前言

1.1 简介

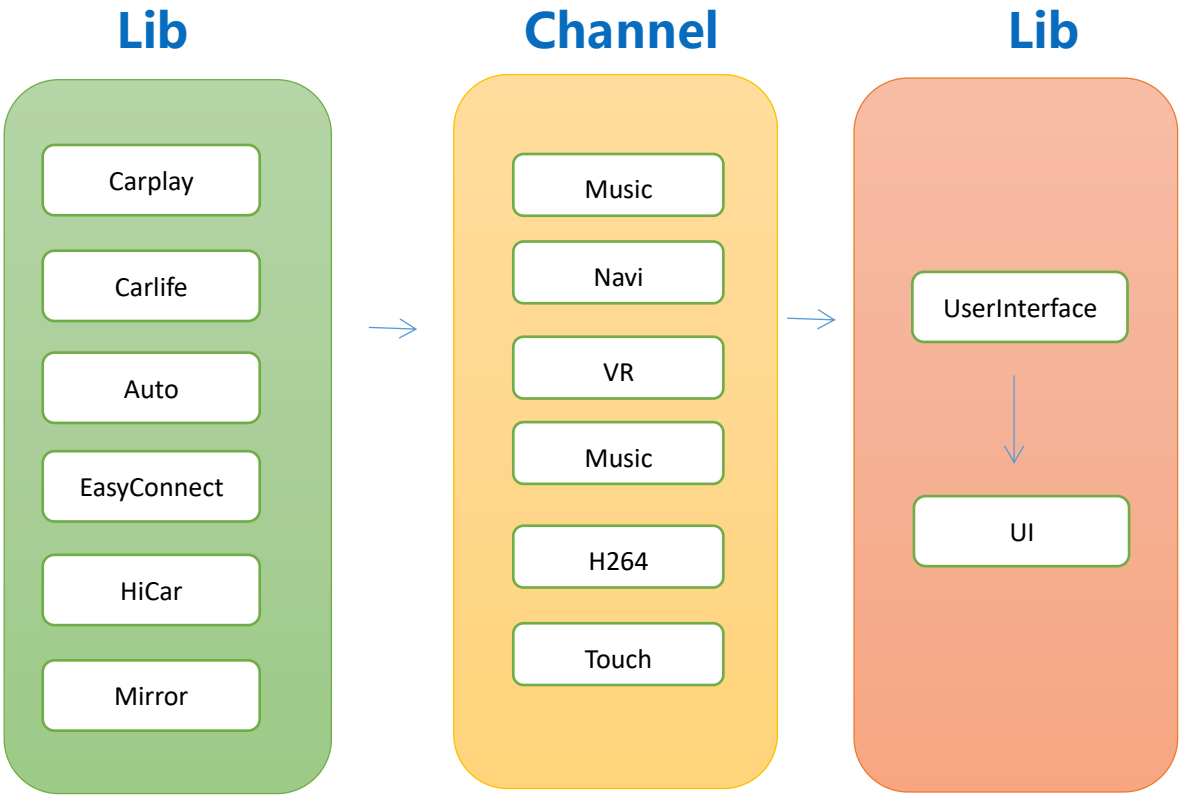
UserInterface 是一套由开阳电子股份有限公司开发的手机和 1668 及 1668e 车机互联集成的 SDK 包。它采用 C/C++ 开发语言，可以应用在 Linux、车机上。目前 UserInterface 对外仅提供 Linux 版本。

UserInterface 在手机和车机间建立互联连接后可以传递手机屏幕和音频等数据。车机厂商可以基于 UserInterface 进行二次集成开发，完成手机互联功能。

车机厂商在开发基于 UserInterface 互联程序时，必须借助手机上应用才能完成特定的互联功能。本文将此应用统称为"手机 APP"。

集成了苹果 Carplay，百度 Carlife，安卓 Auto，亿连 EasyConnect，华为 HiCar 的有线和无线连接开发包。

1.2 手机互联架构图



2. 产品介绍

2.1. 产品划分

2.1.1.手机互联有线版本

UserInterface 针对每一个手机互联包括 Carply,Carlife,HiCar,Auto,EasyConnet,Mirror 对应用户可以选择 Usb 连接. 支持多个平台 Linux,1668,1668e 系列.

2.1.2.手机互联无线版本

UserInterface 针对每一个手机互联包括 Carply，Carlife，HiCar，Auto，EasyConnet，Mirror 对应用户可以选择 WiFi，支持多个平台 Linux,1668,1668e 系列. 无线 Carplay 支持蓝牙地址连接车机，然后启动无线连接， Carlife 通过手机热点发射，车机端连接同一局域网实现无线互联，HiCar 车机发送一个验证码给手机输入验证码实现无线互联，Auto 暂未实现。

EasyConnet 暂时未实现，Mirror 通过手机互联发送热点，车机端连接同一局域网实现无线互联。

3. 硬件要求

3.1. CPU

CPU 主频率 800MHz 及以上。

3.2. 内存

512MB 及以上，UserInterface 运行内存不超过 20MB

3.3. 存储空间

512MB 及以上，UserInterface 占用存储空间 20MB，并提供运行时读写区域 5MB（运行时只读写目录/tmp）。

3.4. USB 支持

支持 USB 2.0 及以上 HOST 模式。如果项目需要支持 EAP 互联，多媒体主机须支持 USB 模式动态切换 HOST 与 DEVICE。其他互联方式，多媒体主机始终处于 HOST 模式。

USB 数据最低传输速率 16,000kbit/s，理想传输速率 24,000kbit/s 及以上；

为保证手机与多媒体主机正常互联，多媒体主机需要稳定电流 500mA 及以上。

若需保证互联过程中，手机始终处于充电状态，多媒体主机可提供 Android 手机不低于 1.5A 充电电流，可提供 iPhone 手机不低于 1A 充电电流（可选）。

3.5. 蓝牙支持

蓝牙 2.1 及以上，支持 A2DP（Advanced Audio Distribution Profile）蓝牙音频传输模型；

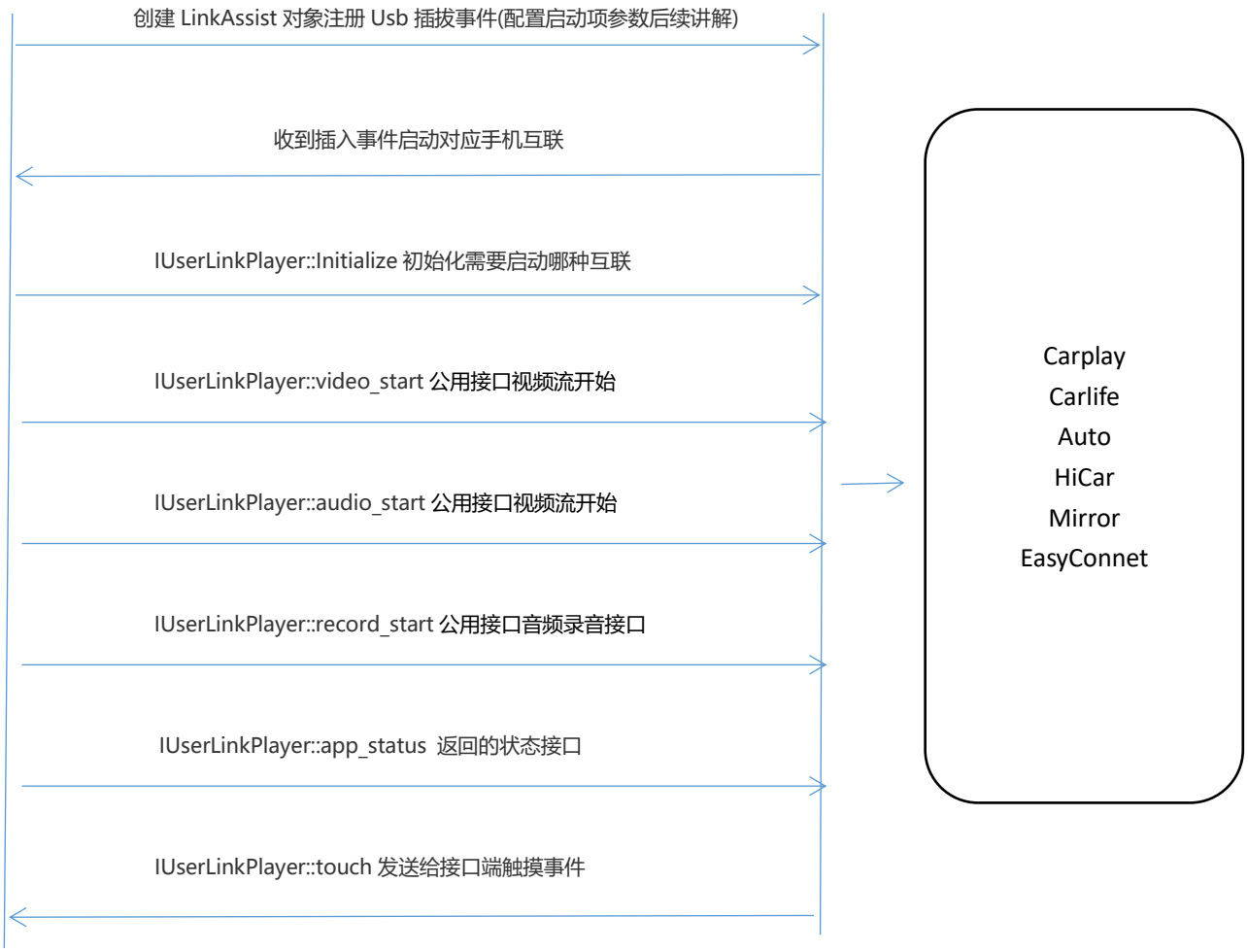
支持 HFP 协议，提供蓝牙拨打电话功能；

手机音频播放通过蓝牙传输至多媒体主机；

多媒体主机通过蓝牙获取手机播放音频的 meta 信息。

4. UserInterface 使用流程

4.1. UserInterface 使用流程图



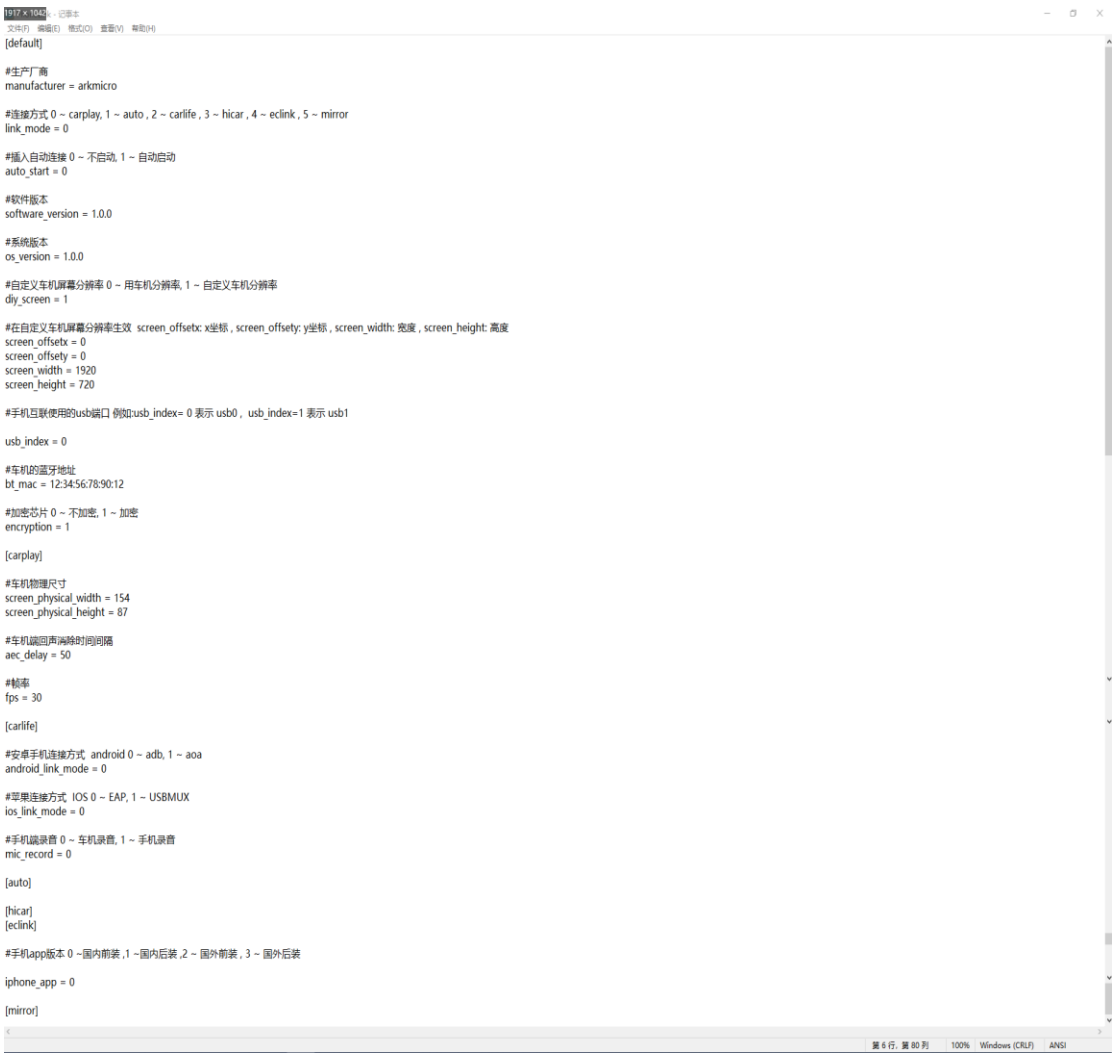
4.2. 基础功能使用

4.2.1.配置文件

配置文件 PhoneLink.ini 存放于/etc/目录下，没有配置文件或者不对的话,选用默认的参数也生效，ini 文件是用的 asic 码保存，其他 Unicode，utf8 等无效.

[default] 下面就是公用的配置比如软件版本，系统版本，车机分辨率设置，usb 端口，加密芯片等。每个手机互联下面有对应的手机互联设置，单独用的比如[carplay]的 fps，[carlife]的安卓连接方式,录音设置等。用户如需要添加一些配置可以自己按照格式添加，LinkAssist 类程序里面去管理配置项.

如一下图:



```
[default]

#生产厂商
manufacturer = arkmicro

#连接方式 0 ~ carplay, 1 ~ auto , 2 ~ carlife , 3 ~ hicar , 4 ~ eclink , 5 ~ mirror
link_mode = 0

#插入自动连接 0 ~ 不启动, 1 ~ 自动启动
auto_start = 0

#软件版本
software_version = 1.0.0

#系统版本
os_version = 1.0.0

#自定义车机屏幕分辨率 0 ~ 用车机分辨率, 1 ~ 自定义车机分辨率
diy_screen = 1

#在自定义车机屏幕分辨率生效  screen_offsetx: x坐标 , screen_offsety: y坐标 , screen_width: 宽度 , screen_height: 高度
screen_offsetx = 0
screen_offsety = 0
screen_width = 1920
screen_height = 720

#手机互联使用的usb端口 例如:usb_index= 0 表示 usb0 ,  usb_index=1 表示 usb1

usb_index = 0

#车机的蓝牙地址
bt_mac = 12:34:56:78:90:12

#加密芯片 0 ~ 不加密, 1 ~ 加密
encryption = 1

[carplay]

#车机物理尺寸
screen_physical_width = 154
screen_physical_height = 87

#车机端回声消除时间间隔
aec_delay = 50

#帧率
fps = 30

[carlife]

#安卓手机连接方式  android 0 ~ adb, 1 ~ aoa
android_link_mode = 0

#苹果连接方式  IOS 0 ~ EAP, 1 ~ USBMUX
ios_link_mode = 0

#手机端录音 0 ~ 车机录音, 1 ~ 手机录音
mic_record = 0

[auto]

[hicar]
[eclink]

#手机app版本 0 ~国内前装 ,1 ~国内后装 ,2 ~ 国外前装 , 3 ~ 国外后装

iphone_app = 0

[mirror]
```

4.2.2.函数使用及参数介绍

接口

IUserLinkPlayer* LinkAssist::Initialize(LinkType linkType)

返回值: 获取 IUserLinkPlayer 的指针

参数: linkType 启动互联的类型

LinkType	说明
Carplay	苹果 carplay
AndroidAuto	安卓 Auto
Carlife	百度 Carlife
HiCar	华为 HiCar
ECLink	亿连互联 EasyConnect
Mirror	安卓投屏
Airplay	苹果 airplay

接口

void LinkAssist::RegisterUsbCallback(FUNCUSBCALLBACK funcUSBCallback);

返回值: void

参数: FUNCUSBCALLBACK　插入拔出的函数指针

typedef std::function<void (ConnectedStatus, PhoneType)> FUNCCONNECTCALLBACK;

ConnectedStatus	**说明**
CONNECT_STATUS_DEVICE_ATTACHED	插入
CONNECT_STATUS_DEVICE_DEATTACHED	拔出
CONNECT_STATUS_CONNECTING	连接中
CONNECT_STATUS_CONNECT_FAILED	连接失败
CONNECT_STATUS_CONNECT_SUCCEED	连接成功
CONNECT_STATUS_DISCONNECTED	连接断开
CONNECT_STATUS_APP_EXIT	手机应用退出(如亿连,百度 Carlife)
CONNECT_STATUS_INTERRUPTED_BY_APP	手机连接中退出(如亿连,百度 Carlife)

PhoneType	**说明**
UnKnown	未知
Phone_Android	安卓
Phone_IOS	苹果

接口

LinkConfig LinkAssist::GetConfigInfo()

返回值: LinkConfig

参数: void

LinkConfig	**说明**
offset_x	屏幕坐标 x
offset_y	屏幕坐标 y
screen_width	屏幕宽度
screen_height	屏幕高度
diy_screen	是否需要自定义屏幕大小(0,1)
autostart	是否插入自动启动手机互联(0,1)
encryption	是否使用加密芯片保证应用正常启动(0,1)
LinkType	互联类型
version	软件版本
manufacturer	厂商
btmac	蓝牙地址

接口

IUserLinkPlayer::Initialize(LinkMode linkMode, PhoneType phoneType)

返回值: bool 类型 是否成功

参数: LinkMode 连接类型无线还是有线

PhoneType 手机类型

LinkMode	说明
Wired	有线
Wireless	无线
Auto	自动选择

PhoneType	说明
UnKnown	未知
Phone_Android	安卓手机
Phone_IOS	苹果手机

接口

bool IUserLinkPlayer::Release();

函数说明:释放资源

返回值: bool 类型 是否成功

参数: void

接口

bool IUserLinkPlayer::Start();

函数说明:启动互联

返回值: bool 类型 是否成功

参数: void

接口

bool IUserLinkPlayer::Stop();

函数说明:停止互联

返回值: bool 类型 是否成功

参数: void

接口

bool IUserLinkPlayer::StartMirror();

函数说明:启动投屏

返回值: bool 类型 是否成功

参数: void

接口

bool IUserLinkPlayer::StopMirror();

函数说明:停止投屏

返回值: bool 类型 是否成功

参数: void

接口

bool IUserLinkPlayer::SetBackground();

函数说明:车机端互联切换到后台

返回值: bool 类型 是否成功

参数: void

接口

bool IUserLinkPlayer::SetForeground();

函数说明:车机端互联切换到前台

返回值: bool 类型 是否成功

参数: void

接口

bool IUserLinkPlayer::SendScreenSize(int width, int height);

函数说明:发送车机端长度和宽度给手机端

返回值: bool 类型 是否成功

参数: int 类型 宽度 和 长度

接口

bool IUserLinkPlayer::SendTouch(int x, int y, TouchCode press);

函数说明:发送给车机端触摸坐标

返回值: bool 类型 是否成功

参数: int 类型 x 坐标

int 类型 y 坐标

TouchCode	**说明**
Touch_Up	抬起
Touch_Press	按下
Touch_Move	移动

接口

bool IUserLinkPlayer::SendKey(KeyCode keyCode);

函数说明:发送给车机端按键值

返回值: bool 类型 是否成功

KeyCode	**说明**
KEY_HOME	主界面
KEY_PHONE	拨号界面
KEY_TALKIE	对话（仅亿连用）
KEY_NAVIGATION	导航
KEY_VOICE_ASSISTANT	语音
KEY_MUSIC_PLAY	音乐播放

KEY_MUSIC_NEXT	下一曲
KEY_MUSIC_PREVIOUS	上一曲
KEY_MUSIC_PAUSE	音乐暂停
KEY_MUSIC_STOP	音乐停止
KEY_MUSIC_PLAY_PAUSE	音乐播放/暂停
KEY_VOLUME_UP	音量加
KEY_VOLUME_DOWN	音量减
KEY_TOPLEFT	焦点移动到左上角（仅百度 Carlife，亿连用）
KEY_TOPRIGHT	焦点移动到右上角（仅百度 Carlife，亿连用）
KEY_BOTTOMLEFT	焦点移动到左下角（仅百度 Carlife，亿连用）
KEY_BOTTOMRIGHT	焦点移动到右下角（仅百度 Carlife，亿连用）
KEY_LIGHT_MODE	白天模式
KEY_NIGHT_MODE	黑夜模式
KEY_APP_FRONT	手机端切换到前台（仅百度 Carlife，亿连用）
KEY_APP_BACK	手机端切换到前台（仅百度 Carlife，亿连用）
KEY_CAR_FRONT	切换车机端互联到前台
KEY_CAR_BACK	切换车机端互联到后台
KEY_ENFORCE_LANDSCAPE	强制手机端 app 横屏（仅亿连,安卓 Mirror 用）
KEY_CANCEL_LANDSCAPE	强制取消手机端 app 横屏（仅亿连,安卓 Mirror 用）
KEY_ENFORCE_OR_CANCEL_LANDSCAPE	强制/取消手机端 app 横屏（仅亿连,安卓 Mirror 用）
KEY_SYSTEM_HOME	回到主界面（仅安卓 Mirror 用）
KEY_SYSTEM_BACK	回到上一个状态（仅安卓 Mirror 用）
KEY_MAX	备用

接口

void IUserLinkPlayer::SendWheel(WheelCode wheelCode, bool bFoucs)

函数说明: 发送旋钮给手机

返回值: void

参数 bFoucs: 是否按下

参数 WheelCode 旋钮值

WheelCode	**说明**
Wheel_Keep	保持
Wheel_Next	向前滚动
Wheel_Previous	向后滚动

接口

bool IUserLinkPlayer::RequestStatus(RequestAppStatus requestAppStatus);

函数说明:发送请求给手机端

返回值: bool 类型 是否成功

参数

RequestAppStatus	说明
QUERYTIME	查询手机时间
QUERYGPS	查询 GPS 定位信息
QUERYBTSTATUS	查询蓝牙状态

接口

void IUserLinkPlayer::onSdkConnectStatus(ConnectedStatus status, PhoneType type);

函数说明:发送手机连接状态

返回值: void 类型

ConnectedStatus	说明
CONNECT_STATUS_DEVICE_ATTACHED	插入
CONNECT_STATUS_DEVICE_DEATTACHED	拔出
CONNECT_STATUS_CONNECTING	连接中
CONNECT_STATUS_CONNECT_FAILED	连接失败
CONNECT_STATUS_CONNECT_SUCCEED	连接成功
CONNECT_STATUS_DISCONNECTED	连接断开
CONNECT_STATUS_APP_EXIT	手机应用退出(如亿连,百度 Carlife)
CONNECT_STATUS_INTERRUPTED_BY_APP	手机连接中退出(如亿连,百度 Carlife)

PhoneType	说明
UnKnown	未知
Phone_Android	安卓
Phone_IOS	苹果

接口

bool IUserLinkPlayer::OpenPage(AppPage appPage);

函数说明:发送请求给手机端

返回值: bool 类型 是否成功

参数

AppPage	说明
APP_PAGE_NAVIGATION	导航界面
APP_PAGE_MUSIC	音乐界面
APP_PAGE_VR	语音界面
APP_PAGE_NAVI_HOME	导航到家的位置
APP_PAGE_NAVI_WORK	导航到工作的位置
APP_PAGE_MAIN	主界面
APP_PAGE_NAVI_GAS_STATION	导航到加油站位置
APP_PAGE_CAR_PARK	停车场的位置
APP_PAGE_4S_SHOP	4s 点位置

接口

void LinkAssist::RegisterAppStatusCallback(FUNCAPPSTATUSCALLBACK funcAppStatusCallback);

返回值: void

参数: FUNCAPPSTATUSCALLBACK　app 状态反馈

typedef std::function<void (AppStatusMessage, void*)> FUNCAPPSTATUSCALLBACK;

AppStatusMessage	**说明**
APP_NOT_RUNNING	手机 app 未运行（仅亿连和百度 Carlife）
APP_FOREGROUND	手机 app 切换前台
APP_BACKGROUND	手机 app 切换到后台
APP_SCREENLOCKED	手机锁屏
APP_UNSCREENLOCKED	手机解锁
APP_NAVI_STARTED	手机导航开始
APP_NAVI_STOPPED	手机导航结束
APP_VR_STARTED	手机语音开始
APP_VR_STOPPED	手机语音结束
LAUNCH_PHONE_APP	启动手机端 app
APP_RESERVED	备注（RequestStatus app 的状态指针数据反馈）

接口

void SendCarBluetooth(const string& name, const string& address, const string& pin);

返回值 void

参数　name :蓝牙名

address:蓝牙地址

pin：蓝牙 PIN 码

接口

void SendPhoneBluetooth(const string& address);

返回值 void

参数 address 手机端蓝牙地址

接口

bool SendIphoneMacAddress(string address);

返回值 bool 类型 是否成功

参数 address 手机端 ip 地址

4.2.3.音视频函数使用及参数介绍

接口

void video_play(const void *data, int32_t len);

返回值 void

参数 data: 视频数据

len: 视频长度

接口

void video_start(int offset_x, int offset_y, int width, int height);

返回值 void

参数　offset_x,: 视频坐标 x

offset_y: 视频坐标 y

width: 视频宽度

height：视频高度

接口

void video_stop ()

返回值 void

参数 void

接口

void audio_play(AudioType audioType, const void* data, uint32_t len);

返回值 void

参数 data： 数据 len：长度

AudioType	**说明**
AUDIO_TYPE_MUSIC	音乐
AUDIO_TYPE_CALL	电话（不包括蓝牙电话）
AUDIO_TYPE_VR	语音
AUDIO_TYPE_TTS	导航
AUDIO_TYPE_Rec	暂未用
AUDIO_TYPE_Alert	提示音（短信，铃声等）

接口

bool audio_start(AudioType audioType, int rate, int bit, int channel);

返回值 bool 是否成功

参数 audioType：音频流类型

rate：采样率

bit ：位

channel ：通道

接口

bool audio_stop(AudioType audioType);

返回值 bool 是否成功

参数 audioType：音频流类型

接口

void record_start(AudioInfo & audioInfo);

返回值 void

参数 **AudioInfo** ：音频参数

AudioInfo	说明
sampleRate	采样率
channel	通道数
format	位数

接口

void record_stop();

返回值 void

参数 void

接口

void record_pause(bool pause);

返回值 void

参数 pause 暂停/开始录音

4.2.4.蓝牙及电话函数使用及参数介绍

接口

void bt_call_action(CallType callType, const char *name, const char* number);

返回值 void

参数 name：电话名字

number：电话号码

CallType	说明
CALL_TYPE_DAIL	拨打/接听
CALL_TYPE_HANG_UP	挂断/拒绝
CALL_TYPE_MAX	备用

5. Dbus 方式通信

5.1. 进程间通信方式目的

目的是除了 lib 方式，需要用户根据具体接口来比较繁琐，难以管理，dbus 方式接口简单，兼容以前 1668 的方便调试，排查问题。改动不算太大。类型定义更加细分。

5.2. dbus 配置和接口

用户发出请求的接口，相对于 carlink 接收的

void requestLink(const int type, const int mode, const int status);

参数 type： LinkType （连接类型 carplay，carlife 等）

参数 mode: LinkMode （无线，有线）

参数 status：DbusReceive

返回值： void

DbusReceive	说明
DBUS_REQUEST_CONNECT	用户发起请求连接
DBUS_REQUEST_FOREGROUND	用户发起请求切到前台
DBUS_REQUEST_BACKGROUND	用户发起请求切到后台
DBUS_REQUEST_EXITED	用户发起请求退出互联

void requestTouch(int x, int y, TouchCode pressed);

函数说明:发送给车机端触摸坐标

返回值: void 类型

参数: int 类型 x 坐标

int 类型 y 坐标

TouchCode	说明
Touch_Up	抬起
Touch_Press	按下
Touch_Move	移动

void requestMultiTouch(int x1, int y1, TouchCode pressed1,int x2, int y2, TouchCode pressed2);

函数说明:发送给车机端多点触摸坐标

返回值: void 类型

参数: int 类型 x1 坐标 x2 坐标

int 类型 y1 坐标 y2 坐标

pressed1 触摸状态 pressed2 触摸坐标

void requestKey(KeyCode key);

函数说明:发送给车机端多点触摸坐标

返回值: void 类型

参数: KeyCode

KeyCode	说明
KEY_HOME	主界面
KEY_PHONE	拨号界面
KEY_TALKIE	对话（仅亿连用）
KEY_NAVIGATION	导航
KEY_VOICE_ASSISTANT	语音
KEY_MUSIC_PLAY	音乐播放
KEY_MUSIC_NEXT	下一曲
KEY_MUSIC_PREVIOUS	上一曲
KEY_MUSIC_PAUSE	音乐暂停
KEY_MUSIC_STOP	音乐停止
KEY_MUSIC_PLAY_PAUSE	音乐播放/暂停
KEY_VOLUME_UP	音量加
KEY_VOLUME_DOWN	音量减
KEY_TOPLEFT	焦点移动到左上角（仅百度 Carlife，亿连用）
KEY_TOPRIGHT	焦点移动到右上角（仅百度 Carlife，亿连用）
KEY_BOTTOMLEFT	焦点移动到左下角（仅百度 Carlife，亿连用）
KEY_BOTTOMRIGHT	焦点移动到右下角（仅百度 Carlife，亿连用）
KEY_LIGHT_MODE	白天模式
KEY_NIGHT_MODE	黑夜模式
KEY_APP_FRONT	手机端切换到前台（仅百度 Carlife，亿连用）

KEY_APP_BACK	手机端切换到前台（仅百度 Carlife，亿连用）
KEY_CAR_FRONT	切换车机端互联到前台
KEY_CAR_BACK	切换车机端互联到后台
KEY_ENFORCE_LANDSCAPE	强制手机端 app 横屏（仅亿连,安卓 Mirror 用）
KEY_CANCEL_LANDSCAPE	强制取消手机端 app 横屏（仅亿连,安卓 Mirror 用）
KEY_ENFORCE_OR_CANCEL_LANDSCAPE	强制/取消手机端 app 横屏（仅亿连,安卓 Mirror 用）
KEY_SYSTEM_HOME	回到主界面（仅安卓 Mirror 用）
KEY_SYSTEM_BACK	回到上一个状态（仅安卓 Mirror 用）
KEY_MAX	备用

void requestWheel(WheelCode wheelCode, bool bFoucs);

函数说明: 请求发送旋钮给手机

返回值: void

参数 bFoucs: 是否按下

参数 wheelCode 旋钮值

WheelCode	**说明**
Wheel_Keep	保持
Wheel_Next	向前滚动
Wheel_Previous	向后滚动

void requestPage(AppPage appPage);

函数说明: 请求发送打开页面

返回值: void

参数 AppPage: 打开指定页面

AppPage	**说明**
APP_PAGE_NAVIGATION	导航界面
APP_PAGE_MUSIC	音乐界面
APP_PAGE_VR	语音界面
APP_PAGE_NAVI_HOME	导航到家的位置
APP_PAGE_NAVI_WORK	导航到工作的位置
APP_PAGE_MAIN	主界面
APP_PAGE_NAVI_GAS_STATION	导航到加油站位置
APP_PAGE_CAR_PARK	停车场的位置
APP_PAGE_4S_SHOP	4s 点位置

void requestPhoneIPAddress(string str);

函数说明: 请求发送手机端 ip 地址

返回值: void

参数 string 手机端 ip

void requestPhoneBTAddress(string str);

函数说明: 请求发送手机端蓝牙地址

返回值: void

参数 string 手机段蓝牙地址

CarLink 反馈给车机端应用

void onLinkStatus(const int type, const int mode, const int status);

函数说明: 发送给车机端应用的状态

返回值: void

参数 type： 互联类型（carplay，carlife 等）

mode： 有线或无线

status：DbusSend 类型

DbusSend	说明
DBUS_DEVICE_ATTACHED	发给用户手机端插入消息
DBUS_DEVICE_DEATTACHED	发给用户手机端拔出消息
DBUS_CONNECTTING	发给用户正在连接
DBUS_CONNECTED	发给用户连接成功
DBUS_FOREGROUND	发给用户手机互联已经切换前台
DBUS_BACKGROUND	发给用户手机互联已经切换后台
DBUS_FAILED	发给用户手机互联连接失败
DBUS_DISCONNECTED	发给用户手机互联连接断开
DBUS_APP_EXIT	发给用户手机互联进程退出
DBUS_INTERRUPTED_BY_APP	发给用户手机互联连接中被杀掉（亿连和 carlife 用）

void SendCarLinkDateTime(const int type, const long long time);

函数说明: 发送给车机端应用的当地时间

返回值: void

参数 type： 互联类型（carplay，carlife 等）

time： long long 类型本地时间

void SendTelephone(const int type, const string name, const string number);

函数说明: 发送给车机端应用的号码信息

返回值: void

参数 type： 互联类型（carplay，carlife 等）

name： string 类型 名字

number： string 类型 号码

5.3. dbus 命令通信

目的可以通过命令行输入快速测试互联

发起连接

例如 carplay 连接

参数 1：int32:0 （carplay 的类型）

参数 2：int32:0 (有线类型)

参数 3：int32:0　(发起连接)

dbus-send --session --type=method_call --print-reply --dest=com.arkmicro.carlink /com/arkmicro/carlink Local.DbusServer.CarLink.requestLink int32:0 int32:0 int32:0

切换前台

参数 1：int32:0　（carplay 的类型）

参数 2：int32:0　(有线类型)

参数 3：int32:0　(发起切换前台)

dbus-send --session --type=method_call --print-reply --dest=com.arkmicro.carlink /com/arkmicro/carlink Local.DbusServer.CarLink.requestLink int32:0 int32:0 int32:1

切换后台

参数 1：int32:0　（carplay 的类型）

参数 2：int32:0　(有线类型)

参数 3：int32:0　(发起切换后台)

dbus-send --session --type=method_call --print-reply --dest=com.arkmicro.carlink /com/arkmicro/carlink Local.DbusServer.CarLink.requestLink int32:0 int32:0 int32:2

发起退出

参数 1：int32:0　（carplay 的类型）

参数 2：int32:0　(有线类型)

参数 3：int32:0　(发起退出)

dbus-send --session --type=method_call --print-reply --dest=com.arkmicro.carlink /com/arkmicro/carlink Local.DbusServer.CarLink.requestLink int32:0 int32:0 int32:3