DOI:10. 19471/j. cnki. 1673-6397. 2023. 02. 008

基于 UDP 的混动商用车巡航控制器 OTA 系统开发

汪志莹[1],李军伟[1*],李兴坤[2],李连强[3],许金鹏[1],刘宝岩[1]

1. 山东理工大学 交通与车辆工程学院,山东 淄博　255049;

2. 北京裕峻汽车技术研究院,北京　100016; 3. 一汽解放青岛汽车有限公司, 山东 青岛　266217

摘要:为实现混动商用车巡航控制器软件及时更新和远程调试,并保障系统的安全性,设计开发一种基于用户数据报协议(user datagram protocol,UDP)的空中下载技术(over the air technology,OTA)升级系统。设计 OTA 系统的通信模块、握手模块、命令行调试模块、文件下载模块、备份升级模块和服务器端界面,并进行测试验证。结果表明:基于 UDP 协议的混动商用车巡航控制器 OTA 系统能够完成报文传输、通信加密、系统调试、软件包传输、软件升级及备份的功能,实现了软件升级的安全性、可靠性、便捷性。

关键词:商用车;巡航控制器;UDP 协议;OTA;加密;解密

中图分类号:U463. 67　　**文献标志码**:A　　**文章编号**:1673-6397(2023)02-0042-08

引用格式:汪志莹,李军伟,李兴坤,等. 基于 UDP 的混动商用车巡航控制器 OTA 系统开发[J]. 内燃机与动力装置,2023,40(2):42-49.

WANG Zhiying, LI Junwei,LI Xingkun, et al. Development of OTA system for hybrid commercial vehicle cruise controller based on UDP[J]. Internal Combustion Engine & Powerplant, 2023,40(2):42-49.

0　引言

随着移动通信技术的飞速发展和汽车新四化(电动化、网联化、智能化、共享化)建设的快速推进,车载控制器的功能越来越复杂,软件更新越来越频繁。空中下载技术(over the air technology,OTA)有效降低了控制器软件升级成本,为用户提供了更多的订阅服务[1-2],实现了车辆驾驶辅助系统、空气悬架系统、车辆导航系统、影音娱乐系统、自动泊车功能等系统的更新[3],已成为未来汽车智能化发展的主要趋势,相比手机等智能设备,车载 OTA 的应用环境更为复杂[4]。

为了提高混动商用车巡航控制器软件升级效率,减少现场升级次数及人工调试成本,本文中从实际工程出发,设计了一套基于用户数据报协议(user datagram protocol,UDP)的混动商用车巡航控制器的 OTA 升级系统,通过移动网络实现软件升级服务,同时为开发人员提供远程调试功能,并通过混动商用车巡航控制器测试验证。

1　系统总体方案和实现功能

1.1　总体方案

混动商用车巡航控制器 OTA 系统基于 Linux 系统开发,分为服务器端和控制器端。为满足数据传输

收稿日期:2022-11-26

基金项目:山东省重大科技创新工程项目(2019JZZY01091)

第一作者简介:汪志莹(1998—),男,山东济宁人,硕士研究生,主要研究方向为电动汽车控制技术,E-mail: m17853314101@163. com。

* **通信作者简介**:李军伟(1964—),男,河南平顶山人,工学博士,教授,主要研究方向为电动汽车关键控制技术,E-mail: ljwhitt@163. com。

的安全性,服务器端和控制器端之间必须有完整的检查机制和不同的加密机制[5]。服务器端安装在云服务器上,用于存储控制器升级所需的软件包,并完成升级任务的分发;控制器端安装在混动商用车巡航控制器上,主要完成升级软件包的接收和安装功能。

设计 OTA 升级系统,使控制器端通过传输控制协议(transmission control protocol,TCP)/网际协议(internet protocol,IP)将要发送的信息打包后,通过 4G 通信模块发送到附近的移动基站,再通过互联网发送到云服务;服务器端解析接收到的报文,并通过互联网将解析结果发送到控制器端,完成与控制器端的通信,实现软件升级[6-7]。混动商用车巡航控制器每次起动时,自动运行 OTA 系统,确认服务器是否有更新任务。

1.2　功能

1)发布升级任务。开发人员将升级软件包上传至服务器端,并在服务器端界面为控制器端指定升级软件包,并选择是否进入命令行调试模块。

2)数字证书交换与验证。由握手模块完成服务器端和控制器端的数字证书交换和验证,实现对称加密密钥的安全传输,并获取控制器端的报文标识符(identification,ID)和软硬件版本号。

3)命令行调试。控制器中加入命令行调试模块,可满足开发人员的远程调试需要,实现调试命令请求,并向服务器端返回调试命令执行结果。

4)软件包下载。由文件下载模块确认控制器端的软件升级版本,实现升级软件包下载,支持文件下载断点续传,下载完成后验证软件包完整性。

5)文件备份与软件升级。在控制器软件升级前,由备份升级模块检测并判断当前车辆状态是否满足升级条件,并进行文件备份;若升级失败,则回滚重刷重新升级[8],确保混动商用车巡航控制器软件成功升级。

2　模块程序设计

基于系统总体方案和功能,设计通信模块、握手模块、命令行调试模块、文件下载模块和备份升级模块,并完成服务器端界面的设计。

2.1　通信模块

通信模块是实现混动商用车巡航控制器 OTA 系统升级的基础,主要功能包括基于 UDP 实现控制器端与服务器端的通信、通过循环冗余校验码(cyclic redundancy check,CRC)校验报文准确性和通过加解密程序保证通信安全性,其中,控制器端与服务器端的通信是设计的重点。

2.1.1　UDP 通信

由于服务器端需通过互联网同时与多个控制器端进行通信,因此选取 Linux 提供的套接字接口作为通信端点,开发套接字初始化函数,设置套接字句柄,完成套接字初始化。选择套接字协议簇为 IPv4,类型为数据报套接字,即使用 UDP 实现套接字传输[9-10],进行网络通信,该通信方式不需要维护连接状态且适合一对多的通信。

在控制器端建立通信目标地址结构体,并在结构体输入服务器端的 IP 地址和端口号[11],完成地址结构体初始化。由于网络延迟时间为 1 ms~1 s,为避免网络波动造成通信失败,设置套接字超时时间为 2 s。套接字和地址结构体初始化完成后,进入发送和接收报文状态。

2.1.2　CRC 校验

网络传输过程中会随机出现位翻转、截断、位缺失等错误,造成控制器软件升级失败,甚至功能失效,需采用报文校验函数检测。由于设计的升级系统中最长报文为 1 410 字节,因此选择 32 位 CRC 进行报文校验。

发送方根据代数编码理论将校验数据编码为原始信息码多项式 $D(x)$,最高次幂为校验数据字节数 m 减 1,生成最高幂次为 m 的固定多项式 $P(x)$;$D(x)$ 乘以 2^m(即左移 m 位),再除以 $P(x)$,所得的商式为

$Q(x)$,余式为 CRC 码多项式 $C_1(x)$,将 $C_1(x)$ 附在 $D(x)$ 之后,即为信息码多项式 $M(x)$,即 $M(x)=2^mD(x)+C_1(x)=Q(x)P(x)$,其中 $M(x)$ 能被 $C_1(x)$ 整除[12]。

接收方将信息码多项式 $M(x)$ 除以 $P(x)$,若余数为 0,则该报文数据通过 CRC 校验,表明报文传输未发生错误;若余数不为 0,则该报文数据未能通过 CRC 校验,表明报文传输发生错误,需重新传输。为了简化程序,本文中设计的升级系统中接收方计算接收报文数据的 CRC 校验码 $C_2(x)$,通过对比 $C_1(x)$ 与 $C_2(x)$ 进行校验。若二者相同,则该报文数据通过 CRC 校验,表明报文传输未发生错误;若二者不同,则该报文数据未能通过 CRC 校验,表明报文传输发生错误。

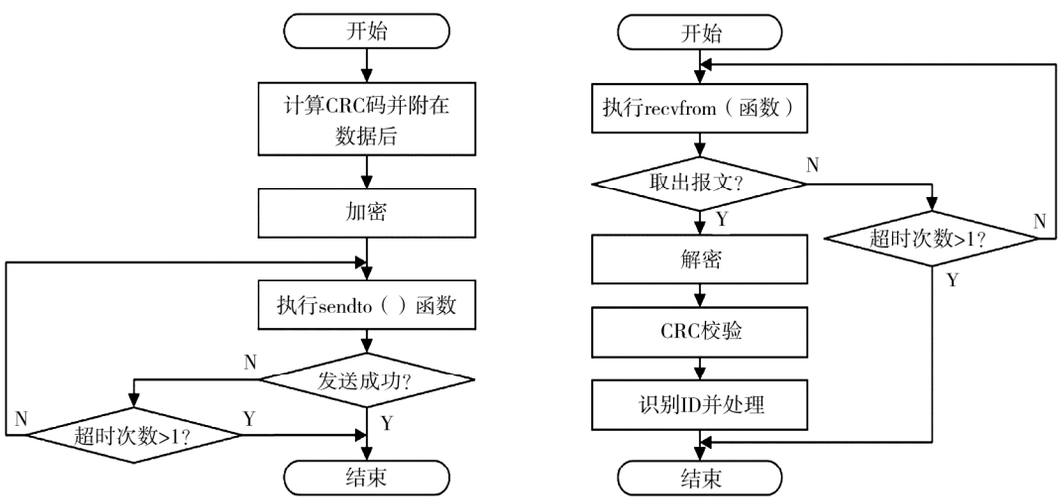
2.1.3 加、解密程序

为保证 OTA 通信报文在互联网上传播过程中的安全性和保密性,调用开放式安全套接层协议(open secure sockets layer,OpenSSL)库,设计加、解密程序使通信模块在网络传输前加密报文,接收后解密报文。

设计加、解密程序时,首先定义加密标志 f_{lag} 的取值,根据 f_{lag} 的值选择对应的加密和解密方式,调用不同的函数和密钥,结合不同的加密算法,完成数据的加密或解密,并返回数据长度。若 f_{lag} 为-1,不对报文内容进行加密。若 f_{lag} 为 0,使用非对称加密算法,非对称加密时,加密程序调用 Linux 内置的 OpenSSL 库内 RSA_public_encrypt( ) 函数,使用接收方的非对称加密密钥 Public_Key 以 117 字节为数据单位加密数据;非对称解密时,解密程序调用 OpenSSL 内的 RSA_private_decrypt( ) 函数,使用解密程序的私钥 Private_Key 以 128 字节为数据单位依次解密数据。若 f_{lag} 为 1,使用对称加密算法,加密或解密时,均调用 OpenSSL 内的 AES_ecb_encrypt( ) 函数,使用对称加密密钥 AES_Key 对数据进行加密或解密,加密时设置模式为 AES_ENCRYPT,解密时设置模式为 AES_DECRYPT。通常,f_{lag} 先置 0,再置 1,这是由于对称加密的密钥相同,只有非对称加密之后,才能传输对称加密的密钥。

2.1.4 发送接收程序

UDP 报文发送与接收过程流程图如图 1 所示。



a)发送过程 b)接收过程

图 1 UDP 报文发送与接收过程流程图

由图 1a)可知:设计 UDP 报文发送程序时,将 $C_1(x)$ 附在 $D(x)$ 后作为完整报文,调用加密程序加密后,执行 Linux 系统内的 sys/socket 库中的 sendto( ) 函数将加密报文写入 4G 通信模块缓冲区,缓冲区按报文写入顺序向移动基站发送报文[13],若 2 s 未发送成功,则重新调用 sendto( ) 函数,若超时 2 次退出程序。

由图 1b)可知:设计 UDP 报文接收程序时,调用 sys/socket 库中的 recvfrom( ) 函数取出 4G 通信模块缓冲区的报文,若 2 s 内未取出报文,则重新调用 recvfrom( ) 函数,超时 2 次退出程序;调用解密程序解密取出的报文内容后,比较 $C_2(x)$ 与 $C_1(x)$,若校验码相同,则校验通过,若不同,则废弃报文;报文校验通

过后,接收函数识别 ID,调用相应的程序,进行数据处理。

2.1.5　通信协议

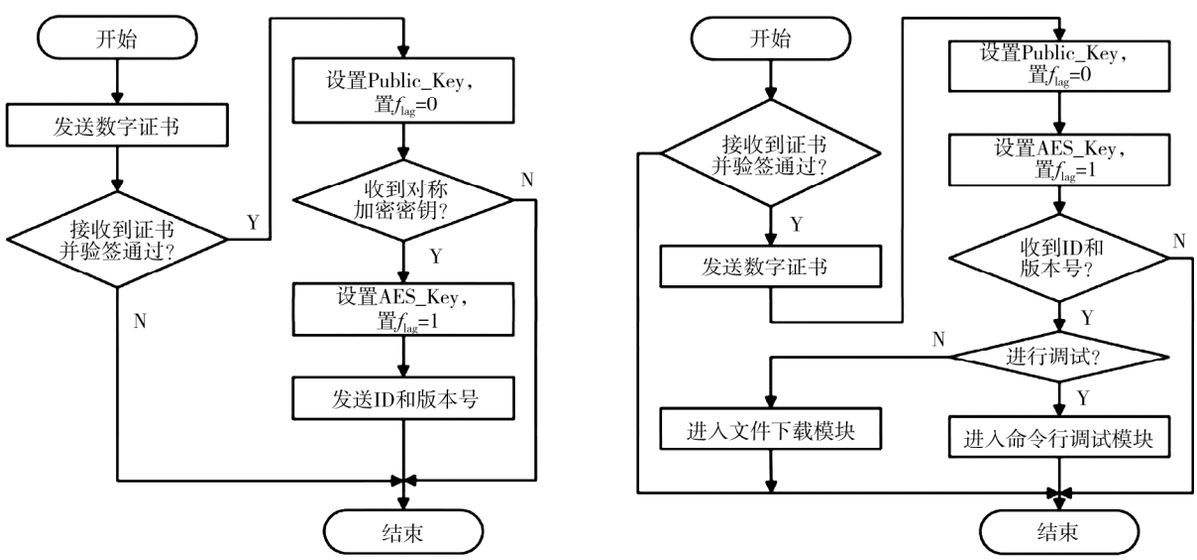
由于 UDP 中报文头不能确认该帧报文的功能,为区分不同报文的功能,本文中设计 OTA 系统时,在报文头部添加报文 ID,用来标识报文的功能,报文 ID 对应的功能如表 1 所示。

表 1　报文 ID 及对应的功能

序号	ID	发送者	功能
1	0x11	服务器端	发送数字证书
2	0x12	控制器端	发送数字证书
3	0x10	服务器端	发送密钥
4	0x50	控制器端	发送控制器端信息
5	0x40	服务器端	重新握手
6	0x28	服务器端	发送调试命令
7	0x68	控制器端	发送调试请求或者调试结果
8	0x34	服务器端	发送软件版本号、大小
9	0x74	控制器端	发送用户同意信息、断点续传信息
10	0x38	服务器端	发送数据
11	0x78	控制器端	发送下载进度
12	0x39	服务器端	发送哈希值

2.2　握手模块

握手模块包括控制器端、服务器端 2 个模块,握手模块流程设计如图 2 所示。



a)控制器端　　b)服务器端

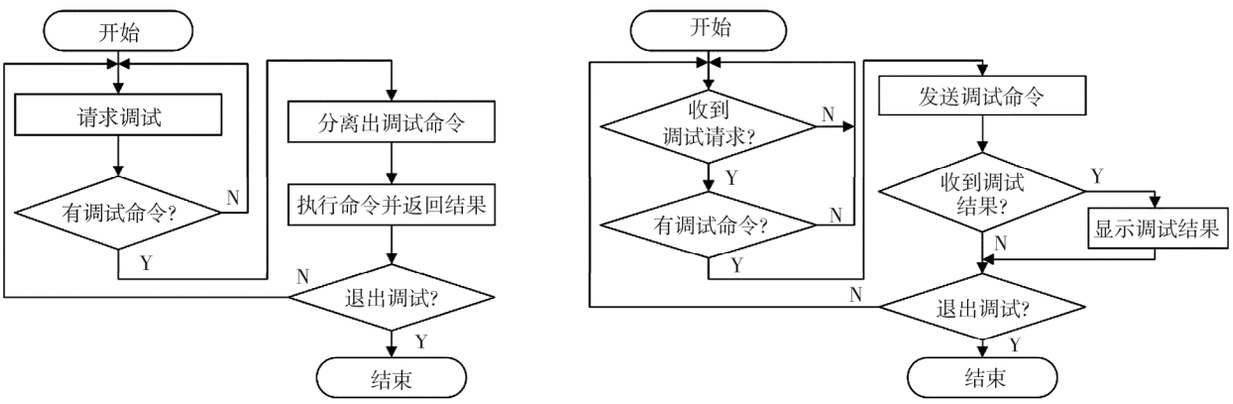
图 2　握手模块流程

由图 2 可知:控制器端将 ID 为 0x12 的报文(内含控制器数字证书),发送给服务器端,服务器端接收到控制器数字证书并验证通过后,发送 ID 为 0x11 的报文;控制器端接收后,调用 OpenSSL 库,验证报文内服务器数字证书的有效期和数字签名;若控制器未接收到 ID 为 0x11 的报文,或报文验证未通过,则退

出 OTA 系统;验证通过后,设置$f_{lag}=0$,解析 Public_Key,通过 ID 为 0x10 的报文(内含加密信息)发送控制器;若控制器接收到 ID 为 0x10 的报文,设置$f_{lag}=1$,解析 AES_Key,同时将 ID 为 0x50 的报文(内含控制器端 ID、软硬件版本号等信息),对称加密后发送至服务器端;若控制器未接收到 ID 为 0x10 的报文,则退出 OTA 系统;服务器端接收 ID 为 0x50 的报文,解密并核对,根据通信之前服务器端的设置,决定进入命令行调试模块或者文件下载模块。若服务器端接收到 ID 为 0x28 的报文,进入命令行调试模块;若接收到 ID 为 0x34 的报文,进入文件下载模块;若未收到 ID 为 0x50 的报文,结束程序。

2.3 命令行调试模块

命令行调试模块无法直接执行命令,需要先新建一个文件流指针 stream,用于存储标准输出流,再调用 C 标准库中的 popen( ) 函数,新建一个管道,用于执行调试命令。由于 popen( ) 函数默认返回标准输出流,为了避免错误输出流,重新定向到标准输出流,将 popen( ) 函数执行结果读入字符串 buf,再关闭文件流指针 stream。命令行调试模块包括控制器端、服务器端 2 个模块,命令行调试模块流程设计如图 3 所示。



a)控制器端 b)服务器端

图 3 命令行调试模块流程

由图 3 可知:控制器端发送 ID 为 0x68 的报文,向服务器端请求调试命令,服务器端先确认是否为调试请求,收到请求后若有调试命令,则向控制器端发送 ID 为 0x28 的报文(内含调试命令);若未收到调试请求或无调试命令,则等待重新接收报文;控制器端解析报文内容,区分命令类型,若是调试命令,则执行该命令,并将执行后的标准输出打包成 ID 为 0x68 的报文,发送到服务器端;若是空命令,则继续向服务器端请求调试命令;服务器端若收到调试结果,则显示结果,否则检测是否退出调试;若不退出调试,则等待接收调试请求;若退出调试,则结束调试程序;控制器重复上述过程,直至接收到服务器发出的 ID 为 0x34 的报文,退出命令行调试模块,进入文件下载模块。

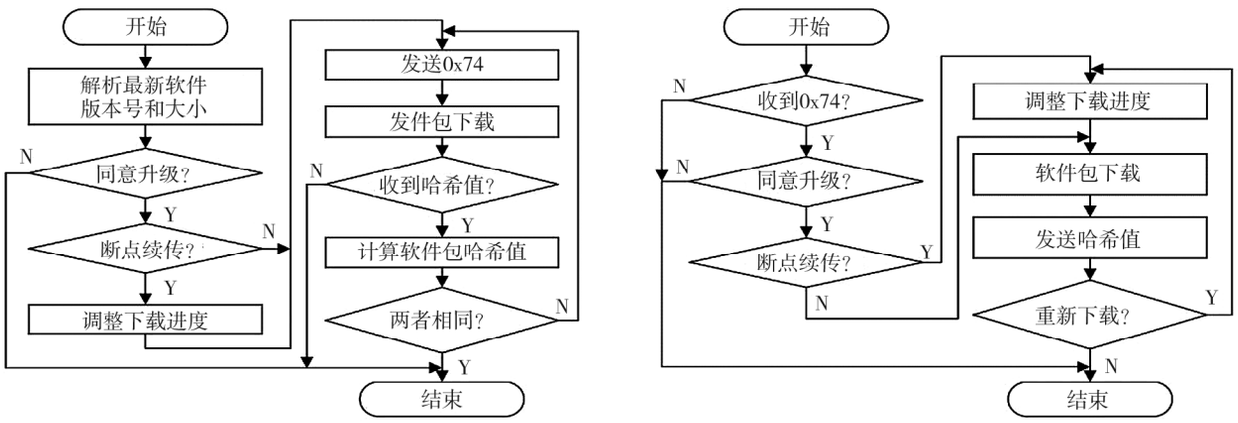
2.4 文件下载模块

设计文件下载模块时,先新建数据缓冲数组$r_{data}[20]$,检测是否需要断点续传,控制器端获取报文发送顺序和进度,并将数据有序储存到$r_{data}[20]$,直至$r_{data}[20]$被写满、剩余报文不足 20 帧或出现超时错误时,检测$r_{data}[20]$是否存在数据缺失。若没有数据缺失,将$r_{data}[20]$写入临时文件,重新计算下载文件大小S_{ize},并将其用 ID 为 0x78 的报文发送到服务器端;若数据缺失,则舍弃该组报文,不重新计算S_{ize};重复接收报文程序,直至下载进度为 100%后,关闭临时文件。

文件下载模块包括控制器端、服务器端 2 个模块,文件下载模块流程设计如图 4 所示。

由图 4 可知:控制器端接收到 ID 为 0x34 的报文并解析,获取最新软件版本号和该软件包大小后,显示升级信息,询问用户是否升级,并监听用户的标准输入流,如果用户输入 Y 或者 y 则视为同意;服务器端接收到 ID 为 0x74 的报文,且同意升级,进入下载程序,否则结束下载程序;控制器判断是否需要断点续传,如需要则先调整下载进度,否则直接下载软件,文件下载完成后,服务器端发送 ID 为 0x39 的报文,传输软件包哈希值,并调用 OpenSSL 库计算软件包哈希值,2 个哈希值若相同,则验证通过,否则服务器

发送 ID 为 0x74 的报文,重新下载。



a)控制器端　　b)服务器端

图 4　文件下载模块流程

若文件下载中断时,先排除中断原因,再检索下载目录是否有与最新软件版本号同名的临时文件;若没有,新建该临时文件并设置 S_{ize} 为 0,重新下载;若存在临时文件,则需要进入断点续传,并设置 S_{ize} 为临时文件大小,将 S_{ize} 打包成 ID 为 0x74 的报文发送给服务器端,服务器端根据报文内容控制下载进度,完成文件下载。

2.5　备份升级模块

备份升级模块流程设计如图 5 所示。如图 5 可知,为避免软件升级失败或引起安全事故,先判断车辆状态是否满足升级条件,不满足条件的调整车辆状态,车辆状态应为发动机转速为 0,车速为 0,驻车制动开启状态,挡位为 P 挡,剩余电量应大于标定最低电量;车辆状态满足升级条件后将之前软件目录内的文件压缩备份,并将压缩包命名为原版软件版本号,解压新版软件包,覆盖原版软件目录;设置 Linux 系统自启动脚本并重启系统,系统重启后运行新版软件包内的 install. sh,运行完后更名为 installed. sh;若软件升级失败,先解压原版软件备份并覆盖原版软件目录,再重启运行原版软件目录内的 installed. sh,提高软件升级过程的安全性和可靠性。

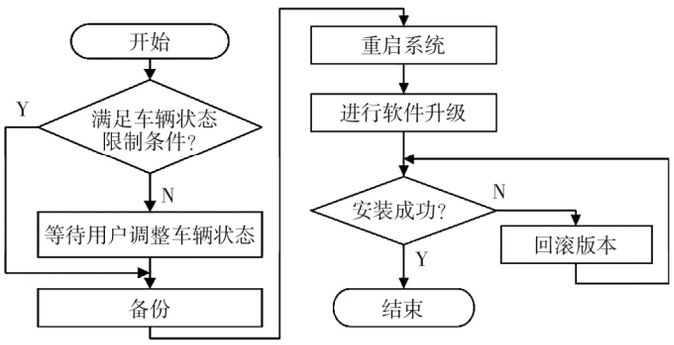


图 5　备份升级模块流程

2.6　服务器端界面设计

通过 Qt Creator,采用多线程机制,设计服务器端界面。主线程负责更新界面信息,子线程负责与控制器端进行信息传输,主线程与子线程之间通过信号槽机制传输信息,保证程序稳定性。定义一个类保存控制器端的通信进度和状态,同时为正在通信的控制器端建立子类,保证不同控制器端的通信互不相扰。服务器端界面设计如图 6 所示。

由图 6 可知:服务器端界面分为控制器端列表区、通信信息状态区和调试控制区;控制器端列表区包括新建控制器端 ID、通信时 IP 地址、端口号、通信状态和下载进度条;通信信息状态区将通信过程中的主要信息显示在新建的文本列表中;调试控制区包含 ID、IP 和端口号文本框,文本框中包含“开启调试、发送命令和结束调试”按键,同时可显示选中的控制器端信息。为指定控制器端更新软件包时,鼠标右键单击控制器端 ID,并单击选择的软件包;控制器端与服务器端建立联系时,控制器端列表区显示通信目标的 IP 和端口号,通信状态为“上线”;鼠标单击控制器端列表区可选中对应控制器端,调试控制区显示被选中的控制器端

信息,点击“开启调试”按键后,控制器端完成通信后,进入命令行调试模块;调试时,先键入调试命令,点击“发送命令”按键发送;结束调试时,点击“结束调试”按键,进入文件下载模块。

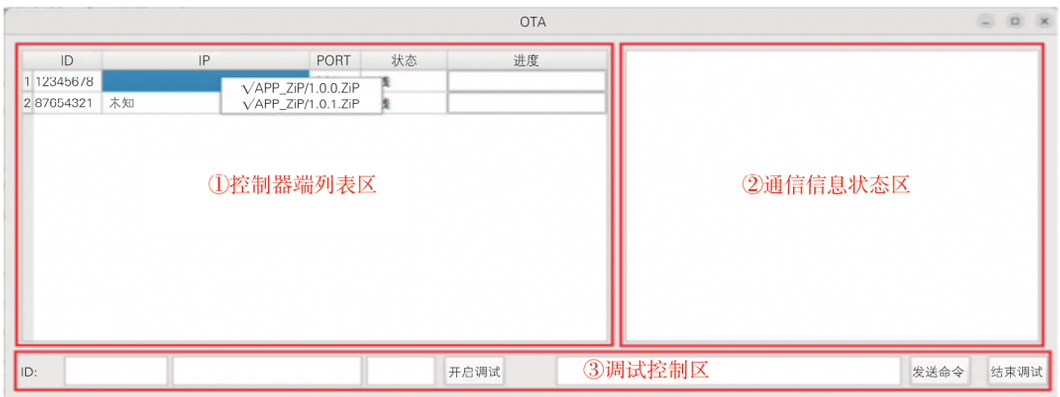


图 6 服务器端界面

3 测试

通过电脑上传测试软件包、发布控制器端升级任务;远程操控公网 IP 的云服务器,实现服务器端部署,并在服务器端界面上实时显示升级过程中的关键信息;控制器端连接显示器,通过 4G 模块与互联网建立通信,实时显示升级进度,混动商用车巡航控制器 OTA 系统测试步骤如下。

1)开启服务器端,设置 ID 为 12345678 的控制器端升级软件包,选择命令行调试,打开控制器端程序,与服务器端建立通信并开始测试。

2)OTA 系统自动进入握手模块,服务器端显示数字证书验证通过、密钥发送成功、成功解密、控制器端版本号 1.0.0 等提示,随后自动进入命令行调试模块。

3)服务器端连续 2 次显示“请求调试命令”的提示后,在服务器端界面键入“./test”调试命令,并点击“发送命令”按钮,显示发送的调试命令和命令执行结果。

4)点击“结束调试”按键,控制器端显示可用更新版本,并询问是否升级,输入“y”后,服务器端显示用户同意,开始下载,进度条显示下载进度;下载过程中强制关闭控制器,并重新打开继续升级,服务器端下载进度条从上次中断处开始,表示实现了断点续传功能;下载完成后,服务器端显示发送哈希值,进行校验。

5)校验完成后,控制器端显示校验通过,提示手刹未拉紧,将手刹拉紧后,显示满足条件;控制器端显示备份完成,重启系统,自动打开命令行窗口,运行 install.sh;运行完毕后,控制器端显示安装成功和测试成功,并要求用户重启系统;重新运行 install.sh,并人为造成测试失败,控制器端提示回滚,解压完备份文件后,运行备份文件中的 installed.sh,确认是否安装成功;若软件安装成功,提示回滚到版本 1.0.0,否则重新运行 installed.sh。

以上测试显示该系统能满足混动商用车巡航控制器的远程升级和调试,实现断点续传和回滚重刷,保证了网络波动时软件包的完整性,解决了控制器调试繁琐的问题。

4 结论

1)混动商用车巡航控制器 OTA 升级,实现了基于 UPD 协议的在线升级,同时对控制器开发阶段的调试工作提供了解决方案。

2)OTA 系统中通信模块完成报文传输,握手模块完成通信加密,命令行调试模块完成命令调试,下载模块完成软件包传输,备份升级模块完成软件的升级和备份。

3)OTA 系统设计时,通过对报文进行加密和 CRC 校验,对下载文件进行哈希校验,确保升级过程中

数据的安全。

参考文献:

[1] 武翔宇,赵德华,郝铁亮. 浅谈汽车 OTA 的现状与未来发展趋势[J]. 汽车实用技术,2019(3):214-216.
[2] 姜楠,姜姗姗,韩小鹏. 汽车在线升级系统(OTA)开发浅析[J]. 时代汽车,2021(21):11-12.
[3] 王栋梁,汤利顺,陈博,等. 智能网联汽车整车 OTA 功能设计研究[J]. 汽车技术,2018(10):29-33.
[4] 李立安,赵帼娟,任广乐. OTA 实现方案及汽车端设计分析[J]. 汽车实用技术,2020(14):16-19.
[5] 陈鹏,徐梅,周传树,等. 客车 OTA 实施及其整车 CAN 通信设计[J]. 汽车电器,2022(3):4-6.
[6] 朱青松,李军伟,王进,等. 基于嵌入式 Linux 的智能重型拖拉机远程监控系统开发[J]. 内燃机与动力装置,2021,38(1):15-20.
[7] 谭文阳,李军伟,朱青松. 重型拖拉机控制器的引导加载程序与上位机设计[J]. 内燃机与动力装置,2020,37(6):57-62.
[8] 陈睿智,石春,吴刚,等. 面向 OTA 需求的汽车电控单元 Bootloader 设计[J]. 仪表技术,2021(2):8-12.
[9] 王进文. Socket 通信过程原理及相关系统调用[J]. 中国新技术新产品,2015(7):25.
[10] 王伟,蓝雯飞,高伟华. 用 Socket 实现 UDP 协议下的网络通信[J]. 软件导刊,2009,8(9):115-117.
[11] 佚名. 简析 IP、UDP 和 TCP 三种协议的关系[J]. 电脑知识与技术(经验技巧),2020(4):90-91.
[12] 王忠,李延社,游智胜. CRC 算法设计与程序实现[J]. 电子测量技术,2007(12):26-28.
[13] 邢卫国,赵亚松. 一种提高传输可靠性的简捷方法[J]. 计算机与网络,2014,40(22):58-60.

Development of OTA system for hybrid commercial vehicle cruise controller based on UDP

WANG Zhiying[1], LI Junwei[1*], LI Xingkun[2],
LI Lianqiang[3], XU Jinpeng[1], LIU Baoyan[1]

1. School of Transportation and Vehicle Engineering, Shandong University of Technology, Zibo 255049, China;
2. Beijing Yujun Automobile Research Institute, Beijing 100016, China;
3. FAW Jiefang Qingdao Automotive Co., Ltd., Qingdao 266217, China

Abstract: To achieve timely updates and remote debugging for hybrid commercial vehicles cruise controllers, and to ensure the safety of the system , an over the air technology (OTA) system based on the user datagram protocol (UDP) is designed and developed, which it includes the communication module, handshake module, command line debugging module, file download module, backup and upgrade module, and server interface of the OTA system. Testing and verification results show that the hybrid commercial vehicle cruise controller OTA system based on UDP protocol can accomplish the functions of message transmission, communication encryption, system debugging, software package transmission, software upgrade and backup, achieving the security, reliability, and convenience of software upgrade.

Keywords: commercial vehicle; cruise controller; UDP protocol; OTA; encryption; decryption

(责任编辑:臧发业)