



Android 10 配置

Android10 rotation 配置

1.0

2020.06.10

文档履历

版本号	日期	制/修订人	内容描述
1.0	2020.06.10		Android10 rotation 配置

目录

1. Android10 rotation 配置	1
1.1 ro.surface_flinger.primary_display_orientation	2
1.2 ro.primary_display.user_rotation	2
1.3 ro.input_flinger.primary_touch.rotation	3
1.4 ro.vendor.sf.rotation	3
1.5 camera.cfg	4
1.6 例子	4
2. Declaration	6

1. Android10 rotation 配置

Android10 大概有以下几个模块设计到了方向的问题：显示方向、显示设备方向、触摸方向、方向类型传感器(加速度传感器、陀螺仪、磁力计等)、camera 方向等等。如 A100 上使用了新的竖屏横用的机制，以解决大部分只针对竖屏机器的应用兼容性问题。这个配置适用于全志平板所有 Android10 的系统上。下面以 A100 为例进行的 rotation 配置教程。

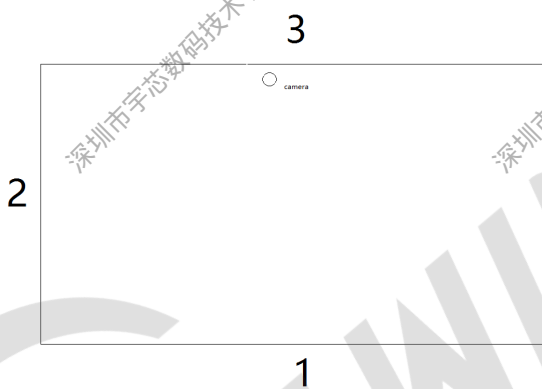


图1: 显示设备方向

上图是 A100 B3 机器的设备方位图，是以显示屏原始方向定位的。0 方向是显示屏默认的方向，即什么配置都不设置后，界面的朝向是 0 方向。其他的 3 个方向就是顺时针旋转的方向。其中 camera 所在的位置是我们用户的正方向，也就是 3 方向是用户拿设备的方向。

A100 B3 是属于大尺寸的设备，定位是平板，方向是横屏，但是目前遇到了不少应用对于横屏机器的不友好(兼容性问题，特别是网络视频播放器和聊天软件)，表现在视频播放时横置全屏、竖屏退出全屏的功能紊乱，聊天软件视频通话时 camera 方向异常等，这些问题在竖屏配置上都没有问题，因此我们将设备配置成了竖屏机器，以兼容这些应用，但产品端需要机器启动后是横屏的，因此在 android 上做了相对应的修改，使系统起来后设备旋转 90 度或者 270 度变成横屏。

下面是一些配置的介绍：

1.1 ro.surface_flinger.primary_display_orientation

这个属性是 mtk 提交到 android SF 的一个配置，这个属性可以取以下值：ORIENTATION_0、ORIENTATION_90、ORIENTATION_180 和 ORIENTATION_270。这个配置的作用是在 SF 层对显示方向进行了旋转。我们的显示设备是有显示方向的，这个方向一般无法修改，但可以通过 SF 进行旋转。

但在我们的方案下进行了一些修改，这个属性是指基于显示屏的默认方向需要旋转的角度，也就是用户的正方向。

如 A100 B3 机器，以 camera 所在的边为我们设备的主方向（启动后界面朝向），但显示设备的原始方向 0 度方向是在右边的，如果需要启动后正方向是在方向 3 上，则需要将 ro.surface_flinger.primary_display_orientation 设置为 ORIENTATION_270。

但是这样做后，android 所拿到的 display size 的 width 会比 height 大，也就是我们所说的横屏机器。这种方案会出现很多应用兼容性的问题，特别是国内的一些应用，主要是针对手机开发的，对这种横屏的应用会有很多问题。因此我们需要将机器设置成竖屏的，但启动后进行默认旋转为横屏的，同时保持刷 GSI 后也是横屏的，因此这个指需要保持为 ORIENTATION_270，然后由下面的一个配置进行配置，在 SF 中再次旋转。这样就保证了用户的正方形就是在 3 方向上。

这个属性在我们的源码中的含义被重新定义了，这个属性的值正确来说，就是用户的正方向所旋转的角度，SF 的主显方向旋转角度计算公式如下：

$$\text{SF display orientation} = \text{ro.surface_flinger.primary_display_orientation} - \text{ro.primary_display.user_rotation}$$

1.2 ro.primary_display.user_rotation

这个属性是我们加的一个配置，这个属性单独使用没有意义，需要与 ro.surface_flinger.primary_display_orientation 相互使用。这个属性的作用是说在 SF display 方向（即系统的 0 度方向）进行旋转的角度（相当于打开自动旋转后旋转到想要的方向，再将自动旋转关闭的效果）使系统旋转到用户的正方向。

ro.primary_display.user_rotation 的驱取值范围为：0, 90, 180, 270。

这个属性与 ro.surface_flinger.primary_display_orientation 的配合关系如下，如上图，0 方向是显示设备的方向，我们需要设置用户的正方向是在 3 方向上，那么我们需要将 ro.surface_flinger.primary_display_orientation 设置为 ORIENTATION_270，然后如果想要将 android 的默认方向（rotation=0）设置为 2 方向上，那么 3 方向相当于是 2 方向的 90 度方向，此时将

ro.primary_display.user_rotation 设置为 90，系统启动后，display 的 rotation=1(90 度)。

这种做法是因为我们将 SF 中对 ro.surface_flinger.primary_display_orientation 的处理进行了修改，将 SF 的主显方向设置成 ro.surface_flinger.primary_display_orientation - ro.primary_display.user_rotation。这么做的原因是解决刷 GSI 后系统方向改变的问题。

当 ro.surface_flinger.primary_display_orientation 小于 ro.primary_display.user_rotation 时，SF 旋转的角度为 ro.surface_flinger.primary_display_orientation +360- ro.primary_display.user_rotation。

1.3 ro.input_flinger.primary_touch.rotation

这个属性也是我们加的一个配置。这个配置是在 android 的 InputFlinger 上对主触摸设备的触摸事件进行旋转。由于我们的旋转配置是在 system 分区的，一旦刷 GSI 就会失效，为了保持与刷了 GSI 后的状态一致，因此触摸的驱动配置应该是与 ro.surface_flinger.primary_display_orientation 配置的方向一致。但同时这里就出个一个问题，如上图，刷 GSI 后的 0 度方向是在 3 方向上，但没有刷 GSI 的固件 0 度方向是在 2 方向上，因此触摸的方向是不同的，因此我们需要在 InputFlinger 上对触摸方向进行旋转。

ro.input_flinger.primary_touch.rotation 的取值范围为：0，90，180，270。

这个值得取值是根据刷了 GSI 后也就是驱动的方向与 android 的默认方向进行设置的，刷 GSI 后的 android 默认方向也就是 ro.surface_flinger.primary_display_orientation 所设定的方向，如驱动上触摸的方向配置为 3 方向上(刷 GSI 后的方向)，而 android 的 0 度方向是在 2 方向上，顺时针旋转了 270 度，因此 ro.input_flinger.primary_touch.rotation=270。

1.4 ro.vendor.sf.rotation

这个属性目前是 A100 sensor HAL 中 AccelSensor 进行旋转的角度，这个角度的取值也是基于用户正方向时的旋转角度，刷 GSI 后与正常固件的方向问题已在 sensor HAL 中进行了处理。按照以前的配置即可。

1.5 camera.cfg

camera.cfg 中的 camera_orientation 也是基于用户正方向时的旋转角度，刷 GSI 后与正常固件的方向问题已在 camera HAL 中进行了处理。按照以前的配置即可。

1.6 例子

这个方案主要针对大屏机器，显示屏为竖屏但是希望启动为横屏的场景。

情景 1: 当显示屏显示方向为 0 方向, 启动后屏幕显示是在 3 方向上, android 系统的 ROTATION_0 方向是在 2 方向上, 则:

```
ro.surface_flinger.primary_display_orientation = ORIENTATION_270
```

```
ro.primary_display.user_rotation = 90
```

```
ro.input_flinger.primary_touch.rotation = 270
```

情景 2: 当显示屏显示方向为 0 方向, 启动后屏幕显示是在 3 方向上, android 系统的 ROTATION_0 方向是在 0 方向上, 则:

```
ro.surface_flinger.primary_display_orientation = ORIENTATION_270
```

```
ro.primary_display.user_rotation = 270
```

```
ro.input_flinger.primary_touch.rotation = 90
```

情景 3: 当显示屏显示方向为 2 方向, 启动后屏幕显示是在 3 方向上, android 系统的 ROTATION_0 方向是在 2 方向上, 则:

```
ro.surface_flinger.primary_display_orientation = ORIENTATION_90
```

```
ro.primary_display.user_rotation = 90
```

```
ro.input_flinger.primary_touch.rotation = 270
```

当不使用此方案时, 只需要将 ro.primary_display.user_rotation 和 ro.input_flinger.primary_touch.rotation 设置为 0 或不设置即可。

情景 4: 当显示屏显示方向为 3 方向, 启动后屏幕显示是在 3 方向上, android 系统的 ROTATION_0 方向是在 0 方向上, 则:

```
ro.surface_flinger.primary_display_orientation = ORIENTATION_0(或者不设置, 默认为 0)
```

```
ro.primary_display.user_rotation = 270
```

```
ro.input_flinger.primary_touch.rotation = 90
```

2. Declaration

This document is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner. The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This document neither states nor implies warranty of any kind, including fitness for any particular application.