基于 UDS 协议的车辆 OTA 开发

吴成加，郑 磊，夏宇生，郝绪程
（劲旅环境科技股份有限公司，合肥 230051）

摘 要：阐述车辆 OTA 的工作原理，参照 UDS 服务协议，针对采用 S32K146 微处理器的车载 ECU 设计一种 OTA 升级系统，在 TBOX 接收升级包后通过车载 EUC 进行 BootLoader 下载。应用结果表明，将 OTA 用于车辆软件升级，可以提高车载 ECU 软件更新的快捷性。

关键词：UDS 协议；OTA 升级；ECU；TBOX；S32K146 微处理器
中图分类号：U469；TP31 **文献标志码**：A **文章编号**：1006-3331(2024)03-0039-06

Application of In-vehicle OTA Based on UDS Protocols

WU Chengjia, ZHENG Lei, XIA Yusheng, HAO Xucheng
(Jinglv Environment Science and Technology Co., Ltd., Hefei 230051, China)

Abstract: This paper describes the working principle of vehicle OTA. According to the UDS service protocols, the paper designs an OTA upgrade system for the on-board EUC using an S32K146 microprocessor, which downloads the upgrade package through the on-board EUC by BootLoader after TBOX receives the upgrade package. The application results show that using OTA for vehicle software upgrade can improve the fastness of vehicle ECU software update.

Key words: UDS protocol; OTA upgrade; ECU; TBOX; S32K146 microprocessor

随着汽车电控系统功能日趋丰富和结构日益复杂，其软件需要不断更新和升级[1]。传统的解决办法是召回返厂或到服务站对系统软件进行统一升级，用户的时间成本较高，特别是偏远地区的用户。因此，通过空间下载技术（OTA）进行软件在线升级变得尤为重要，这不仅可以大幅降低成本，而且在线下载或诊断具有传输速度快、性能稳定等优点。本文基于统一诊断服务（UDS）协议[2]，结合恩智浦 S32K 系列 32 位微控制器，采用 OTA 技术设计一种车辆远程软件升级系统，通过车辆 CAN 总线实现电子控制单元（ECU）程序代码的更新，为用户带来更好的服务和体验[3]。OTA 升级类型多样，包括应用软件更新、配置参数更新和固件版本升级等。

1 总体方案设计

1.1 数据传输系统的组成

数据传输系统包括传输计算机、OTA 云端服务器和车载终端设备，如图 1 所示[4]。传输计算机可以是台式电脑也可以是便携式笔记本电脑，可以放在办公室、车辆内或其他任何地方。云端服务器是车辆运行信息监控和 OTA 升级平台，用于实现与远程车辆之间的运行信息收发、存储和监控。云端服务器分别与传输计算机、车辆上的终端设备相连接。车载终端设备包含远程通信终端模块（TBOX）、一个或多个电子控制单元（ECU）。

TBOX 作为车辆与外媒沟通的桥梁[5]，通过 CAN 总线网络与 ECU 相连接，其主要用于终端设备与云

收稿日期：2024-03-05。
基金项目：安徽省重点研究和开发计划项目（1804a09020010）。
第一作者：吴成加（1971—），男，高级工程师；主要从事新能源汽车电驱动及控制系统核心零部件关键技术研究工作。E-mail：wuchengjia@126.com。

端服务器之间的双向数据传输,实现车辆对云端服务器信息的接收和车辆实时运行信息、诊断信息的上传。

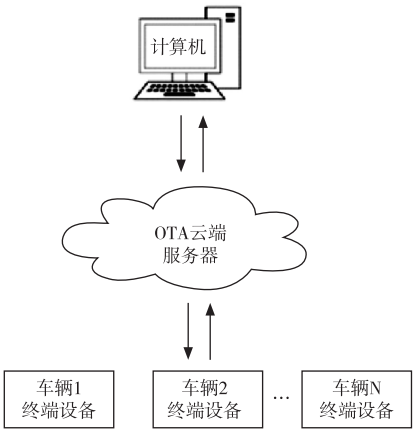

图 1 OTA 数据传输系统的组成框架

正常情况下,各车载 TBOX 接收其车辆一个或多个 ECU 的信息,并将这些信息上传到云端服务器平台。OTA 升级时,传输计算机将升级包发送到云端服务器平台,由云端服务器发送升级包给相应的车载 TBOX,再由 TBOX 传输给相应的 ECU,从而实现远程下载。

OTA 升级软件保存在 TBOX 模块的嵌入式多媒体卡(EMMC)中,当车辆上电后,检测到有升级包且车辆当前状态满足升级条件时,CAN 总线进入升级模式,车辆的 ECU 接收 TBOX 通过 CAN 总线传输的报文,进入在线更新升级模式后,各 ECU 停止发送信息,CAN 总线进入静默状态。TBOX 模块发出的命令帧标识将根据下载目标 ECU 的不同而随之改变。当要升级的 ECU 接收到该命令帧标识信息后,进行应答并连续接收数据,其具体过程为:TBOX 模块接收到待升级 ECU 的应答指令后,将连续数据字节拆分成若干个连续数据帧,进行连续程序下载发送。ECU 将收到的连续数据进行协议解析和校验后,再将这些数据写入到 ECU 处理器内部的 FLASH 指定空间进行存储,直至数据传输结束,完成在线下载。

1.2 UDS 协议中的相关服务

UDS 是汽车电子 ECU 的一种诊断通信协议,它是 ISO 14229(全球通用的汽车诊断技术标准)中的一个应用层协议,具体内容在 ISO 14229-1 中有描述[6]。UDS 提供了一个诊断服务的基本框架,主机厂和零部件供应商可以根据实际情况选择实现其中的一部分或是自定义出一些个性化的诊断服务。在远程 OTA 系统中,采用 UDS 诊断服务协议可便于开发在线检测设备,实现车联网功能,以及进行车辆售后维修保养。

在 ISO 14229 诊断规范中,发送端通过 CAN 总线发送指令和数据,ECU 根据发送端的 UDS 服务请求作出响应,并将接收到的数据写入微处理器的内部 FLASH 空间,实现程序代码更新。发送端与微处理器之间实现诊断、通讯管理功能所需要的服务包括:

1) 0x10 服务。此服务是诊断会话控制服务,该服务包括默认会话、扩展会话、编程会话 3 个模式。发送端通过发送会话指令进入相应的会话模式,使 ECU 进入不同的服务请求状态,以便执行对应的诊断任务和编程操作[7]。

2) 0x22、0x2E 服务。0x22、0x2E 服务分别是数据标识符(DID)读取、写入数据服务。发送端可以通过该服务对 ECU 进行相关参数和配置信息、软件版本号信息、指纹信息的读取或写入。

3) 0x27 服务。该服务为安全访问服务,主要用于修改存储在 ECU 内存中的数据。上位机向下位机请求随机种子,并将接收到的随机种子按照指定的安全加密算法计算出一个密码值,下位机接收上位机算出来的密码并与内部算出的密码进行比较,如果两个密码一致则解锁成功,否则解锁失败[8]。

4) 0x31 服务。此服务为例程控制服务,用于执行特定程序的控制操作,包括启动程序、停止程序以及请求运行结果等。

5) 0x34、0x35 服务。0x34、0x35 服务分别用于请求下载和请求上传数据,用于发送端向接收端请求下载数据,和用于接收端向发送端请求上传数据。

6) 0x36 服务。0x36 服务用于进行数据的传输操作,当 0x36 服务完成数据传输且校验正常后,发送端发出 0x37 服务表示数据传输完成。

7) 0x3E 服务。此服务用于告知 ECU 单元测试工具仍在线,该服务通过周期性发送,维持当前激活的非默认诊断会话模式,以确保诊断服务或者之前激活的通信处于激活状态。

8）0x85、0x28 服务。该服务为通讯控制服务，0x85 服务用于开启和关闭 ECU 的故障代码（DTC）诊断；0x28 服务用于开启和关闭 ECU 的报文传输。在程序升级过程中，可通过这两个服务关闭其他 ECU 单元的 DTC 故障诊断和报文传输，以减少总线负载，提升系统传输的可靠性。

1.3　S32K 系列 32 位微控制器内存分配

电动汽车的 CAN 网络控制单元是由整车控制器、主驱动器、辅驱控制器、电池管理系统、高压控制系统、车身控制系统、仪表显示单元、空调单元、远程监控系统等控制节点组成的一个或多个 CAN 网络。本文 ECU 模块的微控制器采用恩智浦 S32 平台 32 位车规级处理器 S32K146。该处理器采用 ARM 内核（Cortex-M4）IP 和 ARMv7-M 架构集成的片上系统（SoC）。SoC 内部的 FLASH、SRAM、RAM 区域都能保存程序，具有高可扩展性、高性能和低功耗等特点，适用于车身、区域控制和电气化应用。

S32K146 处理器包括一个 1 MB 的 PFlash 空间、一个 128 KB 的 FlexRAM 空间和一个 4 KB 的 FlexNVM 空间，其内存分布如图 2 所示。其中 PFlash 的空间用于 ECU 系统，将其划分为引导加载程序区（BootLoader）、应用代码区（APP1）、备份代码区（APP2）和标记区[9-10]；FlexRAM 空间包括低速静态随机存取存储器（SRAM_L）和高速静态随机存取存储器（SRAM_U），SRAM_L 和 SRAM_U 在物理上和逻辑上形成连续的块内存映射；FlexNVM 空间用于存储数据或用作可擦除可编程存储器空间。

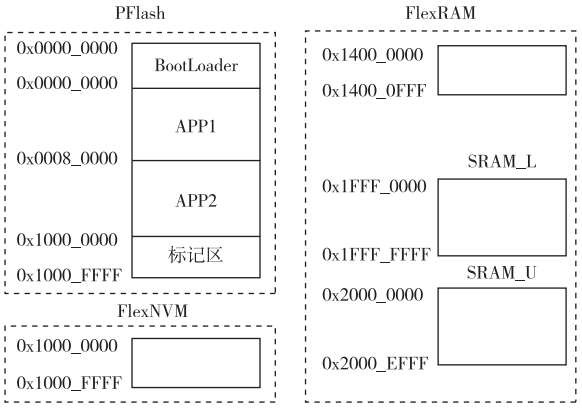

图 2　S32K146 处理器内部存储器空间分布图

引导加载程序区（BootLoader）的主要功能是进行硬件配置、CAN 控制器模块初始化、CAN 中断接收、FLASH 擦写和代码拷贝等任务；应用代码区 APP1 的内存空间为 496 KB，主要用于保存当前用户运行的应用层代码；备份代码区 APP2 的空间大小与应用代码区 APP1 相同，主要用于保存用户备份的代码，当新系统远程下载失败时，可将备份区的代码拷贝到用户区，对用户区的代码进行覆盖，实现代码回滚功能。

标记区的内存空间为 16 KB，用于保存系统运行的动态安全密钥，在代码运行或进行固件更新时，需要通过软件计算获得一个安全密钥，并保存在标记区。如果标记区的密钥与发送的密钥不一致，系统将无法进一步运行。

2　OTA 系统的设计及验证

OTA 升级就是 TBOX 与 ECU 之间按照 UDS 协议进行车辆软件在线升级的过程。在 OTA 升级过程中，ECU 启动时会先运行存储在 BootLoader 区的引导程序，以判断 ECU 是否需要升级。如果不需要升级就从 BootLoader 区跳转到 APP1 区运行用户程序代码；如果需要升级则先通过 TBOX 接收和搬运要升级的软件，保存到 APP1 区，待升级完成后，再从 BootLoader 区跳转到 APP1 区运行升级版的软件，从而实现 OTA 升级。

2.1　车载 TBOX 的设计要求

TBOX 是汽车内部的核心通讯组件，它是车辆与云端之间通信的桥梁，其功能包括远程监控、远程控制、数据传输、OTA 升级等，TBOX 是 OTA 功能实现的关键。

OTA 升级过程中，TBOX 负责接收并保存云端存储的升级包文件。这一过程通过无线网络完成，确保与云端数据的顺畅交互。上传到云端的升级文件被安全存储在云端服务器中，当需要传输数据时，TBOX 设备会通过无线网络与云端服务器建立连接，实现高效、稳定的数据传输。

TBOX 采用严密的验证机制，确保升级包不被篡改，并将传输状态实时反馈给云端服务器。若数据在传输过程中遭遇中断，TBOX 支持断点续传功能，确保数据完整无误地完成传输，这一过程的设计旨在提供稳定、可靠的数据传输环境，确保每一次 OTA 升级

都能顺利完成。

2.2 基于 UDS 协议的 OTA 软件设计

2.2.1 BootLoader 的在线升级设计

基于 UDS 协议的车载 ECU 数据传输首先在于 CAN BootLoader 的在线升级，涉及的 UDS 服务包括握手、安全校准、数据传输、FLASH 擦写、数据写入等。其软件刷写流程可分为预编程、主编程、后编程 3 个阶段。

1）预编程阶段。如图 3 所示，首先系统进入扩展模式并检查预编程条件，然后停止 DTC 设置，禁止无关通讯，最后读取版本号和 ECU 的相关参数信息。

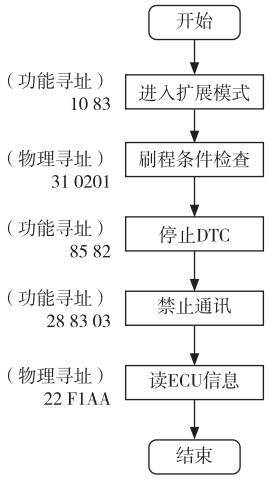

图 3 UDS 预编程流程图

2）主编程阶段。如图 4 所示，系统进入编程会话后，发送请求种子和密钥，当密钥验证成功后，系统启动传输请求，得到正响应信号后，发送端开始 APP 数据传输过程，TBOX 将要发送的 APP 文件进行拆分，分段进行传输，ECU 在擦除 FLASH 空间后接收这些数据并保存，当接收的数据达到 1 KB 时，ECU 将这些数据按页写方式写入到 FLASH 指定的地址中，继续接收并写入 FLASH，直至 TBOX 数据传输结束，ECU 请求传输停止，然后 ECU 检测程序的一致性和完整性，并写入软件相关标识后 ECU 复位，完成主编程过程。

3）后编程阶段。主编程完成后，ECU 复位，如图 5 所示，系统再次进入扩展模式，并发送指令恢复通讯功能，开启 DTC 诊断、清除刷写 ECU 的故障信息，进入默认会话模式。至此完成整个 UDS 的 CAN 传输过程。

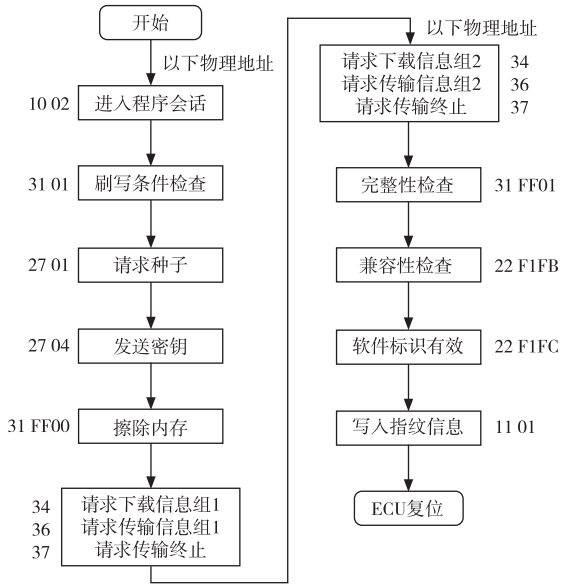

图 4 UDS 主编程流程图

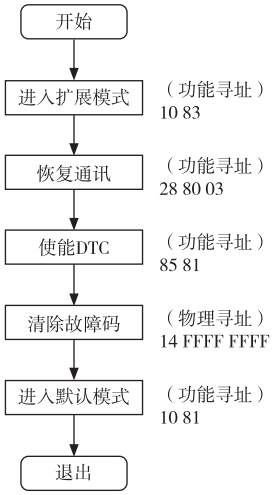

图 5 UDS 后编程流程图

2.2.2 车载 ECU 的数据更新传输软件设计

车载 ECU 的数据更新传输过程主要包括信息握手、系统解锁、ECU 芯片擦除、数据接收和校验等。

1）信息握手。当升级软件由 OTA 云端推送至目标车辆，并由车载端 TBOX 成功接收后，TBOX 将升级包保存在其内部的 EMMC 存储器中，等待下一次车辆重新上电后，TBOX 作为发送端通过 CAN 总线发送升级包更新请求信息，车辆 CAN 总线上的 ECU 根据接收到的信息进行判断。当匹配的 ECU 确认车辆当前状态（包括当前挡位、车速、READY 等信息）满足升级条件后，该 ECU 发送确认信息进行握手，

ECU 进入扩展模式，并检查预编程条件、停止 DTC 功能，禁止应用层通讯，随后系统复位并跳转到 BootLoader 代码中，并在 BootLoader 代码中进入主编程模式。

2）系统解锁。在主编程模式下，发送端（TBOX）发送请求进入编程会话模式，如图 4 所示。系统首先进行刷写条件检查，TBOX 发送请求种子去请求 ECU 解锁，ECU 在肯定应答消息中发送一个随机种子，TBOX 收到 ECU 提供的随机种子，通过解密算法计算，得到密钥并返回给 ECU，ECU 将接收到的密钥信息与自身计算的密钥值进行比对，如果两个密钥匹配，则系统解锁。反之，则提示密码错误，无法进入下一步。

3）ECU 芯片擦除。发送端 TBOX 接着发送擦除内存请求，在擦除内部 FLASH 之前，ECU 先解锁 FLASH，将当前的 APP1 区代码拷贝到 APP2 区中进行软件备份后（即将 0x4000－0x7FFFF 地址中的数据复制到 0x80000－0xFBFFF 地址空间），再将 FLASH 的地址空间从 0x4000－0x7FFFF 位置擦除。

4）数据接收。当发送端 TBOX 接收到 ECU 擦除成功的信息后，请求数据下载，发送端收到正响应信号后，开始数据传输。传输数据是按信息组进行，S32K146 处理器内部 FLASH 数据是按页写入。因此，当 ECU 成功接收到 1024 字节有效数据后，传输过程停止，待 ECU 将该段数据成功写入到指定地址后，发送端再次进行传输，直至数据全部传输结束。

5）数据校验。当所有数据传输结束并通过代码的完整性、兼容性检查后，发送端将软件的有效标识及当前的软件版本、参数等信息传输给 ECU，由 ECU 将这些信息写入到标记区，并复位微控制器，退出传输过程。

2.2.3　ECU 更新后运行及代码回滚设计

当 ECU 的数据更新结束，ECU 上电复位后，数据传输过程进入后编程阶段，发送端再次进入扩展模式，请求系统恢复通讯，并开启 DTC 诊断，清除故障码后，重新进入默认会话模式。至此，更新过程结束。

图 2 所示的 0x000FC000－0x000FFFF 所处的标记区的系统状态值，在进行 OTA 更新后会被 BootLoader 修改。因此，ECU 系统上电复位后首先会读取标记区的系统状态值，判断出当前处于 OTA 的后编程状态。后编程状态下，应用程序软件跳转到更新后的代码区（APP1 区）运行，如果此时代码能正常运行，则在 APP1 区中的软件同样修改标记区中的系统状态值。由此表明，APP1 中更新后的代码能正常运行。

当某些原因导致 APP1 中的代码不能正常运行时，则系统宕机，同时无法修改标记区中的系统状态值。当 ECU 系统再次上电，BootLoader 应用程序发现该值处于异常状态，则在 BootLoader 应用程序中执行代码回滚功能，先将 APP1 代码区擦除，再重新拷贝 APP2 区中的代码到 APP1 区中，使系统还原到 OTA 更新之前的状态。此时，系统仍运行更新前的代码，保证了 OTA 异常后系统仍能正常运行。

2.3　OTA 升级功能验证

完成相关的 UDS 协议及软件开发后，对 18 t 纯电动道路洗扫车的 ECU 进行 OTA 升级测试，如图 6 所示。

图 6　OTA 系统在线升级

1）升级准备。将内含 UDS 通讯协议的 TBOX 模块、ECU、监控计算机均接入整车 CAN 网络。首先将待更新的软件发送到 OTA 云端服务器，再由 OTA 云端服务器将升级包传送给 TBOX 模块，测试车辆的 TBOX 接收到该信息后，车辆仪表上显示终端提示有更新服务。此时确保车辆处于停驶状态，使当前状态符合升级条件。

2）更新阶段。通过计算机实时监控总线通讯状态，在 ECU 重新上电并初始化后，总线监测到 TBOX 与 ECU 之间进入预编程模式，随后，ECU 停止对外发送信息，总线上其他模块的 CAN 活动信息消失。TBOX 进入主编程模式，开始按 UDS BootLoader 的传

输流程将升级程序文件下载到 ECU 中。此时,通过 CAN 接收工具可以实时监测数据流,部分数据流如图 7 所示。

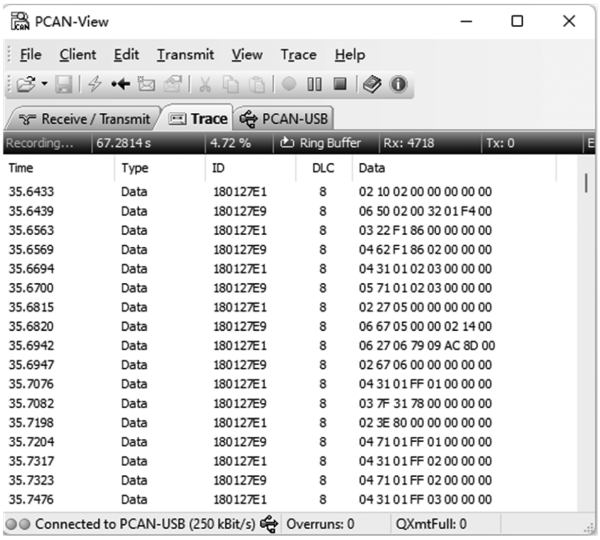

图 7 部分数据流信息

3) 结束阶段。下载结束后 TBOX 进入后编程模式,此时会向 ECU 的标记区写入相关的指纹信息,之后复位 ECU,将车辆重新上电。经过实车测试,代码可以得到正确的更新,上述流程经过多次反复测试,其间不断切换当前代码及备份区中的备份代码进行回滚测试,均满足设计要求,达到了预期效果。

3 结束语

本文论述了以 S32K 系列微控制器为核心的 CAN BootLoader 系统,并基于 UDS 诊断协议进行 OTA 系统升级的设计方案。该设计方案已成功应用于 18 t 纯电动道路洗扫车 ECU 的 OTA 升级。

参考文献:

[1] 陈睿智. 基于 UDS 协议的汽车电控单元故障诊断服务设计与实现[D]. 合肥:中国科学技术大学,2021.

[2] 刘佳楠. 汽车电子控制器硬件抽象与软件开发[D]. 成都:电子科技大学,2011.

[3] 吴成加. 基于 CAN Bootloader 的电动汽车远程数据更新系统设计[J]. 客车技术与研究,2014,36(6):27-30.

[4] 黄悦鹏. 基于 CAN 总线的 UDS 诊断系统的设计与实现[D]. 南京:南京邮电大学,2016.

[5] 李志辉. 北汽新能源汽车 TBOX 软件设计与实现[D]. 大连:大连理工大学,2017.

[6] 马建辉,于良杰,王勇,等. 基于 UDS on CAN 的 BootLoader 设计[J]. 单片机与嵌入式系统应用,2019,19(3):7-9.

[7] 杨朝阳,黄凯金,仝秀峰,等. 基于 UDS 的 Bootloader 上下位机设计[J]. 软件,2023,44(7):42-47.

[8] 吴冬梅. 基于 UDS 协议的车载 Bootloader 设计与实现[D]. 天津:天津科技大学,2022.

[9] 聂幸福,孟晨兴. 基于 UDS 的 BootLoader 上位机实现[J]. 汽车工业研究,2018(7):26-29.

[10] 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.