

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 1 页 共 43 页

EP40&36&37&32 TBOX 与 CDCS 之间基于 TCP 的通讯协议

编制: _____

审核: _____

批准: _____

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 2 页 共 43 页

目录

1. 文档简介	4
1.1 适用范围	4
1.2 参考文档列表	4
1.3 名称术语缩写	4
1.4 版本记录	4
2. 协议数据描述	7
2.1 数据类型含义	7
3. 指令通用格式	8
3.1 指令组成	8
3.2 指令传输位序	9
4. TBOX 与 CDCS 链路架构	9
5. 交互指令	9
5.1 心跳指令	9
5.2 TBOX 发送给 CDCS 用于得到信号强度和制式。	10
5.3 拨打 ECALL/ICALL/BCALL	11
5.4 TBOX 电话状态主动通知车机	12
5.5 挂断 ECALL、BCALL、ICALL	13
5.6 获取 TBOX 的信息	13
5.7 设置 GPS 信息	14
5.8 GPS 信息主动上报	15
5.9 绑车激活	15
5.10 采集视频和图片	16
5.11 安全认证	16
5.12 TBOX 通知 CDCS 上传日志文件（旧-不用）	16
5.13 TBOX 通知 CDCS 更新预约充电状态	17
5.14 CDCS 设置预约充电	19
5.15 CDCS 开启关闭即时充电	20
5.16 CDCS 主动向 TBOX 查询当前预约充电记录	20
5.17 TBOX 通知 CDCS 有 OTA 升级任务（预留）	21
5.18 CDCS 设置电池健康长途模式（预留）	21
5.19 TBOX 通知 CDCS 同步电池健康长途模式状态（预留）	22
5.20 CDCS 主动查询电池健康长途模式状态（预留）	22
5.21 哨兵模式或 360 寻车或生物检测触发	23
5.22 哨兵模式或 360 寻车或生物检测结束	24
5.23 CDCS 通知 TBOX 再次触发生物检测	24

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 3 页 共 43 页

5.24	CDCS 通知 TBOX 有升级任务	25
5.25	CDCS 通知 TBOX 重启	25
5.26	CDCS 设置保温模式	26
5.27	CDCS 主动查询保温模式状态	27
5.28	TBOX 通知 CDCS 同步保温模式	27
5.29	CDCS 设置泊车路线	28
5.30	CDCS 主动查询泊车路线	28
5.31	TBOX 通知 CDCS 同步泊车路线	29
5.32	CDCS 设置代客或隐身模式	30
5.33	CDCS 主动查询代客或隐身模式状态	30
5.34	TBOX 通知 CDCS 同步代客或隐身模式状态	31
5.35	CDCS 设置放电功能和 SOC 目标电量	31
5.36	CDCS 获取放电功能和 SOC 设置	32
5.37	TBOX 通知 CDCS 同步放电功能和 soc 电量	32
5.38	TBOX 通知 CDCS 辅助定位状态信息	33
5.39	CDCS 向 TBOX 查询辅助定位状态信息	33
5.40	TBOX 通知 CDCS 同步网络状态信息	34
5.41	CDCS 设置 TBOX 网络制式(预留)	34
5.42	CDCS 获取 TBOX 网络制式(预留)	35
5.43	TBOX 通知 CDCS 同步网络制式(预留)	35
5.44	TBOX 通知 CDCS 设置日志等级	36
5.45	CDCS 给 TBOX 同步设置日志等级状态	36
5.46	TBOX 通知 CDCS 设置日志指令	37
5.47	CDCS 给 TBOX 同步设置日志指令状态	37
5.48	TBOX 通知 CDCS 设置应用日志开关	38
5.49	CDCS 给 TBOX 同步设置应用日志开关状态	39
5.50	TBOX 通知 CDCS 采集日志并上传(新)	39
5.51	CDCS 给 TBOX 同步采集日志状态(新)	40
5.52	CDCS 向 TBOX 请求上传文件 URL(新)	41
5.53	TBOX 向 CDCS 同步文件上传 URL(新)	41
6.	音频信号传输	42
6.1	RTP 协议概述	42
6.2	使用开源库 ORTP	42
6.3	ORTP 移植	43

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 4 页 共 43 页

1. 文档简介

本文描述了合众汽车 EP40 项目 TBOX 与 CDCS 之间基于 TCP 的通讯协议；其中，TBOX 负责上网拨号，CDCS 通过 AICS 以太网网关与 TBOX 进行路由通讯。

TBOX 与 CDCS 通讯私有协议，以通用类数据协议 TCP 为基础；数据格式使用 protobuf 协议定义。

1.1 适用范围

合众新能源汽车有限公司座舱软件部、TBOX 供应商、CDCS 供应商。

1.2 参考文档列表

参考	文档名称	备注
1	NMEA-0183 协议	
2	protocol buffers 3.0	

1.3 名称术语缩写

术语	术语解释
TBOX	通讯模块总成
CDCS	座舱域控制器
AICS	视觉域控制器（以太网网关）
NMEA	美国国家海洋电子协会的简称，现在是 GPS 导航设备统一的 RTCM 标准协议。
MD5	为计算机安全领域广泛使用的一种散列函数，用以提供消息的完整性保护。

1.4 版本记录

文档版本	变更记录	Protobuf 协议版本	pd 变更点	作者
V1.0	20210809: 初始版本	TBOX_ CDCS_ _v1.0.proto	/	贺俊
V1.1	20211104: 新增“360 寻车”接口	TBOX_ CDCS_ _v1.1.proto	/	贺俊
V1.2	20220224: 5.14 CDCS 设置预约充电（修改） 5.19 CDCS 主动向 TBOX 查询当前预约充电记录（修改） 5.16 TBOX 通知 CDCS 同步电池健康长途模式状态（新增） 5.20 哨兵模式或 360 寻车触发（新增） 5.21 哨兵模式或 360 寻车结束（新增） 5.22 360 寻车图片传输（新增）	TBOX_ CDCS_ _v1.2.proto	/	贺俊

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 5 页 共 43 页

	5.23 CDCS 通知 TBOX 有升级任务（新增） 5.24 CDCS 通知 TBOX 重启（新增） 5.25 CDCS 设置保温模式（新增） 5.26 CDCS 主动查询保温模式状态（新增） 5.27 TBOX 通知 CDCS 同步保温模式（新增）			
V1.3	新增： 5.20 CDCS 主动查询电池健康长途模式状态 5.13、 5.14、5.16 中增加 “iscontinuecharge = 未达目标电量是否继续充” 和 “timelong = 充电时长” 5.21 哨兵模式或 360 寻车或生物检测触发增加 “url= 文件上传地址” 和 “eventid = 事件 id” 5.25 中增加 “event = 预约新增/修改/删除”	TBOX_CDCS_v1.3.proto	/	贺俊
V1.4	修改： 5.22 哨兵模式或 360 寻车或生物检测结束增加 “uploadsts = 上传成功/失败”	TBOX_CD CS_v1.4.pro to	/	贺俊
V1.5	修改： 5.26CDCS 主动查询保温模式状态中的 “battheatmoderecode” 改成 “battheatmoderecord” 5.29CDCS 主动查询泊车路线中的 “parkingrouterecode” 改成 “parkingrouterecord” 5.21 哨兵模式或 360 寻车或生物检测触发接口中增加 “filename = 文件名称; //多文件上传,文件名之间通过逗号分隔。”	TBOX_CD CS_v1.5.pro to	/	贺俊
V1.6	修改： 5.25 CDCS 设置保温模式接口中增加 “cycle = 重复生效” 字段； 5.26 CDCS 主动查询保温模式状态接口中增加 “cycle = 重复生效” 字段； 5.27 TBOX 通知 CDCS 同步保温模式接口中增加 “cycle = 重复生效” 字段；	TBOX_CD CS_v1.6.pro to	修改： 1. BattHeatSingle 2. SetBattHeatMode	贺俊
V1.7	新增： 5.31 CDCS 设置代客或隐私模式； 5.32 CDCS 主动查询代客或隐私模式状态； 5.33 TBOX 通知 CDCS 同步代客或隐私模式状态； 5.34 CDCS 设置放电功能和 SOC 目标电量； 5.35 CDCS 获取放电功能和 SOC 设置； 5.36 TBOX 通知 CDCS 同步放电功能和 SOC 电量；	TBOX_CD CS_v1.7.pro to	新增： 1. enum MessageType 中新增 12 个类型； 2. SubMode Switch； 3. SecurityGuard 4. Discharge State 5. TopMessage 中增加 2 个类型；	贺俊

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 6 页 共 43 页

V1.8	修改: 5.31 CDCS 设置代客或隐身模式; 5.32 CDCS 主动查询代客或隐身模式状态; 5.33 TBOX 通知 CDCS 同步代客或隐身模式状态; 以上三个接口中对于代客模式和隐身模式开关字段类型做了调整, 有原有的结构体数组改成 uint32 类型, 通过每一个 bit 位的 0 和 1 来判断开关;	TBOX_CD CS_v1.8.proto	删除: SubModeSwitch 修改: SecurityGuard	贺俊
V1.9	修改: 5.31 CDCS 设置代客或隐身模式; 5.32 CDCS 主动查询代客或隐身模式状态; 5.33 TBOX 通知 CDCS 同步代客或隐身模式状态; 以上三个接口只保留了“代客模式开关”和“隐身模式开关(位置信息权限-手机 APP)”这两项, 并且取消了“password”密码字段;	TBOX_CD CS_v1.9.proto	修改: SecurityGuard 中删除 password 字段	贺俊
V2.0	修改: 1. “BOX_CDCS_v2.0.proto” 文件中 enum 类型都增加 “unknown=0” 类型; 2. 第 5.1 节中心跳类型明确 NMEA 心跳为 1 和 3CALL 心跳为 2;	TBOX_CD CS_v2.0.proto	修改: enum 类型中增加 unknown=0	贺俊
V2.1	修改: 1. 5.22 哨兵模式或 360 寻车或生物检测结束中增加两个字段 “eventid 和 filename” 用于反馈文件上传成功失败状态; 新增: 1. 5.23 CDCS 通知 TBOX 再次触发生物检测;	TBOX_CD CS_v2.1.proto	修改: Message EndSentinel Mode 增加两个字段	贺俊
V2.2	修改: 1. 5.32、5.33、5.34 中 valetmode 字段增加类型 “3: 未初始化”; 新增: 1. 5.40 TBOX 通知 CDCS 同步网络状态信息;	TBOX_CD CS_v2.2.proto	增加两个消息类型和一个 message NetworkStatus	贺俊
V2.3	修改: 1. 5.28 TBOX 同步 CDCS 保温模式由单个改为列表; 2. 5.32、5.33、5.34 中对 “stealthmode” 隐身模式字段类型定义进行了说明; 新增: 1. 5.41、5.42、5.43 关于 TBOX 网络制式的设置、查询、同步接口;	TBOX_CD CS_v2.3.proto	增加了一个消息类型 message TboxSignalType	贺俊
V2.4	增加 TBOX 计算逻辑: 1. 5.14 CDCS 设置预约充电 CDCS 设置预约充电时未设置结束时间的场景 2. 5.13 TBOX 通知 CDCS 更新预约充电状态 APP 设置预约充电时未设置结束时间的场景	TBOX_CD CS_v2.4.proto	增加 timelong 默认时长为 8hr	朱莉

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 7 页 共 43 页

V2.5	修改 V2.4 中预约充电的默认时长	TBOX_CD CS_v2.5.pro to	修改 timelong 默 认时长为 10hr, 可做 调整	朱莉
v2.6	新增了 5.44、5.45、5.46 日志等级、日志指令、应用 日志开关接口。	TBOX_CD CS_v2.6.pro to	新增接口: LogLevel、 LogLevelRes ult、 LogCmd、 LogCmdResu lt、 LogApp、 LogAppResu lt	贺俊
v2.7	修改并新增 5.44、5.45、5.46、5.47、5.48、5.49、 5.50、5.51 日志等级、日志指令、应用日志开关、日 志采集接口。	TBOX_CD CS_v2.7.pro to	新增接口: LogCollect、 LogCollectR esult	贺俊
v2.8	修改 5.50、5.51 日志采集接口新增一个字段: logid = 日志采集 ID。	TBOX_CD CS_v2.8.pro to	修改接口: LogCollect、 LogCollectR esult	贺俊
v2.9	1. 修改 5.50 日志采集接口删除一个字段: url=采集 文件地址。 2. 新增 5.52、5.53 获取 url 地址和同步 url 地址。	TBOX_CD CS_v2.9.pro to	修改接口: LogCollect 新增接口: GetUploadUr l、 SyncUpload Url	贺俊
v3.0	修改 5.51 日志采集状态同步接口字段定义调整: logcollectresult = 处理结果; //0: 采集失败, 1:采集完 成 (0 和 1 的定义置反, 保持和云平台协议统一)	TBOX_CD CS_v2.9.pro to	不变	贺俊
V3.1	修改 5.21、5.22、5.23, 1) 增加贵重物品检测字段说明; 2) 增加 CDCS 触发生物、贵重物品检测字段定义;	TBOX_CD CS_v3.1.pro to	修改 ReStartSentin elMode 中, 增加 object	丁进

2. 协议数据描述

2.1 数据类型含义

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 8 页 共 43 页

数据类型	说明	取值说明
int32	有符号的 32 位 Integer	
uint32	无符号的 32 位 Integer	
sint32	负数时用的编码方式	
fixed32	4 个字节，如果数值比 228 大的话，这个类型会比 uint32 高效。	
sfixed32	4 个字节	
string	UTF-8 或 7bit 的 ASCII 字符	
bytes	字节	

3. 指令通用格式

3.1 指令组成

每条消息由头、消息体和消息尾组成：

字段名	说明
Protocol Header	协议数据头
Protocol Body	协议数据应用层信息
Protocol Tail	协议数据尾

对应下面的消息结构：

协议数据头		协议数据应用层信息	协议数据尾
开始标志	消息体长度	消息体	结束标志
7 字节（#START*）	2 字节	（消息体长度）个字节	5 字节（#END*）

由于 TCP 协议本身原因，CDCS 接收可能会存在分包/粘包问题，发送数据增加开始标志、数据长度、结束标志 3 个标识符；TBOX 与 CDCS 交互数据格式为：开始标志+protobuf 协议消息体长度+protobuf 协议数据包+结束标志，TBOX 接收做出相应的组包处理。

开始标志：#START*

结束标志：#END*

开始标志：#START*，七个字节。

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 9 页 共 43 页

消息体长度： 用于表示消息体的字节数，不大于 10286。

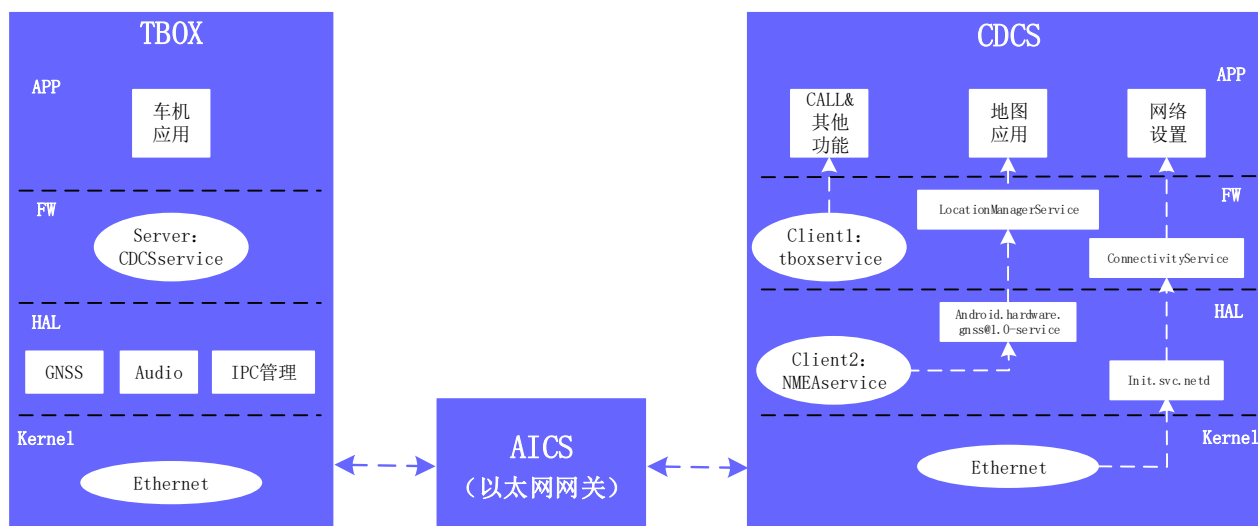
消息体： 传输的内容。

结束标志：#END*， 五个字节。

3.2 指令传输位序

字节位采用大端模式，即传输的第一个字节是最高位。

4. TBOX 与 CDCS 链路架构



TBOX 作为 server 端，地址 172.16.1.50 端口 5757。CDCS 作为 client 端，需要分为两个 client 连上 TBOX，其中一个 client 来请求 NMEA 数据，另一个 client 来请求其他数据。请求 NMEA 的 client 与 TBOX 的交互指令有（5.1、5.7、5.8）、另一个 client 与 TBOX 的交互指令有（5.1、5.2、5.3、5.4、5.5、5.6、5.9、5.10.....）。

注：两个 client 连上之后都需定时发心跳，考虑到 CDCS 有时不需要 NMEA 数据，如果不需要 client 主动关闭链路，需要 NMEA 数据在主动连上来，并且 TBOX 会以 1s 的频率发送 GNSS 数据。

5. 交互指令

5.1 心跳指令

CDCS 发送给 TBOX，用于 TBOX 判断 client 端工作是否正常。

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 10 页 共 43 页

CDCS请求TBOX:

```
message TopMessage{
    message_type = REQUEST_HEARTBEAT_SIGNAL;
    ihu_heart_type {
        heart_type = 心跳类型区分 client (3call 或 nmea)
    }
}
```

TBOX回复CDCS:

```
message TopMessage{
    message_type = RESPONSE_HEARTBEAT_RESULT;
    msg_result{
        Result = 处理结果: true 为成功, false 为失败
        error_code = 错误代码提示(可自定义)
    }
}
```

注: CDCS和TBOX建立通讯连接后, CDCS需要以1s周期定时发送心跳, TBOX判断CDCS心跳超时时间为3s。

5.2 TBOX 发送给 CDCS 用于得到信号强度和制式。

CDCS发送给TBOX, 用于请求获取信号强度的信息:

```
message TopMessage{
    message_type = REQUEST_NETWORK_SIGNAL_STRENGTH;
    tbox_network_ctrl{
        onoff = 网络信号强度和制式发送到 CDCS 使用
        time_cycle = 发送周期毫秒值(默认 1000ms)
    }
}
```

TBOX回复CDCS:

```
message TopMessage{
    message_type = RESPONE_NETWORK_SIGNAL_STRENGTH;
    msg_result{
        Result=处理结果: true 为成功, false 为失败
        error_code=错误代码提示
    }
    signal_type = TBOX 当前网络制式; (制式值 0~3)
    signal_power = TBOX 当前信号强度; (信号强弱值 0~4)
```

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 11 页 共 43 页

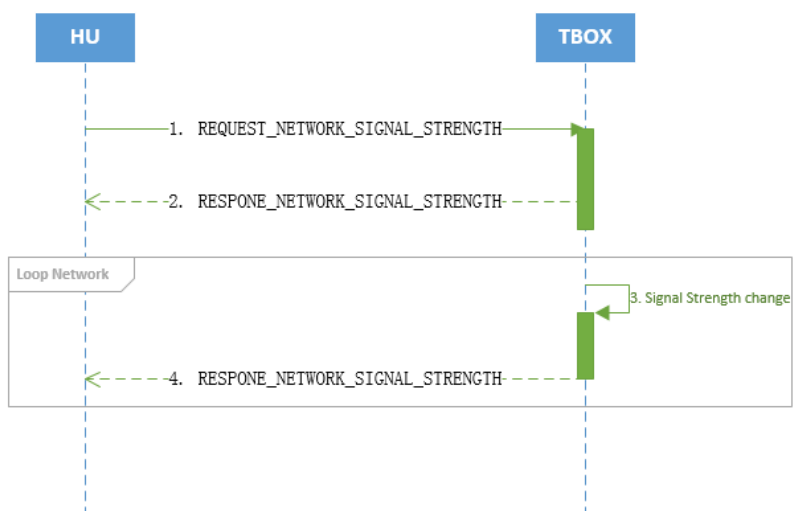
}

网络制式:

```
0: NONE_SIGNAL; // no signal
1: GSM;          //2G
2: UMTS;         //3G
3: LTE;          //4G
```

信号强弱值表示:

```
0: 0% ~ 20%
1: 20% ~ 40%
2: 40% ~ 60%
3: 60% ~ 80%
4: 80% ~ 100%
```



5.3 拨打 ECALL/ICALL/BCALL

CDCS发送给TBOX, 用于拨打ECALL/ICALL/BCALL:

```
message TopMessage{
    message_type = REQUEST_CALL_ACTION;
    call_action{
        type = ECALL/ICALL/BCALL;
        action = START_CALL;
    }
}
```

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 12 页 共 43 页

TBOX 回复 CDCS:

```
message TopMessage{
    message_type = RESPONSE_CALL_ACTION_RESULT;
    msg_result{
        Result = 处理结果: true 为成功, false 为失败
        error_code = 错误代码提示
    }
    call_result{
        type = ECALL/ICALL/BCALL;
        action = START_CALL;
        result = 拨打电话结果;
    }
}
```

拨打电话结果:

CALL_ACTION_SUCCESS 拨打成功

CALL_ACTION_NONSUPPORT 不支持该操作方式——如不支持通过车机拨打 ECALL 电话。

START_FAILED_CALL_PROCESSING 拨打失败, 当前已经有一通电话正在进行

START_FAILED_OTHER_ERROR 拨打失败, 其他未知原因

5.4 TBOX 电话状态主动通知车机

当 TBOX 拨打电话成功, 并且与 CDCS 联通, TBOX ECALL\ICALL\BCALL 电话状态发生变化时, 会主动发消息给 CDCS; 交互如下:

```
message TopMessage{
    message_type = RESPONSE_CALL_STATUS;
    call_status{
        type = ECALL/BCALL/ICALL
        call_status = TBOX 最新电话状态
        reserve = 预留字段
    }
}
```

CDCS回复TBOX (增加CDCS电话状态画面切换响应):

```
message TopMessage{
    message_type = IHU_CALL_STATUS_RESULT;
    msg_result{
        result = 处理结果: true 为成功, false 为失败
        error_code = 错误代码提示
    }
}
```

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 13 页 共 43 页

```

    }
}

```

电话状态:

CALL_IN 来电

CALL_OUT 去电

CALL_RING 对方已振铃

CALL_CONNECTED 接通

CALL_DISCONNECTED 挂断

5.5 挂断 ECALL、BCALL、ICALL

CDCS发送给TBOX，请求挂断电话：

```

message TopMessage{
    message_type = REQUEST_CALL_ACTION;
    call_action{
        type = ECALL/BCALL/ICALL;
        action = END_CALL;
    }
}

```

TBOX回复CDCS：

```

message TopMessage{
    message_type = RESPONSE_CALL_ACTION_RESULT;
    msg_result{
        Result=处理结果：true 为成功，false 为失败
        error_code=错误代码提示
    }
    call_result{
        type = ECALL/ICALL/BCALL;
        action = END_CALL;
        result = 挂断电话结果
    }
}

```

5.6 获取 TBOX 的信息

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 14 页 共 43 页

CDCS发送给TBOX, 请求获取TBOX信息:

```
message TopMessage{
    message_type = REQUEST_TBOX_INFO;
}
```

TBOX回复CDCS:

```
message TopMessage{
    message_type = RESPONSE_TBOX_INFO;
    msg_result{
        Result=处理结果: true 为成功, false 为失败
        error_code=错误代码提示
    }
    tbox_info{
        software_version = TBOX 软件版本;
        hardware_version = TBOX 硬件版本;
        iccid = TBOX ICCID;
        pdid = TBOX PDID;
        imei= TBOX IMEI
        vin =车辆 VIN
    }
}
```

5.7 设置 GPS 信息

CDCS发送给TBOX, 请求获取GPS信息:

```
message TopMessage{
    message_type = REQUEST_TBOX_GPS_SET;
    tbox_gps_ctrl{
        onoff = GNSS 信息从 TBOX 发送到 CDCS 使能。
        time_cycle = 发送周期毫秒值(默认 1000ms)
    }
}
```

TBOX回复CDCS:

```
message TopMessage{
    message_type = RESPONSE_TBOX_GPS_SET_RESULT;
    msg_result{
        Result=处理结果: true 为成功, false 为失败
    }
}
```

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 15 页 共 43 页

error_code=错误代码提示

}

}

注：TBOX收到CDCS获取GPS信号请求后，TBOX以1s的频率转发GPS信号。

5.8 GPS 信息主动上报

TBOX周期发送给CDCS的GPS信息, 依赖于GPS的设置模式:

```
message TopMessage{
    message_type = RESPONSE_TBOX_GPSINFO_RESULT;
    tbox_gpsinfo {
        naea = NMEA; //包括 NMEA-0183 标准当中的 GGA、GSA、GSV、RMC 几项数据；每项数据都遵循 NMEA
        标准的$....；格式：多项数据直接拼接成一个长字符串。
    }
}
```

NMEA数据采样率要求:

NMEA数据项	每次包含的数据个数	采样率 (Hz)	备注
GGA	1	1	
GSA	1	1	
GSV	1-4	1	GPGSV\BDGSV(需要区分GPS和北斗卫星数)
RMC	1	1	

5.9 绑车激活

TBOX得知TSP给的绑车激活信息后，通知CDCS激活成功状态:

```
message TopMessage{
    message_type = RESPONSE_TBOX_ACTIVESTATE_RESULT;
    tbox_activestate {
        active_state = 激活状态; //false: 未激活, true: 已激活
    }
}
```

CDCS回复TBOX:

```
message TopMessage{
    message_type = IHU_ACTIVESTATE_RESULT;
    msg_result{
        result=处理结果: true 为成功, false 为失败
        error_code=错误代码提示
    }
}
```

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 16 页 共 43 页

}

注：绑车激活状态，TBOX是从与TSP的远程配置协议中获取信息。

5.10 采集视频和图片

TBOX通知CDCS录制视频和拍照，有以下两种触发方式：

1. TSP下发远程诊断录制视频和拍照指令时，TBOX需要通知CDCS录制视频和拍照；
2. 触发ECALL时，TBOX需要通知CDCS录制视频和拍照。

```
message TopMessage{
    message_type = REQUEST_TBOX_REMOTEDIAGNOSE;
    tbox_remotediagnose {
        vin = 车辆识别码;
        eventid = 事件ID; //远程诊断TSP触发：eventid（TSP传入）；ECALL触发：eventid不传；
        timestamp = 远程监控触发的时间戳；
        datatype = 数据类型； //1. 图片；2. 视频
        cameraname = 摄像头类型； //1. DVR；2. DMS
        aid = 触发的业务ID； //远程诊断TSP触发：aid（TSP传入）；ECALL触发：aid = 170；
        mid = 触发的消息ID； //远程诊断TSP触发：mid（TSP传入）；ECALL触发：mid = 3；
        effectivetime = 采集有效时长； //远程诊断TSP触发：（TSP传入）；ECALL触发：不传
        sizelimit = 采集数据包大小限制； //远程诊断TSP触发：（TSP传入）；ECALL触发：不传
    }
}
```

CDCS回复TBOX,

```
message TopMessage{
    message_type = RESPONSE_TBOX_REMOTEDIAGNOSE_RESULT;
    msg_result{
        Result=处理结果：true 为成功，false 为失败
        error_code=错误代码提示
    }
}
```

5.11 安全认证

由PKI完成TBOX和HU的认证过程，目前暂未确定在本项目中实现。

5.12 TBOX 通知 CDCS 上传日志文件（旧-不用）

TSP给TBOX下发采集CDCS的日志请求，TBOX通知CDCS采集日志并上传日志文件（CDCS直接将日志文件上传TSP平

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 17 页 共 43 页

台):

```
message TopMessage{
    message_type = REQUEST_IHU_LOGFILE;
    ihu_logfile {
        vin = 车辆识别码;
        eventid = 事件ID; //远程诊断TSP触发: eventid
        timestamp = 远程诊断触发的时间戳;
        aid = 触发的业务ID; //远程诊断TSP触发: aid
        mid = 触发的消息ID; //远程诊断TSP触发: mid
        startTime = 开始采集时间; //从1970-01-01T00:00:00 UTC开始的秒数
        durationTime = 采集持续时间; //以秒为单位0-65535
        channel = 2; //1:TBOX日志, 2: IHU日志
        level = 日志级别; //1:ERROR, 2:WARN, 3:INFO, 4:DEBUG (每个级别代表一种日志类型)
    }
}
```

IHU回复TBOX:

```
message TopMessage{
    message_type = RESPONSE_IHU_LOGFILE_RESULT;
    msg_result{
        Result=处理结果: true 为成功, false 为失败
        error_code=错误代码提示
    }
}
```

5.13 TBOX 通知 CDCS 更新预约充电状态

当TBOX收到TSP发起的预约充电设置指令后（以时间戳判断指令先后顺序），主动通知CDCS更新预约充电状态:

```
message TopMessage{
    message_type = REQUEST_IHU_CHARGEAPPOINTMENTSTS;
    ihu_chargeappointmentsts {
        timestamp = 时间戳(从1970年到现在的秒数);
        hour = 预约小时; //0-23小时
        min = 预约分钟; //0-59分钟
        id = 预约ID; //id由tsp
        targetpower = 目标电量; // 0-100 默认传100
    }
}
```

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 18 页 共 43 页

timelong = 充电时长; //min 默认10小时 (时长可做调整)

effectivestate = 预约是否生效; //0: 预约失效; 1: 预约生效 (每天都有效)

iscontinuecharge = 未达目标电量是否继续充; //0:无需; 1: 需要

}

}

CDCS回复TBOX:

```
message TopMessage{
    message_type = RESPONSE_IHU_CHARGEAPPOINTMENTSTS_RESULT;
    msg_result{
        Result=处理结果: true 为成功, false 为失败
        error_code=错误代码提示
    }
}
```

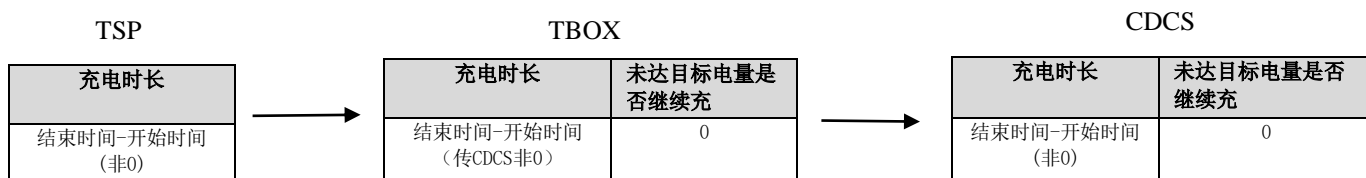
注: TBOX和CDCS每次建立通讯链路后, 如果有有效预约记录, TBOX立即同步一次预约信息给到CDCS。

备注:

若APP未设置预约充电结束时间, SOC需要充满至目标电量;

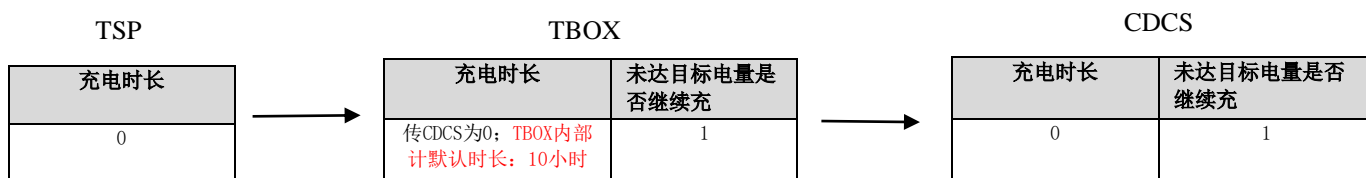
若已过预约充电开始时间, 在充电开启有效时间段内插上充电枪都能立即执行充电流程;

1) APP设置预约充电结束时间



※充电开启有效时间段=结束时间-开始时间=充电时长

2) APP只设置预约充电开始时间



※充电开启有效时间段=开始时间+默认时长 (10小时)

TBOX和TSP协议请参考平台协议

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 19 页 共 43 页

5.14 CDCS 设置预约充电

CDCS主动通知TBOX设置预约充电，TBOX设置预约（以时间戳判断指令先后顺序）后还需要将预约信息同步给TSP。

```

message TopMessage{
    message_type = REQUEST_TBOX_CHARGEAPPOINTMENTSET;
    tbox_chargeappointmentset {
        timestamp = 时间戳(从1970年到现在的秒数);
        hour = 预约小时; //0-23小时
        min = 预约分钟; //0-59分钟
        id = 预约ID; // 如果IHU首次发起的预约设置，id有可能没有就传空
        targetpower = 目标电量; // 0-100 默认传100
        timelong = 充电时长; //min默认10小时（时长可做调整）
        effectivestate = 预约是否生效; //0: 预约失效; 1: 预约生效（每天往复）
        iscontinuecharge = 未达目标电量是否继续充; //0:无需; 1: 需要
    }
}

```

TBOX回复CDCS:

```

message TopMessage{
    message_type = RESPONSE_TBOX_CHARGEAPPOINTMENTSET_RESULT;
    msg_result{
        Result=处理结果: true 为成功, false 为失败
        error_code=错误代码提示
    }
}

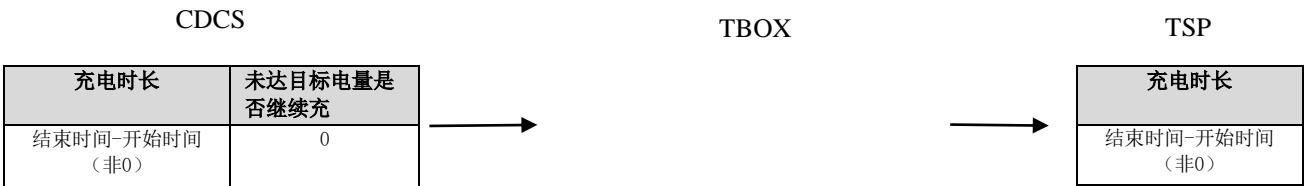
```

备注:

若IHU未设置预约充电结束时间，SOC需要充满至目标电量；

若已过预约充电开始时间，在充电开启有效时间段内插上充电枪都能立即执行充电流程；

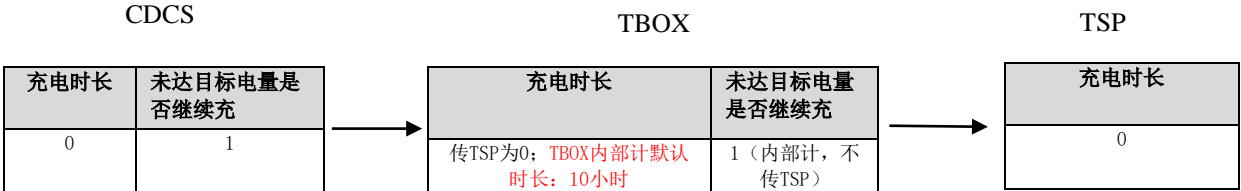
1) IHU设置预约充电结束时间



※充电开启有效时间段=结束时间-开始时间=充电时长

	TBOX 与 CDCS 之间基于 TCP 的		编号:
	座舱系统/	充电时长 未达目标电量是否继续充	
	座舱系统/	充电时长 未达目标电量是否继续充	20 页 共 43 页

2) IHU只设置预约充电开始时间



※充电开启有效时间段=开始时间+默认时长（10小时）

TBOX和TSP协议请参考平台协议

5.15 CDCS 开启关闭即时充电

当CDCS需要发起立即充电请求时，发起此接口通知TBOX发起立即充电控制指令

```
message TopMessage{
    message_type = REQUEST_TBOX_CHARGECTRL;
    tbox_chargectl {
        timestamp = 充电控制触发的时间戳;
        commend = 开启关闭; //0: 关闭; 1: 开启
        targetpower = 目标电量; // 0-100 默认传100
    }
}
```

TBOX回复CDCS:

```
message TopMessage{
    message_type = RESPONSE_TBOX_CHARGECTRL_RESULT;
    msg_result{
        Result=处理结果: true 为成功, false 为失败
        error_code=错误代码提示
    }
}
```

5.16 CDCS 主动向 TBOX 查询当前预约充电记录

CDCS可以主动向TBOX查询当前预约充电的记录:

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 21 页 共 43 页

```

message TopMessage{
    message_type = REQUEST_QUERY_CHARGE_RECORD;
}

```

TBOX回复CDCS:

```

message TopMessage{
    message_type = RESPONSE_CHARGE_RECORD_RESULT;
    charge_record_result{
        timestamp = 时间戳(从1970年到现在的秒数);
        hour = 预约小时; //0-23小时
        min = 预约分钟; //0-59分钟
        id = 预约ID; //id由tsp
        targetpower = 目标电量; // 0-100 默认传100
        timelong = 充电时长; //min
        effectivestate = 预约是否生效; //0: 预约失效; 1: 预约生效(每天都有效)
        iscontinuecharge = 未达目标电量是否继续充; //0:无需; 1: 需要
    }
}

```

5.17 TBOX 通知 CDCS 有 OTA 升级任务（预留）

当TBOX收到TSP发起的OTA升级任务后，主动通知CDCS有新的OTA升级任务：

```

message TopMessage{
    message_type = REQUEST_OTAUUPDATE_TASK;
}

```

CDCS回复TBOX:

```

message TopMessage{
    message_type = RESPONSE_OTAUUPDATE_TASK_RESULT;
    msg_result{
        result=处理结果: true 为成功, false 为失败
        error_code=错误代码提示
    }
}

```

5.18 CDCS 设置电池健康长途模式（预留）

当CDCS需要发起设置电池健康长途模式请求时，发起此接口通知TBOX发起电池健康长途控制指令

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 22 页 共 43 页

```

message TopMessage{
    message_type = REQUEST_TBOX_BATTERYMODE;
    tbox_battery_mode {
        timestamp = 设置电池健康长途模式触发的时间戳；（从1970年到现在的秒数）
        mode = 健康/长途； //0：健康模式-80%； 1：长途模式-100%； 2:健康模式-90%
    }
}

TBOX回复CDCS:
message TopMessage{
    message_type = RESPONSE_TBOX_BATTERYMODE_RESULT;
    tbox_battery_mode_result{
        timestamp = 时间戳(从 1970 年到现在的秒数);
        result = 处理结果； //0:设置失败，条件不满足； 1：设置成功，记忆此状态；
        //2：设置成功，不记忆此状态
    }
}

```

5.19 TBOX 通知 CDCS 同步电池健康长途模式状态（预留）

```

message TopMessage{
    message_type = REQUEST_SYNC_BATTERYMODE;
    tbox_battery_mode {
        timestamp = 设置电池健康长途模式触发的时间戳；（从1970年到现在的秒数）
        mode = 健康/长途； //0：健康模式-80%； 1：长途模式-100%； 2:健康模式-90%
    }
}

TBOX回复CDCS:
message TopMessage{
    message_type = RESPONSE_SYNC_BATTERYMODE_RESULT;
    msg_result{
        result=处理结果： true 为成功， false 为失败
        error_code=错误代码提示
    }
}

```

注：TBOX与CDCS建立连接，TBOX主动同步一次，之后有状态变化再次同步；

5.20 CDCS 主动查询电池健康长途模式状态（预留）

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 23 页 共 43 页

```

message TopMessage{
    message_type = REQUEST_QUERY_BATTERYMODE;
}
TBOX回复CDCS:
message TopMessage{
    message_type = RESPONSE_QUERY_BATTERYMODE_RESULT;
    tbox_battery_mode {
        timestamp = 设置电池健康长途模式触发的时间戳; (从1970年到现在的秒数)
        mode = 健康/长途; //0: 健康模式-80%; 1: 长途模式-100%; 2: 健康模式-90%
    }
}

```

5.21 哨兵模式或 360 寻车或生物检测触发

哨兵模式：当TBOX监测得到车辆出现异常震动或位移时，会触发哨兵模式；此时TBOX会将车辆唤醒上电并通知CDCS采集视频。

360寻车：当远程车控APP触发360寻车，在条件满足的情况下，TBOX将车辆唤醒并通知CDCS采集图片并上传。

生物检测：当CDCS检测到车内有活物时，通知TBOX，TBOX通知CDCS采集图片并上传；

贵重物品检测：当CDCS检测到车内有贵重物品时，通知TBOX，TBOX通知CDCS采集图片并上传；

```

message TopMessage{
    message_type = REQUEST_START_SENTINELMODE;
    start_sentinelmode {
        timestamp = 哨兵模式或360寻车触发的时间戳; (从1970年到现在的秒数)
        mode = 哨兵模式/360寻车/生物监测; //0: 哨兵模式; 1: 360寻车; 2: 生物监测; 3: 贵重物品检测;
        type = 视频/图片; //0: 视频; 1: 图片
        vin = 车辆识别码;
        url = 文件地址; //多文件上传, 文件URL之间通过逗号分隔。
        filename = 文件名称; //多文件上传, 文件名之间通过逗号分隔。
        eventid = 事件ID; //根据TSP平台值提供
    }
}

```

IHU回复TBOX,

```

message TopMessage{
    message_type = RESPONSE_START_SENTINELMODE_RESULT;
    msg_result{
        result=处理结果: true 为已执行, false 为失败
        error_code=错误代码提示
    }
}

```

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 24 页 共 43 页

```

    }
}

```

5.22 哨兵模式或 360 寻车或生物检测结束

哨兵模式：当CDCS采集完视频并上传TSP后，通知TBOX哨兵模式结束。

360寻车：当CDCS采集完图片并上传至TSP后，通知TBOX 360寻车结束。

生物检查：当CDCS采集完图片并上传TSP后，通知TBOX生物检查结束。

贵重物品检测：当CDCS采集完图片并上传TSP后，通知TBOX贵重物品检测结束

```

message TopMessage{
    message_type = REQUEST_END_SENTINELMODE;
    end_sentinelmode {
        timestamp = 哨兵模式或360寻车结束的时间戳；（从1970年到现在的秒数）
        mode = 哨兵模式/360寻车/生物监测； //0：哨兵模式； 1：360寻车； 2：生物监测； 3：贵重
        物品检测；
        eventid = 事件ID；//根据TBOX传的值反馈回来
        filename = 文件名称；//上传成功的文件名称；
        uploadsts = 上传成功/失败； //0:失败； 1：成功；
    }
}

TBOX回复IHU,
message TopMessage{
    message_type = RESPONSE_END_SENTINELMODE_RESULT;
    msg_result{
        result=处理结果：true 为已执行，false 为失败
        error_code=错误代码提示
    }
}

```

5.23 CDCS 通知 TBOX 再次触发生物检测

CDCS通知TBOX首次/再次触发生物检测，当再次触发时，TBOX告知TSP 云平台是1. SOC>20%，且空调已开启60min，但未达到120min；2. SOC<20%，且空调已开启10min；3. SOC>20%，且空调已开启120min；

然后TBOX 在上报云平台的同时，触发通知CDCS 生物检测上传图片流程：

```

message TopMessage{
    message_type = REQUEST_RESTART_SENTINELMODE;
    restart_ sentinelmode {

```


	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 25 页 共 43 页

```

timestamp = 生物检测或贵重物品触发的时间戳；（从1970年到现在的秒数）
mode =生物检测/贵重物品检测；//0：生物检测； 1：贵重物品检测；
object=对象类型；//0：活体（具体未知）；1：儿童；2：宠物；3：儿童和宠物；4：手机；5：包
包；6：手机和包包；
reason = 触发原因；//0：首次触发；1：SOC>20%，且空调已开启120min；2：SOC<20%，且空调已
开启10min；3：SOC>20%，且空调已开启60min，但未达到120min；
    }
}
TBOX回复IHU,
message TopMessage{
    message_type = RESPONSE_RESTART_SENTINELMODE_RESULT;
    msg_result{
        result=处理结果：true 为已执行，false 为失败
        error_code=错误代码提示
    }
}

```

5.24 CDCS 通知 TBOX 有升级任务

CDCS工程模式中有个触发TBOX升级任务的按钮，点击后主动通知TBOX有升级任务：

```

message TopMessage{
    message_type = REQUEST_TBOXUPDATE_TASK;
}

```

TBOX回复CDCS是否收到升级任务：

```

message TopMessage{
    message_type = RESPONSE_TBOXUPDATE_TASK_RESULT;
    msg_result{
        result=处理结果：true 为成功，false 为失败
        error_code=错误代码提示
    }
}

```

5.25 CDCS 通知 TBOX 重启

CDCS工程模式中有个触发TBOX升级任务的按钮，点击后主动通知TBOX有升级任务：

```

message TopMessage{
    message_type = REQUEST_TBOX_REBOOT;
}

```

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 26 页 共 43 页

}

TBOX回复CDCS是否收到升级任务:

```
message TopMessage{
    message_type = RESPONSE_TBOX_REBOOT_RESULT;
    msg_result{
        result=处理结果: true 为成功, false 为失败
        error_code=错误代码提示
    }
}
```

5. 26 CDCS 设置保温模式

当CDCS需要发起设置插枪保温或预约保温请求时，发起此接口通知TBOX发起保温模式设置指令

```
message TopMessage{
    message_type = REQUEST_SET_BATTHEATMODE;
    set_battheatmode {
        timestamp = 时间戳; //(从1970年到现在的秒数)
        hour = 预约小时; //0-23小时（此字段只有预约保温模式有效）
        min = 预约分钟; //0-5分钟（此字段只有预约保温模式有效）
        id = 预约ID; //IHU首次发起预约设置，id有可能没有就传空（此字段只有预约保温模式有效）
        cycle = 重复生效; //无符号整型(0-255)，第一位代表是否重复 0：不重复;1：重复;
            //后七位代表周几重复，如：00000000—不重复 ;11010000—周一、三重重复;
            //11111111—默认，每天重复
        event = 预约新增/修改/删除; //0：新增，1：修改，2：删除
        mode = 关闭/插枪保温/预约保温; //0：关闭；1：插枪保温；2：预约保温
    }
}
```

TBOX回复CDCS:

```
message TopMessage{
    message_type = RESPONSE_SET_BATTHEATMODE_RESULT;
    set_battheatmode_result{
        timestamp = 时间戳; //(从 1970 年到现在的秒数)
        result = 处理结果; //0:设置失败，条件不满足；1：设置成功，记忆此状态;
    }
}
```

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 27 页 共 43 页

5.27 CDCS 主动查询保温模式状态

CDCS可以主动向TBOX查询当前保温模式的状态：

```
message TopMessage{
    message_type = REQUEST_QUERY_BATTHEATMODE;
}
```

TBOX回复CDCS：

```
message TopMessage{
    message_type = RESPONSE_QUERY_BATTHEATMODE_RESULT;
    battheatmoderecord{
        timestamp = 时间戳(从1970年到现在的秒数);
        BattHeatSingle = battheatList [3]; //id 由 tbox 根据 TSP 平台记录（此字段只有预约保温模式有效）
        mode = 关闭/插枪保温/预约保温;    //0: 关闭; 1: 插枪保温; 2: 预约保温
    }
}
```

BattHeatSingle

```
{
    hour = 预约小时;    //0-23小时（此字段只有预约保温模式有效）
    min = 预约分钟;    //0-5分钟（此字段只有预约保温模式有效）
    id = 预约 ID;
    cycle = 重复生效;    //无符号整型(0-255)，第一位代表是否重复 0: 不重复;1: 重复;
                        //后七位代表周几重复，如：00000000—不重复 ;11010000—周一、三重复;
                        //11111111—默认，每天重复
}
```

注：TBOX与CDCS首次连接或保温模式界面打开时，CDCS需要主动查询一次状态。

5.28 TBOX 通知 CDCS 同步保温模式

当 APP 设置保温模式状态时，TBOX 需要把修改记录同步给 CDCS：

```
message TopMessage{
    message_type = REQUEST_SYNC_BATTHEATMODE;
    battheatmoderecord{
        timestamp = 时间戳(从1970年到现在的秒数);
        BattHeatSingle = battheatList [3]; //id 由 tbox 根据 TSP 平台记录（此字段只有预约保温模式有效）
        mode = 关闭/插枪保温/预约保温;    //0: 关闭; 1: 插枪保温; 2: 预约保温
    }
}
```

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 28 页 共 43 页

}

CDCS回复TBOX:

```
message TopMessage{
    message_type = RESPONSE_SYNC_BATTHEATMODE_RESULT;
    set_battheatmode_result{
        timestamp = 时间戳;    //(从 1970 年到现在的秒数)
        result = 处理结果;    //0:设置失败，条件不满足；1：设置成功，记忆此状态；
    }
}
```

5.29 CDCS 设置泊车路线

当CDCS收到ADCS的泊车路线数据后，需要将路线数据同步给TBOX，TBOX再同步给云平台：

```
message TopMessage{
    message_type = REQUEST_SET_PARKING_ROUTE;
    parking_route {
        timestamp = 时间戳; //(从1970年到现在的秒数)
        id = 路线ID;
        name = 路线名称;
        data = 路线数据; //数据格式由ADCS和APP约定
        event = 新增/修改/删除; //0：新增，1：修改，2：删除
    }
}
```

TBOX回复CDCS:

```
message TopMessage{
    message_type = RESPONSE_SET_PARKINGROUTE_RESULT;
    msg_result{
        result=处理结果; //true 为已执行，false 为失败
        error_code=错误代码提示;
    }
}
```

5.30 CDCS 主动查询泊车路线

CDCS可以主动向TBOX查询当前泊车路线的记录：

```
message TopMessage{
```

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 29 页 共 43 页

```

message_type = REQUEST_QUERY_PARKINGROUTE;
}

```

TBOX回复CDCS:

```

message TopMessage{
    message_type = RESPONSE_QUERY_PARKINGROUTE_RESULT;
    parkingroutererecord{
        timestamp = 时间戳(从1970年到现在的秒数);
        ParkingRouteSingle = parkingrouteList [100]; //记录最多 100 条，实际条数以平台反馈为准
    }
}

```

ParkingRouteSingle

```

{
    id = 路线ID;
    name = 路线名称;
    data = 路线数据; //数据格式由ADCS和APP约定，此字段可选
}

```

注：TBOX与CDCS首次连接或泊车路线界面打开时，CDCS需要主动查询一次状态。

5.31 TBOX 通知 CDCS 同步泊车路线

当TBOX收到TSP的泊车路线更新/修改/删除请求后，需要将路线数据同步给CDCS:

```

message TopMessage{
    message_type = REQUEST_SYNC_PARKING_ROUTE;
    parking_route {
        timestamp = 时间戳; //(从1970年到现在的秒数)
        id = 路线ID;
        name = 路线名称;
        data = 路线数据; //数据格式由ADCS和APP约定，此字段可选
        event = 新增/修改/删除; //0: 新增, 1: 修改, 2: 删除
    }
}

```

CDCS回复TBOX同步状态:

```

message TopMessage{
    message_type = RESPONSE_SYNC_PARKINGROUTE_RESULT;
    msg_result{
        result=处理结果; //true 为已执行, false 为失败
    }
}

```

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 30 页 共 43 页

```

        error_code=错误代码提示;
    }
}

```

5.32 CDCS 设置代客或隐身模式

CDCS可以设置代客模式或隐身模式开关，并通知TBOX，再同步给TSP云平台：

```

message TopMessage{
    message_type = REQUEST_SET_SECURITY_GUARD;
    security_guard {
        timestamp = 时间戳; //(从1970年到现在的秒数)
        valetmode = 代客模式开关状态; //0: 无效; 1: 开启; 2: 关闭; 3: 未初始化;
        stealthmode = 是否有位置信息权限-手机APP; //0: 无效; 1: 有; 2: 没有
    }
}

```

TBOX回复CDCS：

```

message TopMessage{
    message_type = RESPONSE_SET_SECURITY_GUARD_RESULT;
    msg_result{
        result=处理结果; //true 为已执行，false 为失败
        error_code=错误代码提示;
    }
}

```

5.33 CDCS 主动查询代客或隐身模式状态

CDCS可以主动向TBOX查询当前代客或隐身模式的记录：

```

message TopMessage{
    message_type = REQUEST_QUERY_SECURITY_GUARD;
}

```

TBOX回复CDCS：

```

message TopMessage{

```

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 31 页 共 43 页

```

message_type = RESPONSE_QUERY_SECURITY_GUARD_RESULT;
security_guard {
    timestamp = 时间戳; //(从1970年到现在的秒数)
    valetmode = 代客模式开关状态; //0: 无效; 1: 开启; 2: 关闭; 3: 未初始化;
    stealthmode = 是否有位置信息权限-手机APP; //0: 无效; 1: 有; 2: 没有
}
}

```

注: TBOX 与 CDCS 首次连接或界面打开时, CDCS 需要主动查询一次状态。

5.34 TBOX 通知 CDCS 同步代客或隐身模式状态

当TBOX收到TSP的代客模式或隐身模式状态变更请求后, 需要将状态同步给CDCS:

```

message TopMessage{
    message_type = REQUEST_SYNC_SECURITY_GUARD;
    security_guard {
        timestamp = 时间戳; //(从1970年到现在的秒数)
        valetmode = 代客模式开关状态; //0: 无效; 1: 开启; 2: 关闭; 3: 未初始化;
        stealthmode = 是否有位置信息权限-手机APP; //0: 无效; 1: 有; 2: 没有
    }
}

```

CDCS回复TBOX同步状态:

```

message TopMessage{
    message_type = RESPONSE_SYNC_SECURITY_GUARD_RESULT;
    msg_result{
        result=处理结果; //true 为已执行, false 为失败
        error_code=错误代码提示;
    }
}

```

5.35 CDCS 设置放电功能和 SOC 目标电量

```

message TopMessage{
    message_type = REQUEST_SET_DISCHARGE;
    tbox_dischargectrl {
        timestamp = 时间戳; //(从1970年到现在的秒数)
        command= 开启关闭; //0: 关闭; 1: 开启
        targetpower= 目标电量; //0-100
    }
}

```

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 32 页 共 43 页

```
    }  
}  
TBOX回复CDCS:  
message TopMessage{  
    message_type =RESPONSE_SET_DISCHARGE_RESULT;  
    msg_result{  
        result=处理结果; //true 为已执行, false 为失败  
        error_code=错误代码提示;  
    }  
}
```

5.36 CDCS 获取放电功能和 SOC 设置

```
message TopMessage{  
    message_type = REQUEST_QUERY_DISCHARGE;  
}  
TBOX回复CDCS:  
message TopMessage{  
    message_type =RESPONSE_QUERY_DISCHARGE_RESULT;  
    tbox_dischargectrl{  
        timestamp = 时间戳(从1970年到现在的秒数);  
        command= 开启关闭; //0: 关闭; 1: 开启  
        targetpower= 目标电量; //0-100  
    }  
}
```

5.37 TBOX 通知 CDCS 同步放电功能和 soc 电量

当APP设置SOC放电状态时, TBOX需把修改记录同步给CDCS

```
message TopMessage{  
    message_type = REQUEST_SYNC_DISCHARGE;  
    tbox_dischargectrl {  
        timestamp = 时间戳; //(从1970年到现在的秒数)  
        command= 开启关闭; //0: 关闭; 1: 开启  
        targetpower= 目标电量; //0-100  
    }  
}
```


	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 33 页 共 43 页

```

}
CDCS 回复 TBOX:
message TopMessage{
    message_type = REQUEST_SYNC_DISCHARGE_RESULT;
    msg_result{
        result=处理结果;  //true 为已执行, false 为失败
        error_code=错误代码提示;
    }
}

```

5.38 TBOX 通知 CDCS 辅助定位状态信息

当TBOX完成卫星定位后，TBOX需把辅助定位状态信息同步给CDCS

```

message TopMessage{
    message_type = REQUEST_SYNC_AGNSSINFO;
    agnss_info{
        timestamp = 时间戳;  //(从1970年到现在的秒数)
        downloadtime = xtra文件下载时间;  //(从1970年到现在的秒数)
        importtime = xtra文件注入定位芯片时间;  //(从1970年到现在的秒数)
        locationtime = 首次定位时间;  //(从1970年到现在的秒数)
        location = 当前定位状态;  //true:已定位; false:未定位
    }
}

```

```

CDCS 回复 TBOX:
message TopMessage{
    message_type = REQUEST_SYNC_AGNSSINFO_RESULT;
    msg_result{
        result=处理结果;  //true 为已执行, false 为失败
        error_code=错误代码提示;
    }
}

```

5.39 CDCS 向 TBOX 查询辅助定位状态信息

当CDCS的导航app启动后，向TBOX主动查询一次辅助定位状态信息

```

message TopMessage{
    message_type = REQUEST_QUERY_AGNSSINFO;

```

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 34 页 共 43 页

```
}
TBOX 回复 CDCS:
message TopMessage{
    message_type = REQUEST_QUERY_AGNSSINFO_RESULT;
    agnss_info{
        timestamp = 时间戳; //(从1970年到现在的秒数)
        downloadtime = xtra文件下载时间; //(从1970年到现在的秒数)
        importtime = xtra文件导入定位芯片时间; //(从1970年到现在的秒数)
        locationtime = 首次定位时间; //(从1970年到现在的秒数)
        location = 当前定位状态; //true:已定位; false:未定位
    }
}
```

5. 40 TBOX 通知 CDCS 同步网络状态信息

当TBOX判断到网络通道状态变化时，需要将状态同步给CDCS:

```
message TopMessage{
    message_type = REQUEST_SYNC_NETWORK_STATUS;
    network_status {
        timestamp = 时间戳; //(从1970年到现在的秒数)
        channel = APN类型; //0: APN1; 1: APN2(默认);
        networksts= 网络状态; //0: 异常; 1: 正常;
    }
}
```

CDCS回复TBOX同步结果:

```
message TopMessage{
    message_type = RESPONSE_SYNC_NETWORK_STATUS_RESULT;
    msg_result{
        result=处理结果; //true为已执行, false为失败
        error_code=错误代码提示;
    }
}
```

5. 41 CDCS 设置 TBOX 网络制式(预留)

```
message TopMessage{
    message_type = REQUEST_SET_SIGNALTYPE;
```

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 35 页 共 43 页

```

tbox_signal_type {
    timestamp = 时间戳; //(从1970年到现在的秒数)
    signaltype = 开启关闭; //0: AUTO(5G优先); 1: 锁定5G; 2: 锁定4G/5G;
                        //3: 锁定4G; 4: 锁定3G; 5: 锁定2G;
}
}
TBOX回复CDCS:
message TopMessage{
    message_type =RESPONSE_SET_SIGNALTYPE_RESULT;
    msg_result{
        result=处理结果; //true 为已执行, false 为失败
        error_code=错误代码提示;
    }
}

```

5.42 CDCS 获取 TBOX 网络制式(预留)

```

message TopMessage{
    message_type = REQUEST_QUERY_SIGNALTYPE;
}
TBOX回复CDCS:
message TopMessage{
    message_type =RESPONSE_QUERY_SIGNALTYPE_RESULT;
    tbox_signal_type {
        timestamp = 时间戳; //(从1970年到现在的秒数)
        signaltype = 开启关闭; //0: AUTO(5G优先); 1: 锁定5G; 2: 锁定4G/5G;
                        //3: 锁定4G; 4: 锁定3G; 5: 锁定2G;
    }
}

```

5.43 TBOX 通知 CDCS 同步网络制式(预留)

当TSP下发配置网络制式时，TBOX需把修改记录同步给CDCS

```

message TopMessage{
    message_type = REQUEST_SYNC_SIGNALTYPE;
    tbox_signal_type {
        timestamp = 时间戳; //(从1970年到现在的秒数)
        signaltype = 开启关闭; //0: AUTO(5G优先); 1: 锁定5G; 2: 锁定4G/5G;

```

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 36 页 共 43 页

//3: 锁定4G; 4: 锁定3G; 5: 锁定2G;

}

}

CDCS 回复 TBOX:

```
message TopMessage{
    message_type = REQUEST_SYNC_SIGNALTYPE_RESULT;
    msg_result{
        result=处理结果; //true 为已执行, false 为失败
        error_code=错误代码提示;
    }
}
```

5.44 TBOX 通知 CDCS 设置日志等级

当TSP下发设置日志等级时, TBOX需把日志等级设置同步给CDCS

```
message TopMessage{
    message_type = REQUEST_SYNC_LOGLEVEL;
    log_level{
        eventid = 事件ID;
        ecutype = ECU缩写;//多ECU之间通过英文字符逗号(,)分隔
        loglevel = 日志等级;//1:ERROR, 2:WARN, 3:INFO, 4:DEBUG, 5:VERBOSE
    }
}
```

CDCS 回复 TBOX:

```
message TopMessage{
    message_type = REQUEST_SYNC_LOGLEVEL_ACK;
    msg_result{
        result=处理结果; //true 为已执行, false 为失败
        error_code=错误代码提示;
    }
}
```

5.45 CDCS 给 TBOX 同步设置日志等级状态

当CDCS设置日志等级完成后, CDCS需要把设置成功失败状态同步给CDCS

```
message TopMessage{
    message_type = REQUEST_SYNC_LOGLEVEL_RESULT;
    log_level_result{
```

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 37 页 共 43 页

```
eventid = 事件ID;
ecutype = ECU缩写;//多ECU之间通过英文字符逗号（,）分隔
loglevelsettingresult = 处理结果; //0:设置成功, 1: 设置失败
}
```

TBOX 回复 CDCS:

```
message TopMessage{
    message_type = REQUEST_SYNC_LOGLEVEL_RESULT_ACK;
    msg_result{
        result=处理结果; //true 为已执行, false 为失败
        error_code=错误代码提示;
    }
}
```

5. 46 TBOX 通知 CDCS 设置日志指令

当TSP下发设置日志指令时，TBOX需把日志指令同步给CDCS

```
message TopMessage{
    message_type = REQUEST_SYNC_LOGCMD;
    log_cmd{
        eventid = 事件ID;
        ecutype = ECU缩写;//多ECU之间通过英文字符逗号（,）分隔
        packagename= 应用标识; //多应用之间通过英文字符逗号（,）分隔
        cmdtype=日志指令; //1:清理日志文件, 2: 暂停日志采集, 3: 恢复初始状态, 4--255 为自定义范围, 业务
        自行定义
    }
}
```

CDCS 回复 TBOX:

```
message TopMessage{
    message_type = REQUEST_SYNC_LOGCMD_ACK;
    msg_result{
        result=处理结果; //true 为已执行, false 为失败
        error_code=错误代码提示;
    }
}
```

5. 47 CDCS 给 TBOX 同步设置日志指令状态

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 38 页 共 43 页

当CDCS设置日志等级完成后，CDCS需要把设置成功失败状态同步给TBOX

```
message TopMessage{
    message_type = REQUEST_SYNC_LOGCMD_RESULT;
    log_cmd_result{
        eventid = 事件ID;
        ecutype = ECU缩写; //多ECU之间通过英文字符逗号（,）分隔
        packagename= 应用标识; //多应用之间通过英文字符逗号（,）分隔
        logsettingcmdresult = 处理结果; //0:设置成功, 1: 设置失败
    }
}
```

CDCS 回复 TBOX:

```
message TopMessage{
    message_type = REQUEST_SYNC_LOGCMD_RESULT_ACK;
    msg_result{
        result=处理结果; //true 为已执行, false 为失败
        error_code=错误代码提示;
    }
}
```

5. 48 TBOX 通知 CDCS 设置应用日志开关

当TSP下发设置应用日志开关时，TBOX需把应用日志开关同步给CDCS

```
message TopMessage{
    message_type = REQUEST_SYNC_LOGAPP;
    log_appsetting{
        eventid = 事件ID;
        ecutype = ECU缩写; //多ECU之间通过英文字符逗号（,）分隔
        packagename = 应用标识; //多应用之间通过英文字符逗号（,）分隔
        logcmdsetting = 日志开关; //0:关闭, 1:打开, 2--255 为自定义范围, 业务自行定义
    }
}
```

CDCS 回复 TBOX:

```
message TopMessage{
    message_type = REQUEST_SYNC_LOGAPP_ACK;
    msg_result{
        result=处理结果; //true 为已执行, false 为失败
        error_code=错误代码提示;
    }
}
```

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 39 页 共 43 页

}

5.49 CDCS 给 TBOX 同步设置应用日志开关状态

当CDCS设置应用日志开关状态完成后，CDCS需要把设置成功失败状态同步给TBOX

```
message TopMessage{
    message_type = REQUEST_SYNC_LOGAPP_RESULT;
    log_appsetting_result{
        eventid = 事件ID;
        ecutype = ECU缩写;//多ECU之间通过英文字符逗号（,）分隔
        packagename = 应用标识; //多应用之间通过英文字符逗号（,）分隔
        logcmdsettingresult = 处理结果; //0:设置成功, 1: 设置失败
    }
}
```

TBOX 回复 CDCS:

```
message TopMessage{
    message_type = REQUEST_SYNC_LOGAPP_RESULT_ACK;
    msg_result{
        result=处理结果; //true 为已执行, false 为失败
        error_code=错误代码提示;
    }
}
```

5.50 TBOX 通知 CDCS 采集日志并上传（新）

当TSP下发采集日志并上传时，TBOX需把采集日志的信息同步给CDCS。

注：如果TSP下发采集任务时，TBOX与CDCS之间socket链路未建立，TBOX需要将本次任务保存在本地，待判断到TBOX与CDCS的socket链路建立起来后，再将信息同步给CDCS；另外TBOX本地只保存最后一次收到的任务信息。

```
message TopMessage{
    message_type = REQUEST_SYNC_LOGCOLLECT;
    log_collect{
        eventid = 事件ID; //平台端生成
        logid = 日志采集ID; //平台端生成
        ecutype = ECU缩写;//多ECU之间通过英文字符逗号（,）分隔
        starttime = 日志开始时间; //采集日志开始的时间戳
        endtime = 日志结束时间; //采集日志结束的时间戳
        loglevel = 日志级别; //1:ERROR, 2:WARN, 3:INFO, 4:DEBUG, 5:VERBOSE
    }
}
```

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 40 页 共 43 页

validity = 日志采集有效期; //数据采集完后, 在有效期时间范围内, 满足上传条件下, 进行上传:
超出有效时间的, 不进行上传。有效期: 3600-43200 秒, 默认为 3600 秒 (1 小时)

```
packagename = 应用标识; //多应用之间通过英文字符逗号 (,) 分隔
}
```

CDCS 回复 TBOX:

```
message TopMessage{
    message_type = REQUEST_SYNC_LOGCOLLECT_ACK;
    msg_result{
        result=处理结果; //true 为已执行, false 为失败
        error_code=错误代码提示;
    }
}
```

5.51 CDCS 给 TBOX 同步采集日志状态 (新)

CDCS采集日志并上传完成后, 通知TBOX采集日志的状态

```
message TopMessage{
    message_type = REQUEST_SYNC_LOGCOLLECT_RESULT;
    log_collect_result{
        eventid = 事件ID;
        logid = 日志采集ID;
        ecutype = ECU缩写; //多ECU之间通过英文字符逗号 (,) 分隔
        packagename = 应用标识; //多应用之间通过英文字符逗号 (,) 分隔
        logcollectresult = 处理结果; //0: 采集失败, 1:采集完成
        Logcollecterrorcause = 日志采集失败原因; //日志采集结果为 FALSE 时, 反馈失败原因:
            //0x00: 采集成功
            //0x01: 没有采集到日志文件
            //0x02: 日志文件上传失败
            //0x03: 当前 ECU 或应用未响应
            //0x04: 当前 ECU 或应用不支持日志采集
            //0x05-0xFD: 保留错误码
            //0xFE: 整车休眠中
            //0xFF: 其他异常
    }
}
```


	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 41 页 共 43 页

TBOX 回复 CDCS:

```
message TopMessage{
    message_type = REQUEST_SYNC_LOGCOLLECT_RESULT_ACK;
    msg_result{
        result=处理结果; //true 为已执行, false 为失败
        error_code=错误代码提示;
    }
}
```

5.52 CDCS 向 TBOX 请求上传文件 URL (新)

CDCS向TBOX请求上传文件的URL地址, TBOX需要向云平台请求URL地址并同步给CDCS

```
message TopMessage{
    message_type = REQUEST_GET_UPLOAD_URL;
    get_upload_url{
        servicetype = 业务类型; //0x00-0x7F为url上传方式
                                //0x00:通用业务; 0x01:敏感数据; 0x02:管理平台; 0x3:OTA业务
                                //0x04:APP业务; 0x05:智能寻车; 0x06:哨兵模式; 0x7:生物检测
                                //0x08:车机埋点; 0x09:实时信息补充上报; 0x0A:主动日志采集
                                //0x0B:NLP语音; 0x0C:素材中心; 0x0D-0x7F:预留
                                //0x80-0xFF为stoken上传方式, 对应0x00-0x7F业务类型。
                                //如: 日志采集获取URL就使用0x8A:主动日志采集

        eventid = 事件ID; //依据TBOX同步的云平台id反馈
        filename = 文件名;//文件名, 多文件上传文件名必传, 文件名之间通过英文字符逗号(,)分隔。无
                    需包括目录, 不能出现“/”
    }
}
```

TBOX 回复 CDCS:

```
message TopMessage{
    message_type = REQUEST_GET_UPLOAD_URL_ACK;
    msg_result{
        result=处理结果; //true 为已执行, false 为失败
        error_code=错误代码提示;
    }
}
```

5.53 TBOX 向 CDCS 同步文件上传 URL (新)

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 42 页 共 43 页

当TBOX从云平台获取到文件上传的URL地址后，TBOX需把URL的信息同步给CDCS

```
message TopMessage{
    message_type = REQUEST_SYNC_UPLOAD_URL;
    sync_upload_url{
        validity = 有效时间; //数据采集完后，在有效期时间范围内，满足上传条件下，进行上传；超出
        有效时间的，不进行上传。有效期：3600-43200 秒，默认为 3600 秒（1 小时）
        fileurl = 文件上传 URL; //文件上传 URL，多个文件上传的，文件 URL 之间通过英文字符逗号（,）
        分隔
    }
}
```

CDCS 回复 TBOX:

```
message TopMessage{
    message_type = REQUEST_SYNC_UPLOAD_URL_ACK;
    msg_result{
        result=处理结果; //true 为已执行，false 为失败
        error_code=错误代码提示;
    }
}
```

6. 音频信号传输

6.1 RTP 协议概述

RTP全名是Real-time Transport Protocol（实时传输协议）。它是IETF提出的一个标准，对应的RFC文档为RFC3550（RFC1889为其过期版本）。RFC3550不仅定义了RTP，而且定义了配套的相关协议RTCP（Real-time Transport Control Protocol，即实时传输控制协议）。RTP用来为IP网上的语音、图像、传真等多种需要实时传输的多媒体数据提供端到端的实时传输服务。RTP为Internet上端到端的实时传输提供时间信息和流同步，但并不保证服务质量，服务质量由RTCP来提供。

RTP协议是一种基于UDP的传输协议，RTP本身并不能为按顺序传送数据包提供可靠的传送机制，也不提供流量控制或拥塞控制，它依靠RTCP提供这些服务。这样，对于那些丢失的数据包，不存在由于超时检测而带来的延时，同时，对于那些丢弃的包，也可以由上层根据其重要性来选择性的重传。比如，对于I帧、P帧、B帧数据，由于其重要性依次降低，故在网络状况不好的情况下，可以考虑在B帧丢失甚至P帧丢失的情况下不进行重传，这样，在客户端方面，虽然可能会有短暂的不清晰画面，但却保证了实时性的体验和要求。

6.2 使用开源库 ORTP

	TBOX 与 CDCS 之间基于 TCP 的 通讯协议	编号:
	座舱系统/智能研究院	第 43 页 共 43 页

ORTP 是一款开源软件, 实现了 RTP 与 RTCP 协议。openRTP, 用C实现的一个RTP库（其实还有C++实现的, JAVA等实现的）；实质是一个视频服务器, 工作时客户端和服务端实时传递视频数据；一般认为RTP工作在传输层, 但是其实RTP比TCP/UDP高一个层次。

ORTP开源详解参考:

<https://www.cnblogs.com/elisha-blogs/p/4029412.html>

<https://www.cnblogs.com/elisha-blogs/p/4029413.html>

https://blog.csdn.net/qq_40732350/article/details/88370782

附上ORTP开源库下载链接: <http://download.savannah.gnu.org/releases/linphone/ortp/sources/>

6.3 ORTP 移植

TBOX 与 CDCS 各自移植匹配系统的 ORTP 库。

TBOX 端移植参考 <https://blog.csdn.net/tainjau/article/details/80399872>, 安卓系统上也可以移植相应的 ORTP。