



Android rotation 配置 使用指南

版本号: 1.5

发布日期: 2024.11.25

版本历史

版本号	日期	制/修订人	内容描述
1.0	2023.3.15	KPA0368	初始版本文档
1.5	2024.11.25	KPA0368	新增问题背景，参数介绍



目 录

1 问题背景	1
2 概念介绍	2
3 配置模块参数介绍	3
4 方案配置步骤	5
4.1 竖屏或横屏机器方案配置	5
4.2 假横屏方案配置	6
5 FAQ	8



1 问题背景

在 Android 的生态中，会出以三种场景：

1. 硬件屏幕方向与硬件结构不匹配，如电源和音量键没有在屏幕显示上方，导致与使用习惯不符，需要调整屏幕方向。
2. 产品定位为横屏机器，但 Android 应用生态都是以竖屏为主，故很多应用会出现视频应用或摄像头应用方向异常问题。
3. 第三方应用为兼容不同设备，提供相关属性给设备产商兼容该设备，需要设备厂商进行配置。

第一种场景：Android 提供修改属性的方式，提供如下 `ro.surface_flinger.primary_display_orientation`（Android 系统启动方向），`ro.minui.default_rotation`（Recovery 方向）允许进一步修改屏幕方向，以满足相关要求。

第二种场景：全志提供了修改属性的方式，`ro.primary_display.user_rotation` 将横屏机器修改为竖屏机器，但启动时以横屏状态启动方式。

第三种场景：目前国内代表应用为 qq, 提供给设备厂商进行适配 `ro.qq.camera.sensor`。

2 概念介绍

• 横屏机器以及竖屏机器

通过查看 `mCurrentOrientation` 为 0 时候机器所处的机器状态，如此时机器是竖屏状态则称为竖屏机器，机器如果是横屏状态则成为横屏机器。

```
console/# dumpsys display | grep mCurrentOrientation  
mCurrentOrientation=0
```

当 0 度角方向为竖屏的机器时，强制竖屏的 APP 就会在此方向进行展示，而强制横屏的 APP 就会在顺时针 90 度进行展示。

当 0 度角方向为横屏的机器时，强制横屏的 APP 就会在此方向进行展示，而强制竖屏的 APP 就会在顺时针 90 度进行展示。

• 假横屏机器

只是启动的时候以横屏状态展示，但实际为竖屏；原因是市面上大部分开发者开发的应用都是针对手机进行开发的，竖屏的机器较横屏能兼容更好的 APP 应用；所以为了满足一些客户需要横屏机器展示，但又能兼容市面上竖屏的应用，提供了 `ro.primary_display.user_rotation`，对横屏机器进行修正。

如果是做横屏平板的，建议采用如下硬件布局：

以便强制横屏或者竖屏的应用打开时，方向符合用户习惯。

3 配置模块参数介绍

1. 屏幕方向相关参数介绍。

通过以下命令进入相关的方案配置路径。

```
source build/envsetup.sh  
lunch 对应版型  
cdevice
```

[device_name]/device-common.mk

```
PRODUCT_PROPERTY_OVERRIDES += \  
ro.surface_flinger.primary_display_orientation=ORIENTATION_0 \#屏幕原始朝向修正, 当设置为ORIENTATION_90  
时, 屏幕原始方向顺时针旋转90  
ro.vendor.sf.rotation=0 \#Gsensor调整方向参数  
ro.vendor.gsi_gsen_rotation=0 \#一般默认为0或者不设置  
  
PRODUCT_SYSTEM_DEFAULT_PROPERTIES += \  
ro.primary_display.user_rotation=0 \#修改机器为假横屏,当设置为ORIENTATION_90时, 屏幕原始方向逆时针旋转  
90  
#Recovery方向, ROTATION_NONE,ROTATION_LEFT,ROTATION_DOWN, ROTATION_RIGHT,分别对应ro.  
surface_flinger.primary_display_orientation的0, 90, 180, 270  
ro.minui.default_rotation=ROTATION_NONE \  
ro.qq.camera.sensor=0 \#QQ应用自己会读取该值, 修正视频通话时候的方向; 如果异常可以改该值, 没有异常可以不动  
ro.primary_display.user_rotation=0
```

新方案的时候建议配置以上的值都为 0, 或者不配置, 如当前的屏幕状态已经满足需求, 则不需要对屏幕方向进行适配, 配置 sensor 以及 TP 的方向即可。

2. 初始状态下的 TP,gsensor 配置路径。

TP 初始方向配置路径: `longan/device/config/chips/[chip_name]/configs/[device_name]/[kernel-version]/board.dts。`

快捷命令到对应路径:

```
source build/envsetup.sh  
lunch \#选择对应的版型  
cbd
```

3. Sensor 初始方向配置路径: `android/device/softwinner/saturn/[board_name]/input/gsensor.cfg。`

快捷命令:

```
source build/envsetup.sh  
lunch #选择对应的版型  
cdevice
```



4 方案配置步骤

1. adb root && adb remount 保证 system 与 vendor 皆可进行读写。
2. 找到对应文件即可在小机上对应参数进行修改验证。

```
system/build.prop  
vendor/build.prop  
system/usr/idc/gslX680.idc
```

4.1 竖屏或横屏机器方案配置

1. 配置板子的 0 度角方向。

```
#adb remount后, busybox vi vendor/build.prop  
ro.surface_flinger.primary_display_orientation=ORIENTATION_0 \ #设置屏幕原始朝向,配置成你想要的显示方向, 此时0度  
方向会顺时针旋转对应的角度  
#adb remount后, busybox vi system/build.prop  
#当如上配置好后即可配置  
ro.minui.default_rotation=ROTATION_NONE
```

如机器物理竖屏, 无需修改; 如机器本来为横屏, 则需修改以上两个属性从而修改屏幕方向。

2. 配置 Gsensor 方向。

```
#adb remount后, busybox vi vendor/build.prop  
ro.vendor.sf.rotation=0
```

3. 配置 TP 的方向。

配置路径为 longan/device/config/chips/[chip-name]/configs/[device_name]

4. 配置 camera.cfg。

```
camera_orientation
```

4.2 假横屏方案配置

在上述修改后，机器已实现竖屏与横屏机器的产品形态。当为兼容当前 Android 竖屏应用为主导的生态环境。还需如下步骤配置机器仅在启动的时候显示为横屏，但实际为竖屏；**如机器为竖屏，或能接受横屏状态下，竖屏应用的兼容性异常问题**，则可不进行下面配置。

配置机器假横屏

一：再次修正屏幕 0 度角方向。

```
#adb remount后, busybox vi system/build.prop
ro.primary_display.user_rotation=0 #逆时针旋转多少角度为屏幕的0度角, 当通过第一步配置的是横屏机器的时候, 可以通过该配置选择一个竖屏方向作为0度方向
```

如按推荐设计产品外观，则该值一般可设置为 90。

二：配置 TP 方向。

• Android12 TP 朝向

ro.input_flinger.primary_touch.rotation 取值为 $(360 - ro.primary_display.user_rotation)$ ，其值有 InputFlinger 所使用，用于支持维持刷 GSI 后触摸位置的正确性，如需支持刷 GSI 后触摸正常，必须设置此属性。

• Android 13 后 TP 朝向

```
#adb remount后, busybox vi system/usr/idc/gslX680.idc
touch.orientation
```

在 Android 12 及之前版本，为了兼容 GSI 和非 GSI 的触摸，需要配置 ro.input_flinger.primary_touch.rotation 属性，来保证触摸旋转正常。

在 Android 13，由于 GMS 要求，配置属性方式或导致 GMS 出现 fail，同时在 Android 13 谷歌支持了通过配置 idc 文件（Input Device Configuration）方式旋转 touch，因此在 Android 13，使用 idc 的方式来支持兼容 GSI 和非 GSI 的触摸。

以 gslX680 触摸为例，讲述如何使用 idc 支持配置。

获取 input 设备名

启动完成后，通过 getevent 或者 dumsys input 获取触摸设备的设备节点名，例如 gsl 触摸驱动的 input 节点名为 gslX680。

```
console:/ # getevent
add device 1: /dev/input/event5
name: "MH248"
add device 2: /dev/input/event4
name: "gslX680"
add device 3: /dev/input/event3
```

```
name: "sunxi-keyboard"
```

方案目录配置 idc 文件

1. 在 Android 方案目录 input 文件夹里，创建一个 idc 文件，命名为 gslX680.idc，参考如下。

```
./ceres-b10/input/gslX680.idc
./ceres-y13/input/gslX680.idc
./ceres-b7/input/gslX680.idc
./ceres-b6/input/gslX680.idc
./ceres-p25/input/gslX680.idc
./ceres-b4/input/gslX680.idc
./ceres-b3/input/gslX680.idc
```

图 4-1: gslX680.idc

2. 在 gslX680.idc 里配置 touch.orientation 信息，其中取值 ro.input.flinger.primary_touch.rotation 的角度值相同，参考如下。

```
touch.orientation=ORIENTATION_90
```

其中，ORIENTATION_90 可配置为 ORIENTATION_0、ORIENTATION_180、ORIENTATION_270，分别对应相同的角度。

3. 在 input 目录下 config.mk 中，将 gslX680.idc 复制到 system 分区下，参考如下。

```
PRODUCT_COPY_FILES += \
$(LOCAL_MODULE_PATH)/gslX680.idc:$(TARGET_COPY_OUT_SYSTEM)/usr/idc/gslX680.idc
```

说明

gslX680.idc 必须放到 system 目录下，确保烧写 GSI 后，该文件会被擦除。

5 FAQ

1. 在配置之前，都首先把所有相关的属性都先配置 0 么，确定好初始状态后，再开始调配置么（类似做一个初始化？）？0 度方向会随这些配置而改变吗？

先配置为 0，是为了确认屏幕物理的屏幕方向；方便后面根据此进行屏幕配置；0 度方向会因为这些配置发生改变。

2. `ro.surface_flinger.primary_display_orientation`，这个是以 0 度方向为参考，逆时针旋转多少度的意思吗？

以初始方向，顺时针方向旋转 `ro.surface_flinger.primary_display_orientation` 为 0 度方向。

3. `ro.minui.default_rotation` 的 4 个值，`ROTATION_NONE`，`ROTATION_LEFT`，`ROTATION_RIGHT`，`ROTATION_DOWN`，分别代表什么意思（或者方向）

这个是进入 recovery 使用的配置，分别对应 `ro.surface_flinger.primary_display_orientation` 中 0，90，270，180。

4. `ro.vendor.sf.rotation` 和 `ro.vendor.gsi_gsen_rotation`，这两个都是 Gsensor 调整方向相关么，一般配置一样的值？

`ro.vendor.gsi_gsen_rotation` 一般配置为 0 或者不配，除非 GSI 中方向异常。

5. 系统是横屏的，但是打开 camera 应用是竖屏，这个是由哪个属性决定？（只跟这个 `camera_orientation` 相关吗？）

应用是否为竖屏是应用自己决定的，`camera_orientation` 决定的是展示在 Surface 上的方向。

6. `touch.orientation` 的值，是不是跟 `ro.surface_flinger.primary_display_orientation` 一致？

否，这个是 TP 的方向配置，配置的时候调整到匹配当前方向即可。




著作权声明

版权所有 ©2024 珠海全志科技股份有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明

、、**全志科技**、（不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。