

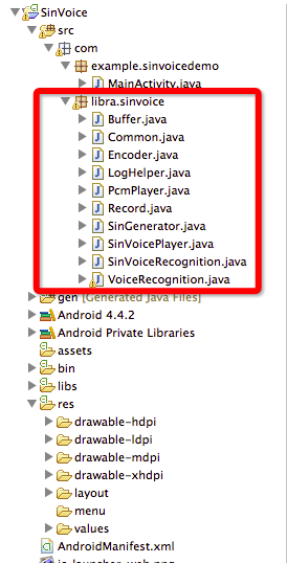


上面说道，如果采样率高，录音的质量就高，那么，是不是采样率越高越好呢？当然不是。随着采样率的提高，虽然质量提高了，但是采样的难度也对应的增加了，而且，采样出来的数据需要存储，采样率越高，产生的数据文件就越大，因此质量高的音乐比一般的音乐体积大。所以，我们通常要选用一个合适的采样率。在信号处理领域有一个定理叫做“采样定理”，也称“奈奎斯特定理”，内容是：如果采样的频率高于信号最高频率的两倍，采样之后的数字信号就可以完整的保留下原始信号中的信息。因为人的听力范围在20HZ-20000HZ，所以一般采样频率在44.1kHZ，也就是一分钟44100次。

在明白了这些预备知识之后，下面开始介绍开源项目SinVoice。



8



1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000

博主专栏



Listview控件使用详解
(<http://blog.csdn.net/column/details/qust-listview.html>)



(<http://blog.csdn.net/column/details/qust-listview.html>)



Android常用自定义控件
详解
(<http://blog.csdn.net/column/details/qust-diy.html>)

20235

在线课程



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0lZ0qnfK9ujYzP1mznWR10Aw-5y9YlZ0lQzq-



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0lZ0qnfK9ujYzP1mznWR10Aw-5y9YlZ0lQzq-



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0lZ0qnfK9ujYzP1mznWR10Aw-5y9YlZ0lQzq-



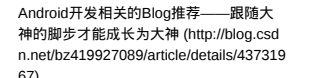
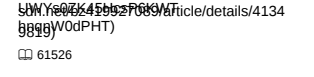
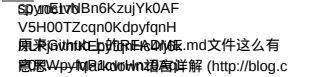
(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0lZ0qnfK9ujYzP1mznWR10Aw-5y9YlZ0lQzq-

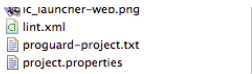


(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0lZ0qnfK9ujYzP1mznWR10Aw-5y9YlZ0lQzq-



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0lZ0qnfK9ujYzP1mznWR10Aw-5y9YlZ0lQzq-





上面是整个项目的结构，圈中的主要的类，下面把几个重要的类的功能和注意点介绍一下。
为了便于理解，我自己添加了一些注释，并不是故意侵占原作者的版权哈。
首先，我们先看一下到底怎么用，下面是MainActivity的代码：

```
[java]
1. package com.example.sinvoicedemo;
2.
3. import android.app.Activity;
4. import android.os.Bundle;
5. import android.os.Handler;
6. import android.os.Message;
7. import android.view.View;
8. import android.view.View.OnClickListener;
9. import android.widget.TextView;
10.
11. import com.libra.sinvoice.LogHelper;
12.
13. import com.libra.sinvoice.SinVoicePlayer;
14. import com.libra.sinvoice.SinVoiceRecognition;
15.
16. /**
17.  * @ClassName: com.example.sinvoicedemo.MainActivity
18.  * @Description: 声波通信
19.  * @author zhaokaiqiang
20.  * @date 2014-11-15 下午12:36:32
21.  */
22.
23. public class MainActivity extends Activity implements
24.     SinVoiceRecognition.Listener, SinVoicePlayer.Listener {
25.
26.     private final static String TAG = "MainActivity";
27.     // 最大数字
28.     private final static int MAX_NUMBER = 5;
29.     // 识别成功
30.     private final static int MSG_SET_RECG_TEXT = 1;
31.     // 开始识别
32.     private final static int MSG_RECG_START = 2;
33.     // 识别结束
34.     private final static int MSG_RECG_END = 3;
35.
36.     private final static String CODEBOOK = "12345";
37.
38.     private Handler mHandler;
39.     // 播放
40.     private SinVoicePlayer mSinVoicePlayer;
41.     // 录音
42.     private SinVoiceRecognition mRecognition;
43.
44.     @Override
45.     protected void onCreate(Bundle savedInstanceState) {
46.         super.onCreate(savedInstanceState);
47.         setContentView(R.layout.activity_main);
48.
49.         mSinVoicePlayer = new SinVoicePlayer(CODEBOOK);
50.         mSinVoicePlayer.setListener(this);
51.
52.         mRecognition = new SinVoiceRecognition(CODEBOOK);
53.         mRecognition.setListener(this);
54.
55.         final TextView playTextView = (TextView) findViewById(R.id.play_text);
```

【Android进阶】android:configChanges
属性总结 (<http://blog.csdn.net/bz419927089/article/details/19921703>)
33323

【凯子哥带你学Framework】Activity启动
过程全解析 (<http://blog.csdn.net/bz419927089/article/details/49428287>)
25017



内容举报

返回顶部

内容举报

```
55.         findViewById(R.id.start_play).setOnClickListener(new OnClickListener() {
56.             @Override
57.             public void onClick(View arg0) {
58.                 String text = genText(15);
59.                 playTextView.setText(text);
60.                 mSinVoicePlayer.play(text);
61.             }
62.         });
63.
64.         // 停止播放声音
65.         findViewById(R.id.stop_play).setOnClickListener(new OnClickListener() {
66.             @Override
67.             public void onClick(View arg0) {
68.                 mSinVoicePlayer.stop();
69.             }
70.         });
71.
72.         // 开始声音识别
73.         findViewById(R.id.start_reg).setOnClickListener(new OnClickListener() {
74.             @Override
75.             public void onClick(View arg0) {
76.                 mRecognition.start();
77.             }
78.         });
79.
80.         // 停止声音识别
81.         findViewById(R.id.stop_reg).setOnClickListener(new OnClickListener() {
82.             @Override
83.             public void onClick(View arg0) {
84.                 mRecognition.stop();
85.             }
86.         });
87.     }
88.
89.     // 获取长度为count且最大值为MAX_NUMBER的随机数
90.     private String genText(int count) {
91.         StringBuilder sb = new StringBuilder();
92.         int pre = 0;
93.         while (count > 0) {
94.             int x = (int) (Math.random() * MAX_NUMBER + 1);
95.             if (Math.abs(x - pre) > 0) {
96.                 sb.append(x);
97.                 --count;
98.                 pre = x;
99.             }
100.        }
101.        return sb.toString();
102.    }
103.
104.    private static class RegHandler extends Handler {
105.
106.        private StringBuilder mTextBuilder = new StringBuilder();
107.        private TextView mRecognisedTextView;
108.
109.        public RegHandler(TextView textView) {
110.            mRecognisedTextView = textView;
111.        }
112.
113.        @Override
114.        public void handleMessage(Message msg) {
115.            switch (msg.what) {
116.                case MSG_SET_RECV_TEXT:
117.            }
118.        }
119.    }
120.
121.}
```





```
122.         char ch = (char) msg.arg1;
123.         mTextBuilder.append(ch);
124.         if (null != mRecognisedTextView) {
125.             mRecognisedTextView.setText(mTextBuilder.toString());
126.         }
127.         break;
128.
129.     case MSG_REC_START:
130.         mTextBuilder.delete(0, mTextBuilder.length());
131.         break;
132.
133.     case MSG_REC_END:
134.         LogHelper.d(TAG, "recognition end");
135.         break;
136.     }
137. }
138.
139. @Override
140. public void onRecognitionStart() {
141.     mHandler.sendMessage(mHandler.obtainMessage(MSG_REC_START));
142. }
143.
144. @Override
145. public void onRecognition(char ch) {
146.     mHandler.sendMessage(mHandler.obtainMessage(MSG_SET_REC_TEXT, ch, 0));
147. }
148.
149. @Override
150. public void onRecognitionEnd() {
151.     mHandler.sendMessage(mHandler.obtainMessage(MSG_REC_END));
152. }
153.
154. @Override
155. public void onPlayStart() {
156.     LogHelper.d(TAG, "start play");
157. }
158.
159. @Override
160. public void onPlayEnd() {
161.     LogHelper.d(TAG, "stop play");
162. }
163.
164. }
165. }
```

我们可以看出，声波播放和识别的代码封装的非常简单易用，我主要强调以下几点

- 1.常量CODEBOOK是一个编码本，因为是这个功能可以商用，因此开源的代码中只给出了使用12345这5个数字进行编码的实例，所以这个常量不要修改。
- 2.SinVoicePlayer和SinVoiceRecognition是两个非常重要的类，前者可以实现将数字转化成单频率的音频进行输出，后者则可以根据音频进行识别。我们可以设置监听器，来监听识别成功的事件回调。
- 3.genText(int count) 方法是为了获取一个长度是count的随机数，而且这个随机数是有要求的，因为示例代码只实现了1到5的编码和解码，因此，生成的随机数必须在1到5之间才能进行正确的编解码，所以使用MAX_NUMBER进行随机数的大小控制

如果只是想简单的使用这个功能，了解上面的知识之后，就完全可以用了，下一篇文章中，我将介绍实现过程中的一些细节问题，下一篇再见。


项目的Github地址：<https://github.com/JesseGu/SinVoice>
(<https://github.com/JesseGu/SinVoice>)



版权声明：本文为博主原创文章，未经博主允许不得转载。


 发表你的评论

(http://my.csdn.net/weixin_35068028)

 qq_25591037 (/qq_25591037) 2016-03-16 10:02 2楼

(/qq_25591037),我把声波频率改成17500之后，相应的采样数算出来是带小数的，要怎么改呢？，谢谢

回复

 qq_25551151 (/qq_25551151) 2015-12-30 17:17 1楼


(/qq_25551151)我想上传图片 文件 或文字 改怎么做呢 谢谢

回复 1条回复

相关文章推荐


【Android开发经验】移动设备的“声波通信/验证”的实现——SinVoice开源项目介绍(三) (http://blog.csdn.net/zhaokaiqiang1992/article/details/41149621)

转载请注明出处：<http://blog.csdn.net/zhaokaiqiang1992> 前两篇介绍了声波验证/通信的原理和声音播放的实现，这一篇将介绍最重要，也是最难懂的东西，就是SinVoice。

 bz419927089 (<http://blog.csdn.net/bz419927089>) 2014年11月16日 19:50 4472

android自定义控件：根据声音波动的声波控件 (http://blog.csdn.net/carlos1992/article/details/41149621)

直接上图，有两个效果，如下：效果一：效果二：大家可能觉得效果二难看，但其实，在项目中的运行以后，是这样的：因为这个控件的许多东西，...

 carlos1992 (<http://blog.csdn.net/carlos1992>) 2016年03月07日 11:40 7005




程序员该学Python吗？90%的码农这么说！

日前我们被一条消息刷屏：我国的程序员更爱Python。一石激起千层浪，身边的程序员对待 Python 究竟是何种态度？他们是这么说...

(http://www.baidu.com/cb.php?c=lgF_pyfqhHmknjnvPjc0lZ0qnfk9ujYzP1f4PjDs0Aw-5Hc3rHnYnHb0TAq15HfLPWRznjb0T1Y4PyFWnjKBuyDdPj0YmHR30AwY5HDdnHc3rj0vrHc0lgF_5y9YlZ0IQzq-uZR8mLPbUB48ugfElAqspynETZ-YpAq8nWqdlAdxTvqdThP-5yF_UvTknOKzujYk0AFV5H00TZcqn0KdpYfqhHRLPjnvnfKEpyfqhHc4rj6kP0KWpyfqP1civrHnz0AQLUWYs0ZK45HcsP6KWThnqPjTdPj6)

Android 录音声波图 (http://blog.csdn.net/aikongmeng/article/details/50427678)

图像类:package com.akm.test;/** * Created by toge on 15/12/9. */ import android.content.Context; impo...

 aikongmeng (<http://blog.csdn.net/aikongmeng>) 2015年12月29日 16:42 3558

android自定义控件：根据声音波动的声波控件模仿Siri (http://blog.csdn.net/koubi1234/article/details/41149621)

 内容举报

 返回顶部




 8







转载于。。。。。。http://blog.csdn.net/carlos1992/article/details/50817997

 koubi1234 (http://blog.csdn.net/koubi1234)2017年03月17日 15:51👤1230



Delphi7高级应用开发随书源码 (http://download.csdn.net/download/chenx...

<http://download.csdn.net/download/chenx...>2003年04月30日 00:00676KB下载




一学就会的 WordPress 实战课

学习完本课程可以掌握基本的 WordPress 的开发能力，后续可以根据需要开发适合自己的主题、插件，打造最个性的 WordPress 站点。

(http://www.baidu.com/cb.php?c=lgF_pyfqnHmknjfvP1m0lZ0qnfK9ujYzP1f4Pjnz0Aw-5Hc4nj6vPjm0TAq15Hf4rjn1n1b0T1Y4njnkPjwbP1N-nyfzrH6z0AwY5HDdnHc3rj0vrHc0lgF_5y9YlZ0lQzqMpgwBUvqoQhP8QvIGIAPCmgfEmvq_lyd8Q1N9nHmvnj7hnHPW5HDknWw9mhkEusKzujYk0AFV5H00TZcqn0KdpyfqnHRLPjnvnfKEpyfqnHnsnj0YnsKWpyfqP1cvrHnz0AqLUWYs0ZK...


实现Android支付宝声波支付时的波纹视图 (http://blog.csdn.net/bboyfeiyu/article/details/42...

转载请注明出处，本文来自【 Mr.Simple 的博客 】。我正在参加博客之星，点击这里投我一票吧，谢谢~前言 自从支付宝声波支付的波纹效果出来以后，这种形式就慢慢流行开来，比如各种安全软件在扫描时会...

 bboyfeiyu (http://blog.csdn.net/bboyfeiyu)2015年01月09日 11:37👤10329


3分钟为你的应用添加声波通讯功能 (http://blog.csdn.net/softlgh/article/details/40507623)

准确性95%以上（保守一点，其实一般是不会出错的）。接口非常简单，有完整的示例，3分钟就可以让你的应用增加声波通讯功能 抗干扰性强，基本上无论外界怎么干扰，信号都是准确的 可自己任意调整通信频段，支...

 softlgh (http://blog.csdn.net/softlgh)2014年10月27日 14:56👤12420

声波配网：通过特点的声波序列将wifi密码发到无屏幕的设备上 (http://blog.csdn.net/qq_1506...

声波配网，即通过手机发出声波，将ssid、password等信息传给设备的一种配网方式。适用于没有触屏或触屏较小不易于信息输入，但是拥有麦克风的智能设备，如智能音箱、智能家庭助手等。其优点是配网速度快...

 qq_15062527 (http://blog.csdn.net/qq_15062527)2017年08月31日 17:12👤552

声波通信原理及源代码 (http://blog.csdn.net/mirkerson/article/details/38305511)

声波通信原理及源代码 分类： 声波通信 声音传输 android 源代码2013-10-17 11:28 2965人阅读 评论(15) 收藏 举报 声波通信声音传输android源...

 mirkerson (http://blog.csdn.net/mirkerson)2014年07月30日 19:42👤12937

手机的超声波通信---对于特殊声音信号的相关资料 (http://blog.csdn.net/vivien_shaw/article/...

目的：若使用超声波信号进行定位，使用笔记本电脑产生超声波信号，要求这种信号不易被人察觉，并且方便接收端的检测，在手机这种便携式硬件上完成超声波信号的接收。 存在的问题：如何让电脑连续...




 内容举报

 返回顶部


 内容举报

 返回顶部

vivien_shaw (http://blog.csdn.net/vivien_shaw)2016年03月02日 10:18👤1065


为您的Android，iOS等应用添加声波传输功能 (<http://blog.csdn.net/x32sky/article/details/4...>)

记得12年左右的时候，美国出现了chirp应用，该应用可以使用声波在iphone手机间传输文本，图片、甚至视频，当时觉得非常高大上。再后来，到13年的时候国内也出现了很多声波应用，比如支付宝的声波支付...

x32sky (<http://blog.csdn.net/x32sky>)2015年01月03日 12:22👤4634


移动设备的“声波通信/验证”的实现——SinVoice开源项目介绍(一) (<http://blog.csdn.net/rsgzs...>)

转载请注明出处：<http://blog.csdn.net/zhaokaiqiang1992> 在APP市场上，经常有一些充满新意的应用让我们眼前一亮，比如微信的面对面加好友，支付宝的声...

rsgzs (<http://blog.csdn.net/rsgzs>)2015年12月12日 23:59👤793


移动设备的“声波通信/验证”的实现——SinVoice开源项目介绍(二) (<http://blog.csdn.net/rsgzs...>)

转载请注明出处：<http://blog.csdn.net/zhaokaiqiang1992> 在上一篇文章中，我们介绍了声波通信/验证的原理和基本使用，这一篇，我们将就一些细节进行谈...

rsgzs (<http://blog.csdn.net/rsgzs>)2015年12月12日 23:59👤533


移动设备的“声波通信/验证”的实现——SinVoice开源项目介绍(三) (<http://blog.csdn.net/rsgzs...>)

转载请注明出处：<http://blog.csdn.net/zhaokaiqiang1992> 前两篇介绍了声波验证/通信的原理和声音播放的实现，这一篇将介绍最重要，也是最难懂的东西，就...

rsgzs (<http://blog.csdn.net/rsgzs>)2015年12月13日 00:00👤543


移动设备的“声波通信/验证”的实现——SinVoice开源项目介绍(二) (<http://blog.csdn.net/dzc37...>)

在上一篇文章中，我们介绍了声波通信/验证的原理和基本使用，这一篇，我们将就一些细节进行谈论。 再来一张项目的结构图 SinVoicePlayer类是我...

dzc372787439 (<http://blog.csdn.net/dzc372787439>)2015年04月08日 15:07👤607


声波通信开源项SinVoice介绍二 (<http://blog.csdn.net/u013286409/article/details/49996365>)

在上一篇文章中，我们介绍了声波通信/验证的原理和基本使用，这一篇，我们将就一些细节进行谈论。 再来一张项目的结构图 SinVoicePlayer类是我...

u013286409 (<http://blog.csdn.net/u013286409>)2015年11月23日 16:00👤992

android开发如何更好的解决适配不同屏幕大小移动设备的问题 (<http://blog.csdn.net/suijing/a...>)

解决这个问题的办法网上有很多，经过不断的查找资料、尝试、寻求帮助后，经过尝试之后初步整理出一种解决方法：理论的东西我就不多说了，主要说一下如何实现的问题 我主要介绍，通过drawable.layout...

suijing (<http://blog.csdn.net/suijing>)2013年12月26日 14:38👤1098



内容举报

返回顶部

