文章编号：2096-1472(2020)-08-05-04
DOI:10.19644/j.cnki.issn2096-1472.2020.08.002

基于统一诊断协议的控制器在线升级系统设计

马宏伟，吴长水

(上海工程技术大学机械与汽车工程学院，上海 201620)
✉1007224517@qq.com; wuchangshui@sues.edu.cn

摘　要：统一诊断协议UDS是诊断服务的标准化规范，控制器ECU是电子控制单元。针对车载ECU在使用过程中，传统的控制器ECU应用软件升级方案比较烦琐，需要将ECU拆卸下来在进行升级。本文以MPC5634M单片机作为控制器，基于UDS协议设计了一种ECU应用软件在线升级方案。Bootloader是ECU上电后首先运行的软件程序，CAN是一种串行通信。在CodeWarrior集成开发环境下，进行MPC5634M单片机Bootloader底层驱动开发，在VS平台下进行上位机设计，通过周立功USB转CAN，实现上位机和下位机CAN通信，并对ECU在线升级功能进行验证。实验结果表明：此设计通过CAN总线能稳定可靠的进行ECU在线系统升级，解决了传统控制器ECU拆卸下来在进行应用程序升级的烦琐问题，达到了应用软件升级方便的设计需求。

关键词：MPC5634M；UDS协议；在线升级
中图分类号：TP311.52　**文献标识码**：A

Design of Online Updating System for Controllers based on Unified Diagnostic Protocol

MA Hongwei, WU Changshui

(*School of Mechanical and Automotive Engineering*, *Shanghai University of Engineering Science*, *Shanghai* 201620, *China*)
✉1007224517@qq.com; wuchangshui@sues.edu.cn

Abstract: The unified diagnostic protocol (UDS) is the standard specification for diagnostic services, and the controller ECU is the electronic control unit. The current updating scheme for the traditional controller ECU (Electronic Control Unit) application is so complicated that it should be dismantled for updating. In order to solve the above problem, a new scheme for online updating the ECU application is designed based on UDS protocol, using MPC5634M single-chip microcomputer as the controller. Bootloader is the first software program to run after ECU is powered on, and CAN (Controller Area Network) is a kind of serial communication. Under the CodeWarrior integrated development environment, the MPC5634M single-chip Bootloader bottom-layer driver development is carried out, and the host computer is designed on the Visual Studio platform. Through transferring Zhou Ligong USB to CAN (i.e, a product provided by Zhou Ligong Company), the communications between the host computer and the slave computer CAN is realized, and the ECU online updating function is verified. The results of the experiment show that this newly-designed scheme enables the ECU online system to update stably through the CAN bus, which meets the design requirements of convenient application updating and solve the problem of the existing scheme.

Keywords: MPC5634M; UDS protocol; online updating

1 引言(Introduction)

在车载控制器ECU的使用过程中，通常需要进行应用层软件的更新升级。传统的ECU应用层程序升级方案比较烦琐，需要将控制器ECU拆卸下来，使用专用的程序下载仿真器进行烧写，严重影响了产品的开发周期。相比之下，控制器ECU在线升级可以很好的解决这一问题。

UDS协议[1]是在汽车电子ECU环境下的一种诊断通信协议，结合CAN通信机制可以缩短汽车电子开发周期和降低生

基金项目：上海市科委“创新行动计划”项目(17030501300).

www.rjgcz.com

产成本。目前，进行控制器ECU在线升级的研究有很多，例如：文献基于CAN总线对CCP协议进行自定义，实现了控制器的在线升级[2]。文献采用S19格式文件进行程序烧写，实现了ECU在线升级[3]。文献在控制器的Bootloader设计中，对CAN驱动和Flash驱动设计进行了详细介绍[4]。文献基于CAN FD软件升级设计中，采用外部触发方式进行ECU软件升级[5]，如果升级过程中出现掉电等现象，无法保证ECU再次上电后正确运行程序。本文在详细分析UDS协议下载流程的基础下，通过对MPC5634M底层相关驱动的配置开发，以及基于VS平台进行上位机的设计，最终实现基于UDS协议的ECU在线升级。

2 UDS协议及系统组成(UDS protocol and system composition)

如表1所示，UDS协议定义了一系列服务命令，不同的服务命令都有自己独立标识符。在CAN通信机制下，由上位机发起请求，下位机根据不同的请求命令进行响应。

表1 UDS协议服务命令

Tab.1 UDS protocol service command

Command	ID
ECU Reset	0x11
Diagnostic Session Control	0x10
Security Access	0x27
Tester Present	0x3E
Routine Control	0x31
Request Download	0x34
Transfer Data	0x36
Request Transfer Exit	0x37

2.1 UDS协议帧类型

根据传输的数据字节数是否大于8字节，可将UDS帧类型分为单帧和多帧，其中多帧包含首帧、连续帧和流控制帧，单帧的传输比较简单，通常的一问一答数据传输即为单帧传输；多帧的传输相对复杂，在传输的过程中，首先需要对多包数据进行拆分，然后再按照一定序列进行重组。

多帧的传输过程，如图1所示，发送端首先发送首帧，告知接收端将要接收的数据字节数；接收端接收到首帧后返回流控制帧，流控制帧主要有三个参数，流状态(FS)；块的大小(BS)主要包含了接收端可以接收数据大小等信息；STmin表示两个连续帧的时间间隔，超过规定的时间间隔会认为发送错误；接着发送端发送连续帧，参数SN表示帧的连续号，在发送连续帧的过程中，需要对连续帧进行编号，每发送一个连续帧，帧的连续号就会增加1，当增加到15时自动清零并重新计数，当数据发送完成后，发送端停止发送连续帧。

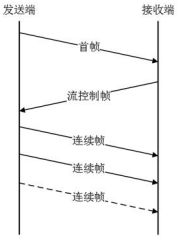

图1 多帧传输流程

Fig.1 Multi-frame transmission flowchart

2.2 UDS编程步骤

基于UDS协议的ECU在线升级系统，是按照UDS规定的相关服务命令进行应用程序更新[6]。UDS编程步骤在遵循标准化步骤的基础下，可根据需要进行自定义。本文设计的编程步骤，如图2所示。

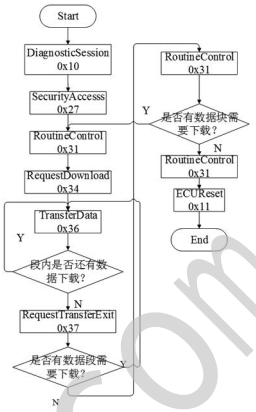

图2 UDS编程步骤

Fig.2 UDS programming steps

ECU收到上位机诊断会话服务请求后，ECU由默认的诊断会话子模式切换到编程会话子模式，为编程做准备。

在进行ECU软件升级前，需要对ECU进行解锁，即上位机向ECU发送Seed请求，ECU返回一个随机种子，上位机通过加密算法计算出密钥发送给ECU，ECU根据该密钥检测是否和自身密钥匹配。如果两者密钥相等则安全访问成功，否则返回错误消息。

通过例程控制服务调用底层Flash擦除函数，擦除原有的应用程序内存区，为下载新应用程序做准备。

ECU收到请求下载服务请求后，根据当前所处的编程会话模式，对下载前地址进行校验，以确保用户对该地址具有访问权限。获得访问权限后，进入下一步的数据传输服务。

ECU收到数据传输服务请求后，如果具备访问应用程序内存地址权限，则开始进入数据传输。在数据传输期间，数据传输服务会多次执行直到数据传输结束。

ECU收到请求退出服务后，终止上位机与ECU之间的数据传输。

通过调用例程控制服务采用CRC校验算法检验下载到ECU中数据块完整性，即ECU计算下载数据CRC值并与上位机发送的CRC值进行比较，若两者相等，则程序数据块下载成功。再次调用该服务主要是用来检查数据块的逻辑性和兼容性。

在校验完下载程序的正确性后，ECU会收到复位服务请求。该服务主要用来进行ECU复位，并将程序启动地址跳转到应用程序首地址。

2.3 系统组成

本文设计的ECU在线升级方案主要由下位机控制器ECU、上位机在线升级软件和USB转CAN三部分组成，如图3所示。

图3 系统组成

Fig.3 System composition

下位机采用具有CAN接口的MPC5634M单片机作为控制器，在CodeWarrior集成开发环境下进行开发。上位机基于Visual Studio平台进行开发，主要采用周立功公司提供的一套完整附带函数声明的Virtual CAN Interface(VCI)函数库进行设计，根据需要调用相关库函数即可完成上位机软件的开发。通信模块采用周立功USB转CAN接口卡，主要实现下位机和上位机的数据通信。

3 Bootloader设计(Bootloader design)

Bootloader程序是ECU上电后首先运行软件程序，主要完成各个模块的初始化，使系统处于恰当的状态，为应用程序的运行提供正确的环境[7−10]。

3.1 硬件选型

NXP公司针对车用而设计的MPC5634M微控制器，提供了两路FlexCAN通信和Flash编程等模块。FlexCAN通信模块符合CAN2.0B标准协议，支持扩展帧和标准帧，每次传输数据长度最大为8字节，数据Buffer为64字节，通信速率高达1Mbit。Flash编程模块提供了高达1.5MB的可编程非易失性闪存，支持64位数据总线，时钟频率高达80MHz,可配置软硬件读写访问保护，可设置擦除块地址空间，每次可写入64位地址空间。

3.2 Flash内存分配

Bootloader程序通常固化在Flash内存中[11]，一般通过专用仿真器下载完成后，不能进行修改，而Application程序需要不断地进行升级完善。为保证Application程序更新时不发生程序混乱，必须进行合理的内存空间分配，以达到Bootloader程序和Application程序无交叉覆盖[12−14]。

本文设计的Flash内存分布，如图4所示。将0x00000000—0x0000FFFF低地址64kB空间划分为Bootloader程序区，其中0x00000000—0x00000008地址空间用于存放Bootloader程序的启动半字和复位向量，0x00000009—0x0000FFFF地址空间用于存放底层相关驱动程序。0x00010000—0x0017FFFF高地址空间划分为Application程序区，其中0x00010000—0x00010008地址空间为Application程序地启动半字和复位向量，0x00010009—0x0017FFF7地址空间存放Application程序，0x0017FFF8—0x0017FFFF地址空间作为Application程序软件升级标志位。本文分别设置Bootloader程序和Application程序的启动半字和复位向量，这样可以同时独立开发、调试Bootloader程序和Application程序，降低产品的开发周期。

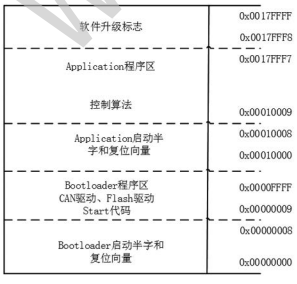

图4 Flash内存空间划分

Fig.4 Flash memory space partition

3.3 CAN和Flash驱动配置

控制器MPC5634M的Bootloader CAN和Flash配置主要是在CodeWarrior集成开发环境下进行编程设置，具体的配置如下：

CAN通信的驱动配置主要涉及CAN节点初始化、波特率、接收中断函数等。CAN初始化函数void CANA_Init(unsigned char speed)主要完成CAN节点使能，定义最大缓冲Buffer为64字节，提供125KB、250KB、500KB三种模式的波特率，可由用户根据需要进行软件选择，SIU.PCR[83].R定义为CAN Tx引脚，为开漏输出，SIU.PCR[84].R定义为CAN Rx引脚。CAN接收中断函数void CANA_RX_ISR(void)主要处理来自上位机请求会话，其中UDS编程函数uds_handler(RxDATA，RxLENGTH)就在CAN中断函数中调用。

Flash驱动配置主要是Flash的初始化、Flash扇区擦除和Flash扇区编程。Flash初始化函数unsigned char FlashInitAndUnlock(void)主要完成Flash各个扇区的初始化，提供访问Flash内存的权限，初始化成功将返回0，否则返回1。Flash擦除函数unsigned char EraseFlash(void)主要完成擦除Flash内存区域的Application应用程序，主要配置是将要擦除内存块的首地址，该函数在UDS编程过程中的例程控制服务中被调用。Flash编程函数unsigned char ProgramFlash(UINT32 dest，UINT32 source，UINT32 size)主要完成对特定内存块的编程，需要配置编程的首地址，并添加校验算法，即下载文件格式是否正确，下载程序是否完整，该函数也是在UDS编程过程中的例程控制服务中被调用。

3.4 Bootloader工作流程

本文设计的控制器ECU复位或重启后，由地址0x00000008跳出进入启动程序地址0x00000009，在启动程序内首先判断软件升级标志是否置位，若软件升级标志置位，则进入Bootloader下载模式，进行Application程序升级，否则，跳转到Application程序首地址0x00010000，执行Application程序。软件升级标志位处于Flash内存的最后一个双字节地址0x0017FFF8—0x0017FFFF，这样可保证Application程序在更新过程中如果出现掉电等现象，Flash最后一个双字节地址软件升级标志位值不会发生变化，再次重新启动控制器可以继续执行Application程序更新，避免了控制器重新启动后执行有错误Application程序的可能。

本文设计的Bootloader具体工作流程如下：在Application程序升级模式下运行Bootloader程序，首先判断下载文件格式是否是S19文件，若是S19文件，则调用Flash擦除函数擦除Application程序所在内存空间为Flash编程做准备，接着调用Flash编程函数进行S19文件下载，S19文件下载完成后校验数据完整性，若S19文件下载完成，则清除软件升级标志位，否则将S19文件未下载完成记录写入Flash，并清除软件升级标志位进行ECU复位。在Application程序运行模式下，如果接收到Application程序升级请求，则软件升级标志置位，ECU进入复位模式，随后进行Application程序在线更新。

4 上位机设计(Upper computer design)

上位机软件采用C++语言编程，在VS2015平台上进行GUI编程和通信驱动设计。由于本文上位机和下位机通信主要是进行Application程序升级，因此，用户需要进行自定义编程。

周立功USB转CAN通信设备提供了一套完整Virtual CAN Interface(VCI)函数库，包括ControlCAN.lib，

www.jgzz.com

ControlCAN.DLL库文件，以及VC版本函数ControlCAN.h声明文件[15]。主要函数库调用设计如下：首先自定义源文件usb_can.cpp包含ControlCAN.h头文件，以方便调用相关函数，在Visual Studio2015编辑环境下设置连接到ControlCAN.lib文件。接着进行打开CAN设备函数char Open_USBCAN()编程，该函数的主要作用是进行CAN通信打开，初始化CAN通信以及启动CAN通信，中间任何一个环节出现错误，都会进行错误提示。其次进行CAN发送函数MY_VCI_Transmit()和CAN接收函数MY_VCI_Receive()编程，函数内部添加CAN帧检测机制，主要用来检测CAN收发过程中出现的单帧、多帧、首帧、流控制帧、连续帧格式是否错误。

本文在MFC框架下设计的上位机GUI界面，如图5所示。该界面设计简单，方便操作，具有识别S19文件功能，同时还具有CAN通信握手检测功能、UDS报文显示功能和程序下载进度显示功能，以及程序完整性检测功能。

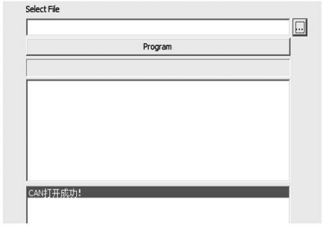

图5 上位机GUI界面

Fig.5 Upper computer GUI interface

5 软件测试(Software testing)

本次测试中，将CAN通信波特率设置为500KB，选用的下载S19文件大小为17kB，在自主设计的上位机和下位机软件上进行测试，如图6所示。下面仅对最后一列报文进行分析：上位机发送021101CCCCCCCCCC报文，其中02中的0表示单帧，2表示有效数据字节数为2字节，11表示ECU复位，01表示ECU复位服务子功能的硬件复位，后四字节数据无效用CC进行填充，上位机接收025101AAAAAAAAAA报文，其中02中的0表示单帧，2表示有效数据字节数为2字节，UDS报文规定肯定响应报文格式第二字节为请求报文第二字节加40，否则为7F，51表示ECU复位服务请求响应为积极响应，01表示ECU复位服务子功能的硬件复位，后四字节数据无效用AA进行填充。

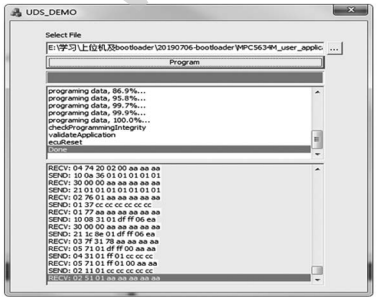

图6 Application程序升级界面

Fig.6 Updating interface of application program

实验结果表明：本文设计的ECU在线升级系统，能准确按照UDS协议帧格式进行单帧、首帧、流控制帧、连续帧的传输，达到了设计需求，满足控制器Application程序在线升级功能。

6 结论(Conclusion)

通过该研究对UDS协议编程流程、Bootloader功能设计以及上位机开发有了更深的理解，设计的ECU在线升级系统满足Application程序在线升级功能，降低了ECU开发周期，为ECU在线升级提供了一种设计思路。同时，本文设计的ECU在线升级系统程序采用两个独立的启动半字和复位向量，可同时开发Bootloader程序和Application程序，降低程序开发、调试难度，缩短开发周期。

参考文献(References)

[1] ISO14229:2013,Road vehicles-Unified diagnostic services(UDS)-Specificationand requirements[S].

[2] 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(1):24-27.

[3] 李娇娇,张宏伟,陈金干.基于LabVIEW新能源汽车控制器刷新软件设计[J].软件工程,2020,2(23):16-18.

[4] 张成雨,杨朝阳.基于CAN总线的车载应用Bootloader设计[J].湖北汽车工业学院学报,2017,4(31):68-69.

[5] 杨朝阳,阮海庭,罗永革,等.基于CAN FD的在线编程系统设计[J].计算机与嵌入式系统应用,2019(5):1-5.

[6] 汪春华,白稳峰,刘胤博,等.基于CAN总线UDS服务Bootloader应用开发[J].汽车电子与测量技术,2017,40(2):166-170.

[7] 罗峰,孙泽昌.汽车CAN总线系统原理设计与应用[M].北京:电子工业出版社,2010.

[8] 曾其林,肖大伟,王志民,等.基于CAN Bootloader的整车控制器程序更新系统设计[J].东方电气评论,2016,30(120):20-23.

[9] 王琦.基于CAN总线的Bootloader研究与实现[D].南京:南京邮电大学,2016.

[10] 王涛.基于CAN诊断汽车控制器刷新软件的设计与实现[D].成都:电子科技大学,2015.

[11] 温暖,杨维明,彭菊红,等.基于MCU的嵌入式系统的Bootloader设计[J].微电子学与计算机,2018,35(3):80-82.

[12] 顾东袁,傅晓婕,陈爱军,等.基于STM32的Bootloader实验系统设计[J].实验技术与管理,2019,36(11):89-93.

[13] 詹克旭.基于UDS协议的汽车ECU升级方案[J].计算机应用与软件,2019,36:191-196.

[14] 刘乐乐.基于CAN FD的车载Bootloader研究与实现[D].北京:中国地质大学,2017.

[15] 广州周立功单片机发展有限公司.CAN-bus通用测试软件及接口函数使用手册[Z].广州周立功单片机有限公司,2004.

作者简介：

马宏伟(1994-)，男，硕士生.研究领域：汽车电子控制技术.

吴长水(1978-)，男，博士，副教授.研究领域：汽车电子控制技术.

www.cnpcz.com