

Android手机直播（三）声音采集



风从影 (/u/c3dcf28e69f2) [+ 关注](#)

2017.01.12 15:30* 字数 4547 阅读 7279 评论 46 喜欢 88 赞赏 1

(/u/c3dcf28e69f2)

一、文章说明

开始写文章了，才知道写文章真心耗费心力，希望自己尽量做到快速更新，也希望这些文章真心能帮助到开发者们。

这篇文章主要讲述Android声音采集相关的知识，首先介绍声音的基础知识，然后介绍如何采集声音，最后再讲述Android上声音录制和回声消除的相关步骤。

整个项目已经开源，开源地址：SopCastComponent (<https://github.com/LaiFeng-Android/SopCastComponent>)

二、基础知识

声音的原理

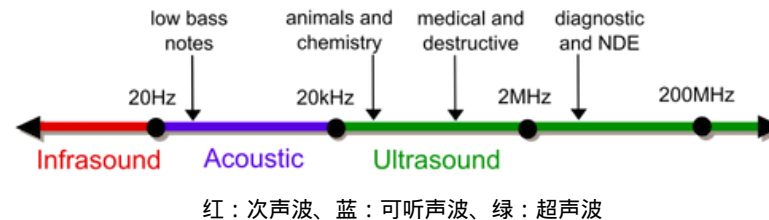
声音是振动产生的声波，通过介质（空气或固体、液体）传播并能被人或动物听觉器官所感知的波动现象。声音的频率一般会以赫兹表示，记为Hz，指每秒钟周期性振动的次数。而分贝是用来表示声音强度的单位，记为dB。

声音是一种波动，当演奏乐器、拍打一扇门或者敲击桌面时，声音的振动会引起介质——空气分子有节奏的振动，使周围的空气产生疏密变化，形成疏密相间的纵波，这就产生了声波，这种现象会一直延续到振动消失为止。

任何器官所接收的声音频率都有其范围限制。人类的耳朵一般只能听到约在20Hz至20000 Hz（20kHz）范围内的声音，其上限会随年龄增加而降低。其他物种动物的听觉频率范围也有所不同，像狗可以听到超过20kHz的声音，但无法听到40Hz以下的声音。不同物种动物的听觉频率范围如下：



蝙蝠：1000 ~ 120000Hz
海豚：2000 ~ 100000Hz
猫：60 ~ 65000Hz
狗：40 ~ 50000Hz
人：20 ~ 20000Hz



麦克风采集

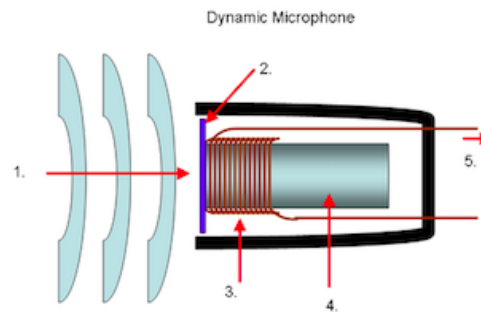
麦克风（又称微音器或话筒，正式的中文名是传声器），译自英文microphone，是一种将声音转换成电子信号的换能器。根据麦克风的制作原理，分为以下几类：

动圈麦克风

动圈式麦克风基本构造包含线圈、振膜、永久磁铁三部分。当声波进入麦克风，振膜受到声波的压力而产生振动，与振膜连接在一起的线圈则开始在磁场中移动，根据法拉第定律以及楞次定律，线圈会产生感应电流。

动圈式麦克风因为含有线圈和磁铁，不像电容式麦克风轻便，灵敏度较低，高低频响应表现较差。优点是声音较为柔润，适合用来收录人声。

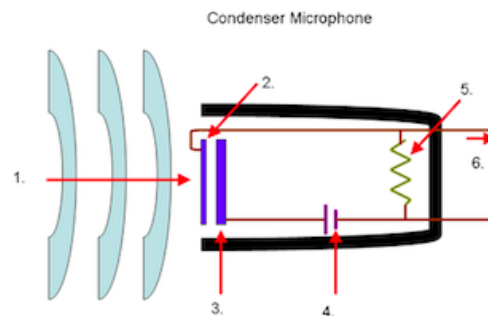




1、声波 2、振动膜 3、线圈 4、磁铁 5、输出信号

电容式麦克风

电容式麦克风并没有线圈及磁铁，靠着电容两片隔板间距离的改变来产生电压变化。当声波进入麦克风，振动膜产生振动，因为基板是固定的，使得振动膜和基板之间的距离会随着振动而改变，根据电容的特性，当两块隔板距离发生变化时，电容值C会产生改变，又由于 $Q = C * V$ ，当C改变时就会造成电量Q的改变。因为在电容式麦克风中需要维持固定的极板电压V，所以此类麦克风需要额外的电源才能运作，一般常见的电源为电池。电容式麦克风因灵敏度较高，常用于高质量的录音。



1、声波 2、振动膜 3、基板 4、电池 5、电阻 6、输出信号

驻极体电容麦克风

电容式麦克风一般需要额外的电源才能运作，但是驻极体电容麦克风却可以不需要额外的电源。驻极体又叫“永电体”，本身会带有固定数量的电荷数，整个线路没有电量消耗（线路去除上图的电池和电阻），根据公式： $Q = CU$ 所以当C变化时必然引起电容器两端电压U的变化，从而输出电信号，实现声—电的变换。由于实际电容器的电容量很小，



输出的电信号极为微弱，输出阻抗极高，可达数百兆欧以上。因此，它不能直接与放大电路相连接，必须连接阻抗变换器。通常用一个专用的场效应管和一个二极管复合组成阻抗变换器。由于场效应管是有源器件，需要一定的偏置和电流才可以工作在放大状态，因此，驻极体话筒都要加一个直流偏置才能工作。

微机电麦克风

微机电麦克风指使用微机电技术做成的麦克风，也称麦克风芯片或硅麦克风。微机电麦克风的压力感应膜是以微机电技术直接蚀刻在硅芯片上，此集成电路芯片通常也集成入一些相关电路，如前置放大器。大多数微机电麦克风的设计，在基本原理上是属于电容式麦克风的一种变型。微机电麦克风也常内置模拟数字转换器，直接输出数字信号，成为数字式麦克风，以利与现今的数字电路连接。微机电麦克风的主要应用于部分的手机、PDA等小型移动产品。

还有其他类型的麦克风在这就不多做讲述。

麦克风降噪

随着科技的发展，现在即使在非常嘈杂的环境下，接听电话的另一方也能听得清清楚楚，这主要得益于手机降噪技术的发展。在现在的手机我们常常看到不仅仅只有一个麦克风，而是有2个甚至是3个，而这多出来的几个就是手机降噪的关键。





一般来说手机都有两个麦克风，顶部和底部都各有一个。这两个麦看起来都非常小，但是两者的作用有着明显的区别，其中底部的麦是用来提供清晰通话，而顶部的麦是用来消除噪音。

由于顶部和底部在通话时距离音源的距离不同，所以两个麦拾取的音量大小也是有不同的，利用这个差别，我们就可以过滤掉噪声保留人声了。在打电话时，两个麦克风所拾取的背景噪声音量是基本相同的，而记录的人声会有6dB左右的音量差。顶端麦收集噪声后，通过解码生成补偿信号后就可以用来消除噪音了。

回声

回声（或称回音）是指障碍物对声音的反射。声波在遇到障碍物时，一部分声波会穿过障碍物，而另一部分声波会反射回来形成回声。若障碍物具有坚硬光滑的表面易产生回声；反之，具有柔软的表面则易吸收声音；另外，粗糙的表面易散射声音。回声相比那些直接传播的声音所经过的路程更长，所以会比直接传播的声音晚被听到。如果两列声

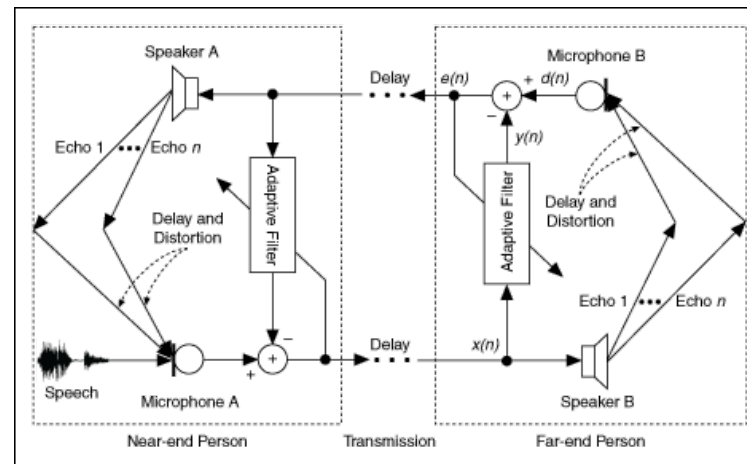


波的时间间隔小于0.1秒，人耳边无法分辨，只能听到被延长的声音。因为室温（20℃）时空气中的声速是343米每秒，所以站在声源处的人要听到回声需要障碍物到声源的距离至少17米。

回声消除

很多时候直播有连麦的需求，这时候就需要对采集的声音进行回声消除。当处在连麦的情况下，手机一边播放对方的声音，一边用麦克风进行采集，然后又将采集的声音传送给对方，这样的话对方就会听到自己的回声，由于这个循环回路一直进行，从而就会使得回声越来越多，最后出现嗡鸣声。

回声消除就是在麦克风录制外音的时候去除掉手机自身播放出来的声音，这样就将对方的声音从采集的声音中过滤出去，从而就避免了回声的产生。下面一张图片很好展示了回声消除的机制。



回声消除

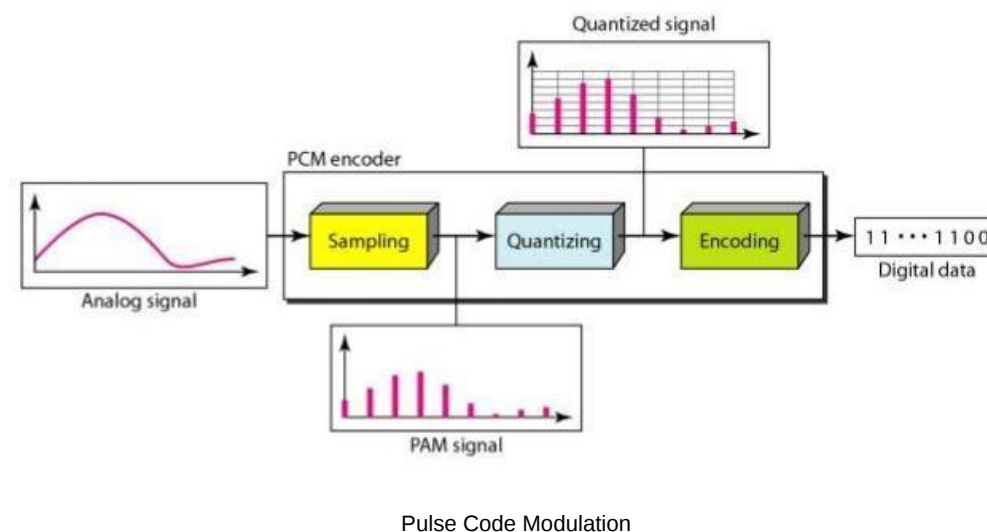
在近端，麦克风会采集到扬声器播放出来的远端声音，假设这路声音为 $y(n)$ ，当然由于需要将远端传来播放出来，我们当然能得到远端传来的声音信号，假设这路声音为 $x(n)$ 。不难发现 $x(n)$ 经过扬声器的播放，然后经过空气的传播，最后被麦克风采集，然后变为 $y(n)$ ， $x(n)$ 和 $y(n)$ 具有明显的相关性。假设麦克风采集到的总声音信号为 $z(n)$ ，这时候需要通过自适应滤波器根据 $x(n)$ 找出 $z(n)$ 中的 $y(n)$ ，然后从 $z(n)$ 中过滤掉 $y(n)$ 。



三、声音采集

在之前已经讲述了麦克风的工作原理，麦克风采集到声音后转化为模拟电信号，之后需要将模拟电信号数字化转化为计算机能够识别的模拟信号。

Android中利用AudioRecord可以录制声音，录制出来的声音可以设置为PCM声音。想要将声音用计算机语言表述，则必须将声音进行数字化。将声音数字化，最常见的方式是通过脉冲编码调制PCM(Pulse Code Modulation)。声音经过麦克风，转换成一连串电压变化的信号。要将这样的电压变化的信号转化成为PCM信号则需要进行三个过程：抽样、量化、编码。要实现这三个过程，则需要使用三个参数，它们是：采样频率、采样位数和声道数。



1、采样频率

采样频率即取样频率，指每秒钟取得声音样本的次数。采样频率越高，声音的质量也就越好，声音的还原也就越真实，但同时它占的资源比较多。由于人耳的分辨率很有限，太高的频率并不能分辨出来。在16位声卡中有22KHz、44KHz等几级，其中，22KHz相当于普通FM广播的音质，44KHz已相当于CD音质了，目前的常用采样频率都不超过48KHz。



2、采样位数

采样位数即采样值或取样值（就是将采样样本幅度量化）。它是用来衡量声音波动变化的一个参数，也可以说是声卡的分辨率。它的数值越大，分辨率也就越高，所发出声音的能力越强。

在计算机中采样位数一般有8位和16位之分，8位不是说把纵坐标分成8份，而是分成2的8次方即256份；同理16位是把纵坐标分成2的16次方65536份。

采样率和采样大小的值越大，记录的波形更接近原始信号。

3、声道数

很好理解，有单声道和立体声之分，单声道的声音只能使用一个喇叭发声（有的也处理成两个喇叭输出同一个声道的声音），立体声的pcm可以使两个喇叭都发声（一般左右声道有分工），更能感受到空间效果。

那么，现在我们就可以得到pcm文件所占容量的公式：

存储量 = (采样频率 · 采样位数 · 声道 · 时间) / 8 (单位：字节数)

四、Android声音录制

Android中使用AudioRecord录制声音，根据上面讲述的声音采集原理，需要传递给AudioRecord采样频率、采样位数和声道数，除此之外还需要传入两个参数，一个是声音源，一个是缓冲区大小。

权限

在Android中录制声音需要相应的权限，注意动态申请权限的问题。

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
```

音频源

下面是Android支持的音频源：




```
/** 默认声音 */
public static final int DEFAULT = 0;
/** 麦克风声音 */
public static final int MIC = 1;
/** 通话上行声音 */
public static final int VOICE_UPLINK = 2;
/** 通话下行声音 */
public static final int VOICE_DOWNLINK = 3;
/** 通话上下行声音 */
public static final int VOICE_CALL = 4;
/** 根据摄像头转向选择麦克风*/
public static final int CAMCORDER = 5;
/** 对麦克风声音进行声音识别，然后进行录制 */
public static final int VOICE_RECOGNITION = 6;
/** 对麦克风中类似ip通话的交流声音进行识别，默认会开启回声消除和自动增益 */
public static final int VOICE_COMMUNICATION = 7;
/** 录制系统内置声音 */
public static final int REMOTE_SUBMIX = 8;
```

缓冲区大小

接下来便是要设置缓冲区大小。麦克风采集到的数据先放置在一个缓冲区里面，之后我们再从这个缓冲区里面读取数据，从而获取到麦克风录制的音频数据。在Android中不同的声道数、采样位数和采样频率会有不同的最小缓冲区大小，当AudioRecord传入的缓冲区大小小于最小缓冲区大小的时候则会初始化失败。大的缓冲区大小可以达到更为平滑的录制效果，相应的也会带来更大一点的延时。



通过下面的方法可以获得最小缓冲区的大小：

```
AudioRecord.getMinBufferSize(frequency, channelConfiguration, audioEncoding);
```



当获取失败后会返还负数，根据错误码可以得到相应的错误信息。

初始化

在对AudioRecord进行录音前需要对采样率进行设置，对于采样率，Android官方文档要求所有的手机需要对44100Hz的采样率进行支持，可是国内的一些极其少数的手机依然不支持44100Hz的采样率。以下是几种常见的采样率：

```
8000, 11025, 16000, 22050, 44100, 48000
```

在设置采样率之前需要对手机对设置的采样率是否支持进行检测，下面是一段代码是获取手机支持的音频采样率：

```
public void getValidSampleRates() {  
    for (int rate : new int[] {8000, 11025, 16000, 22050, 44100}) { // add the rate  
        int bufferSize = AudioRecord.getMinBufferSize(rate, AudioFormat.CHANNEL_CONF  
        if (bufferSize > 0) {  
            // buffer size is valid, Sample rate supported  
  
        }  
    }  
}
```

为了达到立体声的效果，Android也是支持多个声道采集的，一些情况下为了完成在所有手机上的视频，我们需要把声道设置为

AudioFormat.CHANNEL_CONFIGURATION_MONO（单声道）。

对于采样位数，选的值也是以常量的形式定义在 AudioFormat 类中，常用的是

ENCODING_PCM_16BIT（16bit），ENCODING_PCM_8BIT（8bit），

ENCODING_PCM_16BIT可以保证兼容所有Android手机的。

下面便是AudioRecord的初始化方法：



```
public AudioRecord(int audioSource, int sampleRateInHz, int channelConfig,
                   int audioFormat, int bufferSizeInBytes) throws IllegalArgumentException
```

当传入的参数出现问题时会抛出异常。AudioRecord有一个状态量用来表示AudioRecord是否被成功初始化，通过getState()方法可以获取，当返回为STATE_UNINITIALIZED表示未成功初始化，当返回为STATE_INITIALIZED表示已经成功初始化了。

读取数据

AudioRecord通过下面的方法可以读取到相应的录音数据：

```
public int read(@NonNull byte[] audioData, int offsetInBytes, int sizeInBytes) {
    return read(audioData, offsetInBytes, sizeInBytes, READ_BLOCKING);
}
```

当读取失败的时候会返回相应的负数错误码。

参数选择

Android手机有很多厂商，对于开发者来说兼容性一直以来都是一个重要的问题。在录音过程中，Android推荐的参数如下：

```
sampleRateInHz = 44100;
channelConfig = AudioFormat.CHANNEL_CONFIGURATION_MONO;
audioFormat = AudioFormat.ENCODING_PCM_16BIT
```

五、Android回声消除

在Android中回声消除可以通过三种方式进行处理：1、通过VOICE_COMMUNICATION模式进行录音，自动实现回声消除；2、利用Android自身带的AcousticEchoCanceller进行回声消除处理；3、使用第三方库（Speex、WebRTC）进行回声消除处理。



使用AudioRecord模式进行录音的时候，需要将AudioManager设置模式为MODE_IN_COMMUNICATION，还需要将麦克风打开。有一点需要特别注意，**音频采样率必须设置8000或者16000，通道数必须设为1个。**

```
AudioManager audioManager = (AudioManager)mContext.getSystemService(Context.AUDIO_SE  
audioManager.setMode(AudioManager.MODE_IN_COMMUNICATION);  
audioManager.setSpeakerphoneOn(true);
```

使用AcousticEchoCanceler过程比较简单，录制声音的时候可以通过AudioRecord得到AudioSessionId，在创建AudioTrack的时候也可以传入一个AudioSessionId，这时候将这个统一的AudioSessionId传入AcousticEchoCanceler，那么AcousticEchoCanceler将根据之前讲过的回声消除的原理进行回声消除。



```
private void initAec(int audioSessionId) {
    if(!AudioAecUtils.isAcousticEchoCancellerApproved()) {
        aecSwitch.setEnabled(false);
        return;
    }
    aec = AcousticEchoCanceller.create(audioSessionId);
    if (aec == null) {
        Log.e(TAG, "AcousticEchoCanceller.create failed");
        aecSwitch.setEnabled(false);
        return;
    }
}

private boolean setEnableAec(boolean enable) {
    if (aec == null) {
        return false;
    }
    int ret = aec.setEnabled(enable);
    if (ret != AudioEffect.SUCCESS) {
        Log.e(TAG, "AcousticEchoCanceller.setEnabled failed");
        return false;
    }
    if(enable) {
        Log.d(TAG, "Aec On");
    } else {
        Log.d(TAG, "Aec Off");
    }
    return true;
}
```

当使用Speex或者Webrtc第三方库进行回声消除的时候，需要将采集到的音频数据传入作为源数据，需要将此刻播放的音频数据传入作为参考数据，然后还需要传入一个延时间隔，这样第三方库就能工作，从而得到回声消除后的声音。因为播放的声音需要传播，而且麦克风采集声音还有相应的缓冲区，因此需要传入一个延时间隔。关于Speex和Webrtc在github上能找到相应的Android ndk库。

本人三种方式都进行了尝试，发现第一种效果最好，兼容性也较好，因为手机免提通话的时候就进行了回声消除处理，所以基本上所有的手机是支持的。第二种方式支持的很少，Nexus 5支持第二种方式。理论上第三种方式兼容性最好，但是本人多次实验发现要设置合适的延时间隔很难，有些时候设置好了，但是通话一段时间效果又变差。



六、相关链接

Android手机直播（一）总览 (<http://www.jianshu.com/p/7ebbcc0c5df7>)

Android手机直播（二）摄像机 (<http://www.jianshu.com/p/39a015f2996e>)

Android手机直播（三）声音采集 (<http://www.jianshu.com/p/2cb75a71009f>)

Android手机直播（四）Android Media API (<http://www.jianshu.com/p/d26e7d788c0e>)

七、结束语

终于写完了，各位看官觉得文章不错的话不妨点个喜欢～

📖 安卓直播 (/nb/8464631)

举报文章 © 著作权归作者所有



风从影 (/u/c3dcf28e69f2)

写了 30717 字，被 660 人关注，获得了 696 个喜欢

(/u/c3dcf28e69f2)

+ 关注

如切如磋，如琢如磨。

小礼物走一走，来简书关注我

赞赏支持



(/u/d82bd37b1d29)

♡ 喜欢 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-like-button) | 88



更多分享



(http://cwb.assets.jianshu.io/notes/images/8167022/weibo/image_7)



(/apps/download?utm_source=nbc)



登录 (/sign-in?source=desktop&utm_medium=not-signed-in-comment-form) 后发表评论

46条评论

只看作者

按喜欢排序 按时间正序 按时间倒序



zhh_happig (/u/d82bd37b1d29)

3楼 · 2017.01.12 16:56

(/u/d82bd37b1d29)
牛bbbb

👍 1人赞 💬 回复



zhh_happig (/u/d82bd37b1d29)

13楼 · 2017.03.08 11:08

(/u/d82bd37b1d29)
牛啊

👍 1人赞 💬 回复



zhh_happig (/u/d82bd37b1d29)

14楼 · 2017.03.08 11:08


(/u/d82bd37b1d29)
快更新，等不及

👍 1人赞 💬 回复

风从影 (/u/c3dcf28e69f2) : @zhh_happig (/users/d82bd37b1d29) 🤔🤔

2017.03.08 12:49 💬 回复



 添加新评论



More_Efficient (/u/45a75e771244)

2楼 · 2017.01.12 16:52

(/u/45a75e771244)

不错，写的很详细，作者真是这方面的牛人，受教了，向你学习！

 赞  回复



偶尔犯贱 (/u/5c86e01fae2c)

4楼 · 2017.01.13 12:26

(/u/5c86e01fae2c)

mark

 赞  回复



止念观息 (/u/89c695c5c04b)


5楼 · 2017.01.13 13:45


(/u/89c695c5c04b)

lz是直播行业的么

 赞  回复

风从影 (/u/c3dcf28e69f2)：是啊

2017.01.13 14:29  回复

 添加新评论



蜉杰辰螭 (/u/94233006ce3d)

6楼 · 2017.01.22 14:12


(/u/94233006ce3d)

楼主，我现在想做录制视频并且添加水印，这该如何做？


 赞  回复

风从影 (/u/c3dcf28e69f2)：@蜉杰辰螭 (/users/94233006ce3d) 等我更新以后的教程




2017.01.22 14:15  回复

风从影 (/u/c3dcf28e69f2) : @蜉杰辰蜉 (/users/94233006ce3d) 可以参考项目源码

2017.01.22 14:29  回复

蜉杰辰蜉 (/u/94233006ce3d) : @风从影 (/users/c3dcf28e69f2) 看的不是很懂。😳

2017.01.22 14:55  回复

 添加新评论 | 还有1条评论, 展开查看



日了狗 (/u/89172709aaba)

7楼 · 2017.02.16 16:56

(/u/89172709aaba)

第三篇看完顶下

 赞  回复





woniu0936 (/u/0d19c5b5b083)

8楼 · 2017.02.16 23:01

(/u/0d19c5b5b083)

太感谢作者了, android这方面资料的缺失真的让人很头疼, 希望作者能把这一系列的全部写完, 真的真的 非常感谢!

 赞  回复



97f39b674cd3 (/u/97f39b674cd3)


9楼 · 2017.02.17 09:47


(/u/97f39b674cd3)

求大牛后续的文章、

 赞  回复

风从影 (/u/c3dcf28e69f2) : 等等 ~ 最近比较忙 ~

2017.02.17 09:51  回复

 添加新评论





你123哦哦哦 (/u/8fd1d1e14034)

10楼 · 2017.02.21 15:10

(/u/8fd1d1e14034)
期待作者的下一篇文章

👍 赞 💬 回复



白点黑羽 (/u/23e1075b2151)

11楼 · 2017.02.22 10:55

(/u/23e1075b2151)
很详细 学习到了很多

👍 赞 💬 回复



山木19900726 (/u/1cb8ee49b944)

12楼 · 2017.02.28 11:09

(/u/1cb8ee49b944)
坐等跟新，真的好辛苦

👍 赞 💬 回复

风从影 (/u/c3dcf28e69f2)：@山木19900726 (/users/1cb8ee49b944) 最近太忙了，过些时候更新

2017.02.28 13:06 💬 回复

好文章_feiyun (/u/8a84721f4f19)：坐等更新，谢谢楼主

2017.03.01 18:04 💬 回复

✍️ 添加新评论



CoorChice (/u/cfec7d70bbec)

15楼 · 2017.03.14 08:57

(/u/cfec7d70bbec)
期待更新👍

👍 赞 💬 回复





熊熊熊孩子 (/u/daba5093f75e)

16楼 · 2017.03.20 16:39

(/u/daba5093f75e)

楼主借鉴了你的文章写了一篇blog

赞 回复

风从影 (/u/c3dcf28e69f2) : 你表明来源了吗?

2017.03.20 17:15 回复

熊熊熊孩子 (/u/daba5093f75e) : @风从影 (/users/c3dcf28e69f2) 已经加了链接了

2017.03.21 08:35 回复

添加新评论

1

2

下一页

被以下专题收入, 发现更多相似内容



Android知识 (/c/3fde3b545a35?utm_source=desktop&utm_medium=notes-included-collection)



Android开发 (/c/d1591c322c89?utm_source=desktop&utm_medium=notes-included-collection)



首页投稿 (/c/bDHhpK?utm_source=desktop&utm_medium=notes-included-collection)



今日看点 (/c/3sT4qY?utm_source=desktop&utm_medium=notes-included-collection)



程序员 (/c/NEt52a?utm_source=desktop&utm_medium=notes-included-



collection)



手机移动程序开发 (/c/52b64f3155e1?

utm_source=desktop&utm_medium=notes-included-collection)



Android (/c/d76f9327b3a6?utm_source=desktop&utm_medium=notes-included-collection)

展开更多 ∨

