SLAMTEC

Slamware Connect SDK

android_1.0_beta

2016-03-14 Rev. 1

上海思岚科技有限公司
www.slamtec.com

目录

SDK 内容

SlamwareConnectSDK(Android)主要包括 libSlamwareConnectSDK.jar 文件。使用该 SDK 可以对 Slamware 的如下功能进行开发：

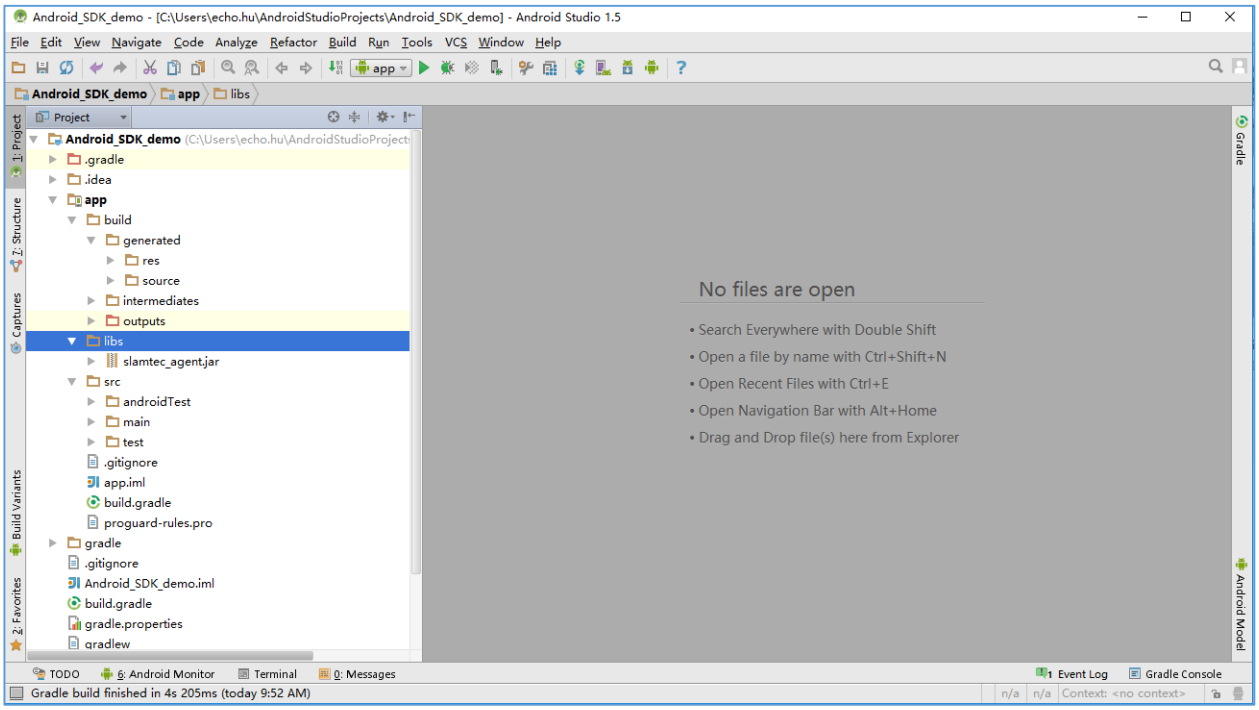
- 通过 WIFI 自动发现设备
- 通过蓝牙对设备进行配置

开发环境要求

要基于 SlamwareConnectSDK 进行应用开发，需要您的开发环境满足如下条件：

- 您的计算机应当安装 Android Studio 1.5, 也可使用最新版本。本文档将以 Android 1.5 为例进行介绍。

a. 在 Android Studio 1.5 中打开 Android SDK 文件;

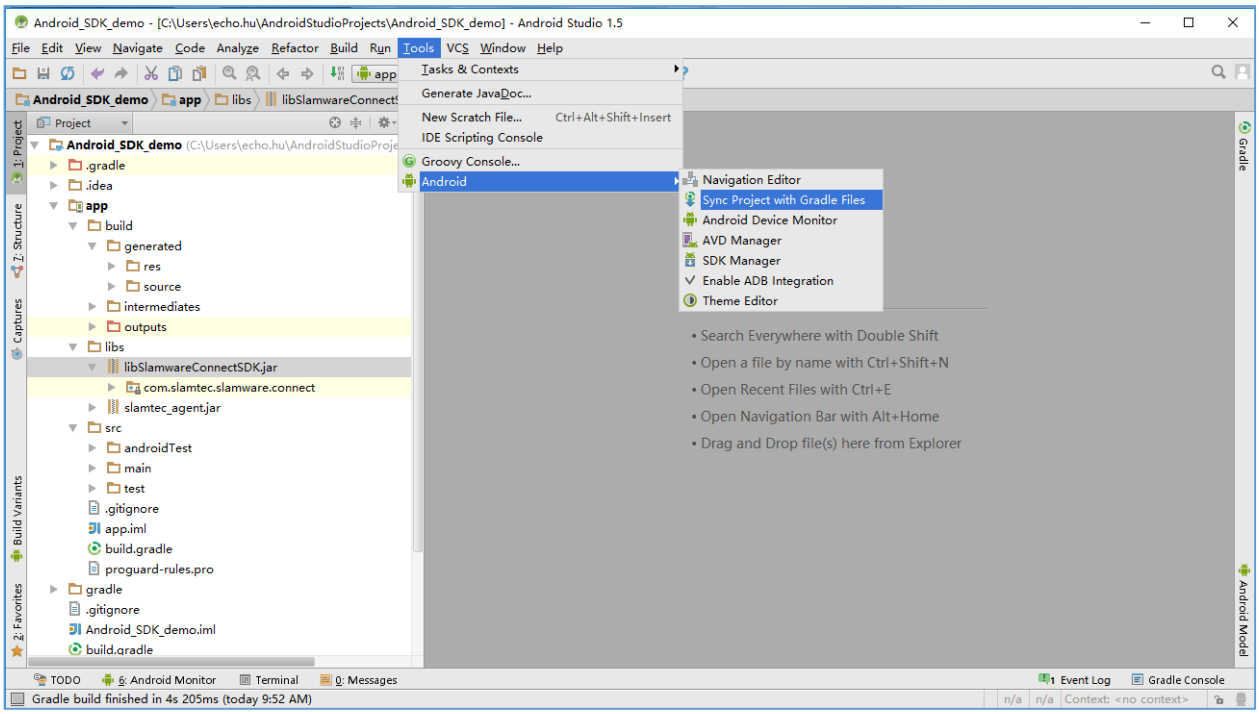


图表 2 创建项目工程步骤 a

b. 将下面的 libSlamwareConnectSDK.jar 文件复制粘贴到步骤 a 中的 libs 文件夹下；

图表 2 创建项目工程步骤 b

c. 粘贴成功后该文件将显示在 libs 文件夹下，请点击工具栏中 Tools>Android>Sync Project with Gradle Files，接下来即可进行开发。



图表 2 创建项目工程步骤 c

概览

对象	说明
AutoDiscovery 类	通过 WIFI 自动发现设备。
AutoDiscoveryDevice 类	自动发现的设备类。
BaseDeviceModel 类	设备基类。
IAutoDiscoveryCallBack 接口	AutoDiscoveryCallBack 接口定义。
ISmartConfigCallBack 接口	IsmartConfigCallBack 接口定义。
SmartConfig 类	通过蓝牙对设备进行配置。

AutoDiscovery 类

概览

通过 WIF 自动发现设备。

常量

```
public static final int STATE_START = 0
public static final int STATE_STOP = 1
public static final int STATE_ERROR_START_FAIL = 10
public static final int STATE_ERROR_STOP_FAIL = 11
public static final int STATE_ERROR_RESOLVE_FAIL = 12
```

构造器

```
public AutoDiscovery(Context var1, IAutoDiscoveryCallback var2)
```

方法

public boolean startDiscovery()

public void stopDiscovery()

public static final int STATE_START = 0

表示启动。

public static final int STATE_STOP = 1

表示终止。

public static final int STATE_ERROR_START_FAIL = 10

表示启动失败。

public static final int STATE_ERROR_STOP_FAIL = 11

表示终止失败。

public static final int STATE_ERROR_RESOLVE_FAIL = 12

表示处理失败。

public AutoDiscovery(Context var1, IAutoDiscoveryCallback var2)

通过 WIF 自动发现设备。

public boolean startDiscovery()

开始发现设备。

public void stopDiscovery()

停止发现设备。

AutoDiscoveryDevice 类

概览

自动发现的设备类。

父类

继承自 BaseDeviceModel。

构造器

```
public AutoDiscoveryDevice(String var1, String var2, String var3, int var4)
```

方法

```
public String getIpAddress()
public int getPort()
```

public AutoDiscoveryDevice(String var1, String var2, String var3, int var4)

自动发现设备。

public String getIpAddress()

获取 IP 地址。

public int getPort()

获取接口。

BaseDeviceModel 类

概览

设备基类。

常量

```
public static final int SOURCE_SMART_CONFIG = 1
public static final int SOURCE_AUTO_DISCOVERY = 2
```

构造器

```
public BaseDeviceModel(String var1, String var2, int var3)
```

方法

```
public String getSdpName()
public String getModel()
public int getSource()
```

public static final int SOURCE_SMART_CONFIG = 1;

表明是 SmartConfig 发现的设备。

public static final int SOURCE_AUTO_DISCOVERY = 2;

表明是 AutoDiscovery 发现的设备。

public BaseDeviceModel(String var1, String var2, int var3)

底盘设备型号。

public String getSdpName()

获取 SDP 名称。

public String getModel()

获取型号。

public int getSource()

获取资源。

IAutoDiscoveryCallBack 接口

概览

接口。

方法

```
void onAutoDiscoveryStateChanged(int var1)
void onAutoDiscoveryFound(AutoDiscoveryDevice var1)
void onAutoDiscoveryLost(AutoDiscoveryDevice var1)
```

void onAutoDiscoveryStateChanged(int var1)

AutoDiscovery 状态改变。

void onAutoDiscoveryFound(AutoDiscoveryDevice var1)

AutoDiscovery 发现。

void onAutoDiscoveryLost(AutoDiscoveryDevice var1)

AutoDiscovery 丢失。

ISmartConfigCallback 接口

概览

接口 。

方法

```
void onSmartConfigFound(BaseDeviceModel var1)
void onSmartConfigStateChanged(int var1)
```

void onSmartConfigFound(BaseDeviceModel var1)

SmartConfig 发现。

void onSmartConfigStateChanged(int var1)

SmartConfig 状态改变。

SmartConfig 类

概览

通过蓝牙对设备进行配置。

常量

```
public static final int SC_FAILED = 0
public static final int SC_SUCCESS = 1
public static final int SC_CONFIGING = 2
public static final int SC_WRONG_SSID_PASSWORD = 3
```

构造器

```
public SmartConfig(Context var1, ISmartConfigCallback var2)
```

方法

```
public void setWifiInfo(String var1, String var2)
public boolean isBluetoothEnabled()
public boolean startScan()
public void stopScan()
public void startConfig(BaseDeviceModel var1)
public boolean supportSmartConfig()
```

public static final int SC_FAILED = 0;

SmartConfig 失败。

public static final int SC_SUCCESS = 1;

SmartConfig 成功。

public static final int SC_CONFIGING = 2;

SmartConfig 正在配置。

public static final int SC_WRONG_SSID_PASSWORD = 3;

SmartConfig 配置时 SSID 或密码不正确。

public SmartConfig(Context var1, ISmartConfigCallback var2)

通过蓝牙对设备进行配置。

public void setWifiInfo(String var1, String var2)

设置 WIFI 信息。var1 为 ssid，var2 为密码。

public boolean isBluetoothEnabled()

蓝牙是否打开。

public boolean startScan()

开始扫描。

public void stopScan()

停止扫描。

public void startConfig(BaseDeviceModel var1)

开始配置。

public boolean supportSmartConfig()

支持配置。

注：调用 SmartConfig 函数之前需要调用 setWifiInfo 函数。

版本历史

日期	版本	描述
2016-03-11	0.1	初始版本

图表索引