**参考1、Android使用AccessibilityService实现USB扫码枪数据抓取**

android单屏机，通过扫码枪扫描二维码的场景非常多，扫码枪的种类也有蓝牙、USB、串口等等

目前USB的扫码枪主流的就是以下两种

**1、USB HID-KBW：扫码器会将扫描出来的内容转化为键盘事件，就是Android中KeyEvent里面对应的常量(KeyEvent.KEYCODE\_\*)。**

**2、USB 虚拟串口：可使用[android-serialport-api](https://github.com/cepr/android-serialport-api" \t "_blank) 连接到UsbDevice进行通信，读取数据。(设备要支持串口)**

支持 Android 热插拔USB扫描枪会在有EditText时，扫描枪扫描内容自动输入到编辑框了，但是有很多输入法兼容的问题，比如搜狗输入法识别到HID设备时会隐藏无法弹出，如果输入法切换成中文时会输入中文等等。

通过串口的方式直接获取原始数据，不再跟输入法产生冲突，可惜设备是USB HID的，通过大量的尝试（包括USB虚拟串口）都不支持（对串口不了解的同学可以先看看这篇文章[上半年最好的Android串口开发入门指南 - 简书](https://www.jianshu.com/p/6f650481dde0" \t "_blank) ）

扫码枪是基于键盘输入的，尝试从获取焦点的Activity中的dispatchKeyEvent(KeyEvent event)进行拦截，可惜只能解决掉中文的问题，事件还是先走到输入法才能回到Activity。于是强大的AccessibilityService就上场了，使用AccessibilityService可以优先获取到键盘事件。

使用强大的AccessibilityService（Google为了让Android系统更实用,为用户提供了无障碍辅助服务），但需要到**系统设置->无障碍->服务** 开启当前服务。对AccessibilityService不了解的同学看看[http://www.jianshu.com/p/4cd8c109cdfb](https://www.jianshu.com/p/4cd8c109cdfb" \t "_blank)

废话不多说看实现步骤

1、先创建扫码Service直接继承AccessibilityService就OK

public class ScanService extends AccessibilityService {

private static OnKeyEvent onKeyEvent;

@Override

public int onStartCommand(Intent intent, int flags, int startId) {

return START\_STICKY;

}

@Override

public void onAccessibilityEvent(AccessibilityEvent event) {

}

@Override

public void onInterrupt() {

}

@Override

protected boolean onKeyEvent(KeyEvent event) {

if(onKeyEvent!=null){

//这里通过回调的方式将事件传出去统一处理

//返回true事件就会拦截不会继续传递

return onKeyEvent.onKeyEvent(event);

}

return super.onKeyEvent(event);

}

/\*\*

\* 设置监听

\* @param onKeyEvent

\*/

public static void setOnKeyEvent(OnKeyEvent onKeyEvent){

ScanService.onKeyEvent=onKeyEvent;

}

public interface OnKeyEvent{

boolean onKeyEvent(KeyEvent event);

}

}

2、创建好自己的ScanService后需要在manifest中进行注册

<service

android:name="包名.service.ScanService"

android:enabled="true"

android:exported="true"

android:permission="android.permission.BIND\_ACCESSIBILITY\_SERVICE">

<intent-filter>

<action android:name="android.accessibilityservice.AccessibilityService"/>

</intent-filter>

<meta-data

android:name="android.accessibilityservice"

android:resource="@xml/accessibility"

/>

</service>

创建android:resource需要用到的xml ，在res下新建xml文件夹，新建accessibility.xml

<?xml version="1.0" encoding="utf-8"?>

<accessibility-service xmlns:android="http://schemas.android.com/apk/res/android"

android:accessibilityEventTypes="typeAllMask"

android:accessibilityFeedbackType="feedbackGeneric"

android:accessibilityFlags="flagRequestFilterKeyEvents"

android:canRetrieveWindowContent="true"

android:notificationTimeout="100"

android:canRequestFilterKeyEvents="true"

android:description="@string/accessibility\_description"

android:packageNames="包名" />

android:description指定一个String作为描述文案

<string name="accessibility\_description">这里是描述辅助功能的文案</string>

到此为止AccessibilityService就配置好了，你的应用就会出现在**系统设置->辅助功能**列表里，只需要手动在设置中打开辅助功能，扫码枪的键盘事件就会触发ScanService的onKeyEvent

接下来是对事件的处理  
1、过滤非扫码枪的设备

/\*\*

\* 检测输入设备是否是扫码器

\*

\* @param context

\* @return 是的话返回true，否则返回false

\*/

public boolean isInputFromScanner(Context context, KeyEvent event) {

if (event.getDevice() == null) {

return false;

}

// event.getDevice().getControllerNumber();

if (event.getKeyCode() == KeyEvent.KEYCODE\_BACK || event.getKeyCode() == KeyEvent.KEYCODE\_VOLUME\_DOWN || event.getKeyCode() == KeyEvent.KEYCODE\_VOLUME\_UP) {

//实体按键，若按键为返回、音量加减、返回false

return false;

}

if (event.getDevice().getSources() == (InputDevice.SOURCE\_KEYBOARD | InputDevice.SOURCE\_DPAD | InputDevice.SOURCE\_CLASS\_BUTTON)) {

//虚拟按键返回false

return false;

}

Configuration cfg = context.getResources().getConfiguration();

return cfg.keyboard != Configuration.KEYBOARD\_UNDEFINED;

}

2、处理事件

Runnable mScanningFishedRunnable = new Runnable() {

@Override

public void run() {

String code = mStringBufferResult.toString();

//做相应处理....

mStringBufferResult.setLength(0);

}

};

/\*\*

\* 扫码枪事件解析

\*

\* @param event

\*/

public void analysisKeyEvent(KeyEvent event) {

int keyCode = event.getKeyCode();

//字母大小写判断

checkLetterStatus(event);

if (event.getAction() == KeyEvent.ACTION\_DOWN) {

char aChar = getInputCode(event);

// char aChar = (char) event.getUnicodeChar();

if (aChar != 0) {

mStringBufferResult.append(aChar);

}

if (keyCode == KeyEvent.KEYCODE\_ENTER) {

//若为回车键，直接返回

mHandler.removeCallbacks(mScanningFishedRunnable);

mHandler.post(mScanningFishedRunnable);

} else {

//延迟post，若500ms内，有其他事件

mHandler.removeCallbacks(mScanningFishedRunnable);

mHandler.postDelayed(mScanningFishedRunnable, MESSAGE\_DELAY);

}

}

}

//检查shift键

private void checkLetterStatus(KeyEvent event) {

int keyCode = event.getKeyCode();

if (keyCode == KeyEvent.KEYCODE\_SHIFT\_RIGHT || keyCode == KeyEvent.KEYCODE\_SHIFT\_LEFT) {

if (event.getAction() == KeyEvent.ACTION\_DOWN) {

//按着shift键，表示大写

mCaps = true;

} else {

//松开shift键，表示小写

mCaps = false;

}

}

}

//获取扫描内容

private char getInputCode(KeyEvent event) {

int keyCode = event.getKeyCode();

char aChar;

if (keyCode >= KeyEvent.KEYCODE\_A && keyCode <= KeyEvent.KEYCODE\_Z) {

//字母

aChar = (char) ((mCaps ? 'A' : 'a') + keyCode - KeyEvent.KEYCODE\_A);

} else if (keyCode >= KeyEvent.KEYCODE\_0 && keyCode <= KeyEvent.KEYCODE\_9) {

//数字

aChar = (char) ('0' + keyCode - KeyEvent.KEYCODE\_0);

} else if (keyCode == KeyEvent.KEYCODE\_ENTER) {

aChar = 0;

} else {

//其他符号

aChar = (char) event.getUnicodeChar();

}

return aChar;

}

扫描完成，获取扫描的数据后，自己想怎么处理就怎么处理

最后附上一些工具类  
跳转到系统辅助功能页

/\*\*

\* 打开设置-辅助功能页

\* @param context

\*/

public void openAccessibilitySetting(Context context){

context.startActivity(new Intent(Settings.ACTION\_ACCESSIBILITY\_SETTINGS));

}

判断当前应用的辅助功能在设置中是否打开

/\*\*

\*

\* @param context

\* @return true辅助功能开 false辅助功能关

\*/

public boolean isAccessibilitySettingsOn(Context context) {

int accessibilityEnabled = 0;

final String service = context.getPackageName() + "/" + ScanService.class.getCanonicalName();

try {

//获取setting里辅助功能的开启状态

accessibilityEnabled = Settings.Secure.getInt(

context.getApplicationContext().getContentResolver(),

android.provider.Settings.Secure.ACCESSIBILITY\_ENABLED);

} catch (Settings.SettingNotFoundException e) {

}

TextUtils.SimpleStringSplitter mStringColonSplitter = new TextUtils.SimpleStringSplitter(':');

if (accessibilityEnabled == 1) {

//获取辅助功能里所有开启的服务 包名列表

String settingValue = Settings.Secure.getString(

context.getApplicationContext().getContentResolver(),

Settings.Secure.ENABLED\_ACCESSIBILITY\_SERVICES);

if (settingValue != null) {

//转换程集合

mStringColonSplitter.setString(settingValue);

while (mStringColonSplitter.hasNext()) {

String accessibilityService = mStringColonSplitter.next();

//判断当前包名是否在服务集合里

if (accessibilityService.equalsIgnoreCase(service)) {

return true;

}

}

}

}

return false;

}

完成。

# 参考2 Android 获取USB扫描枪扫描数据（无界面）

2017年11月14日 12:19:29 [飞雪连天射白鹿](https://me.csdn.net/zhouyuanjing) 阅读数 9353

Android 平板热插拔USB扫描枪应用场景多，在有界面如EditText时，扫描枪扫描内容自动输入到编辑框了，在没有界面的情况下，怎么拿到扫描枪数据呢？

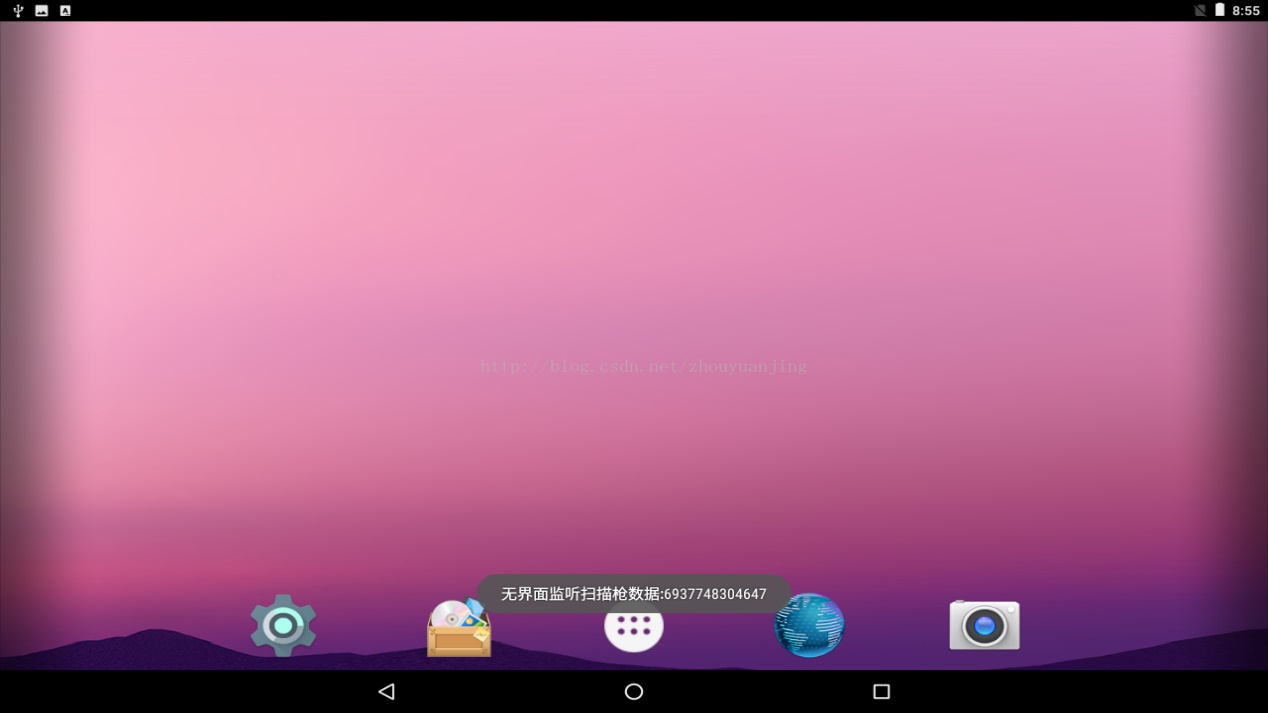
本文类似Android平台抢微信红包思路，使用强大的AccessibilityService（Google为了让Android系统更实用,为用户提供了无障碍辅助服务），但需要到系统设置->无障碍->服务 开启当前服务。

对AccessibilityService不了解的同学看看<http://www.jianshu.com/p/4cd8c109cdfb>

关键：标准扫描枪扫描数据会触发KEYCODE\_ENTER键，感兴趣的同学可以在onKeyEvent函数里打log，看看这个扫描过程中出发的keyCode和条形码（或二维码）内容比较。

处理很简单，ScannerGunService(继承AccessibilityService)，代码如下：

1. package com.android.accessibility;
3. import com.android.accessibility.scanergun.ScannerGun;
4. import com.android.accessibility.scanergun.ScannerGun.ScannerGunCallBack;
6. import android.accessibilityservice.AccessibilityService;
7. import android.content.Intent;
8. import android.text.TextUtils;
9. import android.util.Log;
10. import android.view.KeyEvent;
11. import android.view.accessibility.AccessibilityEvent;
12. import android.widget.Toast;
14. public class ScannerGunService extends AccessibilityService {
16. final static String TAG = "ScannerGunService";
17. private ScannerGun mScannerGun = null;
19. @Override
20. public int onStartCommand(Intent intent, int flags, int startId) {
21. return START\_STICKY;
22. }
24. @Override
25. public void onAccessibilityEvent(AccessibilityEvent arg0) {
26. *// TODO Auto-generated method stub*
28. }
30. @Override
31. public void onInterrupt() {
32. *// TODO Auto-generated method stub*
34. }
36. @Override
37. protected boolean onKeyEvent(KeyEvent event) {
38. *// TODO Auto-generated method stub*
39. if (event.getAction() == KeyEvent.ACTION\_DOWN) {
40. int keyCode = event.getKeyCode();
41. if (keyCode <= 6) {
42. return false;
43. }
44. if (mScannerGun.isMaybeScanning(keyCode, event)) {
45. return true;
46. }
47. }
48. return super.onKeyEvent(event);
49. }
51. @Override
52. public void onCreate() {
53. *// TODO Auto-generated method stub*
54. mScannerGun = new ScannerGun(new ScannerGunCallBack() {
56. @Override
57. public void onScanFinish(String scanResult) {
58. *// TODO Auto-generated method stub*
59. if (!TextUtils.isEmpty(scanResult)) {
60. Toast.makeText(ScannerGunService.this.getBaseContext(),
61. "无界面监听扫描枪数据:" + scanResult, Toast.LENGTH\_SHORT)
62. .show();
63. }
64. }
65. });
66. mScannerGun.setMaxKeysInterval(50);
67. super.onCreate();
68. }
70. }



ScannerGun相关处理就不在这里贴了，有兴趣下载 [ScanGunAccessibility.zip](http://download.csdn.net/download/zhouyuanjing/10117749)

# 参考3 Android USB 扫码枪获取扫描内容

最近做了关于在Android设备上外接扫码的项目，在此记录一下关于Android USB扫码枪获取内容的问题

首先我这边使用是USB HID的扫码枪，即插即用，只需要在界面上有一个带有焦点的EditText ，就可以获取扫码枪扫描的内容。

是不是感觉很简单，但是今天我这里是讲的是在没有EditText 的情况下获取扫码枪扫描的内容。

USB HID扫码枪会将扫描出来的内容转化为键盘事件，对应的就是Android中的KeyEvent事件，所以我们只需要在我们的activity中

重写onKeyDown方法

1. @Override
2. public boolean onKeyDown(int keyCode, KeyEvent event) {
3. checkLetterStatus(event);
4. keyCodeToNum(keyCode);
5. if (keyCode == KeyEvent.KEYCODE\_ENTER && event.getAction() == KeyEvent.ACTION\_DOWN) {
6. Log.e("键盘事件", buffer.toString());
7. buffer.delete(0, buffer.length());
8. return true;
9. }
11. return false;
12. }

上面我们说了扫码枪是响应我们的键盘事件，所以当扫码枪每扫描到一个字符的时候，就相当于按下了我们键盘上的对应按键，也就是keyCode，因此我们只需要处理这个keyCode就行了。

下面我把checkLetterStatus()检查是否大小写的方法跟keyCodeToNum（）根据对应的keycode转换对应的数字跟字母

1. *//检查shift键*
2. private void checkLetterStatus(KeyEvent event) {
3. int keyCode = event.getKeyCode();
4. if (keyCode == KeyEvent.KEYCODE\_SHIFT\_RIGHT || keyCode == KeyEvent.KEYCODE\_SHIFT\_LEFT) {
5. if (event.getAction() == KeyEvent.ACTION\_DOWN) {
6. *//按着shift键，表示大写*
7. mCaps = true;
8. } else {
9. *//松开shift键，表示小写*
10. mCaps = false;
11. }
12. }
13. }
14. *//根据keycode得到对应的字母和数字*
15. private void keyCodeToNum(int keycode) {
16. if (keycode >= KeyEvent.KEYCODE\_A && keycode <= KeyEvent.KEYCODE\_Z) {
17. if (mCaps) {
18. buffer.append(map.get(keycode).toUpperCase());
19. } else {
20. buffer.append(map.get(keycode));
21. }
23. } else if ((keycode >= KeyEvent.KEYCODE\_0 && keycode <= KeyEvent.KEYCODE\_9)) {
24. buffer.append(keycode - KeyEvent.KEYCODE\_0);
25. } else {
26. *//暂不处理特殊符号*
27. }
29. }

上面的方法里面有个map，是用来存储字母的

1. Map<Integer, String> map = new HashMap<>();
2. map.put(29, "a");
3. map.put(30, "b");
4. map.put(31, "c");
5. map.put(32, "d");
6. map.put(33, "e");
7. map.put(34, "f");
8. map.put(35, "g");
9. map.put(36, "h");
10. map.put(37, "i");
11. map.put(38, "g");
12. map.put(39, "k");
13. map.put(40, "l");
14. map.put(41, "m");
15. map.put(42, "n");
16. map.put(43, "0");
17. map.put(44, "p");
18. map.put(45, "q");
19. map.put(46, "r");
20. map.put(47, "s");
21. map.put(48, "t");
22. map.put(49, "u");
23. map.put(50, "v");
24. map.put(51, "w");
25. map.put(52, "x");
26. map.put(53, "y");
27. map.put(54, "z");

最后用一个StringBuffer 来接收处理过的数据。大概就这样吧！