手机app与点烟器硬件通信协议

目录

[修订历史 2](#_Toc459304491)

[1. 文档说明 3](#_Toc459304492)

[2. Socket数据交互 3](#_Toc459304493)

[2.1. 通信安全 3](#_Toc459304494)

[2.2. json内容格式 3](#_Toc459304495)

[2.2.1. json示例 4](#_Toc459304496)

[3. app与硬件交互 5](#_Toc459304497)

[3.1. 接口汇总 5](#_Toc459304498)

[3.2. getDeviceConfig查询设备配置接口 6](#_Toc459304499)

[3.2.1. 参数配置项 6](#_Toc459304503)

[3.3. restartWifi重启路由器 8](#_Toc459304504)

[3.4. setWifi设置wifi 9](#_Toc459304505)

[3.5. resetFactory恢复出厂设置 9](#_Toc459304506)

[3.6. setFm设置fm发射频率 10](#_Toc459304507)

[3.7. setPanelLight设置面板灯状态 10](#_Toc459304508)

[3.8. setGesture设置手势 11](#_Toc459304509)

[3.9. setPlayMode设置收听模式 12](#_Toc459304510)

[3.10. playAudioTone播放提示音 13](#_Toc459304511)

[3.11. buttonClick点烟器按键 13](#_Toc459304512)

[3.12. bindDevice绑定设备 14](#_Toc459304513)

[3.13. unbindDevice解绑设备 14](#_Toc459304514)

[4. 硬件与内置播放器交互 15](#_Toc459304515)

[4.1. 接口汇总 15](#_Toc459304516)

[4.2. setPlayMode设置收听模式 15](#_Toc459304517)

[4.3. playAudioTone播放提示音 15](#_Toc459304518)

[4.4. wavePlayNext挥手播放下一曲上一曲 16](#_Toc459304519)

[4.5. waveAnswerPhone挥手接电话 16](#_Toc459304520)

[4.6. buttonClick点烟器按键 16](#_Toc459304521)

[4.7. bluetoothConnect蓝牙连接状态广播 16](#_Toc459304522)

[4.8. 设置白名单 17](#_Toc459304523)

[4.8.1. 白名单格式 17](#_Toc459304524)

[4.8.2. 白名单更新策略 17](#_Toc459304525)

# 修订历史

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 修订人 | 修订内容 | 时间 |
| 1.0 | 尹徐明 | 编写整体文档 | 2016-7-28 |
| 1.1 | 尹徐明 | 2.2增加json数据交互项： requestCode、requestFrom、requestTo；  增加设置手势接口setGesture；  增加按键操作广播协议 | 2016-7-29 |
| 1.2 | 尹徐明 | 3.2.1 增加蓝牙是否连接状态参数  3.10、4.3增加内置播放器播放提示音接口 | 2016-08-02 |
| 1.3 | 尹徐明 | 4.8 白名单更新策略调整 | 2016-08-03 |
| 1.4 | 尹徐明 | 4.8.1 增加白名单更新地址及具体格式 | 2016-08-09 |
| 1.5 | 尹徐明 | 3.2.1增加了部分参数项，并增加了参数维护方，使用ContentProvider进程共享数据  3.12 增加了硬件提交配置信息接口 | 2016-08-16 |

# 文档说明

手机app与点烟器硬件采用wifi socket通信，实现app控制硬件和内置播放器的功能。手机app与硬件端通过socket长连接实现通信，手机app负责client端发起通信，点烟器硬件内置固定IP地址，并负责启动ServerSocket 监听固定端口号，事先告知ecarx方开发人员固定ip地址和端口号。

当sever端接收到输入流，解析接口命令，根据请求中的requestTo字段判断是否该自己处理，如果requestTo不为0，则将消息通过3中的广播协议转发给内置播放器，其余的硬件控制命令请求，由点烟器server端接收到自己处理。

# Socket数据交互

手机app与点烟器硬件通过socket长连接进行交互，并且每次请求都是异步的，即当app发起一个请求给点烟器后，点烟器并不会立即响应结果，等到点烟器处理完后，才会向socket连接中发起写响应流的操作，这就需要app端缓存每次的请求码requestCode，并维护一个线程，去监听socket端的响应流，当app端接收到响应流时，取出响应的requestCode并去缓存中找到当初的请求requestCode，对比一致后，确定为对应请求的响应结果，返回给界面展示。

两端统一交互的数据格式为json，并且用<ECARX></ECARX>标签将json内容包裹起来，以方便socket读取时确定结束位置。

<ECARX>

json内容

</ECARX>

## 通信安全

目前暂定RSA加密。点烟器硬件上传信息时，ecarx server会返回一段密钥。

## json内容格式

其中json内容中有几个固定key

|  |  |  |
| --- | --- | --- |
| 字段 | 名称 | 说明 |
| code | 操作码 | 操作码0：处理成功；1：处理失败。只有为0的时候才去解析data |
| msg | 消息说明 |  |
| action | 接口名称 | 具体的请求业务接口，数据加密。 |
| requestCode | 请求码 | 由发起请求方生成，保证自身的请求码的唯一性，当请求处理方处理完请求时，将该请求码原样返回给请求方。 |
| requestFrom | 请求发起方 | 0：点烟器硬件；  1：点烟器内置播放器；  2：Ecarx Server；  3：IOS APP；  4：Android APP； |
| requestTo | 请求最终处理方 | 同requestTo |
| data | 数据 | 通信返回内容，数据做加密处理。固定为jsonObject类型，不同的业务接口返回数据里面可以有jsonObject、jsonArray项 |

### json示例

#### 加密后

{

"code": "0",

"msg": "操作成功",

"action": "ni8If5lC13Y7ehvPCdseb5V",

"requestCode": "10001",

"requestFrom": "4",

"requestTo": "0",

"data": "U2FsdGVkX1957OXVqJPni8If5lC1b0sb13Y7ehvPCdseb5VxDqtc2Yign1UA="

}

#### 解密后

取出data的值，进行解密，解密后的数据相当于下列内容：

{

"code": "0",

"msg": "操作成功",

"action": "getDeviceConfig",

"requestCode": "10001",

"requestFrom": "4",

"requestTo": "0",

"data": {

"wifiName": "ecrx点烟器",

"imei": "0000000000",

"bluetoothStatus": "1",

"wifiConnectList": [

{

"name": "iPhone 6s",

"ip": "192.168.1.2"

},

{

"name": "Honor 6p",

"ip": "192.168.1.3"

},

{

"name": "oppo R9",

"ip": "192.168.1.4"

}

]

}

}

# app与硬件交互

## 接口汇总

|  |  |  |
| --- | --- | --- |
| action接口 | 功能 | 说明 |
| getDeviceConfig | 查询设备配置接口 |  |
| restartWifi | 重启路由器 |  |
| setWifi | 设置wifi | wifi名称、wifi密码，可同时设置1项或多项 |
| resetFactory | 恢复出厂设置 |  |
| setFm | 设置fm发射频率 |  |
| setPanelLight | 设置面板灯状态 |  |
| setGesture | 设置手势 |  |
| setPlayMode | 设置收听模式 | 智能、随机、收藏，广播转发命令给播放器 |
| playAudioTone | 播放提示音 |  |
| buttonClick | 点烟器按键 | 蓝牙已连接情景 |
| bindDevice | 绑定设备 | app与ecarx Sever交互 |
| unbindDevice | 解绑设备 | app与ecarx Sever交互 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## getDeviceConfig查询设备配置接口

获取硬件配置信息以及内置播放器信息。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 参数项 | 参数值 | | 说明 |
| 输入参数 | action | getDeviceConfig | |  |
| requestCode |  | | 请求方生成，每次请求都不同 |
| requestFrom | 3或者4 | | 0：点烟器硬件；  1：点烟器内置播放器；  2：Ecarx Server；  3：IOS APP；  4：Android APP； |
| requestTo | 0 | |  |
| 输出参数 | code | 0、1 | | 响应成功、失败 |
| msg |  | |  |
| action | getDeviceConfig | |  |
| requestCode |  | | 同请求值 |
| requestFrom | 0 | |  |
| requestTo | 3或者4 | |  |
| data | key | value |  |
| wifiName | wifi名称 |
| imei |  |  |
| playMode |  |  |
| ...其他参数详见下表 | |  |



### 参数配置项

参数配置包括两个部分，硬件部分配置参数（wifi，蓝牙等状态）和点烟器播放器配置参数。

为了便于向app一次返回参数配置信息建议：

硬件和播放器采用ContentProvider进行进程间数据共享，由播放器读写自己的配置项，由硬件方读写自己的配置项，每次状态发生改变后都更新ContentProvider，并且当恢复出厂设置后各自重建自己的配置项。

当调用getDeviceConfig接口时，server端一次性读取所有参数项配置返回给手机app。

表格 1

|  |  |  |  |
| --- | --- | --- | --- |
| 参数项 | 取值范围（类型String） | 参数维护方 | 说明 |
| wifiName |  | 硬件方 | wifi名称 |
| wifiPwd |  | 硬件方 | wifi密码 |
| mac |  | 硬件方 | mac地址 |
| imei |  | 硬件方 | imei号 |
| phone |  | 硬件方 | sim卡对应的手机号 |
| panelLight | 0:关闭；1:开启 | 硬件方 | 面板灯状态 |
| gesture | 0:关闭；1:开启 | 硬件方 | 手势开启状态 |
| wifiStatus | 0:关闭；1:开启 | 硬件方 | wifi开启状态 |
| bindStatus | 0:未绑定；1:已绑定 | 硬件方 | 设备和账号绑定状态 |
| bluetoothStatus | 0:关闭；1:开启 | 硬件方 | 蓝牙开启状态 |
| bluetoothConnectStatus | 0:未连接；1:已连接 | 硬件方 | 蓝牙连接状态 |
| fm |  | 硬件方 | fm发射频率 |
|  |  |  |  |
| playMode | 0:智能；1:收藏；2:随机 | 内置播放器 | 播放模式 |
| wifiPlayerVersion |  | 内置播放器 | 点烟器内置播放器版本 |

## restartWifi重启路由器

|  |  |  |  |
| --- | --- | --- | --- |
|  | 参数项 | 参数值 | 说明 |
| 输入参数 | action | restartWifi |  |
| requestCode |  | 请求方生成，每次请求都不同 |
| requestFrom | 3或者4 | 0：点烟器硬件；  1：点烟器内置播放器；  2：Ecarx Server；  3：IOS APP；  4：Android APP； |
| requestTo | 0 |  |
| 输出参数 | code | 0、1 | 响应成功、失败 |
| msg |  |  |
| action | restartWifi |  |
| requestCode |  | 同请求值 |
| requestFrom | 0 |  |
| requestTo | 3或者4 |  |

## setWifi设置wifi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 参数项 | 参数值 | | 说明 |
| 输入参数 | action | setWifi | |  |
| requestCode |  | | 请求方生成，每次请求都不同 |
| requestFrom | 3或者4 | | 0：点烟器硬件；  1：点烟器内置播放器；  2：Ecarx Server；  3：IOS APP；  4：Android APP； |
| requestTo | 0 | |  |
| data | key | value | 可选 |
| wifiName | wifi名称值 |
| wifiPwd | wifi密码值 | 可选 |
| 输出参数 | code | 0、1 | | 响应成功、失败 |
| msg |  | |  |
| action | setWifi | |  |
| requestCode |  | | 同请求值 |
| requestFrom | 0 | |  |
| requestTo | 3或者4 | |  |

## resetFactory恢复出厂设置

|  |  |  |  |
| --- | --- | --- | --- |
|  | 参数项 | 参数值 | 说明 |
| 输入参数 | action | resetFactory |  |
| requestCode |  | 请求方生成，每次请求都不同 |
| requestFrom | 3或者4 | 0：点烟器硬件；  1：点烟器内置播放器；  2：Ecarx Server；  3：IOS APP；  4：Android APP； |
| requestTo | 0 |  |
| 输出参数 | code | 0、1 | 响应成功、失败 |
| msg |  |  |
| action | resetFactory |  |
| requestCode |  | 同请求值 |
| requestFrom | 0 |  |
| requestTo | 3或者4 |  |

## setFm设置fm发射频率

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 参数项 | 参数值 | | 说明 |
| 输入参数 | action | setFm | |  |
| requestCode |  | | 请求方生成，每次请求都不同 |
| requestFrom | 3或者4 | | 0：点烟器硬件；  1：点烟器内置播放器；  2：Ecarx Server；  3：IOS APP；  4：Android APP； |
| requestTo | 0 | |  |
| data | key | value | 发射频率 |
| fm | 频率值 |
| 输出参数 | code | 0、1 | | 响应成功、失败 |
| msg |  | |  |
| action | setFm | |  |
| requestCode |  | | 同请求值 |
| requestFrom | 0 | |  |
| requestTo | 3或者4 | |  |

## setPanelLight设置面板灯状态

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 参数项 | 参数值 | | 说明 |
| 输入参数 | action | setPanelLight | |  |
| requestCode |  | | 请求方生成，每次请求都不同 |
| requestFrom | 3或者4 | | 0：点烟器硬件；  1：点烟器内置播放器；  2：Ecarx Server；  3：IOS APP；  4：Android APP； |
| requestTo | 0 | |  |
| data | key | value | 0:关闭；1:开启 |
| panelLight | 0、1 |
| 输出参数 | code | 0、1 | | 响应成功、失败 |
| msg |  | |  |
| action | setPanelLight | |  |
| requestCode |  | | 同请求值 |
| requestFrom | 0 | |  |
| requestTo | 3或者4 | |  |

## setGesture设置手势

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 参数项 | 参数值 | | 说明 |
| 输入参数 | action | setGesture | |  |
| requestCode |  | | 请求方生成，每次请求都不同 |
| requestFrom | 3或者4 | | 0：点烟器硬件；  1：点烟器内置播放器；  2：Ecarx Server；  3：IOS APP；  4：Android APP； |
| requestTo | 0 | |  |
| data | key | value | 0:关闭；1:开启 |
| gestureStatus |  |
| 输出参数 | code | 0、1 | | 响应成功、失败 |
| msg |  | |  |
| action | setGesture | |  |
| requestCode |  | | 同请求值 |
| requestFrom | 0 | |  |
| requestTo | 3或者4 | |  |

## setPlayMode设置收听模式

硬件ServerSocket接收到app发出的该指令，将指令通过广播转发给内置播放器，并立即返回响应给app，具体广播通信协议见4.2。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 参数项 | 参数值 | | 说明 |
| 输入参数 | action | setPlayMode | |  |
| requestCode |  | | 请求方生成，每次请求都不同 |
| requestFrom | 3或者4 | | 0：点烟器硬件；  1：点烟器内置播放器；  2：Ecarx Server；  3：IOS APP；  4：Android APP； |
| requestTo | 1 | | 内置播放器为最终响应方 |
| data | key | value | 0:智能；1:收藏；2:随机 |
| playMode |  |
| 输出参数 | code | 0、1 | | 响应成功、失败 |
| msg |  | |  |
| action | setPlayMode | |  |
| requestCode |  | | 同请求值 |
| requestFrom | 0 | |  |
| requestTo | 3或者4 | |  |

## playAudioTone播放提示音

当设置好fm频率后，app端点击播放提示音，硬件接收到该指令，将命令以广播的方式转发给内置播放器，具体广播通信协议见4.3。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 参数项 | 参数值 | | 说明 |
| 输入参数 | action | playAudioTone | |  |
| requestCode |  | | 请求方生成，每次请求都不同 |
| requestFrom | 3或者4 | | 0：点烟器硬件；  1：点烟器内置播放器；  2：Ecarx Server；  3：IOS APP；  4：Android APP； |
| requestTo | 1 | | 内置播放器为最终响应方 |
|  |  |  |  |
|  |  |
| 输出参数 |  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |

## buttonClick点烟器按键

已连接蓝牙情景下按键，点烟器按键作为蓝牙音箱的标准按键操作，

未连接蓝牙的情况下，点烟器作为内置播放器的按键操作，详细通信协议见4.5。

已连接蓝牙情况下，音箱正在播放app音源，点击按键，app监听到蓝牙按键操作，暂停播放，同理，当app暂停播放时，点击按键，app开始播放。

**app端需要做的工作，监听蓝牙标准按键协议广播**，以android为例：需要监听广播

action == Intent.ACTION\_MEDIA\_BUTTON

|  |  |  |
| --- | --- | --- |
| 命令 | 广播参数项 | |
| buttonClick | action | android.intent.action.MEDIA\_BUTTON |

## bindDevice绑定设备

|  |  |
| --- | --- |
| 通信端 | 通信流程 |
| 点烟器硬件端 | 点烟器硬件初始化后，向ecarx server上传硬件信息，ecarx server返回绑定信息（通信密钥） |
| 手机app端 | app通过getDeviceConfig获取设备配置信息，用户提交绑定信息到ecarx server，ecarx server 返回通信密钥。 |

点烟器硬件提交设备信息

|  |  |  |  |
| --- | --- | --- | --- |
|  | 参数项 | 参数值 | 说明 |
| 请求url | <http://hardware.ecarx.com.cn/encrypt/equipment/add.do> | | |
| 输入参数 | eid |  | 设备型号 |
| eType | 3 | 产品类型 固定3 |
| eversion |  | 硬件版本 |
| appVersion |  | 应用版本 |
| playVersion |  | 内置播放器版本， |
| mac |  | mac地址 |
| phone |  | sim卡手机号 |
| 输出参数 | status | 0或1 |  |
| message |  |  |
| data |  | 通信密钥 |
|  |  |  |
|  |  |  |
|  |  |  |

## unbindDevice解绑设备

|  |  |
| --- | --- |
| 通信端 | 通信流程 |
| 点烟器硬件端 | 点烟器硬件初始化后，向ecarx server上传硬件信息，ecarx server返回绑定信息（通信密钥），此时得到的密钥应该为空（已解除绑定）。 |
| 手机app端 | 用户提交解除绑定信息到ecarx server，ecarx server 解除硬件与用户的绑定关系，app端清除原有的通信密钥。 |

# 硬件与内置播放器交互

点烟器硬件与内置播放器通过android 标准广播进行通信。

## 接口汇总

|  |  |  |
| --- | --- | --- |
| 接口 | 功能 | 说明 |
| setPlayMode | 设置收听模式 | 智能、随机、收藏，广播转发命令给播放器 |
| wavePlayNext | 挥手播放下一曲上一曲 | 如果开启手势功能，硬件接收到该命令，广播转发命令给播放器 |
| waveAnswerPhone | 挥手接听电话 | app电话和普通电话 |
| buttonClick | 点烟器按键 | 蓝牙未连接情景 |
| bluetoothConnect | 蓝牙连接状态广播 | 连接上蓝牙、断开蓝牙广播 |
|  |  |  |

## setPlayMode设置收听模式

硬件将该命令以广播的方式发送给内置播放器

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 命令 | 广播参数项 | | | |
| setPlayMode | action | cn.com.ecarx.xiaoka.music.service.audioplay.ACTION\_PLAY\_MODE | | |
| intent | key | 类型 | value |
| playMode | String | 0:智能；1:收藏；2:随机 |

## playAudioTone播放提示音

当设置好fm频率后，app端点击播放提示音，硬件接收到该指令，将命令以广播的方式转发给内置播放器，内置播放器接收到该指令后，播放预先设置好的提示音。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 命令 | 广播参数项 | | | |
| setPlayMode | action | cn.com.ecarx.xiaoka.music.service.audioplay.ACTION\_PLAY\_TONE | | |
|  |  |  |  |
|  |  |  |

## wavePlayNext挥手播放下一曲上一曲

硬件检测到挥手，**先判断是否开启手势功能**，如果开启，将检测到的手势转换位上一曲、下一曲指令，以广播的方式发送给内置播放器，如果没有开启手势功能不做处理。

手势转换规则：

从左到右挥手：0，下一曲

从右到左挥手：1，上一曲

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 命令 | 广播参数项 | | | |
| wavePlayNext | action | cn.com.ecarx.xiaoka.music.service.audioplay.ACTION\_PLAY\_NEXT | | |
| intent | key | 类型 | value |
| playNext | String | 0:下一曲；1：上一曲 |

## waveAnswerPhone挥手接电话

硬件检测到挥手，**先判断是否开启手势功能，以及手机蓝牙与硬件是否已连接**，如果已开启手势并且蓝牙已连接，发出标准的蓝牙耳机接听电话的广播，。

对于传统电话，系统会自动接起、挂断电话；

对于有手机app内的网络电话，手机app负责捕获蓝牙按键广播事件，接听、挂断电话。

## buttonClick点烟器按键

已连接蓝牙情景下，查看3.10。

未连接蓝牙情景下按键，点烟器按键作为内置播放器控制。

双击：内置播放器播放模式切换（收藏、智能、随机）

单击：内置播放器播放暂停

广播协议：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 命令 | 广播参数项 | | | |
| buttonClick | action | cn.com.ecarx.xiaoka.music.service.audioplay.ACTION\_PLAY\_BUTTON | | |
| intent | key | 类型 | value |
| clickNum | String | 1:单击；2:双击 |

## bluetoothConnect蓝牙连接状态广播

当点烟器蓝牙连接状态改变时，发出该广播。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 命令 | 广播参数项 | | | |
| bluetoothConnect | action | cn.com.ecarx.xiaoka.music.service.audioplay.ACTION\_BLE\_CONNECT | | |
| intent | key | 类型 | value |
| bluetoothStatus | String | 0:未连接；1:已连接 |

## 设置白名单

由硬件每次启动时，去ecarx服务器获取白名单列表，同时ecarx服务器会有更新白名单推送消息发给硬件，硬件接收到这样的消息时，去ecarx服务器获取更新后的白名单列表。

### 白名单格式

1. 下载白名单的地址，是固定的

<http://upgrade.ecarx.com.cn/appdownload/downloadResource.json?appType=10>

<ecarx version=“1.0.0”>

<domain-list>

<domain>ecarx.com.cn</domain>

<domain>ecarx.com.cn/interface/</domain>

<domain>auto.sohu.com/</domain>

</domain-list>

</ecarx>

### 白名单更新策略

硬件每次启动的时候，更新一次白名单，ecarx会提供一下几个接口：

a) 根据硬件编号 查询最新版本的接口（使用于软件更新，白名单比较小，可以直接使用b更新一次）

b) 根据硬件编号 下载最新版本的接口

c) 提交当前硬件版本的接口（服务器维护所有硬件的版本信息，更新成功后提交一次当前版本，白名单文件上有版本号）

消息推送方式提示更新白名单

{

“order”：“1”， //根据order确定执命令类型 1表示白名单更新指令 2 更新操作密码

“data”:”1.1.13”, //不同指令数据不同，data有可能是空值”data”:””

“time”:”123455665554”, //时间字段确定实效性

“type”:”1” //1表示推送给播放器，0 表示推送给硬件 10 同时推送给硬件+内置播放器

}