Research and Implementation of Real-Time Voice Control System based on Android

Qian Zhang
Wuhan Qingchuan University, Wuhan, China
329437365@qq.com

Abstract. At present most of the mobile phone users must have to constantly change their mobile phone’s contextual model to adapt to different occasions in every busy life. In order to solve this problem. This paper designed and implemented the real time auto control system of mobile phone’ voice based on Android platform. The paper introduces the sound overall framework of real-time control system around three functional modules: the sound acquisition, getting sound’s loudness and adjusting ring and expounded key technology of the system in the implementation process. The result of experiment shows that the system can change ringtone loudness on the degree of noisy environment in real time and the system run reliably, meanwhile it has certain applicability and practical value.

1. Introduction
The emergence of Android system, in the current era of rapid development of information technology, smart phones have become a necessity of people's daily life. However, when smart phones bring convenience to users, sometimes it brings a lot of inconvenience because of the unexpected ringing or the inability to hear the ringing. Based on this, this paper studies the external sound environment of mobile phone and the internal sound module of mobile phone on the platform of Android operating system, establishes the corresponding relationship between them, and realizes the automatic adjustment of the ringing sound of mobile phone according to this relationship, so as to realize the real-time adjustment of the ringing sound of mobile phone according to the changes of the external environment, and solve the problem that the ringing sound of mobile phone is too loud or too loud in different occasions Small trouble, for the user's life to provide convenience.

2. Android system architecture
Android operating system is the first fully open operating system based on Linux kernel. It consists of four parts as below:
   Application layer. It includes applications such as SMS client, phone dialer, picture browser, web browser and so on.
   Application framework layer. Application framework layer is the foundation of Android development. Many core applications realize their core functions through this layer, which simplifies the reuse of components.
System operation library layer. The system runtime layer can be divided into two parts: system library and Android runtime layer. Among them, system library is the support of application framework and the important link between application framework layer and Linux kernel layer.

Linux kernel layer. Android is based on Linux 2.6 kernel. Its core system services such as security, memory management, process management, network protocol and driver model all depend on Linux kernel.

3. System design

3.1. Overall design framework of the system

Through the analysis of the demand of the system, the application has the following functions: it can make real-time control of the ring tone of the mobile phone according to the noise level of the external environment, so that the mobile phone can automatically increase the volume of the ring tone in a relatively noisy environment, and automatically reduce the volume of the ring tone in a relatively quiet environment.

In order to enable the user to receive the incoming call in a quiet night, the system must also be able to provide the function of timing, so that the user can start or shut down the voice real-time control service according to the need.

According to the above, the system mainly realizes the real-time control function of mobile phone ring tone through the following three steps, “Sound sampling”, “Get loudness”, and “Adjusting ringtone”.

3.2. System modules analysis

Through the analysis in the previous section, the specific functions of the system can be divided into the following modules:

User settings module. Provide some regular settings for the software, for example, the user can set a group of base numbers according to personal characteristics, so as to control various occasions, and the appropriate loudness of the mobile phone ring tone can be automatically adjusted.

Sound monitor. Let the mobile phone get the samples of the voice of the user's external environment in real time.

The obtained sound samples of the external environment are quantized.

Adjusting the ring volume. According to the sound loudness of the quantified external environment and the previously set base number, the ring tone of the mobile phone is adjusted appropriately.

Timing switch. Users can turn on and off the voice real-time control service according to their needs.

The functional modules of the system are shown in Figure 1.
4. Key technologies

sound sampling

Sound sampling is the process of converting analog audio into digital audio. The main equipment used is analog to digital converter (ADC), which corresponds to digital to analog converter (DAC). The process of sampling is to convert the electrical signal of the normal analog audio signal into binary code 0 and 1, which constitute the digital audio file. The higher the sampling frequency, the better the sound quality.

According to the sampling theorem, the sampling frequency must be higher than twice the highest recording frequency to avoid distortion. The range of human hearing is 20Hz-20kHz, so the sampling frequency must be at least $20k \times 2 = 40kHz$ to ensure that low-frequency distortion does not occur. The number of sampling bits is the resolution of the sound processed by the sound card. The higher the value, the higher the resolution, and the more realistic the sound is recorded and played back. The sound file in the computer is represented by the numbers 0 and 1. Therefore, the essence of recording on the computer is to convert the analog sound signal into a digital signal. On the contrary, when playing, it is to restore the digital signal to an analog sound signal for output.

The bit of sound card refers to the binary digit of digital sound signal used by sound card when collecting and playing sound files. The bits of the sound card objectively reflect the accuracy of the description of the digital sound signal to the input sound signal. 8 bits represent the 8th power of $2 - 256$, 16 bits represent the 16th power of $2 - 64k$.

Sampling frequency refers to the number of times the recording equipment samples the sound signal in one second. The higher the sampling frequency, the more realistic the sound restoration will be. In today's mainstream sound cards, the sampling frequency is generally divided into 22.05 kHz, 44.1kHz and 48kHz. 22.05 can only achieve the sound quality of FM broadcasting, 44.1kHz is the theoretical CD sound quality limit, 48kHz is more accurate.

For Android system, two classes of MediaRecorder and MediaRecorder are directly provided in the API provided by the application framework layer to realize the recording function. The MediaRecorder class is to read the audio stream of MIC and analyze the data of the stream while recording. Compared with the MediaRecorder class, the MediaRecorder class provides a simpler API, and the MediaRecorder class can directly store the data of MIC to file, and can be encoded (such as AMR, MP3, etc.). The following details the process of sound sampling with MediaRecorder class:

Whether using MediaRecorder class or MediaRecorder class, first of all, you must add the application to the permission.

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

Construct an object of MediaRecorder class according to the API provided by MediaRecorder class, and initialize each property:

```java
mediaRecorder = new MediaRecorder();
mediaRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
mediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
mediaRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
mediaRecorder.setOutputFile("/dev/null");
```

obtaining sound loudness

The sampled data are only discrete values, and the loudness of the acquired sound is represented by a number of binary bits in the computer. This process is called quantification. In general, it will lose some precision to transform the discretized data into binary representation, because the computer can only represent limited values. The magnitude of the quantization level determines the dynamic range of the sound. The 16 bit quantization level represents the range of the sound that the human ear has just heard from the extremely subtle sound to the intolerable huge noise. The binary digits corresponding to the quantization level are called quantization digits, sometimes directly called sampling digits or sampling digits. Obviously, the more quantized digits, the better the sound quality, but the greater the amount of data.
For Android system, AudioRecord and MediaRecorder are directly provided in the API provided by the application framework layer to realize the function of voice quantization [3]. Here's how to get the loudness of the recording through the method provided by the MediaRecorder class.

After setting the properties of the MediaRecorder according to the steps described in 3.1, the maximum instantaneous amplitude can be obtained by calling the method MediaRecorder.Getmaxamplitude(); once a second through the thread.

After the maximum amplitude is obtained, the loudness (in decibels) of the sound can be obtained by using \( DB = 20 \times \log(\text{max Amplitude}) \).

In order to simplify the steps and reduce the overhead of the system, the method provided by the MediaDecorder class is used to sample and obtain the sound loudness in the design process. In this way, we can get the decibel per second, but when we get the decibel of every moment, it is not appropriate to use this value as the basis for judging the environment.

Figure 2 below is the decibel test result obtained by using this software. After multiple tests, the result error is small.

![Test results of the software](image)

**Figure 2.** Test results of the software

4.1. *Adjusting the phone ring*

According to the voice loudness of the quantified external environment and the previously set base number, this application will adjust the ringing tone of the mobile phone appropriately.
In Android system, the volume control is implemented through AudioManager class. The AudioManager class, located in the android.media package, provides access to control volume and ringing mode operations.

The AudioManager instance object can be obtained through the “getSystemService (Context.AUDIO_SERVICE)” method.

The method of “adjustStreamVolume (int streamType, int direction, int flags)” provided by AudioManager class can be used to adjust the volume of ringing tone.

4.2. Timer

The application also provides the timing function for users. Users can set the time period for opening and closing the voice real-time control service according to their needs, which makes users more handy.

In Android development, timers are generally implemented in the following three ways [8].

- Adopt the sleep (long) method of handler and thread;
- Using handler’s postDelayed (runnable, long) method;
- The method of combining handler with timer and TimerTask is adopted.

The first method is the method used in this paper. Handler is mainly used to process received messages. This is only the most important method. Of course, there are other methods in handler for implementation, which are not explained in this article.

Define a handler class to handle the received message.

The code is as follows:

```java
Handler handler = new Handler () {
    public void handleMessage (Message msg) {
        // What to do
        super.handleMessage(msg);  
    }
};
```

Create a new thread class that implements the runnable interface.

The code is as follows:

```java
public class MyThread implements Runnable {
    @Override
    public void run () {
        // TODO Auto-generated method stub
        while (true) {
            try {
                Thread.sleep(10000);  // Thread pause 10 milliseconds
                Message message = new Message ();
                message.what = 1;
                handler.sendMessage(message);  // Send message
                // TODO Auto-generated catch block
                e.printStackTrace();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}
```

Add the following statement where the thread needs to be started:

```java
new Thread (new MyThread ()). start ();
```

5. System implementation

The software is compatible with Android 5.0 and above, and supports screen resolution of 480 * 800 and above; the best screen size is 5 inches. Install APK file on Android smartphone, the software can run perfectly.

Select the spectrum option to view the sound spectrum of the current environment, as shown in Figure 3.
6. Conclusion
In this paper, the sound signal processing based on Android platform is studied, and a real-time control system of mobile phone voice based on Android platform is implemented. The experimental results show that the system is reliable and convenient for mobile phone users, which has certain application value and practical significance.

Reference
[1] Barrena, S., Klotz, L., Landes, V., Page, A., Ying Sun. Designing Android applications with both online and offline voice control of household devices[P]. Bioengineering Conference (NEBEC), 2012 38th Annual Northeast, 2012.
[2] Whitwam, Ryan. Google Now vs. S Voice: What's the best voice control app on Android? (video)[J]. EN, 2012.
[3] Wolf, Alan. Dacor Updates Discovery Wi-Fi Range With Voice Control[J]. TWICE, 2014, 29(23).
[4] Ullah Saleem, Mumtaz Zain, Liu Shuo, Abubaq Mohammad, Mahboob Athar, Madni Hamza Ahmad. Single-Equipment with Multiple-Application for an Automated Robot-Car Control System.[J]. Sensors (Basel, Switzerland), 2019, 19(3).