**安卓语音录制播放及实时发送**

在[Android](http://lib.csdn.net/base/android)中音频编解码有两种方式：

1、使用AudioRecord采集音频，用这种方式采集的是未经压缩的音频流；用AudioTrack播放实时音频流。用这两个类的话，如果需要对音频进行编解码，就需要使用JNI移植编解码库。

2、使用MediaRecorder获取编码后的AMR音频，但由于MediaRecorder的特点，只能将流保存到文件中；使用MediaPlayer来播放AMR音频流，但同样MediaPlayer也只能播放文件流。

这两种方式的优缺点是：

1、AudioRecord

主要是实现边录边播以及对音频的实时处理,这个特性让他更适合在语音方面有优势

优点：语音的实时处理，可以用代码实现各种音频的封装

缺点：输出是PCM格式文件，如果保存成音频文件，是不能够被播放器播放的，所以必须先写代码实现数据编码以及压缩

2、MediaRecorder

已经集成了录音、编码、压缩等，支持少量的录音音频格式，大概有,aac,amr,3gp等

优点：集成，直接调用相关接口即可，代码量小

缺点：无法实时处理音频；输出的音频格式不是很多，例如没有输出mp3格式文件

由于电台项目功能需要，选择第一种方式来实现音频的实时录制及传输。

AndioRecord类的主要功能是让各种JAVA应用能够管理音频资源，以便它们通过此类来录制音频输入硬件所收集的声音。

AudioRecord类提供了三个获取声音数据的方法分别是read(byte[], int, int)，read(short[], int, int)，read(ByteBuffer, int)。在录音过程中，app所需要做的就是通过这三个类方法中的一个去及时地获取AudioRecord对象的录音数据。

开始录音的时候，AudioRecord需要初始化一个相关联的声音buffer, 这个buffer主要是用来保存新的声音数据。buffer的大小可以在对象构造期间去指定。它表明一个AudioRecord对象还没有被读取（同步）声音数据前能录多长的音(即一次可以录制的声音容量)。声音数据从音频硬件中被读出，数据大小不超过整个录音数据的大小，即每次读取初始化buffer容量的数据。

一般情况下录音实现的简单流程如下：

1.创建一个数据流。

2.构造一个AudioRecord对象，其中需要的最小录音缓存buffer大小可以通过getMinBufferSize方法得到。如果buffer容量过小，将导致对象构造的失败。

3.初始化一个buffer，该buffer大于等于AudioRecord对象用于写声音数据的buffer大小。

4.开始录音。

5.从AudioRecord中读取声音数据到初始化buffer，将buffer中数据导入数据流（保存文件或socket发送）。

6.停止录音。

7.关闭数据流。