

杰理AC系列OTA库的开发说明_小程序

声明

- 本项目所参考、使用技术必须全部来源于公知技术信息，或自主创新设计。
- 本项目不得使用任何未经授权的第三方知识产权的技术信息。
- 如个人使用未经授权的第三方知识产权的技术信息，造成的经济损失和法律后果由个人承担。

版本历史

版本	日期	修改者	修改记录
1.0.0	2022/03/16	张焕明	初始版本，接口说明，开发说明

一、概述

本文档是为了方便后续项目维护和管理、记录开发内容而创建。
小程序OTA库是基于小程序RCSP库开发，所以包含了许多RCSP接口。

二、接口说明

主要讲述比较重要的接口，只是作为参考。

2.1 OTA相关接口

```
/**
 * function: 初始化蓝牙库
 */
export function init()
/**
 * function 注册回调
 * @param {Object} 全局事件回调
 */
export function addCallback(callback)
/**
 * function 移除回调
```

```

    * @param {Object} 全局事件回调
    */
export function removeCallback(callback)
/**
    * function 读取设备的信息，在onTargetInfo中接收设备的返回的数据,建议一次连接只读取一
    次。
    *
    * @param {int} mask 设备配置信息掩码
    *
    *      BIT0:协议版本号获取
    *      BIT1:POWERUP SYS_INFO(开机就要获取的系统信息)
    *      BIT2:EDR INFO
    *      BIT3:PLATFORM
    *      BIT4:FUNCTION INFO
    *      BIT5:固件版本号获取
    *      BIT6:AI SDK or ST SDK
    *      BIT7:uboot版本
    *      BIT8:ota升级标识
    *      BIT9:升级状态
    *      BIT10:厂商id和产品id
    *      BIT11:产品authKey
    *      BIT12:产品ProCode
    *      BIT13:协议MTU
    *
    * @param {Object} callback 命令操作回调
    */
export function getTargetInfo(mask, callback)
/**
    *
    * 关闭或者打开固件tws的信息推送
    * @param {boolean} open
    * @param {Object} callback 命令操作回调
    */
export function openOrCloseAdvInfoUpdate(open, callback)
/**
    *
    * 开始OTA
    * @param {Uint8Array} data 升级数据
    * @param {Object} listener OTA流程监听
    */
export function startOta(data, listener)

```

2.2 命令操作回调

与设备通信，命令回复回调。

```

object callback {

    /**
    * 命令回复成功
    * @param cmd 命令对象
    */
    function success(cmd);

    /**
    * 命令回复失败
    *
    * @param failObj {

```

```

    *   code: 命令status,
    *   errMsg: 错误信息,
    *   res: 命令对象,
    *}
    */
function fail(failObj);
}

```

2.3杰理RCSP全局事件回调

通过addCallback/removeCallback接口增加和移除

```

let callback = {
  // 蓝牙适配器状态
  onAdapter: (status) => { },
  // 发现设备
  onFound: (devices) => { },
  // 连接状态变化
  // obj: {status, device}
  onConnection: (obj) => { },
  // 接收固件向设备推送的命令
  onCmdReceive: (cmd) => { },
  // 设备信息读取回调
  onTargetInfo: (info) => { },

  /**
   * 以下回调为音箱功能相关
   */
  // 公共属性变化
  onCommonSysInfo: (info) => { },
  // 音乐模式属性变化
  onMusicInfo: (info) => { },
  // linein模式属性变化
  onLineinInfo: (info) => { },
  // FM模式属性变化
  onFmInfo: (info) => { },
  // RTC模式属性变化
  onRtcInfo: (info) => { },
  // 灯光模式属性变化
  onLightInfo: (info) => { },
  /**
   * 以下回调为目录浏览相关
   */
  // 卡设备状态变化
  onDevicesChange: (devices) => { },
  // 点播成功
  onPlaySuccess: () => { },
  // 点播失败
  onPlayFailed: (reason) => { },
  // 文件读取失败
  onFileFailed: (reason) => { },
  // 开始读取文件
  onFileStart: (device: first) => { },
  // 读取到的文件列表回调
  onFileReceive: (devices, files) => { },
  // 文件读取结束
  onFileStop: (device, isEnd) => { },

```

```

    //twS adv信息回调
    onAdvInfo: (advInfo)=>{}
  }

```

2.4 OTA流程监听

```

let callback ={
  onMsg: () => {},
  //开始升级
  onStart: () => {},
  //升级进度
  onProgress: res => {
    res.type;//0x00:'发送uboot升级数据' 0x01:'发送sdk升级数据'
    res.progress;//发送数据进度
  },
  //升级结束
  onFinish: () => {},
  //升级错误
  onError: e => {
    e.code;//OTA升级错误代码
    e.msg;//OTA升级错误信息
  }
}
})
}

```

2.5 OTA库的蓝牙连接状态

```

export const CONNECT_STATUS_CONNECTING = 0;//连接中
export const CONNECT_STATUS_CONNECTED = 1;//已连接
export const CONNECT_STATUS_AUTHING = 6;//认证中，###不需要处理判断
export const CONNECT_STATUS_DISCONNECTING = 2;//断开中
export const CONNECT_STATUS_DISCONNECTED = 3;//已断开
export const CONNECT_STATUS_FAILED = 4;//连接失败

```

2.6 OTA升级错误代码

```

export const ERR_CODE_NOT_CONNECTED = 1; //设备未连接
export const ERR_CODE_READ_HEAD_OFFSET = 2; //读取文件头偏移
export const ERR_CODE_SEND_HEAD_DATA = 3; //发送文件头
export const ERR_CODE_READ_TARGET_INFO = 4; //读取信息
export const ERR_CODE_ENTER_UPGRADE = 5; //进入升级模式错误
export const ERR_CODE_READ_UPGRADE_STATUS = 6; //读取升级状态
export const ERR_CODE_SWITCH_WAY = 7; //切换连接方式
export const ERR_CODE_RECONNECT = 8; //重新连接
export const ERR_CODE_DISCONNECT = 9; //异常断开
export const ERR_CODE_NOT_UPGRADE_DATA = 10; //升级数据为空
export const ERR_CODE_SCAN = 11; //发现设备失败
export const ERR_WAIT_PULL_TIMEOUT = 13; //等待数据拉取命令超时

```

2.7 杰理RCSP库接口

方法名	功能	参数	返回类型
init	初始化	无	void
addCallback	添加回调	callback	void
removeCallback	移除回调	callback	void
startScan	开始扫描	operationCallback	void
stopScan	结束扫描	无	void
isConnectedDevice	设备是否连接	device	boolean
connect	连接设备	device	void
getConnectedDevice	获取已连接的设备	无	Object
disconnect	断开设备连接	无	void
sendCmdToDevice	发送命令到设备	cmd	void
append	读取下一级文件目录	floder	void
back	返回上一级	device 当前卡设备 hasEvent: 是否有回调	void
loadMore	读取更多文件	device 当前卡设备 start: 开始索引位置	void
playFile	播放音乐文件	file	void
getFileDevices	获取当前在线的usb、sd卡列表	无	Array
getFileDeviceByIndex	通过卡设备索引获取卡设备	index: 设备索引	Object
getTargetInfo	获取设备信息	mask: 掩码位 operationCallback	void
getCommonInfo	读取设备通用的系统属性信息	mask: 掩码位 operationCallback	void
getMusicInfo	getMusicInfo		
setVolume	设置音量		
setEq	设置eq		
setFmtx	设置fmtx的值		
musicFun	音乐模式的功能命令,		
lineinFun	linein模式的功能命令		
fmFun	FM模式的功能命令,		

方法名	功能	参数	返回类型
switchFunMode	模式切换功能命令，		
startOTA	开始OTA	data(升级数据), listener(升级回调)	void
openOrCloseAdvInfoUpdate	关闭或者打开固件tws的信息推送	open	void