

iOS的AVFoundation框架提供了基本的音视频播放工具，我们基本上可以靠其中提供的类完成绝大部分的音视频播放任务。但是在音频播放的输出音量的处理上，苹果的策略比较保守。尽管 `AVPlayer` 和 `AVPlayer` 这些类提供了音量调节功能，但这些音量控制属于App级别的控制。好处就是音量调节独立于系统音量，调节大小时不会影响系统音量。但有时候我们可能希望修改系统音量，以免在调节声音的时候，如果系统音量过小，App调节音量效果不明显。一般来说要调节系统音量会有以下方法：

请注意：修改系统音量无法在模拟器上看到效果，必须使用真机调试才能看到效果！

## 使用 `MPVolumeView`

这个方法是苹果官方推荐的方法。`MPVolumeView` 是Media Player Framework中的一个UI组件，直接包含了对系统音量和Airplay设备的音频镜像路由的控制功能。其中包含一个 `MPVolumeSlider` 的subview用来控制音量。这个 `MPVolumeSlider` 是一个私有类，我们无法手动创建此类，但这个类是 `UISlider` 的子类。`MPVolumeView` 的使用很简单，只需要将其加入到一个父视图中，给予父视图合适的大小，再创建 `MPVolumeView` 示例，将其加入到父视图中即可，苹果官方的文档<sup>1</sup>中有示例代码可以参考。

这个方法的缺点如下：

- **UI可定制的程度低。** `MPVolumeView` 只提供了有限的几个方法来定制其中的Slider和Route Button的样式，而且基本上只能靠换图片解决。如果你想把Slider操作换成Button或者其他UI组件，那是不可能的。
- **没有额外的音量控制API。** 目前为止没有发现iOS的公开API中有可以直接操作系统音量的，所以修改系统音量只能使用这个UI组件。

如果还想给UI加入手势操作来控制音量，这种直接使用 `MPVolumeView` 是做不到的，那么有没有什么方法可以绕过这限制呢？办法还是有的。

## 编程实现系统音量调节<sup>2</sup>

上一小节我们提到了 `MPVolumeView` 这个组件中，有一个subview来控制音量，即 `MPVolumeSlider`。其实我们可以通过遍历 `MPVolumeView` 实例的subviews来得到 `MPVolumeSlider` 的实例，从而通过这个UI组件来操作系统音量。

### 通过 `MPVolumeSlider` 的实例来操作系统音量

我们首先通过创建一个 `MPVolumeView`，然后遍历找出 `MPVolumeSlider` 的实例。这个实例提供 `setValue:animated:` 方法来设置系统音量。我们也可以通过 `volumeSlider.value` 这个属性来获取当前的系统音量。具体的代码如下：

```
1  MPVolumeView *volumeView = [[MPVolumeView alloc] init];
2  UISlider* volumeViewSlider = nil;
3  for (UIView *view in [_instance.volumeView subviews]){
4      if ([view.class.description isEqualToString:@"MPVolumeSlider"]){
5          volumeViewSlider = (UISlider*)view;
6          break;
7      }
8  }
9  // retrieve system volume float systemVolume = volumeViewSlider.value;
10 // change system volume, the value is between 0.0f and 1.0f
11 [volumeViewSlider setValue:1.0f animated:NO];
```



```
12 // send UI control event to make the change effect right now.
13 [volumeViewSlider sendActionsForControlEvents:UIControlEventTouchUpInside];
```

上面的代码演示如何获取和修改系统音量，注意音量取值为0到1之间的浮点数。

## 有问题！我不喜欢系统弹出音量提示

上面通过编程的方法可以很完美的调节系统音量，但是每次修改都会弹出系统提示框告知：



有时候这种提示我们未必会需要，那么怎么取消掉这个提示呢？实际上 `MPVolumeView` 没有提供任何接口来调节是否需要显示系统音量提示。但是我们发现一点：当 `MPVolumeView` 处在当前视图的层级之中时，系统就不会显示音量提示。那么事情好办了，我们只要确保两点：

- `MPVolumeView` 视图处在屏幕上看不见的地方，比如某个不透明视图的下方，或者本视图的非可见区域，一个常见的做法就是把该视图的frame设置为区域以外的地方，比如 `volumeView.frame = CGRectMake(-1000, -100, 100, 100);`
- 确保 `MPVolumeView` 视图的hidden属性值为 `NO`。因为当hidden为 `YES` 时，同样会弹出提示。

## 还有问题，我修改了系统音量但是不是通过我的UI

另一个可能的情况就是用户自己通过硬件的音量调节按钮（位于设备侧边）来调节音量，这种情况会使得你的业务逻辑出现问题，因为你只为自己的App UI写了回调，那么怎么为硬件按钮的事件添加回调呢？我们可以使用Notification Center来完成。

这里只需要监听 `AVSystemController_SystemVolumeDidChangeNotification` 事件即可。具体代码如下：

- 首先在资源载入阶段加入监听事件的代码

```
NSError *error;
// Active audio session before you listen to the volume change event.
// It must be called first.
// The old style code equivalent to the line below is:
//// AudioSessionInitialize(NULL, NULL, NULL, NULL);
// AudioSessionSetActive(YES);
//// Now the code above is deprecated in iOS 7.0, you should use the new
// code here.
[[AVAudioSession sharedInstance] setActive:YES error:&error];
// add event handler, for this example, it is `volumeChange:` method
[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(volumeChanged:) name:
```

- 然后实现事件回调方法

```
- (void)volumeChanged:(NSNotification *)notification
{
    // service logic here.
}
```

- 最后记得在资源回收时取消掉事件监听

```
- (void)dealloc
{
    [[NSNotificationCenter defaultCenter] removeObserver:self name:@"AVSystemController_SystemVolumeChanged" object:nil];
}
```

这样，每次用户使用硬件按钮调节音量的时候也会执行你写好的逻辑。

以上除了第一个方案以外，所有的解决方案都属于非官方的hack性质的方法，但是都没有调用私有API，所以没有被Apple审核拒掉的风险。