一种改良安全机制的嵌入式远程升级系统的研究*

邓 力 周新志
(四川大学电子信息学院 成都 610065)

摘 要：针对嵌入式设备在远程升级过程中，存在因通信质量不佳而发生升级中断、文件传输错误等异常，导致升级失败的问题，在传统的程序备份机制的基础上改良了存储方案，设计并实现了一套嵌入式远程升级系统。该系统采用一段只具有应用程序核心功能的小型安全代码作为紧急备份，此代码可以完成应用程序的主要功能并进行硬件自检，尝试修复异常并恢复远程升级。升级中心以嵌入式设备存储器的最小擦写区块为单位生成差异文件，下放到升级设备，降低了流量的消耗并且延长了储存器的擦写使用寿命。系统采用了多重校验机制，确保升级文件在传输过程中的完整性。最后通过模拟异常情况的测试，证明了该系统在远程升级过程中的安全与稳定性。

关键词：远程升级；安全；多重校验；差异文件
中图分类号：TP311.52；TN709 **文献标识码**：A **国家标准学科分类代码**：510.8060

Research of remote update system for embedded devices with improved security mechanism

Deng Li Zhou Xinzhi
(College of Electronics and Information Engineering, SiChuan University, Chengdu 610065, China)

Abstract: To solve problems like communication failure, data transmission error during the remote updating process of embedded devices. This article design and implement a remote update system with an improved security mechanism based on traditional application backup method. With the help of a simplified safe application which contains only core code of normal application, this System can remain stable during the time when updating process is held up due to data communication failure. The system uses multi-verification mechanism and it uses differential data instead of updating the whole software, reducing information flow consumption and extend wipe/write operational life of FLASH. The test result proved the security and stability of the system during remote updating process.

Keywords: remote update; security; multi-verification; differential data

0 引 言

随着物联网技术的发展，具有数据通信功能的嵌入式设备在工业控制、智慧城市、水利自动化等领域得到广泛的应用。安装到现场的设备可以通过数据通信模块远程升级自身的软件，提高了嵌入式设备的可维护性[1]。目前针对嵌入式设备的远程升级，有许多不同的方案，这些方案的侧重点和对于升级过程中异常情况的处理各有不同。文献[2]提出了一种安全备份方案：将嵌入式存储器划分为两个部分，使用其中一部分存储旧版程序作为备份，但该方案要求设备的存储空间足够大，能存下两份应用程序，适用范围受限。文献[3]对用字节差分算法生成升级信息进行了研究。该方法能提取出新旧应用程序的不同，生成差分文件。但算法中的数据插入和删除操作不符合嵌入式设备存储器(闪存)的擦写特性，设备收到字节差分文件以后仍需进行大量复杂的操作才能完成对旧版程序的更新，而在远程升级过程中，设备需要执行的操作应尽量简单，避免发生错误。此外，目前的升级方案，对升级数据采用的校验机制都比较简单。

本文就目前的一些升级方案中存在的问题，着重研究了升级过程中的安全、数据校验机制和升级信息传输的形式。设计并实现了一整套包含召测客户端和升级中心软件的嵌入式远程升级系统。

1 远程升级方案的设计

受到 Windows 系统中的安全模式的启发，本系统在升

收稿日期：2017-01
* 基金项目：国家自然科学基金项目(611721810)资助

级中断或是应用程序错误而无法重新请求升级的时候，使用一段具备应用程序最小功能的代码来替代残缺的应用程序，这段安全代码体积较小并具备基本的硬件检测功能，能够在维持设备正常运行的同时排除设备自身的故障。

升级中心软件将升级文件按照嵌入式设备存储器的最小擦写区块生成差异文件，设备只需要更新差异部分即可，执行效率高且延长了 FLASH 擦写寿命。此外，系统使用安全性更高的 Crc 和 MD5 混合文件校验机制，能够确保数据在传输和写入过程中的完整性。本系统是以意法半导体公司生产的 STM32 F103 芯片为平台来设计的，但系统方案只需做少量修改就可以兼容其他的嵌入式芯片。系统的整体结构图如图 1 所示。

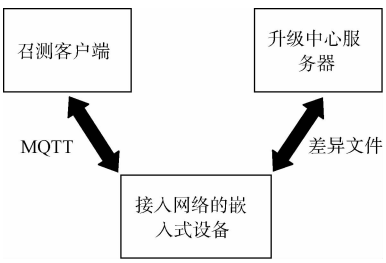

图 1　系统整体结构

系统包含升级中心服务器、召测客户端和嵌入式设备 3 部分。召测客户端通过 MQTT 标准物联网协议来完成嵌入式设备版本的召测。但由于升级文件的数据量较大，使用 MQTT 来传输效率会比较低且耗费的流量会更多。故升级中心服务器直接通过 GPRS 网络连接设备，使用一个简洁的自定义通信协议进行升级。

2　嵌入式升级软件设计

2.1　存储方案的设计

嵌入式设备的远程升级是由一段引导加载程序来更新应用程序代码。前者被存储在 FLASH 的起始位置，被称为 BootLoader，而被更新的应用程序称为 App。嵌入式设备每次复位先进入 BootLoader，它判断跳转标志，决定是进行软件升级还是直接跳转 App，这个跳转标志通常被存储在 FLASH 上一个独立区域[4]。

在嵌入式软件编程中，可以通过_attribute_(at(Some_Addr))指令来将一段数组指定到 FLASH 中的 Some_Addr 位置。故在嵌入式设备出厂的时候，可以将 App 程序精简，生成一个具有 App 核心控制功能的小型代码，称作 safeApp，它还包含了数据通信和硬件测试功能，能完成硬件自检并连接数据中心，类似于 Windows 的安全模式。safeApp 被编译成数组存放在 FLASH 最后的区段，在升级过程中一旦发生数据通信中断、物联卡流量耗尽或者是 App 文件错误，又无法重新请求升级等情况，BootLoader 会将 safeApp 读出，通过_attribute_(at())指令指定到 App 的起始位置。这样可以在紧急情形下完成现场恢复，等待网络恢复后重新连接中心完成升级。

由于升级中断的异常通常来源于通信模块的故障，所以 safeApp 中的硬件检测代码主要是检测通信模块，它会通过 AT 指令测试数据通信模块是否正常：检测 IP 状态、信号强度、DNS 配置等，修改备用的配置参数并重启模块，不断尝试重新建立 IP 链接。一旦连接上中心便上报错误参数。在实际应用中，通信中断的异常一般不会持续太久，safeApp 只需要在此期间完成设备的主要控制功能，如控制阀门或闸门等设施，就能保持现场稳定。

本系统将嵌入式设备的 FLASH 划分为 4 部分：BootLoader 段、预留参数段、App 段和 safeApp 段。本系统使用的 STM32 F103 单片机的 FLASH 存储器划分情况如表 1 所示。

表 1　FLASH 存储器划分情况

名称	大小/KB	地址区间
BootLoader	32	0x08000000～0x08007FFF
参数段	8	0x08008000～0x08009FFF
App	412	0x0800A000～0x08070FFF
safeApp	60	0x08071000～0x0807FFFF

参数段不会在升级过程中被擦除[5]，所以用于存储例如：跳转标志、App 程序的配置参数、采集的历史数据等内容，避免了远程升级过后重要数据的丢失，所以预留给参数段的空间不宜过小[6]。

预留参数段中有一个参数 appBootFailure 用于记录跳入新版程序失败的次数，每次跳转之前该值加 1，如果成功进入 App，且 App 稳定运行并成功向中心上报一次数据，则将其置零；而如果跳转失败，则由独立看门狗复位设备，再次跳转。如果 appBootFailure 达到 3，且 BootLoader 重新请求升级失败[7]，则进入 safeApp。

2.2　通讯数据校验方案设计

升级文件传输和写入 FLASH 的过程中都可能发生错误，故本系统采用了 Crc 和 MD5 混合校验的机制：每次将数据写入存储器之前，把即将写入的数据做一次 Crc16 校验，数据写入之后，再把写到 FLASH 上的数据做一次校验，确保没有发生写入错误。嵌入式设备升级的流程如图 2 所示。

本系统采用芯讯通公司的 SimCom800c 模块进行 GPRS 通信，该模块支持短信收发功能，也可以通过向物联卡发送短信来进行版本召测或发送升级指令。

此外，嵌入式设备向中心请求升级时，中心程序会将整个新版软件的 bin 文件做一次 MD5 校验，将校结果和文件总长度编入回复帧。设备收到回复帧，将这两个参数存储起来，等所有升级数据传输完毕之后，BootLoader 会按照之前的参数，将存储器上 App 区域的数据再做一次 MD5 校验。与中心的校验结果比较，如果相等，则表明升级文件

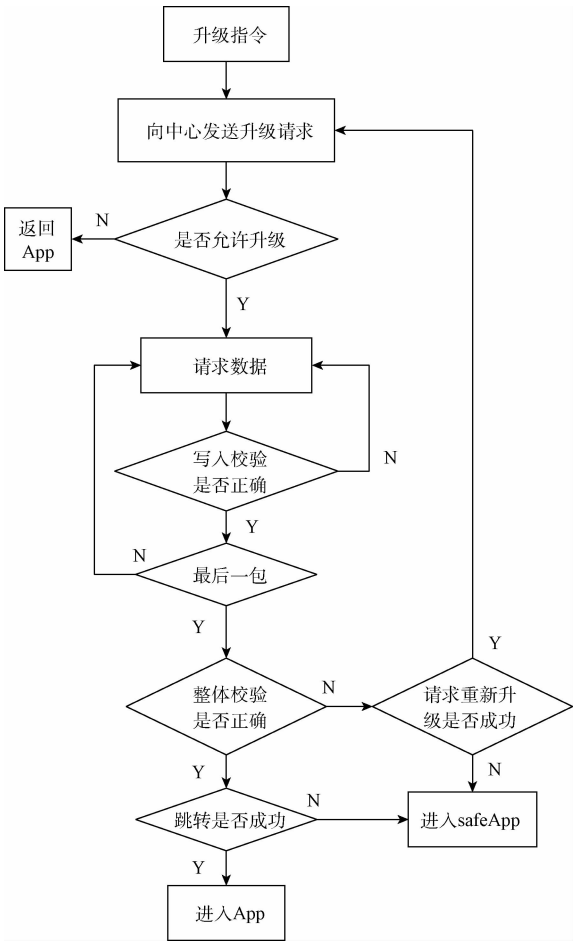

图 2　嵌入式设备远程升级流程

传输无误。

此升级过程中,MD5 校验数据量很大,将整个 App 文件读到内存再传给校验函数是不现实的。可以将 App 的起始地址强制转换成无符号 8 位整数类型的指针,作为参数传递给校验函数[8],就可以轻便地完成校验计算。

MD5 校验比 Crc 的安全性高很多,是一种流行的文件校验算法,128 位的校验值能够精确地保证升级文件数据在传输过程中的完整性。

2.3　App 程序中断向量表偏移设置

正常的嵌入式程序复位之后,PC 指针会从 FLASH 存储器上的 0x08000004[9] 位置处取出复位中断向量地址,从而进入复位中断程序,该程序执行完毕之后才会进入 main 函数。而 App 程序存放于预留参数段的后面,其复位中断向量的存储地址相对于正常程序有大小为 0xA000 的偏移,如果不预先在 IDE 中对 App 进行相应设置,PC 指针将无法指向 App 的复位中断程序,导致跳转失败[10]。下面给出修改偏移量的部分代码:

```
#define VECT_TAB_OFFSET  0xA000
SCB->VTOR = FLASH_BASE | VECT_TAB_
```

OFFSET;

如果 App 程序使用了分散加载,那么还要注意修改其分散加载文件. sct,将加载和执行区域向后分别移动 0xA000 大小的距离,通过上述设置,App 才能正常地运行于设定的区域[11]。

3　召测客户端设计

召测客户端通过 MQTT 协议对安装在野外的设备进行版本召测,以图形化界面显示,方便管理人员整体调度。MQTT 协议的全称是消息队列遥感传输协议,是一种基于“发布/订阅”模式的轻量级通讯协议,运行于 TCP 协议栈之上。它通过往应用消息上添加一个称为“Topic”的标签来完成发送和接收者的身份定位。本系统使用百度云的物联网服务器作为 MQTT 代理服务器,每个设备至少会订阅一个编有自身设备编号的 Topic。本系统中,版本信息的召测流程使用最高级别服务质量 QoS2,这是一种“仅此一次”的通信机制,可靠性很高,最大限度避免了嵌入式设备因错误的升级指令而进行误升级的情况。

召测软件可以从数据库中读出所有设备的设备编号,操作人员可以对这些设备选择性发送召测信息,嵌入式设备收到指令后立即上报本机的软件版本信息,根据这些信息可以决定是否进行升级,直观而高效。远程召测客户端的界面如图 3 所示。

图 3　召测软件界面

4　升级中心设计

本系统使用 C# 编写的升级中心软件,在实际应用中,有时升级文件相对于旧文件只有少量的修改,如果服务器每次都将升级程序不加预处理全部传输,会浪费很多流量[12],故本系统采用如下的改良方案:

BootLoader 对 FLASH 存储器上数据的修改,都是以其最小擦写区块为单位进行的,对于 STM32 F103 芯片来说,该单位是大小为 2 KByte 的“页”,哪怕新旧版文件在一页内只有一处不同,也需要整页擦除重写。如果有的页完全相同,则这部分就根本不需传输。升级中心收到携带设备当前软件版本号和升级软件版本号的请求指令后,将两

版文件从硬盘中读出并进行比对，生成一个二进制的差异文件 Rom_Differ，大小为新版文件的大小。然后以嵌入式设备的页大小为尺度对其进行划分，去掉完全相同的页，留下存在差异的页，并将处于差异页内的新版文件数据和差异页的相对位置下放给升级设备，设备只会擦写差异区块，延长了 FLASH 的擦写寿命。

新版文件长于旧版文件的部分也作为差异传输，若新版文件短于旧版文件，设备会根据新版文件长度将多余的旧版文件擦除。因为升级过程中个服务器通常处理的大多是相同类型的设备的升级服务，只有一份差异文件占用内存空间，不会对服务器程序造成太大负荷。

实验证明，如果只修改程序中一些常量值的话，差异部分通常只有一页。嵌入式设备的最小擦写区块越小，这种方案的优势就越明显，数据传输流量的消耗就越低，同时也减少了错误发生的几率。服务器工作流程如图 4 所示。

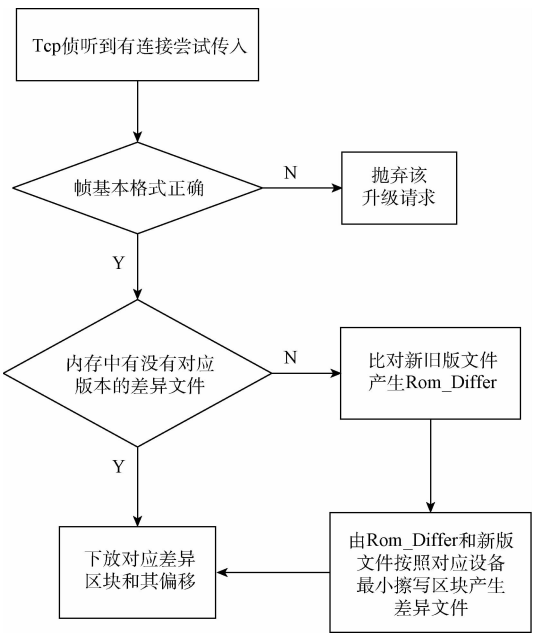

图 4　服务器工作流程

差异文件如果超过一定时间都未被调用，会被自动删除以节省资源。

5　系统测试

为了测试系统的可靠性和稳定性，使用 20 台载有 STM32 F103 芯片的嵌入式设备同时进行升级，先后升级 5 次，共计 100 次。对于这 20 台设备，分别设置 2 种不同差异程度的升级文件：一半只修改少量代码和配置参数常量，另一半做出相对较大的修改。升级文件平均大小为 130 K 左右。由于本系统采用的通讯模块 SimCom800c 单次能处理的数据量最大为 1 460 Byte，所以单包传输 1 KB 即半页的差异数据。表 2 是升级测试的结果情况统计。

表 2　设备升级测试结果

升级批次	成功台数	失败台数	平均耗时/m
1	19	1	8.5
2	20	0	5.6
3	20	0	5.3
4	20	0	5.95
5	20	0	4.25

升级失败的设备运行旧版本的 App，升级失败的原因是：设备的通信质量不佳，多次请求升级数据没有收到中心的回复而放弃升级。

然后，增加一组模拟通信中断的测试：升级过程中拔掉其中 10 台设备的无线模块，通过调试信息发现这些设备都正常运行 safeApp，每隔 40 s 尝试一次连接中心。将无线模块放回后，这些设备均在一定时间内向升级中心重新请数据，完成了自身的软件升级。整个测试过程，没有设备死机。通过中国移动的物联卡的管理平台查询到本次测试平均每台设备耗费流量 104 KB 左右。

6　结　　论

测试结果表明，本系统的安全备份和校验机制，能够保证嵌入式设备在远程升级过程中数据传输完整可靠。本系统并不需要对整个应用程序进行备份，就能从升级中断的异常中恢复正常。适用于应用程序比较大而嵌入式设备存储空间又不是那么充足的情况。而在实际使用中，使用差异文件能一定程度地减少远程升级过程中数据流量的消耗并提高远程升级的效率。

此外，要使引导升级程序 BootLoader 兼容其它型号的芯片，只需添加其他芯片的 FLASH 写入/读出部分代码、存储位置标志、看门狗等为数不多的模块并编译对应芯片的库文件即可，可移植性很强。整套系统具有较高的稳定性和可行性。

参考文献

[1]　雷卫延，敖振浪，周钦强. 基于 STM32 的在应用编程(IAP)开发[J]. 电子测量技术，2015，38(5)：62-66.

[2]　彭井花，蔡声镇，吴允平，等. 基于 GPRS 的嵌入式系统软件的远程在线升级[J]. 现代电子技术，2009(4)：47-49，52.

[3]　王广辉. 嵌入式固件远程升级技术的研究与实现[D]. 成都：电子科技大学，2011.

[4]　阙凡博. 基于 stm32 的程序远程升级设计[J]. 仪器仪表用户，2013(5)：90-92.

[5]　朱伟斌，张涛，顾海涛，等. 基于 CDMA 网络的嵌入式设备远程升级系统[J]. 电子技术应用，2014(2)：135-138.

[6] MENG H, PAN L. Realization of remote update technology for embedded equipment based on μC/OS-Ⅱ[J]. Journal of Measurement Science and Instrumentation,2014(3):69-72.
[7] 李宗卿,刘忠富,吴学富,等. 无线智能家居舒适度测控系统[J]. 国外电子测量技术,2016,35(11):103-107.
[8] 李远茂,刘桂雄,曾成刚. 基于 GPS 的室外放射源信息监控系统设计[J]. 电子测量与仪器学报,2016,30(8):1244-1254.
[9] 罗文,王莉娜,肖鲲. 基于 GPRS 的嵌入式系统远程监控和升级[J]. 电子技术应用,2010(5):159-162.
[10] 吴佳敏. 嵌入式远程工业监控系统的终端设备软件设计与实现[D]. 成都:电子科技大学,2010.
[11] 邱丽芳. 基于 ISP 技术的远程升级智能仪表的设计[J]. 电子测量技术,2007,30(2):125-128.
[12] 黄庆卿,汤宝平,邓蕾,等. 机械振动无线传感网络数据分块无损压缩方法[J]. 仪器仪表学报,2015,36(7):1605-1610.

作者简介

邓力,1993 年出生,硕士研究生,主要研究方向为智能控制系统和水利信息化。
E-mail: 1903411129@qq.com

周新志,1966 年出生,工学博士,教授,博士生导师,主要研究方向为智能控制系统、新型传感技术、智能信息处理技术以及水利信息化等。

(上接第 206 页)

4 结　论

本文汇总介绍了 Linux 最小系统的搭建和移植,并给出基本的移植流程。Linux 最小系统的移植是掌握嵌入式操作系统移植烧写的有效途径。移植了最小 Linux 系统 a9 架构可以进行多任务开发,相比五操作系统的裸机开发有着更大的优势。本文还通过系统编程对该平台进行了测试,表明 Cortex-a9 也有着更高的运算效率,这也是后续进行 Linux 系统开发的基础[12]。

参考文献

[1] R GEHR J. Reel time and embedded system-keaching reliability[J]. Ieee Computer Society,2006,7(5):1-3.
[2] NI Week 2015,以大数据开启物联网之门[J]. 电子测量与仪器学报,2015,30(8):1245-1246.
[3] 蒋涛,宫琴. 基于嵌入式脉冲控制方式的电子耳蜗调试平台的体内系统的研发[J]. 仪器仪表学报,2015,36(7):1673-1680.
[4] 肖景,杨会平,贺达江. 参数化的嵌入式乘法器测试技术研究[J]. 电子测量技术,2016,39(6):98-101.
[5] 陆兴华. 嵌入式 Linux 环境下飞机稳定性惯导控制系统设计[J]. 国外电子测量技术,2016,35(9):110-115.
[6] 夏兰,林凌云. 嵌入式 ARM Linux 血氧饱和度监测系统设计[J]. 电子测量技术,2016,39(3):74-79.
[7] 耿鹏. 嵌入式系统课程教学体系研究[J]. 江苏科技信息,2012(6):42-48.
[8] 李良,姚凯. 嵌入式 Linux 系统的开发环境搭建与移植[J]. 电脑编程技巧与维护,2014(12):16-18.
[9] 郝秉华. 基于 S3C6410 处理器的嵌入式 Linux 系统移植[J]. 电脑与信息技术,2013(6):36-38.
[10] 符意德. 嵌入式系统设计原理及应用[M]. 2 版. 北京:清华大学出版社,2010.
[11] 弓雷,等. ARM 嵌入式 Linux 系统开发详解[M]. 2 版. 北京:清华大学出版社,2014.
[12] 施威铭. Android APP 开发入门[M]. 北京:机械工业出版社,2016.

作者简介

董华,工学硕士,工程师,主要研究方向为物联网技术以及信息安全。
E-mail:13577007632@139.com

苗晟(通信作者),工学博士,讲师,主要研究方向为嵌入式系统开发及信号与信息处理。
E-mail:wumingzhicao@126.com