1040607 OTA车端集成说明

修订记录

版本号	更新时间	修改人	更新内容简述
V1.0.0	2024-7-9	王凯	内容新增
V2.0.0	2024-7-23	王凯	完善第五章系统依赖及第六章集成方案

1 文档介绍

本文档旨在提供智能网联汽车整车OTA升级系统的功能规范、接口定义、验证、集成和部署等详细信息。根据客户需求，为他们提供相关的整车OTA升级技术解决方案。针对FOTA项目的要求，编写一份系统集成设计方案文档，明确各方责任分工。

1.1 目的

- 明确一汽大众OTA系统对IVI供应商的集成要求；
- 明确一汽大众与IVI供应商职责和分工；
- 描述一汽大众OTA系统集成环境，指导车机供应商如何集成部署OTA系统。

1.2 范围

本规范适用一汽大众FOTA项目各车型。

1.3 术语与缩写

缩写(Abbreviation)	**英文全称(Full Name)**	**中文描述（Chinese description）**
OTA	Over-The-Air Technology	远程无线升级技术
FOTA	Firmware Over-The-Air	固件空中升级技术
DOTA	Diagnostic Over-The-Air	云诊断
SOTA	Software Over-The-Air	软件升级
ECU	Electronic Control Unit	电子控制器单元
TSP	Telematics Service Provider	汽车远程通信服务供应商（智能互联平台）
HMI	Human Machine Interface	人机交互接口，主要负责人机交互，HMI界面展示，为UCM模块提供OTA相关的车机系统能力，如车辆预约升级，采集车辆零件信息等
FBL	Flash Bootloader	闪存引导加载程序
CDN	Content Delivery Network	内容分发网络服务器系统模块，用于管理升级包
OTA Server	Over The Air Server	OTA管理平台，负责车辆的版本管理，任务发布，升级结果展示等
MCU	Micro Controller Unit	微控制单元，作为单独升级模块
APK	Android Application Package	OTA 人机交互Android应用程序包
OEM	Original Equipment Manufacturer	原始设备制造商
UCM	Update Control Master	OTA控制主模块，用来负责OTA升级主流程控制管理
Lite	Update Control Lite	OTA 控制从模块，用来负责提供OTA升级所需的关键能力支持（如文件下载，车辆状态信息获取等）。
UDS TP	Unified Diagnostic Services Transport Protocol	一个用于汽车诊断通信的协议层
PKI	Public Key Infrastructure	一种用于管理数字证书和公开密钥加密的系统

VDI	vehicle data interface	车辆数据接口，负责功能如下: 1. 诊断脚本解析。 2. UDS 诊断报文组装和发送。
UA	Update Agent	一汽大众差分升级模块，负责车机SOC的升级。

1.4 参考文件

- 《一汽大众整车OTA咨询项目_车端_产品需求规格说明书》
- 《一汽大众整车OTA咨询项目_Lite_软件详细设计文档》

1.5 阅读对象

- 一汽大众OTA开发工程师
- 车机系统开发工程师
- 车机系统集成工程师
- TBOX或网关开发工程师
- TBOX或网关集成工程师

1.6 偏差

如果车机系统或TBOX等未能符合本规范的要求，任何偏差应首先获得OEM和一汽大众工程师的认可，达成一致后更新本技术要求。

2 OTA系统概述

2.1 车端系统架构

图 2-1 车端系统架构图

系统中各个模块功能描述如下:

名称	描述	实现方
UCM	OTA主控程序主要负责控制业务流程;车云交互，处理外部刺激，收集零件信息;解析和控制下载策略和安装策略	艾拉比
UCL	主要负责响应UCM控制指令，负责收集该域下的零件交互完成零件信息收集，升级包下载，升级能力调用等业务完成该域下零件OTA业务	艾拉比
HMI	负责人机交互接受用户指令(检测，下载，立即安装，预约安装等)，显示OTA业务流程	艾拉比
VDM	实现车辆总线数据收发、UDS(ISO14229)诊断数据收发、UDS刷写数据收发，定制化脚本解析及执行的功能模块	艾拉比
DoIP协议栈	Diagnostic Communication Over Internet Protocol基于以太网协议的诊断通信	艾拉比
DTM	DTM委托传输服务(Delegate Transfer Manager)是服务组件中的一个关键模块，旨在提供高效可靠的传输能力(车内文件下载、车云文件下载和车内文件上传等)	艾拉比

UA	差分升级模块，负责车机soc的升级	艾拉比
Integeration	域内更新代理该项目负责管理了IC，DMS，TBOX等下挂件升级和控制了车机升级（UA：刷写非当前区域+Active：激活）	艾拉比
D2B	设备提供的服务功能(Device to Business)，OTA系统中有些业务需要UCM宿主ECU设备提供接口功能， OTA系统把这类接口功能归类到D2B服务模块，由D2B服务模块统一给OTA业务提供服务	艾拉比
SM	安全模块，与KPI对接完成文件解密验签等安全相关功能	艾拉比

2.2 业务时序图

// TODO: 待补充

3 功能模块描述

3.1 车辆标识信息获取

3.1.1 功能描述

获取车辆唯一标识信息，如车辆VIN码。用于 OTA管理平台判断车辆的合法性。

3.1.2 业务时序

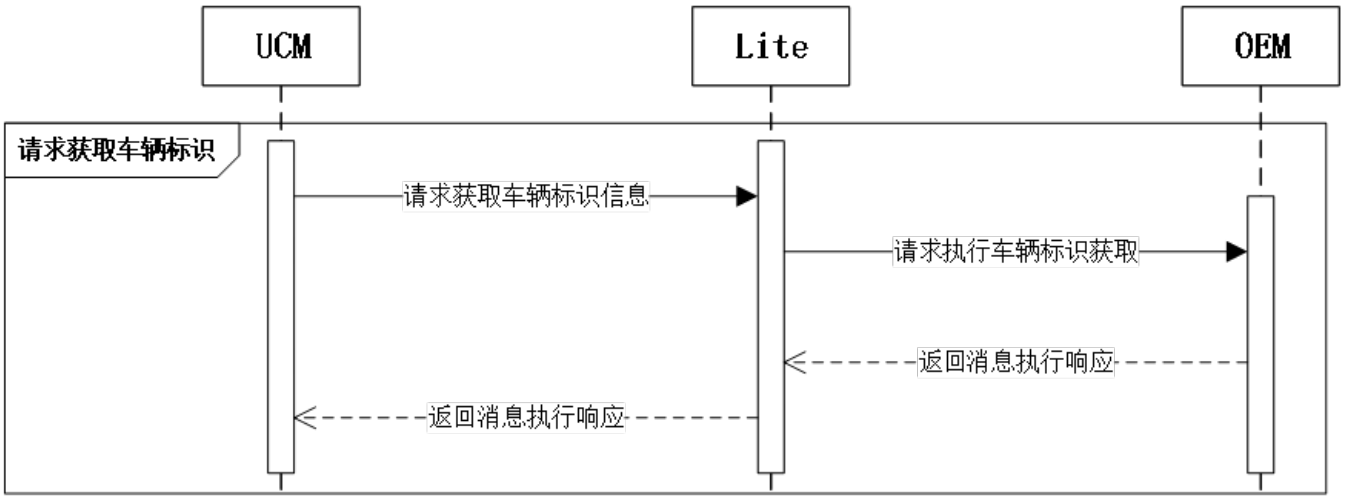


图 3-1 车辆标识信息获取时序图

3.1.3 接口设计

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**在OTA过程中，UC-Master需要获取车辆标识信息VIN，与OTA云平台进行交互时需要携带
- **接口原型**

```
/**
 * @brief 获取车辆标识信息
 *
 * @param[in] pBuffer     描述: 标识信息的存储空间 \n
 * 范围值: 数组地址 \n
 * 单位: DCHAR [] \n
 *
 * @param[in] uiInBuffSize     描述: pBuffer的数组长度，sizeof(pBuffer) \n
 * 范围值: >=0 \n
 * 单位: Byte \n
 *
 * @param[out] pOutBufferSize     描述: 获取到的标识信息的长度 \n
 * 范围值: >=0 \n
 * 单位: Byte \n
 *
 * @return D32S 获取车辆唯一标识码信息的结果
 */
extern D32S oem_getVehicleIden(DCHAR* pBuffer[], const D32U uiInBuffSize, D32U* pOutBufferSize);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pBuffer	DCHAR*	input	non-nullptr	待获取的车辆唯一标识的存储地址
uiInBuffSize	const D32U	input	>0	待获取的车辆唯一标识的长度
pOutBufferSize	D32U*	output	>0	获取的车辆唯一标识的长度
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.2 车辆大版本信息读/写

3.2.1 功能描述

用于获取/设置车辆大版本信息。

大众需要梳理对整车大版本号的管理及产线的初始烧录问题@宋吉

3.2.2 业务时序

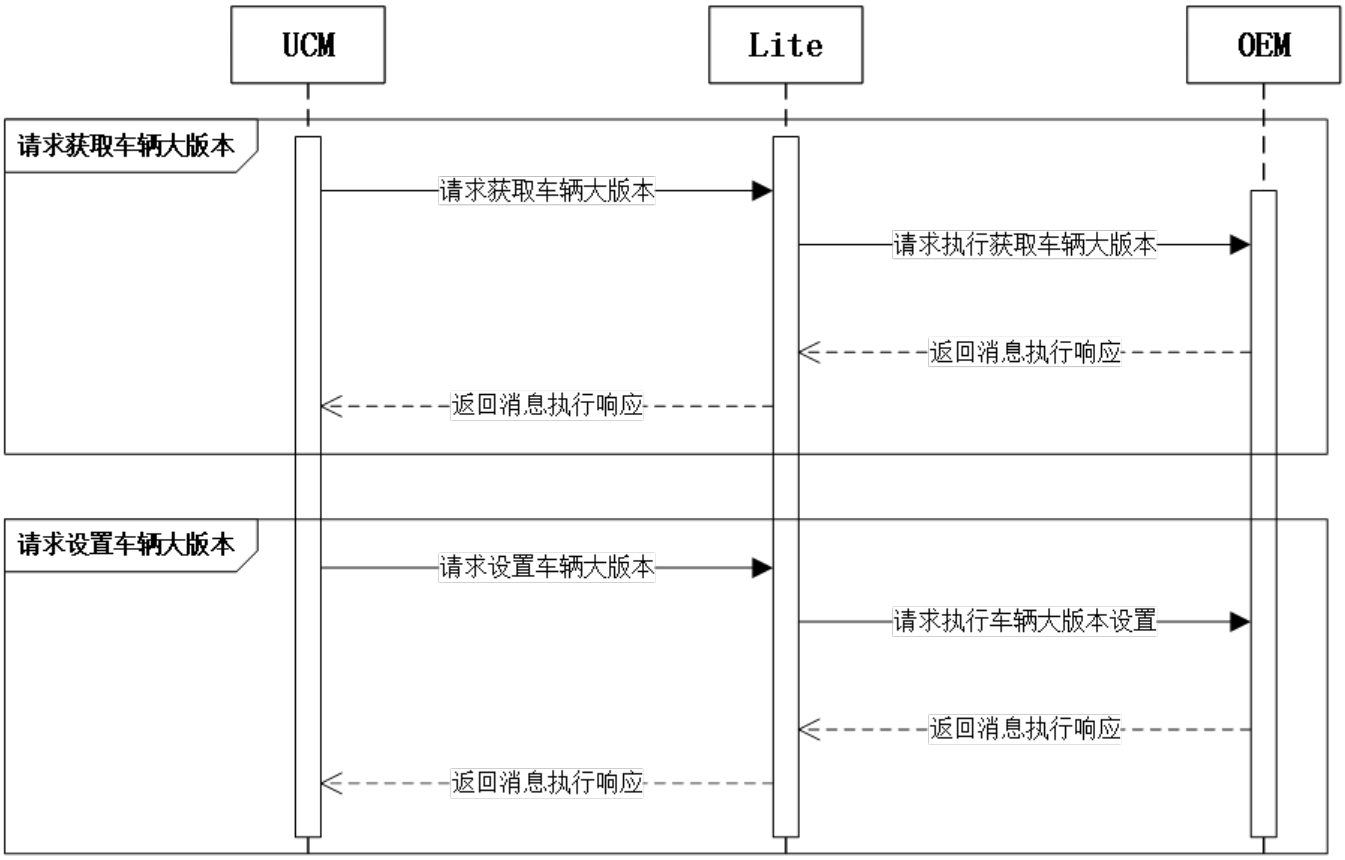


图 3-2 车辆大版本信息时序图

3.2.3 接口设计

3.2.3.1 读车辆大版本信息接口

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**在OTA检测版本、升级前版本校验等业务中，OTA需要调用该接口读取系统大版本号操作
- **接口原型**

```
/**
 * @brief 获取车辆大版本信息
 *
 * @param[in] pBuffer    描述: 大版本信息的存储空间 \n
 * 范围值: 数组地址 \n
 * 单位: DCHAR [] \n
 *
 * @param[in] uiInBuffSize    描述: pBuffer的数组长度，sizeof(pBuffer) \n
 * 范围值: >=0 \n
 * 单位: Byte \n
 *
 * @param[out] pOutBufferSize    描述: 获取到的大版本信息的长度 \n
 * 范围值: >=0 \n
 * 单位: Byte \n
 *
 * @return D32S 获取车辆大版本信息的结果
 */
extern D32S oem_getVehicleBigVersion(DCHAR* pBuffer[], const D32U uiInBuffSize, D32U* pOutBufferSize);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pBuffer	DCHAR*	input	non-nullptr	待获取的车辆大版本的存储地址
uiInBuffSize	const D32U	input	>0	待获取的车辆大版本信息的长度
pOutBufferSize	D32U*	output	>0	获取的车辆大版本信息的长度
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.2.3.2 写车辆大版本信息接口

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**在OTA升级完成后，系统需要更新大版本号，OTA需要调用该接口进行版本写入操作
- **接口原型**

```
/**
 * @brief 设置车辆大版本信息
 *
 * @param[in] pBigVersion     描述: 车辆大版本信息 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @param[in] uiBigVersionLen     描述: 车辆大版本信息的长度 \n
 * 范围值: >=0 \n
 * 单位: Byte \n
 *
 * @return D32S 设置车辆大版本信息的结果
 */
extern D32S oem_setVehicleBigVersion(const DCHAR* pBigVersion, const D32U uiBigVersionLen);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pBigVersion	const DCHAR*	input	non-nullptr	待设置的车辆大版本的信息
uiBigVersionLen	const D32U	input	>0	待设置的车辆大版本信息的长度
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.3 预约定时器设置/查询/预约时间到达通知

3.3.1 功能描述

车机上OTA主控UC-Master接收到用户触发的预约升级指令，UC-Master通过Lite请求TBOX设置预约定时器时间。定时器时间到达后，唤醒整车。Lite-APP检测到预约时间到达后，通知UC-Master可以执行预约升级。

3.3.2 业务时序

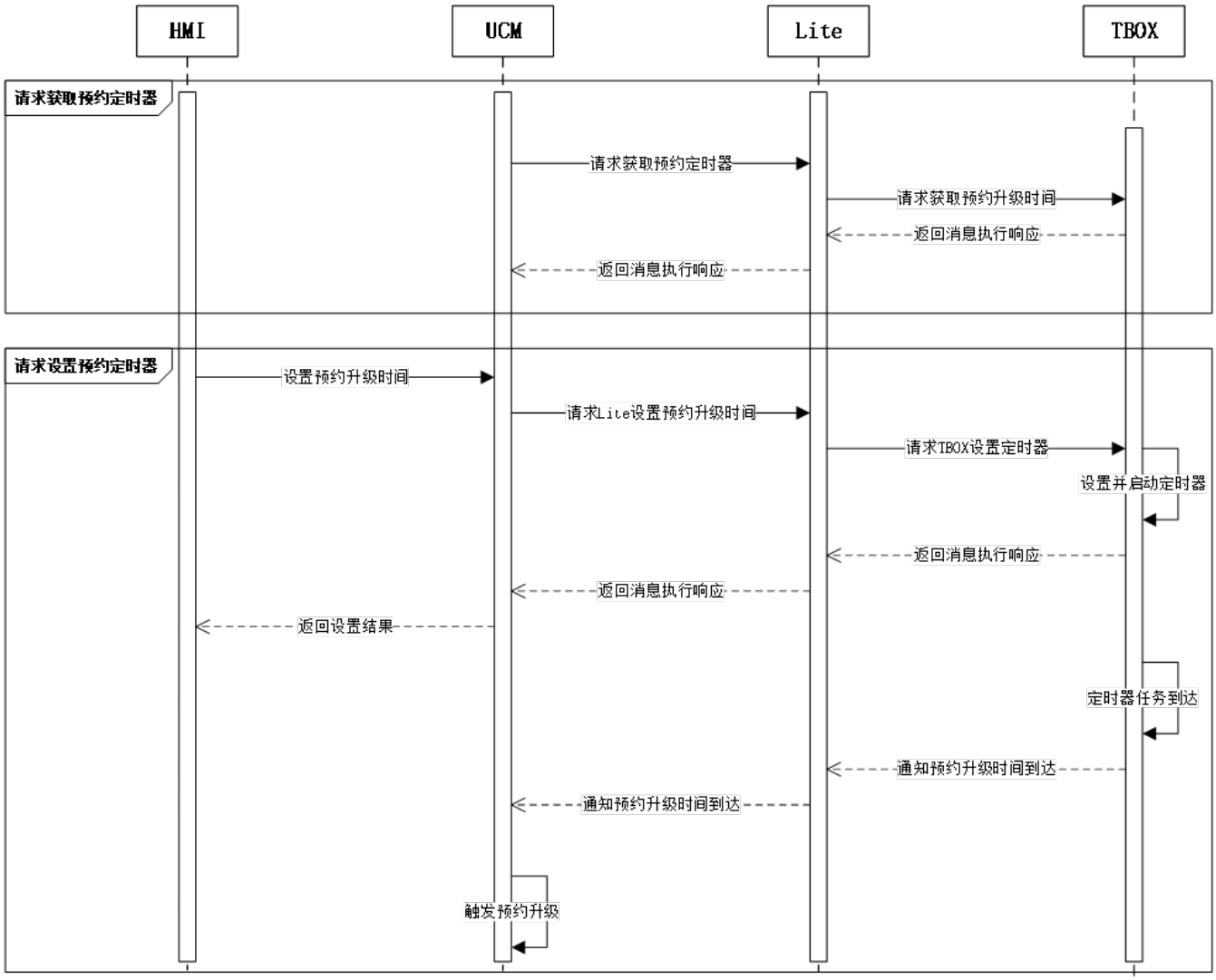


图 3-3 预约升级时序图

3.3.3 接口设计

3.3.3.1 注册预约升级时间到达监听

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**当OTA设置的预约升级时间到达后，调用此接口通知ABUP，预约升级时间到达，需要执行预约升级
- **接口原型**

```
/**
 * @brief 注册预约升级时间到达回调接口
 *
 * @param[in] pfnAppointNotify     描述: 预约时间到达通知处理 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @param[in] pPrivate     描述: 私有参数 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 预约时间到达后通知处理结果
 */
extern D32S oem_registerEventNotifyCallback(ADM_OEM_D2B_APPOINTMENT_ARRIVAL_NOTIFY pfnAppointNotify, DVOID* pPrivate);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pfnAppointNotify	ADM_OEM_D2B_APPOINTMENT_ARRIVAL_NOTIFY	input	non-nullptr	预约到达处理回调
pPrivate	DVOID*	input	non-nullptr	私有参数
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.3.3.2 预约定时器查询接口

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**为了确保OTA升级过程的顺利进行，同时考虑到用户的便利性和车辆的正常使用，提供查询预约安装时间的查询接口
- **接口原型**

```
/**
 * @brief 查询预约定时器信息
 *
 * @param[in] eType     描述: 预约类型 \n
 * 范围值: ADM_D2B_APPOINTMENT_TYPE_E \n
 * 单位: N/A \n
 *
 * @param[out] pAppointmentS     描述: 查询预约时间（相对时间，相对于设置时的时刻到目标时间的秒数。单位：s） \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 查询预约定时器信息的结果
 */
extern D32S oem_getAppointment(const ADM_D2B_APPOINTMENT_TYPE_E eType, const D64U* pAppointmentS);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
eType	const ADM_D2B_APPOINTMENT_TYPE_E	input	枚举	预约类型
pAppointmentS	const D64U*	output	non-nullptr	查询设置的预约定时器定时时间值（相对时间值，返回当前时刻，到设置的目标时间的秒数。单位：秒。若设置的定时时间已过，则返回查询的时间为零）
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.3.3.3 预约定时器设置接口

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**用户可以根据自己的日程设置升级时间，减少对用户日常形成的干扰
- **接口原型**

```
/**
 * @brief 设置预约定时器，预约时间到达后，设备唤醒并通知调用者
 *
 * @param[in] eType     描述: 预约类型 \n
 * 范围值: ADM_D2B_APPOINTMENT_TYPE_E \n
 * 单位: N/A \n
 *
 * @param[in] uAppointmentS     描述: 预约定时器时间（相对时间值，相对于当前时刻到目标时间的秒数。单位：s） \n
 * 范围值: >0 \n
 * 单位: Second \n
 *
 * @return D32S 设置预约定时器的执行结果
 */
extern D32S oem_setAppointment(const ADM_D2B_APPOINTMENT_TYPE_E eType, const D64U uAppointmentS);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
eType	const ADM_D2B_APPOINTMENT_TYPE_E	input	枚举	预约类型
uAppointmentS	const D64U	input	>0	设置的预约定时器时间值（相对时间值，相对于当前时刻，到目标时间的秒数。单位：秒），一旦设置，系统就开始计时，直到到达设置的秒数后，系统需要唤醒整车并通知预约时间到达
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.4 OTA模式进入/退出

3.4.1 功能描述

执行OTA升级时，告知Host当前已经在升级模式中，需要Host释放相应的资源给到OTA，即：非重要的工作需要停止。

3.4.2 业务时序

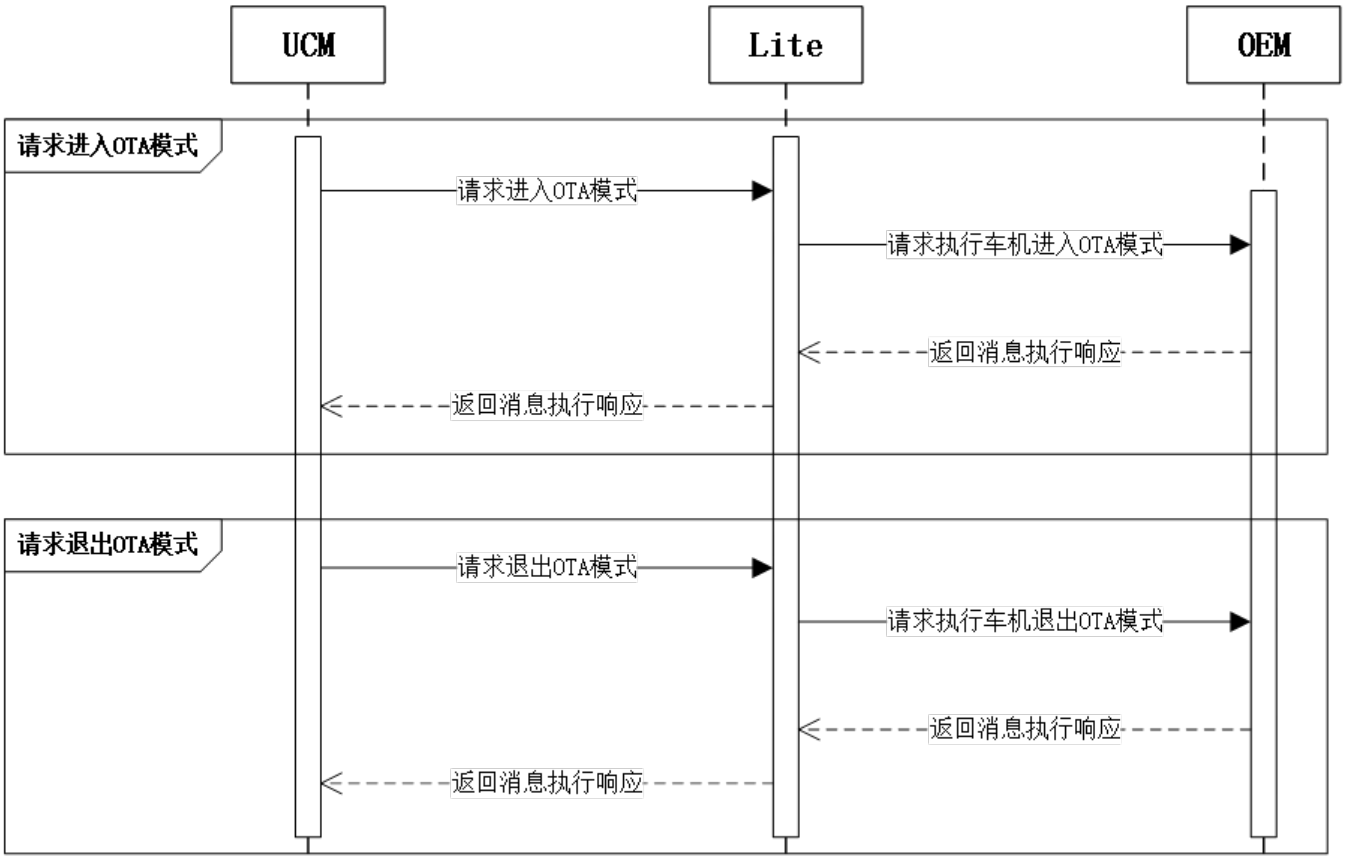


图 3-4 OTA模式时序图

3.4.3 接口设计

3.4.3.1 OTA模式设置接口

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**当OTA进入安装过程整车需要进入OTA模式，OTA安装结束后需要退出OTA模式
- **接口原型**

```
/**
 * @brief OTA模式进入/退出
 *
 * @param[in] uiType     描述: 设置类型 \n
 * 范围值: >=0 \n
 * 单位: Second \n
 *
 * @param[in] uiStayTime     描述: 当进入OTA模式时该值表示维持时间，当退出OTA模式时该值无意义 \n
 * 范围值: >=0 \n
 * 单位: Second \n
 *
 * @return D32S 设置结果
 */
extern D32S oem_setOTAMode(D32U uiType, const D64U uiStayTime);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
uiType	D32U	input	OTA模式类型标识, 取值范围[1,2] 1：进入OTA模式 2：退出OTA模式	取值范围为：[1,2]，其他均为非法值。
uiStayTime	const D64U	input	>=0	1、进入OTA模式的维持时间值（相对时间，相对于当前时刻，到目标时间的描述。单位：秒） 2、当为退出OTA模式时，该值无意义，默认填0即可。
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.4.3.2 OTA模式查询接口

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**在安装前置条件准备中，首先查询整车是处于OTA模式中，如果不在OTA模式中或者进入OTA模式的剩余维持时间不够，那么需要根据安装所需的时间重新设置整车进入OTA模式
- **接口原型**

```
/**
 * @brief 查询当前车辆是否处于OTA模式
 *
 * @param[out] pType     描述: 查询的状态 \n
 * 范围值: 指针 \n
 * 单位: N/A\n
 *
 * @param[out] uiStayTime     描述: 剩余维持时间 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 查询操作结果
 */
extern D32S oem_queryOTAMode(D32U* pType, D64U* pStayTime);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pType	D32U*	output	OTA模式类型标识, 取值范围[1, 4] 1：进入OTA模式中 2：OTA模式维持中 3：退出OTA模式中 4：不在OTA模式下	取值范围为：[1,4]，其他均为非法值。
pStayTime	D64U*	output	>=0	1、当处于进入OTA模式中或OTA模式维持中时，该时间表示剩余维持OTA模式的时间值（相对时间，相对于当前时刻，到设置的目标维持时间的秒数，单位：秒）。 2、当处于退出OTA模式中或不在OTA模式下时该时间没有意义，忽略。
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.5 Host ECU状态维持唤醒/取消维持唤醒通知

3.5.1 功能描述

在远程控制升级或预约升级时，需要维持Host设备在唤醒状态，不进入休眠，防止OTA升级过程被中断。升级结束时取消持续唤醒的状态。

（需要指定保持Host唤醒状态的最大持续时间）

3.5.2 业务时序

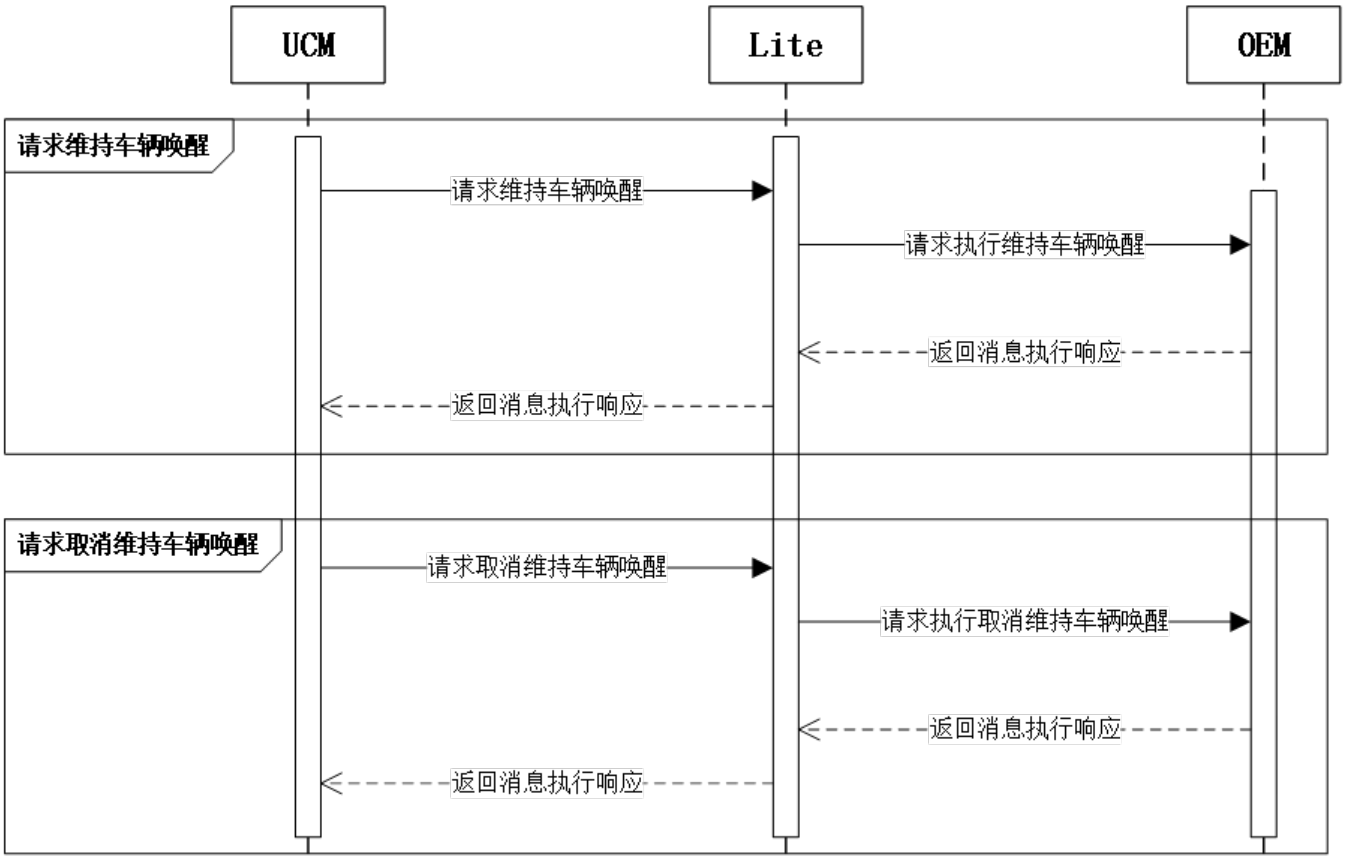


图 3-5 唤醒状态时序图

3.5.3 接口设计

3.5.3.1 Host ECU状态维持唤醒

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**当OTA进入安装过程，需要保持HOST ECU处于维持唤醒状态，避免HOST ECU休眠而影响整车升级
- **接口原型**

```
/**
 * @brief 请求告知设备需要维持Host ECU处于唤醒
 *
 * @param[in] uiStayTimeS     描述: 维持唤醒的最大超时时间 \n
 * 范围值: >0 \n
 * 单位: Second \n
 *
 * @return D32S 硬件收到请求且维持唤醒的结果
 */
extern D32S oem_stayVehicleWakeup(const D64U uiStayTimeS);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
uiStayTimeS	const D64U	input	>0	维持唤醒的最大超时时间（相对时间值，相对于设置时的时刻，到目标时间的秒数。单位：秒）
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.5.3.2 Host ECU状态取消维持唤醒通知

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**OTA安装结束后，通知HOST ECU OTA升级不需要维持唤醒状态了
- **接口原型**

```
/**
 * @brief 请求告知设备结束Host ECU 维持唤醒，进入休眠状态
 *
 * @return D32S 硬件收到请求且结束维持唤醒的结果
 */
extern D32S oem_stopVehicleWakeup(DVOID);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.5.3.3 Host ECU状态维持唤醒查询接口

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**在安装前置条件准备中，首先查询宿主ECU维持唤醒的状态，如果不在维持唤醒状态中或者剩余维持唤醒的时间不够，那么需要根据安装所需的时间重新设置宿主ECU维持唤醒状态
- **接口原型**

```
/**
 * @brief 查询Host ECU维持唤醒信息
 *
 * @param[in] pStayTimeS     描述: 维持唤醒的最大超时时间 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 查询整车唤醒操作的结果
 */
extern D32S oem_queryVehicleWakeup(D32U* pState, D64U* pStayTimeS);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注

pState	D32U*	output	non-nullptr	查询是否在维持唤醒状态，取值范围：[1,2] 1：表示已设置维持唤醒状态 2：表示未在维持唤醒状态
pStayTimeS	D64U*	output	non-nullptr	若已在维持唤醒状态则结果表示剩余维持唤醒的时间，若未在维持唤醒状态则忽略该结果（相对时间值，相对于设置时的时刻，到目标时间的秒数。单位：秒）
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.6 校验host的存储空间

3.6.1 功能描述

检查host设备剩余空间是否满足OTA需要。

3.6.2 业务时序

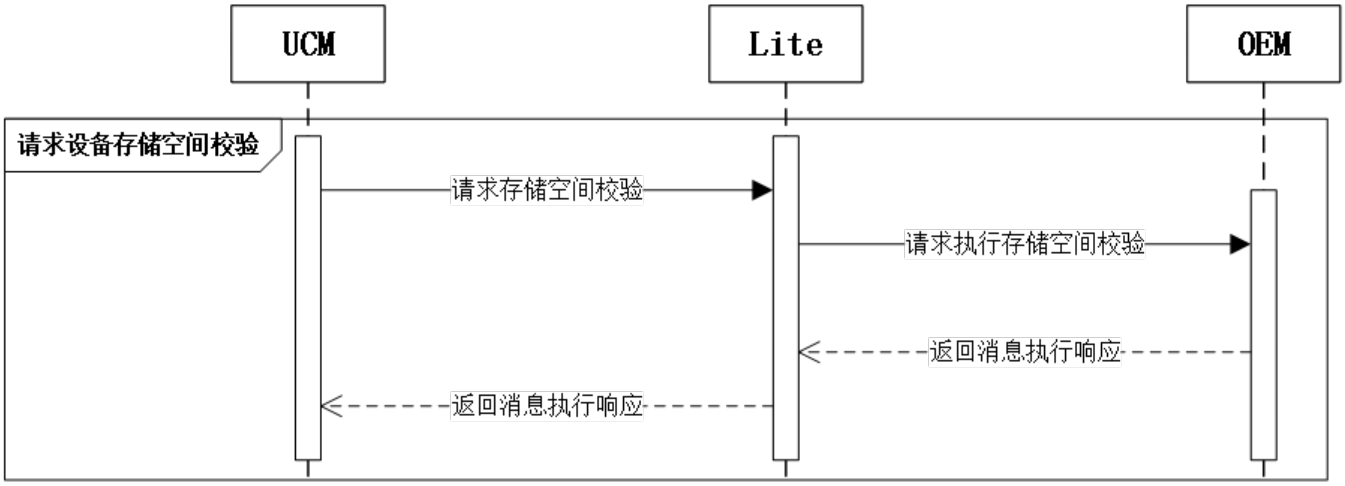


图 3-6 存储空间校验时序图

3.6.3 接口设计

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**OTA下载、安装前需要校验设备存储空间是否满足
- **接口原型**

```
/**
 * @brief 请求设备校验存储空间是否满足
 *
 * @param[in] uiNeedSize     描述: 需要的存储空间大小 \n
 * 范围值: >=0 \n
 * 单位: Byte \n
 *
 * @return D32S 请求校验存储空间是否满足操作的结果
 */
extern D32S oem_verifyStorageSpace(const D64U uiNeedSize);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
uiNeedSize	const D64U	input	>=0	需要的存储空间大小，单位Byte
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.7 获取4G开关状态

3.7.1 功能描述

获取当前Host设备的4G开关状态，用于判断当前设备网络是否可用。

3.7.2 业务时序

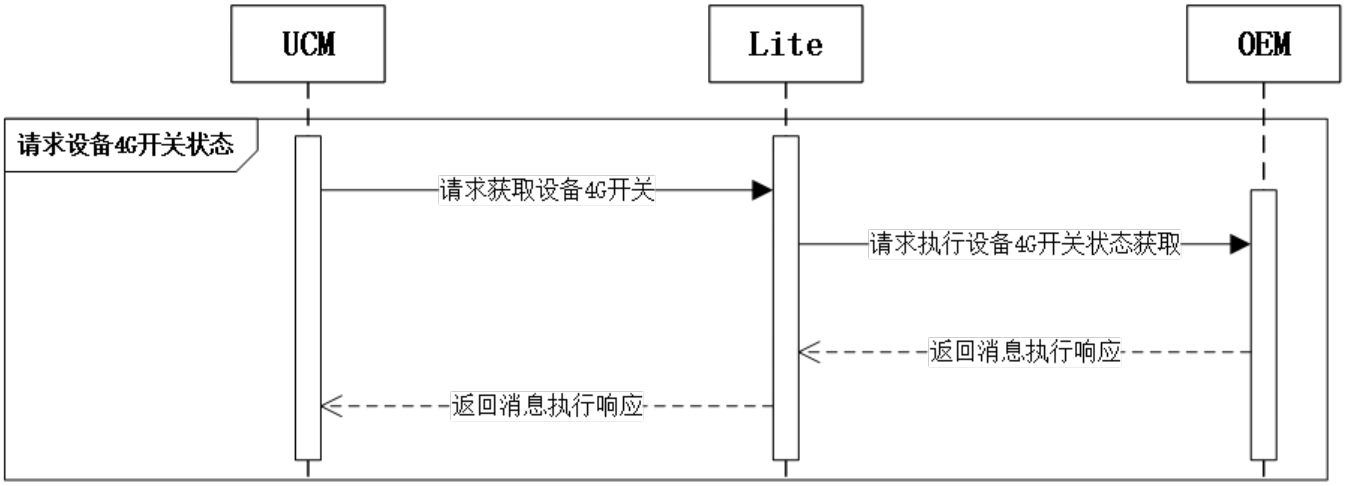


图 3-7 4G开关状态获取时序图

3.7.3 接口设计

- **接口说明**
  - **调用方**：ABUP
  - **实现方**：车机供应商
  - **同步/异步**：同步接口，接口调用的超时时间为30秒
  - **业务场景**：OTA下载前获取当前的4G网络的开关状态
- **接口原型**

```
/**
 * @brief 请求获取设备的4G开关状态
 *
 * @param[out] pState     描述: 获取的4G开关状态 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 获取设备4G开关状态的结果
 */
extern D32S oem_getNetworkState(AMD_NETWORK_STATE_E* pState);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pState	AMD_NETWORK_STATE_E*	output	non-nullptr	待获取的状态存储地址
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.8 TSP消息推送

3.8.1 功能描述

云端通过TSP发送消息信息到车端，车端TBox收到的TSP消息后通过该接口通知到OTA 主控。

3.8.2 业务时序

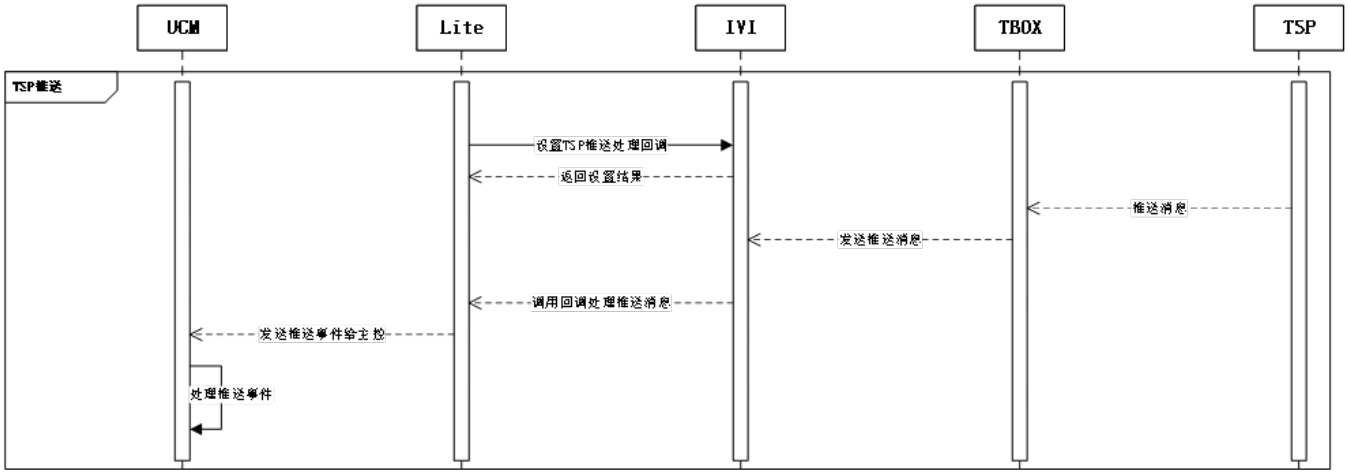


图 3-8 TSP消息推送时序图

3.8.3 接口设计

3.8.3.1 主控TSP推送回调

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **业务场景：**云端通过TSP发送任务信息到车端，车端将收到的任务信息转给车机供应商

注：暂定消息为字符串，具体待TSP平台确认消息协议及触发消息指令（检测、下载、安装？）

- **接口原型**

```
/**
 * @brief 推送处理
 *
 */
typedef D32S (*ADM_OEM_TSP_PUSH_CB_F)(const DCHAR* pMessage, const D32U uiMessageLength);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pMessage	const DCHAR*	input	non-nullptr	推送消息暂定字符串，待TSP平确认消息解析协议
uiMessageLength	const D32U	input	>=0	推送消息长度，单位：Byte
返回值	D32S	output	int	0: 成功 1: 解析错误 2: 解析成功，但TSP任务触发失败 暂提供以上错误码，其他错误码预留

3.8.3.2 主控TSP推送回调注册

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**云端通过TSP发送任务信息到车端，主控将收到的任务信息数据转给车机供应商
- **接口原型**

```
/**
 * @brief 注册TSP推动消息的回调函数
 *
 * @param[in] tspPushCb     描述: TSP消息推送接口 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 注册TSP推送消息的回调结果
 */
extern D32S oem_registerTspPushCallback(const ADM_OEM_TSP_PUSH_CB_F tspPushCb);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
tspPushCb	ADM_OEM_TSP_PUSH_CB_F	input	non-nullptr	TSP消息推送接口
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.9 零件信息采集

3.9.1 功能描述

用于采集待升级的ECU的DID信息，如ECU软件版本号等。（UDS）

3.9.2 业务时序

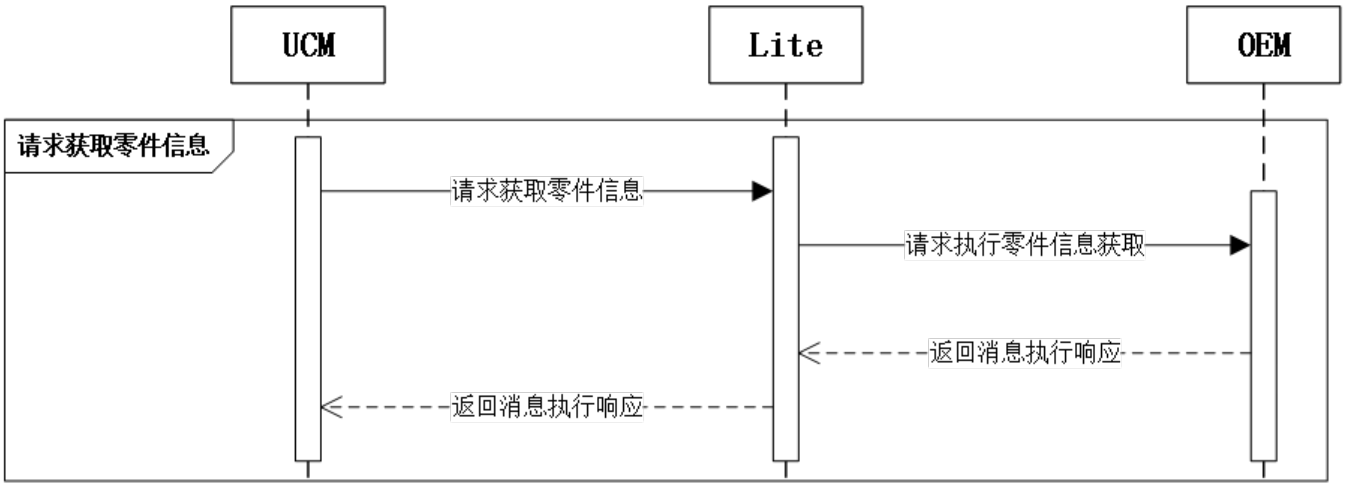


图 3-9 零件信息采集时序图

3.9.3 接口设计

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**零件信息为判断该零件是否有更新的依据，由车机供应商负责
- **接口原型**

```
/**
 * @brief 零件信息收集
 *
 * @param[in] pstReqData     描述: 请求获取的零件标识信息 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 请求获取零件信息的结果
 */
extern D32S oem_collectEcuInfo(ADM_UAM_ECU_COLLECT_REQ_T* pstReqData);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pstReqData	ADM_UAM_ECU_COLLECT_REQ_T*	input	non-nullptr	请求获取的零件标识信息
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.10 车辆状态查询

3.10.1 功能描述

用于OTA过程中查询和控制车辆状态，如查询蓄电池电量、车速、档位等。控制车辆上下高压等。

- 车速，单位km/h
- 发动机转速，单位rpm
- 车辆钥匙状态，0为OFF档，1为ACC档，2为ON档，3为START档
- 车辆档位，1为P档，2为R档，3为空挡，4为D档
- 车辆手刹状态，1为未拉起手刹，2为拉起手刹
- 车辆充电状态，1正在充电，2非充电中
- OBD状态
- 蓄电池电压，可配置范围值
- 动力电池电量百分比
- 辅助驾驶状态
- 车辆放电状态

3.10.2 业务时序

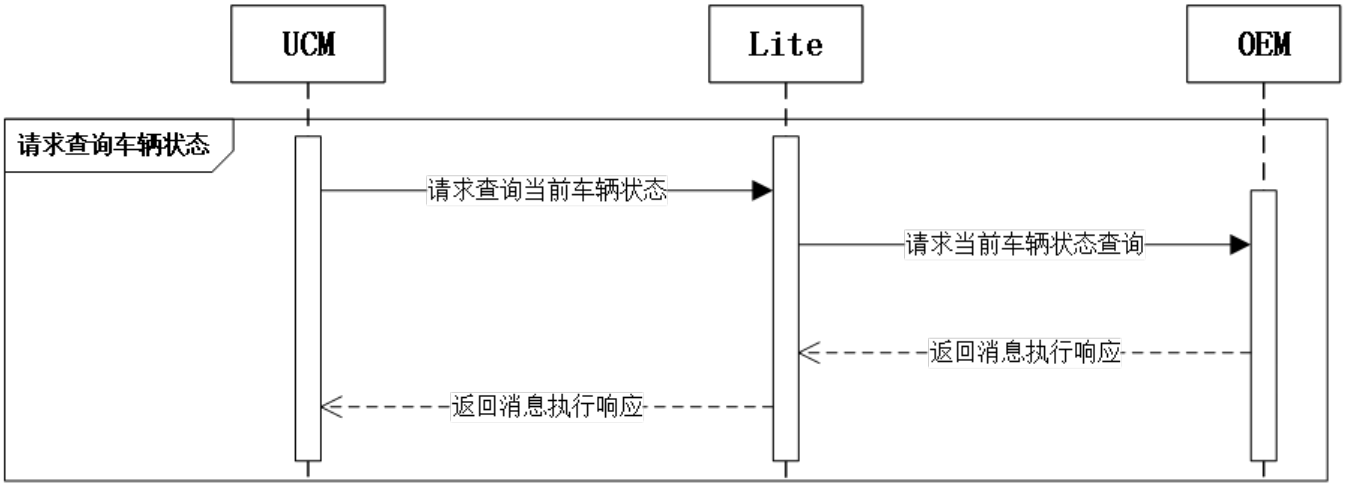


图 3-10 车辆状态时序图

3.10.3 接口设计

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**下载、升级前置条件判断，获取车辆是否处于READAY状态
- **接口原型**

```
/**
 * @brief 查询车辆状态
 *
 * @param[out] pstValue     描述: 获取的车辆状态信息 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 车辆状态查询操作结果
 */
extern D32S oem_getVehicleState(AMD_GET_VEHICLE_STATE_T* pstValue);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pstValue	AMD_GET_VEHICLE_STATE_T*	output	non-nullptr	查询的车辆状态信息
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.10.3.1 查询车辆车速

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**OTA在下载完成后，HMI需要根据当前车速是否静止，来判断是否弹窗提示下载完成
- **接口原型**

```
/**
 * @brief 查询车辆车速
 *
 * @param[in] eType     描述: 查询的车辆状态类型 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @param[out] pstValue     描述: 对应的查询结果信息 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 车辆车速查询的操作结果
 */
extern D32S oem_getCarSpeed(AMD_GET_VEHICLE_STATE_E eType, ADM_VSM_VARIABLE_T* pstValue);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注

eType	AMD_GET_VEHICLE_STATE_E	input	枚举	待查询的车辆状态类型，当查询的是车速信息取值为：ADM_VEHICLE_STATE_CARSPEED
pstValue	ADM_VSM_VARIABLE_T*	output	non-nullptr	车速取值大于等于0，单位km/h
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.10.3.2 查询发动机转速

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**下载、升级前置条件/过程车辆状态获取并校验
- **接口原型**

```
/**
 * @brief 查询发动机转速
 *
 * @param[in] eType     描述: 查询的车辆状态类型 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @param[out] pstValue     描述: 对应的查询结果信息 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 车辆发动机转速查询的操作结果
 */
extern D32S oem_getEngineSpeed(AMD_GET_VEHICLE_STATE_E eType, ADM_VSM_VARIABLE_T* pstValue);
```

- **接口参数**

参数名称	**数据类型**	**输入/输出**	**有效值范围**	**备注**
eType	AMD_GET_VEHICLE_STATE_E	input	枚举	待查询的车辆状态类型，当查询的是发动机转速信息取值为：ADM_VEHICLE_STATE_ENGINESPEED
pstValue	ADM_VSM_VARIABLE_T*	output	non-nullptr	转速取值大于等于0，单位rpm
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.10.3.3 查询钥匙状态

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**下载、升级前置条件/过程车辆状态获取并校验
- **接口原型**

```
/**
 * @brief 查询钥匙状态
 *
 * @param[in] eType     描述: 查询的车辆状态类型 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @param[out] pstValue     描述: 对应的查询结果信息 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 钥匙状态查询的操作结果
 */
extern D32S oem_getKeyState(AMD_GET_VEHICLE_STATE_E eType, ADM_VSM_VARIABLE_T* pstValue);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
eType	AMD_GET_VEHICLE_STATE_E	input	枚举	待查询的车辆状态类型，当查询的是钥匙状态信息取值为：ADM_VEHICLE_STATE_KEYSTATE
pstValue	ADM_VSM_VARIABLE_T*	output	non-nullptr	0表示OFF，1表示ACC，2表示ON，3表示START
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.10.3.4 查询车辆档位

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**HMI模块在执行重启操作前，需要对挡位状态进行校验
- **接口原型**

```
/**
 * @brief 查询车辆档位
 *
 * @param[in] eType     描述: 查询的车辆状态类型 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @param[out] pstValue     描述: 对应的查询结果信息 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 车辆档位查询的操作结果
 */
extern D32S oem_getGearSignal(AMD_GET_VEHICLE_STATE_E eType, ADM_VSM_VARIABLE_T* pstValue);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
eType	AMD_GET_VEHICLE_STATE_E	input	枚举	待查询的车辆状态类型，当查询的是档位信息取值为：ADM_VEHICLE_STATE_GEARSTATE
pstValue	ADM_VSM_VARIABLE_T*	output	non-nullptr	车辆挡位状态,取值范围为[1,3] 1: 车辆处于P挡 2: 车辆处于非P挡 3: 获取挡位状态异常
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.10.3.5 查询车辆手刹状态

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**下载、升级前置条件/过程车辆状态获取并校验
- **接口原型**

```
/**
 * @brief 查询手刹状态
 *
 * @param[in] eType     描述: 查询的车辆状态类型 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @param[out] pstValue     描述: 对应的查询结果信息 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 手刹状态查询的操作结果
 */
extern D32S oem_getHandBrakeState(AMD_GET_VEHICLE_STATE_E eType, ADM_VSM_VARIABLE_T* pstValue);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
eType	AMD_GET_VEHICLE_STATE_E	input	枚举	待查询的车辆状态类型，当查询的是手刹状态信息取值为：ADM_VEHICLE_STATE_HANDBRAKE
pstValue	ADM_VSM_VARIABLE_T*	output	non-nullptr	手刹状态取值[1,2] 1: 未拉起 2: 拉起
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.10.3.6 查询充电状态

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**下载、升级前置条件/过程车辆状态获取并校验
- **接口原型**

```
/**
 * @brief 查询车辆的充电状态
 *
 * @param[in] eType     描述: 查询的车辆状态类型 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @param[out] pstValue     描述: 对应的查询结果信息 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 车辆充电状态查询的操作结果
 */
extern D32S oem_getChargeState(AMD_GET_VEHICLE_STATE_E eType, ADM_VSM_VARIABLE_T* pstValue);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
eType	AMD_GET_VEHICLE_STATE_E	input	枚举	待查询的车辆状态类型，当查询的是充电状态信息取值为：ADM_VEHICLE_STATE_CHARGE
pstValue	ADM_VSM_VARIABLE_T*	output	non-nullptr	当前是否在充电中 1: 正在充电 2: 非充电中
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.10.3.7 查询OBD状态

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**下载、升级前置条件/过程车辆状态获取并校验
- **接口原型**

```
/**
 * @brief 查询车辆OBD状态
 *
 * @param[in] eType     描述: 查询的车辆状态类型 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @param[out] pstValue     描述: 对应的查询结果信息 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 车辆OBD状态查询的操作结果
 */
extern D32S oem_getOBDState(AMD_GET_VEHICLE_STATE_E eType, ADM_VSM_VARIABLE_T* pstValue);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
eType	AMD_GET_VEHICLE_STATE_E	input	枚举	待查询的车辆状态类型，当查询的是OBD状态信息取值为：ADM_VEHICLE_STATE_OBDSTATE
pstValue	ADM_VSM_VARIABLE_T*	output	non-nullptr	车辆OBD状态,取值范围为[1,2] 1: OBD接口未接入 2: OBD接口已接入
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.10.3.8 查询蓄电池电压

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**下载、升级前置条件/过程车辆状态获取并校验
- **接口原型**

```
/**
 * @brief 查询车辆蓄电池电压
 *
 * @param[in] eType     描述: 查询的车辆状态类型 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @param[out] pstValue     描述: 对应的查询结果信息 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 车辆蓄电池电压查询的操作结果
 */
extern D32S oem_getBatteryVoltage(AMD_GET_VEHICLE_STATE_E eType, ADM_VSM_VARIABLE_T* pstValue);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
eType	AMD_GET_VEHICLE_STATE_E	input	枚举	待查询的车辆状态类型，当查询的是蓄电池电压信息取值为：ADM_VEHICLE_STATE_BATTERYVOL
pstValue	ADM_VSM_VARIABLE_T*	output	non-nullptr	车辆蓄电池电压，取值范围为：[0-100]
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.10.3.9 查询动力电池电量百分比

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**下载、升级前置条件/过程车辆状态获取并校验
- **接口原型**

```
/**
 * @brief 查询动力电池电量百分比
 *
 * @param[in] eType     描述: 查询的车辆状态类型 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @param[out] pstValue     描述: 对应的查询结果信息 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 车辆动力电池电量百分比查询的操作结果
 */
extern D32S oem_getPowerBatteryRatio(AMD_GET_VEHICLE_STATE_E eType, ADM_VSM_VARIABLE_T* pstValue);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
eType	AMD_GET_VEHICLE_STATE_E	input	枚举	待查询的车辆状态类型，当查询的是动力电池电量百分比取值为：ADM_VEHICLE_STATE_POWERBATTERY
pstValue	ADM_VSM_VARIABLE_T*	output	non-nullptr	动力电池电量百分比，取值范围为：[0-100]
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.10.3.10 查询辅助驾驶状态

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**下载、升级前置条件/过程车辆状态获取并校验
- **接口原型**

```
/**
 * @brief 查询辅助驾驶状态
 *
 * @param[in] eType     描述: 查询的车辆状态类型 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @param[out] pstValue     描述: 对应的查询结果信息 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 辅助驾驶状态查询的操作结果
 */
extern D32S oem_getAssistedDriving(AMD_GET_VEHICLE_STATE_E eType, ADM_VSM_VARIABLE_T* pstValue);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
eType	AMD_GET_VEHICLE_STATE_E	input	枚举	待查询的车辆状态类型，当查询的是辅助驾驶状态取值为：ADM_VEHICLE_STATE_ASSISTEDDRIVING
pstValue	ADM_VSM_VARIABLE_T*	output	non-nullptr	车辆辅助驾驶状态,取值范围为[1, 2] 1: 在辅助驾驶状态中 2: 未在辅助驾驶状态中
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.10.3.11 查询放电状态

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**下载、升级前置条件/过程车辆状态获取并校验
- **接口原型**

```
/**
 * @brief 查询放电状态
 *
 * @param[in] eType     描述: 查询的车辆状态类型 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @param[out] pstValue     描述: 对应的查询结果信息 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 车辆放电状态查询的操作结果
 */
extern D32S oem_getPowerBatteryRatio(AMD_GET_VEHICLE_STATE_E eType, ADM_VSM_VARIABLE_T* pstValue);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
eType	AMD_GET_VEHICLE_STATE_E	input	枚举	待查询的车辆状态类型，当查询的是车辆放电状态取值为：ADM_VEHICLE_STATE_DISCHARGESTATE
pstValue	ADM_VSM_VARIABLE_T*	output	non-nullptr	车辆放电状态,取值范围为[1, 2] 1: 在放电中 2: 未在放电中
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.11 车辆上下高压

3.11.1 功能描述

设置整车上、下高压

3.11.2 业务时序

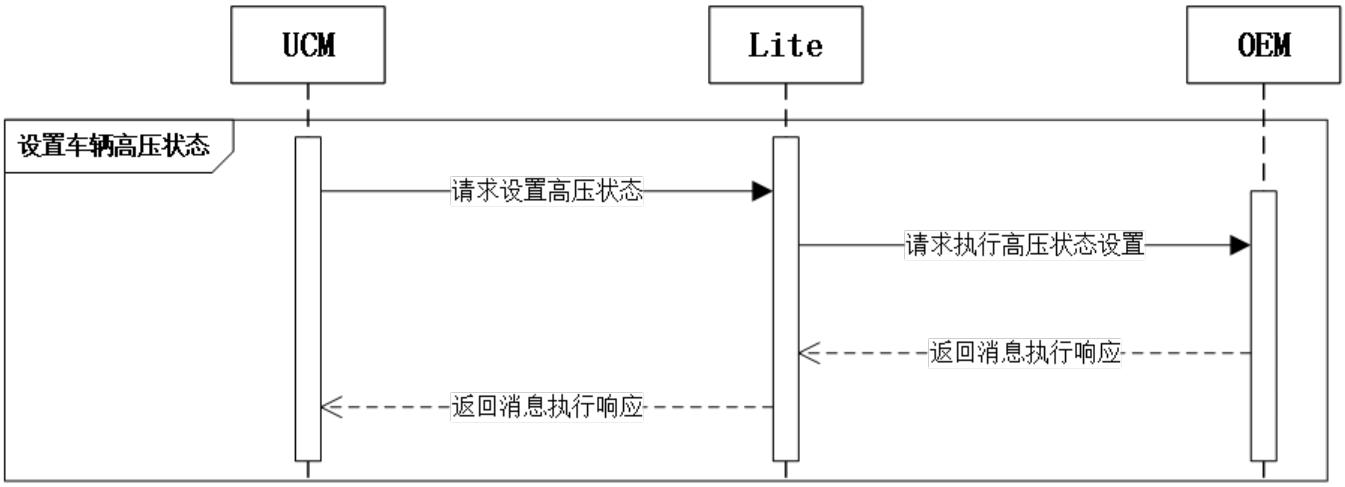


图 3-11 车辆上下高压时序图

3.11.3 接口设计

3.11.3.1 查询车辆上/下高压状态

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**当OTA进入安装过程，遇到低压件需要上高压升级，反之，高压件需要下高压 （即：低压）升级
- **接口原型**

```
/**
 * @brief 查询当前车辆的上/下高压状态
 *
 * @param[out] pOptType     描述: 返回的状态 \n
 * 范围值:指针 \n
 * 单位: N/A \n
 *
 * @param[out] pStayTime     描述: 上、下高压的剩余持续时间 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 查询上/下高压的执行结果
 */
extern D32S oem_queryVoltageState(D32S* pOptType, D32U* pStayTime);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pOptType	D32S*	output	non-nullptr	操作指令，1表示上高压，2表下高压
pStayTime	D32U *	output	non-nullptr	整车上/下高压的剩余持续时间（相对时间，相对于当前时刻，到设置的目标时间的秒数。单位：秒）

返回值	D32S	output	int	0:成功，否则返回具体错误码

3.11.3.2 设置车辆上/下高压

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**当OTA进入安装过程，遇到低压件需要上高压升级，反之，高压件需要下高压 （即：低压）升级
- **接口原型**

```
/**
 * @brief 设置上下高压
 *
 * @param[in] iOptType     描述: 操作指令，上/下高压 \n
 * 范围值:[1,2] \n
 * 单位: N/A \n
 *
 * @param[in] uiTimeOutS     描述: 上、下高压的持续时间 \n
 * 范围值: >0 \n
 * 单位: Second \n
 *
 * @return D32S 设置上下高压的执行结果
 */
extern D32S oem_setVoltageState(D32S iOptType, D32U uiTimeOutS);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
iOptType	D32S	input	[1,2]	操作指令，1表示上高压，2表下高压
uiTimeOutS	D32U	input	>=0	整车上、下高压的持续时间（相对时间，相对于当前时刻，到目标时间的秒数。单位：秒）
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.12 整车上下电信号监控

3.12.1 功能描述

用于监控整车上下电信号。

3.12.2 业务时序

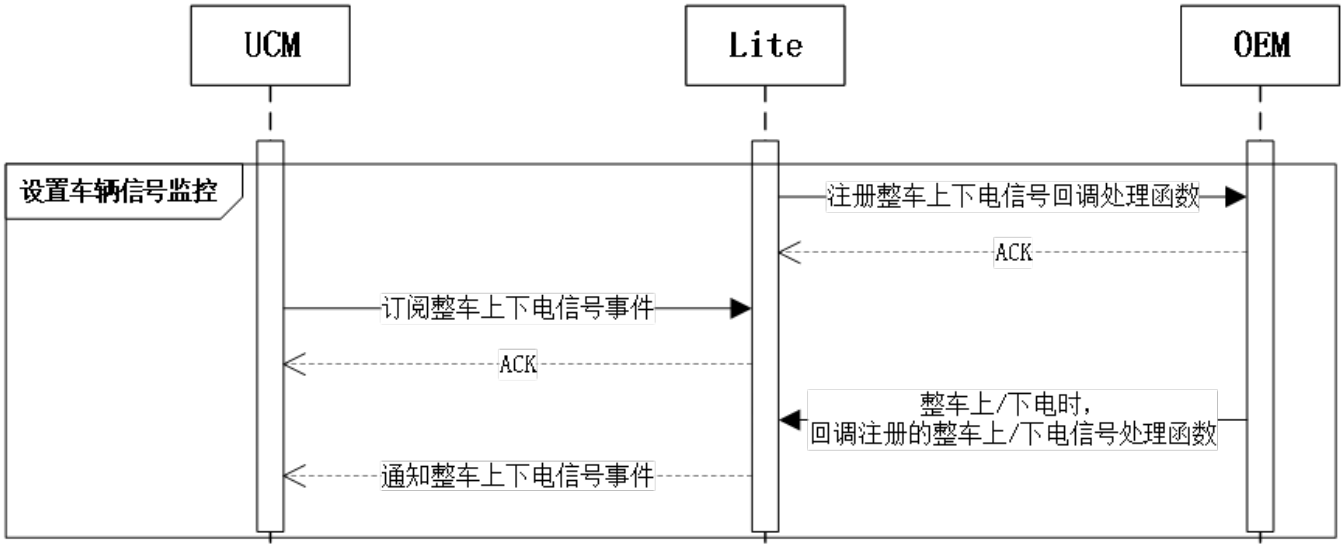


图 3-12 车辆信息监控时序图

3.12.3 接口设计

3.12.3.1 注册整车上下电信号接收回调处理函数

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**注册整车上/下电信号接收回调处理函数，当OEM监控到整车上/下电后调用该注册的回调函数通知上层
- **接口原型**

```
/**
 * @brief 注册整车上下电信号接收回调处理函数
 *
 * @param[in] pfnSignalCB     描述: 接收到整车上下电信号的回到处理函数 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S  注册整车上下电信号接收回调处理函数操作的结果
 */
extern D32S oem_registerVehicleSignalCB(ADM_VSM_VEHICLE_SIGNAL_CALLBACK_F  pfnSignalCB);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pfnSignalCB	ADM_VSM_VEHICLE_SIGNAL_CALLBACK_F	input	non-nullptr	接收到整车上下电信号的回调处理函数
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.12.3.2 查询整车上下电信号接口

- **接口说明**
  - **调用方：**ABUP

    - **实现方：**车机供应商
    - **同步/异步：**同步接口，接口调用的超时时间为30秒
    - **业务场景：**提供查询当前是否已经发送了整车上/下电信号，避免由于整车上下电时由于启动时序问题导致的OTA主控错过OEM发送的整车上下电信号
- **接口原型**

```
/**
 * @brief 查询车辆信号
 *
 * @param[in] pSignalType     描述: 查询到的整车上下电信号类型 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @return D32S 车辆信号查询操作的结果
 */
extern D32S oem_registerVehicleSignalCB(D32S* pSignalType);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pSignalType	D32S*	output	non-nullptr	查询的整车上/下电信号类型 1：已推送整车上电信号 2：已推送整车下电信号 3：未推送整车上下电信号
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.12.3.3 设置整车上下电信号监控接口

- **接口说明**
    - **调用方：**ABUP
    - **实现方：**车机供应商
    - **同步/异步：**同步接口，接口调用的超时时间为30秒
    - **业务场景：**启动整车上下电信号监控，一般为OTA主控启动时监控
- **接口原型**

```
/**
 * @brief 设置整车上下电信号监控
 *
 * @param[in] iSignalType     描述: 待监控的车辆信号类型 \n
 * 范围值: >0 \n
 * 单位: N/A  \n
 *
 * @return D32S 设置车辆信号监控的结果
 */
extern D32S oem_setVehicleSignalSpy(D32S iSignalType);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
iSignalType	D32S	input	>0	待监控的车辆信号类型 1：整车上电信号 2：整车下电信号 3：整车上下电信号
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.12.3.4 取消车辆信号监控接口

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**取消整车上下电信号监控，一般为整车下电时OTA主控进行反初始化时调用
- **接口原型**

```
/**
 * @brief 取消整车上下电信号监控接口
 *
 * @param[in] iSignalType     描述: 待取消的车辆信号类型\n
 * 范围值: >0 \n
 * 单位: N/A  \n
 *
 * @return D32S 取消整车上下电信号监控接口操作的结果
 */
extern D32S oem_cancleVehicleSpy(D32S iSignalType);
```

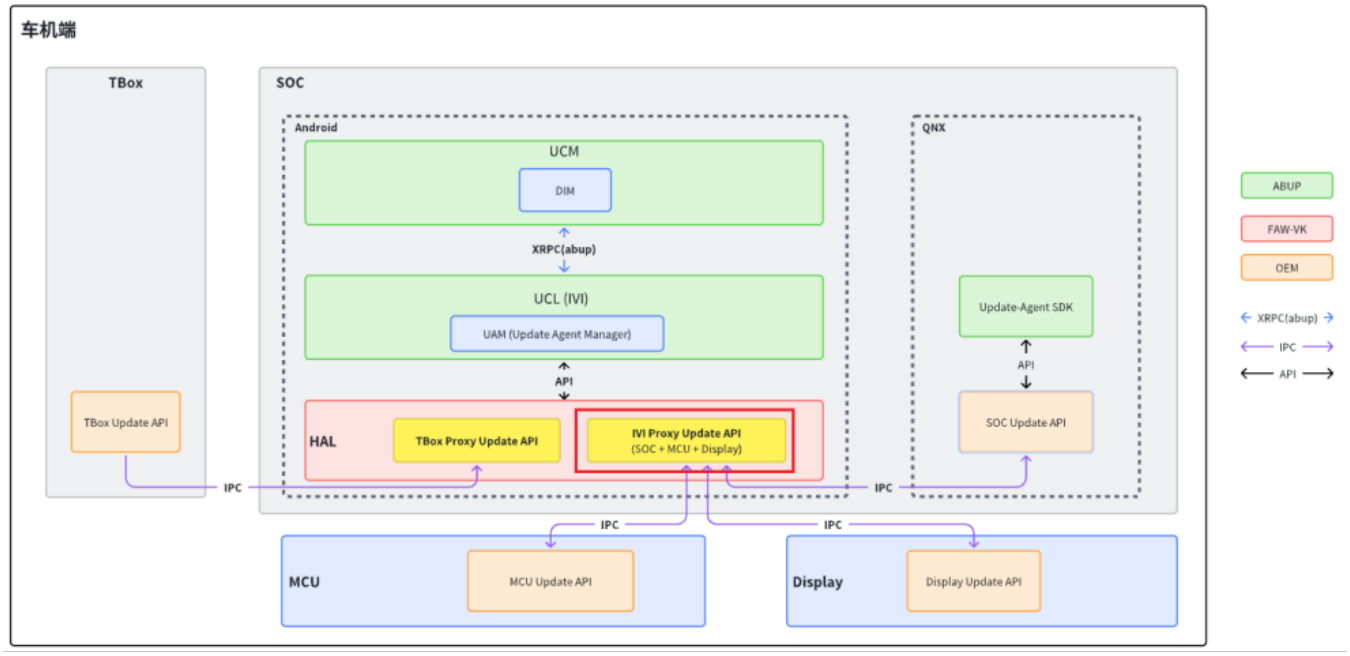
- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
iSignalType	D32S	input	>0	待取消的车辆信号类型 1：整车上电信号 2：整车下电信号 3：整车上下电信号
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.13 IVI Upgrade（IVI供应商）

3.13.1 功能描述

用于智能件IVI的升级对接，IVI的升级范围包括三部分：车机SOC（双系统，Android + QNX 升级）、MCU以及中控/仪表升级。具体如下图：



3.13.2 业务时序

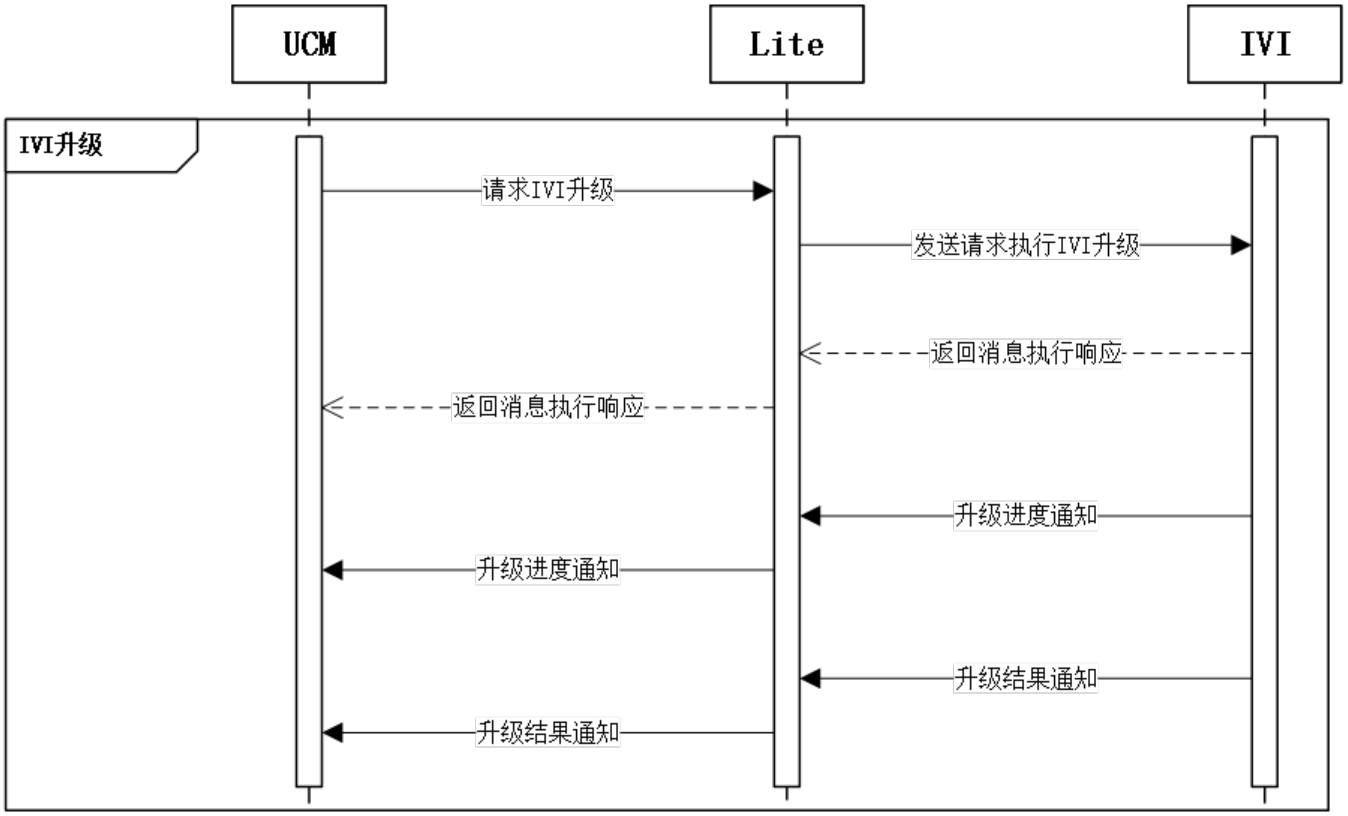


图 3-13 IVI升级时序图

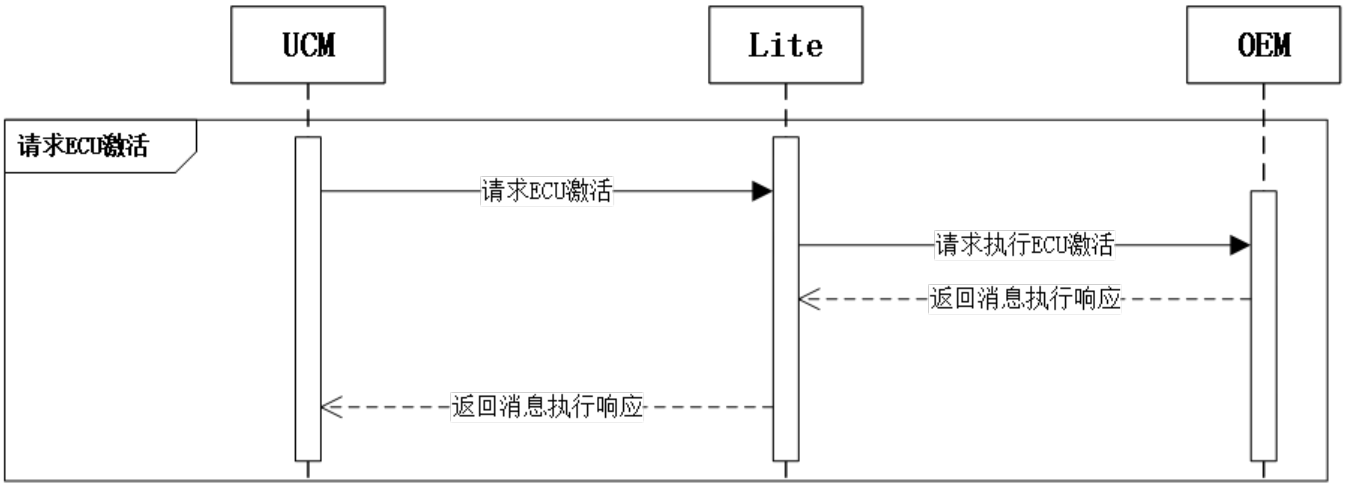


图 3-14 ECU激活时序图

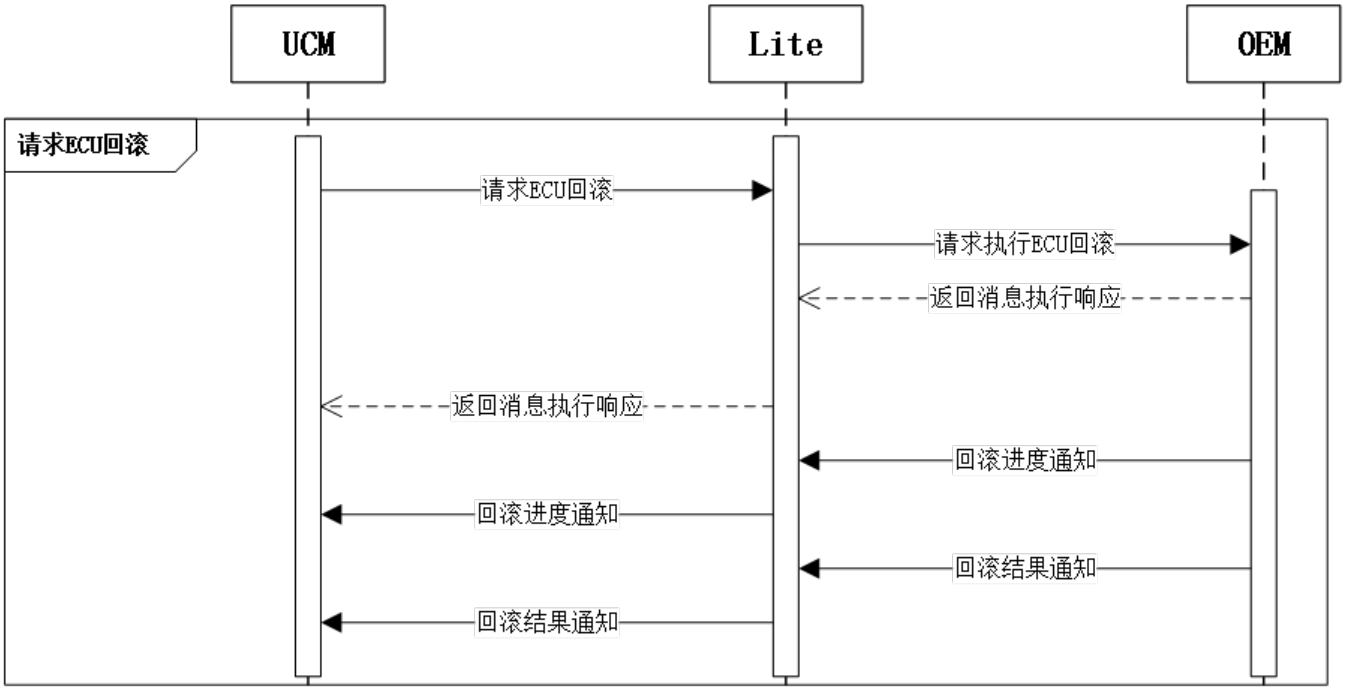


图 3-15 ECU回滚时序图

3.13.3 接口设计

3.13.3.1 IVI升级回调注册

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**注册升级过程的升级/回滚进度、升级/回滚结果、激活结果回调
- **接口原型**

```
/**
 * @brief 注册IVI升级接口回调
 *
 * @param[in] pfnFlashProgressCb 描述: 升级/回滚进度回调函数 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @param[in] pfnFlashResultCb 描述: 升级/回滚结果回调函数 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @param[in] pfnActiveResultCb 描述: 激活结果回调函数 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @return D32S 注册IVI升级接口回调操作的结果
 */
extern D32S oem_registerIVIupgradeCB
(
    FLASH_PROGRESS_CALLBACK_F pfnFlashProgressCb,
    FLASH_RESULT_CALLBACK_F pfnFlashResultCb,
    FLASH_RESULT_CALLBACK_F    pfnActiveResultCb
);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pfnFlashProgressCb	FLASH_PROGRESS_CALLBACK_F	input	non-nullptr	升级/回滚进度回调函数
pfnFlashResultCb	FLASH_RESULT_CALLBACK_F	input	non-nullptr	升级/回滚结果回调函数
pfnActiveResultCb	FLASH_RESULT_CALLBACK_F	input	non-nullptr	激活结果回调函数
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.13.3.2 ECU升级接口

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**异步接口，刷写进度、结果通过注册的升级进度、结果回调函数通知回来
  - **业务场景：**IVI触发升级后，需要回调升级进度和结果给到ABUP
  - **依赖：**实现放需要根据传入的升级类型做对对应的处理
    - **无感升级：**实现方需要对A/B分区的零件进行B分区刷写，同时因为是无感升级，所以升级时资源占用不能过高（CPU及内存占用），具体可参考UA对内存和CPU的参数进行限制处理
    - **无感升级失败的有感升级：**实现方需要对域内各零件进行刷写。同时由于时有感升级，所以升级时资源占用可以高一些
    - **无感升级成功的有感升级：**实现方需要对域内除了无感升级成功的零件进行刷写

- **接口原型**

```
/**
 * @brief 请求刷写
 *
 * @param[in] eUpdateType     描述: 升级类型 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @param[in] pstReqData     描述: 刷写请求参数 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 请求刷写的结果
 */
extern D32S oem_updateIVIRequest
(
    D32S  eUpdateType,
    ADM_UAM_UPDATE_REQ_T* pstReqData
);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
eUpdateType	D32S	input	int	升级类型
pstReqData	ADM_UAM_UPDATE_REQ_T*	input	non-nullptr	刷写请求参数
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.13.3.3 ECU激活接口

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**异步接口，激活结果通过注册的激活结果回调函数通知回来
  - **业务场景：**具有A/B分区的ECU，其中一个分区升级成功后，需要进行激活操作（A/B系统切换的过程，即从A系统切换至B系统的过程），来完成设备系统切换
  - **依赖：**ECU需要根据升级结果，完成系统切换并且生效（即根据升级信息记录当前刷写面，激活时切换到刷写面并使系统生效），分区生效结果需要通过异步通知到激活的结果回调

    注：若激活失败，实现方需要在激活结果通知前将系统保持在未刷写的分区系统，并保证系统的正常运行。（若激活结果失败，即实现方需要通过上述注册的激活的结果回调函数通知上层激活失败，并且若实现方已经将系统成功切换到B分区，则实现方需要将系统切回A分区。此时激活的结果回调通知需要细化对应的错误码）

- **接口原型**

```
/**
 * @brief 请求激活
 *
 * @param[in] pstReqData     描述: 激活请求参数 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 请求激活的结果
 */
extern D32S oem_activeRequest(ADM_UAM_ACTIVE_REQ_T* pstReqData);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pstReqData	ADM_UAM_ACTIVE_REQ_T*	input	non-nullptr	刷写请求参数
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.13.3.4 ECU回滚接口

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**异步接口，回滚进度、结果通过注册的回滚进度、结果回调函数通知回来
  - **业务场景：**升级失败或升级成功但依赖的零件需要该零件回滚时，将该ECU的固件版本恢复到之前的一个稳定版本
  - **依赖：**对手件需要记录上次升级的接口执行结果，并根据如下策略执行回滚：
    - **升级成功的全面回滚：**
      - 若上次升级成功，则执行全面回滚操作。
      - 对于有回滚包（对于A/B分区的零件， 回滚包为整包形式）的情况，则根据回滚包信息进行回滚。
      - 对于A/B分区且无回滚包的情况，实现方需要通过A面还原B面，来完成回滚。
    - **升级失败的回滚：**
      - 若域内部分零件升级成功，则仅回滚升级成功的零件。
      - 对于有回滚包（对于A/B分区的零件， 回滚包为整包形式）的情况，则根据回滚包信息进行回滚。
      - 对于A/B分区且无回滚包的情况，实现方需要通过A面还原B面，来完成回滚。
- **接口原型**

```
/**
 * @brief 请求回滚
 *
 * @param[in] pstReqData     描述: 回滚请求参数 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 请求回滚的结果
 */
extern D32S oem_rollbackRequest(ADM_UAM_ROLLBACK_REQ_T* pstReqData);
```

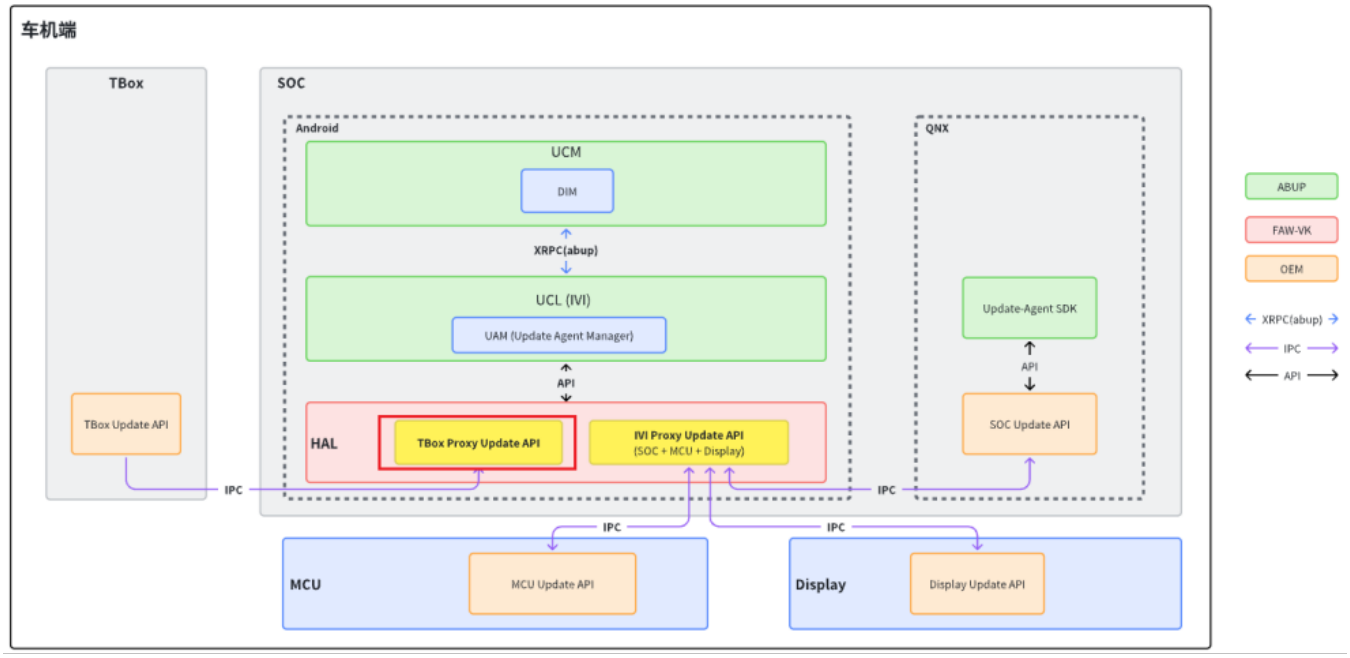
- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pstReqData	ADM_UAM_ROLLBACK_REQ_T*	input	non-nullptr	回滚请求参数
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.14 TBox Update（IVI供应商）

3.14.1 功能描述

用于车机端TBox的升级对接。该接口由IVI提供，IVI提供TBOX的硬件抽象层，并实现TBOX的真实刷写。如下图：



3.14.2 业务时序

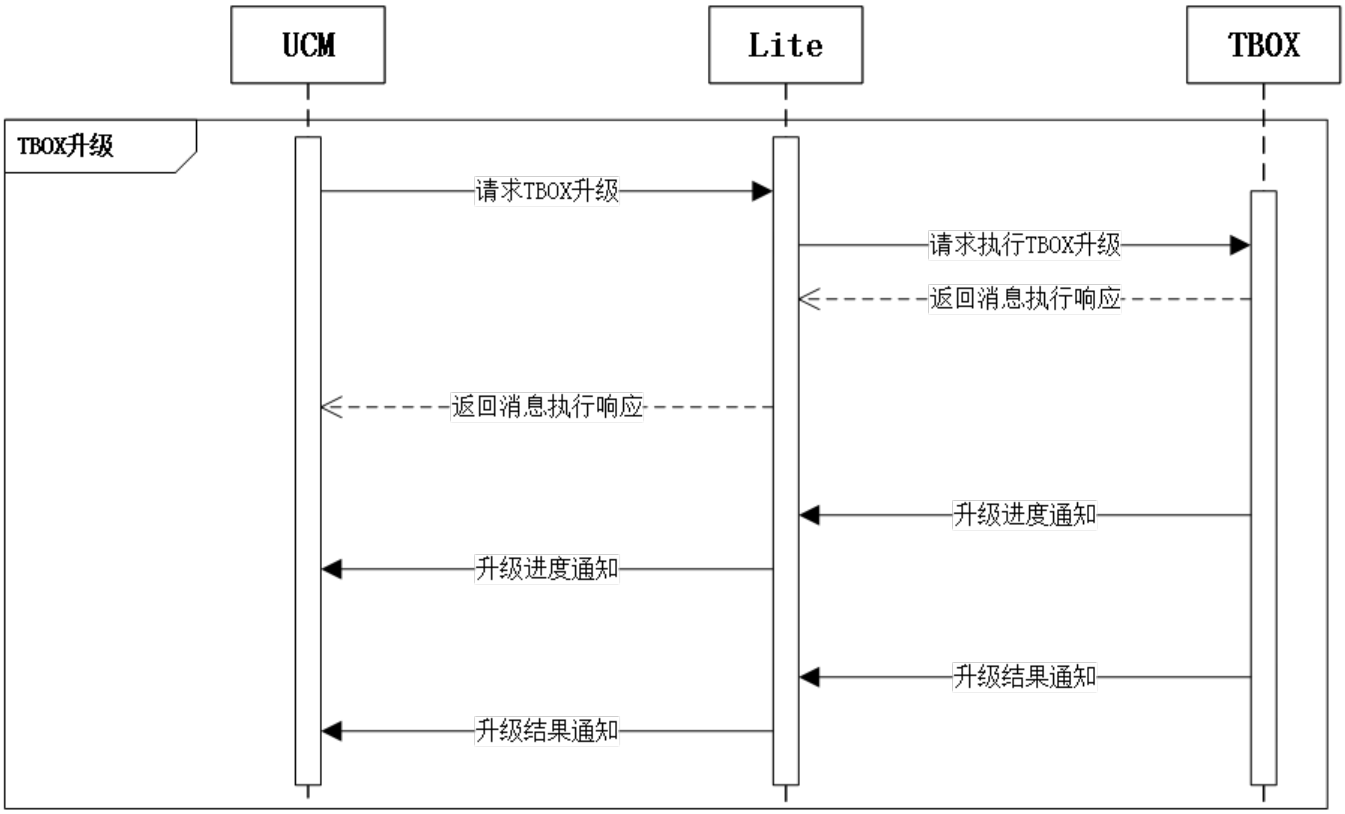


图 3-16 TBOX升级时序图

3.14.3 接口设计

3.14.3.1 注册升级进度/结果通知回调函数

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**注册升级过程的升级进度/结果通知回调
- **接口原型**

```
/**
 * @brief 注册TOBX升级接口回调
 *
 * @param[in] pfnFlashProgressCb 描述: 升级进度回调函数 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @param[in] pfnFlashResultCb 描述: 升级结果回调函数 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @return D32S 注册TOBX升级接口回调操作的结果
 */
extern D32S oem_registerIVIupgradeCB
(
    FLASH_PROGRESS_CALLBACK_F pfnFlashProgressCb,
    FLASH_RESULT_CALLBACK_F   pfnFlashResultCb
);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pfnFlashProgressCb	FLASH_PROGRESS_CALLBACK_F	input	non-nullptr	升级进度回调函数
pfnFlashResultCb	FLASH_RESULT_CALLBACK_F	input	non-nullptr	升级结果回调函数
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.14.3.2 ECU升级接口

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**异步接口
  - **业务场景：**TBOX触发升级后，需要回调升级进度和结果给到ABUP
- **接口原型**

```
/**
 * @brief 请求TBOX刷写
 *
 * @param[in] pstReqData     描述: 刷写请求参数 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 请求TBOX刷写的结果
 */
extern D32S oem_updateTBOXRequest(ADM_UAM_UPDATE_REQ_T* pstReqData);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pstReqData	ADM_UAM_UPDATE_REQ_T*	input	non-nullptr	刷写请求参数
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.15 UDS Flash（UDS刷写）

3.15.1 功能描述

用于非智能件的UDS刷写。

3.15.2 业务时序

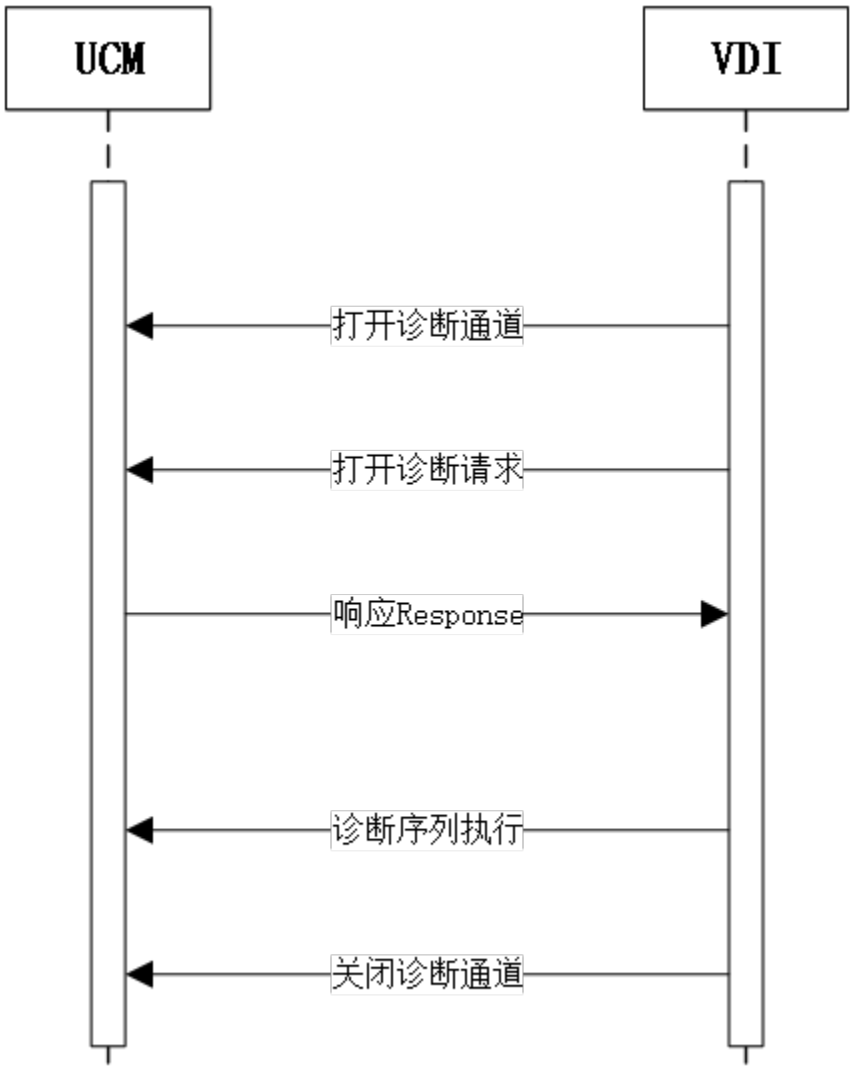


图 3-17 UDS刷写时序图

3.15.3 接口设计

3.15.3.1 UDS TP接口初始化

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**供应商初始化时候调用主控HOST库进行UDS TP初始化
- **接口原型**

```
/**
 * @brief 主控HOST库进行UDS TP初始化
 *
 * @param[in] pArgs     描述: 初始化参数 \n
 * 范围值: 指针 \n
 * 单位: N/A \n
 *
 * @return D32S 初始化的结果
 */
extern D32S oem_udstpInit(const void* pArgs);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pArgs	const void*	input	non-nullptr	args参数在具体项目中由HOST指定
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.15.3.2 UDS TP接口反初始化

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**供应商反初始化时候调用主控HOST库进行UDS TP反初始化
- **接口原型**

```
/**
 * @brief 主控HOST库进行UDS TP反初始化
 *
 * @return D32S 反初始化的结果
 */
extern D32S oem_udstpDeinit(DVOID);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.15.3.3 UDS TP打开通道

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**ABUP调用HOST库打开UDS TP通道。如果是doip协议，这里会激活通道
- **接口原型**

```
/**
 * @brief 打开UDS TP通道
```

```
 *
 * @param[in] uiChannel     描述: 通道值 \n
 * 范围值: >=0 \n
 * 单位: N/A \n
 *
 * @param[in] uiPhyical_addr     描述: 请求地址 \n
 * 范围值: >=0 \n
 * 单位: N/A  \n
 *
 * @param[in] uiFunction_addr     描述: 功能寻址地址 \n
 * 范围值: >=0 \n
 * 单位: N/A  \n
 *
 * @param[in] uiRespond_addr     描述: UDS应答地址 \n
 * 范围值: >=0 \n
 * 单位: N/A  \n
 *
 * @param[in] pfnConfirmCb     描述: 发送确认回调函数 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @param[in] pfnIndicationCb     描述: 指示数据回调函数 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @param[in] pPriv     描述: 透传参数，具体可协商 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @param[in] pHandle     描述: UDStp打开时返回的UDS TP Client的句柄 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @return D32S 打开UDS TP通道的结果
 */
extern D32S oem_openUdstp
(
D32U uiChannel,
```

```
D32U uiPhyical_addr,
D32U uiFunction_addr,
D32U uiRespond_addr,
OEM_UDSTP_CONFIRM_FUNC pfnConfirmCb,
OEM_UDSTP_INDICATION_FUNC pfnIndicationCb,
const void* pPriv,
D64U* pHandle
);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
uiChannel	D32U	input	>=0	通道值
uiPhyical_addr	D32U	input	>=0	请求地址
uiFunction_addr	D32U	input	>=0	功能寻址地址
uiRespond_addr	D32U	input	>=0	UDS应答地址
pfnConfirmCb	OEM_UDSTP_CONFIRM_FUNC	input	non-nullptr	发送确认回调函数
pfnIndicationCb	OEM_UDSTP_INDICATION_FUNC	input	non-nullptr	指示数据回调函数
pPriv	const void*	input	non-nullptr	透传参数，具体可协商
pHandle	D64U*	input	non-nullptr	UDStp打开时返回的UDS TP Client的句柄
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.15.3.4 UDS TP请求发送数据

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**ABUP 发送DOIP报文。request_addr为请求地址，HOST在收到 response_addr的ECU应答数据时，调用oem_openUdstp 接口注册的oem_udstp_indication回调函数通知响应数据
- **接口原型**

```
/**
 * @brief UDS TP请求发送数据
 *
 * @param[in] pHandle     描述: UDStp打开时返回的UDS TP Client的句柄 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @param[in] uiRequest_addr     描述: 功能寻址地址 \n
 * 范围值: >=0 \n
 * 单位: N/A  \n
 *
 * @param[in] uiRespond_addr     描述: UDS应答地址 \n
 * 范围值: >=0 \n
 * 单位: N/A  \n
 *
 * @param[in] pBuf     描述: 请求数据Buffer \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @param[in] uiLen     描述: 请求数据长度 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @return D32S UDS TP请求发送数据的结果
 */
extern D32S oem_udstpRequest
(
D64U pHandle,
D32U uiRequest_addr,
D32U uiRespond_addr,
const D8U* pBuf,
D32U uiLen
);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pHandle	D64U*	input	non-nullptr	UDStp打开时返回的UDS TP Client的句柄
uiRequest_addr	D32U	input	>=0	功能寻址地址请求地址，可以是物理地址或者功能地址

uiRespond_addr	D32U	input	>=0	UDS应答地址
pBuf	const D8U*	input	non-nullptr	请求数据Buffer
uiLen	D32U	input	>=0	请求数据长度
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.15.3.5 UDS TP发送确认回调

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **业务场景：**ABUP调用HOST库进行发送诊断数据
- **接口原型**

```
/**
 * @brief 发送UDS TP数据结果确认回调
 *
 */
typedef D32S (*OEM_UDSTP_CONFIRM_FUNC)
(
    D64U uiHandle,
    D32U uiRequest_addr,
    D32U uiResponse_addr,
    OEM_UDSTP_CONFIRM_E eResult
);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
uiHandle	D64U	input	>=0	UDStp打开时返回的UDS TP Client的句柄
uiRequest_addr	D32U	input	>=0	请求地址
uiResponse_addr	D32U	input	>=0	UDS应答地址
eResult	OEM_UDSTP_CONFIRM_E	input	枚举	确认结果
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.15.3.6 UDS TP回包数据回调

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **业务场景：**ABUP调用HOST oem_udstpRequest接口发送UDS数据，ECU响应的UDS数据，HOST将数据通过该回调函数通知给ABUP
- **接口原型**

```
/**
 * @brief 返回UDS TP数据回调
 *
 */
typedef D32S (*OEM_UDSTP_INDICATION_FUNC)
(
    D64U uiHandle,
    D32U uiRequest_addr,
    D32U uiResponse_addr,
    const D8U* pBuf,
    D32U uiLen
);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
uiHandle	D64U	input	>=0	UDStp打开时返回的UDS TP Client的句柄
uiRequest_addr	D32U	input	>=0	请求地址
uiResponse_addr	D32U	input	>=0	UDS应答地址
pBuf	const D8U*	input	non-nullptr	请求数据Buffer
uiLen	D32U	input	>=0	请求数据长度
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.15.3.7 UDS TP关闭通道

- **接口说明**
  - **调用方：**ABUP
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**供应商调用ABUP库关闭UDS TP通道
- **接口原型**

```
/**
 * @brief 关闭UDS TP通道
 *
 * @param[in] pHandle     描述: UDStp打开时返回的UDS TP Client的句柄 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @return D32S 关闭UDS TP通道的结果
 */
extern D32S oem_closeUdstp(D64U* pHandle);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pHandle	D64U*	input	non-nullptr	UDStp打开时返回的UDS TP Client的句柄
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.16 UA模块

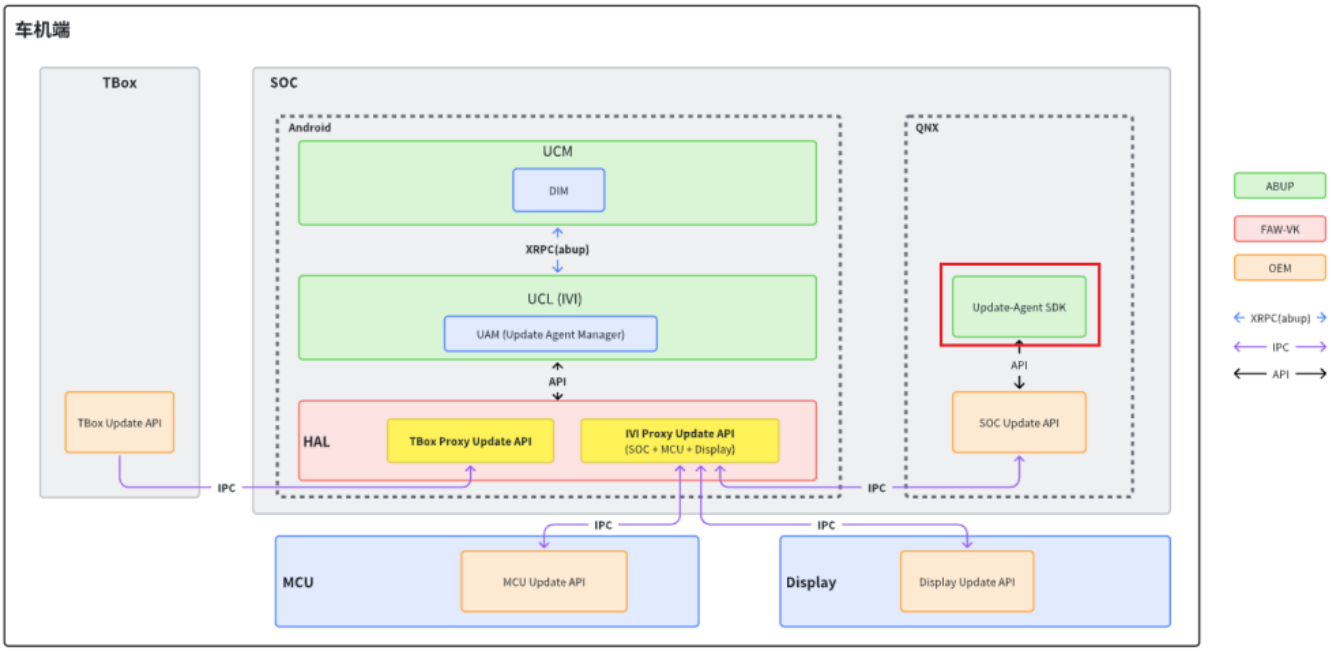
3.16.1 功能描述

该模块由一汽大众实现，主要功能：

1. 实现SOC的升级，包括整包升级和差分升级。
2. 反馈升级状态信息，包括升级进度、升级结果信息。

UA模块支持断电续升功能，当断电重启后Installer模块重新触发升级，传入升级包路径开始续升流程。在安装过程中，出现安装错误时UA模块退出安装流程并返回错误码给到Lite模块。

UA模块以so动态库的形式提供，由Lite模块触发升级刷写流程，UA模块升级过程中需要获取系统当前slot信息以及对升级分区（或升级文件）进行读写，因此需要IVI供应商提供获取分区slot信息接口和分区读写接口供UA模块使用。



3.16.2 业务时序

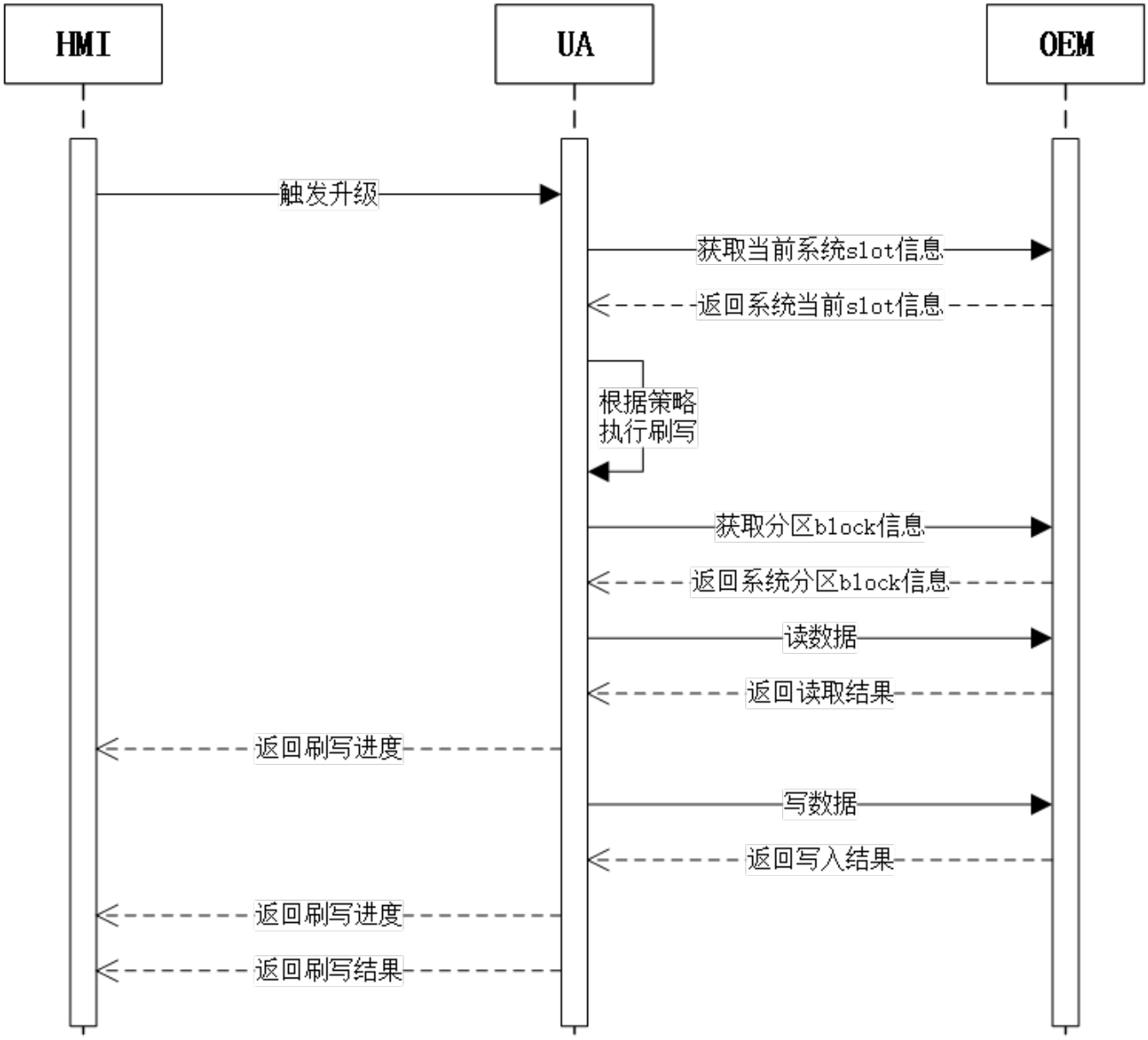


图 3-18 UA模块时序图

3.16.3 接口设计

3.16.3.1 获取系统当前slot信息

- **接口说明**
  - **调用方：**UA模块
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**UA模块升级前需获取系统当前slot ID信息，读取当前slot数据，更新数据到备份slot
- **接口原型**

```
/**
 * @brief 获取系统当前的slot ID信息
 *
 * @return OEM_SLOT_INFO_E 返回的系统当前分区结果
 */
extern OEM_SLOT_INFO_E oem_GetCurrentSlot();
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
返回值	OEM_SLOT_INFO_E	output	枚举	系统当前slot ID信息。 A_SLOT 表示当前系统在A slot， B_SLOT 表示当前系统在B slot， 其它值表示异常，该函数执行失败。

3.16.3.2 获取分区block信息

- **接口说明**
  - **调用方：**UA模块
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**UA模块更新分区时，读写分区前需获取分区block信息，用于后续读写接口调用
- **接口原型**

```
/**
 * @brief 获取系统分区block信息
 *
 * @param[in] pstVolCluster     描述: 分区描述结构体指针 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @param[out] pstBlockInfo     描述: 分区block信息，用于后续分区读写操作 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @return D32S 获取分区block信息结果
 */
extern D32S oem_getBlockInfo(OEM_VOLCUSTER_T* pstVolCluster, OEM_BLOCKINFO_T* pstBlockInfo);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pstVolCluster	OEM_VOLCUSTER_T*	input	non-nullptr	分区描述结构体指针
pstBlockInfo	OEM_BLOCKINFO_T*	output	non-nullptr	分区block信息，用于后续分区读写操作
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.16.3.3 读分区数据

- **接口说明**
  - **调用方：**UA模块
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**UA模块升级分区数据时，从分区节点读取数据
- **接口原型**

```
/**
 * @brief 读取系统分区数据
 *
 * @param[in] pstVolCluster     描述: 分区描述结构体指针 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @param[out] pBuf     描述: 用于读取数据的指针 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @param[in] pBuf     描述: 读取数据位置偏移量 \n
 * 范围值: >=0 \n
 * 单位: N/A  \n
 *
 * @param[in] pBuf     描述: 读取数据大小 \n
 * 范围值: >=0 \n
 * 单位: N/A  \n
 *
 * @return D32S 读取系统分区数据的结果
 */
extern D32S oem_readOtaUpdate
(
    OEM_VOLCUSTER_T* pstVolCluster,
void* pBuf,
size_t uiBlockOffset,
size_t uiBlockCounts
);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pstVolCluster	OEM_VOLCUSTER_T*	input	non-nullptr	分区描述结构体指针
pBuf	void*	output	non-nullptr	用于读取数据的指针
uiBlockOffset	size_t	input	>=0	读取数据位置偏移量，以block为单位

uiBlockOffset	size_t	input	>=0	读取数据大小，以block为单位
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.16.3.4 写分区数据

- **接口说明**
  - **调用方：**UA模块
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**UA模块升级分区数据时，将计算后的数据写入分区
- **接口原型**

```
/**
 * @brief 系统分区数据写入
 *
 * @param[in] pstVolCluster     描述: 分区描述结构体指针 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @param[in] pBuf     描述: 用于读取数据的指针 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @param[in] pBuf     描述: 读取数据位置偏移量 \n
 * 范围值: >=0 \n
 * 单位: N/A  \n
 *
 * @param[in] pBuf     描述: 读取数据大小 \n
 * 范围值: >=0 \n
 * 单位: N/A  \n
 *
 * @return D32S 系统分区数据写入的结果
 */
extern D32S oem_writeOtaUpdate
(
    OEM_VOLCUSTER_T* pstVolCluster,
void* pBuf,
size_t uiBlockOffset,
size_t uiBlockCounts
);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注

pstVolCluster	OEM_VOLCUSTER_T*	input	non-nullptr	分区描述结构体指针
pBuf	void*	input	non-nullptr	用于写入数据的指针
uiBlockOffset	size_t	input	>=0	读取数据位置偏移量，以block为单位
uiBlockOffset	size_t	input	>=0	读取数据大小，以block为单位
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.17 HMI模块

3.17.1 功能描述

车机交互操作

3.17.3 业务时序

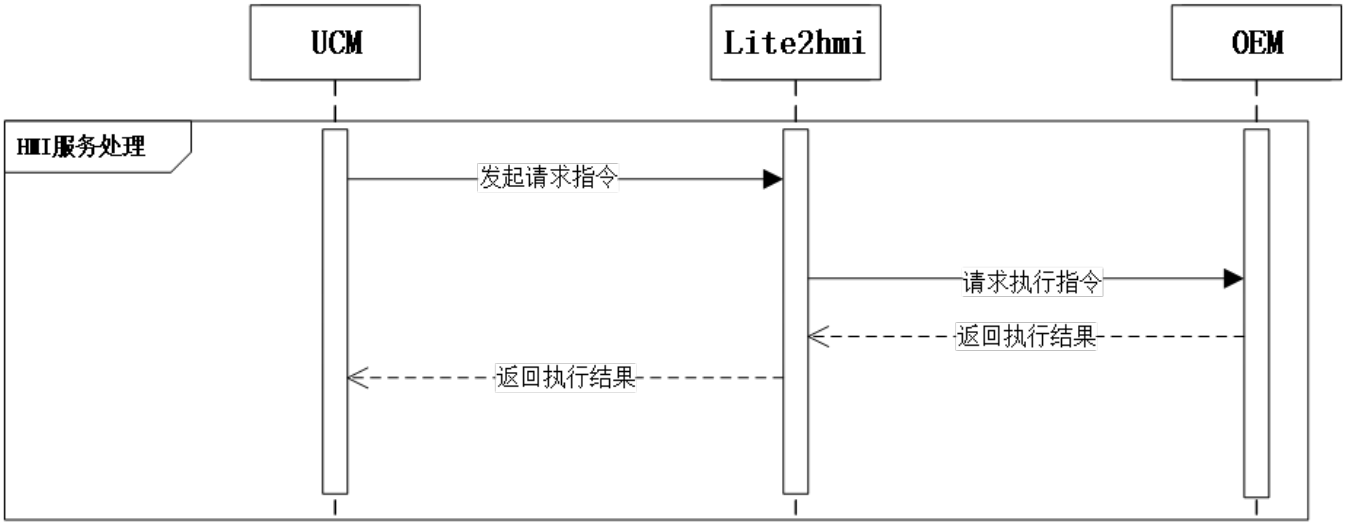


图 3-19 HMI模块时序图

3.17.2 接口设计

3.17.3.1 HMI小红点显示与隐藏

HMI通知HMI Service模块，在系统设置页面进行小红点的显示与隐藏。

- **接口说明**
  - **调用方：**HMI服务模块
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**当UC-Master检测到升级任务后，HMI通知Settings模块，Settings界面展示小红点；当升级任务已经被UC-Master模块处理后，HMI通知Settings模块，Settings界面隐藏小红点
- **接口原型**

```
/**
 * @brief 小红点展示设置
 *
 * @param[in] bShow     描述: 是否展示 \n
 * 范围值: >=0 \n
 * 单位: N/A  \n
 *
 * @return D32S 设置小红点展示设置的结果
 */
extern D32S oem_setRedDotShow(bool bShow);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
bShow	bool	input	[false, true]	显示/隐藏小红点的标志 true: 显示 false: 隐藏
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.17.3.2 获取车机当前用户语言

- **接口说明**
  - **调用方：**HMI服务模块
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**OTA轮播图、升级提示语等需根据当前车机的语言模式（中/英文）区分展**示**
- **接口原型**

```
/**
 * @brief 获取当前车机用户语言
 *
 * @param[out] pLanguage     描述: 获取的语言模式 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @return D32S 返回获取操作的结果
 */
extern D32S oem_getOtaLanguageMode(D32S* pLanguage);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注

pLanguage	D32S*	output	non-nullptr	返回车机当前的语言模式，返回值范围为 1：中文 2：英文 其他：预留语言 注：与供应商沟通协定一个默认主题
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.17.3.3 获取当前车机模式

- **接口说明**
  - **调用方：**HMI服务模块
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**HMI界面启动，需要获取车机当前主题来判断显示对应的界面，如OTA轮播图等需根据当前车机的白天/黑夜模式区分展示
- **接口原型**

```
/**
 * @brief 获取当前车机模式
 *
 * @param[out] pTheme     描述: 获取的车机模式 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @return D32S 返回获取当前模式的操作结果
 */
extern D32S oem_getOtaTheme(D32S* pTheme);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pTheme	D32S*	output	non-nullptr	返回车机当前切换后的主题，返回值范围为：[1,2,3] 1：主题1 2：主题2 3：主题3 注：与供应商沟通协定一个默认主题
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.17.3.4 车机霸屏模式控制

- **接口说明**
  - **调用方：**HMI服务模块
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**当OTA进入安装过程，为了防止用户操作屏幕影响升级，安装界面需要霸屏展示OTA界面，升级完成需要退出霸屏。
- **接口原型**

```
/**
 * @brief 霸屏模式设置
 *
 * @param[in] uiType     描述: 进入/退出霸屏模式 \n
 * 范围值: >0 \n
 * 单位: N/A  \n
 *
 * @return D32S 霸屏模式设置的结果
 */
extern D32S oem_controlOverScreen(D32U uiType);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
uiType	D32U	input	[1, 2]	霸屏控制的类型标识, 取值范围[1,2] 1：进入霸屏模式 2：退出霸屏模式
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.17.3.5 OTA应用入口

- **接口说明**
  - **调用方：**HMI服务模块
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**作为OTA应用入口，用户可以点击入口进入HMI程序查看流程或者进行人机交互。
- **接口原型**

```
OTA应用APK的启动信息如下：
包名（PackageName）：com.faw-vw.ota
启动Acitivty类名（ClassName）：.ui.activity.MainActivity
```

3.17.3.6 消息中心显示事件

通知消息中心的标题与内容，具体消息展示，通知栏的展示频率与次数，车机方根据接口的desc、taskID来实现。

- **接口说明**
  - **调用方：**HMI服务模块
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**当OTA有检测到新版本、下载完成、安装进度、安装结果等状态时，HMI需要通知消息中心进行消息展示
- **接口原型**

```
/**
 * @brief 通知中心显示事件
 *
* @param[in] pTitle     描述: 消息通知的消息标题 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @param[in] pDesc     描述: 消息通知的消息内容 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @param[in] pTaskID     描述: OTA的任务ID \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @return D32S 通知中心显示事件操作的结果
 */
extern D32S oem_showNotification(DCHAR* pTitle, DCHAR* pDesc, DCHAR* pTaskID);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pTitle	DCHAR*	input	non-nullptr	消息通知的消息标题, ASCII编码的字符串
pDesc	DCHAR*	input	non-nullptr	消息通知的消息内容, ASCII编码的字符串
pTaskID	DCHAR*	input	non-nullptr	OTA的任务ID
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.17.3.7 车辆设防状态获取

获取车辆是解防状态还是设防状态

- **接口说明**
  - **调用方：**HMI服务模块
  - **实现方：**车机供应商
  - **同步/异步：**同步接口，接口调用的超时时间为30秒
  - **业务场景：**
    - 设防定义：车门车窗关闭、主副驾无压力、蓝牙钥匙不在车内、车辆未启动
    - 用于判断车辆是否正在使用中，如升级前置条件不满足时需根据设防状态区分弹窗提示时机
    - 静默升级前需判断车辆处于设防状态
- **接口原型**

```
/**
 * @brief 获取当前车机设防状态
 *
 * @param[out] pStatus     描述: 获取的设防状态 \n
 * 范围值: 指针 \n
 * 单位: N/A  \n
 *
 * @return D32S 返回获取当前车机的设防状态操作结果
 */
extern D32S oem_getFortifyStatus(D32S* pStatus);
```

- **接口参数**

参数名称	数据类型	输入/输出	有效值范围	备注
pStatus	D32S*	output	non-nullptr	车辆挡位状态,取值范围为[1,3] 1: 解防状态 2: 设防状态 3: 获取状态异常
返回值	D32S	output	int	0:成功，否则返回具体错误码

3.18 数据类型定义

3.18.1 ADM_D2B_APPOINTMENT_TYPE_E

```
/**
 * @brief 预约类型
 *
 */
typedef enum ADM_D2B_APPOINTMENT_TYPE_
{
    D2B_APPOINTMENT_UPGRADE = 0x01,        ///< 预约升级
    D2B_APPOINTMENT_DIAGNOSTIC,            ///< 预约诊断
    D2B_APPOINTMENT_WAKEUP,                ///< 预约唤醒
    D2B_APPOINTMENT_RESERVE,               ///< 预约的预留类型
    D2B_APPOINTMENT_INVALID = 0xFF         ///< 无效类型
} ADM_D2B_APPOINTMENT_TYPE_E;
```

3.18.2 ADM_OEM_D2B_APPOINTMENT_ARRIVAL_NOTIFY

```
/**
 * @brief 预约定时器时间到达后，通过此回调通知调用者
 *
 */
typedef D32S (*ADM_OEM_D2B_APPOINTMENT_ARRIVAL_NOTIFY)(const ADM_D2B_APPOINTMENT_TYPE_E eType, DVOID* pPrivate);
```

3.18.3 ADM_PUB_ECU_ID_INFO_T

```
/**
 * @brief 零件ID信息
 *
 */
typedef struct
{
    DCHAR aRequestId[65];          ///< ECU物理请求地址
    DCHAR aSwDid[65];              ///< ECU升级对象软件版本
} ADM_PUB_ECU_ID_INFO_T;
```

3.18.4 ADM_UAM_VARIABLE_DATA_T

```
/**
 * @brief ecu零件信息采集(采用异步的方式,请求后会有事件通知)
 *
 */
typedef struct
{
    D16U uiDataType;                ///< 数据类型
    D16U uiDataLength;              ///< 数据长度
    DCHAR* pValue;                  ///< 数据值
} ADM_UAM_VARIABLE_DATA_T;
```

3.18.5 ADM_UAM_DID_DATA_T

```
/**
 * @brief 零件DID数据结构
 *
 */
typedef struct
{
    D32U uiDid;                         ///< did
    D32U uiStartByte;                   ///< 开始字节
    D32U uiEndByte;                     ///< 中止字节
    D32U uiErrorCode;                   ///< 响应码
    ADM_UAM_VARIABLE_DATA_T stDidData;     ///< did结构数据
} ADM_UAM_DID_DATA_T;
```

3.18.6 ADM_UAM_DID_DATA_ARRAY_T

```
/**
 * @brief 零件DID数组结构
 *
 */
typedef struct
{
    D32U uiDidArrayLength;              ///< did数组长度
    ADM_UAM_DID_DATA_T* pstDidArray;        ///< did数组
} ADM_UAM_DID_DATA_ARRAY_T;
```

3.18.7 ADM_UAM_ECU_COLLECT_REQ_T

```
/**
 * @brief 作为信息收集的请求以及事件通知的结构
 *
 */
typedef struct
{
    ADM_PUB_ECU_ID_INFO_T stEcuIdenInfo;     ///< 域ECU信息(eim进行零件信息收集,此时并没有升级策略信息)
    ADM_UAM_DID_DATA_ARRAY_T* pstDidList;    ///< ecu did列表信息
} ADM_UAM_ECU_INFO_COLLECT_T;
```

3.18.8 ADM_VSM_VARIABLE_T

```
/**
 * @brief 车辆状态管理结构
 *
 */
typedef struct ADM_VSM_VARIABLE_
{
    D16U uiType;            ///< 数据类别
    D16U uiLength;          ///< 数据长度
    union
    {
        D8U aui[1];
        D32U ui32;
        D32S i32;
        D64U ui64;
        D64S i64;
        DOUBLE dDouble;
        DCHAR a[1];
    } v;
} ADM_VSM_VARIABLE_T;
```

3.18.9 ADM_VSM_VEHICLE_SPY_TRIGGER_F

```
/**
 * @brief 监控触发回调函数
 *
 */
typedef DBOOL (*ADM_VSM_VEHICLE_SPY_TRIGGER_F)(D64U, ADM_VSM_RESULT_T*, D32U, const DVOID*);
```

3.18.10 ADM_VSM_SPY_VARIABLE_T

```
/**
 * @brief 使用可变的数据结构体的监控参数列表
 *
 */
typedef struct ADM_VSM_VARIABLE_
{
    D32U uiVehicleId;                    ///< vehicle状态、触发事件枚举值
    D32U uiCompareSymbol;                ///< 监控逻辑
    D32U uiMultiRelate;                  ///< 当存在多个监控message时，多个message的逻辑关系
    ADM_VSM_VARIABLE_T stFirstSpyData;   ///< 监控值：第1个参数必填
    ADM_VSM_VARIABLE_T stSeondSpyData;   ///< 监控值：第2个参数选填
    D32U uiEnumCount;                    ///< 枚举数量
    D32S aEnums[100];                    ///< 枚举值
    D32U uiPeriod;                       ///< 信号上报周期
} ADM_VSM_SPY_VARIABLE_T;
```

3.18.11 ADM_VSM_SPY_ARRAY_VARIABLE_T

```
/**
 * @brief 使用可变的数据结构体的监控参数列表
 *
 */
typedef struct ADM_VSM_VARIABLE_
{
    D32U uiMsgIdArrayLength;             ///< 监控数组长度
    ADM_VSM_SPY_VARIABLE_T* pstSpyArray;    ///< 监控数组
} ADM_VSM_SPY_ARRAY_VARIABLE_T;
```

3.18.12 ADM_UPGRADE_IMAGE_INFO_ATOM_T

```
/**
 * @brief 镜像原子对象
 *
 */
typedef struct
{
    D64U    uiImageSize;    ///< 文件大小
    DCHAR*  pImagePath;     ///< 文件路径,包含文件名
    DCHAR*  pImageHash;     ///< 文件hash值
} ADM_UPGRADE_IMAGE_INFO_ATOM_T;
```

3.18.13 ADM_UAM_UPDATE_REQ_T

```
/**
 * @brief 智能键升级请求结构
 *
 */
typedef struct
{
    ADM_PUB_ECU_ID_INFO_T stEcuIdenInfo;          ///< 域ECU信息
    ADM_UPGRADE_IMAGE_INFO_ATOM_T stImageInfo;    ///< 升级压缩包
} ADM_UAM_UPDATE_REQ_T;
```

3.18.14 ADM_PUB_ECU_FUNC_INFO_T

```
/**
 * @brief 作为信息收集的请求以及事件通知的结构
 *
 */
typedef struct
{
    ADM_PUB_ECU_ID_INFO_T stEcuIdInfo;        ///< ECUID信息
    DCHAR aResponseId[151];           ///< 诊断响应ID-扩展
    DCHAR aFunctionId[151];           ///< 诊断功能ID-扩展
} ADM_PUB_ECU_FUNC_INFO_T;
```

3.18.15 ADM_UPGRADE_PROGRESS_NOTIFY_T

```
/**
 * @brief 零件刷写进度结构
 *
 */
typedef struct
{
    ADM_PUB_ECU_ID_INFO_T stEcuIdenInfo;       ///< ECU信息
    D32U uiUpgradeProgress;                    ///< 升级/回滚进度
} ADM_UPGRADE_PROGRESS_NOTIFY_T;
```

3.18.16 FLASH_PROGRESS_CALLBACK_F

```
/**
 * @brief 刷写进度回调处理
 *
 */
typedef D32S (*FLASH_PROGRESS_CALLBACK_F)(ADM_UPGRADE_PROGRESS_NOTIFY_T* pstProgress);
```

3.18.17 ADM_UPGRADE_RESULT_INFO_T

```
/**
 * @brief 零件刷写结果信息
 *
 */
typedef struct
{
    D32S    iResult;              ///< 升级结果
    DCHAR* pExpectVal;            ///< 期望值
    DCHAR* pRealVal;              ///< 真实值
} ADM_UPGRADE_RESULT_INFO_T;
```

3.18.18 ADM_UPGRADE_RESULT_NOTIFY_T

```
/**
 * @brief 零件刷写结果结构
 *
 */
typedef struct
{
    ADM_PUB_ECU_ID_INFO_T stEcuIdenInfo;        ///< ECU信息
    ADM_UPGRADE_RESULT_INFO_T stUpgradeResultInfo;        ///< 升级/回滚结果
} ADM_UPGRADE_PROGRESS_NOTIFY_T;
```

3.18.19 FLASH_RESULT_CALLBACK_F

```
/**
 * @brief 刷写结果回调处理
 *
 */
typedef D32S (*FLASH_RESULT_CALLBACK_F)(ADM_UPGRADE_RESULT_NOTIFY_T* pstResult);
```

3.18.20 ADM_UAM_ACTIVE_TYPE_E

```
/**
 * @brief 激活类型
 *
 */
typedef enum
{
    ADM_ACTIVE_NO_REBOOT = 1,          ///< 直接激活,不需要进行重启
    ADM_ACTIVE_REBOOT                  ///< 重启激活
} ADM_UAM_ACTIVE_TYPE_E;
```

3.18.21 ADM_UAM_ACTIVE_REQ_T

```
/**
 * @brief 零件激活请求结构
 *
 */
typedef struct
{
    ADM_PUB_ECU_ID_INFO_T stEcuIdenInfo;        ///< 待激活ECU信息
    ADM_UAM_ACTIVE_TYPE_E eActiveType;          ///< 激活方式
} ADM_UAM_ACTIVE_REQ_T;
```

3.18.22 OEM_UDSTP_CONFIRM_E

```
/**
 * @brief 确认结果的类型值
 *
 */
typedef enum
{
    OEM_UDSTP_CONFIRM_OK = 0,         ///< OK
    OEM_UDSTP_CONFIRM_BUSOFF = 1, ///< OFF
    OEM_UDSTP_CONFIRM_DISCONNECT = 2, ///< 断开
    OEM_UDSTP_CONFIRM_NOTOPEN = 3,    ///< 没有打开
    OEM_UDSTP_CONFIRM_SENDFAIL = 4,   ///< 发送失败
    OEM_UDSTP_CONFIRM_CONNECT = 5,    ///< 连接成功
    OEM_UDSTP_CONFIRM_INVALID,
} OEM_UDSTP_CONFIRM_E;
```

3.18.23 OEM_SLOT_INFO_E

```
/**
 * @brief 系统当前slot ID信息
 *
 */
typedef enum
{
    A_SLOT = 0, ///< A slot
    B_SLOT = 1, ///< B slot
    INVALID_SLOT = -1 ///< 异常
} OEM_SLOT_INFO_E;
```

3.18.24 OEM_VOLCUSTER_T

```
/**
 * @brief 分区描述结构体指针
 *
 */
typedef struct
{
    const DCHAR* pVolName;;        ///< 卷名，例如“/dev/block/mmcblk0p1”
    size_t uiVolOffset;          ///< 卷偏移，从存储设备零地址的偏移，单位是存储设备块大小
} OEM_VOLCUSTER_T;
```

3.18.25 OEM_BLOCKINFO_T

```
/**
 * @brief block信息描述结构体
 *
 */
typedef struct
{
    size_t uiBlockSize;          ///< 块大小，必须与升级包的块大小相同，不一定与存储设备块大小相同
    size_t uiBlockCount;         ///< 分区中的块数
} OEM_BLOCKINFO_T;
```

3.18.26 AMD_NETWORK_STATE_E

```
/**
 * @brief 当前4G 网络状态
 *
 */
typedef enum
{
    ADM_NETWORK_ON,
    ADM_NETWORK_OFF,
    ADM_NETWORK_INVAILD
} AMD_NETWORK_STATE_E;
```

3.18.27 AMD_GET_VEHICLE_STATE_T

```
/**
 * @brief 车辆状态信息结构
 *
 */
typedef struct
{
    D32U uiCarSpeed;                    ///< 车速
    D32U uiEngineSpeed;                 ///< 发动机转速
    D32U uiKeyState;                    ///< 钥匙状态
    D32U uiGears;                       ///< 档位
    D32U uiHandbrakeState;              ///< 手刹状态
    D32U uiChargeState;                 ///< 充电状态
    D32U uiOBDState;                    ///< OBD状态
    D32U uiBatteryVolMv;                ///< 蓄电池电压
    D32U uiPowerBatteryRatio;           ///< 动力电池电量百分比
    D32U uiAssistedDriving;             ///< 辅助驾驶状态
    D32U uiDischargeState;              ///< 放电状态
} AMD_GET_VEHICLE_STATE_T;
```

3.18.28 AMD_GET_VEHICLE_STATE_E

```
/**
 * @brief 车辆状态类型
 *
 */
typedef enum
{
    ADM_VEHICLE_STATE_CARSPEED,                 ///< 车速
    ADM_VEHICLE_STATE_ENGINESPEED,              ///< 发动机转速
    ADM_VEHICLE_STATE_KEYSTATE,                 ///< 钥匙状态
    ADM_VEHICLE_STATE_GEARSTATE,                ///< 档位
    ADM_VEHICLE_STATE_HANDBRAKE,                ///< 手刹
    ADM_VEHICLE_STATE_CHARGE,                   ///< 充电状态
    ADM_VEHICLE_STATE_OBDSTATE,                 ///< OBD状态
    ADM_VEHICLE_STATE_BATTERYVOL,               ///< 电池电压
    ADM_VEHICLE_STATE_POWERBATTERY,             ///< 动力电池电量百分比
    ADM_VEHICLE_STATE_ASSISTEDDRIVING,          ///< 辅助驾驶状态
    ADM_VEHICLE_STATE_DISCHARGESTATE,           ///< 放电状态
    ADM_VEHICLE_STATE_INVALID
} AMD_GET_VEHICLE_STATE_E;
```

3.18.29 ADM_UPGRADE_IMAGE_INFO_ATOM_T

```
/**
 * @brief 回滚文件对象
 *
 */
typedef struct
{
    D64U    uiImageSize;    ///< 文件大小
    DCHAR*  pImagePath;     ///< 文件路径,包含文件名
    DCHAR*  pImageHash;     ///< 文件hash值
} ADM_UPGRADE_IMAGE_INFO_ATOM_T;
```

3.18.30 ADM_UAM_ROLLBACK_REQ_T

```
/**
 * @brief 回滚结构
 *
 */
typedef struct
{
    ADM_PUB_ECU_ID_INFO_T stEcuIdenInfo;        ///< 待回滚ECU信息
    ADM_UPGRADE_IMAGE_INFO_ATOM_T* pstImageInfo;          ///< 回滚压缩包
} ADM_UAM_ROLLBACK_REQ_T;
```

4 接口验证

1.上述定义的OTA所需接口文档和头文件，IVI供应商根据接口文档和头文件来实现功能。

2.一汽大众提供接口验证demo工具和使用说明，以协助供应商进行自测验证，从而减少双方不必要的联调和沟通成本，提高效率并确保项目实施顺利。

5 系统依赖

5.1 开源库依赖

软件开发依赖的第三方开源组件库依赖情况如下：

开源组件			许可证		大小
名称	版本	下载链接	类型	链接	库
openssl	1.1.1n	https://www.openssl.org/source/old/1.1.1/openssl-1.1.1n.tar.gz	GNU	https://github.com/openssl/openssl/blob/OpenSSL_1_1_1n/LICENSE	3.64MB（静态库）
curl	7.88.1	https://github.com/curl/curl/releases/download/curl-7_88_1/curl-7.88.1.tar.gz	MIT	https://github.com/curl/curl/tree/curl-7_88_1/LICENSES	1000KB（静态库）
leveldb	1.22	https://github.com/google/leveldb/archive/refs/tags/1.22.tar.gz	BSD	https://github.com/google/leveldb/blob/1.22/LICENSE	989KB（静态库）
iniparser	4.1	https://github.com/ndevilla/iniparser/archive/refs/tags/v4.1.tar.gz	MIT	https://github.com/ndevilla/iniparser/blob/v4.1/LICENSE	12KB（静态库）
EasyLogger	2.2.0	https://github.com/armink/EasyLogger/archive/refs/tags/2.2.0.tar.gz	MIT	https://github.com/armink/EasyLogger/blob/2.2.0/LICENSE	22KB（静态库）
cJSON	1.7.15	https://github.com/DaveGamble/cJSON/archive/refs/tags/v1.7.15.tar.gz	MIT	https://github.com/DaveGamble/cJSON/blob/v1.7.15/LICENSE	27KB（静态库）
zlib	1.2.11	https://github.com/madler/zlib/archive/refs/tags/v1.2.11.tar.gz	GNU	https://github.com/madler/zlib/blob/v1.2.11/README	115KB（静态库）

civetweb	1.13	https://github.com/civetweb/civetweb/archive/refs/tags/v1.13.tar.gz	MIT	https://github.com/civetweb/civetweb/blob/master/LICENSE	206KB （静态库）

上述所选用的第三方开源组件已在我司应用多年，并且在广汽、长安、赛力斯等多个项目中应用，有厚重的技术积累和应用经验，也是业内的主流方案。在稳定性、易用性、扩展性、性能、成本、复杂度，安全性等方面都有优秀的表现，故选择该三方开源组件库使用。

5.2 系统与编译链要求

基于立项信息，本项目中OTA软件输出物（OTA-APP程序、Lite-APP程序、Lite2hmi程序以及HMI和UA中的动态库），基于下表中的交叉编译链编译生成。如果编译链信息有变更，需要走需求变更流程。

系统(System)	版本(Version)	编译链名称	使用方(user)
Android	~~Android 11~~ 待定	~~android-ndk-r23b~~ 待定	ABUP
QNX	待定	待定	ABUP

基于立项信息，本项目中OTA的HMI的软件输出物（Android APK），基于下表中车机系统系统信息生成。如果车机系统类型、车机系统版本、车机系统签名文件有变更，需要走需求变更流程。

系统(System)	版本号(Version)	系统签名文件	使用方(user)
Android	Android 11	车机供应商提供（不可变更）	一汽大众

5.3 操作系统要求

车机操作系统需要支持以下能力：

- 支持标准POSIX接口；
- 支持BSD的UNIX文件操作集合，包括读、写、创建文件或创建目录接口；
- 支持BSD API类型的SOCKET编程；
- 支持多线程接口编程。

5.4 文件系统要求

文件系统的函数接口需要支持两类接口，举例如下：

标准C接口 ANSI C	FILE *fopen(const char *path, const char *mode);
	int fclose(FILE *stream);
	size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream);
	size_t fwrite(const void *ptr, size_t size, size_t nmemb, FILE *stream);
POSIX	int open(const char *pathname, int flags, mode_t mode);
	int close(int fd);
	ssize_t write(int fd, const void *buf, size_t count);
	ssize_t read(int fd, void *buf, size_t count);
	void sync(void); int fsync(int fd);

5.5 Socket

ICC需要支持TCP/IP协议栈，TCP/IP的功能需支持POSIX接口，举例如下

int socket(int domain, int type, int protocol);
int accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen);
int bind(int sockfd, const struct sockaddr *addr, socklen_t addrlen);
int listen(int sockfd, int backlog);
ssize_t recv(int sockfd, void *buf, size_t len, int flags);
ssize_t send(int sockfd, const void *buf, size_t len, int flags);
int connect(int sockfd, const struct sockaddr *addr, socklen_t addrlen);

5.6 TLS

TLS需要支持完整的openssl API

5.7 调试要求

主要包括程序在开发调试阶段，软件对车机系统的一些调试要求如下：

- 车机系统支持开启ADB调试选项，以便通过USB接口连接至计算机进行调试；
- 调试设备车机具备可用的网络连接，若需要相关联网步骤，请提前告知；
- 车机系统支持开启Root权限，以便进行一些高级系统调试操作；
- 车机系统支持ADB Remount操作，方便在调试阶段通过ADB命令将升级包Push到设备系统目录；
- 车机系统开启日志记录功能，以便在系统调试阶段更好地跟踪问题和分析错误；
- 系统支持SeLinux权限的打开和关闭操作；
- 提供系统刷机方法以及设备线束，以防止调试过程中设备异常无法恢复；

提供两个及以上固件版本包，用于制作升级包进行升级测试；

5.8 存储要求

宿主ECU供应商需根据OTA业务需求，分配文件存储空间，用于缓存从OTA管理平台下载的升级文件。存储空间还需符合如下要求：

- 预留存储空间由ECU供应商、车厂和艾拉比根据实际业务情况共同协商定义；
- 建议有独立分区，该分区被OTA应用独占使用，用于存放升级包；
- 该存储空间具备一定的安全防护条件，确保数据库、配置文件、升级文件不被用户篡改，或者被第三方窃取；

5.9 网络要求

车机系统需要满足以下要求，以提供OTA系统程序的上网能力：

- 网络权限：OTA应用需要获得操作系统或用户的网络访问权限，以允许程序进行网络连接和数据传输。
- 白名单要求：需要提前告知，并将OTA平台的IP地址加入到设备白名单中。
- 网络限速要求：为了确保车云之间数据传输的效率，车机对于OTA应用不做限速处理。
- 车机需要有对车云通讯证书的下发，存储，更新，维护等管理能力，并提供接口和说明文档供OTA程序在OTA过程中使用

5.10 程序运行要求

5.10.1 OTA-APP运行要求

OTA-APP模块相关的性能参数如下，供车机供应商参考，如车机性能无法满足，请及时与一汽大众沟通。

配置参数	参数要求	备注
RAM	35MB~45MB	软件运行时占用大小
ROM	30MB	软件自身占用空间大小
CPU	6000DMIPS	基于8155芯片设备测试

5.10.2 Lite-App运行要求

Lite-APP模块相关的性能参数如下，供车机供应商参考，如车机性能无法满足，请及时与一汽大众沟通。

配置参数	参数要求	备注
RAM	25MB~35MB	软件运行时占用大小
ROM	30MB	软件自身占用空间大小
CPU	6000DMIPS	基于8155芯片设备测试

5.10.3 HMI运行要求

HMI模块相关的性能参数如下，供车机供应商参考，如车机性能无法满足，请及时与一汽大众沟通。

配置参数	参数要求	备注
RAM	100MB~160MB	若HMI设计时包含动画视频等显示，需要提高参数
ROM	30MB	软件自身占用空间大小
CPU	7000DMIPS	基于8155芯片设备测试

5.10.4 UA模块运行要求

UA模块相关的性能参数如下，供车机供应商参考，如车机性能无法满足，请及时与一汽大众沟通。

配置参数	参数要求	备注
RAM	50MB	若采用block方式升级，则内存消耗额外增加200MB左右
ROM	3MB	软件自身占用空间大小
CPU	7000DMIPS	基于8155芯片设备测试

5.11 OTA权限要求

以下为本模块用到的权限，需要注意的是，车机系统供应商需要确保应用在集成到车机之后就已经拥有了以下权限，基于OTA的产品特性，OTA应用不做程序运行时动态申请权限的操作。

5.11.1 UCM-APP文件目录权限要求

目录(Directory)	权限(Permission)	备注(Notes)
/system/bin/admota/ucm/ /system/lib64/ /system/etc/admota/ucm 具体需要与宿主件沟通	R	系统集成路径，包括可执行程序，动态库路径，配置文件路径。程序以ROOT权限启动。
/data/fawvk/admota/ucm/ 具体需要与宿主件沟通	RW	OTA程序运行路径，包括日志、数据库和下载的文件。

5.11.2 Lite-APP文件目录权限要求

目录(Directory)	权限(Permission)	备注(Notes)
/system/bin/admota/lite /system/lib64/ /system/etc/admota/lite 具体需要与宿主件沟通	R	系统集成路径，包括可执行程序，动态库路径，配置文件路径。程序以ROOT权限启动。
/data/fawvk/admota/lite/ 具体需要与宿主件沟通	RW	OTA程序运行路径，包括日志、数据库和下载的文件。

5.11.3 HMI目录权限要求

目录(Directory)	权限(Permission)	备注(Notes)
/sdcard/usb/ 路径可按照车机的规划做修改	RW	USB 接入后，U盘内文件的存储路径 注：路径可按照车机的规划做修改
/system/app/FAW-VK-Ota/ HMI APK 安装目录及缓存目录	RW	HMI集成及运行过程中的信息存储目录

5.11.4 UA模块权限要求

权限名称(Permission Name)	权限说明(Permission Description)	使用场景(Usage Scenario)
/**/abupi/	RW	升级过程用于保存缓存文件以及日志文件
待升级分区设备节点	RW	升级过程中对于升级分区节点进行读写操作

5.11.5 SeLinux权限调试要求

1. 针对开启了SeLinux权限的设备，一汽大众将相关的Te文件供车机供应商集成。
2. 由于Android系统版本的迭代、车机厂商对系统定制的确定性以及SeLinux权限修改后验证的独特性，一汽大众的Te文件可能存在无法覆盖所有权限的场景，调整权限的过程中，需车机供应商协助，诸如编译固件等工作，在此表示歉意和感谢。

OTA相关的Selinux的文件（Te文件），一汽大众作为发版的附件提供给车机供应商

5.11.6 OTA SDK自升级要求

车机提供APK自升级服务和权限

5.12 自启动要求

5.12.1 UCM-APP自启动要求

1. UCM-APP模块的相关文件在指定路径，路径要求在存储要求集成方案中。
2. OTA系统UCM-APP程序，采用独立的进程启动，需要车机实现UCM-APP程序的自启动。
3. 为了确保OTA应用程序在系统启动时能够自动运行，车机供应商应将我们启动脚本添加到系统启动的rc文件中，并以root权限拉起。
4. UCM-APP需要作为系统应用常驻到后台，因此当某种原因程序crash或者被系统进程kill掉，UCM-APP进程需要由车机系统的守护进程重新拉起。

5.12.2 Lite-APP自启动要求

1. Lite-APP模块的相关文件在指定路径，路径要求在存储要求集成方案中。
2. OTA系统Lite-APP程序，采用独立的进程启动，需要车机实现Lite-APP程序的自启动。
3. 为了确保OTA应用程序在系统启动时能够自动运行，车机供应商应将我们启动脚本添加到系统启动的rc文件中，并以root权限拉起。
4. Lite-APP需要作为系统应用常驻到后台，因此当某种原因程序crash或者被系统进程kill掉，Lite-APP进程需要由车机系统的守护进程重新拉起。

5.12.3 HMI自启动要求

- HMI APK 开机自启动

OTA需求需要HMI开机自启动，因此车机供应商需要支持HMI作为独立进程开机正常启动，且不能重复拉起。

- HMI启动入口

HMI需要支持用户手动点击进入应用进行人机交互，因此HMI需要与车机供应商进行入口集成。

- HMI进程守护要求

HMI需要作为系统应用常驻到后台，因此当某种原因程序crash或者被系统进程kill掉，HMI应用需要由车机系统的守护进程重新拉起。

5.13 系统签名要求

由于HMI模块以及UA模块在产品流程中涉及Notification bar、开机自启动、获取系统属性等要求，都需要系统签名才能解决权限问题，因此车机供应商需要在OTA开发前释放系统签名文件或者签名工具给到一汽大众集成。

6 集成方案

6.1 集成交付输出物

交付模块	交付物名称	交付物类型	交付物说明
UCM-APP	admUcmApp	可执行程序	UCM-APP进程
	ucmconfig.ini	配置文件ini	UCM-APP进程启动的配置信息文件 配置文件的使用使得应用程序的设置和选项可以在不改变源代码的情况下灵活地进行调整和修改
	libOtaClient.so	动态库	负责车云通信交互的库（FAW-VK实现）
	libarpc.so	动态库	ABUP负责进程间通信的库（主从控通信库）
	libcrypto.so.1.1	动态库	负责安全相关库
	libssl.so.1.1	动态库	提供网络通信中进行安全加密和认证的功能
	libmisc.so	动态库	进程需要的相关基础库
	libfoundation.so	动态库	进程需要的相关基础库
	admUcmApp.te	te文件	SeLinux权限添加使用，需要车机编译到系统中。
Lite-APP	admLiteApp	可执行程序	Lite-APP进程
	liteconfig.ini	配置文件ini	Lite-APP进程启动的配置信息文件 配置文件的使用使得应用程序的设置和选项可以在不改变源代码的情况下灵活地进行调整和修改
	libarpc.so	动态库	ABUP负责进程间通信的库（主从控通信库）
	libcrypto.so.1.1	动态库	负责安全相关库
	libssl.so.1.1	动态库	提供网络通信中进行安全加密和认证的功能
	Libmisc.so	动态库	进程需要的相关基础库
	libcore.so	动态库	诊断刷写引擎库
	libfoundation.so	动态库	进程需要的相关基础库
	admLiteApp.te	te文件	SeLinux权限添加使用，需要车机编译到系统中。
	依赖的宿主ECU相关库	动态库	依赖的宿主ECU实现的相关接口库文件
HMI APK	OTAHMI.apk 具体名称待定	apk	OTA从控程序，负责为UC-Master提供所需的系统能力，实现人机交互和HMI界面展示的android应用程序安装包
UA	sample	可执行程序及源码	升级示例程序及参考源码
	libabinstaller.so	动态库	核心升级库，实现升级能力
	cfgjson.txt	json配置文件	参考升级策略文件

	mkotapackage	做包工具，可执行文件	Linux系统下可执行文件，用于制作升级包
	《Installer集成操作手册-mkotapackage》 《Installer集成操作手册-Installer解》	文档	集成指导手册

6.2 集成方案

6.2.1 OTA-APP集成方案

1. UCM-APP提供集成相关文件，具体文件参照集成交付输出物集成交付输出物；
2. 为了确保程序能够被系统正确识别和执行，可执行程序和启动脚本放在/system/bin/admota/ucm/路径；
3. 配置文件通常包含程序运行所需的设置和参数需放在/system/etc/admota/ucm/路径下；
4. OTA应用程序所需的动态链接库应放在/system/lib64/路径下以便程序在运行时可调用；
5. 为了确保OTA应用程序在系统启动时能够自动运行，车机供应商应将我们启动脚本添加到系统启动的rc文件中，并以root权限拉起。

6.2.2 Lite-APP集成方案

1. Lite-APP提供集成相关文件，具体文件参照集成交付输出物集成交付输出物；
2. 为了确保程序能够被系统正确识别和执行，可执行程序和启动脚本放在/system/bin/admota/lite/路径；
3. 配置文件通常包含程序运行所需的设置和参数需放在/system/etc/admota/lite/路径下；
4. OTA应用程序所需的动态链接库应放在/system/lib64/路径下以便程序在运行时可调用；
5. 为了确保OTA应用程序在系统启动时能够自动运行，车机供应商应将我们启动脚本添加到系统启动的rc文件中，并以root权限拉起。

6.2.3 HMI集成方案

1. 车机供应商提供系统签名文件；
2. 一汽大众对OTA的HMI应用进行系统签名；
3. 一汽大众提供签名后的OTA的HMI安装包给车机供应商；
4. 车机供应商将OTA的HMI安装包集成至车机Android系统的/system/app/FAW-VK-Ota/路径下；
5. 车机供应商将OTA的HMI应用添加到应用白名单中，以确保OTA的HMI进程不会因系统资源（如内存）紧张等原因被系统回收，从而避免导致OTA的HMI应用无法正常使用。

6.2.4 UA集成方案

1. 差分升级模块以动态库形式提供，提供给到零件供应商集成适配。

6.3 版本迭代

考虑到OTA信息安全、车机系统安全等因素，与OTA相关的软件不能自行进行升级，而必须跟随系统一起进行升级。

在项目进入联调和车辆量产等阶段时，如果出现OTA程序异常或客户需求变更等原因，导致OTA软件需要更新迭代，车机供应商需要进行配合，进行OTA相关软件系统集成发版。

7 参照文件(Reference)

序号	文件名	版本号	存储位置
1	J01-OTA 车机端PRD	V0.2	06. J01-OTA 车机端PRD - Jetta BEV - Confluence (vc.in)
2	OTA车端软件详细设计	V2.0	1040604 OTA车端软件详细设计 - Jetta BEV - Confluence (vc.in)
3	OTA安全方案	V2.0	1040605 OTA安全方案 - Jetta BEV - Confluence (vc.in)
4	OTA车云交互设计	V1.0.1	1040602 OTA车云交互设计 - Jetta BEV - Confluence (vc.in)
5	差分升级设计	V1.3	1040606 差分升级设计 - Jetta BEV - Confluence (vc.in)
6			

8 接口头文件

oem_interface.h