

CarLife动态链接库使用说明

修改记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | CarLife协议版本 | 修改内容简介 | 修改日期 | 修改人 |
| V0.1 | V0.10 | 创建 | 9th June 2015 | 刘才权 |
| V0.2 | V0.10 | 添加Video、Media、TTS、VR、Control通道的接口定义；  添加CarLifeLib动态库的测试； | 17th June 2015 | 刘才权 |
| V0.3 | V0.15 | 在VR通道添加对手机端数据接收的调用接口和回调注册接口（part 11-2， 11-3）；  增加了对VR通道新增加接口的测试（part 13-6）； | 1th September 2015 | 刘才权 |
| V0.4 | V0.19 | 增量添加命令通道接口：  0x00018025-0x00018039(part 7-1, part 7-3) | 14th September 2015 | 刘才权 |
| V0.5 | V0.19 | 增加IPv6支持；  兼容IPv4；  对于使用用户使用CarLifeLib不需要区分IPv6/IPv4; | 20th October 2015 | 刘才权 |
| V0.6 | V0.20 | 完善video通道测试描述 | 29th October 2015 | 刘才权 |
| V0.7 | V0.20 | 代码优化，避免通道反复重连导致的内存栈区溢出；  添加 CarLifeVehicleLib资源释放方法； | 10th December 2015 | 刘才权 |
| V0.8 | V1.03 | 增加消息0x00010040-0x00018050；  修改log机制，适配不同平台环境的log策略；  修复CVirtualShell::doCmd()方法bug；  修复MSG\_CMD\_MODULE\_STATUS（0x00010026）消息的处理bug； | 11th January 2016 | 刘才权 |
| V0.9 | V1.03 | 为Log打印添加关键字“####CarLifeLog:”；  添加通道数据接收内存越界处理；  添加修改记录自动更新脚本-changeRecoredTool.sh； | 4th February 2016 | 刘才权 |
| V0.10 | V1.03 | 更新NaviNextTurnInfo消息定义；  修复sendCtrlTouchSigleClick消息头定义错误；  增大doCmdWithResult接口返回值buffer；  优化Log打印机制； | 23th February 2016 | 刘才权 |
| V0.11 | V1.0.11 | 增加消息接口0x00010051-0x00018055；  增加使用于Linux车机的连接建立接口int connectionSetup(string mdIP, string interfaceName)；  修复MSG\_CMD\_MD\_AUTHEN\_RESULT消息的错误定义； | 4th March 2016 | 刘才权 |
| V0.12 | V1.0.11 | 修复CCarLifeLib.cpp文件中确实类作用域符的问题（在某些平台会导致编译错误）； | 16th March 2016 | 刘才权 |
|  |  |  |  |  |

目录

[1. 前言 5](#_Toc422251405)

[2. 环境搭建 5](#_Toc422251406)

[3. 代码结构 5](#_Toc422251407)

[4. libcarlifevehicle.so使用方法 7](#_Toc422251408)

[13-1 交叉编译protobuffer 7](#_Toc422251409)

[13-2 车机端CarLife交叉编译 7](#_Toc422251410)

[13-3 车机端CarLife运行环境搭建 7](#_Toc422251411)

[5. 动态库初始化接口定义 7](#_Toc422251412)

[6. 连接建立接口定义 8](#_Toc422251413)

[7. Command通道接口定义 8](#_Toc422251414)

[7-1 车机端向手机端发送数据接口 8](#_Toc422251415)

[7-2 车机端接收手机端发送数据接口 14](#_Toc422251416)

[7-3 手机端向车机端发送数据回调注册接口 14](#_Toc422251417)

[8. Video通道接口定义 20](#_Toc422251418)

[8-1 车机端接收手机端发送数据接口 20](#_Toc422251419)

[8-2 手机端向车机端发送数据回调注册接口 20](#_Toc422251420)

[9. Media通道接口定义 21](#_Toc422251421)

[9-1 车机端接收手机端发送数据接口 21](#_Toc422251422)

[9-2 手机端向车机端发送数据回调注册接口 21](#_Toc422251423)

[10. TTS通道接口定义 24](#_Toc422251424)

[10-1 车机端接收手机端发送数据接口 24](#_Toc422251425)

[10-2 手机端向车机端发送数据回调注册接口 24](#_Toc422251426)

[11. VR通道接口定义 25](#_Toc422251427)

[11-1 车机端向手机端发送数据接口 25](#_Toc422251428)

[12. Control通道接口定义 26](#_Toc422251429)

[12-1 车机端向手机端发送数据接口 26](#_Toc422251430)

[12-2 车机端接收手机端发送数据接口 28](#_Toc422251431)

[12-3 手机端向车机端发送数据回调注册接口 29](#_Toc422251432)

[13. CarLifeLib 动态库测试 29](#_Toc422251433)

[13-1 通道连接建立测试 29](#_Toc422251434)

[13-2 command通道测试 30](#_Toc422251435)

[13-3 video通道测试 31](#_Toc422251436)

[13-4 media通道测试 32](#_Toc422251437)

[13-5 tts通道测试 33](#_Toc422251438)

[13-6 vr通道测试 34](#_Toc422251439)

[13-7 control通道测试 34](#_Toc422251440)

# 前言

本文档是供车机端CarLife使用的动态链接库（libcarlifevehicle.so）说明文档。该动态链接库主要实现了基于《CarLife车机互联项目数据通信协议》中各数据通道（command channel, video channel, media channel, tts channel, vr channel, control channel）的通道建立和数据收发（包括数据包的报文封装和解析过程）。

百度提供动态链接库(libcarlifevehicle.so)的全部源码，不但可以帮助使用者更好的理解CarLife通信协议，同时，源码能够直接通过交叉编译后，集成到使用者的车机端CarLife项目中，加快车机端CarLife的开发。

# 环境搭建

CarLife的测试环境如表2-1所示：

表2- 1

|  |  |  |
| --- | --- | --- |
| 百度测试环境 | 版本 | 说明 |
| 运行系统 | Ubuntu 12.04 | Linux系统即可 |
| ProtoBuffer | 2.5.0 | 版本必须一致 |
| 交叉编译链 | QNX650 | 本示例中使用基于QNX 的armv7交叉编译工具链，可根据具体项目需求选择对应的交叉编译环境 |
|  |  |  |

# 代码结构

代码目录如图3-1所示：

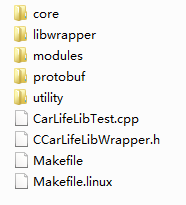


图3- 1

其中，

core：数据包报文封装和解析功能类，包含《CarLife车机互联项目数据通信协议》中对数据包报文封装和解析的全部操作。core中的内容如图3-2所示：

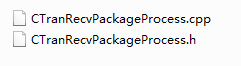


图3- 2

modules：包含通道连接建立功能类，cmmand通道数据收发功能类，video数据收发功能类，media数据收发功能类，tts数据收发功能类，vr（voice recognition）数据收发功能类，control数据收发功能类。用户可以使用这些功能类，实现各个通道的连接建立和数据收发操作。modules中的内容如图3-3所示：

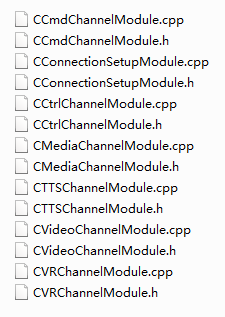


图3- 3

utility：包括底层套接字操作功能类和命令解析功能类，供Demo中其他的上层模块调用。utility中的内容如图3-4所示：

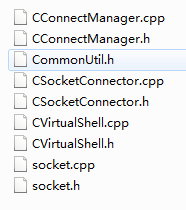


图3- 4

libwrapper：包括CCarLifeLib.h/CCarLifeLib.cpp，CCarLifeLib实现了对该动态链接库的内容封装，对外提供的接口。libwrapper中的内容如图3-5所示：

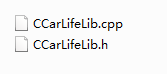


图3- 5

CCarLifeLibWrapper.h：该文件提供libcarlifevehicle.so对外暴露的所有接口和数据定义。

CarLifeLibTest.cpp：该文件为libcarlifevehicle.so的测试程序，提供了libcarlifevehicle.so的具体使用和测试步骤。

# libcarlifevehicle.so使用方法

## 交叉编译protobuffer

CarLife对数据的传输和接收采用了protobuffer机制进行报文封装和解析，因此，在使用libcarlifevehicle.so之前需要对protobuffer源码进行交叉编译，获得protobuffer的动态连接库文件libprotobuf.so.8文件。

## 车机端CarLife交叉编译

将交叉编译后的libprotobuf.so.8和libcarlifevehicle.so以及对应的头文件（CCarLifeLibWrapper.h）拷贝到车机系统的交叉编译环境中，并设置好编译和链接路径，同车机端CarLife项目一起进行交叉编译。

## 车机端CarLife运行环境搭建

将交叉编译后的libprotobuf.so.8和libcarlifevehicle.so文件拷贝到车机系统的运行环境中，并设置好链接路径。

# 动态库初始化接口定义及资源释放方法

动态库初始化接口定义如表5-1：

表5- 1

|  |  |  |
| --- | --- | --- |
| 函数定义 | CCarLifeLib\* getInstance() | 获取动态库实例对象 |
| 参数说明 | 无 |  |
| 返回值 | CCarLifeLib\* | 动态库实例对象指针 |
|  | | |
| 函数定义 | int carLifeLibInit() | 动态库初始化 |
| 参数说明 | 无 |  |
| 返回值 | 0：success  -1：failed |  |
|  |  |  |

在需要停止使用CarLifeVehicleLib时，只需通过delete方法释放CarLifeVehicleLib对象即可，具体使用方法如下：

|  |
| --- |
| //release CarLifeVehicleLib resources  CCarLifeLib \*pCCarLifeLib=CCarLifeLib::getInstance();  delete pCCarLifeLib; |

# 连接建立接口定义

动态库初始化接口定义如表6-1：

表6- 1

|  |  |  |
| --- | --- | --- |
| 函数定义 | int connectionSetup() | 车机端与手机端通道连接建立（适用于ADB） |
| 参数说明 | 无 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int connectionSetup(string mdIP) | 车机端与手机端通道连接建立（适用于QNX） |
| 参数说明 | mdIP：手机端IP地址 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int connectionSetup(string mdIP, string interfaceName) | 车机端与手机端通道连接建立（适用于Linux） |
| 参数说明 | mdIP：手机端IP地址  interfaceName：网卡节点名（可通过ifconfig查询获取） |  |
| 返回值 | 0：success  -1：failed |  |

# Command通道接口定义

## 车机端向手机端发送数据接口

车机端向手机端发送数据接口定义如表7-1：

表7- 1

|  |  |  |
| --- | --- | --- |
| 函数定义 | int cmdHUProtoclVersion  (S\_HU\_PROTOCOL\_VERSION\* version) | 车机端将自身的协议版本信息发送给手机端（对应消息0x00018001） |
| 参数说明 | version：车机端协议版本信息 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdHUInfro(S\_HU\_INFO\* huInfo) | 车机端将自身车身系统信息发送给手机（对应消息0x00018003） |
| 参数说明 | huInfo： 车机端车身系统信息 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdHUBTOOBInfro  (S\_HU\_BT\_OOB\_INFO\* oobInfo) | 车机端将自身蓝牙信息发送给手机，（对应消息0x00018005） |
| 参数说明 | oobInfo：车机端蓝牙信息 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdVideoEncoderInit  (S\_VIDEO\_ENCODER\_INIT\* initParam | 车机端通知手机端初始化视频编码器（对应消息0x00018007） |
| 参数说明 | initParam：视频编码器初始化参数 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdVideoEncoderStart() | 车机端通知手机端视频编码器开始发送视频（对应消息0x00018009） |
| 参数说明 | 无 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdVideoEncoderPause() | 车机端通知手机端视频编码器暂停发送视频（对应消息0x0001800A） |
| 参数说明 | 无 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdVideoEncoderReset() | 车机端通知手机端视频编码器reset（对应消息0x0001800B） |
| 参数说明 | 无 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdVideoEncoderFrameRateChange  (S\_VIDEO\_ENCODER\_FRAME\_RATE\_CHANGE\* videoParam) | 车机端通知手机端视频编码器改变帧率（对应消息0x0001800C） |
| 参数说明 | videoParam：帧率改变信息 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| ~~函数定义~~ | ~~int cmdPauseMedia()~~ | ~~车机端通知手机端停止Media播放（对应消息0x0001800E，该消息已废弃）~~ |
| ~~参数说明~~ | ~~无~~ |  |
| ~~返回值~~ | ~~0：success~~  ~~-1：failed~~ |  |
|  | | |
| 函数定义 | int cmdCarVelocity  (S\_CAR\_VELOCITY\* carVelocity) | 车机端将车身车速信息发送给手机端（对应消息 0x0001800F） |
| 参数说明 | carVelocity：车身车速信息 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdCarGPS(S\_CAR\_GPS\* gps) | 车机端将车身GPS信息发送给手机端（对应消息0x00018010） |
| 参数说明 | gps：车身GPS信息 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdCarGyroscope(S\_CAR\_GYROSCOPE\* gyro) | 车机端将车身陀螺仪信息发送给手机端（对应消息0x00018011） |
| 参数说明 | gyro：车身陀螺仪信息 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdCarAcceleration  (S\_CAR\_ACCELERATION\* acceleration) | 车机端将车身加速度信息发送给手机端（对应消息0x00018012） |
| 参数说明 | acceleration：车身加速度信息 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdCarOil() | 车机端将车身油量信息发送给手机端（对应消息0x00018013） |
| 参数说明 | Reserved |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdLaunchModeNormal() | 车机通知手机以正常模式启动（对应消息0x0001801D） |
| 参数说明 | 无 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdLaunchModePhone() | 车机通知手机以电话模式启动（对应消息0x0001801E） |
| 参数说明 | 无 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdLaunchModeMap() | 车机通知手机以地图模式启动（对应消息0x0001801F） |
| 参数说明 | 无 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdLaunchModeMusic() | 车机通知手机以音乐模式启动（对应消息0x00018020） |
| 参数说明 | 无 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdGoToForeground() | 车机端通知手机端CarLife到前台运行（对应消息0x00018025） |
| 参数说明 | 无 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdStatisticInfo(S\_STATISTICS\_INFO\* ) | 车机端发送给手机端的统计信息，其中包括觉得CarLife车机Log显示的信息（对应消息0x00018027） |
| 参数说明 | S\_STATISTICS\_INFO\* | 车机端统计相关信息 |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdModuleControl(S\_MODULE\_STATUS\_CONTROL\*) | 车机通知手机端音乐，语音，导航等改变当前状态（对应消息0x00018028） |
| 参数说明 | S\_MODULE\_STATUS\_CONTROL\* | 手机端需要状态改变的模块信息 |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdCarDataGear(S\_GEAR\_INFO\*) | 车机端通知手机端倒车状态（对应消息0x00018029） |
| 参数说明 | S\_GEAR\_INFO\* | 倒车状态 |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdCarDataSubscribeDone(S\_VEHICLE\_INFO\_LIST\*) | 车机端反馈给手机端车机端的信息订阅情况（对应消息0x00018032） |
| 参数说明 | S\_VEHICLE\_INFO\_LIST\* | 车身信息订阅情况 |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdBtHfpIndication(S\_BT\_HFP\_INDICATION\*) | 车机发送给移动设备，用来告诉移动设备蓝牙电话的状态信息。其中需要指明该状态是来自于那个蓝牙设备（地址）。主要考虑到有的车机能够支持同时连接多个蓝牙设备（对应消息0x00018041） |
| 参数说明 | S\_BT\_HFP\_INDICATION\* | 电话状态 |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdBtHfpConnection(S\_BT\_HFP\_CONNECTION\*) | 车机发送该消息到移动设备用来告诉移动设备关于蓝牙的连接状态，依据HFP规范（对应消息0x00018042） |
| 参数说明 | S\_BT\_HFP\_CONNECTION\* | 蓝牙连接状态 |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdCarLifeDataSubscribe  (S\_SUBSCRIBE\_MOBILE\_CARLIFE\_INFO\_LIST\*) | 车机发送给移动设备的订阅移动设备CarLife相关数据的消息（对应消息0x00018043） |
| 参数说明 | S\_SUBSCRIBE\_MOBILE\_CARLIFE\_INFO\_LIST\* | 需要订阅的手机信息 |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdCarLifeDataSubscribeStart  (S\_SUBSCRIBE\_MOBILE\_CARLIFE\_INFO\_LIST\*) | 车机端要求手机端开始发送订阅的相关消息（对应消息0x00018045） |
| 参数说明 | S\_SUBSCRIBE\_MOBILE\_CARLIFE\_INFO\_LIST\* | 需要开始接收的手机信息 |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdCarLifeDataSubscribeStop  (S\_SUBSCRIBE\_MOBILE\_CARLIFE\_INFO\_LIST\*) | 车机端要求手机端停止发送订阅的相关消息（对应消息0x00018046） |
| 参数说明 | S\_SUBSCRIBE\_MOBILE\_CARLIFE\_INFO\_LIST\* | 需要手机停止发送的订阅消息 |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdHuAuthenRequest(S\_AUTHEN\_REQUEST\*) | 车机CarLife在本地随机产生一个随机字符串，并将该值发送给手机端CarLife（对应消息0x00018048） |
| 参数说明 | S\_AUTHEN\_REQUEST\* | 车机端产生的随机码字符串信息 |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdHuAuthenResult(S\_HU\_AUTHEN\_RESULT\*) | 车机端CarLife将验证结果发送给手机端CarLife（对应消息0x0001804A） |
| 参数说明 | S\_HU\_AUTHEN\_RESULT\* | 认证结果 |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdBTHfpResponse(S\_BT\_HFP\_RESPONSE\*) | 车机端在处理完成HFP的请求后，需要通过该消息告诉手机该命令的执行结果（对应消息0x0001804E） |
| 参数说明 | S\_BT\_HFP\_RESPONSE\* | 手机端HFP请求结果 |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdBTHfpStatusResponse  (S\_BT\_HFP\_STATUS\_RESPONSE\*) | 当车机收到手机请求状态时,需要通过该消息将指定的状态反馈给手机端。目前主要用来反馈车机MIC的状态（对应消息0x00018050） |
| 参数说明 | S\_BT\_HFP\_STATUS\_RESPONSE\* | 反馈给手机的申请状态信息 |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdFeatureConfigResponse(S\_FEATURE\_CONFIG\_LIST\*) | 车机端根据自身情况进行功能定制，并将定制信息发送给手机（对应消息0x00018052） |
| 参数说明 | S\_FEATURE\_CONFIG\_LIST\* | 车机端配置信息 |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdBTStartIdentifyReq(S\_BT\_START\_IDENTIFY\_REQ\*) | 车机发送该消息来让手机开始蓝牙标识过程（对应消息0x00018053） |
| 参数说明 | S\_BT\_START\_IDENTIFY\_REQ\* | 请求信息 |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int cmdErrorCode(S\_ERROR\_CODE\*) | 车机将错误码统计信息发送给手机（对应消息0x00018055） |
| 参数说明 | S\_ERROR\_CODE\* | 错误码信息 |
| 返回值 | 0：success  -1：failed |  |
|  |  |  |
|  |  |  |
|  | | |

## 车机端接收手机端发送数据接口

车机端接收手机端发送数据接口如表7-2所示：

表7- 2

|  |  |  |
| --- | --- | --- |
| 函数定义 | int cmdReceiveOperation() | 车机端CarLife调用该方法实现对手机端命令通道数据的接收 |
| 参数说明 | 无 |  |
| 返回值 | 0：success  -1：failed |  |

## 手机端向车机端发送数据回调注册接口

手机端向车机端发送数据回调注册接口如表7-3所示：

表7- 3

|  |  |  |
| --- | --- | --- |
| 函数定义 | void cmdRegisterProtocolVersionMatchStatus  (void (\*pFunc)(S\_PROTOCOL\_VERSION\_MATCH\_SATUS\*)); | 手机端发送给车机端的协议版本匹配结果（对应消息0x00010002） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterMDInfro(void (\*pFunc)(S\_MD\_INFO\*)) | 手机端发送给车机端的手机系统信息（对应消息0x00010004） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterMDBTOOBInfro(void (\*pFunc)(S\_MD\_BT\_OOB\_INFO\*)) | 手机端发送给车机端的手机蓝牙信息（对应消息0x00010006） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterVideoEncoderInitDone(void (\*pFunc)(S\_VIDEO\_ENCODER\_INIT\_DONE\*)) | 手机端通知车机端视频编码器初始化完成（对应消息0x00010008） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterVideoEncoderFrameRateChangeDone(void (\*pFunc)  (S\_VIDEO\_ENCODER\_FRAME\_RATE\_CHANGE\_DONE\*)) | 手机端通知车机端视频编码器帧率调整完成（对应消息0x0001000D） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | oid cmdRegisterTelStateChangeIncoming(void (\*pFunc)(void)) | 手机端通知车机端有电话接入（对应消息0x00010014） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterTelStateChangeOutGoing(void (\*pFunc)(void)) | 手机端通知车机端有电话打出（对应消息0x00010015） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterTelStateChangeIdle(void (\*pFunc)(void)) | 手机端通知车机端电话空闲（对应消息0x00010016） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterTelStateChangeInCalling(void (\*pFunc)(void)) | 手机端通知车机端手机处于通话中（对应消息0x00010017） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterScreenOn(void (\*pFunc)(void)) | 手机端通知车机端手机屏幕点亮（对应消息0x00010018） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterScreenOff(void (\*pFunc)(void)) | 手机端通知车机端手机屏幕关闭（对应消息0x00010019） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterScreenUserPresent(void (\*pFunc)(void)) | 手机端通知车机端手机屏幕解锁（对应消息0x0001001A） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterForeground(void (\*pFunc)(void)) | 手机端通知车机端手机CarLife处于前台（对应消息0x0001001B） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterBackground(void (\*pFunc)(void)) | 手机端通知车机端手机CarLife处于后台（对应消息0x0001001C） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterGoToDeskTop(void (\*pFunc)(void)) | 手机端通知车机端手机恢复到主界面（对应消息0x00010021） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterMicRecordWakeupStart(void (\*pFunc)(void)) | 手机端通知车机端手机开始VR的唤醒录音（对应消息0x00010022） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterMicRecordEnd(void (\*pFunc)(void)) | 手机端通知车机端手机结束VR的录音（对应消息0x00010023） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterMicRecordRecogStart(void (\*pFunc)(void)) | 手机端通知车机端手机开始VR的识别后录音（对应消息0x00010024） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterModuleStatus(void (\*pFunc)(S\_MODULE\_STATUS\_LIST\_MOBILE\*)) | 手机端通知车机端音乐、导航、电话、VR等模块的状态（对应消息0x00010026） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterNaviNextTurnInfo(void (\*pFunc)(S\_NAVI\_NEXT\_TURN\_INFO\*)) | 手机端通知车机端导航在路况转向的信息（对应消息0x00010030） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterCarDataSubscribe(void (\*pFunc)(S\_VEHICLE\_INFO\_LIST\*)) | 手机端通知车机端需要订阅的车身信息（对应消息0x00010031） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterCarDataSubscribeStart(void (\*pFunc)(S\_VEHICLE\_INFO\_LIST\*)) | 手机端通知车机端开始发送订阅信息（对应消息0x00010033） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterCarDataSubscribeStop(void (\*pFunc)(S\_VEHICLE\_INFO\_LIST\*)) | 手机端通知车机端停止发送车身订阅信息（对应消息0x00018034） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterMediaInfo(void (\*pFunc)(S\_MEDIA\_INFO\*)) | 手机端通知车机端当前的媒体播放信息（对应消息0x00010035） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterMediaProgressBar(void (\*pFunc)(S\_MEDIA\_PROGRESS\_BAR\*)) | 手机端通知车机端媒体播放进度条信息（对应消息0x00010036） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | |  |
| 函数定义 | void cmdRegisterConnectException(void (\*pFunc)(S\_CONNECTION\_EXCEPTION\*)) | 手机端通知车机端连接异常信息（对应消息0x00010037） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterRequestGoToForeground(void (\*pFunc)(void)) | 手机端通知车机端将手机端CarLife提升到前台运行（对应消息0x00010038） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  |  |  |
| 函数定义 | void cmdRegisterUIActionSound(void (\*pFunc)(void)) | 手机端通知车机端发出焦点点中通知声（对应消息0x00010039） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterBtHfpRequest(void (\*pFunc)(S\_BT\_HFP\_REQUEST\*)) | 车机发送给移动设备，用来告诉移动设备蓝牙电话的状态信息。其中需要指明该状态是来自于那个蓝牙设备（地址）。主要考虑到有的车机能够支持同时连接多个蓝牙设备（对应消息0x00010040） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterCarLifeDataSubscribeDone(void (\*pFunc)(S\_SUBSCRIBE\_MOBILE\_CARLIFE\_INFO\_LIST\*)) | 移动设备针对车机订阅CarLife相关数据的反馈信息（对应消息0x00010044） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterNaviAssistantGuideInfo(void (\*pFunc)(S\_NAVI\_ASSITANT\_GUIDE\_INFO\*)) | 移动设备发送给车机的导航辅助诱导信息。该消息用于车机在主频之外的其他的屏幕显示相关的辅助诱导信息（对应消息0x00010047） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterMdAuthenResponse(void (\*pFunc)(S\_AUTHEN\_RESPONSE\*)) | 手机CarLife将接收到的车机随机数使用私有密钥进行加密（对应消息0x00010049） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterMdAuthenResult(void (\*pFunc)(S\_MD\_AUTHEN\_RESULT\*)) | 手机端CarLife将验证结果发送给车机端CarLife（对应消息0x0001804B） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterGotoForgroundResponse(void (\*pFunc)(void)) | 车机端发送MSG\_CMD\_GO\_TO\_FOREGROUND消息给移动设备端将CarLife切回到前台以后，移动设备端如果在5秒内没有切换成功则发送该条消息（对应消息0x0001004C） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterStartBtAutoPairRequest(void (\*pFunc)(S\_BT\_START\_PAIR\_REQ\*)) | 手机端发送给车机，断开已经存在的HFP连接，并开始蓝牙自动匹配的流程。如果是IOS则不需要开始自动匹配流程（对应消息0x0001004D） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterBTHfpStatusRequest(void (\*pFunc)(S\_BT\_HFP\_STATUS\_REQUEST\*)) | 手机发送该消息来获取指定的状态，目前主要用来获取车机MIC的状态（对应消息0x0001004F） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterFeatureConfigRequest(void (\*pFunc)(void)) | 双方握手认证通过后，手机端会发送该消息来请求车机端的功能定制项（对应消息0x00010051） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void cmdRegisterBTIdentifyResultInd(void (\*pFunc)(S\_BT\_INDENTIFY\_RESULT\_IND\*)) | 手机发送给车机，告诉车机 蓝牙标识的结果，同时当标识状态发生变化时，通知车机（对应消息0x00018054） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  |  |  |
|  |  |  |

# Video通道接口定义

## 车机端接收手机端发送数据接口

车机端接收手机端发送数据接口如表8-1所示：

表8- 1

|  |  |  |
| --- | --- | --- |
| 函数定义 | int videoReceiveOperation() | 车机端CarLife调用该方法实现对手机端Video通道数据的接收 |
| 参数说明 | 无 |  |
| 返回值 | 0：success  -1：failed |  |

## 手机端向车机端发送数据回调注册接口

手机端向车机端发送数据回调注册接口如表8-2所示：

表8- 2

|  |  |  |
| --- | --- | --- |
| 函数定义 | void videoRegisterDataReceive(void (\*pFunc)(u8 \*data, u32 len)) | 车机端接收手机端发送的视频数据（对应消息0x00020001） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void videoRegisterHeartBeat(void (\*pFunc)(void)) | 车机端接收手机端发送的心跳数据包（对应消息0x00020002） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  |  |  |

# Media通道接口定义

## 车机端接收手机端发送数据接口

车机端接收手机端发送数据接口如表9-1所示：

表9- 1

|  |  |  |
| --- | --- | --- |
| 函数定义 | int mediaReceiveOperation() | 车机端CarLife调用该方法实现对手机端Media通道数据的接收 |
| 参数说明 | 无 |  |
| 返回值 | 0：success  -1：failed |  |

## 手机端向车机端发送数据回调注册接口

手机端向车机端发送数据回调注册接口如表9-2所示：

表9- 2

|  |  |  |
| --- | --- | --- |
| 函数定义 | void mediaRegisterInit(void (\*pFunc)(S\_AUDIO\_INIT\_PARAMETER\*)) | 手机端发送给车机端的media初始化信息（对应消息0x00030001） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void mediaRegisterNormalData(void (\*pFunc)(u8 \*data, u32 len)) | 车机端接收手机端发送的media数据（对应消息0x00030006） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| ~~函数定义~~ | ~~void mediaRegisterStop(void (\*pFunc)(void))~~ | ~~手机端通知车机端media停止状态（已废弃）~~ |
| ~~参数说明~~ | ~~pFunc:被回调函数指针~~ |  |
| ~~返回值~~ | ~~无~~ |  |
|  | | |
| 函数定义 | void mediaRegisterPause(void (\*pFunc)(void)) | 手机端通知车机端media暂停状态 |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void mediaRegisterResume(void (\*pFunc)(void)) | 手机端通知车机端media恢复播放状态（对应消息0x00030003） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| ~~函数定义~~ | ~~void mediaRegisterSeek(void (\*pFunc)(void))~~ | 手机端通知车机端media seek状态（已废弃） |
| ~~参数说明~~ | ~~pFunc:被回调函数指针~~ |  |
| ~~返回值~~ | ~~无~~ |  |

# TTS通道接口定义

## 车机端接收手机端发送数据接口

车机端接收手机端发送数据接口如表10-1所示：

表10- 1

|  |  |  |
| --- | --- | --- |
| 函数定义 | int ttsReceiveOperation() | 车机端CarLife调用该方法实现对手机端TTS通道数据的接收 |
| 参数说明 | 无 |  |
| 返回值 | 0：success  -1：failed |  |

## 手机端向车机端发送数据回调注册接口

手机端向车机端发送数据回调注册接口如表10-2所示：

表10- 2

|  |  |  |
| --- | --- | --- |
| 函数定义 | void ttsRegisterInit(void (\*pFunc)(S\_AUDIO\_INIT\_PARAMETER\*)) | 手机端发送给车机端的tts初始化信息（对应消息0x00040001） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void ttsRegisterNormalData(void (\*pFunc)(u8 \*data, u32 len)) | 车机端接收手机端发送的tts数据（对应消息0x00040003） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void ttsRegisterStop(void (\*pFunc)(void)) | 手机端通知车机端tts停止状态（对应消息0x00040002） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |

# VR通道接口定义

## 车机端向手机端发送数据接口

车机端向手机端发送数据接口定义如表11-1：

表11- 1

|  |  |  |
| --- | --- | --- |
| 函数定义 | int sendVRRecordData(u8\* data, u32 size, u32 timeStamp) | 车机将MIC录制的音频数据传输到手机端（对应消息0x00058001） |
| 参数说明 | data：MIC录制的音频数据；  size：音频数据大小（单位byte）；  timestamp：音频数据时间戳； |  |
| 返回值 | 0：success  -1：failed |  |

## 车机端接收手机端发送数据接口

车机端接收手机端发送数据接口如表11-2所示：

表11- 2

|  |  |  |
| --- | --- | --- |
| 函数定义 | int vrReceiveOperation() | 车机端CarLife调用该方法实现对手机端VR通道数据的接收 |
| 参数说明 | 无 |  |
| 返回值 | 0：success  -1：failed |  |

## 手机端向车机端发送数据回调注册接口

手机端向车机端发送数据回调注册接口如表11-3：

表11- 3

|  |  |  |
| --- | --- | --- |
| 函数定义 | void vrRegisterInit(void (\*pFunc)(S\_AUDIO\_INIT\_PARAMETER\*)) | 手机端发送给车机端的vr初始化信息（对应消息0x00050002） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void vrRegisterNormalData(void (\*pFunc)(u8 \*data, u32 len)) | 车机端接收手机端发送的vr数据（对应消息0x00050003） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |
|  | | |
| 函数定义 | void vrRegisterStop(void (\*pFunc)(void)) | 手机端通知车机端vr停止状态（对应消息0x00050004） |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |

# 12.Control通道接口定义

## 车机端向手机端发送数据接口

车机端向手机端发送数据接口定义如表12-1：

表12- 1

|  |  |  |
| --- | --- | --- |
| ~~函数定义~~ | ~~int ctrlTouchAction(S\_TOUCH\_ACTION\* touchAction)~~ | ~~车机端发送给手机端的车机触屏事件（已废弃）~~ |
| ~~参数说明~~ | ~~touchAction：车机触屏事件~~ |  |
| ~~返回值~~ | ~~0：success~~  ~~-1：failed~~ |  |
|  | | |
| 函数定义 | int ctrlTouchActionDown(S\_TOUCH\_ACTION\_DOWN\* touchActionDown) | 车机端发送给手机端的车机触屏按下事件（对应消息0x00068002） |
| 参数说明 | touchActionDown：触摸屏按下动作信息 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int ctrlTouchActionUp(S\_TOUCH\_ACTION\_UP\* touchActionUp) | 车机端发送给手机端的车机触屏抬起事件（对应消息0x00068003） |
| 参数说明 | touchActionUp：触摸屏抬起动作信息 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int ctrlTouchActionMove(S\_TOUCH\_ACTION\_MOVE\* touchActionMove) | 车机端发送给手机端的车机触屏移动事件（对应消息0x00068004） |
| 参数说明 | touchActionMove：触摸屏移动信息 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int ctrlTouchSigleClick(S\_TOUCH\_SIGNAL\_CLICK\* touchSingleClick) | 车机端发送给手机端的车机触屏单击事件（对应消息0x00068005） |
| 参数说明 | touchSingleClick：触摸屏单击事件信息 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int ctrlTouchDoubleClick(S\_TOUCH\_DOUBLE\_CLICK\* touchDoubleClick) | 车机端发送给手机端的车机触屏双击事件（对应消息0x00068006） |
| 参数说明 | touchDoubleClick：触摸屏双击事件信息 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int ctrlTouchLongPress(S\_TOUCH\_LONG\_PRESS\* touchLongPress) | 车机端发送给手机端的车机触屏长按事件（对应消息0x00068007） |
| 参数说明 | touchLongPress：触摸屏长按事件 |  |
| 返回值 | 0：success  -1：failed |  |
|  | | |
| 函数定义 | int ctrlTouchCarHardKeyCode  (S\_TOUCH\_CAR\_HARD\_KEY\_CODE\* touchCarHardKeyCode) | 车机端发送给手机端的车机硬按键信息（对应消息0x00068008） |
| 参数说明 | touchCarHardKeyCode：车机硬按键信息 |  |
| 返回值 | 0：success  -1：failed |  |

## 车机端接收手机端发送数据接口

车机端接收手机端发送数据接口如表12-2所示：

表12- 2

|  |  |  |
| --- | --- | --- |
| 函数定义 | int ctrlReceiveOperation() | 车机端CarLife调用该方法实现对手机端Control通道数据的接收 |
| 参数说明 | 无 |  |
| 返回值 | 0：success  -1：failed |  |

## 手机端向车机端发送数据回调注册接口

手机端向车机端发送数据回调注册接口如表12-3所示：

表12- 3

|  |  |  |
| --- | --- | --- |
| 函数定义 | void ctrlRegisterUIActionSound(void (\*pFunc)(void)) | 手机端通知车机端UI控件被选中 |
| 参数说明 | pFunc:被回调函数指针 |  |
| 返回值 | 无 |  |

# CarLifeLib 动态库测试

CarLifeLib动态库的测试主要包括通道连接建立，command通道数据收发测试，video通道数据收发测试，media通道数据收发测试，tts通道数据收发测试，vr通道数据收发测试，control通道数据收发测试等七项。

## 通道连接建立测试

测试步骤：

1. 使用Wi-Fi连接手机和测试台机（手机和台机处于同一网段）；
2. 启动手机端CarLife；
3. 运行测试台机上的libtest测试程序；

预期结果：连接状态图标将由图13-1变为图13-2，

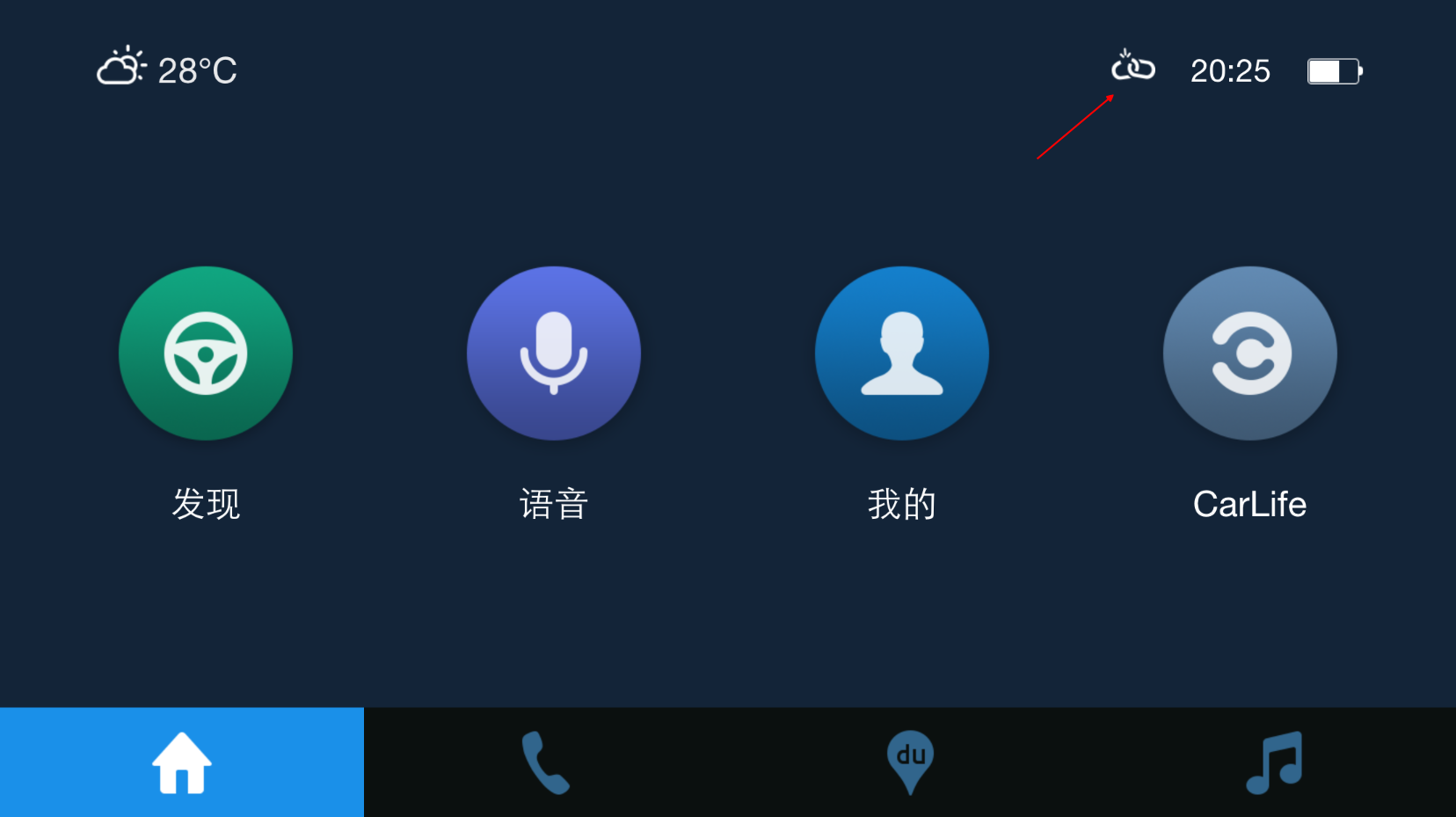


图13- 1

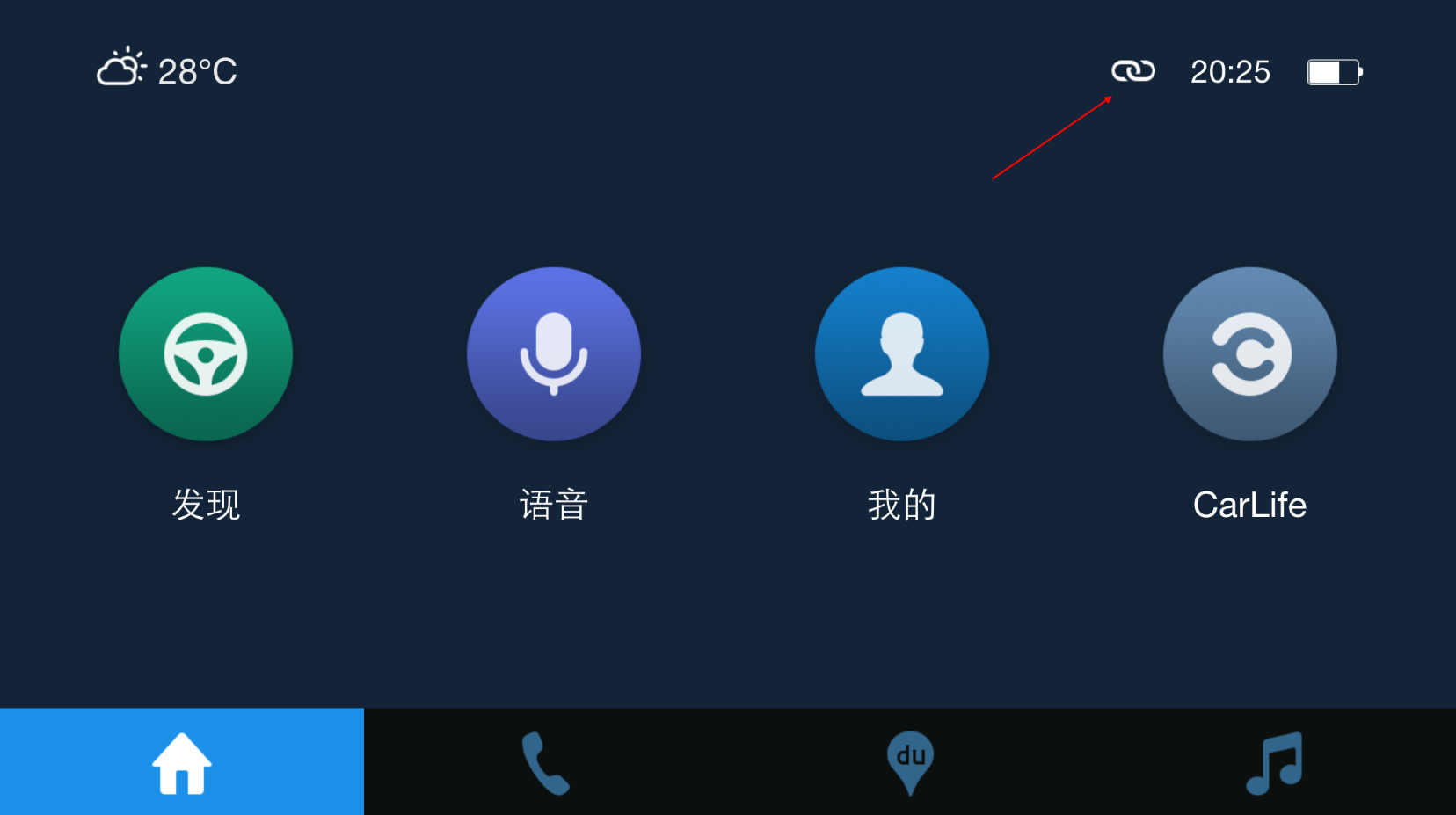


图13- 2

测试说明：当command通道，video通道，media通道，tts通道，vr通道，control通道全部建立连接成功后，连接图标将会发生变化。

## command通道测试

HU->MD/MD->HU数据通路测试

测试步骤：

1. 使用Wi-Fi连接手机和测试台机（手机与台机处于同一网段）；
2. 启动手机端CarLife；
3. 运行测试台机上的libtest测试程序；
4. 在程序的输入选项中选择1；
5. 观察libtest打印的log；

预期结果：如图13-3所示，

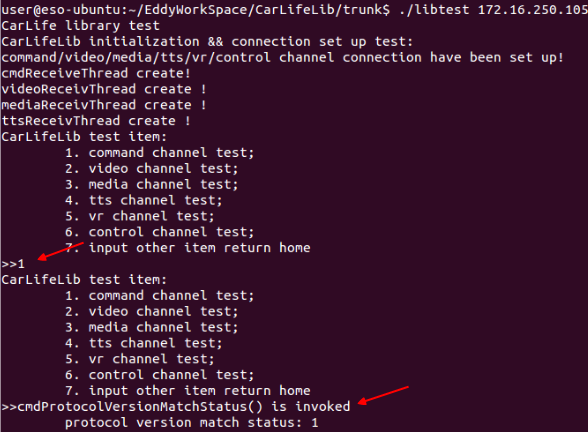


图13- 3

测试说明：手机和车机CarLife建立连接后，车机端将自身版本信息发送给手机（MSG\_CMD\_HU\_PROTOCOL\_VERSION=0x00018001），手机验证版本信息后将验证结果发送给车机（MSG\_CMD\_PROTOCOL\_VERSION\_MATCH\_STATUS=0x00010002）。

## video通道测试

测试步骤：

1. 使用Wi-Fi连接手机和测试台机（手机与台机处于同一网段）；
2. 对应Android 5.0以下版本手机，需要手动启动BDSC：
   * 1. 将对应版本bdsc通过adb push到手机的/data/local/tmp目录下，如Android 4.1:

#adb push bdsc16 /data/local/tmp/

* + 1. 通过adb命令运行bdsc；

bdsc版本对应关系如下：

|  |  |
| --- | --- |
| bdsc版本 | Android版本 |
| Version 16 | 4.1 |
| Version 17 | 4.2 |
| Version 18 | 4.3 |
| Version 19 | 4.4.2 |
| Version 19\_01 | 4.4.4 |

1. 启动手机端CarLife；
2. 运行测试台机上的libtest测试程序；
3. 在程序的输入选项中，选择输入2；
4. 观察libtest的运行log；

预期结果：如图13-4所示，

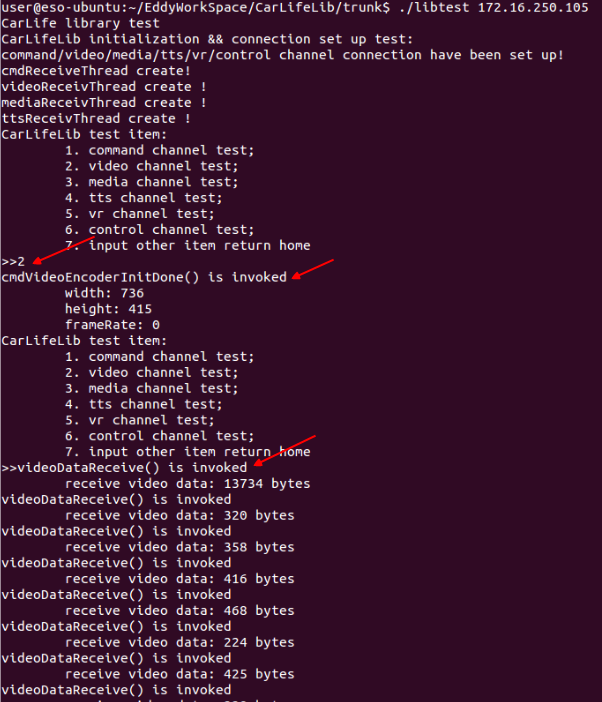


图13- 4

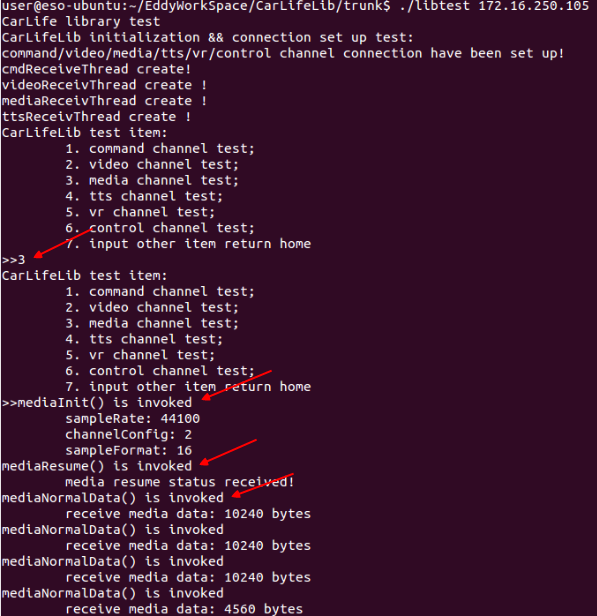
测试说明：手机端CarLife和车机端CarLife在command通道完成对应交互后，车机端CarLife就能从video通道接收video数据。

## media通道测试

测试步骤：

1. 使用Wi-Fi连接手机和测试台机（手机与台机处于同一网段）；
2. 启动手机端CarLife；
3. 运行测试台机上的libtest测试程序；
4. 在程序的输入选项中，选择输入3；
5. 播放手机端音乐；
6. 观察libtest运行的log；

预期结果：如图13-5所示，

图13- 5

测试说明：音乐开始播放后，手机端CarLife将会通过media通道将音乐的初始化参数和数据发送给车机端的CarLife。

## tts通道测试

测试步骤：

1. 使用Wi-Fi连接手机和测试台机（手机与台机处于同一网段）；
2. 启动手机端CarLife；
3. 运行测试台机上的libtest测试程序；
4. 在程序的输入选项中，选择输入4；
5. 开始手机端导航；
6. 观察libtest运行的log；

预期结果：如图13-6所示，

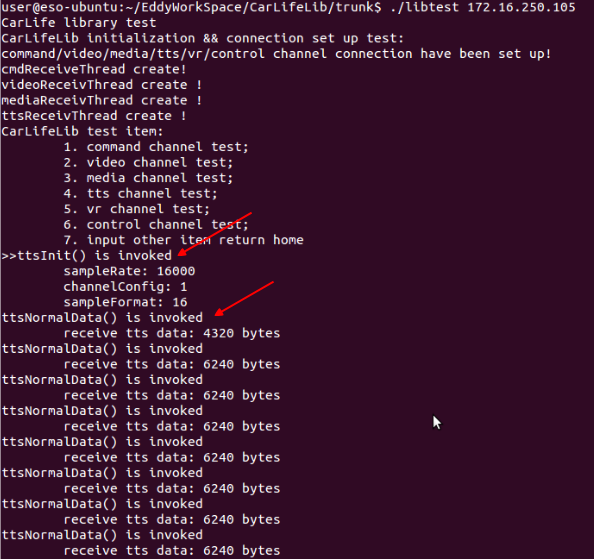


图13- 6

测试说明：手机端导航开始后，手机端CarLife将会通过tts通道将tts的初始化参数和数据发送给车机端的CarLife。

## vr通道测试

1. VR命令通道测试

测试步骤：

1. 使用Wi-Fi连接手机和测试台机（手机与台机处于同一网段）；
2. 启动手机端CarLife；
3. 运行测试台机上的libtest测试程序；
4. 开启并退出手机端CarLife的VR功能；
5. 观察libtest运行过程中的log；

预期结果如图13-7所示：

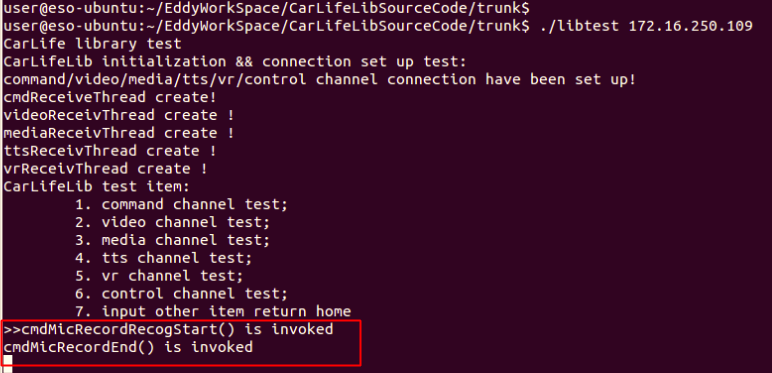


图13- 7

测试说明：VR功能开始时，手机通过命令通道通知车机开始录音，并触发从cmdMicRecordRecogStart()回调；VR功能关闭时，手机通过命令通道通知车机结束录音，并触发cmdMicRecordEnd()回调；

1. VR 数据通道测试

测试步骤：

1. 使用Wi-Fi连接手机和测试台机（手机与台机处于同一网段）；
2. 启动手机端CarLife；
3. 运行测试台机上的libtest测试程序；
4. 开启手机端CarLife的VR功能；
5. 观察libtest运行过程中的log；

预期结果如图13-8所示：

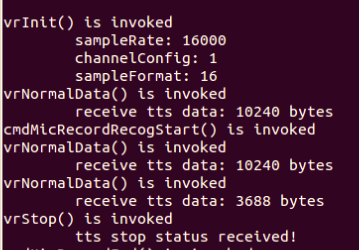


图13- 8

测试说明：手机端VR功能开启时，手机端会将提示音-“叮”通过VR通道发送到车机端，从而触发车机端vrInit(), vrNormalData(), vrStop()的等函数的回调。

1. VR录音测试

测试步骤：

1. 使用Wi-Fi连接手机和测试台机（手机与台机处于同一网段）；
2. 启动手机端CarLife；
3. 运行测试台机上的libtest测试程序，并选择item5；
4. 观察手机端CarLife的运行log（手机端程序使用测试版，开放log打印）；

预期结果：如图13-9所示，

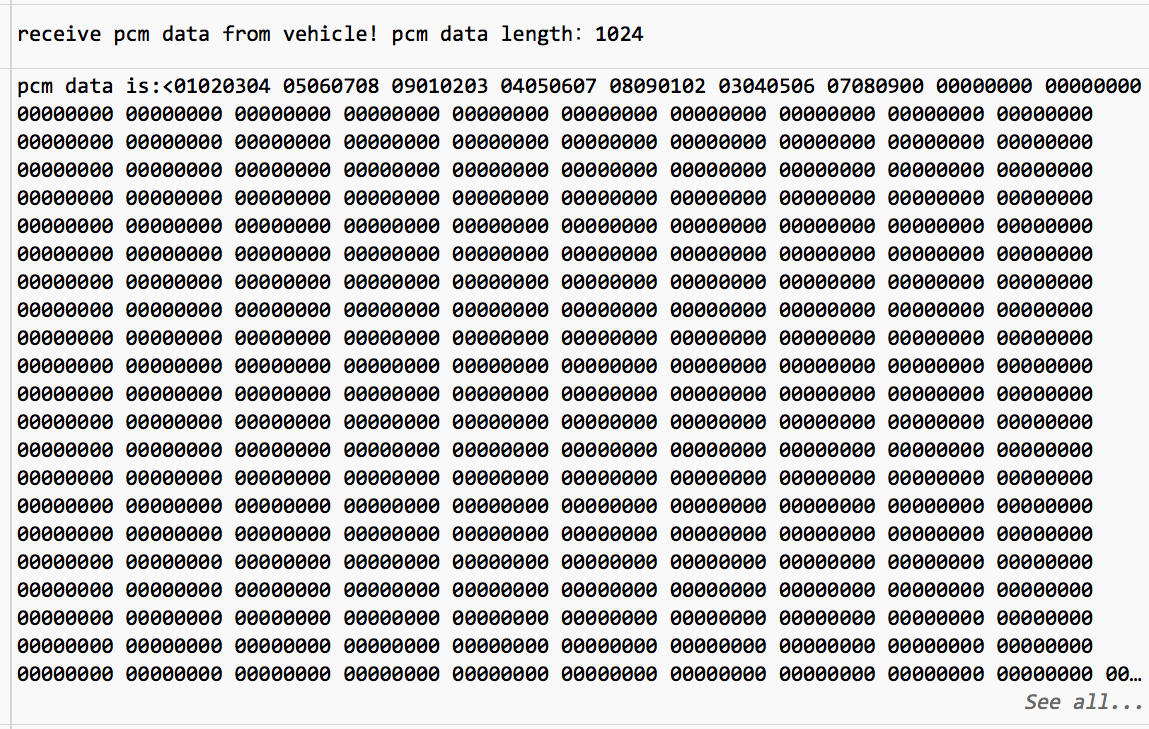


图13- 9

测试说明：通过int sendVRRecordData(u8\* data, u32 size, u32 timeStamp)接口，车机端能够将MIC录音产生的数据发送给手机端。

## control通道测试

（需要手机端代码做适当修改，并打开log）

测试步骤：

1. 使用Wi-Fi连接手机和测试台机（手机与台机处于同一网段）；
2. 启动手机端CarLife；
3. 运行测试台机上的libtest测试程序；
4. 在程序的输入选项中，选择输入6；
5. 观察手机端CarLife的运行log；

预期结果：如图13-10所示，

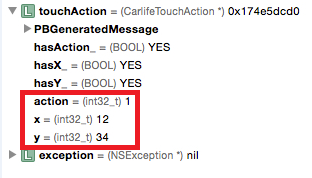


图13- 10

测试说明：车机端CarLife将控制信息通过control通道发给手机端CarLife，控制信息使用PB进行封装。