

基于 UDS 协议的汽车 ECU 升级方案

詹克旭

(博世华域转向系统有限公司 上海 201821)

摘要 近些年,随着汽车行业的迅猛发展,汽车 ECU 的功能也越来越强大、越来越复杂。Bootloader 刷新功能的引入,对于汽车 ECU 的开发起到巨大的促进作用,已经成为汽车 ECU 的一个必备功能。针对汽车 ECU 的特点,提出一种基于 UDS 协议的汽车 ECU 升级方案。详细介绍汽车 ECU 的 Bootloader 刷写软件的实现原理。该方案不仅方便汽车电子研发人员进行开发和测试,而且对日后系统的升级和维护都有着重要作用。

关键词 汽车 ECU Bootloader UDS 协议 诊断服务 Flash

中图分类号 TP3 **文献标识码** A **DOI**:10.3969/j.issn.1000-386x.2019.01.035

AUTO ECU UPGRADE SCHEME BASED ON UDS PROTOCOL

Zhan Kexu

(BOSCH HUAYU Steering System Co., Ltd., Shanghai 201821, China)

Abstract In recent years, with the rapid development of the automobile industry, the function of automobile ECU is becoming more and more powerful and complex. The introduction of Bootloader update function has played a great role in promoting the development of automobile ECU, and has become an essential function of vehicle ECU. In view of the characteristics of automobile ECU, an auto ECU upgrade scheme based on UDS protocol was proposed, and the realization principle of Bootloader updating software for automobile ECU was introduced in detail. The scheme not only facilitates the development and testing of automotive electronics for R&D personnel, but also plays an important role in the upgrading and maintenance of the system in the future.

Keywords Automobile ECU Bootloader UDS protocol Diagnosis service Flash

0 引言

随着汽车制造业和快速发展的电子信息技术的不断变革,汽车工业的发展和进步被汽车电子技术的创新和应用不断地向前推进。汽车电子技术对改善汽车行驶舒适性、稳定性,提高汽车的安全性、经济性、动力性起到了非常关键的作用。此外也使汽车具备了通信、办公和娱乐等丰富功能^[1-3]。

汽车电子按用途和功能分为车身电子、汽车动力系统、安全系统和汽车底盘控制四类。由于汽车电子技术的发展,汽车中有越来越多不同功能的 ECU 控制单元被应用。为了实现全车智能化控制、共享数据信息,将所有的车载电器和 ECU 控制单元都搭载到 CAN 网络总线上。

由于汽车 ECU 应用功能变得越来越强大,实现起来也越来越复杂。为了完成整车的性能匹配和汽车相应模块的功能,在开发汽车 ECU 的过程中,需要频繁更新软件和相关参数。传统方法需要将零件从整车上拆卸下来,然后再通过调试端口进行程序更新,这将增加更多的工作量,并且容易对车辆本身造成损坏。为了解决汽车 ECU 升级的问题,本文提出了一种基于 UDS 协议的汽车 ECU 升级方案,并详细介绍了汽车 ECU 的 Bootloader 刷写软件的实现原理。

1 相关概念

1.1 Bootloader

Bootloader 是嵌入式系统在加电后执行的第一段

代码,在它完成 CPU 和相关硬件的初始化之后,再将操作系统映像或固化的嵌入式应用程序装载到内存中然后跳转到操作系统所在的空间,启动操作系统运行。对于 ECU 的 Bootloader 来说,其工作原理如下^[4-5]:

(1) Bootloader 就是 ECU 启动时候运行的一段程序,这段程序负责应用程序的更新。当进入 Bootloader 后,可以更新,也可以不更新。更新的话,Bootloader 更新完程序后,跳转到新程序运行;不更新的话,Bootloader 直接跳转到原来的程序去运行。

(2) Bootloader 更新完程序后并不擦除自己,下次启动后依然先运行 Bootloader 程序。

(3) 加入了 Bootloader 功能,就可以给 ECU 日后升级程序留出一个接口,方便日后 ECU 程序更新。当然,这就需要创建两个工程项目,一个为 Bootloader 工程,一个为 APP 工程。

(4) Bootloader 工程生成的文件通常下载到 flash 中的首地址,这样可以保证上电后先运行 Bootloader 程序。而 APP 工程生成的文件则下载到 flash 中 Bootloader 后面的地址中。也就是说,存在 flash 中的内容是分为两部分的。

(5) 要实现在同一个 flash 中保存两段程序,并且保证不能相互覆盖,则需要下载程序时指定地址。

1.2 UDS 协议规范

UDS 协议即 ISO14229 统一诊断服务,是诊断服务的规范化标准。比如读取故障码应该向汽车 ECU 发什么指令,读数据流又该发什么指令。作为诊断仪与汽车 ECU 之间进行诊断通信必不可少的一部分,一系列的诊断服务在诊断规范中被描述。诊断规范定义了诊断仪和汽车 ECU 之间的请求响应规则以及对于请求报文汽车 ECU 的处理行为^[6-7]。

汽车 CAN 诊断遵循 ISO15765 标准,其中应用层遵循 ISO15765-3 标准,网络层遵循 ISO15765-2 标准,ISO15765 体系结构如图 1 所示。标准寻址方式被应用诊断报文所采用。汽车 ECU 共有 2 个 CAN ID 用于接收诊断报文,一个是物理寻址 ID,一个是功能寻址 ID。ISO15765-2 网络层诊断报文分为流控帧报文、单帧报文、连续帧报文、第一帧报文^[8-9]。作为 CAN 通信软件设计的一个重要部分,汽车 ECU 需要解析这些收到的报文,从而得出完整的诊断服务。

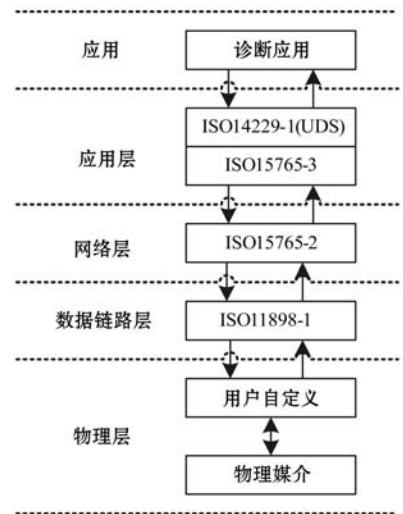


图 1 ISO15765 体系结构

2 总体实现方案

在汽车 ECU 中,Bootloader 是一个独立的程序,与系统的应用程序存放在汽车 ECU 不同位置的 flash 中。车辆故障诊断作为保障汽车安全性和可维护性的技术,在汽车电子的发展中越来越重要,UDS 诊断服务已广泛应用在汽车 ECU 上。根据这一特性,基于 UDS 协议的汽车 ECU 升级方案采用下载工具来更新应用程序。下载工具是一个上位机软件,应用程序以 S19 文件形式,通过 CAN 网络按照通信流程进行数据传输。汽车 ECU 与下载工具之间通过 UDS 协议的诊断服务实现数据交互,最终实现应用程序的升级功能。

本方案 Bootloader 无需拆卸,可通过 CAN 端口对汽车 ECU 编程。同时基于安全考虑,下载工具必须经过安全认证后才可以对 ECU 进行更新操作,安全认证的方式采用 UDS 协议的诊断服务实现。

2.1 系统要求

2.1.1 安全访问 SecurityAccess(27)

此服务的目的是为受限于访问安全、排放或安全原因的数据或服务提供一种访问方法。例如上传/下载程序数据至 ECU、例程控制、从 ECU 中读取特殊内存数据等诊断服务一般需要执行安全访问。安全访问的引入,可以防止因为不恰当地下载程序或数据至 ECU 导致破坏电子设备或其他汽车部件,对汽车的安全性及安全标准造成风险。安全访问的概念使用“种子”和“密钥”来实现。

诊断仪请求 ECU 解锁首先发送“SecurityAccess-RequestSeed”服务报文。ECU 发送一个种子进行响应,此种子是密钥计算算法的输入参数。诊断仪使用该种子计算出相应的密钥。

诊断仪通过发送包含密钥的“SecurityAccess-SendKey”服务报文给 ECU 来请求比较密钥。ECU 须将此密钥与内部存储或计算的密钥进行比较,如果两数相符,ECU 启动(解锁)诊断仪对特定服务和资料的访问权限,并通过“SecurityAccess-SendKey”服务指出。如果两数不相符,此访问被认为是一次错误的访问尝试。安全访问流程如图 2 所示。

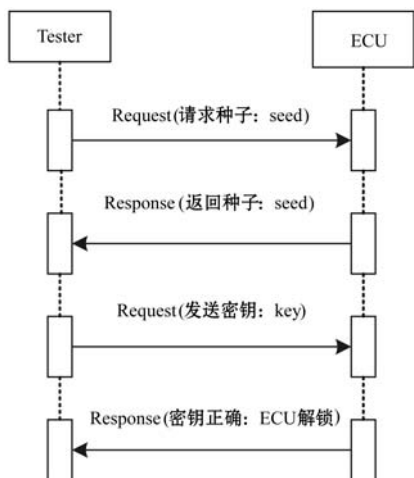


图 2 安全访问流程

2.1.2 升级条件检测

对于一些特殊的汽车 ECU,应该确保进行系统升级之前,整个系统状态满足升级条件。如果不符合系统升级条件,比如有车速或者有发动机转速等,这个时候应该拒绝升级请求。该检测是可选的,根据具体的汽车 ECU 情况和 OEM 需求来确认是否需要加入。

2.1.3 数据完整性验证

汽车 ECU 必须检查下载到内存中的程序的数据完整性。当一个程序下载完成之后,将通过 CRC 算法来核对当前传递过来并写入内存的所有数据是否正确。通过诊断服务的例程控制“CheckProgrammingIntegrity”请求来进行验证。在接收到该诊断服务请求之后,Bootloader 将计算下载的所有数据的 CRC 的值。然后将计算的 CRC 的值与传递过来的数据中 CRC 的值进行比较。如果两数相符,则认为下载程序成功。如果两数不相符,则认为下载程序的数据有错误,下载失败。

2.1.4 软件验证

定义一个标志来表示应用程序软件是否有效。如果程序完整性验证检查的结果是正确的,则认为 ECU 当前的软件状态是有效的,应用程序标志应该设被置为有效的。只有当标志有效时,才能运行应用程序,否则认为下载失败。

2.1.5 Flash 驱动下载

Flash 驱动是硬件相关的软件,提供的功能是对内存初始化、擦除和写入。

内存中的内容必须防止意外擦除和重写。为此,实现了一种软件锁机制,用于重新编程的 flash 驱动不存储在 ECU 非易失存储器中,只是在下载过程中才被下载到 ECU RAM 中。下载完成后,该驱动程序代码在 ECU 恢复正常工作模式之前将完全从 ECU 的 RAM 缓冲中删除。

编程过程中 flash 驱动程序的下载应该遵循指定的步骤,符合 UDS 协议诊断服务的要求。先进行请求下载 RequestDownload(34h),再进行传输数据(36h),最后进行请求退出传输 RequestTransferExit(37h)。

2.1.6 容错机制

BootLoader 程序应存储在受保护的易失性内存中,避免无意擦除。这样即使应用软件被擦除,ECU 也可以支持重新编程。在升级应用软件过程中,发生下列情况,ECU 应该能够重新下载:

- (1) 过/欠电压状态恢复后;
- (2) CAN 通信故障或超时恢复后;
- (3) 复位发生后;
- (4) 部分 flash 被擦除。

2.1.7 诊断服务需求

本系统诊断服务参考 ISO 14229 标准^[10]。为了让汽车 ECU 能够执行相应的操作,通过诊断仪将诊断服务发送给汽车 ECU。应用程序软件和 Bootloader 软件必须至少支持如表 1 所示的诊断服务。

表 1 诊断服务表

SID	诊断服务
0x10	诊断会话控制 DiagnosticSessionControl
0x11	电控单元重置 EcuReset
0x22	读取数据 ReadDataByIdentifier
0x27	安全访问 SecurityAccess
0x28	通信控制 CommunicationControl
0x2E	写入资料 WriteDataByIdentifier
0x31	例程控制 RoutineControl
0x34	请求下载 RequestDownload
0x36	数据传输 TransferData
0x37	请求退出传输 RequestTransferExit
0x3E	诊断设备在线 TesterPresent
0x85	控制 DTC 设置 ControlDTCSetting

2.2 Bootloader 启动序列

在 UDS 协议中,规定了诊断服务具有默认会话、程序设计会话和扩展诊断会话三种会话模式:

默认会话(10h 01h):此诊断会话模式为启动 ECU 的默认诊断会话,该会话模式不需要任何诊断应用程序的超时处理。

程序设计会话(10h 02h):此诊断会话模式支持 ECU 程序设计所需要的诊断服务。

扩展诊断会话(10h 03h):此诊断会话模式支持在 ECU 内存中调整功能和与安全相关的所有诊断服务,如安全访问、通信控制、写入资料、例程控制、控制 DTC 设置等。该会话模式也可以用来启动一些服务,并不限于特定用于功能调整的诊断服务^[11]。

基于不同的诊断会话切换关系,Bootloader 启动序列如图 3 所示。

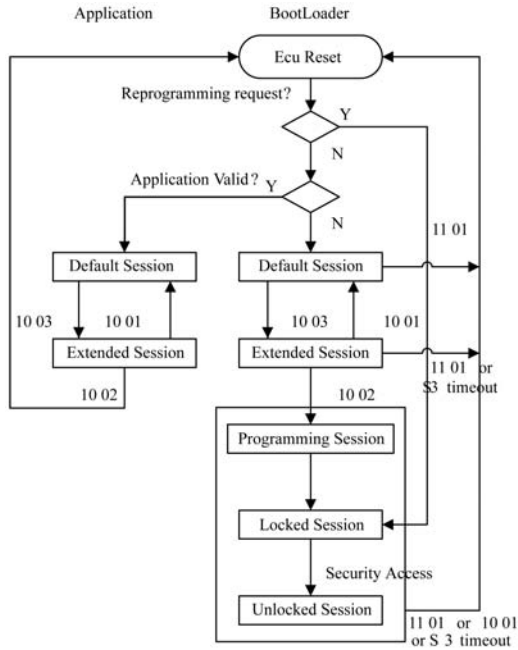


图 3 Bootloader 启动序列

从诊断会话的角度来看,进入编程会话必须通过扩展会话,这意味着 ECU 不支持直接从默认会话切换到编程会话。同样,ECU 不支持从编程会话直接切换到扩展会话。

如果 ECU 在正常条件下收到了“10h 02h”,ECU 将设置升级标志位有效并复位 ECU。ECU 复位后,首先进入 Bootloader,然后检查升级标志:

如果升级标志位有效,则不管应用程序是否有效都会保持在 Bootloader,此时进入程序设计会话模式。

如果升级标志位无效,则检查应用程序是否有效。如果应用程序无效,则继续保持在 Bootloader,ECU 进入默认会话模式;如果应用程序有效,则跳转至应用程序。

在 Bootloader 中,有些情况可导致 ECU 复位:

- (1) “11h 01h”会导致 ECU 复位;
- (2) 在程序设计会话和扩展诊断会话时,S3 超时而导致 ECU 复位;
- (3) 在程序设计会话时,“10h 01h”会导致 ECU 复位。

2.3 升级序列

更新程序顺序分为三个阶段^[12-14]:

- (1) 预编程:为编程设置好 CAN 网络;
- (2) 编程:下载应用程序;
- (3) 编程结束:重新同步 CAN 网络。

如果在预编程、编程和编程结束阶段有任何错误发生,升级过程都要重新再进行一次。

2.3.1 预编程

预编程阶段用于准备 ECU 编程的 CAN 网络。这一步还包含为了更新程序而进行的一些准备工作。预编程步骤如图 4 所示。

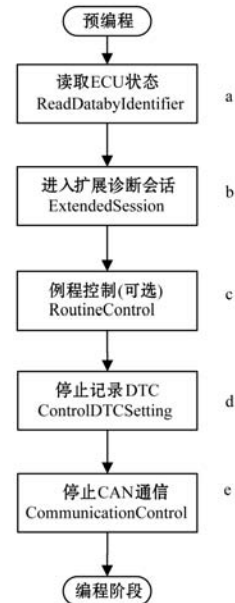


图 4 预编程步骤

(1) ReadDataByIdentifier 读取数据诊断服务(22h XXh YYh):读取 ECU 状态进行编程。从 ECU 读取数据标识,如 ECUHardwareNumber、BootSoftwareIdentification 等。读取数据 ReadDataByIdentifier 诊断服务是可选的,读取的内容由 OEM 定义。

(2) DiagnosticSessionControl 诊断会话控制诊断服务(10h 03h):为了能够停止正常的 CAN 通信和禁止记录 DTC,需要通过 DiagnosticSessionControl(10h)服务的 DiagnosticSessionControl(03h)实现让会话模式先进入到扩展诊断会话。为了进行例程控制、DTC 设置和通信控制做准备。

(3) RoutineControl 例程控制诊断服务(可选) - “检查预编程的条件是否满足”(31h 01h XXh YYh):这是为了确定 ECU 编程的条件,以确保系统的安全性。条件是由 ECU 来估计的,如果有任何不安全因素 ECU 应该拒绝编程。

注意:对于规定了需要该检测条件进行系统升级的 ECU,如果 ECU 收到“10h 02h”编程请求前没有收

到“检查预编程条件是否满足”(31h 01h), ECU 应该发送负响应拒绝跳入 Bootloader 进行编程模式。

(4) ControlDTCSetting 控制 DTC 设置诊断服务(85h 02h):禁止记录 DTC,以防止之后因为通信停止而误报节点丢失的错误。

(5) CommunicationControl 通信控制(28h 03h 03h):禁止 CAN 正常通信,为进入编程模式做准备。

2.3.2 编程

经过预编程之后,开始对 ECU 进行编程。图 5 为编程阶段流程。

输 TransferData (36h),请求退出传输 RequestTransferExit (37h)。flash 驱动所有字节下载后,需要进行“检查程序的完整性”的诊断服务(31h 01h),用来检查是否所有数据下载成功。

(5) RoutineControl 例程控制诊断服务 - “擦除内存”(31h 01h XXh YYh):擦除 ECU 的内存以便允许应用程序下载。通过使用例程控制 RoutineControl(31h)执行擦除内存服务。调用擦除内存诊断服务后,应用程序有效标志应该设置为无效。

(6) 下载过程诊断服务(34h,36h,37h):下载应用程序到 ECU 内存中应遵循一般的数据传输方法,使用以下诊断服务顺序:

- 请求下载 RequestDownload(34h);
- 传输数据 TransferData(36h);
- 请求退出传输 RequestTransferExit(37h)。

应用程序将被分解为若干程序块,通过多个传输数据诊断服务(36h)将应用程序完全下载到 ECU。

(7) RoutineControl 例程控制诊断服务 - “检查程序的完整性”(31h 01h XXh YYh): RoutineControl(31h)是用来检查程序下载是否成功、数据正确与否,这个诊断服务用于检查下载程序的完整性。

(8) EcuReset 电控单元重置诊断服务(11h):确认下载结束后,下载工具发送一个 EcuReset(11h)服务到网络来进行 ECU 硬件复位。Flash 编程过程通过 EcuReset 服务请求结束,跳出 Bootloader 以恢复正常 ECU 操作。硬件复位后,flash 驱动代码完全从 ECU 的 RAM 缓存中删除,以避免误操作出现意外的擦除。

2.3.3 编程结束

编程阶段结束之后,运行新的应用程序,恢复 ECU 到正常工作模式。图 6 为后编程阶段流程。

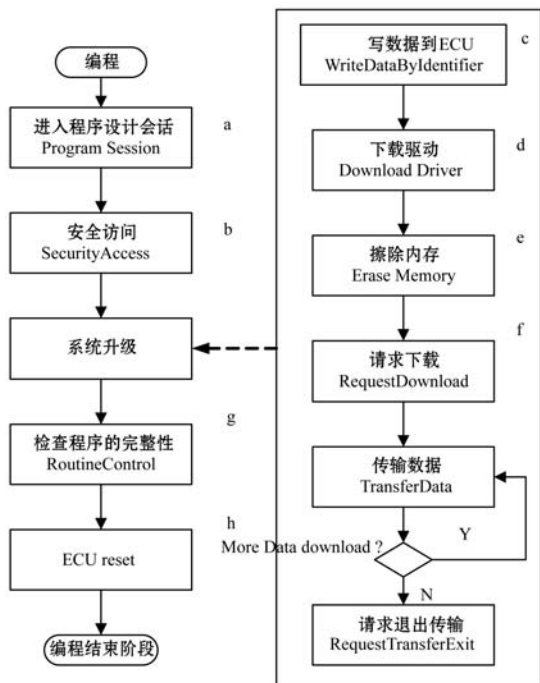


图 5 编程流程

(1) DiagnosticSessionControl 诊断会话控制诊断服务(10h 02h):首先要进入程序设计会话,发送程序设计会话请求。当 ECU 收到请求时,它应分配用于编程所需的所有资源。在 ECU 切换到编程会话之前,应该给出正响应。

(2) SecurityAccess 安全访问诊断服务(27h):更新程序需要在安全进入的情况下才可以进行。安全访问过程在下载前是强制的,安全访问只能允许经过安全认证的工作人员才可以对 ECU 升级程序。

(3) WriteDataByIdentifier 写入资料诊断服务(2Eh XXh YYh):强制性的写一个序列号和升级软件日期到 ECU。序列号和升级软件日期用于识别对 ECU 内存进行了操作记录。

(4) 下载 flash 驱动诊断服务(34h,36h,37h,31h):为了能够正常对 flash 进行下载程序的操作,首先必须先下载 flash 驱动。每次更新程序时,flash 驱动会先下载到 RAM 中。下载 flash 驱动需要用到如下诊断服务,请求下载 RequestDownload(34h),数据传

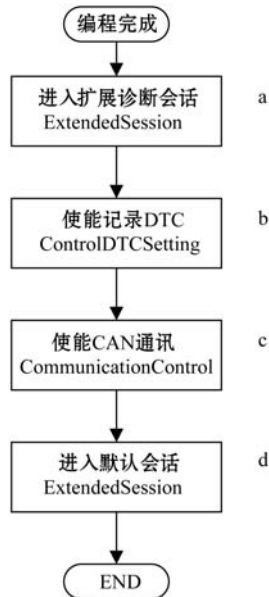


图 6 编程结束流程

(1) DiagnosticSessionControl 诊断会话控制诊断服务(10h 03h): 测试仪发送 DiagnosticSessionControl (10h)请求到 CAN 网络,ECU 切换到扩展诊断会话模式,以便能够使用通信控制和使能 DTC 控制诊断。

(2) ControlDTCSetting 控制 DTC 设置诊断服务(85h 01h):恢复可以正常记录 DTC 的状态。

(3) CommunicationControl 通信控制诊断服务(28h 00h 01h):使能 ECU 功能重新发送/接收消息,系统恢复到正常的通信模式。

(4) DiagnosticSessionControl 诊断会话控制诊断服务(10h 01h):ECU 切换到默认会话模式,整个系统恢复到正常工作状态,系统升级完成。

3 验证

将本文所设计的基于 UDS 协议的 Bootloader 程序经过编译后生成 S19 文件,然后将 S19 文件通过上位机下载工具下载至汽车 ECU 中。经测试,目标文件能够通过下载工具按照升级流程顺利下载到汽车 ECU 中,刷新数据如图 7 所示。下载后重新上电,仍能够通过 Bootloader 自动加载运行。

Request Data	Respon...	...	Response Data
22 F1 93	0.03226s	...	62 F1 93 30 3...
22 F1 80	0.02673s	...	62 F1 80 56 3...
10 83			
3E 80			
85 82			
28 81 01			
10 02	0.07725s	...	50 02 00 32 0...
27 11	0.00025s	...	67 11 0D CB 5...
27 12 13 0C CF D4	0.00287s	...	67 12
2E F1 98 39 38 37 36 35 3...	0.00285s	...	6E F1 98
2E F1 99 20 17 12 25	0.00284s	...	6E F1 99
34 00 44 FE DF 60 00 00 0...	0.00026s	...	74 20 0F FF
36 01 03 00 09 85 22 61 D...	0.00038s	...	76 01
37	0.00027s	...	77
31 01 02 02 6E D2 AD 4C	0.00568s	...	71 01 02 02 04
31 01 FF 00 44 00 00 F0 0...	3.91477s	...	71 01 FF 00 04
3E 80			
34 00 44 00 00 F0 00 00 0...	0.00027s	...	74 20 0F FF
36 01 32 30 31 37 2F 31 3...	0.01833s	...	76 01
36 02 FF FF FF 80 07 84 0...	0.01743s	...	76 02
36 03 06 06 0C FE D1 2D 2...	0.01742s	...	76 03

图 7 刷新数据

本文设计的汽车 ECU 升级方案的主要功能是通过 Bootloader 实现日后对应用程序和标定数据的刷写。因此,为了保证应用程序和标定数据的刷写,Bootloader 程序是不能被擦除和重写的。这一特性要求 Bootloader 在刷新过程中,无论经过多少次正确刷写、无论刷写过程中出现何种情况造成刷新失败,该 Bootloader 应该都具有重复刷写的功能。

在刷新过程中,每一步的实现都应遵照 UDS 诊断服务。当发生异常情况导致刷新流程终止,根据 UDS 诊断服务的刷新流程,其中某一步没有通过或通信超时,Bootloader 都会回复一个负响应码,同时跳出刷新

流程。刷新失败后,可以通过负响应码来判断刷新失败的原因,待问题解决后重新对系统进行刷新。

为了验证基于 UDS 协议的汽车 ECU 升级方案的安全性和可靠性,进行了如下几种测试:

(1) 过/欠电压测试:在系统升级过程中,调整电源电压,人为制造过/欠电压条件,当恢复供电电压后重新开始刷新。经验证,软件仍然可以正确刷新。

(2) CAN 通信异常测试:在系统升级过程中,停止 CAN 通信,待恢复 CAN 通信后重新开始刷新。经验证,软件仍然可以正确刷新。

(3) 复位测试:在更新过程中对汽车 ECU 进行复位操作,复位结束后软件仍然可以正确刷新。

(4) 多次刷新:在多次正确刷新过程中,软件每一次都可以正确地下载到汽车 ECU 中,并且没有出现任何程序故障。

通过人为制造各种异常情况,经过反复多次测试,系统仍能够保证按照正常的刷新流程实现程序的升级。经验证,基于 UDS 协议的汽车 ECU 升级方案安全可靠,系统稳定性高,对于刷新过程中可能出现的各种异常情况,能够通过自身的安全机制避免程序出错,有良好的抗外界干扰能力。测试结果证明,该汽车 ECU 系统升级方案具有良好的可用性。

4 结语

基于 UDS 协议的汽车 ECU 升级方案,实现了基于 CAN 总线应用程序的在线更新。在不需要拆卸的情况下,即可实现汽车 ECU 的应用程序升级,减少了工作量,提高效率。该方案解决了汽车电子产品软件难以升级的问题,同时为了保证汽车 ECU 软件更新的一致性,采用基于 UDS 协议的诊断服务来进行汽车 ECU 系统升级,严格遵守 ISO14229 统一诊断服务的更新流程,为软件更新的可靠性提供了保障。

本文设计的升级解决方案已经被应用到很多不同的汽车电子产品上,覆盖了多个主机厂的不同车型。该方案的提出,对整车的研发制造有着重要的意义,在保证汽车电子软件产品质量的同时大大缩短了软件的开发周期,其作用覆盖了从开发到量产的整个过程。同时,该系统是基于 UDS 协议的诊断服务实现的,具有很好通用性和可扩展性,有利于日后软件升级和维护。基于 UDS 协议的汽车 ECU 升级方案,由于其系统的实现简单、便于理解、稳定性高等优点,相信在不久的将来,将广泛应用到各大主机厂及汽车零部件供应商的汽车电子产品上,有很好的发展前景。

TextRank 算法赋权值后的分类 F1 值高于 TF-IDF 算法。

实验结果表明,用 TextRank 算法对语料赋权值后的分类结果优于用 TF-IDF 算法对语料赋权值后的实验结果。并且由于该方法下的分类准确率、召回率、F1 值均达到 97% 以上,说明本研究提出的基于 SVM 的学科试题分类方法是可行的。

7 结 语

研究表明,基于 SVM 的学科试题分类方法是有效的,该方法可对学科试题进行较为准确地分类。在本研究中,将学科试题按照课本章节分为 12 类,因为类别是按照“章”划分的,故目前根据分类结果只能确定试题知识点的大概范围。下一步将尝试对试题进一步分类,得到更加具体的知识点。

参 考 文 献

- [1] 古丽娜孜·艾力木江,乎西旦·居马洪,孙铁利,等. 基于支持向量的最近邻文本分类方法[J]. 智能系统学报, 2018, 13(5):799-807.
- [2] 贺鸣,孙建军,成颖. 基于朴素贝叶斯的文本分类研究综述[J]. 情报科学, 2016, 34(7):147-154.
- [3] 黄正伟,唐芳艳. 基于 SVM 分类模型的垃圾文本识别研究[J]. 数学的实践与认识, 2016, 46(7):144-153.
- [4] 崔建明,刘建明,廖周宇. 基于 SVM 算法的文本分类技术研究[J]. 计算机仿真, 2013, 30(2):299-302, 368.
- [5] 阿力木江·艾沙,吐尔根·依布拉音,库尔班·吾布力,等. 基于 SVM 的维吾尔文文本分类研究[J]. 计算机工程与科学, 2012, 34(12):150-154.
- [6] 朱刘影,杨思春. 基于 SVM 的地理试题自动分类[J/OL]. 计算机应用研究, 2018, 35(9). [2017-08-28]. <http://www.aocmag.com/article/02-2018-09-025.html>.
- [7] 夏晔,钱松荣. SVM 算法在网站分类中的应用研究[J]. 计算机应用与软件, 2012, 29(11):222-224.
- [8] 周庆平,谭长庚,王宏君,等. 基于聚类改进的 KNN 文本分类算法[J]. 计算机应用研究, 2016, 33(11):3374-3377, 3382.
- [9] 杨帅华,张清华. 基粗糙集近似集的 KNN 文本分类算法研究[J]. 小型微型计算机系统, 2017, 38(10):2192-2196.
- [10] 赵燕,李晓辉,周云成,等. 基于朴素贝叶斯的农业文本分类方法研究[J]. 节水灌溉, 2018(2):98-102.
- [11] 高元,刘柏嵩. 基于集成学习的标题分类算法研究[J]. 计算机应用研究, 2017, 34(4):1004-1007.
- [12] 马丽菲,莫倩,杜辉. 面向中文短影评的分类技术研究

[J]. 山东大学学报(理学版), 2016, 51(1):52-57.

- [13] 李航,唐超兰,杨贤,等. 融合多特征的 TextRank 关键词抽取方法[J]. 情报杂志, 2017, 36(8):183-187.
- [14] 杨林青,余瀚,费宁,等. 一种基于 TextRank 的单文本关键词提取算法[J]. 计算机应用研究, 2018, 35(3):705-710.
- [15] 郑莉,董渊,何江舟. C++ 语言程序设计[M]. 4 版. 北京:清华大学出版社, 2010.
- [16] 李稚楹,杨武,谢治军. PageRank 算法研究综述[J]. 计算机科学, 2011, 38(10):185-188.

(上接第 196 页)

参 考 文 献

- [1] 亓瑞环. 浅析汽车控制的研究现状与展望[J]. 时代汽车, 2017(8):40-41.
- [2] 张情. 汽车电子前沿技术及未来发展趋势[J]. 时代汽车, 2017(12):46-47.
- [3] 闫利利. 汽车电控单元刷新的标准化研究[C]//第十二届中国标准化论坛文集, 2015:1720-1724.
- [4] 汪庆武,郇钲,黎泽清,等. 基于 SPI 接口 DSP 程序引导加载方法实际与实现[J]. 仪表技术, 2016(5):6-8.
- [5] 刘坤,韩朝智. 浅析基于 ARM 嵌入式开发的 BootLoader 设计及其实现[J]. 电子技术与软件工程, 2016(2):203-204.
- [6] 陈姿霖,宋磊锋,张龙岗,等. 基于 UDS 的整车诊断系统设计方法[J]. 汽车电器, 2017(4):14-17.
- [7] 罗峰,郁静华. 汽车统一诊断服务诊断协议网络层测试方法[J]. 同济大学学报(自然科学版), 2016, 44(4):632-636.
- [8] 游长能. 基于 LabVIEW 的 CAN 总线 UDS 诊断工具开发[J]. 电子测试, 2016(19):59-60.
- [9] 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(2):166-170.
- [10] ISO 14229:2013, Road vehicles-Unified diagnostic services (UDS)-Specification and requirements[S].
- [11] 王琦,黄悦鹏,邢正阳,等. 基于 CAN 总线的 Bootloader 设计与实现[J]. 微型机与应用, 2015, 34(18):14-16.
- [12] 刘一平. 基于 HIS 协议的车载 Bootloader 的研究与实现[D]. 成都:电子科技大学, 2015.
- [13] 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(4):20-23.
- [14] 陈彤,黄立梅. 一种用于汽车电控单元 CAN Bootloader 的设计与实现[J]. 汽车实用技术, 2016(9):156-160.