

悠达背景音乐协议

修改记录

日期	修改记录	修改人
20191220	重新整理版本	bcf
20200311	心跳与地址相关描述修正	bcf
20200828	补充自定义触发播放的说明	bcf
20230224	音乐场景相关说明	bcf
20230428	增加云音乐 VIP 相关接口	bcf
20230504	增加云音乐 VIP 支付接口	bcf
20230801	添加新增的设备类型	bcf
20230913	完善通道的配置参数说明	bcf
20230919	一些错别字和描述修正	bcf
20231101	音乐场景指令补全	bcf

目录

1. FAQ.....	5
1.1 协议控制方式?	5
1.2 如何搜索设备?	5
1.3 如何获取设备 IP?	5
1.4 为何发送命令后设备没有任何反应?	5
1.5 什么是通道编号?.....	6
1.6 如何计算校验?	6
1.7 如何控制设备?	6
1.8 party 功能是什么?	6
1.9 主机级联是什么?.....	7
1.10 手机推送 PARTY 是什么?	7
1.11 系统会有哪些通知?.....	7
1.12 如何唯一指定歌曲播放?.....	7
1.13 如何临时播放特定歌曲或者音频.....	8
1.14 安防报警支持.....	8
1.15 TCP 控制注意事项.....	9
1.16 485 协议转换器控制说明.....	9
1.17 常用指令.....	9
2. 命令描述.....	10
2.1 搜寻设备.....	11
2.2 心跳命令.....	12
2.3 JSON 扩展命令.....	13
3. JSON 命令概述.....	14
3.1 命令模板.....	14
3.2 错误说明.....	15
3.3 命令样例.....	15
4. 音乐 JSON 命令说明.....	17
4.1 系统信息.....	17
4.2 通道信息.....	18
4.3 打开通道.....	19
4.4 关闭通道.....	20
4.5 歌曲列表.....	20
4.6 文件夹内容列表.....	21
4.7 媒体信息.....	22
4.8 播放指定歌曲.....	23
4.9 云音乐搜索.....	24
4.10 播放下一曲.....	26
4.11 播放上一曲.....	26
4.12 暂停播放.....	26
4.13 继续播放.....	26
4.14 指定时间播放.....	27

4.15 设置播放模式.....	27
4.16 设置音量.....	27
4.17 设置音效.....	28
4.18 设置静音.....	29
4.19 切换音源.....	29
4.20 播放信息.....	30
4.21 收藏歌曲.....	31
4.22 取消收藏.....	32
4.23 修改主机名称.....	32
4.24 修改通道名称.....	32
5. 高级音乐 JSON 指令说明.....	33
5.1 设置低音.....	33
5.2 设置高音.....	33
5.3 播放下一专辑.....	34
5.4 播放上一专辑.....	34
5.5 下载歌曲.....	34
5.6 随机播放.....	35
5.7 专辑列表.....	35
5.8 专辑内歌曲列表.....	36
5.9 party 操作.....	37
5.10 对讲操作.....	38
5.11 通道配置.....	39
5.12 日志上传.....	40
5.13 软件更新.....	40
5.14 系统操作.....	41
5.15 修改主机时间.....	42
5.16 获取主机时间.....	42
5.17 触发或取消警报.....	42
5.18 定时器.....	43
5.19 延迟关机定时器.....	46
5.20 FM 操作.....	47
5.21 WIFI 配网.....	48
5.22 获取网络信息.....	50
5.23 系统配置.....	51
5.24 本地音乐文件操作列表.....	52
5.25 歌单操作.....	53
5.26 音乐场景.....	55
5.27 云音乐 VIP.....	55
6. 门铃 JSON 命令说明.....	58
6.1 门铃信息.....	58
6.2 配置门铃.....	59
6.3 保持状态.....	59
6.4 查看视频.....	60
6.5 向门铃说话.....	60

6.6 开门.....	60
6.7 门铃触发.....	61
7. JSON 样例指令.....	61
8. 老协议样例指令.....	63
9. 485 命令说明.....	67

1. FAQ

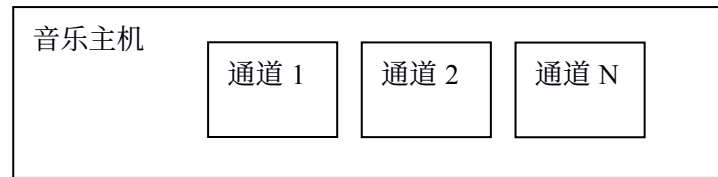
说明:为了方便协议调试,我们有提供一个简单的调试助手,有需要的请向商务索要.

1.1 协议控制方式?

使用 UDP/TCP 控制 , 设备监听端口: 10061

协议控制基本上都是基于通道控制的, 而一台音乐主机可以包含 1~8 个通道(对应通道编

号 0~7), 每个通道都是一个独立的播放单元。控制时, 手机或者其他控制设备需要与音乐主机处于同一局域网内才能控制。



1.2 如何搜索设备?

使用 UDP 广播的方式发送 搜寻主机命令(ce 00 ce), 局域网里所有设备都会返回主机信息。

注意: 主机 DHCP 获取 IP 时, 无论是 UDP 控制还是 TCP 控制, 都需要发送 UDP 广播搜索设备。

主机固定 IP 时, 可以通过预设的 IP 直接控制

1.3 如何获取设备 IP?

a. 使用 UDP 广播的方式发送 搜寻主机命令(ce 00 ce), 设备会有反馈信息

b. 依据 UDP 返回信息可以获得设备 IP, 即反馈 UDP 包的发送 IP。

或者

发送 system.info 命令, 设备会返回系统信息, 其中有 ip 字段。

1.4 为何发送命令后设备没有任何反应?

设备仅会响应发送搜索设备 (ce 00 ce) 然后保持心跳(cf 00 cf)的控制设备的命令, 搜索设备 (ce 00 ce) 和心跳命令(cf 00 cf)任何时候都会有返回。

心跳命令建议 5~10 秒一次, 设备 30 秒未收到心跳会自动忽略控制设备的命令, 并且不再发送通知。

说明: 心跳在 UDP/TCP 控制时必须保持, 串口控制可以忽略。

1.5 什么是通道编号?

悠达设备支持多分区独立控制, 故在控制时需要指定控制的分区编号 (通道编号), 编号从 0 开始。如: y4 的通道编号为 0~3, 分别对应第一~第四通道。t3/w5 等仅有一个分区, 默认为 0。

1.6 如何计算校验?

校验算法(c 代码):

```
uint8_t xorSum(const uint8_t* data, uint16_t size){
```

```

if (data == 0 || size == 0) return 0;
uint8_t sum = data[0];
uint16_t i = 1;
for (; i < size; ++i) sum ^= data[i];
return sum;
}

```

1.7 如何控制设备？

- 使用 UDP 广播发送 ce 00 ce 搜索局域网中的设备(端口 10061)
- 搜索到设备后，开启心跳保持,周期性发送 cf 00 cf 命令（5~10 秒一次，30 秒超时）
- 发送控制命令，开始控制设备
- 设备状态发生变化时，会主动发送信息到控制端(发送到控制端之前发送控制命令对应的 socket 的接收端口)

1.8 party 功能是什么？

party 是一种能让同一局域网中所有背景音乐设备同步播放同一歌曲的功能。当前同一局域网中只能创建一个 party，只支持 MP3 和云音乐两个音源。无线设备也可以加入 party，但是为了稳定，建议使用有线连接。party 功能只要设备处于同一局域网即可，没有其他要求。

- 一个局域网只能发起一个 party. 当通道处于 mp3 或者云音乐下时发起 party，party 的播放歌曲和播放列表会与发起通道保持一致，处于其他音源下则默认播放 mp3 音源的歌曲。
- 任何一个已经打开的通道都可以发起 party，发起 party 后，网络中其他已经打开的通道会自动加入，并且网络中的所有通道都可以手动加入(要先打开通道)，退出，或者关闭 party。
- 处于 party 中的通道的音量是独立的，调节音量不会影响其他处于 party 中通道的音量。
- party 中任意通道切歌，所有其他 party 中的通道会同步切。
- 当 party 中的通道都退出后，或者一直无法正常播放，party 会自动关闭
- 发起 party 的通道关闭，只要 party 中还有其他通道，party 不会关闭。
- 发起 party 后的 mp3 列表与发起 party 的通道保持一致。

一个网络内多 PARTY 的实现方式：默认情况下同一网络中的多台中央机只能开启一个 PARTY,因为 PARTY 是局域网共享的。但是实际上每一台中央机是可以配置为 PARTY 独享的(即与其他中央机隔离的,自己内部的通道组成一个 PARTY 组),这样就可以每一台中央机内部可以组成一个 PARTY 组,也就是物理上的多 PARTY 分组了。

配置单主机 PARTY 的命令为:{"config.set","arg":{"onlyLocalHost":1}}。该选项同时会额外带来一些稳定性的优化。

1.9 主机级联是什么？

一个网络中的多台中央机，可以将一台设为主机(出厂默认都为从机)，其他为从，—

个网络里只能有一台主机(若有多台,则其他主机会自动重启), 设置后设备需要重启。则主机将自己的 U 盘共享给从机, 从机器会忽略自己的 U 盘。并且主机(Y6/Y8)支持 DLNA&AIRPLAY 的 PARTY 推送(最新程序非主机已经可以通过协议开启 party 推送)。

配置主丛机的命令为:{"config.set","arg":{"master":1}}

说明:一般不建议开启该功能

1.10 手机推送 PARTY 是什么?

当 Y6/Y8 设置为主机后(需要重启,最新的程序已经可以在非主机设备通过协议开启, 最新程序可以不设为主机), 手机可以在 DLNA 或 AIRPLAY 的界面中看到一个 xxxx#PARTY 的推送点, 选择它以后, 网络中所有通道会自动打开, 进入 party, 同步播放推送的歌曲, 手机调音所有通道同步调节。

开启推送 PARTY 的命令为:{"config.set","arg":{"airplayParty":1,"dlnaParty":1}}.

1.11 系统会有哪些通知?

操作设备的过程中,设备状态发生变化时会有一些主动通知:

system.dateTime, update.info, timer.list

channel.name,channel.state,talk.info,party.info

player.volume,player.mute,player.eq,player.bass,player.treb,player.source,player.mode,player.time

player.state,player.info,player.list,player.like,player.download

radio.autoScan 等

具体触发条件请参考命令描述

1.12 如何唯一指定歌曲播放?

指定歌曲播放可以通过 ID, URL, NAME 三种方式:

ID: 播放前,实时获取歌曲列表获取 ID, 可以唯一指定歌曲播放, 但是 ID 在插拔 U 盘 SD 卡, 断电或者网络变化等都有可能导致变化.推荐使用的方式是播放前获取列表以获取最新的 ID. (若仅插了 U 盘或者仅插 SD 卡,并且没有修改里面的歌曲之前,ID 是会保持不变的)

URL:可以任何时候唯一指定歌曲播放,但是若歌曲不存在,则会失败.一般只有在需要保存到配置文件中长期使用才通过这种方式.其他大部分时候仅需要 ID 指定即可满足需求.

NAME:通过名字可以模糊指定播放的歌曲,但是当有多首重名歌曲时,不能确定具体播放的歌曲.

获取 ID, URL, NAME 的协议有: list.mediaList(所有歌曲列表), list.dirNodeList(文件夹结构列表)等

指定歌曲播放的协议为: player.play

Id,name 在默认获取列表时就会返回,URL 仅有在额外指定 metaLevel 为 3 才会返回,主要是

由于 URL 数据比较大的缘故,默认是不返回的. 如需要单独获取某个 ID 对应的 URL 可以使用 media.info 获取.

1.13 如何临时播放特定歌曲或者音频

系统是支持临时播放指定音频(报警音, 触发提示音,语音播报等),然后返回之前的状态.使用的命令为:player.play

具体如下:

```
{
  "call": "player.play",
  "arg": {
    "tmp": "true", //临时播放,播放完会返回之前的状态
    "url": "/local/sda1/文件夹名字/歌曲名称.mp3", //例如: /local/sda1 /播报语音/欢迎.wav, 也可以是 rtps://, http://, ftp:// 等网络音频
    "volume": 100, //可选, 0~255, 默认为之前的音量
  }
}
```

说明:也可以使用 pos 或 id 来指定(需要同时指定 source),但是 url 更准确.

1.14 安防报警支持

中央机系列是支持安防报警的, 具体的报警说明请参考命令 alarm.set, 它可以有两种方式接入:

1.通过机器背面的 DB9 接口的针脚短接产生触发报警信号:

火警 1-4 短接 触发后所有通道都会报警

盗警 1-8 短接 触发后仅有第一个通道会报警, 通过关闭通道或者切换音源可以解除

2.通过串口或者网络发送报警触发命令触发报警: alarm.set 命令

3.报警音可以自定义,将自定义文件放置在 U 盘根目录下(支持 mp3,wav 等):

火警: /yodar_custom_media/fire_alarm

盗警: /yodar_custom_media/steal_alarm

注意: 仅有机器设为主机或者不级联(不与其他主机联动)才会响应报警, 具体请使用悠达 APP 在设置修改。

4. **最新的程序可以自定义触发后执行的动作:**

使用 config.set 设置 customAlarm 为 true 可以开启自定动作:

开启自定动作后, 火警触发会让系统自动播放, 盗警会让系统退出报警动作, 报警时的默认参数如下:

播放文件夹: /yodar_custom_media/alarms

播放音乐路径: /yodar_custom_media/alarm 默认音乐文件, 可以与 albumUrl 同时指定

播放时的音量: 160 使用 config.set 设置 customAlarmVolume 修改(0-255)

默认音源: 临时音源 使用 config.set 设置 customAlarmSource 修改(一般不建议修改)

1.15 TCP 控制注意事项

使用 TCP 控制时,协议会有点点区别, 需要在协议的头部添加两个字节的长度描述:

协议字段	位置	长度	
命令长度	第一,二个字节	2	命令长度(不包含本身的两个字节)大端模式,高字节在前,低字节在后
命令	第三~N 个字节	N	协议文档描述命令(UDP 命令)

例如心跳命令, 使用 UDP 时是: cf 00 cf 使用 TCP 时是: 00 03 cf 00 cf

1.16 485 协议转换器控制说明

485 协议转换器支持通过 485 总线控制局域网中的悠达背景音乐设备(可同时控制 1 台或多台设备)。**第三方通过 485 对接转换器，转换器再通过局域网控制音乐设备。**

- a. 485 端支持的是悠达网络协议, 新的 JSON 命令, 老的 485 协议(不支持选曲), 默认为悠达网络协议,其他模式需要额外配置.
 - b. 485 协议转换器默认配置为 9600 8n1, 可在出厂前定制配置.
 - b. 不同音乐设备通过 485 编号来区分,与通道编号是不一样的, 并且编号需填写在协议第二个字节, 0xff 表示广播 .
 - 如: 485 编号为 1-2, 那么第二个字节则需要填写为 0x12.
 - d. 设备搜索可以使用 ce ff 31 或者直接手动配置, 心跳命令 cf 00 cf 不用保持, 忽略.
- 说明:协议转换器有一个配置小工具,可以在配置波特率和协议,需要的请向商务要.**

1.17 常用指令

搜索主机: CE 00 CE

心跳保持: CF 00 CF

触发盗警: 0F 00 00 36 7B 22 63 61 6C 6C 22 3A 22 61 6C 61 72 6D 2E 73 65 74 22 2C 22 61 72 67 22 3A 7B 22 74 79 70 65 22 3A 20 31 2C 20 22 73 74 61 74 65 22 3A 32 7D 7D 1C

关闭盗警: 0F 00 00 36 7B 22 63 61 6C 6C 22 3A 22 61 6C 61 72 6D 2E 73 65 74 22 2C 22 61 72 67 22 3A 7B 22 74 79 70 65 22 3A 20 31 2C 20 22 73 74 61 74 65 22 3A 30 7D 7D 1E

通道 0 参考命令

打开: 0F 00 00 23 7B 22 63 61 6C 6C 22 3A 22 63 68 61 6E 6E 65 6C 2E 6F 70 65 6E 22 2C 22 74 6F 22 3A 31 7D 77

关闭: 0F 00 00 24 7B 22 63 61 6C 6C 22 3A 22 63 68 61 6E 6E 65 6C 2E 63 6C 6F 73 65 22 2C 22 74 6F 22 3A 31 7D 12 7

下一首: 0F 00 00 1F 7B 22 63 61 6C 6C 22 3A 22 70 6C 61 79 65 72 2E 70 6C 61 79 4E 65 78 74 22 7D 30

上一首: 0F 00 00 1F 7B 22 63 61 6C 6C 22 3A 22 70 6C 61 79 65 72 2E 70 6C 61 79 50 72 65 76 22 7D 26

切换本地音源: 0F 00 00 33 7B 22 63 61 6C 6C 22 3A 22 70 6C 61 79 65 72 2E 73 65 74 53 6F 75 72 63 65 22 2C 22 61 72 67 22 3A 7B 22 73 6F 75 72 63 65 22 3A 30 7D 7D 13

切换云音乐音源: 0F 00 00 33 7B 22 63 61 6C 6C 22 3A 22 70 6C 61 79 65 72 2E 73 65 74 53 6F 75 72 63 65 22 2C 22 61 72 67 22 3A 7B 22 73 6F 75 72 63 65 22 3A 35 7D 7D 16

切换 FM 音源: 0F 00 00 33 7B 22 63 61 6C 6C 22 3A 22 70 6C 61 79 65 72 2E 73 65 74 53 6F 75 72 63 65 22 2C 22 61 72 67 22 3A 7B 22 73 6F 75 72 63 65 22 3A 31 7D 7D 12

切换 AUX 音源: 0F 00 00 33 7B 22 63 61 6C 6C 22 3A 22 70 6C 61 79 65 72 2E 73 65 74 53 6F 75 72 63 65 22 2C 22 61 72 67 22 3A 7B 22 73 6F 75 72 63 65 22 3A 32 7D 7D 11

音量+: 0F 00 00 20 7B 22 63 61 6C 6C 22 3A 22 70 6C 61 79 65 72 2E 61 64 64 56 6F 6C 75 6D 65 22 7D 65

音量-: 0F 00 00 20 7B 22 63 61 6C 6C 22 3A 22 70 6C 61 79 65 72 2E 64 65 63 56 6F 6C 75 6D 65 22 7D 66

静音: 0F 00 00 2F 7B 22 63 61 6C 6C 22 3A 22 70 6C 61 79 65 72 2E 73 65 74 4D 75 74 65 22 2C 22 61 72 67 22 3A 7B 22 6D 75 74 65 22 3A 31 7D 7D 0E

取消静音: 0F 00 00 30 7B 22 63 61 6C 6C 22 3A 22 70 6C 61 79 65 72 2E 73 65 74 4D 75 74 65 22 2C 22 61 72 67 22 3A 7B 22 6D 75 74 65 22 3A 20 30 7D 7D 30

暂停播放: 0F 00 00 1C 7B 22 63 61 6C 6C 22 3A 22 70 6C 61 79 65 72 2E 70 61 75 73 65 22 7D 62

继续播放: 0F 00 00 1D 7B 22 63 61 6C 6C 22 3A 22 70 6C 61 79 65 72 2E 72 65 73 75 6D 65 22 7D 08

随机播放: 0F 00 00 21 7B 22 63 61 6C 6C 22 3A 22 70 6C 61 79 65 72 2E 70 6C 61 79 52 61 6E 64 6F 6D

临时播放本地第一首歌: 0F 00 00 4A 7B 22 61 72 67 22 3A 7B 22 70 6F 73 22 3A 30 2C 22 73 6F 75 72 63 65 22 3A 30 2C 22 74 6D 70 22 3A 74 72 75 65 7D 2C 22 63 61 6C 6C 22 3A 22 70 6C 61 79 65 72 2E 70 6C 61 79 22 2C 22 74 6F 22 3A 22 30 22 7D 2F

临时播放本地第二首歌: 0F 00 00 4A 7B 22 61 72 67 22 3A 7B 22 70 6F 73 22 3A 31 2C 22 73 6F 75 72 63 65 22 3A 30 2C 22 74 6D 70 22 3A 74 72 75 65 7D 2C 22 63 61 6C 6C 22 3A 22 70 6C 61 79 65 72 2E 70 6C 61 79 22 2C 22 74 6F 22 3A 22 30 22 7D 2E

临时播放本地第三首歌: 0F 00 00 4A 7B 22 61 72 67 22 3A 7B 22 70 6F 73 22 3A 32 2C 22 73 6F 75 72 63 65 22 3A 30 2C 22 74 6D 70 22 3A 74 72 75 65 7D 2C 22 63 61 6C 6C 22 3A 22 70 6C 61 79 65 72 2E 70 6C 61 79 22 2C 22 74 6F 22 3A 22 30 22 7D 2D

临时播放本地第四首歌: 0F 00 00 4A 7B 22 61 72 67 22 3A 7B 22 70 6F 73 22 3A 33 2C 22 73 6F 75 72 63 65 22 3A 30 2C 22 74 6D 70 22 3A 74 72 75 65 7D 2C 22 63 61 6C 6C 22 3A 22 70 6C 61 79 65 72 2E 70 6C 61 79 22 2C 22 74 6F 22 3A 22 30 22 7D 2C

[更多样例指令请查看文档尾部.](#)

2. 命令描述

设备控制流程:

- 1.使用 ce 命令搜索设备, 所有网络中的支持该协议的设备都会反馈 ef 命令,通过该返回获得设备的类型,唯一 ID,名字等信息, 并且设备会在接收到该命令时记录控制端的 IP 和端口。
- 2.向所有搜索到的设备发送 cf 命令保持心跳(建议 10s 一次, 超过 30s 未发送, 则不再响应 json 控制命令).
- 3.使用具体的 json 命令(封装在 0f 命令中)控制设备.

注意:使用 TCP 控制时,协议头部需要添加两个字节用来描述命令长度,大端模式,高字节在前,低字节在后,例如心跳命令, 使用 UDP 时是 : cf 00 cf 使用 TCP 时是: 00 03 cf 00 cf

老的字节协议和 485 协议请参考文档尾部的说明

协议的第二个字节的低四位默认表示设备通道号, 某些与通道无关的命令会忽略掉 (填 0x00 或者 0xff 都可以)

但是使用 485 协议转换器控制时, 第二个字节用来表示设备的 485 编号, 0xff 表示所有设

备，与通道无关的命令返回默认为 0

2.1 搜寻设备

a) 搜寻设备消息

协议字段	位置	长度	数值定义	备注
Command	第一个字节	1	0xce	命令 id
Address	第二个字节	1	0x00	忽略
Checksum	第三个字节	1	第 1 个字节开始的异或	校验消息正确性

说明:UDP 广播发送，可以尝试间隔几秒多次发送，确保获得完整设备列表。

a) 设备信息返回

协议字段	位置	长度	数值定义	备注
Command	第一个字节	1	0xef	命令 id
Address	第二个字节	1	0xff	忽略
Len	第三个字节	1	Len	信息总长度
Device Type	第四个字节		0x55	55T
			0x56	55T 扩展带蓝牙
			0x57	55T 扩展不带蓝牙
			0x58	i5 扩展带蓝牙
			0x59	i5 扩展不带蓝牙
			0x60	i7
			0x61	I72
			0x62	B5 plus
			0x63	I10
			0x64	I12
			0x71	吊顶主机(T3)
			0x72	Y4
			0x73	W5
			0x74	Y2
			0x77	Y1
			0x75	红外学习器
			0x76	门铃
			0x79	Y6
			0x7A	Y8
			0x7B	预留 Y12
			0x7C	I31
			0x7D	I32
			0x7e	I35
			0x7f	I36

			0x90	I37
			0x0E	中控主机
			0x10	中控主机 V2
			0x80	WIFI 开关
Model	第五个字节	1	1 ~ 16	通道数
Iphone id	第六个字节	1	0x00	忽略
Name id	第七个字节	1	0x01	
Name len	第八个字节	1	n	名字长度
Name	第九~n+8 字节	n	主机名字	若 n 为 0，则忽略该项
No.id	第 n+9 字节	1	0x02	
No.len	第 n+10 字节	1	0x08	唯一标识码长度
No.	第 n+11~n+18 字节	8	8 个字节的唯一标识码	
No.	收藏字段(可选)	1	0x03	收藏
		1	1	收藏字段长度
		1	0	不带收藏功能
			1	带收藏功能
No.	锁定字段(可选)	1	0x04	锁定
		1	1	锁定字段长度
		1	0	不锁定
			1	锁定
End id	第 n+19 字节	1	0xff	
Checksum	第 n+20 字节	1	第 1 个字节开始的异或	校验消息正确性

2.2 心跳命令

a) 心跳命令消息

协议字段	位置	长度	数值定义	备注
Command	第一个字节	1	0xcf	命令 id
Address	第二个字节	1	0	忽略
Checksum	第三个字节	1	第 1 个字节开始的异或	校验消息正确性

说明:心跳在使用 UDP/TCP 控制时(建议 10s 一次)必须保持

b) 心跳命令消息

协议字段	位置	长度	数值定义	备注
Command	第一个字节	1	0xcf	命令 id
Address	第二个字节	1	0	忽略
Data1	第三个字节	1	0 为正常，1 为异常。	
Data2	第四个字节	1	0	

Checksum	第五个字节	1	第 1 个字节开始的异或	校验消息正确性
----------	-------	---	--------------	---------

2.3 JSON 扩展命令

c) JSON 扩展消息

协议字段	位置	长度	数值定义	备注
Command	第一个字节	1	0x0f	命令 id
Address	第二个字节	1	通道编号放低四位(高四位一般忽略,某些特殊场景会使用)	地址
length	第三个字节	1	命令长度高字节	整个命令的长度, 包含校验字节
	第四个字节		命令长度低字节	
JSON 命令	第五~N + 4 字节	N	JSON 文本内容(UTF-8)	
Checksum	第 N + 5 个字节	1	第 1 个字节开始的异或	校验消息正确性

d) JSON 扩展命令返回消息

协议字段	位置	长度	数值定义	备注
Command	第一个字节	1	0x0f	命令 id
Address	第二个字节	1	通道编号放低四位(高四位一般忽略,某些特殊场景会使用)	地址
length	第三个字节	1	命令长度高字节	整个命令的长度, 包含校验字节
	第四个字节		命令长度低字节	
JSON 命令	第五~N + 4 字节	N	JSON 文本内容(UTF-8)	
Checksum	第 N + 5 个字节	1	第 1 个字节开始的异或	校验消息正确性

3. JSON 命令概述

在原有的字节协议中新增 json 命令, 具体定义如下:

1 0x0f 命令 ID
2 0x00 接收地址 通道编号放低四位(高四位一般忽略,某些特殊场景会使用)
3~4 0x00 0x00 命令总长度, 第 3 个字节为长度的高位, 第四字节为长度的低位
5~n 0x00~ JSON 内容
n+1 Checksum 除校验字节外其他字节异或([C 样例代码请参考 FAQ](#))

3.1 命令模板

//CALL

```
{
  "call": "class.method", //调用方法
  "tag": "12345678", //调用 ID,主要是用来识别异步调用的返回,由 call 提供, 可选
  "arg": { //调用参数,可选
    "arg1": "",
    "arg2": ""
  }
}
```

//ACK

```
{
  "ack": "class.method", //调用方法
  "tag": "12345678", //调用 ID,主要是用来识别异步调用的返回, 由 call 提供, 原样返回,可选
  "arg": { //返回参数,可选
    "arg1": "",
    "arg2": ""
  }
}
```

或

```
{
  "ack": "class.method", //调用方法
  "tag": "12345678", //调用 ID,主要是用来识别异步调用的返回,可选, 由 call 提供, 原样返回
  "code": 1, //调用放回状态, 0 或无 成功, 其他值为错误号
  "arg": { //返回参数,可选
    "msg": "" //可选
  }
}
```

3.2 错误说明

0xfe 系统错误
0xfd 无效的参数

0xfc 不支持的命令

0xfb 异步操作中，稍后重试，该错误的 arg 中可能会有一个建议重试时间，若无，则默认为 1000ms

"arg": { //返回参数,可为空,或一个对象,或一个数组,可选

 "wait":100, //ms 毫秒

}

0xfa 验证失败

//NOTIFY，系统主动通知

{

 "notify": "class.method", //调用方法

 "arg": { //返回参数,可为空

 }

}

校验算法(c 代码):

```
uint8_t xorSum(const uint8_t* data, uint16_t size){
```

```
    if (data == 0 || size == 0) return 0;
```

```
    uint8_t sum = data[0];
```

```
    uint16_t i = 1;
```

```
    for (; i < size; ++i) sum ^= data[i];
```

```
    return sum;
```

```
}
```

注意:

"arg"中的可选字段若无,其默认值为: (int 0, string "", boolean false)

3.3 命令样例

//获取当前播放信息

//请求

0f 10 00 21 7b 0a 20 20 22 63 61 6c 6c 22 20 3a 20 22 70 6c 61 79 65 72 2e 69 6e 66 6f 22 0a 7d 33

...!{. "call" : "player.info".}3

//返回

0f 10 01 3a 7b 22 61 63 6b 22 3a 22 70 6c 61 79 65 72 2e 69 6e 66 6f 22 2c 22 61 72 67 22 3a 7b 22 61 6c 62 75 6d 22 3a 22 e6 b5 81 e8 8a b1 20 4c 6f 76 65 20 48 65 72 62 79 22 2c 22 61 72 74 69 73 74 22 3a 22 e8 a2 81 e5 a8 85 e7 bb b4 22 2c 22 62 61 73 73 22 3a 39 2c 22 64 6f 77 6e 6c 6f 61 64 22 3a 30 2c 22 64 75 72 61 74 69 6f 6e 22 3a 30 2c 22 65 71 22 3a 32 35 34 2c 22 69 64 22 3a 33 2c 22 6c 69 6b 65 22 3a 30 2c 22 6d 75 74 65 22 3a 31 2c 22 6e 61 6d 65 22 3a 22 e6 b5 81 e8 8a b1 20 4c 6f 76 65 20 48 65 72 62 79 22 2c 22 70 69 63 55 72 6c 22 3a 22 68 74 74 70 3a

2f 2f 69 6d 67 63 61 63 68 65 2e 67 74 69 6d 67 2e 63 6e 2f 6d 65 64 69 61 73 74 79 6c 65 2f 6d
75 73 69 63 5f 76 31 32 2f 69 6d 67 2f 64 65 66 61 75 6c 74 5f 61 6c 62 75 6d 2e 70 6e 67 22 2c
22 70 6c 61 79 54 69 6d 65 22 3a 30 2c 22 73 6f 75 72 63 65 22 3a 35 2c 22 73 74 61 74 65 22 3a
30 2c 22 74 72 65 62 22 3a 36 2c 22 76 6f 6c 75 6d 65 22 3a 31 36 30 7d 7d 9a

....{"ack":"player.info","arg":{"album":"..... Love
Herby","artist":".....","bass":9,"download":0,"duration":0,"eq":254,"id":3,"like":0,"mute":1,"na
me":"..... Love Herby","picUrl":"http://imgcache.gtimg.cn/mediastyle/music_v12/img/
default_album.png","playTime":0,"source":5,"state":0,"treb":6,"volume":160}}.

//获取系统信息

```
{"call":"system.info","tag":"016"}
{"ack":"system.info","arg":
{"buildTag":"devel@20161112143647","channelCount":4,"cloudType":1,"dateTime":"2016-11-23
15:45:20","deviceType":"YODAR_Y4","internet":1,"ip":"0.0.0.0","isRegister":-
1,"mac":"00:11:22:12:34:56","name":"Yodar-
123456","network":1,"nfs":0,"update":0,"usb":0,"uuid":"001122123456","version":161112},"tag":
"016"}
```

//获取播放信息

```
{"call":"player.info"}
{"ack":"player.info","arg":{"bass":9,"duration":216,"eq":254,"id":0,"mute":0,"name":"Avril
Lavigne - Girlfriend [mqms2]","playTime":9,"source":0,"state":3,"treb":6,"volume":160}}
```

//获取列表

```
{"arg":{"begin":0,"id":0,"size":15,"type":2},"call":"list.dirNodeList"}
{"ack":"list.dirNodeList","arg":{"id":0,"nodeList":[{"id":1,"name":" 新 歌 ","type":1},
{"id":7,"name":" 排 行 ","type":1},{id":8,"name":" 歌 单 ","type":1},{id":9,"name":" 分
类 ","type":1},{id":10,"name":" 电 台 ","type":1},{id":11,"name":" 歌 手 ","type":1},
{"id":30,"name":"专辑","type":1},{id":57,"name":"搜索","type":1}],total":8}}
```

//播放指定歌曲

```
{"arg":{"albumId":2,"id":3},"call":"player.play","tag":"007"}
{"ack":"player.play","tag":"007"}
```

//模糊播放

```
{"call":"player.play","arg":{"name":"周杰伦","source":5,"autoOpen":true}}
```

//临时播放

```
{"call":"player.play","arg":{"url":"/local/欢迎.wav","volume":180,"tmp":true}}
```

//打开通道

```
{"call":"channel.open"}
```

//切换到本地音源


```
{"call": "player.setSource", "arg": {"source": 0}}
```

//获取播放列表

```
{"arg": {"begin": 0, "name": "play", "size": 2}, "call": "list.mediaList", "tag": "017"}  
{"ack": "list.mediaList", "arg": {"id": -2, "name": "play", "nodeList": [{"album": "长", "artist": "李宇春", "id": 0, "name": "西门少年"}, {"album": "间谍同盟", "artist": "张杰", "id": 1, "name": "Give You My World (《间谍同盟》电影全球唯一推广曲)"}], "total": 100}, "tag": "017"}
```

4. 音乐 JSON 命令说明

每条正确的 CALL 命令,都会有对应的 ACK,告知控制端命令是否执行成功,有部分还会触发一条或多条相关的 NOTIFY. 并且 ACK 和 NOTIFY 的返回顺序是不确定的. 其中协议中 tag 字段由控制端提供(是可以选的, 可用来标记或者匹配命令), 主机会在 ACK 中原样返回.

说明: JSON 命令在 485/UDP/TCP 中都是通用的

4.1 系统信息

系统信息描述是整个设备信息

//CALL

```
{  
  "call": "system.info",  
  "arg": { //可选  
    "moreInfo": false //是否获取更多信息,可选  
  }  
}
```

//ACK

```
{  
  "ack": "system.info",  
  "arg": {  
    "name": "", //主机名称,与通道名不一样  
    "deviceType": "", //设备类型 YODAR_Y4,YODAR_T3,  
YODAR_Y2,YODAR_Y6,YODAR_Y8,YODAR_I32,YODAR_W5  
    "deviceTypeName": "", //设备类型的名字 (可选)  
    "version": 0, //版本号  
    "romVersion": "", //ROM 版本号  
    "uuid": "", //机器唯一 ID  
    "isRegister": 0, //是否注册  
    "dateTime": "", //主机时间 yyyy-MM-dd hh:mm:ss  
    "internet": 0, //是否链接外网
```

```

"mac":"","//mac 地址
"ip":0,//路由网络端 IP WAN
"wanMode":"","//可选,外网模式,ethernet 以太网, wifi 无线,disconnect 未连接
"network":0,//是否链接路由
"update":0,//是否有更新, 0 已是最新,1 已经更新完成(需重启才能使用最新),2 正在检测
或更新
"usb":0,//可选,是否插入 SD 卡
"sd":0,//可选,是否插入 U 盘
"tcpControl":0,//可选,是否支持 TCP 控制
"songList":0,//可选,是否支持歌单
"channelCount":0,//通道数
"channelInfoList":[//通道信息列表, 可选
    {
        "name":"","//通道名
        "state":0,//状态
        "source":0//音源
    }
],
}
}

```

4.2 通道信息

通道信息描述的时每个通道的信息，一台设备上可能有多个通道，每个通道的信息都是不一样的，需要单独获取。背景音乐的操作基本上都是基于通道进行的。

```

//CALL
{
    "call":"channel.info",
    "arg">{
        "moreInfo":false
    }
}
//ACK
//NOTIFY
{
    "call":"channel.info",
    "arg":{
        "name":"","
        "state":0,//0 关闭,1 打开, 0xfe 错误
        "source":0,
        "party":0,//是否开启 party,可选,默认 0
    }
}

```

```

    "playerInfo":{//可选,与 player.info 内容一致
    },
    "partyInfo":{//可选,
        "state":0
    }
    "talkInfo":{//可选,
        "state":0,
        "peerIp":"","//可选
        "peerId":"","//可选
    }
    }
}

```

4.3 打开通道

```

//CALL
{
    "call":"channel.open",
    "arg":{//可选
        "source":0,//可选,打开时的音源
        "notPlay":false,//可选, 打开后是否不自动播放, 默认为 false
        "mediaUrl":""," //可选打开播放的歌曲, 专辑
        "albumUrl":"","
        //或
        "mediaId":0,
        "albumId":0
    }
}

```

```

//NOTIFY
{
    "notify":"channel .state ",
    "arg":{
        "state":0 //0 关闭 1 打开
    }
}

```

可能触发的 NOTIFY: channel.state, player.info, player.*

4.4 关闭通道

```

//CALL
{

```

```
"call": "channel.close"
}
```

可能触发的 NOTIFY: channel.state, player.state

4.5 歌曲列表

一层列表结构（列表结构），可以获取设备所有歌曲列表,当前播放列表,收藏列表和历史播放列表等。

```
//CALL
{
  "call": "list.mediaList",
  "arg": {
    "name": "", //列表名称, play, like, history, all, 可选
    "source": 0, //可选, 指定音源, 若无, 默认当前音源
    "begin": 0, //获取列表的开始位置, 可选
    "size": 4, //获取的数目, 可选
    "metaLevel": 0, // 0 自动, 1 最少(仅名字和 id) 2 一般(名字, id, 歌手, 专辑, 长度) 3, 全部
    (会包含 url 等), 可选, 默认 0
  }
}

//ACK
{
  "ack": "list.mediaList",
  "arg": {
    "name": "", //列表名称, play, like, history, all
    "total": 0, //列表总数, 注意不是返回的列表项数目
    "begin": 0, //获取列表的开始位置
    "id": 0,
    "nodeList": [
      {
        "id": 0, //歌曲 ID
        "name": "music", //歌曲名称
        "artist": "zjl", //歌手名称 //可选
        "album": "al", //专辑名称 //可选
        "url": "" //可选
      },
      {
        "id": 0, //歌曲 ID
        "name": "music", //歌曲名称
        "artist": "zjl", //歌手名称 //可选
      }
    ]
  }
}
```

```

        "album": "al", //专辑名称  //可选
        "url": "" //可选
    },
]
}
}
//NOTIFY
{
    "notify": "list.mediaList",
    "arg": {
        "name": "", //列表名称, play, like, history, all, 可选
    }
}

```

4.6 文件夹内容列表

多层结构,类似文件夹浏览.根目录 ID 为 0

```

//CALL
{
    "call": "list.dirNodeList",
    "arg": {
        "id": 0, //可选, 专辑 ID, 若无, 默认为 0
        "source": 0, //可选, 指定音源, 若无, 默认当前音源
        "type": 0, //可选, 0 文件, 1 文件夹, 2 文件和文件夹 若无, 则默认为 2
        "begin": 0, //可选, 获取列表的开始位置, 若无, 默认为 0
        "size": 4, //可选, 获取的数目, 若无, 默认由音乐主机自己决定
        "metaLevel": 0, // 0 自动, 1 最少(仅名字和 id) 2 一般(名字, id, 歌手, 专辑, 长度) 3, 全部(会
        包含 url 等), 可选, 默认 0
    }
}

//ACK
{
    "ack": "list.dirNodeList",
    "arg": {
        "id": 0, //文件夹 ID, 0 表示根文件夹, 若无, 则默认为 0
        "source": 0, //可选, 指定音源, 若无, 默认当前音源
        "total": 0, //列表总数, 注意不是返回的列表项数目
        "begin": 0, //获取列表的开始位置
        "nodeList": [
            {
                "id": 0, //歌曲或文件夹 ID
                "name": "music", //歌曲名称

```

```

        "type":"","//可选,可选,0 文件, 1 文件夹, 默认 0

        //type 为 0 才会有下面的字段
        "artist":"zjl", //歌手名称, 可选
        "album":"al", //专辑名称, 选
        "duration":82, //播放长度, 单位秒, 可选

        //type 为 1 才会有下面的字段
        "picUrl":"http://example.com/xx.jpg" //专辑图片 URL, 可选
    },
    {
        "id":0, //歌曲或文件夹 ID
        "name":"music", //歌曲名称
        "type":"","//可选,可选,0 文件, 1 文件夹, 默认 0

        //type 为 0 才会有下面的字段
        "artist":"zjl", //歌手名称 //可选
        "album":"al", //专辑名称 //可选
        "duration":82, //播放长度 单位秒 //可选

        //type 为 1 才会有下面的字段
        "picUrl":"http://example.com/xx.jpg" //专辑图片 URL, 可选
    },
]
}
}

```

4.7 媒体信息

可以通过该接口使用 ID 获取唯一 URL,设备重新上电后还会保持有效.

```

//CALL
{
    "call":"media.info",
    "arg":{
        "type":0, //0 歌曲, 1 专辑
        "id":0, //可选
        "url":"","//可选
        "source":0, //可选, 指定音源, 若无, 默认当前音源
    }
}
//ACK
{
    "ack":"media.info",

```

```

"arg" : {
    "name": ""
    "id": 0//歌曲 ID
    "url": ""//歌曲 url
    "picUrl": "http://example.com/xx.jpg"//专辑图片 URL,可选

    //type 为 0 才会有下面的字段
    "artist": "zjl", //歌手名称 //可选
    "album": "al", //专辑名称 //可选
    "duration": 82, //播放长度 单位秒 //可选
}
}

```

4.8 播放指定歌曲

播放歌曲默认是需要先打开通道的(可以通过参数 `autoOpen` 或 `tmp` 改变),并且仅能播放当前音源的歌曲(可以通过参数 `source` 改变)

指定歌曲播放可以有好几种方式:

1. 通过 `id` 指定,若同时指定 `albumId` 则会将指定专辑作为播放列表. 否则仅播放指定歌曲,然后继续播放之前的播放列表里的歌曲.`id`, `albumId` 可以通过 `list.mediaList`, `list.dirNodeList` 等获取.
2. 通过 `url` 指定,若同时指定 `albumUrl` 则会将指定专辑作为播放列表. 否则仅播放指定歌曲,然后继续播放之前的播放列表里的歌曲.`url`, `albumUrl` 可以通过 `media.info` 或 `system.dirNodeList` 或 `list.dirNodeList`(需要指定 `metaLevel` 为 3)获取. **URL 可以是网络中的音频(http, ftp 等)**
3. 通过 `name` 指定, 若同时指定 `albumName` 则会将指定专辑作为播放列表. 否则仅播放指定歌曲,然后继续播放之前的播放列表里的歌曲. `name`, `albumName` 可以通过 `list.mediaList`, `list.dirNodeList` 等获取. **当为云音乐时,若当前列表中没有匹配的歌曲,会自动云端搜索并播放**
4. 通过 `pos` 指定, 若同时指定 `albumPos` 则会将指定专辑作为播放列表. 否则仅播放指定歌曲,然后继续播放之前的播放列表里的歌曲. `pos` 是指歌曲或专辑按名字排序的序号.
5. 若指定了 `tmp` 为 `true`,则只能指定歌曲 `id` 或 `url`(其他字段忽略),并且支持本地歌曲和网络歌曲,它是临时播放会确保通道打开并播放指定歌曲,播放完成后自动回到之前的状态.
6. `id`, `pos`, `name`, `url` 同时指定时, 优先级为 `url > id > pos > name`, 并且在使用 URL 播放云音乐时,需要注意歌名显示的问题,若想要一直正确显示歌名,需要在 `meta` 字段中添加歌名.

```

//CALL
{
    "call": "player.play ",
    "arg" : {
        "name": ""//歌曲名字, 可选,
        "id": 0//歌曲 ID,可选, 可以为-1,表示不指定歌曲, 配合 albumId 使用
        "albumId": 0//专辑 ID,可选,若有效,则会将该专辑歌曲作为播放列表, 0 表示所有歌曲
        ,-0xf1 收藏歌曲, 与 id 配合使用 ,可选
    }
}

```

“pos”:0//播放制定下标歌曲,下标顺序由 list.mediaList 获取
 “albumPos”://播放制定下标专辑,下标顺序由 list.albumList 获取,与 pos 搭配使用,可选

“url”:“”// 歌曲 url, 可选 “/system/*” “/local/*” 指定播放本地指定歌曲,如“/local/sda1/test.mp3”,可以为空值,表示不指定歌曲,可以配合 albumUrl 使用
 “albumUrl”:“”//专辑 url, 必须以“/”结尾,与 url 配合使用,可选
 “ttsText”:“”//tts 播报(仅某些机型支持), 可选,
 “aiCmd”:“”//ai 文本命令(仅某些机型支持), 可选,
 “tmp”:false//可选, 若为 true, 则播放完指定歌曲后会跳回原来的音源,并且专辑无法指定

“volume”:0//可选
 “mode”: 0 //可选, 播放的同时修改播放模式: 0 循环播放, 1 随机播放, 2 单曲循环

“autoOpen”:false//可选, 若为 true, 则会先打开通道,否则, 之前未打开的通道无法播放歌曲

“source”:0//可选, 若无则默认当前音源,否则会在播放前切换到指定音源
 “meta”:{}//可选,在使用 URL 指定播放时需要使用
 “name”:“”//指定显示歌名
 }
 }
 }

可能触发的通知: player.info, player.state, player.time, player.list

4.9 云音乐搜索

```
//CALL
{
  "call": "list.search",
  "arg": {
    "key": "", //搜索的 key
    "type": 0, //搜索类型, 可选, 0 自动, 1 歌曲, 2 歌手, 3 专辑, 则默认为 0
    "begin": 0, //获取列表的开始位置
    "size": 0 //获取的数目
  }
}

//ACK
{
  "ack": "list.search",
  "arg": {
    "total": 0, //列表总数, 注意不是返回的列表项数目
    "begin": 0, //获取列表的开始位置
    "key": "", //搜索的 key
  }
}
```



```

"type":0,//搜索类型, 可选,0 自动, 1 歌曲 , 2 歌手, 3 专辑,则默认为 0
"id":0,//专辑 Id
"nodeList":[
  {
    "id":0,//歌曲或文件夹 ID
    "name":"music", //歌曲名称
    "type":"","//可选,可选,0 文件, 1 文件夹, 默认 0

    //type 为 0 才会有下面的字段
    "artist":"zjl", //歌手名称 //可选
    "album":"al", //专辑名称 //可选
    "duration":82,//播放长度 单位秒 //可选

    //type 为 1 才会有下面的字段
    "picUrl":"http://example.com/xx.jpg"//专辑图片的 URL,可选
  },
  {
    "id":0,//歌曲或文件夹 ID
    "name":"music", //歌曲名称
    "type":"","//可选,可选,0 文件, 1 文件夹, 默认 0

    //type 为 0 才会有下面的字段
    "artist":"zjl", //歌手名称 //可选
    "album":"al", //专辑名称 //可选
    "duration":82,//播放长度 单位秒 //可选

    //type 为 1 才会有下面的字段
    "picUrl":"http://example.com/xx.jpg"//专辑图片的 URL,可选
  },
]
}
}

```

4.10 播放下一曲

```

//CALL
{
  "call":"player.playNext"
}

```

可能触发的通知: player.info, player.state,player. time

4.11 播放上一曲

```
//CALL
```

```
{  
    "call": "player.playPrev"  
}
```

可能触发的通知: player.info, player.state, player.time

4.12 暂停播放

```
//CALL
```

```
{  
    "call": "player.pause"  
}
```

```
//NOTIFY
```

```
{  
    "notify": "player.state",  
    "arg" : {  
        "state": 0//当前播放状态 0 停止播放,2 缓冲, 3 播放, 暂停 4, 0xfe 错误  
    }  
}
```

可能触发的通知: player.state

4.13 继续播放

```
//CALL
```

```
{  
    "call": "player.resume"  
}
```

```
//NOTIFY
```

```
{  
    "notify": "player.state",  
    "arg" : {  
        "state": 0//当前播放状态 0 停止播放,2 缓冲, 3 播放, 暂停 4, 0xfe 错误  
    }  
}
```

可能触发的通知: player.state

4.14 指定时间播放

```
//CALL
```

```
{  
    "call": "player.seek",  
    "arg" : {  
        "time": 0//秒  
    }  
}
```

```

    }
}
//NOTIFY
{
    "notify": "player.time",
    "arg" : {
        "time": 0 //秒
    }
}

```

可能触发的通知: player. time

4.15 设置播放模式

```

//CALL
{
    "call": "player.setPlayMode",
    "arg" : {
        "mode": 0 //循环 0x00, 随机 0x01, 单曲 0x02
    }
}
//NOTIFY
{
    "notify": "player.mode",
    "arg" : {
        "mode": 0 //循环 0x00, 随机 0x01, 单曲 0x02
    }
}

```

可能触发的通知: player. mode

4.16 设置音量

```

//设置音量
//CALL
{
    "call": "player.setVolume",
    "arg" : {
        "volume": 0 //0 ~ 255
    }
}

```

"stageId": 0 //ai 文本命令, 可选,

//音量加

```
//CALL
{
  "call": "player.addVolume",
  "arg": {
    "stepValue": 8 //0 ~ 255, 可选, 默认 8
  }
}
```

//音量减

```
//CALL
{
  "call": "player.decVolume",
  "arg": {
    "stepValue": 8 //0 ~ 255, 可选, 默认 8
  }
}
```

```
//NOTIFY
{
  "notify": "player.volume",
  "arg": {
    "volume": 0 //0 ~ 255
  }
}
```

可能触发的通知: player.volume

4.17 设置音效

```
//CALL
{
  "call": "player.setEq",
  "arg": {
    "eq": 0 /// NORMAL 0x00, POP 0x01, SOFT 0x02, CLASSIC 0x03, JAZZ 0x04,
    ROCK 0x05
  }
}
//NOTIFY
{
  "call": "player.eq",
  "arg": {
    "eq": 0 /// NORMAL 0x00, POP 0x01, SOFT 0x02, CLASSIC 0x03, JAZZ 0x04,
    ROCK 0x05
  }
}
```

```
}
```

可能触发的通知: player.eq

4.18 设置静音

```
//CALL
```

```
{  
  "call": "player.setMute",  
  "arg" : {  
    "mute": true, //true, false  
  }  
}
```

```
}
```

```
//NOTIFY
```

```
{  
  "notify": "player.mute",  
  "arg" : {  
    "mute": true, //true, false  
  }  
}
```

```
}
```

可能触发的通知: player.mute

4.19 切换音源

音源主要有:

0 MP3,本地歌曲,包括 SD 卡,U 盘, 内置硬盘等

1 FM 电台

2 AUX2 外接输入 2

3 NET_RADIO 网络电台

4 AUX1 外接输入 1

5 CLOUD 云音乐

6 AIRPLAY IOS 系统内置支持的 WIFI 推送,只能手机推送时触发,无法主动设置

7 DLNA IOS/ANDROID 第三方支持的 WIFI 推送,只能手机推送时触发

8 BLUETOOTH 蓝牙 ,某些机型无法主动设置(由连接的手机端等自动触发)

9 SD 卡(部分机型)

10 U 盘(部分机型)

239 TMP 临时音源(由 player.play 中指定 tmp 字段触发, 某些功能内部实现也会触发, 如定时播放等),播放完会自动回到之前状态

241 LAN_MUSIC 主机共享 MP3 音源,网络中有主机时可以设置,否则与 MP3 保持一致

245 TALK 对讲音源 ,由对讲协议触发,无法主动设置

254 ALARM 报警音源 ,由外部报警触发,无法主动设置

```
//CALL
```

```
{
```

```

    "call": "player.setSource",
    "arg": {
        "source": 0
    }
}

```

```
//NOTIFY
```

```

{
    "notify": "player.source",
    "arg": {
        "source": 0
    }
}

```

可能触发的通知: player.source, player.info, player.state, player.mode, player.eq, player.volume, player.bass, player.treb, player.time

4.20 播放信息

```
//CALL
```

```

{
    "call": "player.info",
}

```

```
//ACK
```

```
//NOTIFY
```

```

{
    "ack": "player.info",
    "arg": {
        "name": "",
        "source": 0,
        "id": 0, //当前播放歌曲 ID
        "album": "", //可选
        "artist": "", //可选
        "duration": 0, //歌曲总时间
        "playTime": 0, //当前播放时间
        "state": 0, //当前播放状态 0 停止播放, 1 打开中, 2 缓冲, 3 播放, 4 暂停, 5 准备播放,

```

0xfe 错误

```

        "treb": 0, //可选
        "bass": 0, //可选
        "eq": 0, //可选
        "volume": 0, //可选
        "playMode": 0, //可选
        "mute": 0, //可选
        "buffering": 0, //可选, 缓冲进度
    }
}

```

```

        "like":1,//可选, 收藏
        "download":0,//可选, 0 未下载 1 已经下载 2 下载中
        "picUrl":""//专辑图片 URL,可选
    }
}

```

4.21 收藏歌曲

```

//CALL
{
    "call":"player.like",
    "arg" : {
        "id":0,//歌曲 ID,若无,则默认当前歌曲
    }
}

//ACK
{
    "ack":"player.like",
    "arg" : {
        "like":0//已经收藏为 1 ,否则为 0
    }
}

//NOTIFY
{
    "notify":"player.like",
    "arg" : {
        "id":0//收藏的歌曲 id,
        "like":1
    }
}
}
可能触发的通知: player.like

```

4.22 取消收藏

```

//CALL
{
    "call":"player.unlike",
    "arg" : {
        "id":0,//歌曲 ID,若无,则默认当前歌曲
    }
}

```

```
//NOTIFY
{
  "notify": "player.like",
  "arg": {
    "id": 0 // 收藏的歌曲 id,
    "like": 0
  }
}
可能触发的通知: player.like
```

4.23 修改主机名称

```
//CALL
{
  "call": "system.setName",
  "arg": {
    "name": ""
  }
}
```

可能触发的 NOTIFY: system.name

```
//NOTIFY
{
  "notify": "system.name",
  "arg": {
    "name": ""
  }
}
```

4.24 修改通道名称

```
//CALL
{
  "call": "channel.setName",
  "arg": {
    "name": ""
  }
}
```

```
//NOTIFY
{
  "notify": "channel.name",
  "arg": {
```



```
    "name":""  
  }  
}
```

可能触发的 NOTIFY: channel.name

5. 高级音乐 JSON 指令说明

5.1 设置低音

```
//CALL  
{  
  "call": "player.setBass",  
  "arg" : {  
    "bass": 0 //0 ~ 15  
  }  
}  
//NOTIFY  
{  
  "notify": "player.bass",  
  "arg" : {  
    "bass": 0 //0 ~ 15  
  }  
}
```

可能触发的通知: player.bass

5.2 设置高音

```
//CALL  
{  
  "call": "player.setTreb",  
  "arg" : {  
    "treb": 0 //0 ~ 15  
  }  
}  
//NOTIFY  
{  
  "notify": "player.treb",  
  "arg" : {  
    "treb": 0 //0 ~ 15  
  }  
}
```

可能触发的通知: player.treb

5.3 播放下一专辑

```
//CALL
{
    "call": "player.playNextAlbum"
}
```

可能触发的通知: player.info, player.state, player.time, player.list

5.4 播放上一专辑

```
//CALL
{
    "call": "player.playPrevAlbum"
}
```

可能触发的通知: player.info, player.state, player.list

5.5 下载歌曲

```
//CALL
{
    "call": "player.download",
    "arg": {
        "id": 0 //歌曲 ID,若无,则默认当前歌曲
    }
}
```

```
//ACK
{
    "ack": "player.download",
    "arg": {
        "download": 0 //已经下载为 1, 否则为 0
    }
}
```

```
//NOTIFY
{
    "notify": "player.download",
    "arg": {
        "id": 0 //已下载的歌曲 id
        "download": 1
    }
}
```

可能触发的通知: player.download

.5.6 随机播放

随机生成一个指定音源的播放列表并播放

```
//CALL
{
  "call": "player.playRandom",
  "arg": { //可选
    "source": 0
  }
}
```

5.7 专辑列表

两层列表结构（分类结构）

第一层使用 `list.albumList` 获取, 第二层使用 `list.albumNodeList` 获取

```
//CALL
{
  "call": "list.albumList",
  "arg": {
    "source": 0, //可选, 指定音源, 若无, 默认当前音源
    "type": "", //列表类型, album, artist, directory, list, 可选
    "begin": 0, //获取列表的开始位置, 可选
    "size": 4, //获取的数目, 可选
  }
}

//ACK
{
  "ack": "list.albumList",
  "arg": {
    "type": "", //列表类型, album, artist, directory, list, 可选
    "total": 0, //列表总数, 注意不是返回的列表项数目
    "begin": 0, //获取列表的开始位置
    "nodeList": [
      {
        "id": 0, //专辑 ID
        "name": "music", //专辑名称,
        "picUrl": "http://example.com/xx.jpg" //专辑图 URL, 可选
      },
      {
        "id": 0, //专辑 ID
        "name": "mc", //专辑名称,
```

```

        "picUrl":"http://example.com/xx.jpg"//专辑图 URL,可选
    },
]
}
}

```

5.8 专辑内歌曲列表

```

//CALL
{
    "call":"list.albumNodeList",
    "arg": {
        "id":0, //专辑 ID
        "begin":0, //获取列表的开始位置
        "source":0, //可选, 指定音源, 若无, 默认当前音源
        "size":4, //获取的数目
        "metaLevel":0, // 0 自动, 1 最少(仅名字和 id) 2 一般(名字,id,歌手,专辑,长度) 3, 全部
        (会包含 url 等), 可选, 默认 0
    }
}

```

```

//ACK
{
    "ack":"list.albumNodeList",
    "arg": {
        "id":0, //专辑 ID
        "total":0, //列表总数, 注意不是返回的列表项数目
        "begin":0, //获取列表的开始位置
        "nodeList":[
            {
                "id":0, //歌曲 ID
                "name":"music", //歌曲名称
                "artist":"zjl", //歌手名称 //可选
                "album":"al", //专辑名称 //可选
                "duration":82, //播放长度 单位秒 //可选
                "dummy":false //是否可以播放, //可选
            },
            {
                "id":0, //歌曲 ID
                "name":"music", //歌曲名称
                "artist":"zjl", //歌手名称 //可选
                "album":"al", //专辑名称 //可选
            }
        ]
    }
}

```

```

        "duration":82,//播放长度 单位秒 //可选
        "dummy":false//是否可以播放, //可选
    },
]
}
}

```

5.9 party 操作

```

//party 信息
//CALL
{
    "call":"party.info"
}
//NOTIFY
{
    "call":"party.info",
    "arg":{
        "state":0//0 未开启 party, 1 未加入 party, 2 party 中
    }
}
}

```

//操作 party

party 操作说明

1. 打开任意通道, 向通道发送 party.set, action 指定为 3, 创建 party(初始播放歌曲, 歌曲列表以该通道为准), 若网络内已经有一个 party, 则默认加入, 否则创建 party, 默认并将局域网内所有已打开的通道拉入 party, 若指定了 channelList 则拉入指定的通道. 当 party 里的通道都退出以后, party 会自动关闭. **发起 party 的通道关闭或者退出, 若还有其他通道在 party 内, party 是不会解散的.**

2. 已加入 party 的通道可以随时发送 party.set, action 指定 0, 退出 party

3. 局域网内后来打开的通道可以发送 party.set, action 指定 1 加入 party

4. 向局域网内任一通道发送 party.set, action 指定 2 都可以解散 party

```

//CALL
{
    "call":"party.set",
    "arg":{
        "action":0//0 退出 party, 1 加入 party, 2 解散 party, 3 若无则创建, 否则加入
        "channelList":["//需要加入的通道列表, 仅在创建时有效. 通道 ID 可以有两种格式
            "channel.xxxxxx.0", //使用通道唯一 ID 标识, 通道唯一 ID 格式为: channel.[主机唯一
            id].[通道 id]
            "192.168.2.1#0"//[主机 IP]#[通道号]
        ]
    }
}

```

```

    ]
  }
}
可能触发的 NOTIFY: party.info,player.*

```

5.10 对讲操作

对讲可以 1 对 1 或 1 对所有

```

//CALL
{
  "call": "talk.set",
  "arg": {
    "talkTag": "",
    "action": 0 // 0 关闭 talk, 1 打开对讲, 2 开始说话
    // action 为 1 时可以有下面字段
    "url": "" // 音频流, 支持 rtsp:// http:// udp://
    "udpInfo": {} // 可选
    "codec": "" // s16L, alaw, ulaw, mp4a 等
    "channel": 0 // 1, 2
    "samplerate": 0 // 8000, 16000, 44100...
  }
  "peerIp": "" // 可选
  "peerId": "" // 可选
  "broadcast": false // 可选, 对所有人说话
}
}
//ACK
{
  "ack": "talk.set",
  "tag": "001",
  "arg": {
    // udp:// 才包含下面字段
    "dataPort": 0 // udp 数据接收端口
  }
}
}

```

可能触发的 NOTIFY: talk.info,player.*

```

//对讲信息
//CALL
{
  "call": "talk.info"
}

```

```
//NOTIFY
{
  "call": "talk.info",
  "arg": {
    "state": 0//0 未打开, 1 打开但未说话, 2 说话中
    "peerIp": "",//可选
    "peerId": "",//可选
    "broadcast": false//可选, 对所有人说话
  }
}
```

.5.11 通道配置

设置通道配置:

channel.setConfig

```
//CALL
{
  "call": "channel.setConfig",
  "arg": {
    "ringEnable": true,//可选 是否响应门铃
    "ringVolume": 0,//可选 门铃的铃声音量
    "ringMediaUrl": "/sda1/ring.wav",//可选 门铃的铃声
    "party": false//可选 内否能被拉入 PARTY
    "talk": false//可选 内否能被拉入对讲
    "airplay": false//可选 是否启用 airplay
    "dlna": false//可选 是否启用 dlna
  }
}
```

获取通道配置

```
//CALL
{
  "call": "channel.configInfo"
}
//ACK
{
  "ack": "channel.configInfo"
  "arg": {
    "ringEnable": true,//可选 是否响应门铃
    "ringMediaUrl": "/sda1/ring.wav",//可选 门铃的铃声
    "ringVolume": 80//可选 门铃铃声音量
  }
}
```

```

}

//NOTIFY
{
    "notify": "channel.configInfo"
    "arg":{
        "ringEnable": true //改变的 KEY 和新值
    }
}

```

5.12 日志上传

```

//CALL
{
    "call": "upload.log"
}

```

5.13 软件更新

系统支持上层软件更新和 ROM 更新两种.一般只需要上层软件更新即可.

```

//查询更新
//CALL
{
    "call": "update.info"
}
//ACK
{
    "ack": "update.info",
    "arg": {
        "type": 0, //更新的类型 0 APP, 1 ROM 可选,默认 APP
        "state": 0, //是否有更新, 0,已是最新,1,已经更新完成(需重启才能使用最新),2,正在检测和更新, 3 检测到新版, 但是未更新
        "version": "", //当前系统版本
        "updateVersion": 0, //可选,将要更新的版本
    }
}
//NOTIFY
{
    "notify": "update.info",
    "arg": {
        "version": 0, //当前运行程序的版本
        "type": 0, //更新的类型 0 APP, 1 ROM 可选,默认 APP

```



```
    "state":0,//是否有更新, 0,已是最新,1,已经更新完成 2,正在检测和更新, 3 检测到新版,
但是未更新
    "updateVersion":0,//可选,若有更新,则为最新版本信息
  }
}
```

//触发更新

更新流程:

- 1.发送 update.set, action 指定 UPDATE
- 2.设备开始进入更新,发送 notify: update.info, state 为 2
- 3.设备更新完成,发送 notify: update.info, state 为 0 或 1

```
//CALL
{
  "call": "update.set",
  "arg": {
    "type": 0, //更新的类型 0 APP, 1 ROM 可选,默认 APP
    "action": "UPDATE" //CHECK 检测更新, UPDATE 检测并更新
  }
}
可能触发的 NOTIFY: update.info
```

5.14 系统操作

```
//CALL
{
  "call": "system.set",
  "arg": {
    "action": "", //FACTORY 回复出厂, REBOOT 重启, REMOTE_CONTROL_RESET 远程
控制重置, WIFI_RESET 清除 WIFI 配置, DISABLE_YD_NOTIFY 禁止非 JSON 的通知
DISABLE_JSON_NOTIFY 禁止 JSON 的通知
  }
}
```

5.15 修改主机时间

时间虽然可以修改,但是接入外网后会自动网络校准

```
//CALL
{
  "call": "system.setDateTime",
  "arg": {
    "date": "yyy-MM-dd", //可选
  }
}
```

```
        "time": "hh:mm:ss", // 可选
    }
}
可能触发的 NOTIFY: system.dateTime
```

5.16 获取主机时间

```
//CALL
{
    "call": "system.dateTime",
}
//ACK
{
    "ack": "system.dateTime",
    "arg": {
        "date": "yyy-MM-dd",
        "time": "hh:mm:ss"
    }
}
```

5.17 触发或取消警报

火警无法取消，只能关闭机器，盗警只要关闭通道或切换音源即可取消

```
//CALL
{
    "call": "alarm.set",
    "arg": {
        "type": 0, // 报警类型 0 火警 1 盗警
        "state": 0 // 报警状态 0 关闭, 1 开启
    }
}
```

5.18 定时器

定时器有三种类型,定时开机,定时关机,定时闹铃. 定时器可以在指定日期时间触发或者周期触发(星期一到星期天). 定时开机和闹铃可以指定音源和播放的歌曲. 每一个定时器都有一个ID来唯一标识(这个ID是添加时,系统分配的)

```
//添加定时器
//CALL
//简单参数
//默认每天指定时间触发定时操作
{
    "call": "timer.add",
```

```

    "arg": {
        "type": 0, //闹铃类型, 0,闹铃, 1 开机, 2 关机, 3 HEX 指令, 4 AI 指令, 5 语音
        播报, 6 搜索播放, 7 重启系统
        "time": "hh:mm:ss", //闹铃启动时间
        "enable": false, //是否开启
    }
}

```

//完整参数

```

{
    "call": "timer.add",
    "arg": {
        "name": "music", //名称, _SYSTEM_DELAY_DELAY_CLOSE 延时关机特定名称,
        _SYSTEM_DELAY_OPEN_TIMER 定时开机, _SYSTEM_DELAY_CLOSE_TIMER 定时关机
        "circleDay": 0, //循环周期, 类似 iphone, 最低位表示星期一,次高位表示星期天是
        否触发,如每周星期一,星期三触发的值为:0x05, 每周星期二,星期天触发则是:0x42
        "type": 0, //闹铃类型, 0,闹铃, 1 开机, 2 关机, 3 HEX 指令, 4 AI 指令, 5 语音
        播报, 6 搜索播放, 7 重启系统
        "date": "yyyy-mm-dd", //闹铃启动日期,若指定了 circleDay 则会忽略该参数
        "time": "hh:mm:ss", //闹铃启动时间
        "enable": false, //是否开启
        "runTime": 0, //闹铃时间,仅 type 为闹铃才有效, 可选
        "volume": 0, //闹铃音量,可选
        "source": 0, //音源, 可选
        "aiText": "" //ai 指令文本
        "ttsText": "" //语音播放文本
        "hexCmd": "" //指令 HEX 文本, 高级功能
        "media": { //歌曲, 可选
            "id": 0, //id 和 url 只需要指定一个即可
            "url": ""
        }
        "album": { //专辑, 可选
            "id": 0, //id 和 url 只需要指定一个即可
            "url": ""
        }
    }
}

```

可能触发的通知: timer.list

//设置定时器

//CALL

//简单参数

```

{
    "call": "timer.set",

```

```

    "arg": {
        "id": 0, //ID, //指定需要设置的定时器
        "time": "hh:mm:ss", //闹铃启动时间, 可选
        "enable": false, //是否开启, 可选
    }
}
//完整参数
{
    "call": "timer.set",
    "arg": {
        "id": 0, //指定需要设置的定时器
        "name": "music", //名称, _SYSTEM_DELAY_CLOSE 延时关机特定名称
        "circleDay": 0, //循环周期, 类似 iphone, 可选
        "type": 0, //闹铃类型, 0, 闹铃, 1 开机, 2 关机, 3 HEX 指令, 4 AI 指令, 5 语音
        播报, 6 搜索播放, 7 重启系统
        "date": "yyyy-mm-dd", //闹铃启动日期, 可选
        "time": "hh:mm:ss", //闹铃启动时间, 可选
        "enable": false, //是否开启, 可选
        "runTime": 0, //闹铃时间, 可选
        "volume": 0, //闹铃音量, 可选
        "source": 0, //音源, 可选
        "media": { //歌曲, 可选
            "id": 0, //id 和 url 只需要指定一个即可
            "url": ""
        }
        "album": { //专辑, 可选
            "id": 0, //id 和 url 只需要指定一个即可
            "url": ""
        }
        "aiText": "" //ai 指令文本
        "ttsText": "" //语音播放文本
        "hexCmd": "" //指令 HEX 文本, 高级功能
    }
}

//删除定时器
//CALL
{
    "call": "timer.del",
    "arg": {
        "id": 0, //ID 可选
    }
}
可能触发的通知: timer.list

```

```

//定时器列表
//CALL
{
    "call": "timer.list",
    "arg": {
        "begin": 0, //获取列表的开始位置
        "size": 4 //获取的数目
    }
}

//ACK
{
    "ack": "timer.list",
    "arg": [
        "begin": 0, //获取列表的开始位置
        "total": 0, //列表总数,注意不是返回的列表项数目
        "nodeList": [
            {
                "id": 0, //iD
                "name": "music", //名称
                "circleDay": 0, //循环周期, 类似 iphone
                "type": 0, //闹铃类型, 0,闹铃, 1 开机, 2 关机, 3 HEX 指令, 4 AI 指令, 5 语音播
报, 6 搜索播放, 7 重启系统
                "date": "yyyy-mm-dd", //闹铃启动日期
                "time": "hh:mm:ss", //闹铃启动时间
                "enable": false, //是否开启
                "runTime": 0, //闹铃时间
                "volume": 0, //闹铃音量
                "source": 0, //音源
                "aiText": "", //ai 指令文本
                "ttsText": "", //语音播放文本
                "hexCmd": "", //指令 HEX 文本, 高级功能
                "media": { //歌曲
                    "name": ""
                }
                "album": { //专辑
                    "name": ""
                }
            }
        ]
    }
}

//NOTIFY
{

```

```
"ack": "timer.list"
}
```

5.19 延迟关机定时器

延时关机定时器只有一个,它通过一个特殊的名称来标识. 如果设置了,那么当到达指定的时间后,通道会自动关闭.

添加延时关机定时器

```
{
  "call": "timer.add",
  "arg": {
    "name": "_SYSTEM_DELAY_TIMER", //名称, _SYSTEM_DELAY_TIMER 延时关机特定名称
    "delayTime": 0 //延迟的时间,秒
  }
}
```

获取延时关机定时器

//CALL

```
{
  "call": "timer.info",
  "arg": {
    "name": "_SYSTEM_DELAY_TIMER", //名称, _SYSTEM_DELAY_TIMER 延时关机特定名称
  }
}
```

//ACK

```
{
  "ack": "timer.info",
  "arg": {
    "name": "_SYSTEM_DELAY_TIMER", //名称, _SYSTEM_DELAY_TIMER 延时关机特定名称
    "delayTime": 0, //; 离下次触发时间秒
  }
}
```

删除延时关机定时器

//CALL

```
{
  "call": "timer.del",
  "arg": {
    "name": "_SYSTEM_DELAY_TIMER"
  }
}
```

5.20 FM 操作

自动搜台

```
//CALL
{
    "call": "radio.autoScan"
}
//NOTIFY
{
    "notify": "radio.autoScan",
    "arg": {
        "state": 0 // 2 正在搜台, 0 搜台完成
    }
}
```

可能触发的通知: radio. autoScan

添加频率

```
//CALL
{
    "call": "radio.addFreq",
    "arg": {
        "name": "", // 可选
        "freq": 0 // 如 101.7Mhz 为 1017
    }
}
```

删除频率

```
//CALL
{
    "call": "radio.delFreq",
    "arg": {
        "id": 0
    }
}
```

可能触发的通知: list. MediaList

获取列表

//使用获取歌曲列表接口即可

指定播放频率

可以直接使用 player.play 命令, 默认歌名类似 91.8Mhz

```
//CALL
{
```

```

    "call": "player.play",
    "arg": {
        "url": "918" //91.8Mhz
    }
}

```

5.21 WIFI 配网

1. 确保机器进入配网模式（机器启动完成，并且拔掉有线）
2. 手机接入配网设备的 ap(yodar-xxxxxx)
3. 发送设备搜索命令(ce 00 ce)，搜索设备，获取设备 ID
4. 发送 wifi.apList 获取 ap 列表
5. 发送 wifi.set 配置 ssid,password 等 wifi 参数
6. 切换到配置的路由器 AP 下(一般手机会自动切回)，发送(ce 00 ce)搜索设备，直到发现刚才配置设备(通过设备 ID 确认)

```

//获取无线 AP 列表
//CALL
{
    "call": "wifi.apList",
    "arg": {
        "begin": 0, //获取列表的开始位置
        "size": 4 //获取的数目
    }
}

//ACK
{
    "ack": "wifi.apList",
    "arg": {
        "total": 0, //APli 列表总数,注意不是返回的列表项数目
        "begin": 0, //获取列表的开始位置
        "apList": [
            {
                "ssid": "WRT", //AP 名称
                "bssid": "E0:46:9A:34:4F:F2", //MAC 地址
                "channel": 2, //WIFI 信道
                "quality": 0, //0~100 表示无线信号质量
                "signal": -82, // 无线信号强度 单位 dbm
                "encryption": "psk" //加密类型,可以是 psk,psk2,wep,none
                "connect": true //是否已经连接成功, 可选, 默认 false
            },
            {

```



```

        "ssid":"Yodar",
        "bssid":"88:25:93:5C:E1:70",
        "channel":11,
        "quality":0,
        "signal":-82,
        "encryption":"psk" //psk,psk2,wep,none
    }
]
}
}

```

//获取当前 WIFI 信息

//CALL

```

{
    "call":"wifi.info"
}

```

//ACK

```

{
    "ack":"wifi.info",
    "arg": {
        "network":false,
        "internet":false,
        "lan": {
            "ssid":"Yodar", //AP 名称
            "channel":11,//0-11 无线信道
            "encryption":"psk2", //加密类型, psk,psk2,wep,none
            "key":"yodarwifi",//WIFI 密码
            "hidden":0,//是否隐藏 AP
            "enable":1//是否开启
            "ip":"192.168.186.1",
            "mac":"00:11:22:33:33:44"
        },
        "wan": { //若无,则无该配置
            "ssid":"WRT", //想要无线连接的上层路由 AP 名称,通过 wifi.apList 命令可以获取
            "encryption":"psk2", //想要无线连接的上层路由加密类型
            "key":"password", //想要无线连接的上层路由的密码
            "enable":1//开启连接上层路由
            "connect":true//是否已经连接成功, 可选, 默认 false
            "ip":"192.168.2.13",
            "mac":"00:11:22:33:33:46"
            "quality":0//0-100 信号强度
        }
    }
}

```

```

}
//设置 WIFI
//CALL
{
  "call": "wifi.set",
  "arg": {
    "lan": { //若有，则配置 AP 已经禁用
      "ssid": "Yodar", //可选
      "encryption": "psk2", //psk,psk2,wep,none//可选
      "key": "yodarwifi", //可选
    },
    "wan": { //若有，则配置 WIFI 无线到其他路由
      "ssid": "WRT",
      "encryption": "psk2", //psk,psk2 可选
      "key": "password", //可选
      "disabled": 0 //可选
    }
  }
}
}

```

5.22 获取网络信息

```

//CALL
{
  "call": "network.info"
}
//ACK
{
  "ack": "network.info",
  "arg": {
    "network": true,
    "internet": true,
    "gateway": "192.168.2.1",
    "dns1": "192.168.2.1",
    "dns2": "192.168.2.1",
    "protocol": "DHCP"
    "ip": "192.168.2.3",
    "netmask": "255.255.255.0",
    "mac": "8C:88:2B:00:02:7E "
  }
}
}

```

5.23 系统配置

//修改系统配置

//CALL

```
{
  "call": "config.set",
  "arg": {
    "ledDelayTime": 0 //可选 led 自动关闭时间, <=0 时表示常亮 ,单位 秒,仅 T3/W5 支持
    "remoteControl": false //可选 远程控制开关
    "alarm": false //可选 报警开关
    "onlyLocalHost": false //可选 是否单主机, 若为 true 则不支持多台设备一起 party,同时可
提升 party 稳定性和网络负载
    "airplayParty": false //可选 ,airplay 的 PARTY 推送开关
    "airplayPartyPassword": false //可选
    "dlnaParty": false //可选, dlna 的 PARTY 推送开关
  }
}
```

//获取系统配置信息

//CALL

```
{
  "call": "config.info"
}
```

//ACK

```
{
  "call": "config.info",
  "arg": {
    "ledDelayTime": 0,
    "remoteControl": false,
    "alarm": false,
    "remoteControlLock": false,
    "onlyLocalHost": false,
    "airplayParty": false,
    "airplayPartyPassword": false,
    "dlnaParty": false
  }
}
```

5.24 本地音乐文件操作列表

直接操作文件系统中的文件和文件夹

//本地音乐文件操作列表

//直接获取系统中文件夹中的内容,会过滤掉非音乐文件,仅能获取本地音乐.

//CALL

```
{
  "call": "system.dirNodeList",
  "arg": {
    "path": "", 文件夹路径,根路径为"/"
    "type": 0//0 文件,1 文件夹,2 全部,默认为 2
    "begin": 0, //获取列表的开始位置
    "size": 4//获取的数目
  }
}
```

//ACK

```
{
  "ack": "system.dirNodeList",
  "arg": {
    "path": "", //文件夹 path
    "total": 0, //列表总数,注意不是返回的列表项数目
    "begin": 0, //获取列表的开始位置
    "nodeList": [
      {
        "name": "download",
        "type": 1//0 文件,1 文件夹,默认为 0
      },
      {
        "name": "music",
        "type": 1//0 文件,1 文件夹,默认为 0
      },
    ]
  }
}
```

//创建文件或文件夹

//CALL

```
{
  "call": "system.createFile",
  "arg": {
    "path": ""//文件或文件夹路径,若路径以"/"结尾表示创建文件夹,否则为文件
  }
}
```

//重命名文件或文件夹

//CALL

```

{
  "call": "system.renameFile",
  "arg": {
    "path": "", // 路径
    "name": "" // 新名称
  }
}

```

// 删除文件或文件夹

// CALL

```

{
  "call": "system.removeFile",
  "arg": {
    "path": "", // 路径
  }
}

```

// 移动文件或文件夹

// CALL

```

{
  "call": "system.moveFile",
  "arg": {
    "path": "", // 路径
    "dstDir": "", // 目的文件夹
  }
}

```

5.25 歌单操作

歌单是用户字节编辑的歌曲列表，属于两层列表结构，歌单列表使用 `list.albumList` 获取(type 指定 list)，歌单内容使用 `list.albumNodeList` 获取

// 创建歌单

// CALL

```

{
  "call": "list.createList",
  "arg": {
    "name": ""
  }
}

```

// 删除歌单

// CALL

```

{

```

```
    "call": "list.removeList",
    "arg": {
        "id": 0
    }
}
```

//重命名歌单

```
//CALL
{
    "call": "list.renameList",
    "arg": {
        "id": 0,
        "name": "",
    }
}
```

//添加歌曲到歌单

```
//CALL
{
    "call": "list.addListNode",
    "arg": {
        "id": 0,
        "mediaId": 0
    }
}
```

//从歌单删除歌曲

```
//CALL
{
    "call": "list.delListNode",
    "arg": {
        "id": 0,
        "mediaId": 0
    }
}
```

5.26 音乐场景

音乐设备可以通过屏幕或者手机端编辑 1~16 个快捷场景（操作），它可以是打开、关闭音乐设备，指定播放某首歌，播放，暂停，调节音量等乐设备，执行语音播放，AI 指令，指定播放某首歌，播放，暂停，调节音量等。

//执行音乐场景

```

//CALL
{
  "call": "scene.run",
  "arg": {
    "id": 0, //场景 ID, 0~8
    "pos": 0, //可选, 场景索引 (第几个场景), 0~8, 若 ID 同时存在, 优先使用 ID
  }
}

//添加场景
{
  "call": "scene.add",
  "arg": {
    "name": "music", //名称
    "type": 0, //场景类型, 0, 播放, 1 开机, 2 关机, 3 HEX 指令, 4 AI 指令, 5 语音
    播报, 6 搜索播放
    "volume": 0, //音量, 可选
    "source": 0, //音源, 可选
    "icon": "", //图标, 可选
    "aiText": "", //ai 指令文本
    "ttsText": "", //语音播放文本
    "searchKey": "", //搜索关键字
    "media": { //歌曲, 可选
      "id": 0, //id 和 url 只需要指定一个即可
      "url": ""
    }
    "album": { //专辑, 可选
      "id": 0, //id 和 url 只需要指定一个即可
      "url": ""
    }
  }
}

```

可能触发的通知: scene.list

//设置场景

//CALL

//完整参数

```

{
  "call": "scene.set",
  "arg": {
    "id": 0, //指定需要设置的场景
    "name": "music", //名称
    "type": 0, //场景类型, 0, 播放, 1 开机, 2 关机, 3 HEX 指令, 4 AI 指令, 5 语音
    播报, 6 搜索播放
  }
}

```

```

        "volume":0//音量,可选
        "source":0//音源, 可选
        "icon": ""//图标, 可选
        "aiText": ""//ai 指令文本
        "ttsText": ""//语音播放文本
        "searchKey": ""//搜索关键字
        "media": { //歌曲, 可选
            "id":0, //id 和 url 只需要指定一个即可
            "url": ""
        }
        "album": { //专辑, 可选
            "id":0, //id 和 url 只需要指定一个即可
            "url": ""
        }
    }
}

```

//删除场景

//CALL

```

{
    "call": "scene.del",
    "arg": {
        "id":0, //iD 可选
    }
}

```

可能触发的通知: scene.list

//场景列表

//CALL

```

{
    "call": "scene.list",
    "arg": {
        "begin":0, //获取列表的开始位置
        "size":4//获取的数目
    }
}

```

//ACK

```

{
    "ack": "scene.list",
    "arg": [
        "begin":0, //获取列表的开始位置
        "total":0, //列表总数,注意不是返回的列表项数目
        "nodeList": [

```



```

        {
            "id":0,//iD
            "name":"music",//名称
            "type":0, //场景类型, 0,播放, 1 开机, 2 关机, 3 HEX 指令, 4 AI 指令, 5 语音
播报, 6 搜索播放
            "volume":0//音量
            "source":0//音源
            "icon":"","//图标
            "aiText": ""//ai 指令文本
            "ttsText":"","//语音播放文本
            "searchKey":"","//搜索关键字
            "media": { //歌曲
                "name":""
            }
            "album": { //专辑
                "name":""
            }
        }
    ]
}
//NOTIFY
{
    "ack":"scene.list"
}

```

5.27 云音乐 VIP

最新的云音乐内容平台为：QQ 音乐， 内容提供方要求用户使用微信扫码登录授权， 并且 VIP 歌曲需要开通 VIP 会员才能完整收听。

system.info 命令中新增：

cloudType: 4 表示 qqmusic

isMusicVip: false 是否 VIP， 注意： 这里的 VIP 不是平台 VIP

//查询云音乐 VIP 信息

//CALL

```

{
    "call":"cloud.music.vipInfo"
}

```

//ACK

```

{
    "ack":"cloud.music.vipInfo",
}

```

```

    "arg": {
      "plat": "", //音乐平台 qqmusic
      "user": "", //用户名称
      "userLogin": 0, //登录状态 -1 未知, 0 未登录, 1 登录
      "isVip": false, //VIP 会员, 也叫平台 VIP
      "vipExpires": "1970-01-01 00:00:00", //平台 VIP 过期时间,
      "isYodarVip": false,
      "yodarVipExpires": "1970-01-01 00:00:00", //悠达会员, 也叫普通会员
    }
  }
}

//用户登录云音乐
//CALL
{
  "call": "cloud.music.login"
}
//ACK
{
  "ack": "cloud.music.login ",
  "arg": {
    "qrCodeUrl": "" //二维码 Url, 需要再 UI 上渲染出来, 让客户扫码登录
    "expireSeconds": 300 //有效时间, 单位: 秒
  }
}
}

```

//查询用户登录状态

//调用登录接口后, 需要轮询登录状态, 一般建议 2~5 秒间隔一次, 登录超时根据二维码有效时间决定, 登录成功后, 主机的 VIP 信息更新可能有几秒钟延时

//注意: 该接口尽在调用 login 后半小时内有效, 其他时候请使用 cloud.music.vipInfo 查询登录状态

```

//CALL
{
  "call": "cloud.music.loginState",
}
//ACK
{
  "ack": "cloud.music.loginState",
  "arg": {
    "state": 0 // 0 未登录, 1 登录 2 登录中
  }
}
}

```

```

//用户退出云音乐
//CALL
{
    "call":"cloud.music.logout "
}
//ACK
{
    "ack":"cloud.music.logout "
}

//VIP 套餐列表
//CALL
{
    "call":"cloud.music.payVipList"
}
//ACK
{
    "ack":"cloud.music. payVipList ",
    "arg": {
        "nodeList": [
            {
                "id":0,//套餐 id
                "title":"","//套餐名
                "detail":"","//套餐说明
                "price":0,//套餐价格,以分为单位
                "origPrice": 0,//套餐原价,以分为单位
                "duration":0,//套餐时长，以月为单位
            },
        ]
    }
}

//购买 VIP 套餐
//CALL
{
    "call":"cloud.music.payVip",
    "arg":{
        "id":0,//套餐 id
        "price":0,//VIP 套餐价格,以分为单位
        "title":"","//VIP 套餐名称
    }
}
//ACK
{

```

```

    "ack": "cloud.music.payVip ",
    "arg": {
        "orderId": "" //订单号
        "payUrl": "" //二维码 Url, 需要再 UI 上渲染出来, 让客户使用微信扫码支付
        "expireSeconds": 600 //有效时间, 单位: 秒
    }
}

```

//查询支付状态

//CALL

```

{
    "call": "cloud.music.payState",
    "arg": {
        "orderId": "" //订单号
    }
}

```

//ACK

```

{
    "ack": "cloud.music.payState ",
    "arg": {
        "orderId": "" //订单号
        "state": 0 // 1 支付成功, 2 已关闭订单, 4 支付中, 6 已取消支付,
    }
}

```

//刷新设备 VIP 信息

//仅用在当用户支付完成后, 让主机尽快同步 VIP 信息(否则可能需要 1-2 分钟才会同步), 并更新歌曲列表数据

//CALL

```

{
    "call": "cloud.music.refresh"
}

```

//ACK

```

{
    "ack": "cloud.music.refresh"
}

```

6. 门铃 JSON 命令说明

6.1 门铃信息

//CALL

```

{

```

```

    "from":"","//设备唯一 ID
    "call":"camera.info"
}

//ACK OR NOTIFY
{
    "from":"","//设备唯一 ID
    "ack":"camera.info",
    "arg":{
        "talk":"","//对讲者的 ID
        "talkTag":"","//对讲 TAG
        "video":false,//摄像头采集
        "audio":false,//MIC 采集
        "speaker":false//扬声器播放
        "speakerVoice":false//扬声器是否有声音
        "remote":false//是否有远程查看
        "sensorMode":0//摄像头模式 0 白天 1 黑夜
        "name":"","//名称
    }
}

```

6.2 配置门铃

```

//CALL
{
    "from":"","//设备唯一 ID
    "call":"camera.set",
    "arg":{
        "sensorMode":0//摄像头模式, 0 白天 1 黑夜
    }
}

```

6.3 保持状态

当查看摄像头或者向命令发送语音数据时,需要周期发送该指令保持(10s 一次,30 秒超时),否则 30 秒后无法查看视频或关闭喇叭.

```

{
    "from":"","//设备唯一 ID
    "call":"camera.keep",
    "arg":{
        "video":false//摄像头采集
        "audio":false//MIC 采集
    }
}

```

```
    "speaker":false//扬声器播放
  }
}
```

6.4 查看视频

```
//CALL
{
  "from":"","//设备唯一 ID
  "call":"camera.view",
  "arg":{
    "video":false//摄像头采集
    "audio":false//MIC 采集
    "remote":false//是否远程查看，可选，默认系统自动识别
  }
}
//ACK
{
  "ack":"camera.view",
  "arg":{
    "url":"","//可播放的 rtsp://流地址
    "resolution":"","//视频分辨率 D1 720x576 720P 1280x720 1080P 1920x1080
  }
}
```

6.5 向门铃说话

```
//CALL
{
  "from":"","//设备唯一 ID
  "call":"camera.talk",
  "arg":{
    "talkTag":"","
    "action":0//0 结束讲话, 1 开始说话
    //action 为 1 时可以有下面字段
    "url":"","//音视频流, 支持 rtsp:// http:// udp://
  }
}
//ACK
{
  "ack":"camera.talk"
}
```

6.6 开门

```
//CALL
{
  "from":"","//设备唯一 ID
  "call":"door.open"
}
```

6.7 门铃触发

当有人按门铃时, 会触发该通知

```
//NOTIFY
{
  "from":"","//设备唯一 ID
  "call":"camera.ring"
}
```

7. JSON 样例指令

```
//搜索主机 反馈 0xef
ce 00 ce
```

```
//保持心跳 反馈 0xcf
cf 00 cf
```

```
//系统信息
{"call":"system.info"}
```

```
//通道信息
{ "call": "channel.info"}
```

```
//修改通道名称, 会触发 channel.name 通知
{ "call": "channel.setName", "arg":{"name": "xxx"}}
```

```
//打开通道, 会触发 channel.state 或 player.info 通知
{ "call": "channel.open"}
{ "call": "channel.open", "arg":{"autoPlay":false}}
{ "call": "channel.open", "arg":{"autoPlay":true, "source":5}}
```

```
//关闭通道,会触发 channel.state 通知
{ "call": "channel.close"}
```

```
//获取云音乐根目录下内容
```

```
{ "arg": { "id": 0, "begin": 0, "size": 8, "type": 2, "source": 5 }, "call": "list.dirNodeList" }
```

//获取本地音乐根目录下内容

```
{ "arg": { "id": 0, "begin": 0, "size": 8, "type": 2, "source": 0 }, "call": "list.dirNodeList" }
```

//播放歌曲,会触发 player.info 通知

```
{ "arg": { "id": 1 }, "call": "player.play" }
```

//自动打开通道并播放本地歌曲 会触发 player.info 或 channel.state 通知

```
{ "arg": { "id": 1, "autoOpen": true, "source": 0 }, "call": "player.play" }
```

//自动打开通道,按指定音量播放歌曲,播放完后恢复到原有状态 会触发 player.info 或 channel.state 通知

```
{ "arg": { "url": "/xx/xx/aa.mp3", "tmp": true, "source": 0, "volume": 120 }, "call": "player.play" }
```

//获取当前播放信息

```
{ "call": "player.info" }
```

//获取当前播放列表

```
{ "arg": { "name": "play", "begin": 0, "size": 8 }, "call": "list.mediaList" }
```

//获取收藏列表

```
{ "arg": { "name": "like", "begin": 0, "size": 8 }, "call": "list.mediaList" }
```

//下一首, 会触发 player.info 通知

```
{ "call": "player.playNext" }
```

//上一首, 会触发 player.info 通知

```
{ "call": "player.playPrev" }
```

//暂停 会触发 player.state 或 player.info 通知

```
{ "call": "player.pause" }
```

//继续播放 会触发 player.state 或 player.info 通知

```
{ "call": "player.resume" }
```

//停止播放 会触发 player.state 或 player.info 通知

```
{ "call": "player.stop" }
```

//切换到本地音源, 会触发 player.source 或 player.info 通知

```
{ "call": "player.setSource", "arg": { "source": 0 } }
```

//切换到云音乐, 会触发 player.source 或 player.info 通知

```
{ "call": "player.setSource", "arg": { "source": 5 } }
```


//设置音量 0~255 会触发 player.volume 或 player.info 通知
{ "call": "player.setVolume", "arg": {"volume": 100}}

//EQ 会触发 player.eq 或 player.info 通知
{ "call": "player.setEq", "arg": {"eq": 0}}

//高音 会触发 player.treb 或 player.info 通知
{ "call": "player.setTreb", "arg": {"treb": 7}}

//低音 会触发 player.bass 或 player.info 通知
{ "call": "player.setBass", "arg": {"bass": 7}}

//静音 会触发 player.mute 或 player.info 通知
{ "call": "player.setMute", "arg": {"mute": true}}

//取消静音 会触发 player.mute 或 player.info 通知
{ "call": "player.setMute", "arg": {"mute": false}}

//指定时间播放 会触发 player.playTime 或 player.info 通知
{ "call": "player.seek", "arg": {"time": 30}}

//收藏 会触发 player.like 或 player.info 通知
{ "call": "player.like" }

//取消收藏 会触发 player.like 或 player.info 通知
{ "call": "player.unlike" }

//播放模式 会触发 player.mode 或 player.info 通知
{ "call": "player.setPlayMode", "arg": {"mode": 0}}

//下载
{ "call": "player.download" }

//获取歌曲信息
{ "call": "media.info", "arg": {"id": 1}}

8. 老协议样例指令

- 1.控制需要先发 ce 00 ce, 设备会记录控制端的 IP 和 PORT
 - 2.10s 一次发送 cf 00 cf, 否则 30s 会无法控制。
 - 3.其他控制端操作设备时, 会通过 ce 命令时记录的 IP 和 PORT 发送通知数据
- 下面所列出来的简单指令可以在所有网络音乐设备上通用, 需要更高级的命令请参考老的网

络协议文档.

注意:

下面的命令仅能通过网络 UDP 控制, 主机监听端口为: 10061

搜索设备 ce 00 ce

心跳保持 cf 00 cf

通道 0

打开 a300070106

关闭 a30003fffc

下一曲 a300090009

上一曲 a300050005

暂停 a300020103

播放 a300020002

静音 ab0000ffff

取消静音 ab0001fffe

音量++ a300010001

音量-- a300080008

MP3 音源 a3000b000b

SD 卡音源(ANDROID 机型) a3000d020f

云音乐音源 a3000f000f

AUX1 音源 a300060006

AUX2 音源 a3000c000c

FM 音源 a3000a000a

FM 搜台 ba00000000

通道 1

打开 a301070107

关闭 a30103fffd

下一曲 a301090008

上一曲 a301050004

暂停 a301020102

播放 a301020003

静音 ab0100fffe

取消静音 ab0101ffff

音量++ a301010000

音量-- a301080009

MP3 音源 a3010b000a

SD 卡音源(ANDROID 机型) a3010d020e

云音乐音源 a3010f000e

AUX1 音源 a301060007

AUX2 音源 a3010c000d

FM 音源 a3010a000b

FM 搜台 ba01000001

通道 2

打开 a302070104

关闭 a30203fffe
下一曲 a30209000b
上一曲 a302050007
暂停 a302020101
播放 a302020000
静音 ab0200fffd
取消静音 ab0201fffc
音量++ a302010003
音量-- a30208000a
MP3 音源 a3020b0009
SD 卡音源(ANDROID 机型) a3020d020d
云音乐音源 a3020f000d
AUX1 音源 a302060004
AUX2 音源 a3020c000e
FM 音源 a3020a0008
FM 搜台 ba02000002
通道 3
打开 a303070105
关闭 a30303ffff
下一曲 a30309000a
上一曲 a303050006
暂停 a303020100
播放 a303020001
静音 ab0300fffc
取消静音 ab0301fffd
音量++ a303010002
音量-- a30308000b
MP3 音源 a3030b0008
SD 卡音源(ANDROID 机型) a3030d020c
云音乐音源 a3030f000c
AUX1 音源 a303060005
AUX2 音源 a3030c000f
FM 音源 a3030a0009
FM 搜台 ba03000003
通道 4
打开 a304070102
关闭 a30403fff8
下一曲 a30409000d
上一曲 a304050001
暂停 a304020107
播放 a304020006
静音 ab0400fffb
取消静音 ab0401fffa
音量++ a304010005

音量-- a30408000c
MP3 音源 a3040b000f
SD 卡音源(ANDROID 机型) a3040d020b
云音乐音源 a3040f000b
AUX1 音源 a304060002
AUX2 音源 a3040c0008
FM 音源 a3040a000e
FM 搜台 ba04000004
通道 5
打开 a305070103
关闭 a30503fff9
下一曲 a30509000c
上一曲 a305050000
暂停 a305020106
播放 a305020007
静音 ab0500fffa
取消静音 ab0501fffb
音量++ a305010004
音量-- a30508000d
MP3 音源 a3050b000e
SD 卡音源(ANDROID 机型) a3050d020a
云音乐音源 a3050f000a
AUX1 音源 a305060003
AUX2 音源 a3050c0009
FM 音源 a3050a000f
FM 搜台 ba05000005
通道 6
打开 a306070100
关闭 a30603fffa
下一曲 a30609000f
上一曲 a306050003
暂停 a306020105
播放 a306020004
静音 ab0600fff9
取消静音 ab0601fff8
音量++ a306010007
音量-- a30608000e
MP3 音源 a3060b000d
SD 卡音源(ANDROID 机型) a3060d0209
云音乐音源 a3060f0009
AUX1 音源 a306060000
AUX2 音源 a3060c000a
FM 音源 a3060a000c
FM 搜台 ba06000006

通道 7
打开 a307070101
关闭 a30703fffb
下一曲 a30709000e
上一曲 a307050002
暂停 a307020104
播放 a307020005
静音 ab0700fff8
取消静音 ab0701fff9
音量++ a307010006
音量-- a30708000f
MP3 音源 a3070b000c
SD 卡音源(ANDROID 机型) a3070d0208
云音乐音源 a3070f0008
AUX1 音源 a307060001
AUX2 音源 a3070c000b
FM 音源 a3070a000d
FM 搜台 ba07000007

9. 485 命令说明

下面所列出来的命令仅支持 T3/W5/I32/Y 系列中央机, 具体其他功能请参考 485 协议文档.

注意:

音乐主机的 485 默认配置是 4800 8n1, 接口为主机背面的 DB9 接口。

json 协议也可以在 485 上使用, 并且 485 仅能控制一台音乐设备,不能同时控制多台.

若想要实现一个 485 控制多台主机需要额外使用"悠达 485 协议转换器".

当调试 485 不通时, 请使用悠达 485 控制面板控制, 若正常则 485 正常, 请检查接线和 485 配置.

通道 0

开机 B900030003
关机 B900040004
MP3 B900050207
FM B900050104
AUX1 B900050005
AUX2 B900050306
云音乐 B900050500
网络电台 B900050603
上一曲 A300050005
下一曲 A300090009
上一专辑 A300030003
下一专辑 A300070007
播放 A300020002
暂停 A300020103

静音 A300040004
音量加 A300060006
音量减 A300080008
蓝牙 A3000E000E
通道 1
开机 B901030002
关机 B901040005
MP3 B901050206
FM B901050105
AUX1 B901050004
AUX2 B901050307
云音乐 B901050501
网络电台 B901050602
上一曲 A301050004
下一曲 A301090008
上一专辑 A301030002
下一专辑 A301070006
播放 A301020003
暂停 A301020102
静音 A301040005
音量加 A301060007
音量减 A301080009
蓝牙 A3010E000F
通道 2
开机 B902030001
关机 B902040006
MP3 B902050205
FM B902050106
AUX1 B902050007
AUX2 B902050304
云音乐 B902050502
网络电台 B902050601
上一曲 A302050007
下一曲 A30209000B
上一专辑 A302030001
下一专辑 A302070005
播放 A302020000
暂停 A302020101
静音 A302040006
音量加 A302060004
音量减 A30208000A
蓝牙 A3020E000C
通道 3
开机 B903030000

关机 B903040007
MP3 B903050204
FM B903050107
AUX1 B903050006
AUX2 B903050305
云音乐 B903050503
网络电台 B903050600
上一曲 A303050006
下一曲 A30309000A
上一专辑 A303030000
下一专辑 A303070004
播放 A303020001
暂停 A303020100
静音 A303040007
音量加 A303060005
音量减 A30308000B
蓝牙 A3030E000D
通道 4
开机 B904030007
关机 B904040000
MP3 B904050203
FM B904050100
AUX1 B904050001
AUX2 B904050302
云音乐 B904050504
网络电台 B904050607
上一曲 A304050001
下一曲 A30409000D
上一专辑 A304030007
下一专辑 A304070003
播放 A304020006
暂停 A304020107
静音 A304040000
音量加 A304060002
音量减 A30408000C
蓝牙 A3040E000A
通道 5
开机 B905030006
关机 B905040001
MP3 B905050202
FM B905050101
AUX1 B905050000
AUX2 B905050303
云音乐 B905050505

网络电台 B905050606
上一曲 A305050000
下一曲 A30509000C
上一专辑 A305030006
下一专辑 A305070002
播放 A305020007
暂停 A305020106
静音 A305040001
音量加 A305060003
音量减 A30508000D
蓝牙 A3050E000B
通道 6
开机 B906030005
关机 B906040002
MP3 B906050201
FM B906050102
AUX1 B906050003
AUX2 B906050300
云音乐 B906050506
网络电台 B906050605
上一曲 A306050003
下一曲 A30609000F
上一专辑 A306030005
下一专辑 A306070001
播放 A306020004
暂停 A306020105
静音 A306040002
音量加 A306060000
音量减 A30608000E
蓝牙 A3060E0008
通道 7
开机 B907030004
关机 B907040003
MP3 B907050200
FM B907050103
AUX1 B907050002
AUX2 B907050301
云音乐 B907050507
网络电台 B907050604
上一曲 A307050002
下一曲 A30709000E
上一专辑 A307030004
下一专辑 A307070000
播放 A307020005

暂停 A307020104

静音 A307040003

音量加 A307060001

音量减 A30708000F

蓝牙 A3070E0009

