**1、扫描蓝牙使用SWDeviceScanManager类**

1.1、使用SWDeviceScanManager类来进行扫描BLE蓝牙设备，获取SWDeviceScanManager实例对象，

scanManager = **new** SWDeviceScanManager(**this**)

1.2、设置扫描蓝牙后的回调scanManager.setScanCallBack(**new** DeviceScanCallBack｛

@Override

**public** **void** onLeScan(BluetoothDevice device, **int** rssi, **byte**[] scanRecord) {

**final** BluetoothDevice device2 = device;

runOnUiThread(**new** Runnable() {

@Override

**public** **void** run() {

mLeDeviceListAdapter.addDevice(device2);

mLeDeviceListAdapter.notifyDataSetChanged();

}

});

}

｝);

1.3、开始扫描蓝牙设备，scanManager.startScan();停止扫描使用scanManager.stopScan();

**2、连接蓝牙或移动设备和蓝牙进行通讯使用SWDevice类**

2.1、获取SWDevice实例对象，swDevice = **new** SWDevice([**Context**](eclipse-javadoc:%E2%98%82=MyTag/D:%5C/Android%20Project%5C/MyTag%5C/libs%5C/MyTagSDK.jar%3Ccom.comime.swdevice(SWDevice.class%E2%98%83SWDevice~SWDevice~Landroid.content.Context;~Ljava.lang.String;~Landroid.bluetooth.BluetoothDevice;~Lcom.comime.swdevice.DeviceListener;%E2%98%82Context) context, [**String**](eclipse-javadoc:%E2%98%82=MyTag/D:%5C/Android%20Project%5C/MyTag%5C/libs%5C/MyTagSDK.jar%3Ccom.comime.swdevice(SWDevice.class%E2%98%83SWDevice~SWDevice~Landroid.content.Context;~Ljava.lang.String;~Landroid.bluetooth.BluetoothDevice;~Lcom.comime.swdevice.DeviceListener;%E2%98%82String) tag,[**BluetoothDevice**](eclipse-javadoc:%E2%98%82=MyTag/D:%5C/Android%20Project%5C/MyTag%5C/libs%5C/MyTagSDK.jar%3Ccom.comime.swdevice(SWDevice.class%E2%98%83SWDevice~SWDevice~Landroid.content.Context;~Ljava.lang.String;~Landroid.bluetooth.BluetoothDevice;~Lcom.comime.swdevice.DeviceListener;%E2%98%82BluetoothDevice) device, [**DeviceListener**](eclipse-javadoc:%E2%98%82=MyTag/D:%5C/Android%20Project%5C/MyTag%5C/libs%5C/MyTagSDK.jar%3Ccom.comime.swdevice(SWDevice.class%E2%98%83SWDevice~SWDevice~Landroid.content.Context;~Ljava.lang.String;~Landroid.bluetooth.BluetoothDevice;~Lcom.comime.swdevice.DeviceListener;%E2%98%82DeviceListener)deviceListener);例如MyTagexample中的swDevice = **new** SWDevice(DeviceScanActivity.**this**, device.getAddress(),

device, DeviceScanActivity.**this**);

或者使用swDevice = **new** SWDevice([**Context**](eclipse-javadoc:%E2%98%82=MyTag/D:%5C/Android%20Project%5C/MyTag%5C/libs%5C/MyTagSDK.jar%3Ccom.comime.swdevice(SWDevice.class%E2%98%83SWDevice~SWDevice~Landroid.content.Context;~Ljava.lang.String;~Landroid.bluetooth.BluetoothDevice;~Lcom.comime.swdevice.DeviceListener;%E2%98%82Context) context, [**String**](eclipse-javadoc:%E2%98%82=MyTag/D:%5C/Android%20Project%5C/MyTag%5C/libs%5C/MyTagSDK.jar%3Ccom.comime.swdevice(SWDevice.class%E2%98%83SWDevice~SWDevice~Landroid.content.Context;~Ljava.lang.String;~Landroid.bluetooth.BluetoothDevice;~Lcom.comime.swdevice.DeviceListener;%E2%98%82String) tag);

然后swDevice.setBluetoothDevice([**BluetoothDevice**](eclipse-javadoc:%E2%98%82=MyTag/D:%5C/Android%20Project%5C/MyTag%5C/libs%5C/MyTagSDK.jar%3Ccom.comime.swdevice(SWDevice.class%E2%98%83SWDevice~SWDevice~Landroid.content.Context;~Ljava.lang.String;~Landroid.bluetooth.BluetoothDevice;~Lcom.comime.swdevice.DeviceListener;%E2%98%82BluetoothDevice) bluetoothDevice);

swDevice.setDeviceListener([**DeviceListener**](eclipse-javadoc:%E2%98%82=MyTag/D:%5C/Android%20Project%5C/MyTag%5C/libs%5C/MyTagSDK.jar%3Ccom.comime.swdevice(SWDevice.class%E2%98%83SWDevice~SWDevice~Landroid.content.Context;~Ljava.lang.String;~Landroid.bluetooth.BluetoothDevice;~Lcom.comime.swdevice.DeviceListener;%E2%98%82DeviceListener)deviceListener);

2.2连接蓝牙使用swDevice.connectGatt();断开重连使用swDevice.connect();

2.3 通过实现[**DeviceListener**](eclipse-javadoc:%E2%98%82=MyTag/D:%5C/Android%20Project%5C/MyTag%5C/libs%5C/MyTagSDK.jar%3Ccom.comime.swdevice(SWDevice.class%E2%98%83SWDevice~SWDevice~Landroid.content.Context;~Ljava.lang.String;~Landroid.bluetooth.BluetoothDevice;~Lcom.comime.swdevice.DeviceListener;%E2%98%82DeviceListener)接口，可以获得蓝牙一些通讯操作的回调方法，例如**public void** onConnected(String tag, BluetoothDevice device)，蓝牙设备连接成功会调用该方法; **public void** onDisconnected (String tag, BluetoothDevice device)，蓝牙设备连接失败或断开连接会调用该方法;使用swDevice.readRssi()方法后;会执行**public void** onGetRssi(String tag, **int** rssi, BluetoothDevice device)，我们可以得到rssi; **public void** onGetValue(String tag, **byte**[] value, BluetoothDevice device),蓝牙往APP发送数据时会调用该回调方法，我们可以通过value来获取到相对应的数据; **public void** onWriteSuccess(String tag, **byte**[] value, BluetoothDevice device)，APP给蓝牙设备写数据时，如果写入成功，将会调用该回调方法，我们可以通过value来查看什么数据写入成功;

2.4、使用swDevice.startRing()给蓝牙设备发送报警命令；使用swDevice.stopRing()给蓝牙设备发送停止报警命令；使用swDevice.closeBluetoothDevice()可以关闭蓝牙设备；