# 一、Gradle引用：

Add it in your root build.gradle at the end of repositories:

**allprojects {**

**repositories {**

**...**

**maven { url 'https://jitpack.io' }**

**}**

**}**

**Step 2.** Add the dependency

**dependencies {**

**implementation 'com.github.ahuangshang:Bluetoothlib:1.0.1'**

**}**

# 二、使用

**1、在合适的地方进行初始化，例如: Application onCreate里面执行以下代码** BTClient.init(this);  
   
**2、在需要读卡的页面加入读卡的代码，示例如下：（详细使用见demo，当前页面关闭的时候记得释放资源）** private void startReadBlueCard() {  
 new Thread(new Runnable() {  
  
  
 @Override  
 public void run() {  
 timer = new Timer();  
 timerTask = new TimerTask() {  
 @Override  
 public void run() {  
 read();  
 }  
 };  
 timer.schedule(timerTask, 0, 200);  
 }  
 }).start();  
 }  
   
 private void read() {  
 ArrayList<String> rfidCardNumList = RFIDReaderUtil.getRFIDCardNum();  
 if (rfidCardNumList == null) return;  
 for (String anUid : rfidCardNumList) {  
 //因为卡号被读取的频率是5次/秒（默认频率，可调），但不能一张卡的卡号被读取了五次就记录五次卡号，这样不符合逻辑。  
 //正确的逻辑是：在当前页面的生命周期内，同一张卡的卡号只能被记录一次。  
 boolean isHaveSameDiscernSuccessUserId = false;  
 if (discernSuccessUserIdList.size() > 0) {  
 for (int i = 0; i < discernSuccessUserIdList.size(); i++) {  
 if (anUid.equals(discernSuccessUserIdList.get(i).getUserId())) {  
 isHaveSameDiscernSuccessUserId = true;  
 }  
 }  
 }  
 if (!isHaveSameDiscernSuccessUserId) {  
 //这里可以通过判断卡号的位数来确认是否是智芯卡，如果不在支持老卡，可以在这里处理  
 UserCardModel userCardModel = new UserCardModel();  
 userCardModel.setUserId(anUid);  
 discernSuccessUserIdList.add(userCardModel);  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 userCardAdapter.notifyDataSetChanged();  
 SoundPlayUtils.play(SoundPlayUtils.getSoundID());  
 }  
 });  
 }  
  
 }  
 }  
**3、读到卡号后有个音效播放，lib中默认内置了一个音效，支持更改音效，更改方法：** （1）在自己项目中加入想要播放的音效  
 （2）在读卡页面读卡前调用  
 SoundPlayUtils.load(Context context,int resId)方法，传入要播放的音效  
   
**4、蓝牙设备选择界面可以参照lib中界面自己重新定义**