

EINKSDK 使用说明

开发部

版本历史

版本/状态	责任人	起止日期	备注
V1.0/正式	Maxeye	2018/04/20	
V1.0.1	Maxeye	2018/05/18	

MAXEYE CONFIDENTIAL FOR HUARUI

目 录

1. EINKSDK 简介	1
2. 工程引用 EINKSDK	1
2.1 拷贝 EINKSDK.AAR 到工程 LIB 目录。	1
2.2 配置 BUILD.GRADLE	2
3. EINKSDK 使用说明	3
3.1 在 APPLICATION 中建立 EINKCLIENT 实例。	3
3.2 实现 EINKCLIENT 回调函数;	4
3.3 初始化 EINKCLIENT	4
3.4 调用 EINKCLIENT API	5
4. EINKSDK API 使用说明	5
4.1 CLASSES	5
4.2 INTERFACES	5
4.3 MATHOD	5
5. FAQ	7
5.1 调用 DRAWPOINTTOEINKVIEW () 没有效果?	7
6. V1.0.1 更新说明	7

1. EINKSDK 简介

EINK SDK 是 MAXEYE 公司为 ANDROID 平台提供的开发工具包（SDK）。EINK SDK 封装了 MAXEYE EINK 手写板软件各种 API，包括连接 Bluetooth 连接，获取数据，和 View 绘图等。用户只需要做少量修改即可快速完成应用开发工作。

2. 工程引用 EINKSDK

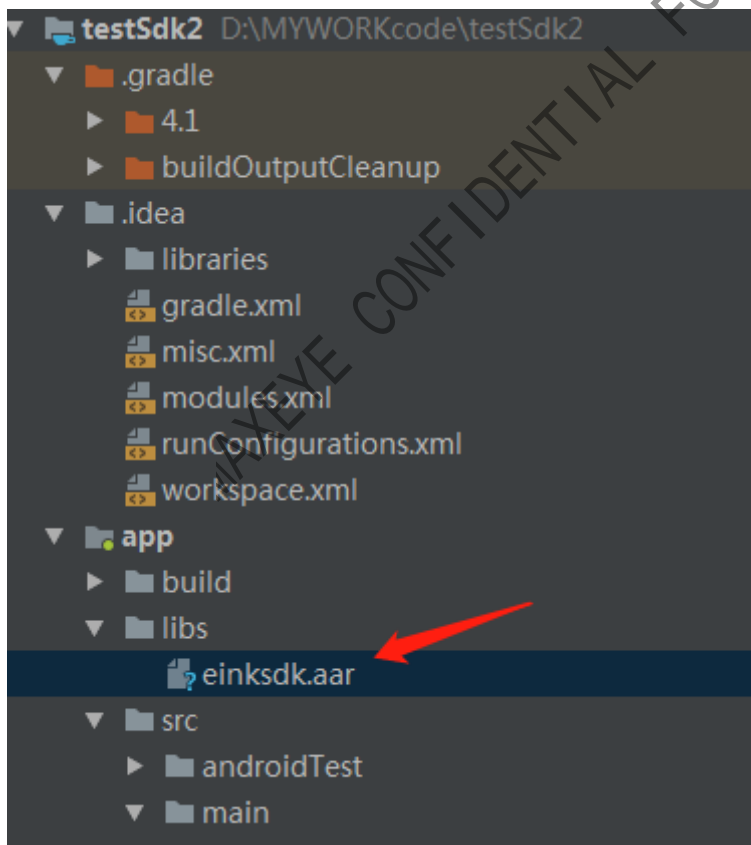
以 Android Studio 3.0.1 版本为例进行说明。主要有以下几个步骤

Step1: 将 einksdk.aar copy 到目标工程的 lib 目录;

Step2: 配置工程 build.gradle;

2.1 拷贝 einksdk.aar 到工程 lib 目录。

然后将 einksdk.aar copy 到工程的 lib 目录。如图:



2.2 配置 build.gradle

1: 设置 minSdkVersion 18

2: 引入以下组件:

```
implementation 'com.polidea.rxandroidble:rxandroidble:1.4.3' //蓝牙连接框架
implementation 'org.greenrobot:eventbus:3.0.0' //事件消息 EventBus
implementation 'org.xutils:xutils:3.5.0' //数据库
implementation fileTree(dir: 'libs', include: ['einksdk.aar']) //einksdk
```

如图:

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 26
    defaultConfig {
        applicationId "com.maxeye.maxeyesdkdemo"
        minSdkVersion 18
        targetSdkVersion 26
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:26.1.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.1'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'

    implementation 'com.polidea.rxandroidble:rxandroidble:1.4.3' //蓝牙连接框架
    implementation 'org.greenrobot:eventbus:3.0.0' //事件消息 EventBus
    implementation 'org.xutils:xutils:3.5.0' //数据库
    implementation fileTree(dir: 'libs', include: ['maxeyelib.aar']) //maxeyelib
}
```

3. EINKSDK 使用说明

在使用 EINKSDK 时，主要有以下五个步骤：

Step1: 在 Application 中建立 EinkClient 实例；

Step2: 在 activity 中实现 EinkClient 回调函数；

Step3: 在 activity 初始化 EinkClient；

Step4: 调用 EinkClient API

3.1 在 Application 中建立 EinkClient 实例。

EinkClient 是用户调用 EinkClient 的接口类，在 Application 中实例化，在各个 Activity 中都可以调用。

参考代码：

MyApplication.java

```
public class MyApplication extends Application {  
    private EinkClient einkClient = null;  
    @Override  
    public void onCreate() {  
        super.onCreate();  
        einkClient = new EinkClient(this);  
    }  
    public EinkClient getEinkClient() {  
        return einkClient;  
    }  
}
```

AndroidManifest.xml 修改：

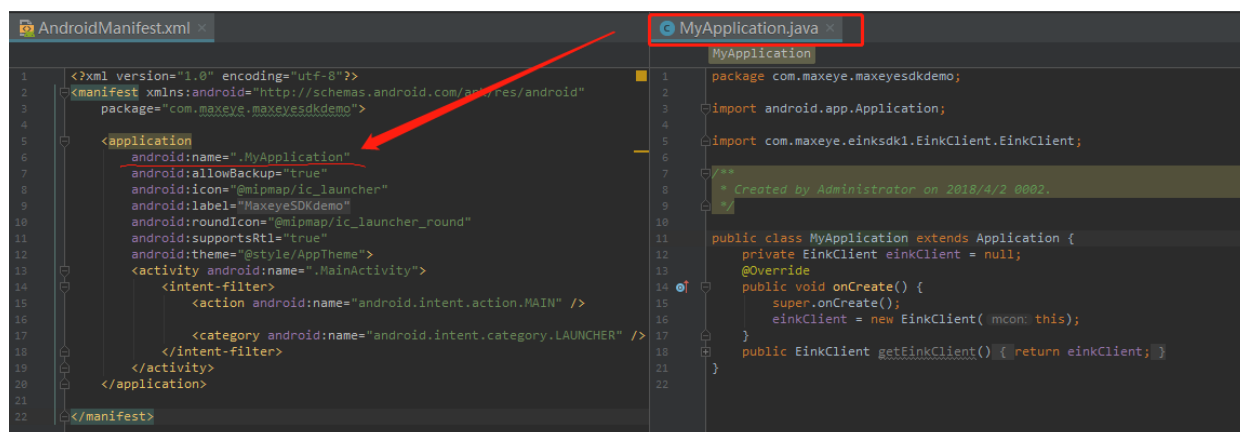
申请权限：

```
<uses-permission android:name="android.permission.BLUETOOTH" />  
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />  
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />  
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

定义属性：

```
<manifest  
    <application android:name=".MyApplication"  
        <!--省略其他。。。。。。-->  
    </application>  
</manifest>
```

如下图所示：



3.2 实现 EinkClient 回调函数；

手写板的相关状态都会通过 EinkClientCallbacks 来通知到应用层。

参考代码：

```
private EinkClient.EinkClientCallbacks callbacks = null;
callbacks = new EinkClient.EinkClientCallbacks() {
    @Override
    public void BluetoothScanResult(RxBleDevice device) {
        // TODO: 2018/4/21 0021 处理 蓝牙搜索 的结果
    }
    @Override
    public void EventBTConnectState(String message) {
        // TODO: 2018/4/21 0021 处理 蓝牙连接 状态
    }
    @Override
    public void UserMessage(EventUserMessage message) {
        // TODO: 2018/4/21 0021 处理 SDK 发布的其他信息，详情请参考附录：消息列表
    }
};
```

3.3 初始化 EinkClient

调用 EinkClient.init()设置相关参数。函数说明如下：

```
/**
 * @param callbacks    消息回调，可以为空
 * @param einkPaintView 手写视图，可以为 null
 * @param mod          开启在线/离线模式
 */
public void Init(EinkClientCallbacks callbacks, EinkPaintView einkPaintView, int mod)
```

在不同的 Activity 中针对不同需求，可以在次调用 init()。

3.4 调用 EinkClient API

根据需求调用 API 完成相关功能。

4. EINKSDK API 使用说明

4.1 Classes

EinkClient()	应用调用 SDK 接口类。在 Application 中实例化。
EinkPaintView	绘制笔记的 View
PageBean	一页笔记数据，此文档统一称其为 page
PointBean	一个点笔记数据，此文档统一称其为 pont

4.2 Interfaces

EinkClinteCallbacks	由用户实现。手写板的相关状态，通知等由此函数通知应用层。请看考 3.2
---------------------	-------------------------------------

4.3 Method

Init()	初始化客户端参考 3.3
SetCallbacks()	设置 EinkClinteCallbacks
SetEinkView()	设置 EinkPaintView
GetAllPageInfo()	Get 所有 page 信息
SetMod()	开启离线/在线模式
CleanOnLineDataOnePage()	清除书写的内容
DelOnePage()	删除一页 page

DrawPointToEinkView()	在 EinkPaintView 中绘制传入的 point
GetEinkView()	取得当前有效的 EinkPaintView
GetPageBeanOneLine()	取得 page 中指定一笔的 所有 point
GetPageInfo()	取得指定 page 的详细信息
UpdateOnePageInfo()	更新指定 page 的详细信息
GetPagePoinDate()	取得 page 的所有 point
SavePicture ()	保存 当前有效的 EinkPaintView，为一张 PNG 图片
SaveOnLineDataOnePage()	保存当前 EinkPaintView 上的 point 信息，在新建 page 时用到。
UpdateOnLineDataOnePage()	保存当前 EinkPaintView 上的 point 信息，在编辑已有 page 时用到。
SqlitOnePage()	将指定的 page 拆分成 2 页 page
StartScanBluetooth()	开始扫描蓝牙设备
StopScanBluetooth()	停止扫描蓝牙设备
StartConnectDevice()	开始连接蓝牙设备
StopConnectDevice()	断开蓝牙连接
IsBluetoothConnected()	判断是否已经连接蓝牙
ReNameBluetoothDevice()	重命名手写板蓝牙设备
GetHPPower()	取得手写电量信息
StartHandWritingUpdate ()	写板固件升级
SavePictureWithBackground()	将 View 导出图片存放到指定的位置。

5. FAQ

5.1 调用 **DrawPointToEinkView**（）没有效果？

请先确认：

1. 已使用 `init（）` 初始化了 `EinkPaintView`。
2. 在 `onWindowFocusChanged()` 中调用此函数绘图。

6. 更新说明

V1.0.1

1. 增加设置数据保存位置 api: `public EinkClient(Context mcon,String path)` 参数 `path` 可以指定保存相关数据的路径。如果为“null”怎默认保存到 `apk getExternalFilesDir()` 对应的目录。
2. 增加手写板固件升级 api: `EinkClient.StartHandWritingUpdate()`
3. 增加导出笔记数据带背景图片 api: `SavePictrueWithBackground()`;
4. 增加蓝牙自动重连功能: `StartConnectDevice(RxBleDevice device,boolean autoConnect)`
 - a) 参数: `autoConnect : true` 当蓝牙异常断开会自动重新连接。
 - b) 参数: `autoConnect : false` 当蓝牙异常断开会不会自动连接。

V1.1.2

1. 增加 `hover` 点显示功能
2. 支持手写板和手指同时书写
3. 初始化函数修改

a) Java

```
public void Init(EinkClientCallbacks callbacks, EinkClientView einkClientView, int mod)
```

a) Xml

```
<com.maxeye.einksdk.EinkClient.EinkClientView
    android:id="@+id/hover_EinkPaintViewLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
/>
```

4. 不推荐使用以下函数，后期会逐步废弃

```
public void Init(EinkClienteCallbacks callbacks, EinkPaintView einkPaintView, int mod)
```

V1.1.3

5. 增加设置分类接口

```
/**
 * 设置页数据分类信息
 * @param pageId      数据页 ID
 * @param strCategory  设置的分裂名称
 * @param set          true  取消此分类  false 设置此分类
 */
public void SetPageDataCategory(int pageId, String strCategory,boolean set){
    dbSerbice.SetPageCategoryInfo(pageId,strCategory,set);
}

/**
 * 去所有分类字符串
 * @return
 */
public List<String> GetCategoryString(){
    return dbSerbice.GetCategoryString();
}

/**
 * 按分类查找页
 * @param category
 * @return
```

```
*/

public List<PageBean> FindPageByCategory(String category){

    return dbSerbice.FindPageByCategory(category);

}

/**

* 查找 一页数据的分类信息

* @param pageId

* @return

*/

public List<String> FindCategoryByPageId(int pageId){

    return dbSerbice.FindCategoryByPageId(pageId);

}

/**

* 提取一页离线数据

*/

public void GetOnePageOfflineData(){

    dbSerbice.SynOfflineData();

}
```