

# 基于LabVIEW的新能源汽车控制器刷写软件设计

李娇娇<sup>1</sup>, 张宏伟<sup>1</sup>, 陈金干<sup>2</sup>

(1.河南理工大学电气工程与自动化学院, 河南 焦作 454000;

2.上海同湛新能源科技有限公司, 上海 201800)

**摘要:** ECU(Electronic Control Unit)汽车电子控制单元的集成功能日益复杂, 为了应对软件更新及对Bootloader的需求问题, 本文研究利用LabVIEW编程语言, 采用CAN总线通信技术, 遵循UDS协议规范, 根据Bootloader刷写流程, 设计了基于LabVIEW的汽车ECU刷写上位机软件。该上位机集成了数据刷写过程中的所有诊断服务, 提供了简单明了、易操作的人机交互界面, 实现了程序在线更新功能, 能够很好地完成汽车ECU程序刷写任务。试验测试结果表明使用CAN网络测试工具和该上位机界面能够成功地实现频繁地刷写更新程序, 已经用在了现实程序刷写操作中, 具有较高的可靠性和有效性。

**关键词:** CAN总线; UDS协议; Bootloader刷写流程; LabVIEW

**中图分类号:** TP26 **文献标识码:** A

## Design of Brushing Software for New Energy Vehicle Controller Based on LabVIEW

LI Jiaojiao<sup>1</sup>, ZHANG Hongwei<sup>1</sup>, CHEN Jinan<sup>2</sup>

(1.School of Electrical Engineering and Automation Henan Polytechnic University, Jiaozuo 454000, China;

2.Shanghai Tongzhan New Energy Technology Co.Ltd., Shanghai 201800, China)

**Abstract:** The integration function of ECU (Electronic Control Unit) automotive electronic control unit is increasingly complex. In order to cope with software update and the demand for bootloader, this paper uses LabVIEW programming language, adopts CAN bus communication technology, follows UDS protocol specification, and writes according to Bootloader data brushing to designed a LabVIEW-based automotive ECU to refresh the host computer software. The upper computer integrates all the diagnostic services in the process of data writing, provides a simple and easy-to-operate human-computer interaction interface, achieves the online program update function, and can complete the task of car ECU program refreshing well. The test results show that the CAN network test tool and the host computer interface can successfully implement the frequent flashing and writing update program. It has been used in the real program flash writing operation, and has high reliability and effectiveness.

**Keywords:** CAN bus; UDS protocol; bootloader data brushing; LabVIEW

### 1 引言(Introduction)

近年来, 新能源电动汽车、车联网技术和无人驾驶技术的浪潮风起云涌, 在一定程度上促进了汽车电子技术的发展<sup>[1]</sup>。同时, 人们对于汽车的各种高端功能、驾驶安全性以及舒适度的要求也越来越高了<sup>[2]</sup>, 为了应对这些需求, 实现这些功能, ECU的数量就需要增多, ECU的控制策略和功能的复杂程度也需要提高, 致使ECU软件在线更新也越来越频繁。在汽车的开发、生产和售后等阶段中, 每个阶段都有软件更新的需求<sup>[3]</sup>。传统的软件的更新方法需要BDM调试器通过BDM调试接口进行程序的下载, 该方法数据下载速度慢, 插拔很不方便, 并且在ECU开发后期一般不再有调试端口,

这样对控制器程序的下载更新调试就非常的麻烦<sup>[4]</sup>。为了在开发过程中更快捷方便地完成程序地烧录更新以及在不用拆下汽车控制器的情况下, 通过CAN总线作为通信介质完成程序地升级更新及调试, 设计完成了该上位机软件<sup>[5]</sup>。

### 2 系统设计方案(The scheme design of system)

作为在线刷写升级系统的上位机软件, 主要是通过CAN测试工具和下位机建立对话服务<sup>[6]</sup>。首先需要对需要更新升级的S19文件进行解析, 转化为CAN报文格式, 依据Bootloader数据下载流程, 然后根据UDS协议中的各项服务规则向下位机发送指令, 完成程序的刷写更新。系统总体架构如图1所示。

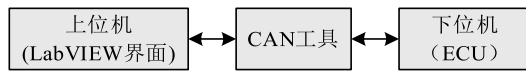


图1 系统总体架构

Fig.1 The overall structure of system

本设计是基于USB CAN测试工具作为硬件接口进行开发,其体积小,成本低廉,使用非常方便,并且支持LabVIEW软件进行二次开发,能够满足各种应用开发需求<sup>[7]</sup>。UDS(Unified Diagnostics Services)协议,是一种面向汽车电子控制单元(ECU)的统一诊断服务,不仅用于汽车故障诊断,还应用于软件刷写等应用场景,可以说是目前汽车电子领域应用最广泛的技术协议之一<sup>[8]</sup>。

### 3 Bootloader数据下载流程(The data download process of Bootloader)

Bootloader数据下载流程主要是依据UDS协议标准。UDS提供了一套标准的诊断服务,给出了车辆诊断通讯服务规范和一些参数说明,以及一些功能单元服务的基本要求<sup>[9]</sup>。UDS是一种客户端与服务端的通信服务,即客户端(Tester)向ECU发送诊断服务请求(Request),ECU则向客户端发送对应服务的请求响应(Response)<sup>[10]</sup>。通过上位机软件向下位机ECU刷写程序的流程主要是依据Bootloader数据下载流程。Bootloader的数据下载流程总体分为三个步骤,分别为:预编程阶段、主编程进行阶段、后编程阶段<sup>[11]</sup>。

#### 3.1 预编程阶段

预编程阶段是用来为要下载更新程序前的CAN网络做准备,流程如图2所示。

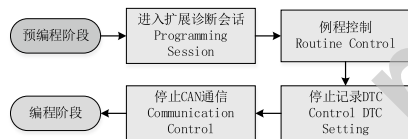


图2 预编程阶段

Fig.2 Preprogramming stage

(1)在设置刷新网络前,ECU需要进入扩展会话模式。一般来说,程序初始化后直接进入默认模式运行,当需要更高级的访问权限时,则使用该模式。

(2)例程控制(0×31):通过例程控制服务来检查编程预条件,为接下来的更新提供一个安全的更新环境。如果条件不满足则退出刷写,从而确保系统的安全。

(3)刷新期间,要求各个ECU停止除诊断以外信息的发送和接收,禁止重新设置故障码<sup>[12]</sup>。

(4)刷写工具通过通信控制服务请求禁止不属于诊断服务的报文数据的发送和接收,尽可能地降低总线的负载率,为接下来的数据传输提供足够的宽带。

#### 3.2 主编程进行阶段

在预编程阶段之后,是主编程进行阶段,是数据刷写的具体阶段。主编程进行阶段时序是单个ECU编程事件的应用,该阶段主要是下载应用软件或应用数据,因此所有的服务请求都使用物理寻址,即一对一的通信<sup>[13]</sup>。如图3所示,描述了在主编程进行阶段中执行的各项具体服务。

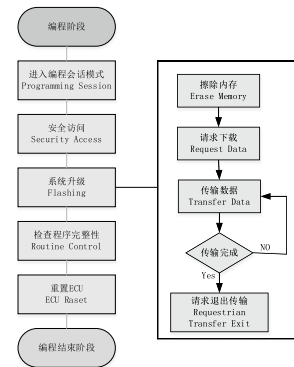


图3 主编程进行阶段

Fig.3 Main programming stage

(1)进入诊断会话控制,ECU启动Bootloader,并分配编程所需的所有资源。

(2)安全访问:主编程阶段必须通过安全访问。进行安全访问模式通过发送0×27服务来获取刷写数据的权限。上位机获取该请求种子,并发送密钥给下位机,下位机将接收到的密钥值与自身通过运算的值进行比较,当上位机发送的密码与ECU相匹配时,则获得进入所要求的诊断模式的权限,然后运行相应的诊断命令,进入相应的模式。

(3)程序升级:当ECU的非易失性存储单元中没有存储内存驱动时,将执行内存驱动的下。下载应该按照如图3所示顺序进行。

(4)例程控制服务(0×31)——“擦除ECU内存”:接下来是需要将应用软件和数据下载至ECU中,在此之前,ECU的内存需要被擦除掉,该步骤通过例程控制服务来执行擦除ECU内存操作。

(5)例程控制——“检查编程完整性”:此服务用来检验逻辑块的完整性,即验证之前传输数据的完整性<sup>[14]</sup>。

(6)重置ECU:所有数据下载完成后,测试工具将应用一个例程来触发ECU从而检查编程的依赖性。这些检查内容一般是由ECU供应商定义的,但是其中的内容必须确保所有逻辑块的兼容性、一致性和完整性<sup>[15]</sup>。

#### 3.3 后编程阶段

后编程阶段按照如图4所示,该阶段用来确认更新已经完成,同时该阶段是在复位后在应用程序中执行一些恢复整车网络通信,以及配置信息的相关操作。

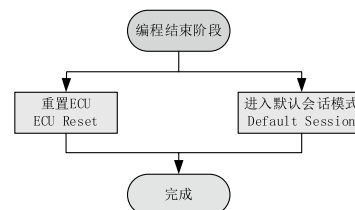


图4 后编程阶段

Fig.4 Post programming stage

(1)重置ECU:测试工具软件使用物理寻址,发送一个复位类型为硬件复位(子功能01)的ECU复位请求报文到CAN网络上,ECU收到请求后,回复一个肯定响应,结束后编程过程,返回到正常的操作模式。

(2)进入默认会话模式。

(3)完成刷写功能。

#### 4 S19文件解析(S19 file parsing)

上位机软件刷写的程序主要用于ECU读取并下载数据更新程序,该程序就是S19文件<sup>[16]</sup>。因此,需要对S19文件进行解析并提取出有效数据,转化为CAN报文格式,对ECU进行程序更新。

S19文件是Freescale CodeWarrior编译器生成的后缀名为.s19的程序文件,使用文本打开是一行行的十六进制数据,文件中的每一行称为一条记录,每行的数据长度不等,其中最长78个字节,156个字符,包括记录类型、记录长度、记录地址、记录数据和校验和<sup>[17]</sup>。

S19数据格式转化为CAN报文格式时,需要遵循UDS诊断协议规范对S19文件进行解析,提取地址信息、数据长度信息和有效数据信息<sup>[18]</sup>。为了方便解析处理S19文件,通过HexView软件观察数据内容,该S19文件共分为13个Block,还显示每个Block的首地址,以及每个Block的字节长度,根据这些信息解析提取S19文件的有效数据。

#### 5 上位机软件系统设计(The design of host software system)

为了实现上述程序刷写功能,本上位机软件设计主要分为系统初始化设置模块、S19文件解析模块、主程序设计模块、实时进度的推送模块和刷写结果的反馈模块及安全登录模块。

##### 5.1 系统初始化设置

初始化阶段包括启动接口卡、配置CAN总线和启动CAN总线<sup>[19]</sup>。通过调用LabVIEW库函数,把各个CAN工具的动态函数库DLL按照顺序创建子VI封装起来,完成初始化配置,保证LabVIEW软件中底层驱动的可靠性、稳定性<sup>[20]</sup>。

##### 5.2 数据发送接收设置

数据发送子程序是用来向下位机发送CAN报文数据,通过调用库函数的VCL\_Transmit函数来实现该功能。

##### 5.3 S19文件解析

S19文件是一种程序文件,可以使用LabVIEW中读取文本文件函数来读取该文件并转化为字符串数组。通过S19文件中的记录类型(前两个字节)来判断S19文件地址长度。通过截取字符串函数和匹配字符串函数提取出S19文件中的首地址,再通过上述程序中获得的地址长度判断提取S19文件中的有效数据,这些有效数据就是我们需要通过上位机刷写到下位机ECU的程序数据。将上述程序获得的有效数据转化为二维数组,并对获得的有效数据的正确性进行判断。

要写入的S19文件的有效数据是按照Block划分的,每个Block的划分是依据首地址和数据长度,划分之后并需要获得该Block字节数的大小,与通过HexView软件观察数据内容而得到的每个Block的首地址和字节数大小来核对程序是否正确,至此,S19文件解析完毕。

##### 5.4 主程序设计

在LabVIEW中,数据的传输是以数据流的方式传输的,数据流是一连串CAN报文数据的发送。根据Bootloader数据

刷写流程规范设计该LabVIEW程序的数据流。主程序设计流程图如图5所示。

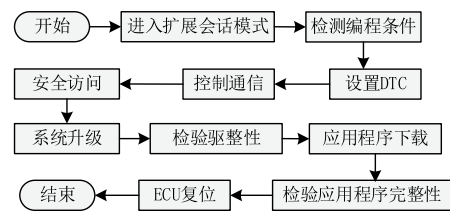


图5 主程序流程图

Fig.5 The flow chart of main program

#### 5.5 实时进度的推送和刷写结果的反馈

刷写功能的执行时间主要取决于用于刷写的S19文件的大小,通常情况下1M大小的S19文件包含超过13000行记录,接近40万字节有效数据,刷写的时间将超过两分钟,对于用户而言,在两分钟内没有得到任何刷写相关的指示将会使用户感到焦虑,所以实时进度的推送是十分有必要的。本设计通过刷写进度条来直观地显示刷写工程的速度和进度,直到百分之百,表明刷写完成。同时,刷写进度还通过字符串文字显示当前刷写状态,可以清楚地知道具体刷写到哪一步骤。

#### 5.6 安全验证

首先,对操作者的身份进行验证。刷写操作人员必须是相关的工程师或者经授权的测试人员,因此需要密码登录来确保非相关人员误操作<sup>[21]</sup>。

#### 6 结论(Conclusion)

首先给ECU上电,使其进入正常的运行模式,然后对LabVIEW上位机的前面板进行设置,配置好参数后,运行上位机,点击CAN通讯设备状态按钮,打开CAN通讯;点击添加,添加S19文件;由客户端切换到服务端,输入登录密码;点击stay in boot,完成stay in boot配置(在启动流程中加入stay in boot这一阶段,是为了增加时间冗余来保证软件的可靠性);点击开始,开始刷写,同时通过上位机界面进度条和当前状态文字信息提示,观察刷写进程。上位机刷写界面如图6所示。



图6 上位机刷写界面

Fig.6 The brushing interface of host computer

#### 参考文献(References)

- [1] 汪春华,白稳峰,刘胤博,等.基于CAN总线UDS服务BootLoader应用开发[J].电子测量技术,2017,40(02):166-170.
- [2] 吴进军,方继根,王西峰,等.基于CAN总线的新能源汽车ECU控制器程序刷写系统设计[J].机电产品开发与创新,2018,31(02):1-3;7.
- [3] 申欣欣.电动汽车控制器上位机监控软件设计[D].吉林大

(下转第8页)