

10 DoIP协议栈详细设计

- 1 文档介绍 Introduction
 - 1.1 范围说明 Scope
 - 1.2 参考文档 References
 - 1.3 读者对象 Target Audience
 - 1.4. 术语和缩写 Definitions and Abbreviations
- 2 架构设计 Architecture Design
- 3 静态设计 Static Design
 - 3.1 详细模块/组件框图
- 4 动态设计 Dynamic Design
 - 4.1 业务流程图
 - 4.1.1 整体流程
 - 4.1.2 Uds On IP流程
 - 4.1.2.1 预编程
 - 4.1.2.2 编程执行
 - 4.1.2.3 后编程
 - 4.1.3 Uds On CAN流程
 - 4.2 业务时序图
 - 4.2.1 DoIP Handler相关时序
 - 4.2.2 UDS实例刷写时序
- 5 接口设计
 - 5.1 外部接口
 - 5.2 内部接口
- 6 单元测试

Version / 版本					

1 文档介绍 Introduction

1.1 范围说明 Scope



1.2 参考文档 References

Ducment Name / 文档名称	Date / 日期	Version / 版本	Location / 位置

1.3 读者对象 Target Audience

系统工程师，开发人员，测试人员，技术支持人员，及其他相关人员。

1.4. 术语和缩写 Definitions and Abbreviations

本章节对本文中使用的术语定义和缩写进行解释。

Definitions or Abbreviations / 术语或缩写	Explanation / 解释

2 架构设计 Architecture Design

按照刷写的需求，我们将DoIP协议栈功能分为三层：

对外接口层：DoIP模块对外提供的功能接口，主要为业务提供OTA的启停、OTA刷写进度查询、OTA刷写错误查询功能。

模块处理层：DoIP模块对于OTA的主流程的状态控制处理，其中包含有DoIP节点本身的刷写处理与DoIP节点作为边缘节点时的刷写处理，还有刷写过程中的UDS诊断的通用服务处理。

传输管理层：负责DoIP模块中对Socket资源的创建、监听、释放等操作，同时处理诊断通道的连接处理（路由激活）。



3 静态设计 Static Design

3.1 详细模块/组件框图

DoIP协议栈主要分为5个模块，分别为主控，UDS实体，DoIP实体，UDS通用服务处理，DoIP协议处理

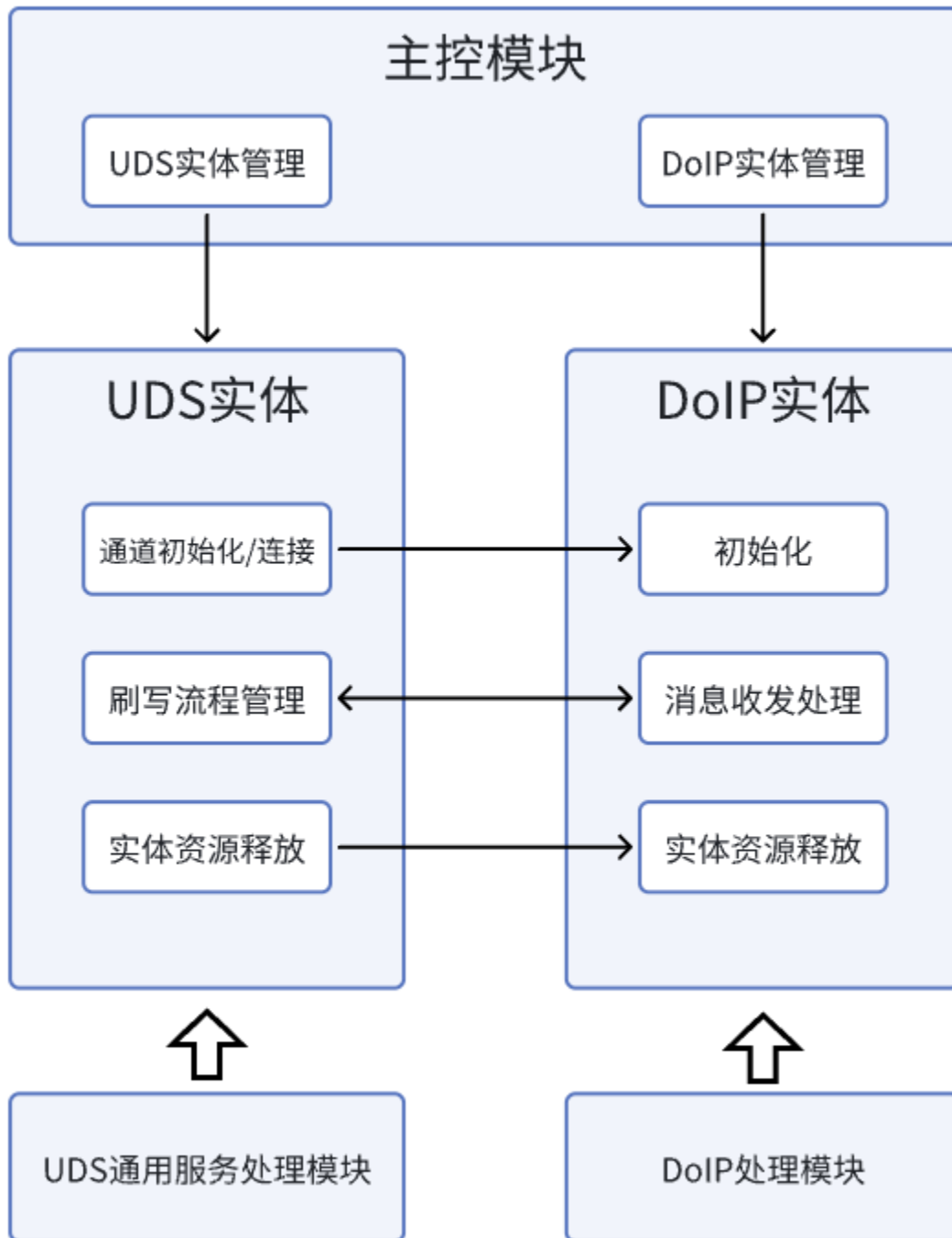
主控：负责对上次服务提供刷写接口，并对整体的刷写流程进行管理，维护UDS实体与DoIP实体的生命周期

UDS实体：进行对ECU的刷写流程控制，包含直连节点升级与下挂节点升级流程

DoIP实体：对UDS实体提供相关的接口，包含DoIP连接，DoIP去连接，DoIP消息发送，DoIP消息接收等接口

UDS通用服务处理：封装标准的UDS服务处理接口供刷写流程使用

DoIP协议处理：负责DoIP底层协议栈的处理，提供诊断通道初始化、去初始化、诊断消息发送的接口，具体包含Socket的创建、连接、监听消息处理等操作



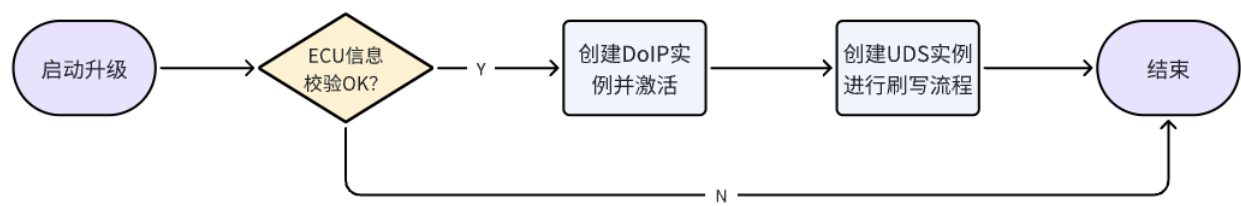
4 动态设计 Dynamic Design

4.1 业务流程图

4.1.1 整体流程

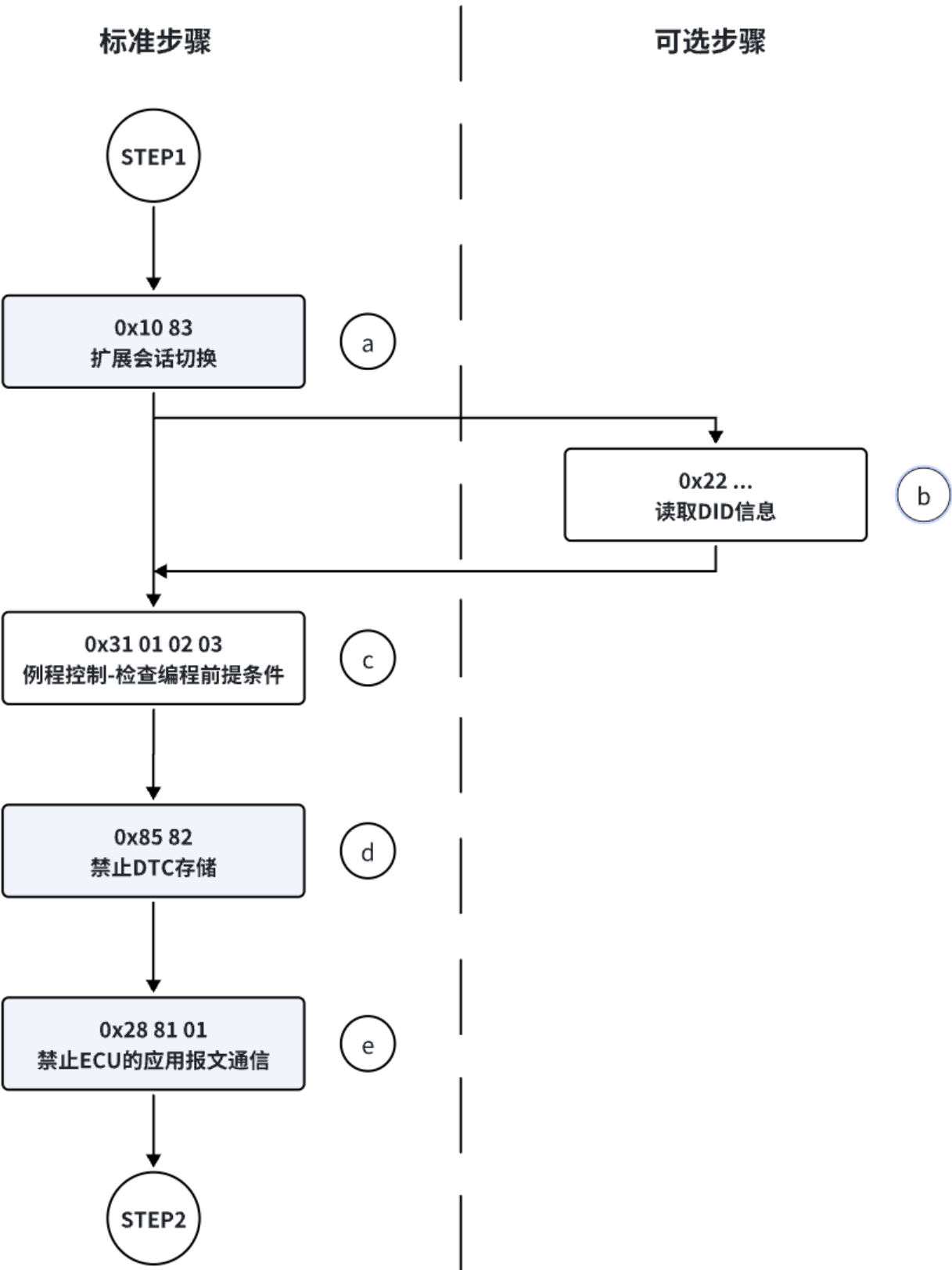
整体刷写流程如下所示，DoIP模块在接受到上层服务的开始升级请求后，依次创建DoIP实例与UDS刷写实例，执行升级操作。

PS：初版只支持同时存在一个DoIP实例与UDS实例，即不支持并行刷写场景



4.1.2 Uds On IP流程

4.1.2.1 预编程



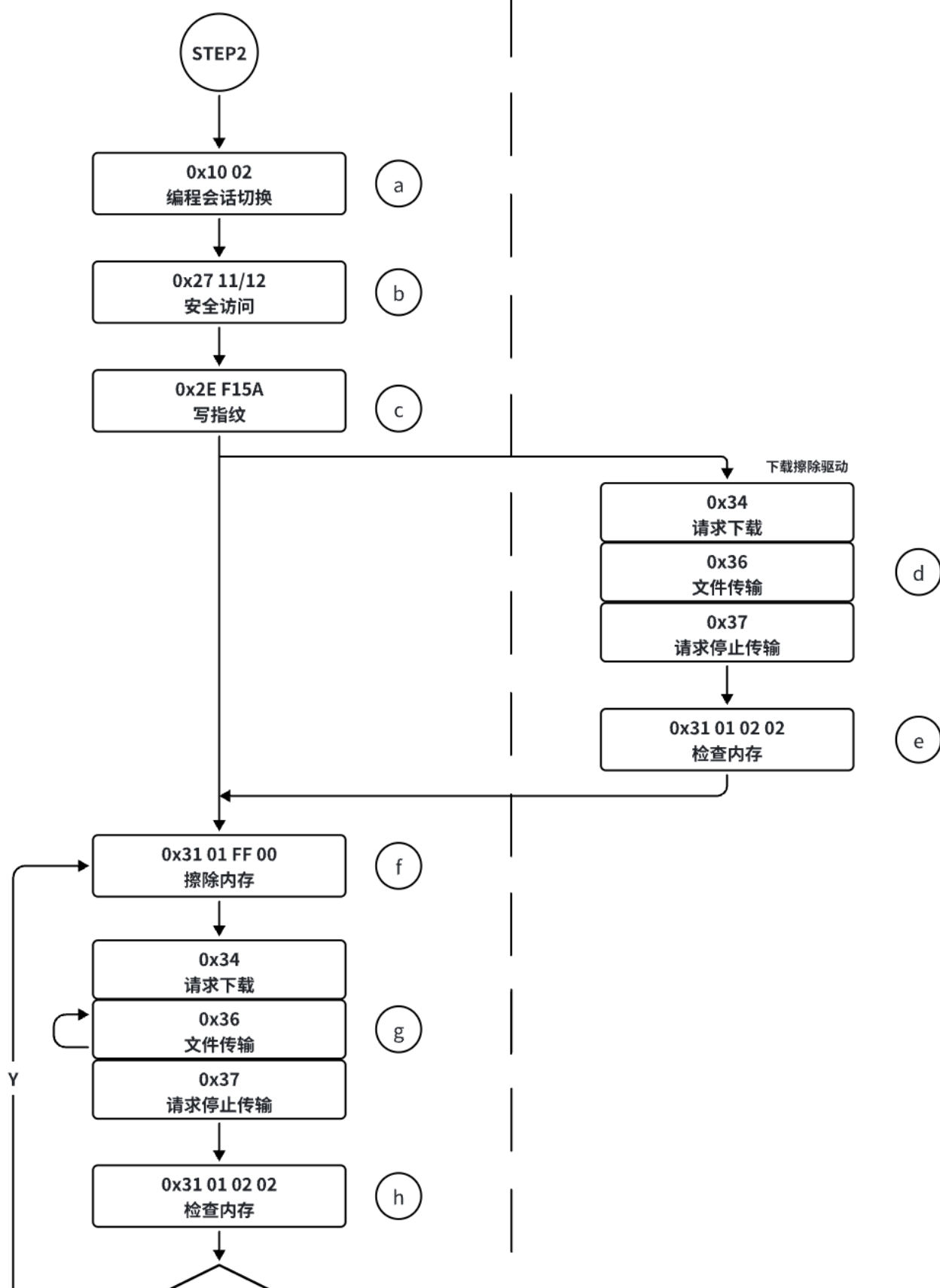
步骤c中，ECU返回响应 71 01 02 03则表示编程前提条件均满足

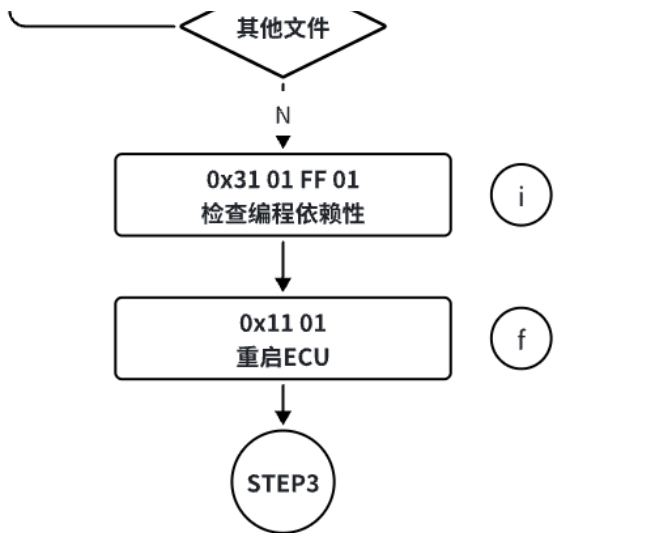
4.1.2.2 编程执行

1. 嵌入式ECU

标准步骤

可选步骤

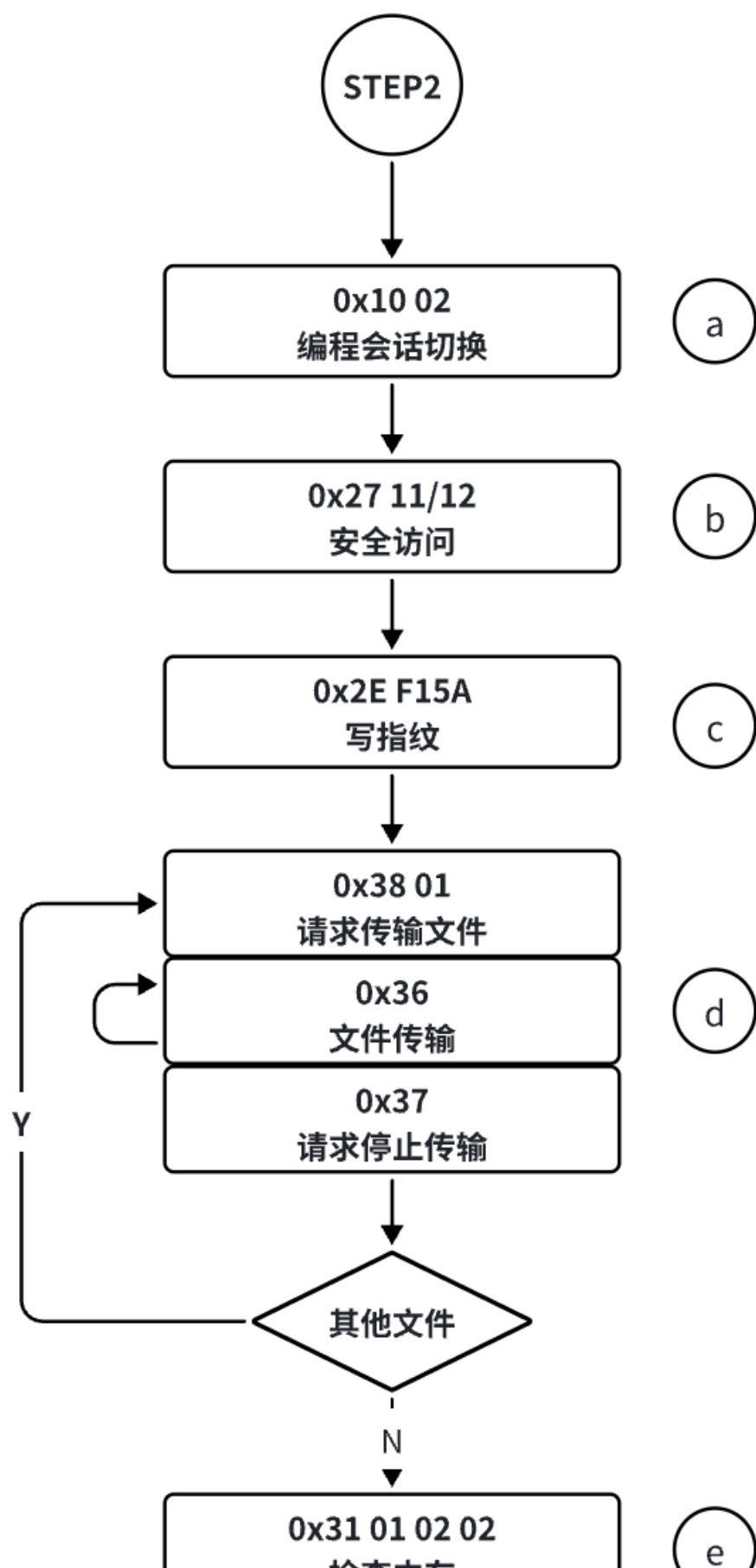


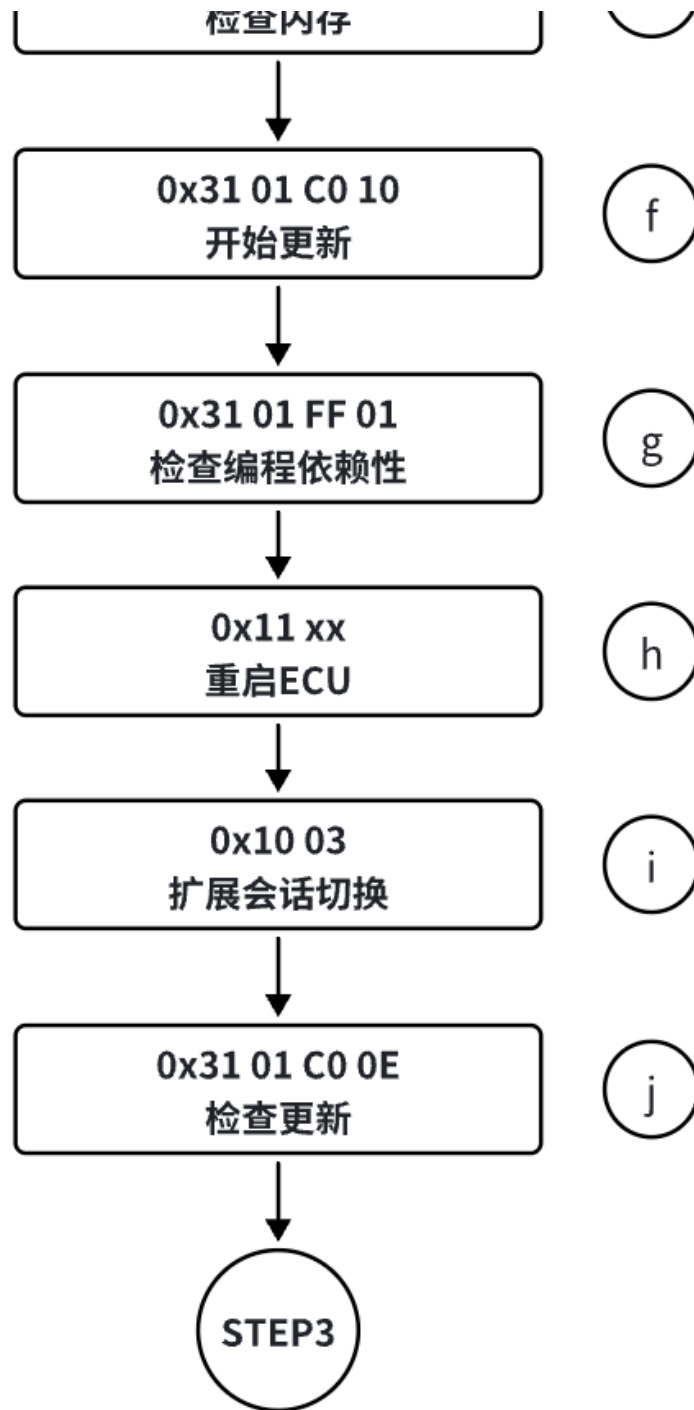


步骤e, f中, 例程的请求数据携带有CRC32、签名或者二者的组合, 响应结果为00则表示内存检查成功

步骤f目前存疑, 是否需要跟着gh循环每次都执行擦除, 从规范描述讲是不需要的

2. 非嵌入式ECU





步骤e中，例程的请求数据携带有CRC32、签名或者二者的组合，响应结果为00则表示内存检查成功

步骤f中，例程的请求数据要向ECU发送作为更新执行的安装包的逻辑块地址信息以启动软件更新

步骤g中，00代表正确结果，01代表不正确结果

4.1.2.3 后编程

功能寻址

物理寻址

标准步骤

STEP3

0x10 83
扩展会话切换

a

0x28 80 01
恢复ECU的应用报文通信

b

0x85 81
恢复DTC存储

c

0x10 01
默认会话切换

d

0x14 FF FF FF
清除诊断信息

e



4.1.3 Uds On CAN流程

同UDS on IP流程的预编程，编程执行的嵌入式ECU，后编程

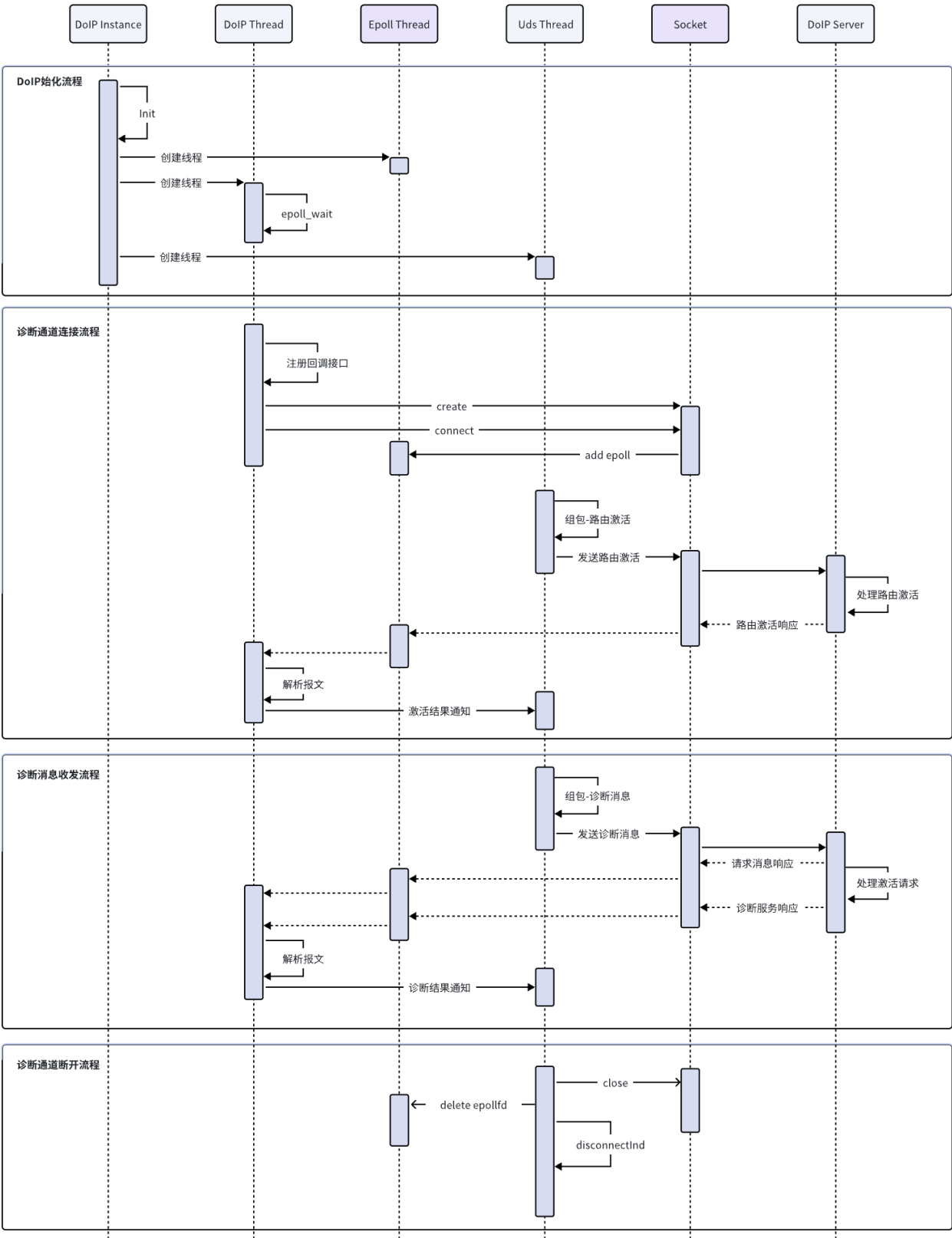
4.2 业务时序图

4.2.1 DoIP Handler相关时序

DoIP实体与DoIP Handler之间主要是通过Handler提供的接口和回调函数进行DoIP在协议栈层面的实现，主要涉及DoIP初始化、诊断通道初始化、诊断消息收发、诊断通道去初始化流程。

1. DoIP初始化：主要进行socket的创建与连接，并创建一个单独的子线程监控socket客户端消息并解析通知上层
2. 诊断通道初始化：根据ECU信息进行路由激活处理，并根据激活结果进行判断是否激活成功
3. 诊断消息收发：上位机在路由激活成功后通过DoIP Handler进行诊断消息的收发过程，Handler会对诊断消息增加对应的DoIP头与SA、TA
4. 诊断通道去初始化：断开DoIP连接，释放诊断通道资源

在实现上，DoIP Handler主要包含DoIP线程，Epoll线程与随之创建的Socket的相关操作，时序流程可以参考下图。

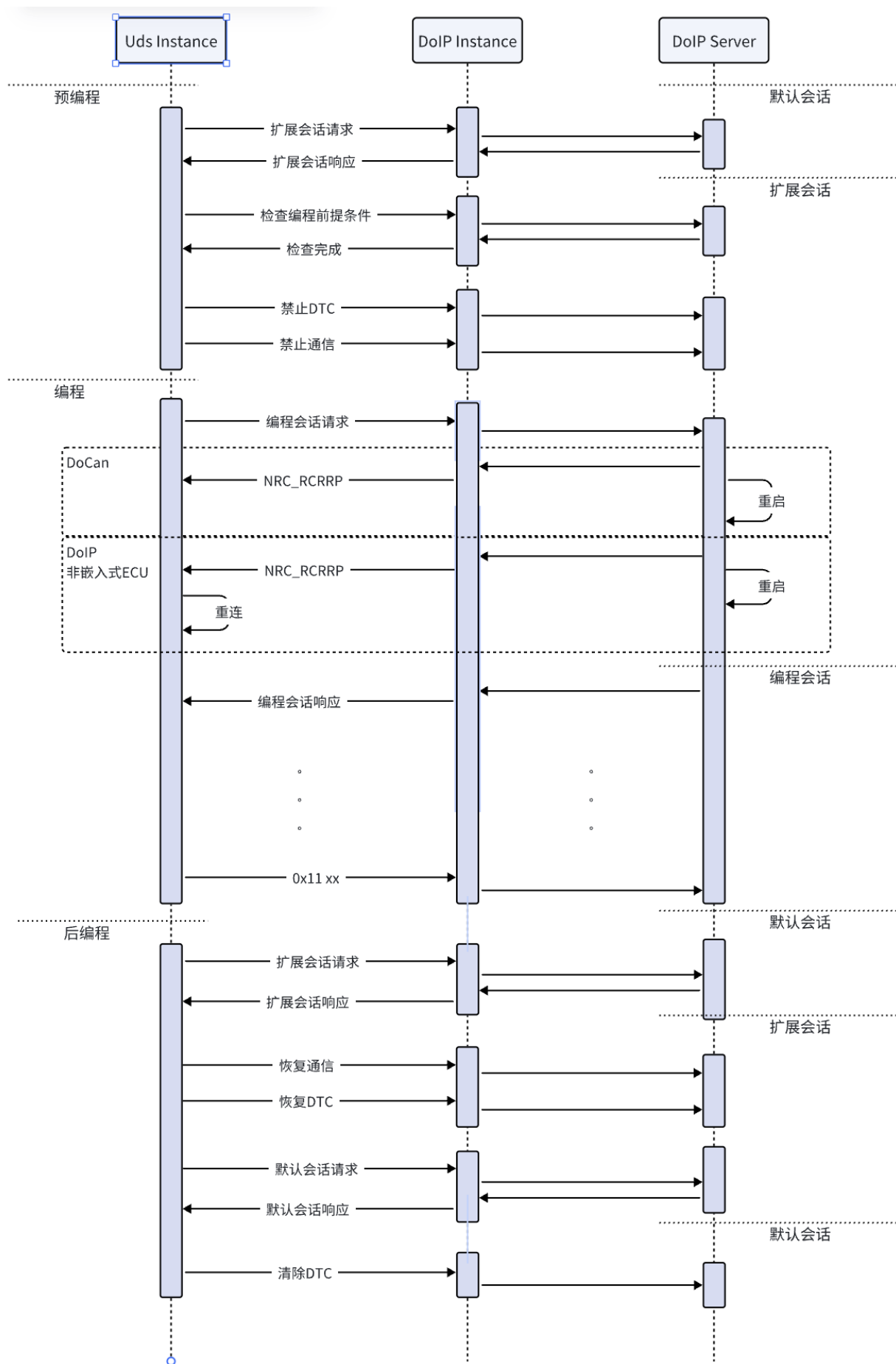


4.2.2 UDS实例刷写时序

待补充，具体刷写流程状态与主控模块有交联，初版以接口形式完成一条诊断消息的收发代替刷写流程

具体刷写流程可参考资料：<https://jqvur2c3qg.feishu.cn/docx/RUBZdAASnoEDzTx1ubhcKIU3nYf>

刷写目标不同（直连节点/下挂节点），两者的主要差异点在于直连节点刷写过程中涉及DoIP的重连操作，下挂节点由于是通过DoIP边缘节点转发消息，所以在重启过程中只需要等待结果不需要重新连接边缘节点，这方面处理会存在差异



5 接口设计

5.1 外部接口

1. 获取升级错误信息接口

函数	void get_doip_last_error(uint8_t portNo, char *errmsg, int msglen);	
同步/异步	同步	
可重入	可重入	
参数 (in)	portNo	端口Index
	msglen	错误信息长度
参数 (out)	errmsg	错误信息内容
返回值	void	
描述	收集最新的错误信息，通过接口上报给上层服务	

2. 查询当前升级件的升级状态

函数	int query_doip_upgrade_status(uint8_t portNo, uint8_t *progress, uint8_t *status, int32_t *err_code);	
同步/异步	同步	
可重入	可重入	
参数 (in)	portNo	端口Index
参数 (out)	progress	升级进度
	status	升级状态
	err_code	升级错误码
返回值	int	0: 查询成功. 其他: 失败.
描述	上层服务调用此接口查询当前的升级状态	

3. ecu启动ota流程

函数	void diag_ota_Start(EcuInfoType *ecuInfoPtr);	
同步/异步	同步	
可重入	可重入	
参数 (in)	ecuInfoPtr	ECU信息
返回值	void	
描述	车辆搜集自身的车辆版本信息，然后上报到云端，请求查看是否有新版本	

4. ecu停止ota流程

函数	void diag_ota_Stop(EcuInfoType *ecuInfoPtr);	
同步/异步	同步	
可重入	可重入	
参数 (in)	ecuInfoPtr	ECU信息
返回值	void	

描述	车辆搜集自身的车辆版本信息，然后上报到云端，请求查看是否有新版本
----	----------------------------------

5.2 内部接口

6 单元测试