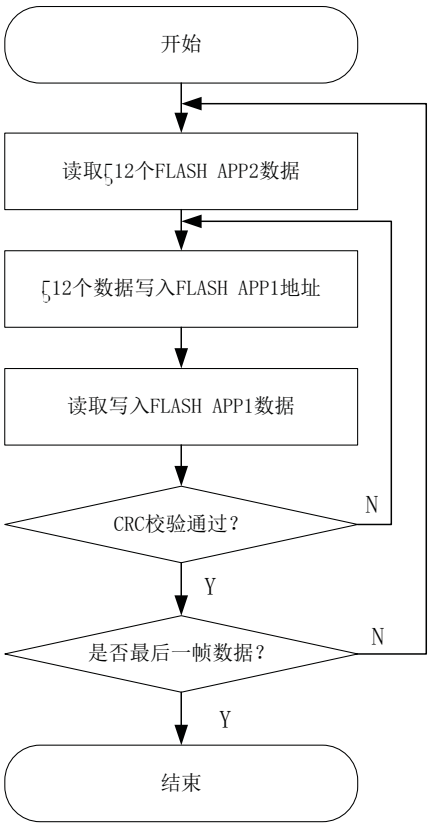

图2　固件搬运软件流程图

三、研究启示

本文主要研究单片机的固件升级原理，详细阐述单片机的FLASH存储原理，基于芯片烧录固件的原理，设计通过BOOTERLODER启动实现单片机内固件搬运的功能。完成上述设计过程中，深入分析了STM32芯片的FLASH区域划分以及芯片的FLASH操作原理，设计了固件搬运的实验，从实验现象分析设计完成了单片机固件搬运功能[9]。

尽管本文完成了固件搬运的功能，但是仍然存在不足，这种固件搬运需要提前将确定的固件烧录芯片的FLASH备用区域，但是对于FLASH大小不足的芯片，这种方案不太实用，可考虑通过其他途径实现BOOTLODER在线烧录方案。

参考文献

[1]吕春艳,靳占军,钟贻兵等.4G无线通信技术的单片机远程升级研究[J].单片机与嵌入式系统应用,2019,19(11):60-64.
[2]沈素霞,金鹰.IAP在交通信号灯运维装置中的远程升级应用[J].单片机与嵌入式系统应用,2019,19(05):51-53,57.
[3]吕春艳,靳占军,张乐君等.STM32单片机在线升级设计及实现[J].信息通信,2017(06):110-111.
[4]李永,李芙玲.基于STM32单片机的监控终端程序代码远程升级功能的实现[J].华北科技学院学报,2016,13(03):72-76.
[5]周振齐.单片机IAP在应用软件升级的方法探究[J].数码世界,2016(05):11-12.
[6]李健行,黎英,郭志伟.基于ZigBee的MSP430单片机无线升级技术[J].化工自动化及仪表,2016,43(01):76-79,92.
[7]赵小录,徐迎晖,罗欢.C8051F410单片机BootLoader的实现[J].电子设计工程,2014,22(08):175-177,181.
[8]马乾,赵俊奎,张宇.基于英飞凌XC2267单片机的远程程序更新系统设计方案[J].自动化与仪器仪表,2014(01):127-128.
[9]朱少辉,夏超英.基于CAN总线的ECU在线编程技术[J].单片机与嵌入式系统应用,2014,14(01):24-27.