

青岛海信医疗设备股份有限公司	文档版本	密 级
	所属范围	共 页

海信医疗终端适配指导书

拟制:	邹彦飞	日期:	2016-12-23
Prepared by	_____	Date	_____
审核:	_____	日期:	_____
Reviewed by	_____	Date	_____
审核:	_____	日期:	_____
Reviewed by	_____	Date	_____
批准:	_____	日期:	_____
Granted by	_____	Date	_____

青岛海信医疗设备股份有限公司

版权所有 侵权必究

修订记录 Revision record

日期 Date	修订版本 Revision version	描述 Description	作者 Author	审核 Reviewer
2016-12-23	V1.00	海信医疗终端适配指导书	邹彦飞	林文

目次

目次.....3

海信医疗终端适配指导书.....1

1 简介1

2 概述1

3 硬件连接和手机准备.....1

 3.1 驱动安装.....1

 3.2 设置调试.....2

4 适配接口说明4

 4.1 获取扫描结果.....4

 4.2 获取指纹绑定账号4

 4.3 通过解锁广播启动第三方应用4

 4.4 第三方应用动态获取指纹绑定账号5

 4.5 网络查看.....7

5 错误原因和现象描述.....7

 5.1 ADB 服务未启动.....7

海信医疗终端适配指导书

1 简介

海信医疗终端依托医院现有HIS系统，通过移动护理APP，将医护业务整合延伸到移动手持终端，形成一个实时、动态的工作平台。医护人员通过手持终端随时随地采集、查询、核对、录入医嘱信息或患者信息，利用4G/WIFI无线网络进行信息的传输或共享，方便快捷、安全可靠，是简化医疗流程、提高医疗效率、保障医疗安全的有效措施。

2 概述

该指导书用于指导客户在海信医疗终端上进行各类功能的适配，包括扫描功能、指纹功能、网络查看功能等，通过适配，简化操作过程，提升用户体验，提供更有竞争力的产品。

3 硬件连接和手机准备



图 1. 计算机与海信移动医疗终端硬件连线图

与计算机连接的海信移动医疗终端需要在设置允许连接的计算机进行调试。需要先在计算机上安装手机终端的驱动程序，然后打开“调试”功能，计算机的“硬件管理”出现ADB端口时就可以正常使用了。

3.1 驱动安装



图 2 手机驱动安装选择界面

当USB线将终端与计算机连接后，终端界面会弹出提示，选择“安装光盘”，计算机会出现一个光驱盘符，打开盘符可以找到Setup.exe程序，双击运行。依图操作，如下图：

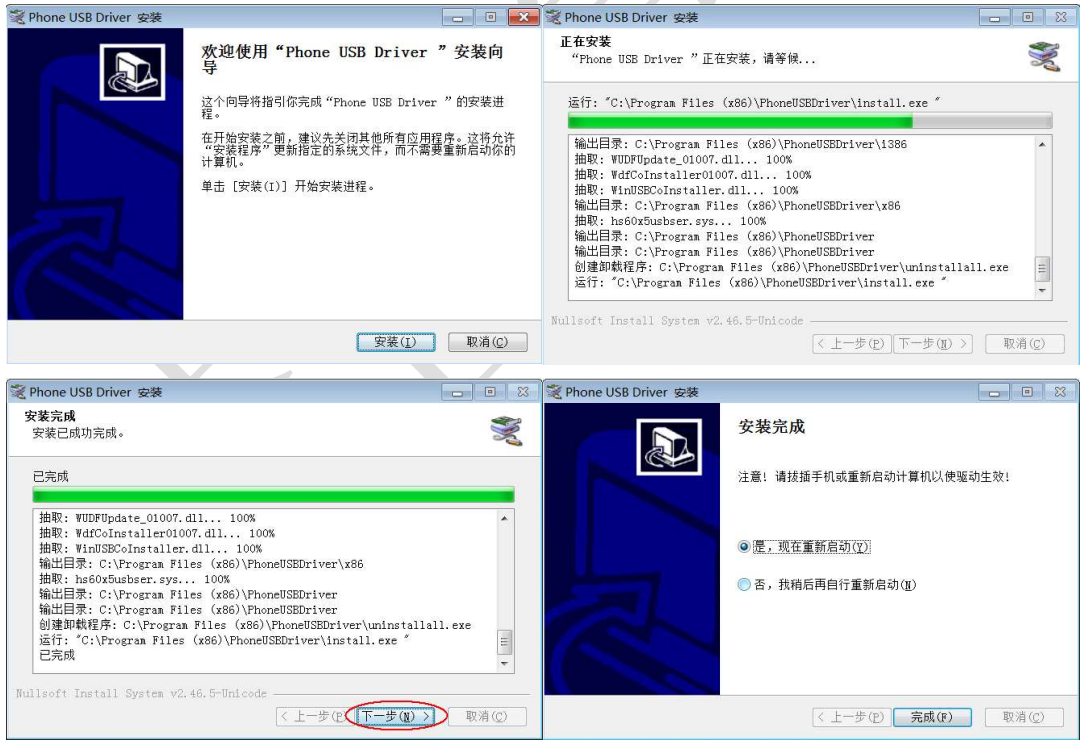


图 3 驱动Setup程序安装步骤

安装完成后，重新启动计算机，至此，驱动已经安装完毕。

3.2 设置调试

当USB再次连接海信移动医疗终与计算机时，计算机的硬件管理器将出现ADB端口，如图4：

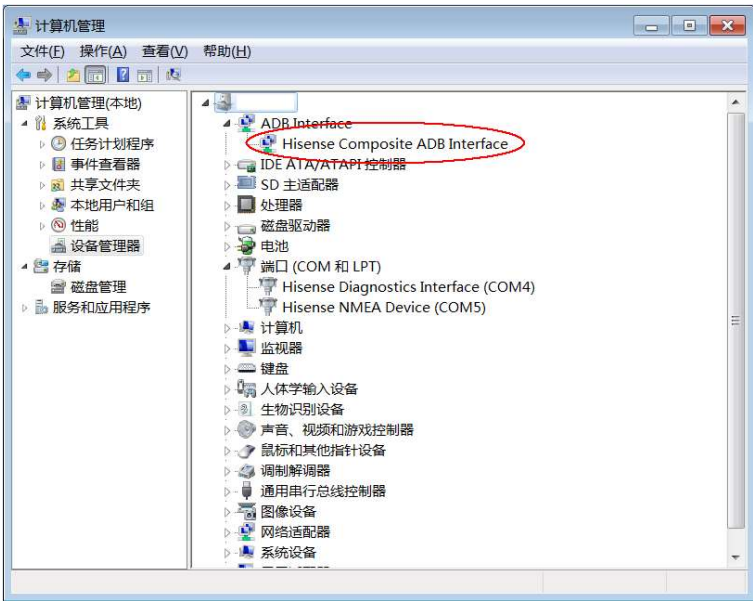


图 4 设备管理器出现ADB端口

请将手机的USB设置成“调试”状态，如图5：

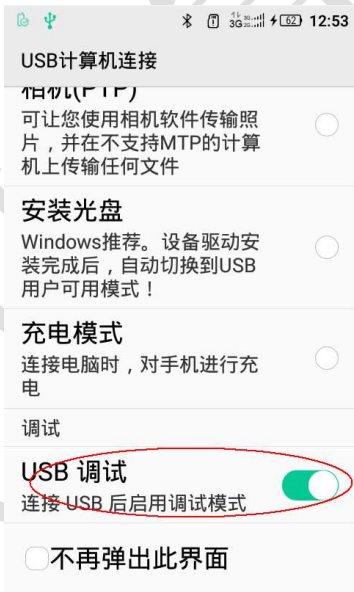


图 5 终端设置为“USB调试”

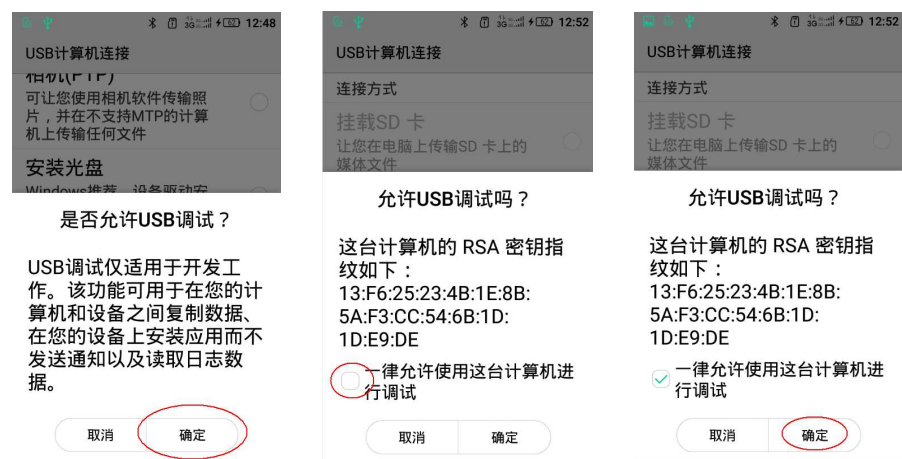


图 6 终端界面操作为调试状态

4 适配接口说明

4.1 获取扫描结果

功能：获取扫描头扫描出来的结果；

返回值：字符串

接口定义：通过 `intent` 获取，需要监听下面的广播名称，通过 `msg` 获取到扫描的最终结果；

广播名: "android.provider.sdlMessage"

扫描结果内容: `intent.getStringExtra("msg");`

4.2 获取指纹绑定账号

功能：录入指纹后会提示绑定账号和密码，通过指纹解锁终端后，动态获取对应指纹绑定的账号和密码；

返回值：字符串

前提条件：录入指纹并绑定账号

接口定义：录入指纹后可以绑定对应指纹的账号，指纹绑定账号后，通过广播发送账号，需要监听下面的广播，通过 `user` 获取账号名称，通过 `pw` 获取账号对应的密码。

广播名: "hmct.keyguard.unlocked"

用户名: `intent.getStringExtra("user");`

密码: `intent.getStringExtra("pw");`

4.3 通过解锁广播启动第三方应用

功能：在指纹解锁后直接启动某一个第三方应用程序

返回值：字符串

前提条件：录入指纹并绑定账号

接口定义：需要做如下处理：

在第三方应用的AndroidManifest.xml文件中添加如下处理，此处理的结果，会在指纹解锁后收到intent后自动启动该应用：

```
<receiver android:name="com.android.internal.app.ScreenLockReceiver"> //这里是调用接收广播的类名或者包名
```

```
<intent-filter>
```

```
    <action android:name="hmct.keyguard.unlocked" /> //之前约定的广播
```

```
</intent-filter>
```

```
</receiver>
```

4.4 第三方应用动态获取指纹绑定账号

功能：应用启动后需要通过触摸指纹动态获取到对应指纹的绑定账号

返回值：字符串

前提条件：录入指纹并绑定账号

接口定义：需要第三方应用自己在使用时调用 Android 标准指纹接口，实时获取指纹服务，以获取对应指纹的绑定账号信息

Step1: 取得指纹服务mFpm

```
private CancellationSignal mFingerprintCancelSignal;
```

```
private FingerprintManager mFpm =
```

```
(FingerprintManager) context.getSystemService(Context.FINGERPRINT_SERVICE);
```

Step2: 实现指纹事件回调

```
private FingerprintManager.AuthenticationCallback mAuthenticationCallback
```

```
    = new AuthenticationCallback() {
```

```
    @Override
```

```
    //指纹匹配失败
```

```
    public void onAuthenticationFailed() {
```

```
    };
```

```
    @Override
```


//指纹匹配成功

```
public void onAuthenticationSucceeded(AuthenticationResult result) {  
    Log.d(FP_TAG,"onAuthenticationSucceeded");  
    int fingerId = 0;  
    if (result != null && result.getFingerprint() != null) {  
        fingerId = result.getFingerprint().getFingerId();  
        // key_fp_account是保存在设置数据库中的Key值。  
        //这个Key值对应的value是json类型的数据  
        //json的Key值fingerId, value是user和密码，用;分割  
        String account = getFP(mContext.getContentResolver(), "key_fp_account",  
String.valueOf(fingerId));  
        if (account != null) {  
            String[] userAndPasswd = account.split(";");  
            if(userAndPasswd != null && userAndPasswd.length == 2) {  
                String user = userAndPasswd[0];  
                String password = userAndPasswd[1];  
                Intent intent = new Intent("hmct.keyguard.unlocked");  
                intent.putExtra("user",user);  
                intent.putExtra("pw",password);  
                mContext.sendBroadcast(intent);  
            }  
        }  
    }  
}
```

Step3: 调用authenticate监听指纹

```
mFingerprintCancelSignal = new CancellationSignal();  
mFpm.authenticate(null, mFingerprintCancelSignal, 0, mAuthenticationCallback, null, userId);
```

getFP方法:

```
public static String getFP(ContentResolver contentResolver, String setting, String key) {  
    int userId = UserHandle.getCallingUserId();  
    String json = Settings.System.getStringForUser(contentResolver, setting, userId);  
    Log.d(TAG,"getFP,json is: "+json);  
    if (json == null || json.equals("")) {
```

```
        return null;
    } else {
        try {
            JSONObject jsonObj = new JSONObject(json);
            return jsonObj.getString(key);
        } catch (JSONException e) {
            Log.d(TAG, "JSONException is :"+e);
            return null;
        }
    }
}
```

锁屏指纹匹配的源码位置：

frameworks/base/packages/Keyguard/src/com/android/keyguard/KeyguardUpdateMonitor.java

4.5 网络查看

功能：获取当前WIFI网络连接状态以及连接的相关信息

返回值：字符串

接口定义：

- 1、获取当前网络连接状态：NetworkInfo.getState(), 5 表示未连接、8 表示已连接
- 2、获取已连接热点的 SSID 接口如下：

```
WifiInfo wifiInfo = WifiManager.getConnectionInfo();
if (wifiInfo != null) {
    String ssid = wifiInfo.getSSID();
    if (ssid != null) {
        return WifiInfo.removeDoubleQuotes(ssid);
    }
}
```

5 错误原因和现象描述

5.1 ADB 服务未启动

原因：

- (1) USB数据线未连接；

(2) 未设置终端进入调试状态；

(3) 软件版本不支持

对策：

(1) 正确连接 USB 线；

(2) 设置终端进入调试，参照 3.2；