



ZS110A VoiceBleTest 二次开发 参考文档

最新版本号：1.0
2019/01/21

目录

1	概述	2
1.1	编写目的.....	2
1.2	术语和缩写词.....	2
1.3	版本历史.....	2
2	工程描述	3
3	基本流程图	3
4	模块介绍	4
5	模块详细分析	4
5.1	BLE 的连接和数据接收模块.....	4
5.2	解码模块.....	6
5.3	语音的识别模块.....	6
5.4	OTA 模块.....	6
5.5	其他.....	8
6	声明.....	9

1 概述

1.1 编写目的

详细介绍 VoiceBleTest 源码的基本组成模块和一些接口。

1.2 术语和缩写词

缩写和术语	解 释

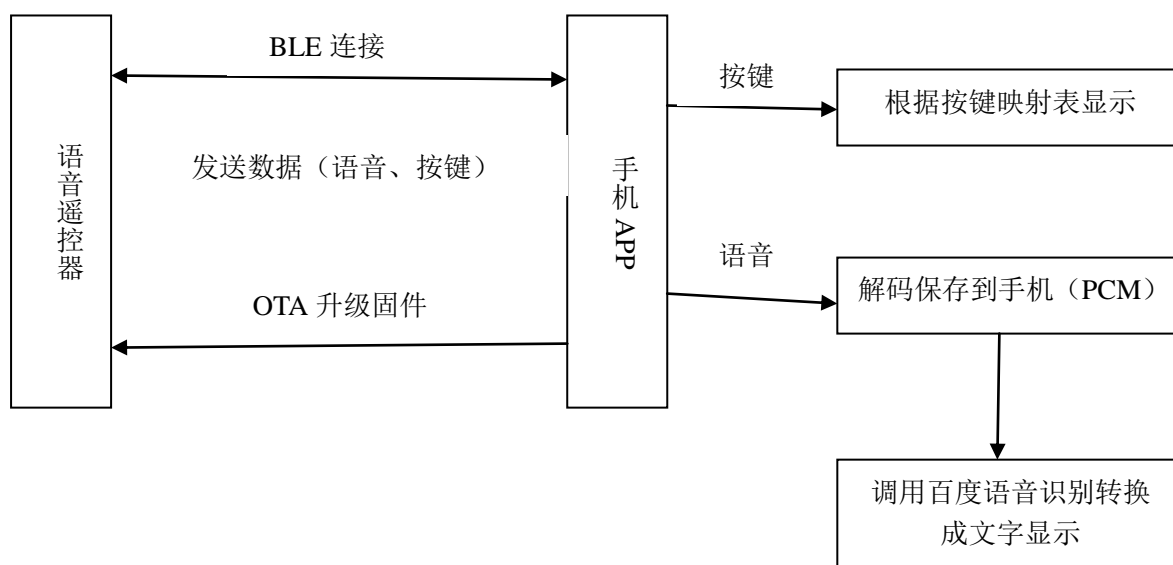
1.3 版本历史

日期	版本号	注释	作者
2018-01-21	1.0	建立初始版本	ZS110A 项目组

2 工程描述

VoiceBleTest 是一个 Android APP，主要用途是与语音遥控器通过低功耗蓝牙进行连接，接收语音（解码保存）和按键信息，显示在页面上，可以将接收的语音翻译成文字，语音识别采用的是百度开源的识别系统。

3 基本流程图



4 模块介绍

VoiceBleTest 主要有四个重要模块，BLE 的连接和数据接收模块，语音的解码模块，语音的识别模块，OTA 模块

5 模块详细分析

5.1 BLE 的连接和数据接收模块

Demo 中使用了一个开源库 RxAndroidBle，进行 BLE 的连接和数据操作。开源库地址 (<https://github.com/Polidea/RxAndroidBle>)，可以参考一下。应用启动进入的是 ScanActivity，在该 Activity 搜索设备，连接设备后跳转到 MainActivity，在 MainActivity 进行连接，成功连接设备后会监听数据通道，

```
UUID_RECEIVE_DATA = "001120a1-2233-4455-6677-889912345678";//语音数据接收通道 UUID
```

```
UUID_CONTROL = "001120a3-2233-4455-6677-889912345678";//按键数据通道 UUID
```

主要调用接口：

ScanActivity:

```
private void startScan() //扫描设备，主要也是使用第三方 RxAndroidBle 接口，详情见源码
```

```
private void stopScan() //停止扫描
```

```
private void connectBle(ScanResult scanResults, boolean autoConnect)//连接某台设备跳转到
```

MainActivity

```
private void createBond(ScanResult itemAtPosition)//配对
```

```
public void askMultiplePermission() //申请需要权限
```

```
private void handleLogic(View contentView, ScanResult itemAtPosition)// 处理弹出显示内容、点击事件等逻辑
```

MainActivity:

创建 ble 连接状态的监听，在回调 onConnectionStateChange 中接收状态改变并更新 UI

```
bleDevice = VoiceBleTestApplication.getRxBleClient(this).getBleDevice(macAddress);
```

```
bleDevice.observeConnectionStateChanges()
```

```
.compose(bindUntilEvent(DESTROY))
.observeOn(AndroidSchedulers.mainThread())
.subscribe(this::onConnectionStateChange);
```

由于 APP 会自动回连，在 `connectTask` 任务中执行

```
private Runnable connectTask = new Runnable() {
    @Override
    public void run() {
        //Log.d(TAG, "connectTask running: " + state);
        mHandler.sendEmptyMessage(RECONNECT);
        mHandler.postDelayed(this, 3000);
    }
};

private void onConnect() //连接 BLE 设备，连接成功跳转 onConnectionReceived，连接失败跳
onConnectionFailureLost

public void onNotify()//连接成功后会调用建立监听函数，建立两个通道的监听
监听建立成功后，就可以接收遥控器发送的数据了，

private void onNotificationReceivedControl(byte[] bytes) //接收按键数据
private void onNotificationReceived(byte[] bytes) //接收语音数据
```

详细介绍一下，按键数据和语音数据的处理过程，因为接收的语音数据是加密的，所以要选择对应的解码参数进行解码。

如何选择解码参数呢？是根据发送的按键值，不同的键值选择不同的解码参数

0C0200, 0F0200 对应算法 3

0D0200 对应算法 1

0E0200 对应算法 2

1E0200 对应算法 5

100200 对应算法 4

当然这个根据遥控器加密算法对应的，可自行修改，只要加解密函数对应就行
CCCC00 语音结尾按键标志，说明语音发送完了，添加界面显示

每段语音是接收到开始录音的按键值和结束键值之前的数据。

解码语音数据，除了算法 1 是每 40 字节解码一次，其余算法都是 20 字节解码一次，所以如果是算法 1 需要特殊处理一下，接收够 40 字节才解码，否则不解码。

5.2 解码模块

解码类的定义是 DecodeJni 类，主要是调用 libasc_dec.so, cpp 调用在 \src\main\jni, VoiceBleTestV1.1 可能没有, VoiceBleTestV1.0 的 com_actions_voicebletest_jni_DecodeJni.cpp, libDecodeJni 是调用 libasc_dec.so 的，是一层 jni 调用。解码库的参数介绍需要算法同事提供，这里不做阐述。

每次解码需要先初始化，即调用：

```
mBitstreamlen = mDecodeJni.decodeInit(mDecodeAlgrithm);
```

然后是

```
short[] ret = mDecodeJni.Decode(shorts, mBitstreamlen, mDecodeAlgrithm);
```

ret 是解码结果返回，解码后写到文件。

解码出来的数据是 PCM 数据，保存路径是 SD 卡的

/com.actions.voicebletest/pcm/文件夹，命名是按时间戳命名。

```
private void getPath()//获取保存语音文件的名字
```

5.3 语音的识别模块

语音的识别模块是使用的百度的语音识别 SDK

百度的 Android SDK 文档：

<http://ai.baidu.com/docs/#/ASR-Android-SDK/414a7d26>

百度是 SDK 下载地址：<https://ai.baidu.com/sdk#asr>

这里不做阐述，百度的文档写的很详细，请自行参考。

5.4 OTA 模块

OTA 模块另有一个文档参考，流程这里也不做阐述，主要接口描述如下。

OTAFragment.java

```
public void showFileChooser() //显示选择 ota 升级文件的对话框
```

```
private void unzipOTA() //解压 ota.zip, 读取 xml 信息
```

```
private void getBatteryValue()//获取电量信息，大于 30%才能升级
```

```
private void onUploadClick()//点击升级按钮，正在升级则暂停，否则开始升级
```

```
private void onCancelUpload()//取消升级
```

```
private void startUploadBin()//升级某个 bin
```

```
public void WdxcGetFiles(int fileHdl)//获取小机分区表信息
private void WdxcFtcSendGetReq(int fileHdl, int offset, int len, int type)//发送获取分区表信息命令
private void onWriteFtc(byte[] data)//写数据到 UUID"005f0003-2ff2-4ed5-b045-4C7463617865"
private void onWriteDc(byte[] data)//写数据到 UUID "005f0002-2ff2-4ed5-b045-4C7463617865"
public void openFile(InputStream stream)//读取 bin 文件信息
private int getNumberOfPackets()//获取文件需要发送的包数
private void startToFoundDfuService()//发现 OTA 服务, 主要监听三个通道
private void onConnectionFailureDfuNotFound(Throwable throwable)//找不到 OTA 服务抛出异常
public void reset()//重置状态信息
private void startSetupNotification() //建立监听服务

private void onNotificationReceivedFtc(byte[] bytes)//接收
"005f0003-2ff2-4ed5-b045-4C7463617865"通道数据
主要是接收小机端接收数据的进度情况, 命令回复, 一些校验情况, 版本信息
private void wdxcParseFtd(byte[] data)//没有用到
private void onNotificationReceivedFtd(byte[] bytes)//接收小机分区表信息并解析
private void onNotificationReceivedDc(byte[] bytes) //接收 UUID
"005f0002-2ff2-4ed5-b045-4C7463617865"的数据, 其实是以前的残留代码, 目前其实没有用到
void wdxcParseFtc(byte[] data)// onNotificationReceivedFtc 的主要处理
public void WdxcPutFiles(String OtaFileName)//根据小机上报的分区表信息, 如果有两个分区选择升级哪一份分区
A/B 备份机制指存在2 个相同类型的分区。
以遥控器方案为例: 定义了2 个BOOT_TYPE 分区和2 个SYSTEM_
TYPE 分区。
初始版本的分区表中: A 份分区的seq=0, B 份分区的seq=-1。
第一次进行OTA 时, 移动端读取到分区表后, 会选择seq 值较小的B 分区进行升级, 升级完成会修改 seq_B。seq_B=seq_A+1。

private void WdxcFtcSendPutReq(int fileHdl, int offset, int len, int fileSize, int type) //
告诉小机要写入文件的偏移和长度
private void WdxcFtcSendEraseReq(int fileHdl, int offset, int len) //擦除数据的偏移和长度
void WdxcFtcSendAbort(int fileHdl) //停止发送
public void WdxcFtcSendGetVersion()//获取版本号
public void WdxcFtcSendReset()//重置小机状态信息
public void WdxcFtcSendUpdateConnParam(int interval_min, int interval_max, int latency, int
timeout)
//更新连接参数信息
public void WdxcDcSetDisconnectAndReset() //断开连接并重启, 没有用到
private void startUploadingFile() //开始发送文件
```



```
public void onFileTransferValidation()//校验完成
public void onFileTransferCompleted() //发送文件完成
public void WdxcVerifyFiles()//校验文件
private void WdxcFtcSendVerifyFile(int fileHdl)//校验文件发送命令
private boolean checkCRC32(String filePath)//校验.zip 包的文件 crc32 是否正确
private void sendPacketAsync() //异步发送文件，使用 RxAndroidBle 库发送
```

5.5 其他

VoiceTestFragment 是接收按键，显示按键，显示语音，点击播放语音和翻译成文本的界面代码，使用 ListView 和 ChatAdapter 实现，按键映射定义在 KeyMapping.java；接收的按键和语音显示条目使用 sqlite 存储（MessageReaderDbHelper.java），日志保存模块在 LogcatManager.java, Utils 目录下的为公共函数类，widget 目录下是一些界面控件的实现。

6 声 明

Disclaimer

Information given in this document is provided just as a reference or example for the purpose of using Actions' products, and cannot be treated as a part of any quotation or contract for sale.

Actions products may contain design defects or errors known as anomalies or errata which may cause the products' functions to deviate from published specifications. Designers must not rely on the instructions of Actions' products marked "reserved" or "undefined". Actions reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

ACTIONS DISCLAIMS AND EXCLUDES ANY AND ALL WARRANTIES, INCLUDING WITHOUT LIMITATION ANY AND ALL EXPRESS OR IMPLIED WARRANTIES OF MERCHANTABILITY, ACCURACY, SECURITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND AGAINST INFRINGEMENT OF INTELLECTUAL PROPERTY AND THE LIKE TO THE INFORMATION OF THIS DOCUMENT AND ACTIONS PRODUCTS.

IN NO EVENT SHALL ACTIONS BE LIABLE FOR ANY DIRECT, INCIDENTAL, INDIRECT, SPECIAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES WHATSOEVER, INCLUDING WITHOUT LIMITATION FOR LOSS OF DATA, PROFITS, SAVINGS OR REVENUES OF ANY KIND ARISING FROM USING THE INFORMATION OF THIS DOCUMENT AND ACTIONS PRODUCTS. REGARDLESS OF THE FORM OF ACTION, WHETHER BASED ON CONTRACT; TORT; NEGLIGENCE OF ACTIONS OR OTHERS; STRICT LIABILITY; OR OTHERWISE; WHETHER OR NOT ANY REMEDY OF BUYER IS HELD TO HAVE FAILED OF ITS ESSENTIAL PURPOSE, AND WHETHER ACTIONS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR NOT.

Actions' products are not designed, intended, authorized or warranted for use in any life support or other application where product failure could cause or contribute to personal injury or severe property damage. Any and all such uses without prior written approval of an Officer of Actions and further testing and/or modification will be fully at the risk of the customer.

Ways of obtaining information

Copies of this document and/or other Actions product literature, as well as the Terms and Conditions of Sale Agreement, may be obtained by visiting Actions' website at: <http://www.actions-semi.com> or from an authorized Actions representative.

Trademarks

The word "Actions" and the logo are the trademarks of Actions Semiconductor Co., Ltd, and

Actions (Zhuhai) Technology Co., Limited is authorized to use them. Word “炬芯” is the trademark of Actions (Zhuhai) Technology Co., Limited. Names and brands of other companies and their products that may from time to time descriptively appear in this document are the trademarks of their respective holders, no affiliation, authorization, or endorsement by such persons are claimed or implied except as may be expressly stated therein.

Rights Reserved

The provision of this document shall not be deemed to grant buyers any right in and to patent, copyright, trademark, trade secret, know how, and any other intellectual property of Actions or others.

Miscellaneous

Information contained or described herein relates only to the Actions products and as of the release date of this publication, abrogates and supersedes all previously published data and specifications relating to such products provided by Actions or by any other person purporting to distribute such information.

Actions reserves the rights to make changes to information described herein at any time without notice. Please contact your Actions sales representatives to obtain the latest information before placing your product order.

Additional Support

Additional products and company information can be obtained by visiting the Actions website at: <http://www.actions-semi.com>

支持:

如欲获得公司及产品的其它信息，欢迎访问我公司网站: <http://www.actions-semi.com>

炬芯（珠海）科技有限公司

地址：珠海市唐家湾镇高新区科技四路 1 号 1#厂房一层 C 区

电话：+86-756-3392353

传真：+86-756-3392251

邮政编码：519085

网址：<http://www.actions-semi.com>

电子邮件（业务）：mp-sales@actions-semi.com

（技术支持）：mp-cs@actions-semi.com