



# (12)发明专利申请

(10)申请公布号 CN 107544900 A

(43)申请公布日 2018.01.05

(21)申请号 201610475999.4

(22)申请日 2016.06.27

(71)申请人 北京优朋普乐科技有限公司

地址 100123 北京市朝阳区高碑店高井文  
化园8号东亿国际传媒产业园三期B座  
6层

(72)发明人 李波

(74)专利代理机构 北京恒都律师事务所 11395

代理人 王清亮

(51) Int. Cl.

G06F 11/36(2006.01)

G06F 9/445(2006.01)

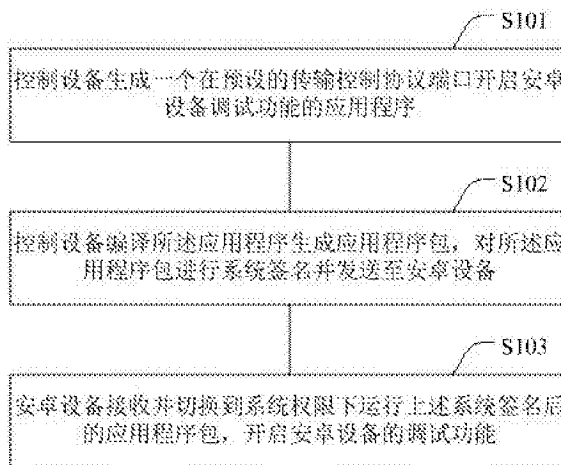
权利要求书2页 说明书6页 附图2页

## (54)发明名称

安卓设备及其调试桥的开启方法和装置、控制系统

## (57)摘要

本申请提供了一种安卓设备及其调试桥的开启方法和装置、控制系统,所述方法包括:控制设备生成一个在预设的传输控制协议端口开启安卓设备调试功能的应用程序;编译所述应用程序生成应用程序包,对所述应用程序包进行系统签名,并发送至安卓设备;安卓设备接收并切换到系统权限下运行上述系统签名后的应用程序包,开启安卓设备的调试功能。本申请通过上述手段,可以在不破坏安卓设备外包装的条件下就能达到开启ADB调试功能的目的,方便开发人员在安卓设备上/App开发及调试。



1. 一种安卓设备调试桥的开启方法,其特征在于,在安卓设备侧,所述方法包括:  
接收应用程序包,所述应用程序包中的应用程序包括有在预设的传输控制协议端口开启安卓设备调试功能的命令,且对编译所述应用程序生成的应用程序包进行系统签名;  
切换到系统权限下运行所述系统签名后的应用程序包,开启安卓设备的调试功能。
2. 根据权利要求1所述的方法,其特征在于,所述在预设的传输控制协议端口开启安卓设备调试功能的命令具体为:在预设的传输控制协议端口重新启动安卓调试桥守护进程的命令。
3. 根据权利要求1或2所述的方法,其特征在于,所述预设的传输控制协议端口的端口号为5555。
4. 根据权利要求1所述的方法,其特征在于,对编译所述应用程序生成的应用程序包进行系统签名,具体为:在对所述应用程序进行编译生成应用程序包时,利用安卓系统的系统密钥以及签名类库对所述应用程序包执行签名操作。
5. 根据权利要求4所述的方法,其特征在于,所述系统密钥包括platform.pk8和platform.x509.pem;所述签名类库为signapk.jar。
6. 一种安卓设备调试桥的开启装置,其特征在于,在安卓设备侧,所述装置包括:  
接收模块,用于接收应用程序包,所述应用程序包中的应用程序包括有在预设的传输控制协议端口开启安卓设备调试功能的命令,且对编译所述应用程序生成的应用程序包进行系统签名;  
调试桥开启模块,用于切换到系统权限下运行所述系统签名后的应用程序包,开启安卓设备的调试功能。
7. 一种安卓设备,其特征在于,包括权利要求6所述的安卓设备调试桥的开启装置。
8. 一种安卓设备调试桥的开启方法,其特征在于,执行所述方法的系统包括有控制设备和安卓设备,所述方法包括:  
在控制设备侧,生成一个在预设的传输控制协议端口开启安卓设备调试功能的应用程序;编译所述应用程序生成应用程序包,对所述应用程序包进行系统签名,并将系统签名后的应用程序包发送至安卓设备;  
在安卓设备侧,接收并切换到系统权限下运行所述系统签名后的应用程序包,开启安卓设备的调试功能。
9. 根据权利要求8所述的方法,其特征在于,所述应用程序包括有在预设的传输控制协议端口重新启动安卓调试桥守护进程的命令。
10. 根据权利要求8或9所述的方法,其特征在于,所述预设的传输控制协议端口的端口号为5555。
11. 根据权利要求8所述的方法,其特征在于,编译所述应用程序生成应用程序包,对所述应用程序包进行系统签名,具体包括:在对所述应用程序进行编译生成应用程序包时,利用安卓系统的系统密钥以及签名类库对所述应用程序包执行签名操作。
12. 根据权利要求11所述的方法,其特征在于,所述系统密钥包括platform.pk8和platform.x509.pem;所述签名类库为signapk.jar。
13. 一种安卓设备调试桥的开启控制系统,其特征在于,包括安卓设备和控制设备,其中:

所述控制设备包括：应用程序生成模块，用于生成一个在预设的传输控制协议端口开启安卓设备调试功能的应用程序；系统签名模块，用于编译所述应用程序生成应用程序包，对所述应用程序包进行系统签名；发送模块，用于将系统签名后的应用程序包发送至安卓设备；

所述安卓设备包括：接收模块，用于接收所述控制设备发送的应用程序包；调试桥开启模块，用于切换到系统权限下运行所述系统签名后的应用程序包，开启安卓设备的调试功能。

## 安卓设备及其调试桥的开启方法和装置、控制系统

### 技术领域

[0001] 本申请涉及安卓设备维护技术领域,特别地,涉及一种安卓设备及其调试桥的开启方法和装置、控制系统。

### 背景技术

[0002] 安卓调试桥(ADB,Android Debug Bridge)是针对安卓设备开发的一套设备调试接口,当用户使用ADB模式从电脑连接到安卓设备时,电脑端就会发现这个ADB设备,此时用户就可以使用ADB工具对该安卓设备进行控制,包括安卓设备上程序的安装与卸载、系统的诊断以及应用程序(App,Application)的逐行调试等。

[0003] 目前市面上出售的基于安卓系统的播放设备,如机顶盒、电视等,为了系统安全,一般不允许安装额外的App,也不允许对App进行调试,只能从系统内置的应用商店里下载安装App。开发人员购买到这种设备后,为了调试自己编写的App,需要开启设备的ADB调试接口,进行App的实时调试。

[0004] 在现有技术条件下,开发人员一般只能通过拆开这些安卓设备,找到电路板上串口的连接位置,在该连接位置焊接针脚,然后通过串口转USB线,将另一端连接到电脑的USB接口,从而在电脑上通过串口操作软件打开安卓设备的ADB调试接口。但是这种方式由于破坏了设备包装,可能会使安卓设备失去保修资格。

### 发明内容

[0005] 本申请提供一种安卓设备调试桥的开启方法和装置、控制系统,用于解决现有技术为开启安卓设备的ADB接口需要破坏设备包装而可能失去保修资格的问题。

[0006] 本申请公开的一种安卓设备调试桥的开启方法,在安卓设备侧,所述方法包括:接收应用程序包,所述应用程序包中的应用程序包括有在预设的传输控制协议端口开启安卓设备调试功能的命令,且对编译所述应用程序生成的应用程序包进行系统签名;切换到系统权限下运行所述系统签名后的应用程序包,开启安卓设备的调试功能。

[0007] 优选的,所述在预设的传输控制协议端口开启安卓设备调试功能的命令具体为:在预设的传输控制协议端口重新启动安卓调试桥守护进程的命令。

[0008] 优选的,所述预设的传输控制协议端口的端口号为5555。

[0009] 优选的,对编译所述应用程序生成的应用程序包进行系统签名,具体为:在对所述应用程序进行编译生成应用程序包时,利用安卓系统的系统密钥以及签名类库对所述应用程序包执行签名操作。

[0010] 优选的,所述系统密钥包括platform.pk8和platform.x509.pem;所述签名类库为signapk.jar。

[0011] 本申请公开的一种安卓设备调试桥的开启装置,在安卓设备侧,所述装置包括:接收模块,用于接收应用程序包,所述应用程序包中的应用程序包括有在预设的传输控制协议端口开启安卓设备调试功能的命令,且对编译所述应用程序生成的应用程序包进行系统

签名;调试桥开启模块,用于切换到系统权限下运行所述系统签名后的应用程序包,开启安卓设备的调试功能。

[0012] 本申请还公开了一种设置有上述安卓设备调试桥的开启装置的安卓设备。

[0013] 本申请公开的一种安卓设备调试桥的开启方法,执行所述方法的系统包括有控制设备和安卓设备,所述方法包括:在控制设备侧,生成一个在预设的传输控制协议端口开启安卓设备调试功能的应用程序;编译所述应用程序生成应用程序包,对所述应用程序包进行系统签名,并将系统签名后的应用程序包发送至安卓设备;在安卓设备侧,接收并切换到系统权限下运行所述系统签名后的应用程序包,开启安卓设备的调试功能。

[0014] 优选的,所述应用程序包括有在预设的传输控制协议端口重新启动安卓调试桥守护进程的命令。

[0015] 优选的,所述预设的传输控制协议端口的端口号为5555。

[0016] 优选的,编译所述应用程序生成应用程序包,对所述应用程序包进行系统签名,具体包括:在对所述应用程序进行编译生成应用程序包时,利用安卓系统的系统密钥以及签名类库对所述应用程序包执行签名操作。

[0017] 优选的,所述系统密钥包括platform.pk8和platform.x509.pem;所述签名类库为signapk.jar。

[0018] 本申请公开的一种安卓设备调试桥的开启控制系统,包括安卓设备和控制设备,其中,所述控制设备包括:应用程序生成模块,用于生成一个在预设的传输控制协议端口开启安卓设备调试功能的应用程序;系统签名模块,用于编译所述应用程序生成应用程序包,对所述应用程序包进行系统签名;发送模块,用于将系统签名后的应用程序包发送至安卓设备;所述安卓设备包括:接收模块,用于接收所述控制设备发送的应用程序包;调试桥开启模块,用于切换到系统权限下运行所述系统签名后的应用程序包,开启安卓设备的调试功能。

[0019] 与现有技术相比,本申请优选实施例通过对应用程序包进行系统签名后执行的方式开启ADB调试功能,方法实施简单,且不破坏android设备外包装的条件下就能达到开启ADB调试功能的目的,方便开发人员在安卓设备上进行App开发及调试。

## 附图说明

[0020] 附图仅用于示出优选实施方式的目的,而并不认为是对本发明的限制。而且在整个附图中,用相同的参考符号表示相同的部件。在附图中:

图1为本申请安卓设备调试桥的开启方法第一实施例的流程图;

图2为本申请安卓设备调试桥的开启方法第二实施例的流程图;

图3为本申请安卓设备调试桥的开启装置一实施例的结构示意图;

图4为本申请安卓设备调试桥的开启控制系统一实施例的结构示意图。

## 具体实施方式

[0021] 为使本申请的上述目的、特征和优点能够更加明显易懂,下面结合附图和具体实施方式对本申请作进一步详细的说明。

[0022] 在本申请的描述中,需要理解的是,术语“第一”、“第二”仅用于描述目的,而不能

理解为指示或暗示相对重要性或者隐含指明所指示的技术特征的数量。由此,限定有“第一”、“第二”的特征可以明示或者隐含地包括一个或者更多个该特征。“多个”的含义是两个或两个以上,除非另有明确具体的限定。术语“包括”、“包含”及类似术语应该被理解为是开放性的术语,即“包括/包含但不限于”。术语“基于”是“至少部分地基于”。术语“一实施例”表示“至少一个实施例”;术语“另一实施例”表示“至少一个另外的实施例”。其他术语的相关定义将在下文描述中给出。

[0023] 参照图1,示出了本申请安卓设备调试桥的开启方法第一实施例的流程,执行所述方法的系统包括有控制设备和安卓设备,所述方法包括以下步骤:

步骤S101:控制设备生成一个在预设的传输控制协议(TCP,Transmission Control Protocol)端口开启安卓设备调试功能的应用程序。

[0024] 在进一步的优选实施例中,具体可以采用在上述应用程序中设置在预设的传输控制协议端口重新启动安卓调试桥守护进程(ADB,ADB Daemon)命令的方式实现。

[0025] 在另一优选实施例中,预设的传输控制协议端口的端口号优选采用默认端口5555,之后进行ADB连接时可以不输入端口号即可完成。当然,选用其它端口号也是可以的,只是ADB连接时稍显麻烦,需要输入连接端口号。

[0026] 具体实施时,应用程序可以采用如下代码:

```
public class MainActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button button = (Button)findViewById(R.id.getadbserver);
        button.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View arg0) {
                try {
                    Process p = Runtime.getRuntime().exec("setprop
service.adb.tcp.port 5555");
                    p.waitFor();
                    p = Runtime.getRuntime().exec("stop adb");
                    p.waitFor();
                    p = Runtime.getRuntime().exec("start adb");
                    p.waitFor();
                } catch (IOException e) {
                    e.printStackTrace();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        })
    }
}
```

```
    });  
  }  
}
```

上述代码中：

语句“`Runtime.getRuntime().exec("setprop service.adb.tcp.port 5555").waitFor()`”的作用是将ADB调试端口设置为默认端口5555。

[0027] 语句“`Runtime.getRuntime().exec("stop adbd").waitFor()`”的作用是停止ADB服务,其中adbd表示安卓调试桥守护进程(ADB Daemon)。

[0028] 语句“`Runtime.getRuntime().exec("start adbd").waitFor()`”的作用是开启ADB服务。以上两条命令连续执行,相当于是重启系统ADB服务。

[0029] 上述代码在用户点击执行之后,会在设备上的5555端口打开ADB调试功能。5555端口是ADB调试功能的默认端口。

[0030] 另外,需要在应用程序的AndroidManifest.xml文件中的manifest节点加入`android:sharedUserId="android.uid.system"`这个属性。

[0031] 步骤S102:控制设备编译所述应用程序生成应用程序包,对所述应用程序包进行系统签名,并将系统签名后的应用程序包发送至安卓设备。

[0032] 具体实施时,可以采用下述两种方式之一实现系统签名的目的:

方式一:将应用程序代码放在AOSP(Android Open Source Project)项目源代码目录的packages/apps/下,并且在Android.mk中添加`LOCAL_CERTIFICATE := platform`字段,之后设置编译AOSP的初始环境,进入此App所在的源码目录,执行mm命令编译即可。编译完成后会提示安卓应用程序包(APK,Android Application Package)存放的路径。这时生成的APK已具有系统权限。

[0033] 方式二:在AOSP项目源代码目录的build/target/product/security/下,存放有platform.pk8和platform.x509.pem两个系统密钥。另外找到out/host/linux-x86/framework/signapk.jar,如果没有,就在build/tools/signapk/目录下编译生成上述签名类库。之后执行如下命令进行apk的系统签名:`java -jar signapk.jar platform.x509.pem platform.pk8 your.apk your_signed.apk`。

[0034] 步骤S103:安卓设备接收并切换到系统权限下运行上述系统签名后的应用程序包,开启安卓设备的调试功能。

[0035] 具体的,可以将步骤S102生成的apk文件(即系统签名后的安卓应用程序包)拷贝到U盘或SD卡,并将上述U盘或SD卡插入到安卓设备的相应接口中。这时安卓设备会弹出提示窗口。用遥控器(或鼠标等)在安卓设备上点击此apk,查看此apk是否能执行安装。如果没有提示窗口,也可以通过安卓设备内置的应用商店安装一个文件管理器。在文件管理器中选择并安装这个apk。如果能正常安装完毕此apk,则在运行此apk后,安卓设备已打开了ADB调试功能。如果不能正常安装,说明这台安卓设备的系统已经做了系统签名的比对及校验,本申请方案对这台设备无效。

[0036] 最后,可以在电脑端使用ADB命令连接此设备,命令格式如下:“adb connect 设备ip”。连接之后即可在此安卓设备上进行App的开发与调试工作。

[0037] 参照图2,示出了本申请安卓设备调试桥的开启方法第二实施例的流程,执行主体

为安卓设备,所述方法包括以下步骤:

步骤S201:接收系统签名后的应用程序包。

[0038] 其中,系统签名后的应用程序包可以在其他设备生成,应用程序包中的应用程序包括有在预设的传输控制协议端口开启安卓设备调试功能的命令,且该其他设备需要对编译所述应用程序生成的应用程序包进行系统签名。

[0039] 步骤S202:切换到系统权限下运行上述系统签名后的应用程序包,开启安卓设备的调试功能。

[0040] 对于前述的各方法实施例,为了描述简单,故将其都表述为一系列的动作组合,但是本领域的技术人员应该知悉,本申请并不受所描述的动作顺序的限制,因为根据本申请,某些步骤可以采用其他顺序或同时执行;其次,本领域技术人员也应该知悉,上述方法实施例均属于优选实施例,所涉及的动作和模块并不一定是本申请所必须的。

[0041] 本申请还公开了一种在其上记录有用于执行上述方法的程序的计算机可读记录介质。所述计算机可读记录介质包括配置为以计算机(以计算机为例)可读的形式存储或传送信息的任何机制。例如,机器可读介质包括只读存储器(ROM)、随机存取存储器(RAM)、磁盘存储介质、光存储介质、闪速存储介质、电、光、声或其他形式的传播信号(例如,载波、红外信号、数字信号等)等。

[0042] 参照图3,示出了本申请安卓设备调试桥的开启装置一实施例的结构示意图,所述装置设置在安卓设备中,所述装置包括:

接收模块11,用于接收系统签名后的应用程序包;

本实施例中系统签名后的应用程序包可以在其他设备中生成,其中的应用程序包括有在预设的传输控制协议端口开启安卓设备调试功能的命令,该其他设备需要对编译所述应用程序生成的应用程序包进行系统签名;

调试桥开启模块12,用于切换到系统权限下运行上述系统签名后的应用程序包,开启安卓设备的调试功能。

[0043] 参照图4,示出了本申请安卓设备调试桥的开启控制系统一实施例的结构示意图,所述系统包括控制设备2和安卓设备1。

[0044] 控制设备2包括应用程序生成模块21、系统签名模块22和发送模块23其中:

应用程序生成模块21,用于生成一个在预设的传输控制协议端口开启安卓设备调试功能的应用程序。

[0045] 具体实施时,可以在上述应用程序中设置在预设的传输控制协议端口重新启动安卓调试桥守护进程的命令;其中,预设的传输控制协议端口的端口号优选采用TCP的默认端口5555。

[0046] 系统签名模块22,用于编译所述应用程序生成应用程序包,对所述应用程序包进行系统签名。

[0047] 其中,系统签名模块22的可以采用如下二种方式之一实现:

方式一:系统签名模块22在对所述应用程序进行编译生成应用程序包时,将本地的签名证书设置为平台级别。

[0048] 方式二:系统签名模块22利用安卓系统的系统密钥以及签名类库对所述应用程序包执行签名操作;其中,上述系统密钥可以采用platform.pk8和platform.x509.pem;签名



类库可以采用signapk.jar。

[0049] 发送模块23,用于将系统签名后的应用程序包发送至安卓设备。

[0050] 安卓设备1中设置有安卓设备调试桥的开启装置,该安卓设备调试桥的开启装置包括接收模块11和调试桥开启模块12,其中:

接收模块11,用于接收系统签名后的应用程序包;

调试桥开启模块12,用于切换到系统权限下运行上述系统签名后的应用程序包,开启安卓设备的调试功能。

[0051] 调试桥开启模块12开启安卓设备的调试功能后,即可在电脑端使用ADB命令连接此设备,并在此安卓设备上进行App的开发与调试工作。

[0052] 需要说明的是,上述装置和系统实施例属于优选实施例,所涉及的单元和模块并不一定是本申请所必须的。

[0053] 本说明书中的各个实施例均采用递进的方式描述,每个实施例重点说明的都是与其他实施例的不同之处,各个实施例之间相同相似的部分互相参见即可。对于本申请的装置和系统实施例而言,由于其与方法实施例基本相似,所以描述的比较简单,相关之处参见方法实施例的部分说明即可。

[0054] 以上对本申请所提供的一种安卓设备调试桥的开启方法和装置、控制系统,进行了详细介绍,本文中应用了具体个例对本申请的原理及实施方式进行了阐述,以上实施例的说明只是用于帮助理解本申请的方法及其核心思想;同时,对于本领域的一般技术人员,依据本申请的思想,在具体实施方式及应用范围上均会有改变之处,综上所述,本说明书内容不应理解为对本申请的限制。

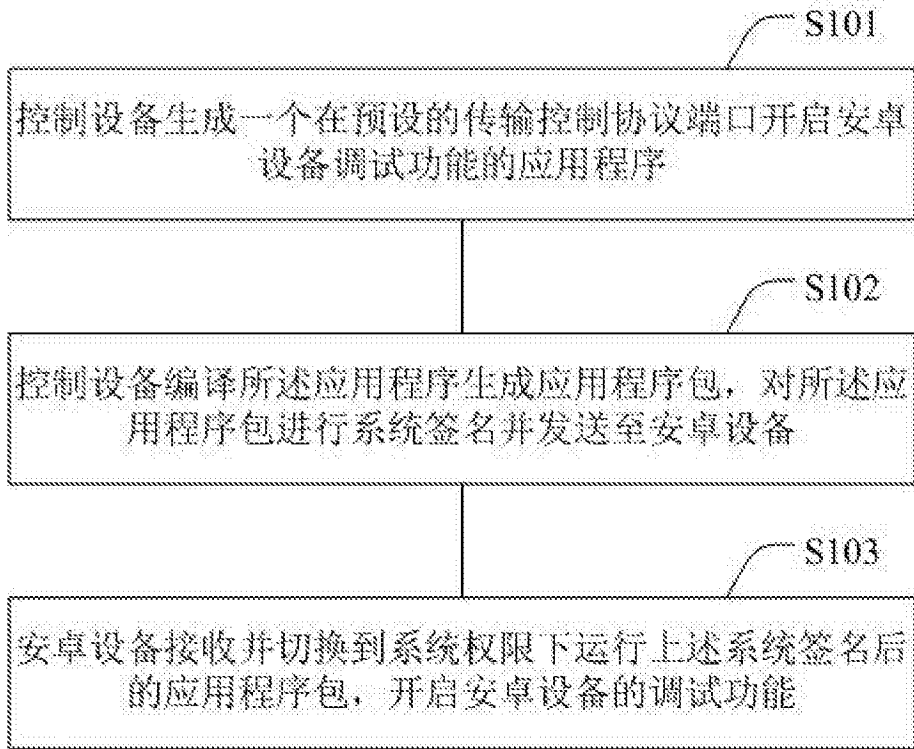


图1

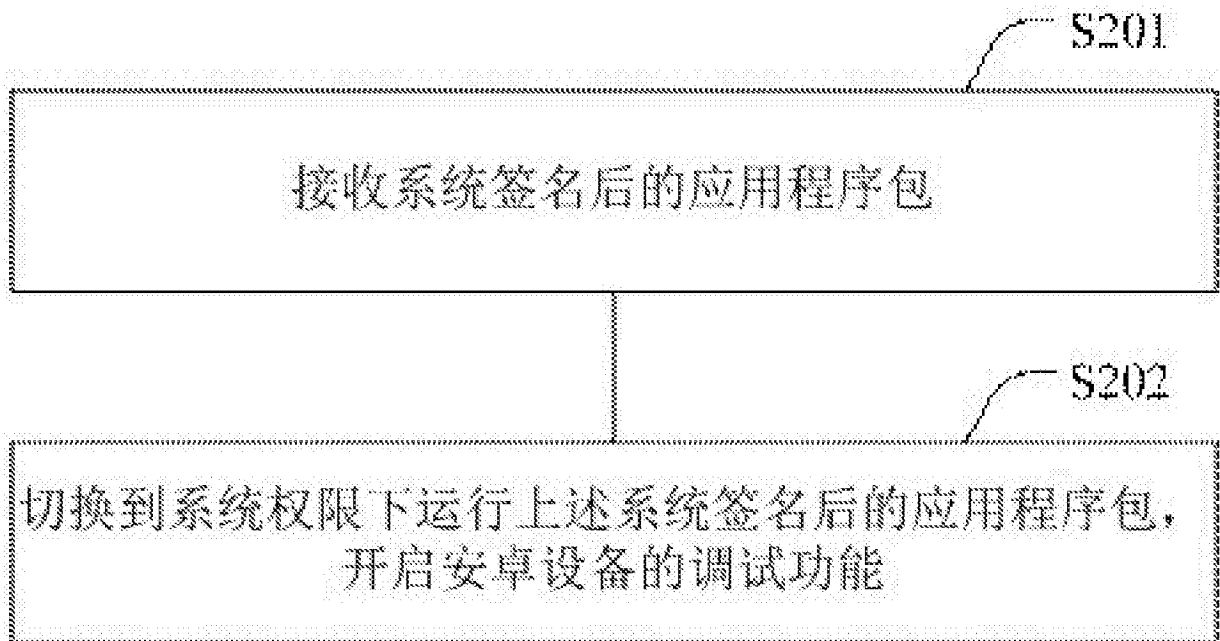


图2

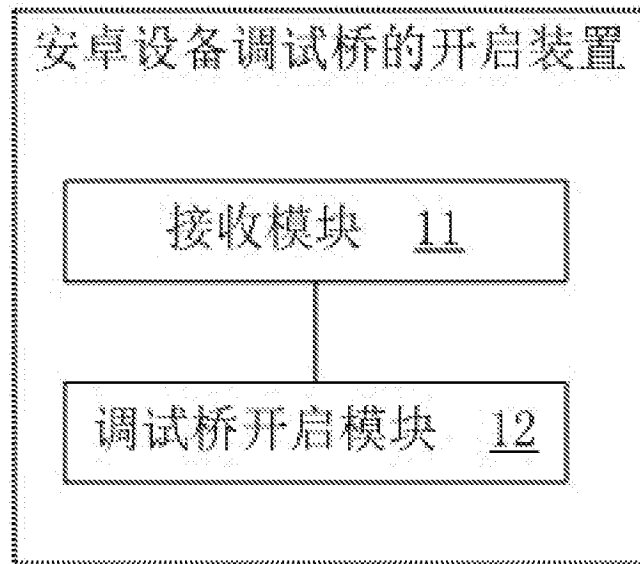


图3

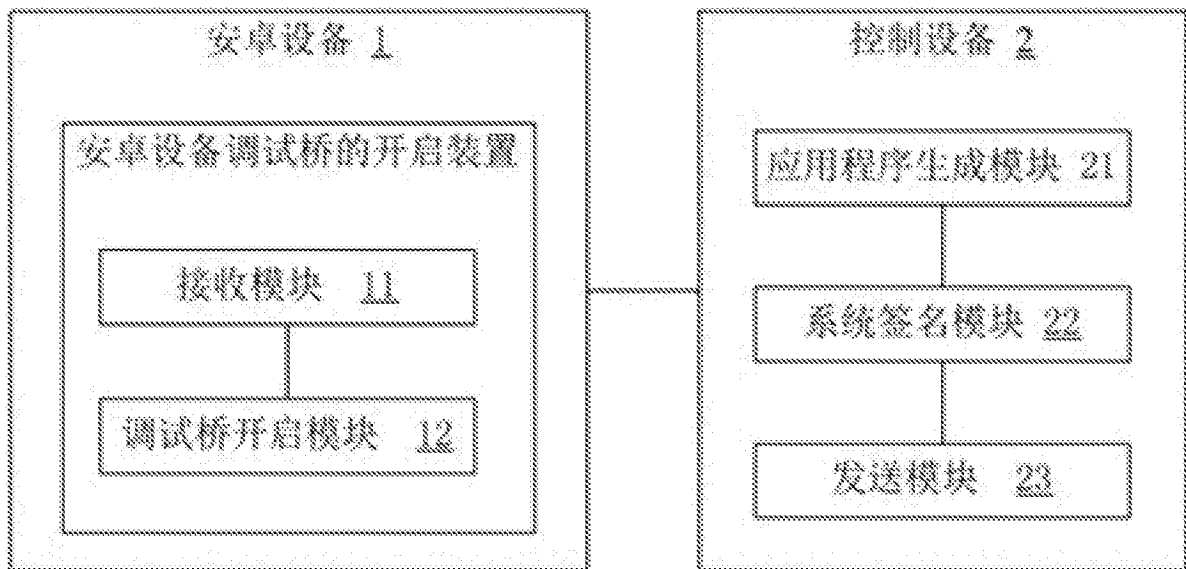


图4