

基于以太网实现 F28M35H52C 在线程序更新设计

王健

杭州应用声学研究所, 浙江 杭州 310023

DOI: 10.61369/TACS.2026070001

摘 要 : 在众多水下无人项目中在线程序更新功能扮演着极为重要的角色, 主要承担功能迭代和修复漏洞的任务。在更新过程中, 相比非引导式在线程序更新方式, 依靠引导式在线更新方式, 可实现设备可靠地迭代, 同时降低设备维护成本。本文基于以太网实现 F28M35H52C 双核在线程序更新的设计展开研究, 借助 CCS 和 LabVIEW 开发环境, 实现上位机与设备进行交互式在线更新功能。

关 键 词 : 在线程序更新; 水下无人设备; F28M35H52C; 以太网通信; LabVIEW 上位机

The Design of Online Program Update for F28M35H52C is Implemented Based on Ethernet

Wang Jian

Hangzhou Applied Acoustics Research institute, Hangzhou, Zhejiang 310023

Abstract : Among numerous underwater unmanned projects, the online program update function plays an extremely important role, mainly undertaking the tasks of function iteration and bug fixes. During the update process, compared with the non-guided online program update method, relying on the guided online update method can achieve reliable equipment iteration and reduce equipment maintenance costs at the same time. This paper conducts research on the design of online program update for F28M35H52C dual-core based on Ethernet. With the help of CCS and LabVIEW development environments, the interactive online update function between the upper computer and the device is realized.

Keywords : online program update; underwater unmanned equipment; F28M35H52C; ethernet communication; LabVIEW host computer

引言

传统固件更新方式需要通过专用下载器进行硬件连接才能完成操作, 该流程耗时且繁琐。随着技术的发展和市场需求的变化, 众多设备需要不断更新功能和修复漏洞, 尤其在无人设备产品中在线更新成为不可或缺的一项功能。有关 F28M35H52C 设备已实现了设备具备非引导式在线更新功能, 弥补了设备在线程序更新的不足, 使工作人员在无需拆卸设备或使用额外的硬件工具的情况下进行在线更新, 降低现场维护的难度和成本, 提高设备的便捷性和效率。但在产品更新的过程中, 外界环境因素有时会导致设备更新失败, 比如更新过程中突然掉电的现象, 设备原有程序已擦除, 新的用户程序又没成功写入, 从而使设备进入不可逆转的死机状态, 设备更新过程的可靠性需进一步提高。目前, 针对嵌入式软件的在线升级也得到了广泛的研究^{[1][2][3][4][6]}, 故本文设计了基于以太网实现 F28M35H52C 引导式在线更新程序, 该更新方式可有效解决掉电死机的现象, 具有显著研究设计意义。

一、基本原理

根据在线程序更新方式可分为非引导式更新和引导式更新。非引导式在线更新原理框图如图 1 所示。

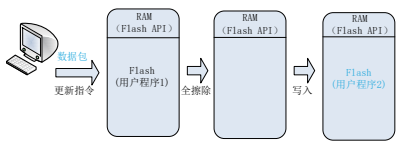


图 1 非引导式在线更新原理框图

F28M35H52C 非引导式在线程序更新流程大致如下, 通常先将 Flash API 从 Flash 区域搬运到 RAM 区域运行, 然后在 RAM 中调用相关的擦除和写入函数将新的程序代码直接覆盖旧的代码区。这种更新结构简单且易于实现, 并且不会占用大量的存储空间和内存资源, 面对资源受限的嵌入式设备比较友好。但缺点是设备受外界环境因素影响较大, 更新过程出现错误将导致系统进入不可逆转的死机状态。

引导式更新方式具有独立的引导程序来管理更新过程, 设备中设计引导程序 and 用户程序, 它们相互独立并存放在不同的存储

区域。引导式在线更新原理框图如图2所示。

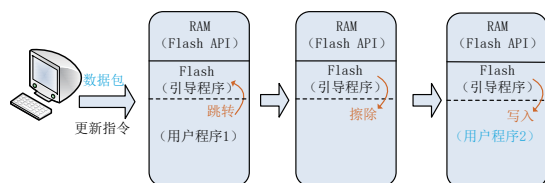


图2 引导式在线更新原理框图

F28M35H52C 引导式在线程序更新流程大致如下，当设备收到更新指令时，Flash API 从 Flash 区域搬运到 RAM 区域运行，程序 PC（用于指向当前运行指令的下一条指令）指针跳转并运行引导程序，随后 F28M35H52C 在 RAM 中调用 Flash API 的相关函数对用户程序 Flash 区域进行擦除旧程序和写入新程序。其优点是 F28M35H52C 在更新过程中稳定可靠，当在更新过程中新程序未能成功写入时，实施重新上电操作再次进行引导更新操作，不会因更新失败进入不可逆转的死机状态，实现系统的更新过程更加稳定。

综上所述，比较两种方案，非引导式在线更新方式方案已经不能满足无人设备实现更加可靠的更新操作，本设计中采用的双核 F28M35H52C 内部资源丰富，M3 子系统 Flash 为 512K 字节，RAM 为 36K 字节，C28 子系统 Flash 为 512K 字节，RAM 为 36K 字节，同时 F28M35H52C 内部的共享 RAM 高达 64K 字节，所以在芯片内部资源充足的情况下，F28M35H52C 有必要采用引导式更新方式设计相对稳定、可靠的在线更新程序功能。

F28M35H52C 双核更新设计方案总框图如图3所示。

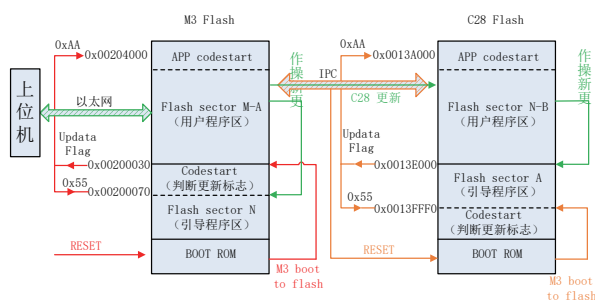


图3 F28M35H52C 双核更新设计方案框图

F28M35H52C 集成了 ARM Cortex-M3 核与 TMS320C28x-32 位 CPU，这种双核架构结合了 ARM 处理器在通信和控制方面的优势以及 DSP 在数据处理的实时控制方面的能力，可实现高效的数据处理和复杂的控制任务^[5]。如上图所示，M3 子系统的引导程序放在 Flash N 扇区，其它扇区用于存放用户程序。如图中红色线所示，系统上电复位后，M3 子系统从 Flash 引导到地址 0x00200030 处执行防升级失败程序（Codestart）进行判断更新标志位，当设备未更新或更新失败，程序 PC 指针跳转到地址 0x00200070 执行引导程序，当程序已更新成功，程序 PC 指针直接跳转到地址 0x00204000 执行用户程序。如图中绿色线所示，当 M3 子系统正在运行用户程序时接收到上位机发送的更新命令，M3 子系统停止当前运行的用户程序，执行跳转指令并运行引导程序，完成新旧程序的更改。

如上图所示，C28 子系统的引导程序放在 Flash A 扇区，其它

扇区用于存放用户程序，如图中橙色线所示，C28 子系统上电复位后，M3 子系统通过内部的 IPC (Inter Processor Communications) 控制 C28 子系统从 Flash 引导到地址 0x0013FFF0 处执行防升级失败代码进行判断更新标志位，同 M3 子系统一样，判断是否更新成功跳转到指定的更新区域或者用户区域。在 C28 子系统的更新过程中需要 M3 子系统作为媒介，C28 子系统的应用程序时刻监视 IPC 传输的命令，当 M3 子系统接收到更新 C28 子系统的命令时，并通过 IPC 将更新命令转发给 C28 子系统，如图中绿色线所示，C28 子系统跳转到地址 0x0013E000 处执行引导程序，然后 C28 子系统调用 Flash 擦除和写入函数进行用户程序的升级。

M3 子系统和 C28 子系统是相互独立的控制系统，拥有单独的程序存储区域和 RAM 区域。两个子系统都依靠 M3 子系统的以太网通信进行更新升级，M3 子系统可独立完成自己的升级，C28 子系统需借助共享 RAM 和 IPC 通道实现系统的升级。每个子系统都设计了引导功能代码和用户程序代码，在更新过程中两个子系统的引导代码不进行擦除操作，根据更新指令实现引导区和用户区之间的跳转并执行相应的功能代码。

二、软件设计

软件设计流程图如图4所示：

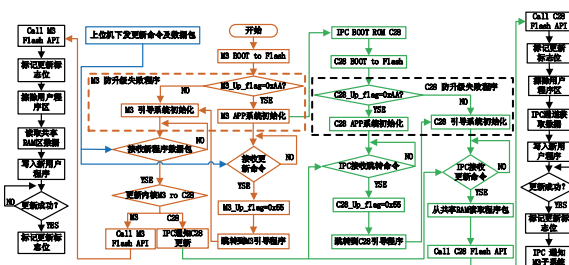


图4 软件设计流程图

本文所设计与开发的双核引导程序的设计软件在 CCS 环境中完成，采用 C 语言编写指令控制程序，上位机使用 LabVIEW 进行开发。当设备上电后，首先 M3 子系统检测更新标志位，确定跳转到引导程序还是用户程序。M3 的应用程序监视到以太网接收到更新命令时，M3 子系统停止当前应用程序的执行，跳转到 M3 引导程序的入口地址开始执行，在引导程序的主函数中，重新初始化 M3 子系统，包括系统时钟、Flash、共享 RAM、以太网等模块。等待接收上位机发送更新开始命令和数据包，上位机首先发送开始更新命令通知 M3 子系统进入准备状态，然后上位机开始发送用户程序数据包到 M3 子系统，M3 子系统接收完成并将新的用户程序数据包放置到共享 RAM 中，M3 子系统接收到上位机发送的更新结束命令，跳转到 Call_M3_Flash_API() 函数，该函数执行用户程序区 Flash 的擦除和写入操作。

C28 子系统的更新需要借助 M3 子系统的引导程序，当上位机发送更新 C28 子系统的命令，M3 子系统通过 IPC 通道将跳转命令和更新传送到 C28 子系统，随后 C28 子系统跳转到引导程序，执行与 M3 子系统类似操作，等待 C28 子系统更新成功并回送给 M3 子系统更新完成命令。

在 F28M35H52C 更新时，需要引导程序中增加更新标志位的记录，在调用 Flash API 擦除应用程序的 Flash 之前，必须在相应的更新

标志位地址调用 Flash API 写入 0x55，并在 F28M35H52C 成功升级完成后，再次调用 Flash API 在该地址增加 8 个单元写入 0xAA 表示升级成功。记录更新标志位的程序框图如图 5 所示。

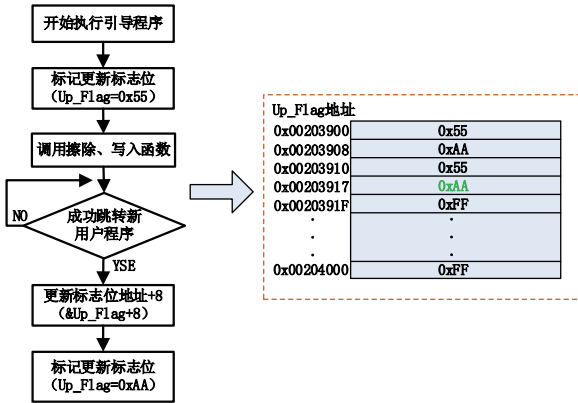


图5 记录更新标志位的程序框图

三、实试验证与测试

上位机工作流程图如图 6 所示。

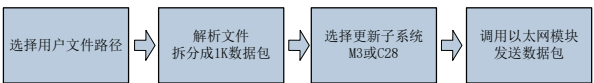


图6 上位机工作流程图

本设计中上位机软件是基于 NI 公司的 LabVIEW 软件进行开发，该上位机主要包括：以太网通信模块、文件的读取和解析模块、命令选择按钮及相关状态指示灯。其中文件读取和解析模块为上位机软件设计的关键部分，该模块读取用户升级代码并解析拆分成多个 1K 字节的数据包，通过以太网通信模块下发到 F28M35H52C。

将 F28M35H52C 通过网线连接到电脑端，选择要更新的子系统，添加用户程序的文件路径，点击发送按钮开始更新操作。上位机系统界面如图 7 所示。

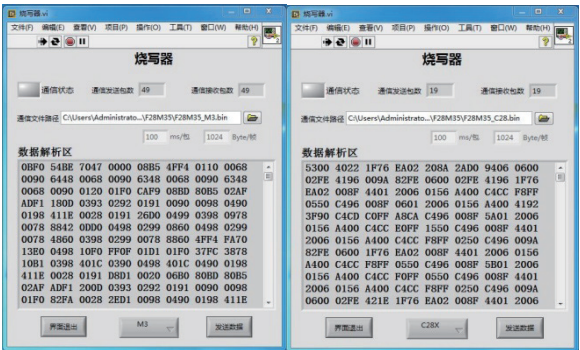


图7 上位机系统界面图

(一) F28M35H52C 防升级失败功能验证

为了方便验证，M3子系统的引导程序使用“...0b...”心跳报文，M3子系统的用户程序使用“...0c...”心跳报文。

当上位机发送一半的新用户程序数据包时，进行掉电操作，模拟 F28M35H52C 在线升级过程中突然掉电的情况，则使用抓包工具抓取心跳报文发现 F28M35H52C 停止发送心跳包，再次复位上电，F28M35H52C 正常运行并发送引导程序的心跳包。实验现象表明：M3子系统更新失败跳转到引导程序开始运行。M3子系统的引导程序心跳报文如图 8 所示。

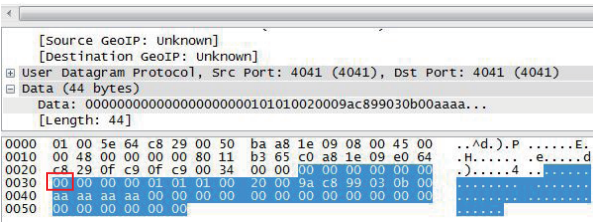


图8 M3子系统的引导程序心跳报文

当 F28M35H52C 的 M3子系统更新成功时，进行复位上电操作，使用抓包工具抓取心跳报文进行观察，抓取到用户程序的心跳报文，实验现象表明：M3子系统成功跳转到用户程序开始运行。M3子系统用户程序心跳报文如图 9 所示。

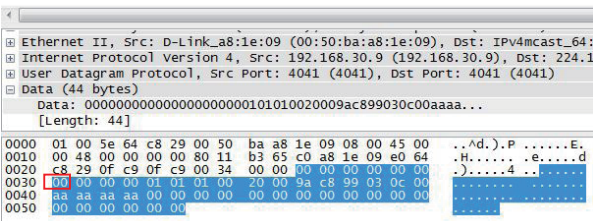


图9 M3子系统用户程序心跳报文

经上述的实验验证，防更新失败程序实现判断更新状态值进入相应的程序区，同时 F28M35H52C 在更新失败后也不会出现系统崩溃的现象。

(二) F28M35H52C 引导式更新用户程序验证

为了方便验证，M3子系统的“用户程序1”使用“...0d...”心跳报文，M3子系统的“用户程序2”使用“...0e...”心跳报文。当 M3子系统正在运行“用户程序1”时，使用上位机发送更新 M3子系统更新指令和“用户程序2”数据包。使用抓包工具抓取心跳报文进行观察，抓取到“用户程序2”的心跳报文。M3子系统“用户程序1”心跳报文如图 10 所示，M3子系统“用户程序2”心跳报文如图 11 所示。

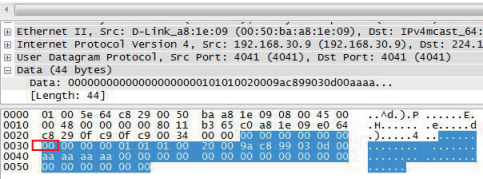


图10 M3子系统“用户程序1”心跳报文

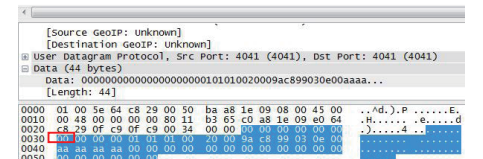


图11 M3子系统“用户程序2”心跳报文

实验现象表明：M3子系统成功完成更新操作，擦除“用户程序1”，开始运行“用户程序2”。C28子系统通过同样的实验方法进行验证，可实现正常更新用户程序。

四、结论

本文按照需求设计了基于以太网实现 F28M35H52C 引导式在

线程序更新，借助上位机的指令来控制更新对象、更新开始、更新结束操作，实现对该芯片的 M3子系统和 C28子系统引导式在线程序更新功能，同时具备防升级失败功能，当 F28M35H52C 更新过程中出现掉电现象，通过重新上电，能够再次进行程序更新，避免更新失败进入死机状态，有效提高更新过程的稳定性。

经过一系列实验可以得出，本文所设计的引导式在线程序更新方案符合设计要求，具备应有的相关功能，能够满足实际需求。

参考文献

[1] 钟洪念,丁杰,顾瀚戈,等.基于 TMS320C2000 系列 SCI 接口的实时在线程序升级方法[J].电子与封装,2023,23(5):44-48.DOI:10.16257/j.cnki.1681-1070.2023.0040.

[2] 梁学修,刘涛,姚路,等.基于 EtherCAT 的 DSP 应用软件在线更新方法研究[J].电子产品世界,2020,27(9):59-62.

[3] 周洋洋,赵昶宇.基于 CAN 总线的 DSP28335 在线烧写方法研究[J].科技与创新,2019,(6):58-59.DOI:10.15913/j.cnki.kjycx.2019.06.058.

[4] 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,(6):110-111.

[5] TEXAS INSTRUMENTS.Concerto F28M-35x Technical Reference Manual (Rev. I).

[6] 胡琪,颜科峰,蒋国军.阵列信号源程序更新模块设计[J].声学电子工程,2014,(2):36-37+40.