System Requirement of Development under Android Platform

Table of Contents

1. Introduction

Except using Android SDK offered by Astra, properly setting the Android system is necessary in order to do the secondary development to Astra Camera under Android Platform. This document described the setting process for Android System which can meet Astra's development requirement.

2. Scope of Document

This document applies to Astra series' camera including Astra, Astra Pro, Astra Mini.

Minimum Platform Required: Windows 7 or Ubuntu 14.04

3. Add VID on Orbbec's Camera

Adding VID support in hasPermission function of UsbSettingsManager.java:

```
public boolean hasPermission(UsbDevice device) {
        synchronized (mLock) {
            int uid = Binder.getCallingUid();
            if (uid == Process.SYSTEM_UID) {
                return true;
            }
            //add support Orbbec VID
            if(device.getVendorId()==0x2bc5){
                return true;
            }
            //end support Orbbec VID
    ……..
}
```

Problem Solved:

Popping USB authorizing alert while access USB in game or using gesture control - It will influence user experience.

Unable to use the device since some models don not include pop-up USB authorizing.

4. Control Code for Adding Permissions

Adding control code for property in init.xxx.rc:

```
on property:persist.sys.orbbec.runtime=true
    chmod 0666 /dev/uinput
    chmod 0666 /dev/video0
    chmod 0666 /dev/video1
    chmod 0666 /dev/video2
    chmod 0666 /dev/video3
    setenforce 0
```

```
on property:persist.sys.orbbec.runtime=false
    chmod 0660 /dev/uinput
    chmod 0660 /dev/video0
    chmod 0660 /dev/video1
    chmod 0660 /dev/video2
    chmod 0660 /dev/video3
      setenforce 1
```

Control Part:

In frameworks/base/services/java/com/android/server, adding directory orbbec and attachment document.

In SystemServer.java, adding code to start OrbbecService. Code shown below:

```
/*start add orbbec */
OrbbecService orbbecService=new OrbbecService(context);
ServiceManager.addService("orbbec_service", orbbecService);
/*end add orbbec */
ActivityManagerService.self().systemReady(new Runnable() {
```

Problem Solved:

Needs to change /dev/uinput permission while using gesture control. Uinput can only be used after obtaining system permission.

Some system starts SELinux control mechanism which need to be paused when using gesture control.

5. Check Commonly Used Key

Check if the following keys exist in frameworks/base/data/keyboards/qwerty.kl:

```
key 102     HOME                WAKE
key 113     VOLUME_MUTE
key 114     VOLUME_DOWN         WAKE
key 115     VOLUME_UP           WAKE
```

If not exist, adding the keys.

Problem Solved:

Some key may not exist in qwerty.kl in some model. That may cause the problem when controlling silence mode, home and volume under gesture control.

6. Check if USB Supports Host Mode

Checking if android.hardware.usb.host.xml file exists in frameworks/native/data/etc, and checking if the

following contents exist:

```
<permissions>
    <feature name="android.hardware.usb.host" />
</permissions>
```

If all the contents exist, no need to change.

Problem Solved:
Some devices do not support Host mode of USB and fail to identify.

7. Driver for Virtual Touch Screen (Gesture Control)

For virtual touch (gesture control), Orbbec has ob_sim_driver and instruction guide available. It is recommended to directly compile with kernel. The detail explanation and code are in the attachment.
After re-compiling kernel, please install the TouchCursor.apk from the Test directory to achieve universal gesture control of the system.

Problem Solved:
Third-party app cannot be controlled with Orbbec Touch Cursor.

8. 3D Sensor Node Permission

When connecting Orbbec 3D sensor, please allow R/W permission for the created video or audio nodes.

9. Game Mode

Game Mode refers to video input delay which under 150ms (avoid obvious delay).

Problem Solved: Large time delay on Television or set-top boxes. Game mode can offer good game experience.