

UNISOC ANDROID 10 LAUNCHER 客制化指南

适用产品信息	SC7731E/SC9863A/SC9832E/UMS512(T)/ UIS8581E/SL8541E
适用版本信息	Android 10/Android 10-Go
关键字	布局、图标、Search 框、应用抽屉、单层桌面、通知、未读、动态图标、文件夹、最近任务、主题、循环滑动

Unisoc Confidential For hiar

声明

本文件所含数据和信息都属于紫光展锐所有的机密信息，紫光展锐保留所有相关权利。本文件仅为信息参考之目的提供，不包含任何明示或默示的知识产权许可，也不表示有任何明示或默示的保证，包括但不限于满足任何特殊目的、不侵权或性能。当您接受这份文件时，即表示您同意本文件中内容和信息属于紫光展锐机密信息，且同意在未获得紫光展锐书面同意前，不使用或复制本文件的整体或部分，也不向任何其他方披露本文件内容。紫光展锐有权在未经事先通知的情况下，在任何时候对本文件做任何修改。紫光展锐对本文件所含数据和信息不做任何保证，在任何情况下，紫光展锐均不负任何与本文件相关的直接或间接的、任何伤害或损失。

请参照交付物中说明文档对紫光展锐交付物进行使用，任何人对紫光展锐交付物的修改、定制化或违反说明文档的指引对紫光展锐交付物进行使用造成的任何损失由其自行承担。紫光展锐交付物中的性能指标、测试结果和参数等，均为在紫光展锐内部研发和测试系统中获得的，仅供参考，若任何人需要对交付物进行商用或量产，需要结合自身的软硬件测试环境进行全面的测试和调试。

版本历史

版本	日期	备注
V1.0	2019/10/11	初稿
V1.1	2020/1/10	更新适用产品信息 更新章节 3.4.1 双行显示 3.5.2 Search 框移到顶部 3.7.2 定制 App 在应用抽屉位置 4.3.1 通知标记增强 4.6 下拉手势
V1.2	2020/7/21	缩略语更新 RRO、增加 SRO 更新章节 2.1.1 Launcher3GolconRecents 形态 3.4.3 隐藏图标 增加 3.8 资源 Overlay 章节

Unisoc Confidential For hiar

目录

1 前言	1
1.1 范围	1
1.2 缩略语	1
1.3 参考文献	1
2 概览	2
2.1 Android 10 Launcher3 升级变化	2
2.1.1 Launcher3GolconRecents 形态	2
2.1.2 Developer Options 功能调试开关界面	3
2.1.3 Go 版本差异	3
2.1.4 全面屏手势功能	4
3 UI 客制化	5
3.1 桌面布局	5
3.2 布局配置	5
3.2.1 布局配置客制化	7
3.2.2 显示大小对布局的影响	9
3.3 预置图标	9
3.3.1 图标预置类型介绍	9
3.3.2 预置 workspace 图标	13
3.3.3 预置 Hotseat 图标	14
3.4 图标客制化	14
3.4.1 双行显示	14

3.4.2 Hotseat 应用图标名称显示.....	15
3.4.3 隐藏图标.....	17
3.4.4 图标形状.....	19
3.5 Search 框客制化.....	19
3.5.1 移除 Smartspace 小部件.....	20
3.5.2 Search 框移到顶部.....	21
3.5.3 移除 Search 框.....	22
3.5.4 Search 框可拖拽.....	23
3.5.5 替换 Search 框.....	24
3.6 待机界面.....	24
3.6.1 待机箭头.....	24
3.6.2 待机引导动画.....	25
3.6.3 待机界面上下阴影.....	25
3.7 应用抽屉.....	26
3.7.1 背景透明.....	26
3.7.2 定制 App 在应用抽屉位置.....	27
3.7.3 模糊查找.....	28
3.8 资源 Overlay.....	28
3.8.1 RRO 预置样例.....	29
4 功能客制化.....	31
4.1 桌面样式.....	31
4.2 桌面网格.....	32

4.3 图标角标.....	34
4.3.1 通知标记增强.....	34
4.3.2 未读标记.....	35
4.3.3 调整角标颜色.....	38
4.3.4 设置角标位置.....	38
4.4 动态图标.....	38
4.5 循环滑动.....	40
4.6 下拉手势.....	41
4.7 文件夹图标样式.....	42
4.8 Hotseat 图标自适应	44
4.9 最近任务扩展功能.....	45
4.9.1 任务锁定.....	45
4.9.2 显示内存信息.....	45
4.9.3 内存清理提示语	46
4.9.4 清除按钮显示在底部.....	47
4.10 禁止横屏.....	48

图目录

图 2.1 Launcher3QuickStepGo (左图) 和 Launcher3GolconRecents (右图)	2
图 2.2 Developer Options 功能调试开关界面.....	3
图 2.3 全面屏手势	4
图 3.1 基本布局.....	5
图 3.2 双行显示.....	15
图 3.3 Hotseat 显示标题	17
图 3.4 图标形状设置.....	19
图 3.5 Search 框显示位置	20
图 3.6 移除 Smartspace 小部件.....	21
图 3.7 替换搜索框	24
图 3.8 应用抽屉透明.....	26
图 3.9 定制 APP 位置.....	28
图 4.1 双层和单层桌面	31
图 4.2 桌面网格切换.....	33
图 4.3 通知标记增强效果	34
图 4.4 未读标记效果.....	36
图 4.5 未读标记设置.....	37
图 4.6 动态时钟和动态日历图标	39
图 4.7 动态图标设置界面	39
图 4.8 循环滑动设置开关	41
图 4.9 文件夹样式	42

图 4.10 Hotseat 图标自适应.....	44
图 4.11 任务锁定和解锁.....	45
图 4.12 显示内存信息.....	46
图 4.13 内存清理提示语.....	47
图 4.14 全部清除按钮移到底部.....	48

Unisoc Confidential For hiar

表目录

表格 1.1 缩略语.....	1
表格 3.1 Profile 文件中的 Launcher 属性	7
表格 3.2 决定应用图标和名称大小的三组配置	8
表格 3.3 appwidget 配置属性介绍.....	10
表格 3.4 favorite 配置属性介绍.....	11
表格 3.5 shortcut 配置属性介绍	11
表格 3.6 folder 配置属性介绍.....	12
表格 3.7 resolve 属性配置介绍	13
表格 4.1 桌面样式配置.....	32

Unisoc Confidential For hiar

1 前言

1.1 范围

本文主要介绍 UNISOC 平台 Launcher3 Android 10 版本模块相比 Android 9 版本的主要变化，并详细说明 Launcher3 客制化的具体方法。希望读者通过本文对 UNISOC Android 10 平台 Launcher3 模块有一定的了解，并可以完成常见的 Launcher 客制化需求。

本文档大量词汇参考《UNISOC ANDROID 10 LAUNCHER 简介》，所以为了便于理解，建议先阅读该文档，该文档可以通过 isupport 进行获取。

1.2 缩略语

名称	全称	定义
AOSP	Android Open Source Project	Android 开源项目
RRO	Runtime Resource Overlay	运行时 Overlay，简称 RRO，该机制的资源替换发生在运行时。
SRO	Static Resource Overlay	静态 Overlay，简称为 SRO，发生在编译时，需要在 Android 系统源码环境中进行配置。
单层桌面	无应用抽屉风格的桌面	移除了应用抽屉，所有应用图标都显示在待机界面
双层桌面	有应用抽屉风格的桌面	待机界面放置应用快捷方式和小部件，所有应用均存在于应用抽屉中。

表格 1.1 缩略语

1.3 参考文献

《UNISOC ANDROID 10 LAUNCHER 简介》

2 概览

Launcher3 Android 10 版本相对于 Android 9 版本，主要增加了针对 Android Go 设备的全新 Launcher3GolconRecents 形态。本章主要介绍 Android 10 Launcher3 的新特性。

2.1 Android 10 Launcher3 升级变化

在 Android 10 Launcher3 中，UI 与功能方面均有一些变化，下文会介绍几个主要的变化。

2.1.1 Launcher3GolconRecents 形态

针对 Android 10-Go 新增加的标准形态，包含了全新设计的 Recents UI 见图 2.1 (右图)，使用更小的缩略图，具有节省内存的好处。此形态下不支持手势导航功能。

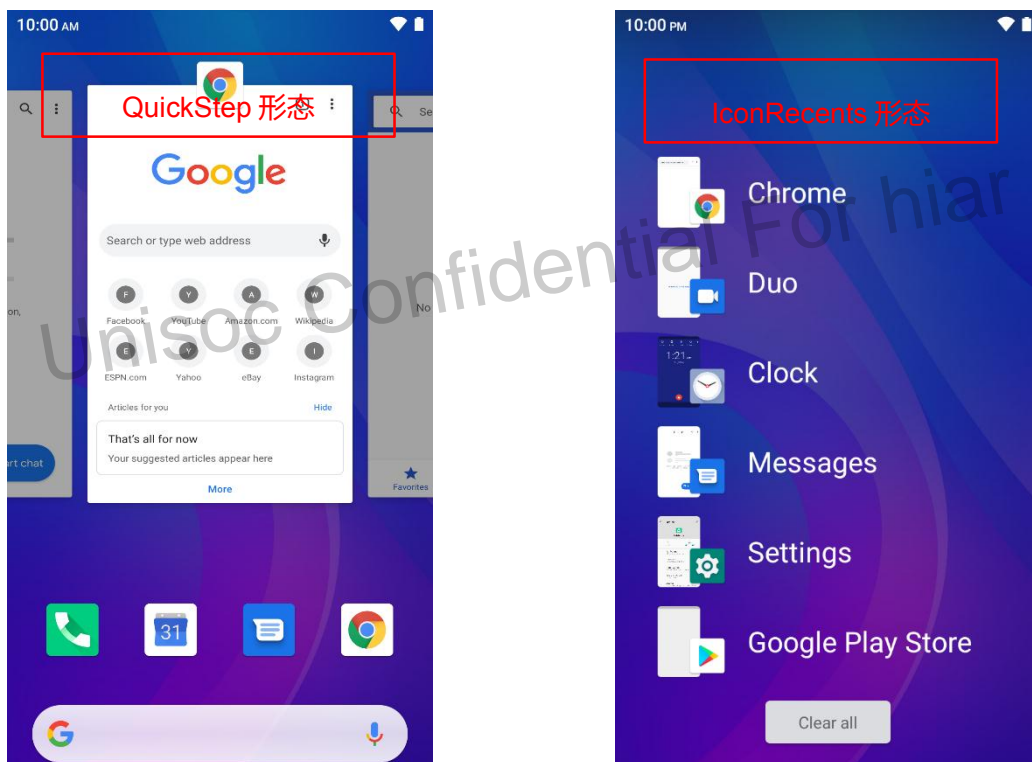


图 2.1 Launcher3QuickStepGo (左图) 和 Launcher3GolconRecents (右图)



注意：

Launcher3GolconRecents 形态下不支持 Android10 的新特性“全面屏手势导航功能”，如需支持要使用 Launcher3QuickStepGo。

2.1.2 Developer Options 功能调试开关界面

Android 10 新增特性，方便开发者调试和维护 Launcher。Debug 版本中，用户在系统设置中打开开发者选项时，主屏幕设置中会新增 Launcher 的开发者选项。该选项可以帮助开发者快速关闭和开启一些 Features。

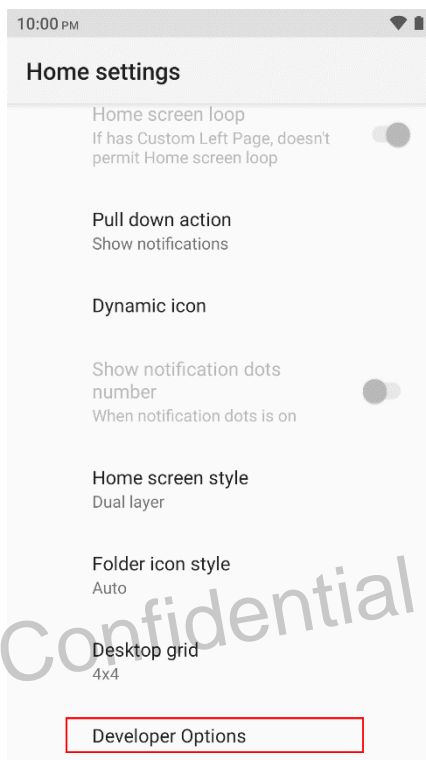


图 2.2 Developer Options 功能调试开关界面

2.1.3 Go 版本差异

Android 10-Go 版本相比 Android 10 No-Go 版本主要存在如下差异：

- 关闭长按应用图标弹出菜单
- 禁止小部件（Widget）及快捷方式（DeepShortCut）
- 关闭通知小圆点功能
- Recents UI 使用 Launcher3GolconRecents 形态（仅 512M 内存 Go 版本使用）
- 无负一屏功能

2.1.4 全面屏手势功能

从 Android 10 开始，用户可以在设置中启用手势导航，替代设备中的虚拟或实体按键，完成返回、回到待机界面、进入最近任务等操作。

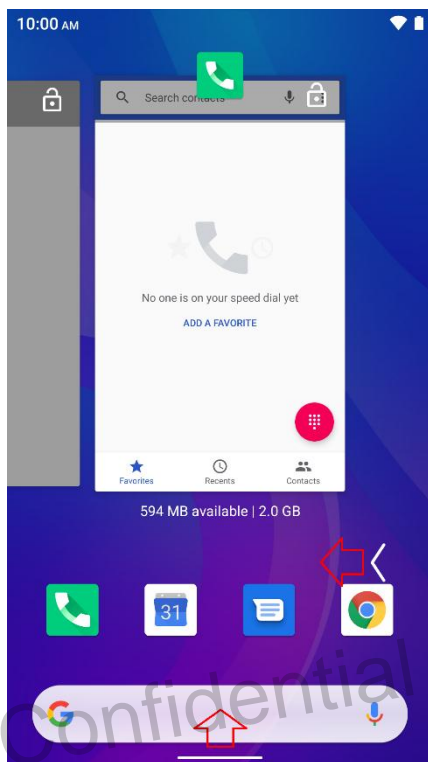


图 2.3 全面屏手势



说明：

全面屏手势功能与 Launcher 强相关，如果项目中不使用平台自带 Launcher，则设置中的全面屏手势菜单会自动隐藏。

3 UI 客制化

Launcher 待机界面是用户使用频率最高的界面，待机界面的 UI 会影响用户对设备的第一印象。因此，为了增加产品竞争力，通常需要对桌面 UI 进行一些客制化。本章会逐一介绍常用的桌面 UI 客制化方法和注意事项。

3.1 桌面布局

图 3.1 是 Launcher3 的基本布局，主要分为 Workspace（左图）和 All apps（右图）两个界面。

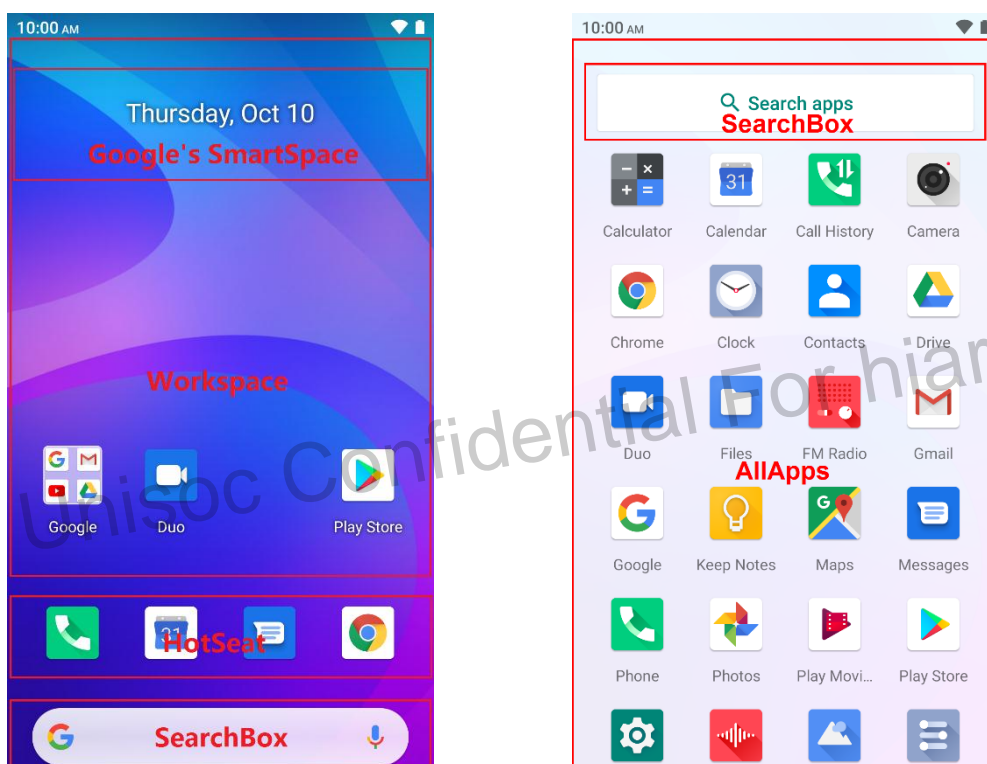


图 3.1 基本布局

3.2 布局配置

布局配置文件路径：

Go 版本：Launcher3\go\res\xml\device_profiles.xml

No-Go 版本：Launcher3\res\xml\device_profiles.xml

配置文件中定义了不同分辨率对应的 grid-option 和 display-option 标签，Launcher 根据设备分辨率自适应最佳 UI 配置，通常情况下无需调整。如需客制化，请参考 3.2.1 章节。

下面以 No-Go 版本的 profile 文件内容为例，说明各属性表示的含义，具体见表格 3.1。

```
<profile
  <grid-option
    launcher:name="4_by_5"
    launcher:numRows="5"
    launcher:numColumns="4"
    launcher:numFolderRows="4"
    launcher:numFolderColumns="4"
    launcher:numHotseatIcons="4"
    launcher:defaultLayoutId="@xml/default_workspace_4x5" >
  <display-option
    launcher:name="Nexus 5"
    launcher:minWidthDps="335"
    launcher:minHeightDps="567"
    launcher:iconImageSize="54"
    launcher:iconTextSize="13.0"
    launcher:canBeDefault="true" />
  />
/>
```

名称	
<grid-option>	网格配置项
name	名称
numRows	待机界面应用行数
numColumns	待机界面和应用抽屉应用列数
numFolderRows	文件夹中应用行数
numFolderColumns	文件夹中应用列数
numHotseatIcons	Hotseat 应用图标数量
defaultLayoutId	默认布局文件名
<display-option>	显示属性配置项
name	配置信息名称
minWidthDps	应用被允许的最小宽度，单位 dp

minHeightDps	应用被允许的最小高度，单位 dp
iconImageSize	图标大小
iconTextSize	字体大小
canBeDefault	是否可以作为默认配置

表格 3.1 Profile 文件中的 Launcher 属性



备注：

一种 grid-option 和一种 display-option 为一组，对应一种桌面布局。其中，前者决定桌面应用图标行列数、文件夹内图标行列数、Hotseat 区域图标数目以及桌面预置图标位置；后者决定桌面应用图标和应用名称的大小。

3.2.1 布局配置客制化

1. 获取当前设备使用的配置信息

设备在待机界面时，在 PC 终端执行如下命令：

Windows 环境

```
$ adb shell dumpsys activity com.android.launcher3.Launcher | findstr DeviceProfile
```

Ubuntu 环境

```
$ adb shell dumpsys activity com.android.launcher3.Launcher | grep DeviceProfile
```

以上命令在 SeachLauncherQRef 上无效，请替换类名如下：

```
com.android.searchlauncher.SearchLauncher
```

输出结果如下：

```
$ DeviceProfile: IDP [mGridName:4_by_4 mDisplayOptionName:<4_by_4,Nexus 5>&<4_by_4,Nexus 4>&<4_by_4,Nexus S> numRows:4 numColumns:4 numHotseatIcons:4 numFolderRows:3 numFolderColumns:4 iconSize:53.981415 iconShapePath:M50,0L92,0C96.42,0 100,4.58 100 8L100,92C100, 96.42 96.42 100 92 100L8 100C4.58, 100 0 96.42 0 92L0 8 C 0 4.42 4.42 0 8 0L50 0Z landscapelconSize:53.981415 iconTextSize:13.0]
```



注意：

由于 SearchLauncherQRef 形态的主 Activity 名称与其他形态不同。因此在搭配了 SearchLauncherQRef 形态的版本中（Android 10 GMS 版本），上述命令中的类名请替换成如下类名“com.android.searchlauncher.SearchLauncher”。

2. 根据配置信息客制化

以上面输出的 log 为例：

➤ mGridName 等于 4_by_4 说明当前桌面布局是由 profile 文件中 launcher:name 为 4_by_4 的 grid-option 标签指定的。

✧ 找到该 grid-option 标签项，可通过其下的各个 launcher 属性自定义桌面图标行列数、文件夹内图标行列数、Hotseat 区域图标个数以及桌面默认图标及其位置，Launcher 属性具体含义见表格 3.1。

➤ mDisplayOptionName 等于<4_by_4, Nexus 5>&<4_by_4,Nexus 4>&<4_by_4,Nexus S>说明当前桌面应用图标大小和应用名称大小是由表格 3.2 中三组配置的 iconImageSize 和 iconTextSize 共同决定的。

✧ 调整应用图标或者应用名称的大小，只需分别微调三组配置中的 iconImageSize 和 iconTextSize 即可。

grid name	display name	Grid name 的 display name 中定义应用图标和应用名称大小
4_by_4	Nexus 5	launcher:iconImageSize="48" launcher:iconTextSize="13.0"
4_by_4	Nexus 4	launcher:iconImageSize="54" launcher:iconTextSize="13.0"
4_by_4	Nexus S	launcher:iconImageSize="54" launcher:iconTextSize="13.0"

表格 3.2 决定应用图标和名称大小的三组配置



注意：

设备配置信息中的 mDisplayOptionName 打印为 mDisplayOptionName:Nexus 5 这种形式时，说明当前桌面布局即由 grid name 为 xxx 且 display name 为 Nexus 5 的这组配置决定。此时只需更改这组配置中的 iconImageSize 和 iconTextSize 即可客制化桌面应用图标和应用名称大小。

3.2.2 显示大小对布局的影响

在系统设置→显示→显示大小，可以改变手机的显示大小。每次修改显示大小后 Launcher 会重新适配布局配置。由于各布局配置参数可能不同，所以修改显示大小后，可能会引起桌面行列数等布局发生变化。



备注：

如果期望修改显示大小后，桌面布局不发生变化，可以仿照 Go 版本配置，只保留一套 profile 中的 grid-option。但会导致桌面网格功能失效。

桌面网格功能开启时，更改显示大小不会影响桌面行列数。

3.3 预置图标

设备首次启动完成后，待机界面已经预置了部分图标。配置文件会被 Partner 机制替换，建议先检查是否存在 Partner app。可以通过查看 Launcher 启动时的 Log：

```
D/UNISOC-LAUNCHER3: Launcher.Partner, packageName is:com.google.android.gmsintegration
```

可以根据包名查找对应配置文件：

- **GMS 版本配置文件路径：**其中 res_XXX 请根据项目配置情况确认：

vendor\partner_gms\apps\GmsSampleIntegration\res_XXX\xml\partner_default_layout.xml

- **Native 版本配置文件路径：**

Launcher3\NativeLauncherLayout\res\xml\partner_default_layout.xml

下文均是在不存在 Partner 的情况下做的修改，如果存在 Partner，只需参考该修改即可。

默认图标是在表格 3.1 中 defaultLayoutId 配置的。对应如下文件：

Launcher3\res\xml\default_workspace_XXX.xml

如果修改此文件无效，需要考虑该文件是否被 Launcher Partner 机制替换。

3.3.1 图标预置类型介绍

预置图标是在上述 default_workspace_XXX 配置文件中添加，共 5 种标签类型。

- **appwidget(待机界面小部件)，格式如下：**

```
<appwidget
```

```
launcher:screen="0"
launcher:x="0"
launcher:y="1"
launcher:spanX="4"
launcher:spanY="2"
launcher:packageName="com.android.deskclock"
launcher:className="com.android.alarmclock.DigitalAppWidgetProvider" />
```

名称	定义
screen	页面位置，第几页
x	X 坐标位置
y	Y 坐标位置
spanX	Widget 宽度占用格子数
spanY	Widget 高度占用格子数
packageName	Widget 包名
className	Widget 类名

表格 3.3 appwidget 配置属性介绍



注意：

其他图标或 widget 的预置位置不能与本 widget 所预置的区域冲突，否则会导致加载失败。

➤ favorite(待机界面应用图标)，格式如下：

```
<favorite
launcher:screen="0"
launcher:x="2"
launcher:y="3"
launcher:packageName="com.android.settings"
launcher:className="com.android.settings.Settings" />
```

名称	定义
screen	页面位置，第几页
x	X 坐标位置
y	Y 坐标位置
packageName	包名
className	类名

表格 3.4 favorite 配置属性介绍

favorite 预置，一般预置带有 Launcher 属性的应用图标。

➤ shortcut(待机界面快捷方式图标)：格式如下：

```
<shortcut
  launcher:icon="@drawable/app_icon"
  launcher:title="@string/app_name"
  launcher:uri="http://www.baidu.com/"
  launcher:screen="0"
  launcher:x="0"
  launcher:y="0" />
```

名称	定义
icon	图标
title	名称
screen	页面位置，第几页
x	X 坐标位置
y	Y 坐标位置
uri	Uri 类型的地址

表格 3.5 shortcut 配置属性介绍

shortcut 预置，一般预置的是网址书签、应用中某个 Activity（不带 Launcher 属性）、打开某个特殊文件等快捷方式，可以自定义其 icon、title，通常建议使用“launcher:uri”标签。

例如：打开预置在某个目录的视频文件，可以使用如下 uri 格式：

```
launcher:uri="file:///storage/emulated/0/Movies/1.mp4#Intent;type=video/*;
action=android.intent.action.VIEW;category=android.intent.category.BROWSABLE;end"
```

若需指定系统默认播放器就增加 component=com.android.gallery3d/app.MovieActivity;这一项，若需用户选择播放器，就将这项删除掉。



注意：

使用 shortcut 标签，必须设置 title，否则预置会失效。

➤ folder(待机界面文件夹)，格式如下：

```
<folder
  launcher:title="@string/folder_name"
  launcher:screen="0"
  launcher:x="0"
  launcher:y="3">
  <favorite
    launcher:packageName="com.android.settings"
    launcher:className="com.android.settings.Settings"
    launcher:x="0"/>
  <favorite
    launcher:packageName="com.android.settings"
    launcher:className="com.android.settings.Settings"
    launcher:x="1"/>
</folder>
```

名称	定义
title	名称
screen	页面位置，第几页
x	X 坐标位置
y	Y 坐标位置
favorite	文件夹中预置的图标，详见表格 3.4，此时 x 代表文件夹中的排序

表格 3.6 folder 配置属性介绍

folder 预置，不能包含 appwidget 类型。

➤ resolve(待机界面图标)，格式如下：

```
<resolve
  launcher:container="-101"
  launcher:screen="1"
  launcher:x="1"
  launcher:y="0" >
  <favorite
    launcher:uri="#Intent;action=android.intent.action.MAIN;category=android.intent.category.APP_MESSAGING;end" />
    <favorite launcher:uri="sms:" />
    <favorite launcher:uri="smsto:" />
    <favorite launcher:uri="mms:" />
    <favorite launcher:uri="mmsto:" />
  </favorite>
</resolve>
```

名称	定义
container	图标的容器， -101 代表 Hotseat， -100 代表 Workspace
screen	页面位置，也是 Hotseat 自左到右的顺序位置
x	X 坐标位置，也是 Hotseat 自左到右的顺序位置，与 screen 一致
y	Y 坐标位置，由于 Hotseat 只有一行，故默认一直是 0
favorite	这里一般指地址，详见表格 3.4

表格 3.7 resolve 属性配置介绍

resolve 预置，一般预置需要自适应的应用图标，favorite 中配置 uri。

3.3.2 预置 workspace 图标

用户可以直接修改如下 worksapce 待机界面图标布局文件。

packages\apps\Launcher3\res\xml\default_workspace_XXX.xml

```
<favorite
  launcher:screen="0"
  launcher:x="-1"
  launcher:y="-1"
  launcher:packageName="com.android.settings"
  launcher:className="com.android.settings.Settings" />
```



说明：

其中 x 轴或者 y 轴位置若是“-1”，代表待机界面 x 轴或者 y 轴的最后一个位置。



注意：

如 3.2 章节注意事项，如果存在 Parnter app，请在 Parnter app 中修改待机界面默认布局文件。

3.3.3 预置 Hotseat 图标

Hotseat 图标位于待机界面最下面的一行，通常预置常驻应用，是在如下文件配置的。

Launcher3\res\xml\default_workspace_XXX.xml

```
<resolve
  launcher:container="-101"
  launcher:screen="1"
  launcher:x="1"
  launcher:y="0" >
  <favorite
    launcher:uri="#Intent;action=android.intent.action.MAIN;category=android.intent.category.APP_MESSAGING;end" />
  <favorite launcher:uri="sms:" />
  <favorite launcher:uri="smsto:" />
  <favorite launcher:uri="mms:" />
  <favorite launcher:uri="mmsto:" />
</resolve>
```

Hotseat 图标预置使用的是 resolve 标签和 uri 属性，好处是可以自适应。例如用户预置了其他第三方短信应用而没有使用原生短信应用，则该位置会根据 resolve 标签和 uri 属性解析出第三方短信应用，进而填充 Hotseat 图标。当然也可以使用 favorite、shortcut、folder 标签，请注意坐标和包名、类名的准确性。

3.4 图标客制化

对图标进行客制化，请参考 3.2 章节，确认设备使用的是哪一个 grid-option 标签，然后调整属性即可。

其他复杂的情况，则需要调整代码来实现，例如双行显示、Hotseat 应用图标名称显示和隐藏图标等。

3.4.1 双行显示

➤ Feature 简介

当应用名称太长的时候，可根据需要开启双行显示。如图 3.2 所示，修改后的效果如下：

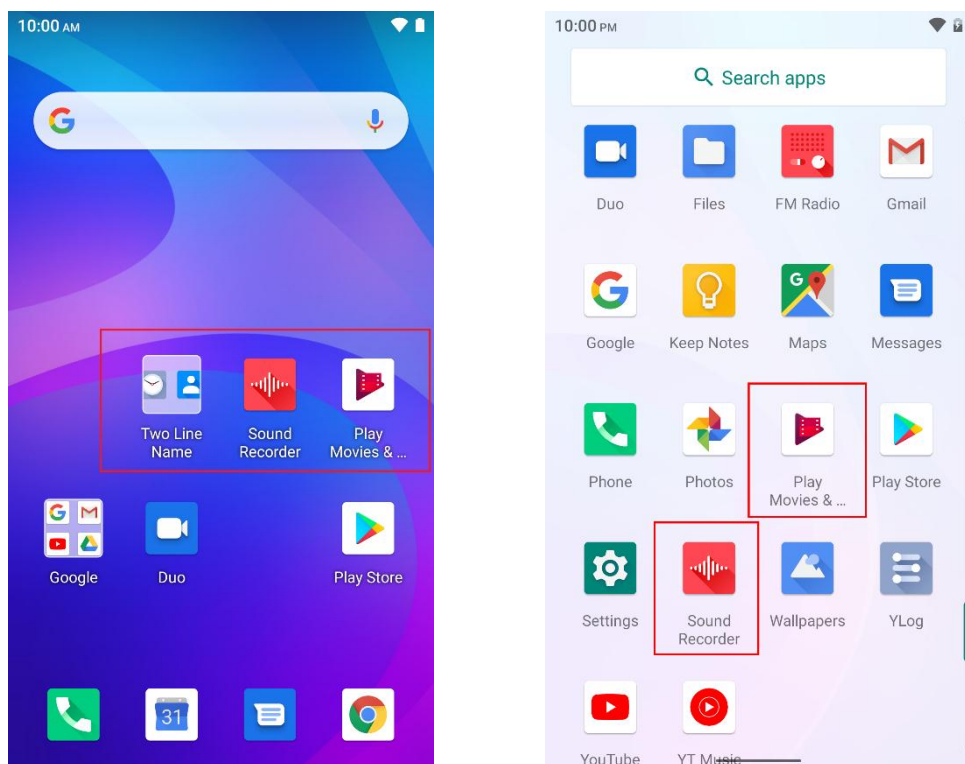


图 3.2 双行显示

➤ Feature 开关

本 feature 通过如下 prop 属性控制，可根据项目需求使用该开关开启或者关闭本 feature。

ro.launcher.iconlabelline

➤ Feature 默认状态

此功能默认关闭。

➤ Feature 配置

在功能开关开启的情况下，可以通过如下配置调整是否默认双行显示，在主屏幕设置中打开或者关闭双行显示。

Launcher3\res\values\config_ext.xml

```
<!-- add for app icon label display double lines -->
<bool name="enable_icon_label_show_double_lines">false</bool>
```

3.4.2 Hotseat 应用图标名称显示

Hotseat 区域，应用图标名称默认是隐藏的，若需显示，请做如下修改：

➤ 显示 Hotseat 应用图标名称

Launcher3\src\com\android\launcher3\CellLayout.java

```
// Hotseat icons - remove text
if (child instanceof BubbleTextView) {
    BubbleTextView bubbleChild = (BubbleTextView) child;
-    bubbleChild.setTextVisibility(mContainerType != HOTSEAT);
+    bubbleChild.setTextVisibility(true);
}
```

- 显示 Hotseat 区域文件夹图标名称

Launcher3\src\com\android\launcher3\WorkspaceLayoutManager.java

```
// Hide folder title in the hotseat
if (child instanceof FolderIcon) {
-    ((FolderIcon) child).setTextVisible(false);
+    ((FolderIcon) child).setTextVisible(true);
}
```

- 拖拽动画结束后，显示 Hotseat 区域应用名称

Launcher3\src\com\android\launcher3\BubbleTextView.java

```
public boolean shouldTextBeVisible() {
-    // Text should be visible everywhere but the hotseat.
-    Object tag = getParent() instanceof FolderIcon ? ((View) getParent()).getTag() : getTag();
-    ItemInfo info = tag instanceof ItemInfo ? (ItemInfo) tag : null;
-    return info == null || info.container != LauncherSettings.Favorites.CONTAINER_HOTSEAT;
+    return true;
}
```

- 调整 Hotseat 图标大小

修改完成后，若发现 Hotseat 图标名称显示不全，需要调整 Hotseat 大小，使图标名称显示完全。

Launcher3\res\values\dimens.xml

```
-    <dimen name="dynamic_grid_hotseat_extra_vertical_size">34dp</dimen>
+    <dimen name="dynamic_grid_hotseat_extra_vertical_size">48dp</dimen>
```

请根据实际情况调整该值。

如图 3.3 所示，修改前后的效果图如下：

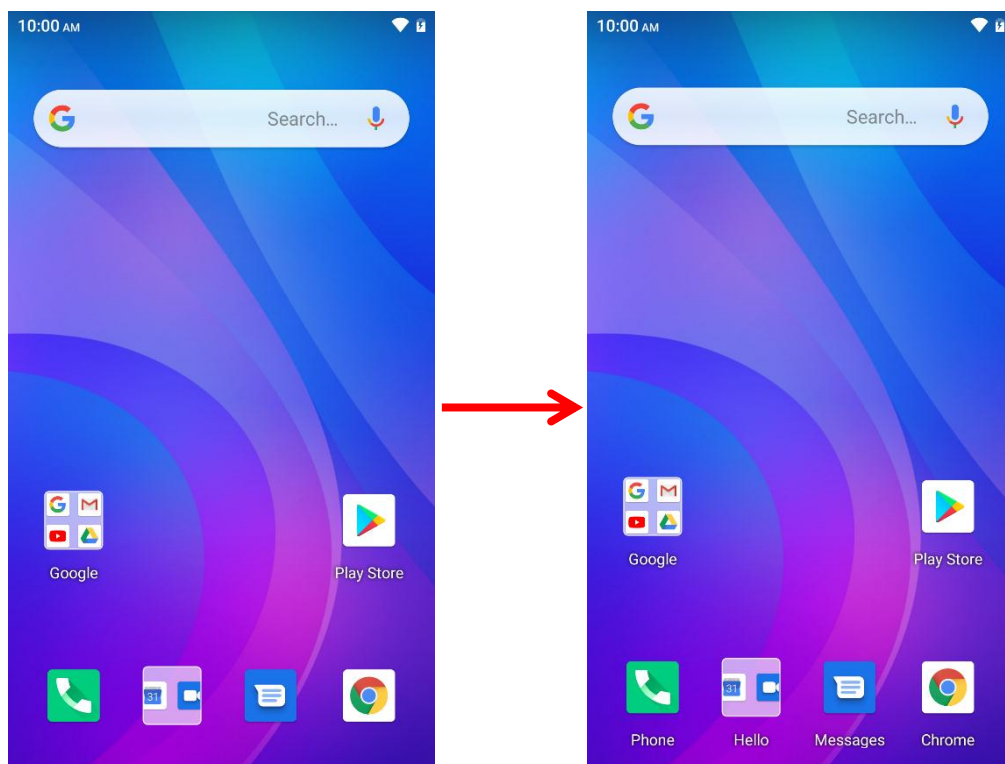


图 3.3 Hotseat 显示标题



注意：

由于 Hotseat 上的图标为用户最常用的图标，google 设计为通过图标来辨识应用，所以隐藏掉了 Hotseat 图标的 title。非特殊情况，不建议显示该 title，因为显示 Hotseat title 会导致待机界面可用区域变小，甚至会导致某些 widget 无法显示完全，请慎重修改。

3.4.3 隐藏图标

在 Launcher 中隐藏某个应用图标及其 widget，以达到预期效果。目前，Launcher 已经预留了隐藏某个应用的接口类，具体如下：

- 双层 Launcher 下隐藏图标：

Launcher3\src\com\android\launcher3\AppFilter.java

- 新增一个类继承于接口类：

例如：Launcher3\src\com\android\launcher3\CustomAppFilter.java

```
package com.android.launcher3;
import android.content.ComponentName;
import android.content.Context;

public class CustomAppFilter extends AppFilter {
    public CustomAppFilter(Context context) {
    }

    @Override
    public boolean shouldShowApp(ComponentName app) {

        // 实现此方法即可过滤相应的应用图标

        return true;
    }
}
```

- 在配置文件中配置该类：

Launcher3\res\values\config.xml

```
<!-- Various classes overridden by projects/build flavors. -->
- <string name="app_filter_class" translatable="false"></string>
+ <string name="app_filter_class"
translatable="false">com.android.launcher3.CustomAppFilter</string>
```

- 混淆：该子类为隐式启动，其他地方没有调用，在编译过程中会被忽略。需要做如下修改，避免被忽略。

Launcher3\proguard.flags

```
+ -keep class com.android.launcher3.CustomAppFilter {
+     *;
+ }
```

- 单层 Launcher 下隐藏图标：

双层 Launcher 下隐藏图标的基础上增加如下代码：

src/com/sprd/ext/multimode/VerifyIdleAppTask.java

```
+ import com.android.launcher3.AppFilter;

- if (matcher.filterItemInfos(LauncherModel.getBgDataModel().workspaceItems).isEmpty()) {
+ AppFilter appFilter = AppFilter.newInstance(mContext);
+ if (matcher.filterItemInfos(LauncherModel.getBgDataModel().workspaceItems).isEmpty()
+     && appFilter.shouldShowApp(componentKey.componentName)) {
        Object obj = entry.getValue();
        if (obj instanceof AppInfo) {
            verifyShortcutHighRes(context, (AppInfo) obj);
        }
    }
```

3.4.4 图标形状

➤ Feature 简介

Android 10 可以在设置→系统→高级→开发者选项→“图标形状”对图标形状进行设置。设置后全局图标形状都会随之改变，如图 3.4 所示，是该功能的使用步骤和效果。

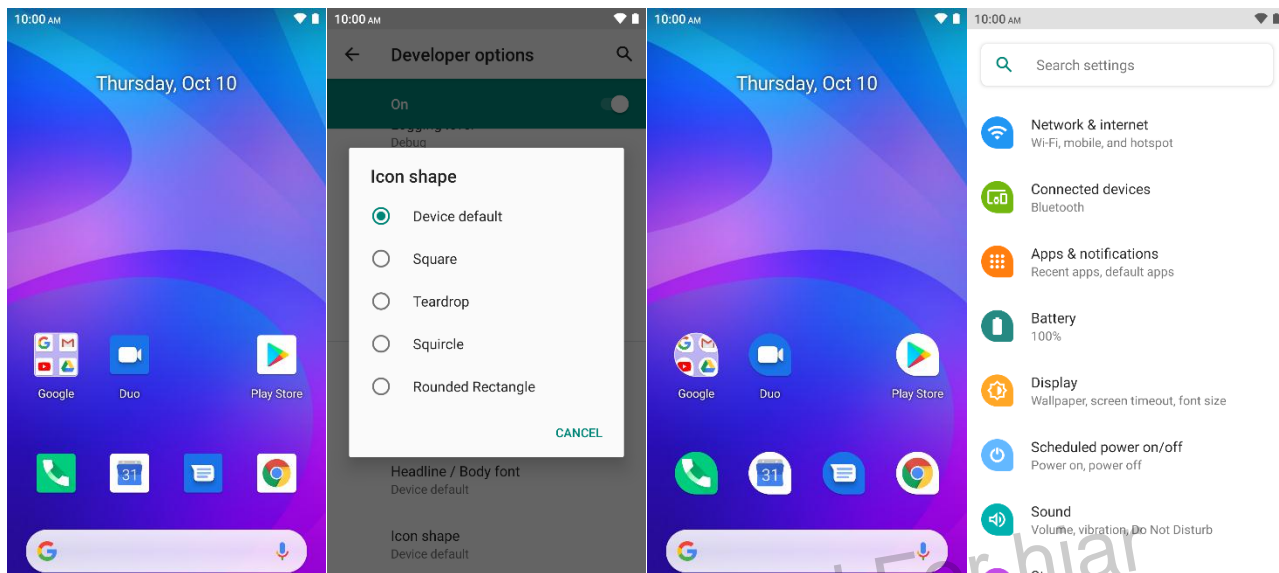


图 3.4 图标形状设置

系统默认图标形状客制化，可以使用 RRO 方式做如下修改：

frameworks\base\packages\overlays\IconShapeXxxxOverlay\AndroidManifest.xml

```
<overlay android:targetPackage="android"
    android:category="android.theme.customization.adaptive_icon_shape"
+   android:isStatic="true"
    android:priority="1"/>
```

3.5 Search 框客制化

首页 search 框为 google 的强制要求，非特殊原因不建议移除、修改，否则会导致无法通过 Google GTS 测试。

如图 3.5 所示，Search 框有两种显示方式

- ✧ SearchLauncherQRef 的 Search 框默认是在待机界面底部显示。
- ✧ 除 SearchLauncherQRef 形态之外，其它 Launcher 形态的 Search 框均显示在待机首页顶部。

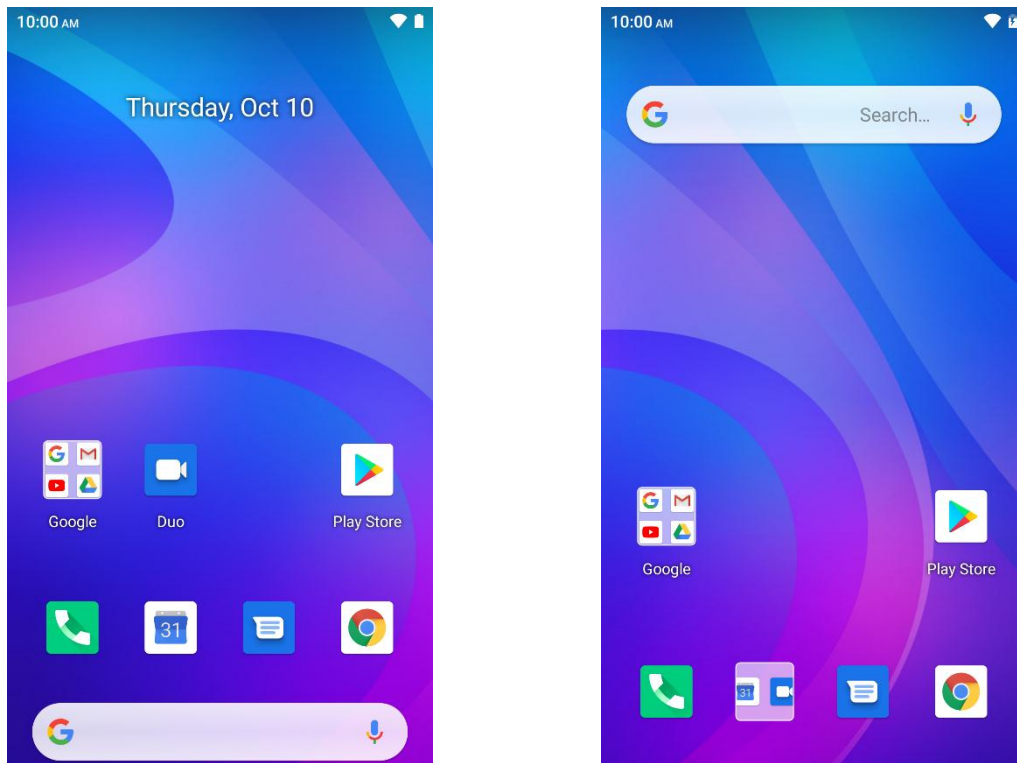


图 3.5 Search 框显示位置

3.5.1 移除 SmartSpace 小部件

SearchLauncherQRefp 上方有个固定的 SmartSpace 小部件如图 3.5 (左图), 替换了 Launcher3 中的 Search 框小部件, 是无法拖动、移除的。

1. 移除此小部件：

vendor\partner_gms\apps\SearchLauncher\quickstep\src\com\android\launcher3\config\FeatureFlags.java

```
private FeatureFlags() {}
```

```
+ public static final boolean QSB_ON_FIRST_SCREEN = false;
```

2. 移除“主屏幕设置”中的“At a glance”：

vendor\partner_gms\apps\SearchLauncher\res\xml\launcher_preferences.xml

```
- <androidx.preference.PreferenceScreen
-     android:key="pref_smartspace"
-     android:persistent="false"
-     android:summary="@string/smartspace_preferences_in_settings_desc"
-     android:title="@string/smartspace_preferences_in_settings"/>
```

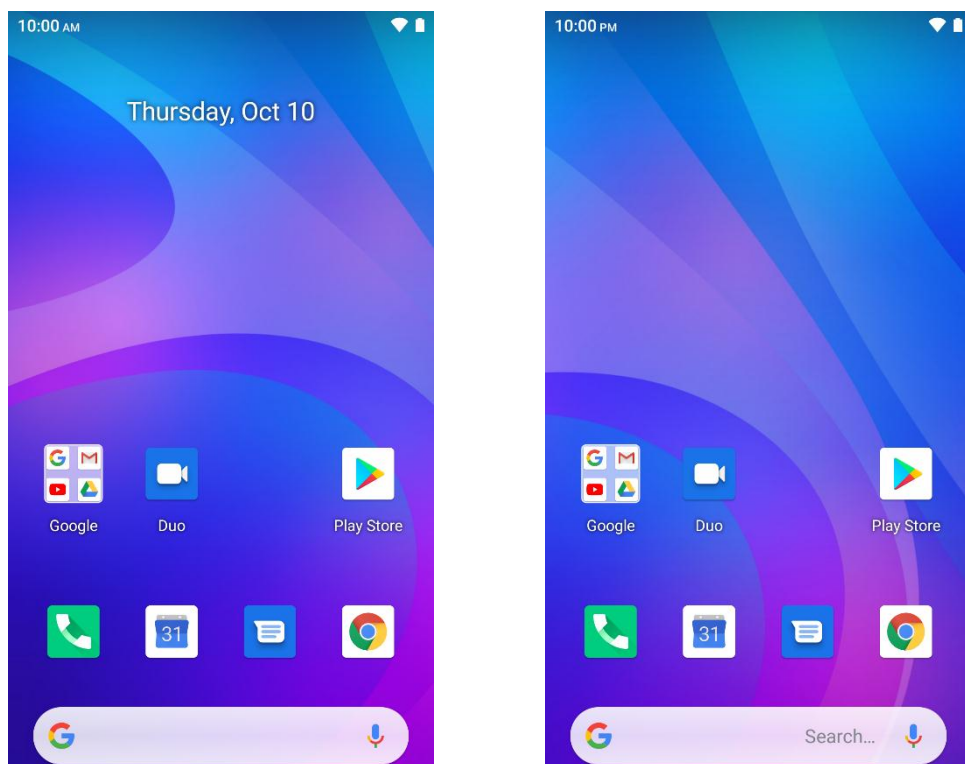


图 3.6 移除 SmartSpace 小部件

3.5.2 Search 框移到顶部

SearchLauncherQRef 待机界面 Search 框默认显示在底部 Hotseat 区域，如图 3.5（左图）。

如果需要 Search 框显示在顶部，需要将顶部固定的日期小部件和底部的 Search 框移除，并默认使用

Launcher3 中的 Search 框，请做如下修改：

删除如下文件：

vendor/partner_gms/apps/SearchLauncher 目录下：

// 删除底部 Search 框的布局

deleted: quickstep/res/layout/search_container_all_apps.xml

// 删除顶部日期小部件的容器布局

deleted: quickstep/res/layout/search_container_workspace.xml

// 删除顶部日期小部件的布局

deleted: quickstep/res/layout/smart_space_date_view.xml

// 删除 overlay 的 Hotseat 大小以及搜索框的配置值，目前有 4 项

deleted: quickstep/res/values/dimens.xml


```
<dimen name="dynamic_grid_hotseat_size">70dp</dimen>
<dimen name="dynamic_grid_hotseat_top_padding">0dp</dimen>
<dimen name="dynamic_grid_hotseat_bottom_padding">72dp</dimen>
<dimen name="search_widget_top_shift">24dp</dimen>

// 删除设置界面 overlay 的配置

deleted:    quickstep/res/values/settings_overrides.xml

// 删除布局对应的 java 文件

deleted:    quickstep/src/com/android/searchlauncher/HotseatQsbWidget.java
deleted:    quickstep/src/com/android/searchlauncher/QuickstepSettingsFragment.java
deleted:    quickstep/src/com/android/searchlauncher/SmartSpaceHostView.java
deleted:    quickstep/src/com/android/searchlauncher/SmartSpaceQsbWidget.java
```



注意：

所删除的文件，可能由于 GMS 包的版本不同而导致修改失败，修改思路就是将 SearchLauncher 中对 Search 框进行 overlay 的相关文件移除，然后修改 Hotseat 的高度。

3.5.3 移除 Search 框

移除 Search 框分为两种情况，一种是在 GMS 版本的 SearchLauncherQRef 中移除，另一种是在非 GMS 版本的 Launcher3 中移除。

- 在 GMS 版本 SearchLauncherQRef 中移除：

删除如下文件，移除后主菜单中的搜索框位置会有稍许变动：

```
vendor/partner_gms/apps/SearchLauncher 目录下：

deleted:    quickstep/res/layout/search_container_all_apps.xml
deleted:    quickstep/res/values/dimens.xml
deleted:    quickstep/src/com/android/searchlauncher/HotseatQsbWidget.java
```

- 在非 GMS 版本 Launcher3 中移除：

这种情况下移除比较简单，只需要在：

Launcher3\src\com\android\launcher3\config\BaseFlags.java

// Enable moving the QSB on the 0th screen of the workspace

```
- public static final boolean QSB_ON_FIRST_SCREEN = true;
+ public static final boolean QSB_ON_FIRST_SCREEN = false;
```

3.5.4 Search 框可拖拽

目前待机界面默认 Search 框是固定显示的，无法拖拽删除。若需要 Search 框可移动删除，需要做如下修改：

1. 移除固定的 Search 框小部件

方法可以参考 3.5.1 章节、3.5.3 移除 Search 框章节 将固定的元素移除。

2. 添加可以拖动的搜索小部件，请保证小部件的包名类名正确

小部件的包名类名可以通过拖动一个小部件到桌面，然后导出并查看 Launcher 数据库获得。

- ✧ 对于无 Partner app 的版本，定位到：

Launcher3\res\xml\default_workspace_XX.xml

```
<appwidget
    launcher:screen="0"
    launcher:x="0"
    launcher:y="0"
    launcher:spanX="4"
    launcher:spanY="1"
    launcher:packageName="com.google.android.googlequicksearchbox"
    launcher:className="com.google.android.googlequicksearchbox.SearchWidgetProvider" />
```

- ✧ 对于有 Partner app 的版本，定位到：

Partner app 下的\xml\partner_default_layout.xml

```
<appwidget
    screen="0"
    x="0"
    y="0"
    spanX="4"
    spanY="1"
    packageName="com.google.android.googlequicksearchbox"
    className="com.google.android.googlequicksearchbox.SearchWidgetProvider"/>
```



注意：

此修改可能会导致无法通过 Google GTS 测试。

3.5.5 替换 Search 框

以替换为 chrome search 框为例，如图 3.7 所示，修改前后效果如下：

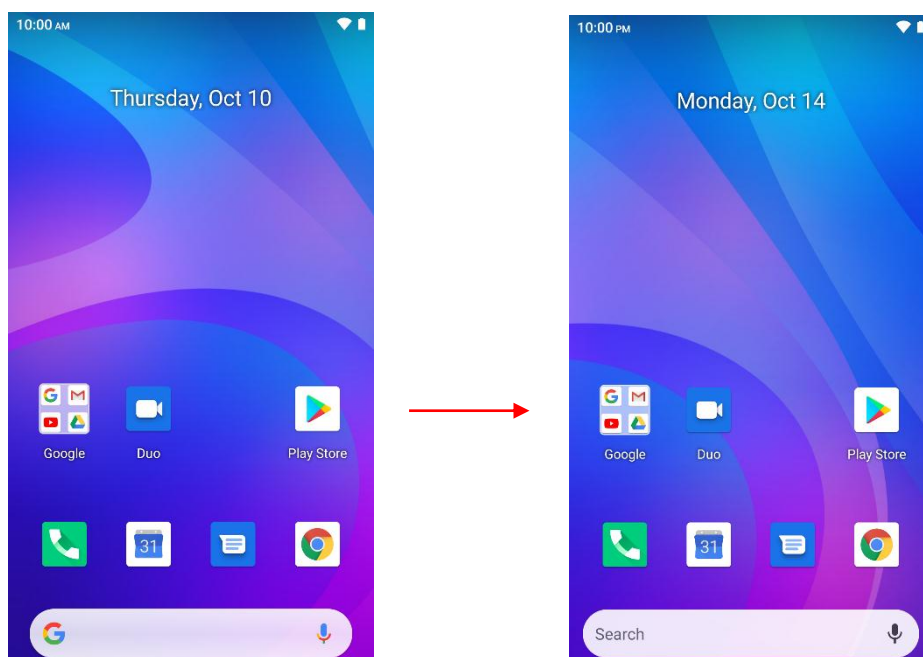


图 3.7 替换搜索框

方案修改如下：

Launcher3\res\values\config_ext.xml

```
+ <!--add for use custom search widget-->
+ <string name="custom_search_component" translatable="false">
com.android.chrome/org.chromium.chrome.browser.searchwidget.SearchWidgetProvider</string>
```

若需要修改为其他 search 框，请自行调整 `custom_search_component`。

 注意：

配置 config 开关，可以使用静态 overlay 的方式开启此该功能，也可以通过 RRO 的方式开启。请注意保证 Chrome 浏览器的包名类名正确。

3.6 待机界面

待机界面是指开机解锁后或者按 Home 键后进入的界面。

3.6.1 待机箭头

在 Launcher 待机界面 workspace 下面中间区域有一个小箭头，该箭头提示用户可以通过向上滑动进

入应用抽屉界面。目前 Android 10 版本，该箭头默认只在横屏模式下或者设置→无障碍→无障碍功能菜单开启的情况下才会显示，且在单层桌面下不允许显示。

若需要默认显示，修改如下：

Launcher3\src\com\android\launcher3\views\ScrimView.java

```
+ private static final boolean ALWAYS_SHOW_DRAGHANDLE = true;
+
private void updateDragHandleVisibility(Drawable recycle) {
    boolean visible = mLauncher.getDeviceProfile().isVerticalBarLayout() || mAM.isEnabled();
+   visible |= ALWAYS_SHOW_DRAGHANDLE;
    visible &= !MultiModeController.isSingleLayerMode();
    boolean wasVisible = mDragHandle != null;
    if (visible != wasVisible) {
```

3.6.2 待机引导动画

首次开机后，Hotseat 区域会连续向上跳动六次，提示用户可以通过向上滑动进入应用抽屉界面，这是一个进入应用抽屉引导动画。一旦用户进入过应用抽屉后，该动画就不会再播放。

该动画是在如下文件中定义的，若需修改效果，请自行在该文件中修改。

packages\apps\Launcher3\res\animator-v23\discovery_bounce.xml

若需关闭此动画，请定位到如下文件：

packages\apps\Launcher3\src\com\android\launcher3\allapps\DiscoveryBounce.java

```
+ private static final boolean HIDE_DISCOVERY_BOUNCE = true;
+
public static void showForHomelfNeeded(Launcher launcher) {
+   if (HIDE_DISCOVERY_BOUNCE) return;
    showForHomelfNeeded(launcher, true);
}
```

直接 return 即可关闭此动画。

3.6.3 待机界面上下阴影

待机界面从状态栏向下有一个渐变的阴影效果，若需去除此效果，请做如下修改：

Launcher3\res\values\styles.xml

```
- <item name="workspaceStatusBarScrim">@drawable/workspace_bg</item>
+ <item name="workspaceStatusBarScrim">@null</item>
```

3.7 应用抽屉

待机界面可以通过上滑操作，进入应用抽屉界面，即所有应用界面。

3.7.1 背景透明

➤ Feature 简介

调整应用抽屉背景使其可以显示壁纸。如图 3.8 所示，修改前后效果图如下，应用抽屉与待机界面

面文字颜色一致：

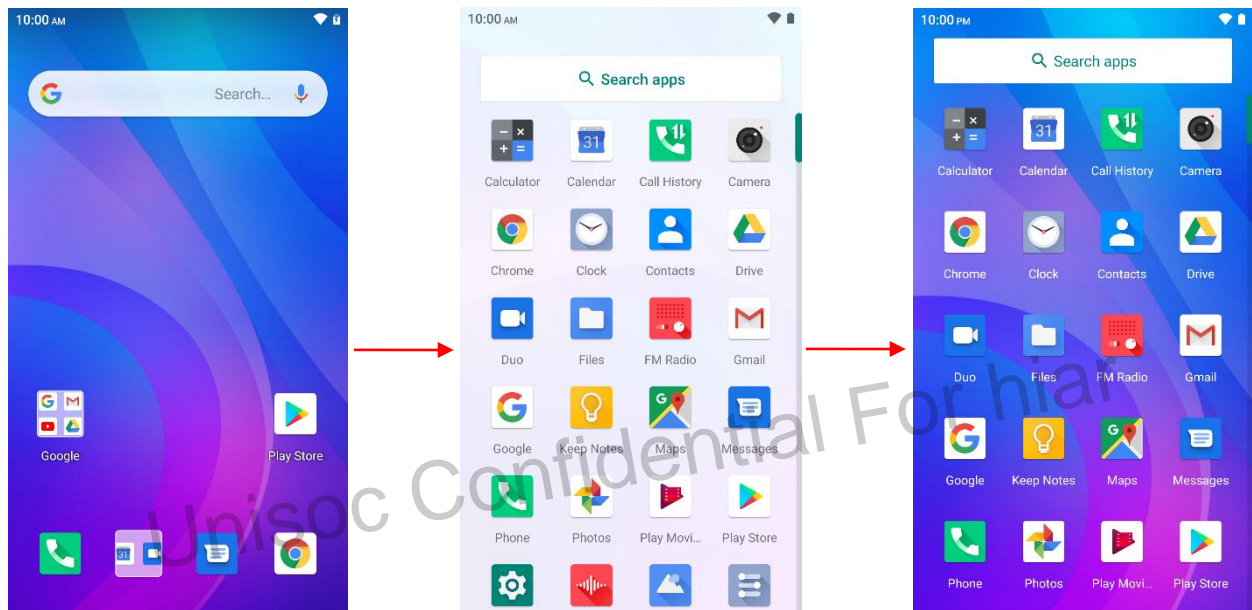


图 3.8 应用抽屉透明

➤ Feature 开关

本 feature 通过如下 prop 属性控制，可根据项目需求使用该开关开启或者关闭本 feature。

ro.launcher.allapp.bgtransp

➤ Feature 默认状态

此功能默认关闭。

➤ Feature 配置

调整透明度请定位到如下文件，透明度不要超过 200，否则会显示不美观。

Launcher3\res\values\config_ext.xml

```
<!--All apps background transparent alpha value should 0 ~ 200-->
<integer name="allapps_bg_alpha">64</integer>
```

3.7.2 定制 App 在应用抽屉位置

➤ Feature 简介

可以指定某些 app 在应用抽屉中的位置。

➤ Feature 开关

本 feature 通过如下 prop 属性控制，可根据项目需求使用该开关开启或者关闭本 feature。

ro.launcher.allapp.customize

➤ Feature 默认状态

该功能默认关闭。

➤ Feature 配置项

本功能可通过如下方式配置：

item 项格式：包名/类名#位置

修改如下配置文件：

packages/apps/Launcher3\ext\res\xml\arrays.xml

```
<string-array name="customize_app_position" translatable="false">
  <item> com.android.contacts#0</item>
  <item> com.android.settings#1</item>
</string-array>
```



注意：

类名属于可选配置，包名和位置属于必选，则位置是从 0 开始；当位置大于 all app 的数量时，则排列在最后；位置不能小于 0。

修改前后显示效果，如图 3.9 所示：

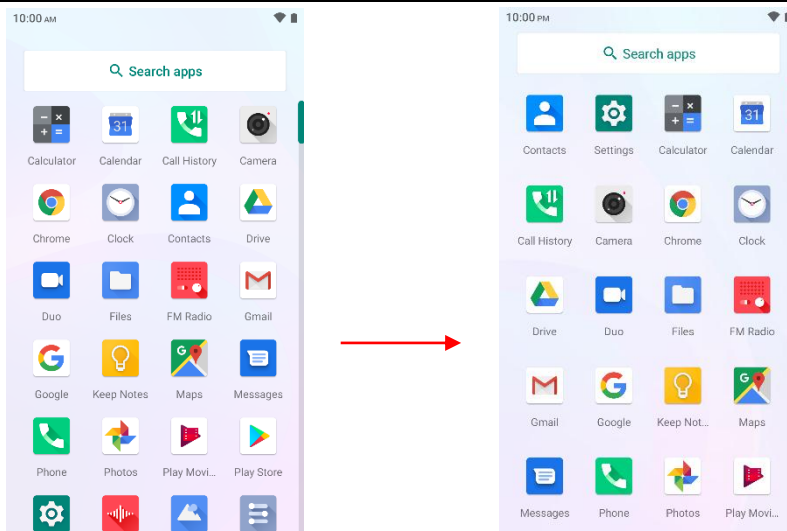


图 3.9 定制 APP 位置



注意：

该功能目前默认只有 CUCC 版本开启，且是通过 RRO 方式开启的，具体代码请参考 [/vendor/sprd/feature_configs/carriers/cucc/overlay/packages/apps/Launcher3/](#)。

3.7.3 模糊查找

➤ Feature 简介

可以根据应用名称中的任意字符，在应用抽屉界面上的搜索框查找此应用。

➤ Feature 开关：

本 feature 通过如下 prop 属性控制，可根据项目需求使用该开关开启或者关闭本 feature。

ro.launcher.allapp.fuzzysearch

➤ Feature 默认状态：

只在 Android 10 No-Go 版本开启。由于此算法会导致搜索时间变长，Android 10-Go 版本默认关闭。

3.8 资源 Overlay

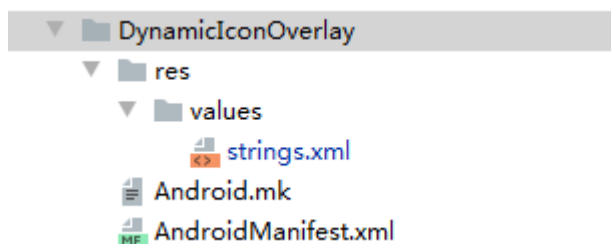
Launcher 存在多种形态，在编译时会引用不同目录下的 res 文件（具体引用可以查看 Android.mk），使用 SRO 方式替换资源，需要考虑 SRO 文件是否被引用，否则会导致 SRO 不生效。

因此：建议使用 google 推荐的 RRO 方式做 overlay。

3.8.1 RRO 预置样例

本节只简单实现一个 RRO apk 的模板，更详细介绍请参考 Google 相关文档。

样例：



1. strings.xml 中包含需要替换的字符串资源，布局文件不能被替换

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="app_name">Launcher3Overlay</string>
</resources>
```

2. Android.mk 实现将 RRO 命令行编译成 apk 应用

```
#####
# Various Dynamic Icon targets
LOCAL_PATH:= $(my-dir)

# NativeDynamicIconOverlay
include $(CLEAR_VARS)
LOCAL_PACKAGE_NAME := NativeDynamicIconOverlay
LOCAL_MODULE_OWNER := unisoc
LOCAL_MODULE_TAGS := optional
LOCAL_SYSTEM_EXT_MODULE := true
LOCAL_CERTIFICATE := platform
LOCAL_RESOURCE_DIR := $(LOCAL_PATH)/res
LOCAL_SDK_VERSION := current
include $(BUILD_RRO_PACKAGE)
```

3. AndroidManifest.xml 定义了本应用包名、替换目标包名、版本、优先级（取值范围[-1000,1000]，

默认 0，值越高优先级越高），是否是 Static 类型

```
<manifest xmlns:android=http://schemas.android.com/apk/res/android
  package="com.android.overlay.dynamiciconconfig"
  android:versionCode="1"
  android:versionName="1.0">
```

```
<overlay android:targetPackage="com.android.launcher3"  
    android:priority="0"  
    android:isStatic="true" />  
</manifest>
```

4. 最后，NativeDynamicIconOverlay 在对应项目中参与编译。

```
PRODUCT_PACKAGES += NativeDynamicIconOverlay
```

Unisoc Confidential For hiar

4 功能客制化

本章节主要对 Launcher 上新增的常用功能进行介绍。

✧ PROP 属性：ro.launcher.XXX

PROP 属性的默认配置及修改方法请参考《UNISOC ANDROID 10 LAUNCHER 简介》文档。

4.1 桌面样式

➤ Feature 简介

桌面样式主要有单层桌面、双层桌面两种(如图 4.1)。动态切换开启后用户可以在“主屏幕设置”中的“桌面样式”选择样式。

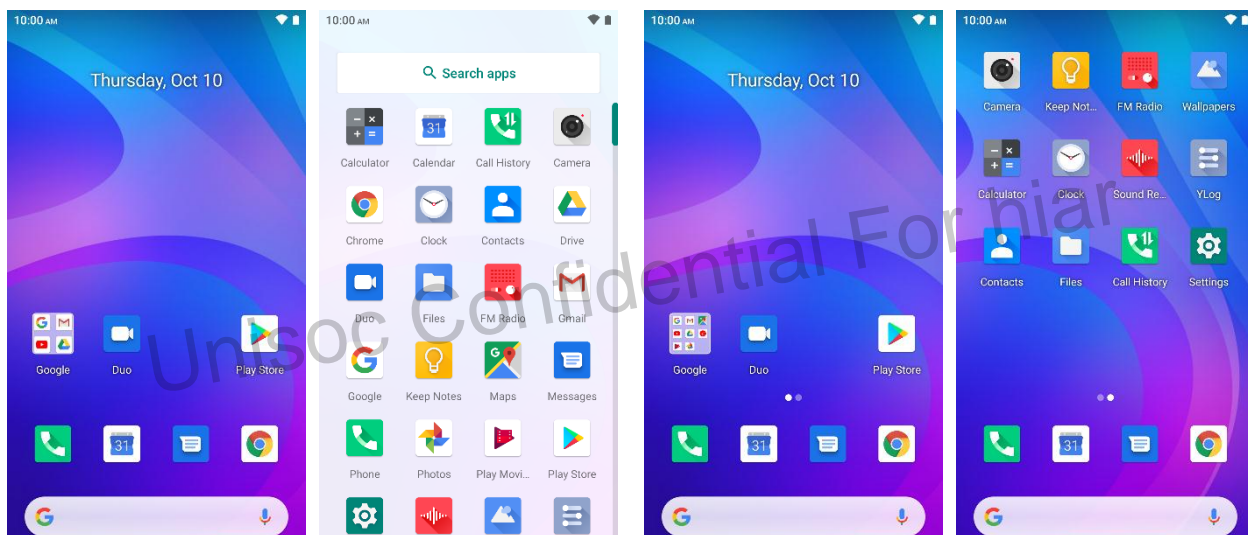


图 4.1 双层和单层桌面

➤ Feature 开关

本 feature 通过如下 prop 属性控制，可根据项目需求使用该开关开启或者关闭本 feature。

ro.launcher.multimode

➤ Feature 默认状态

✧ Android 10 No-Go 版本默认开启该功能。

✧ Android 10-Go 版本默认关闭。

➤ Feature 配置

目标模式		ro.launcher.multimode	show_home_screen_style_settings	default_home_screen_style
仅双层桌面		false	false	无需配置
仅单层桌面		true	false	single
支持动态切换	默认单层	true	true	single
	默认双层	true	true	dual

表格 4.1 桌面样式配置

主要代码目录：

Launcher3\src\com\sprd\ext\multimode

功能自定义调整：

请定位到：Launcher3\res\values\config_ext.xml

✧ 配置在“主屏幕设置”中是否显示“桌面样式”选项

```
<bool name="show_home_screen_style_settings">true</bool>
```

✧ 配置默认样式，默认值请使用 home_screen_style_values 定义的 value 值，若配置错误，则

默认为双层桌面。

```
<string name="default_home_screen_style" translatable="false">dual</string>
```



说明：

home_screen_style_values 数组在 Launcher3\res\values\arrays_ext.xml 中定义。

4.2 桌面网格

➤ Feature 简介

可在“主屏幕设置”的“桌面网格”中选择并设置桌面图标的行列数，对待机界面和主菜单界面布

局均有影响。如图 4.2 所示，从 4×4 布局调整为 5×5 布局后的效果。

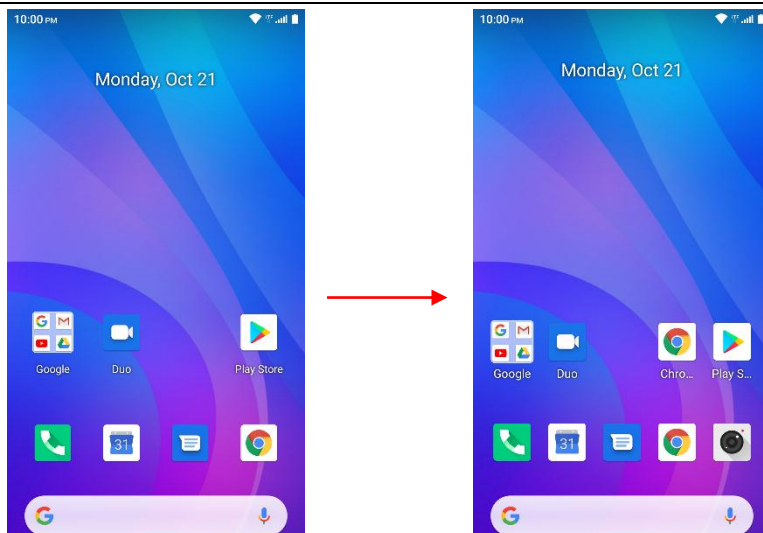


图 4.2 桌面网格切换

➤ Feature 开关

本 feature 通过如下 prop 属性控制，可根据项目需求开启或者关闭。

ro.launcher.desktopgrid

➤ Feature 默认状态

- ✧ Android 10 No-Go 版本默认开启
- ✧ Android 10-Go 版本默认关闭

➤ Feature 配置

本功能主要代码路径：

/packages/apps/Launcher3/src/com/sprd/ext/grid/

可以通过修改以下配置项，进行功能调整。

/packages/apps/Launcher3/res/values/config_ext.xml

- ✧ 自定义刷机开机后桌面默认行列数。

```
<string name="default_desktop_grid_name" translatable="false"></string>
```

- ✧ 隐藏桌面网格弹框中的某些选项，一般不建议修改此值。

```
<string-array name="grid_options_ignored" translatable="false">
    <item>3_by_3</item>
</string-array>
```



注意：

如需在 Android 10-Go 版本开启此功能，可在/vendor/sprd/generic/misc/launchercfg/LauncherFeatures.mk 中配置。

同时，请参考 No-Go 版本的 device_profiles.xml 文件，在 Go 版本的 device_profiles.xml 中至少增加一项 grid-option 选项，否则，在主屏幕设置中不会显示桌面网格设置项。

4.3 图标角标

图标角标显示在桌面图标上，主要实现方式有两种：

- 通知标记增强
- 未读标记

4.3.1 通知标记增强

- Feature 简介

通知标记增强基于 Google 原生通知圆点框架，读取通知栏中应用通知个数并在应用图标右上角的通知圆点中显示。如图 4.3 所示，AOSP 通知圆点和打开通知标记增强后的效果。

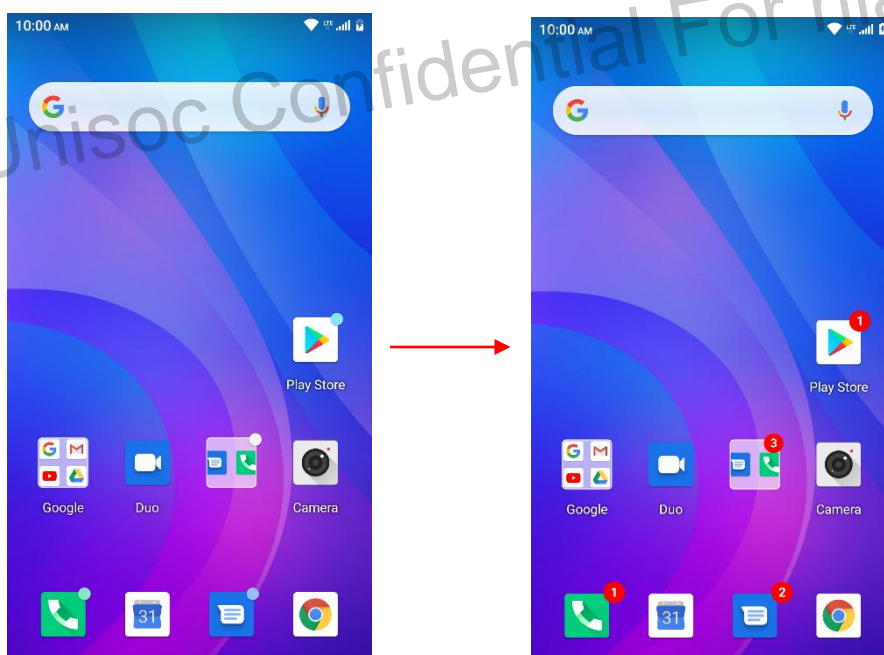


图 4.3 通知标记增强效果

- Feature 开关

本 feature 通过如下 prop 属性控制，可根据项目需求使用该开关开启或者关闭本 feature。

ro.launcher.notifbadge.count

➤ Feature 默认状态

- ✧ Android 10 No-Go 版本 Feature 默认开启。显示数目功能默认关闭，需“主屏幕设置”选项中开启“通知圆点中显示数目”。
- ✧ Android 10-Go 版本默认关闭。

➤ Feature 配置

本功能的主要代码在如下目录：

Launcher3\src\com\android\launcher3\notification
 Launcher3\src\com\android\launcher3\dot
 Launcher3\src\com\sprd\ext\notificationdots

- ✧ 可以通过修改以下配置项，默认打开通知标记增强开关。

Launcher3\res\values\config_ext.xml

```
<bool name="show_notification_dots_num">false</bool>
```



注意：

- 应用可以在发送通知时候配置 setShowBadge 来决定是否显示通知圆点。
- 应用可以通过控制通知的条数来控制通知圆点的个数显示，例如未接电话在电话应用中将通知限制成 8 条，故在 Launcher 中未读小圆点及条数最多只能显示 8（但是由于广播发送延迟，会先显示 9，被应用 remove 后又马上变为 8），如需修改最大条数限制，则需要对应模块的 app 对其条数限制进行修改。
- 由于通知发送存在延时问题，当切换语言后，应用通知小圆点会出现随通知栏中条数增加而递增的问题，具体表现为通知圆点条数从 1 变为 2、3、4……直到通知栏通知全部显示。

4.3.2 未读标记

➤ Feature 简介

未读标记是在 Launcher 上显示未接电话、未读信息、未读邮件和未读日历的数目。其中，未读邮件和未读日历仅对 UNISOC 自研 app 有效，对 GMS app 无效。如图 4.4 所示，未读标识显示效果如下。



图 4.4 未读标记效果

➤ Feature 开关

本 feature 通过如下 prop 属性控制，可根据项目需求使用该开关开启或者关闭本 feature。

ro.launcher.badge

➤ Feature 默认状态

✧ 本功能仅在 Android 10-GO 开启，但是设置项默认未勾选。

✧ Android 10-GO (512M) 关闭。

如图 4.5 所示，未读标记功能在主屏幕设置菜单中的开关，开关打开后该功能才会生效。

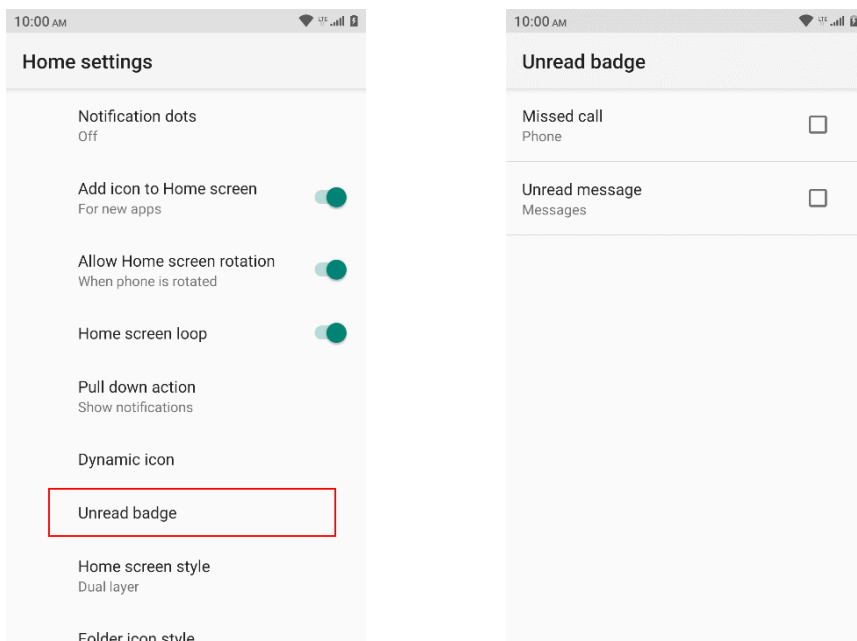


图 4.5 未读标记设置

➤ Feature 配置

本功能的主要代码在如下目录：

Launcher3\src\com\sprd\ext\unreadnotifier

在功能开关开启的情况下，配置以下 prop 属性开启或关闭未接电话、未读信息、未读邮件和未读日历四个子功能。

ro.launcher.badge.phone //未接电话

ro.launcher.badge.message //未读信息

ro.launcher.badge.email //未读邮件

ro.launcher.badge.calendar //未读日历



注意：

如图 4.5 所示，对应选项没有默认勾选，这里请务必不要默认勾选。因为默认勾选后，首次开机进入待机界面会有权限提示框。根据 google 要求，所有敏感权限必须由用户确认，不允许规避。另外，首次开机弹出权限提示框会导致 gts 测试项 fail。

4.3.3 调整角标颜色

未读标记与通知标记增强使用了同一套绘制流程，未读圆点字体颜色和背景颜色可以修改如下配置：

Launcher3\res\values\colors_ext.xml

✧ 背景颜色

```
<color name="unread_dot_bg_color">#FFFF0000</color>
```

✧ 数字颜色

```
<color name="unread_dot_text_color">#FFFFFFFF</color>
```

4.3.4 设置角标位置

圆点位置可以通过配置位置模式来进行调整。

Launcher3\src\com\sprd\ext\unreadnotifier\DotDrawUtils.java

✧ 四种位置模式

```
private enum DotMode {LT_DOT, RT_DOT, LB_DOT, RB_DOT}
```

✧ 默认使用右上角模式

```
static void drawDot(Canvas canvas, View v, String text) {
    drawDot(canvas, v, text, DotMode.RT_DOT);
}
```

✧ 也可以根据图标类型，进行位置微调：

Launcher3\res\values\dimens_ext.xml

```
<dimen name="unread_pref_customize_padding">8dp</dimen>
<dimen name="hotseat_dot_margin_x">0dp</dimen>
<dimen name="hotseat_dot_margin_y">4dp</dimen>
<dimen name="workspace_dot_margin_x">6dp</dimen>
<dimen name="workspace_dot_margin_y">6dp</dimen>
<dimen name="app_dot_margin_x">5dp</dimen>
<dimen name="app_dot_margin_y">8dp</dimen>
<dimen name="folder_dot_margin_x">5dp</dimen>
<dimen name="folder_dot_margin_y">12dp</dimen>
```

4.4 动态图标

➤ Feature 简介

本功能开启后，UNISOC 平台和 Google 平台的时钟、日历应用图标可分别实时显示当前时间和日期。如图 4.6 所示：

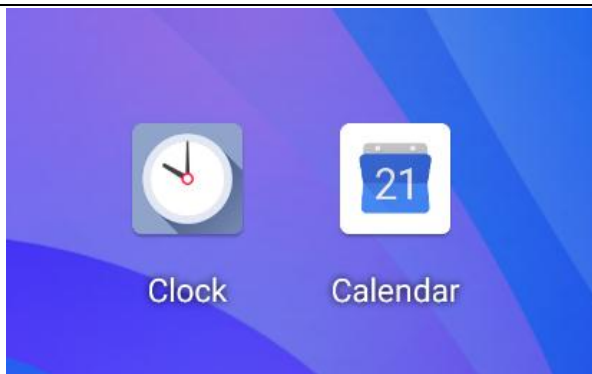


图 4.6 动态时钟和动态日历图标

➤ Feature 开关

本 feature 通过如下 prop 属性控制，可根据项目需求开启或者关闭。

ro.launcher.dynamic

➤ Feature 默认状态

Android 10 No-Go 版本中默认开启，Android 10-GO 默认关闭。

主屏幕设置中，时钟和日历的图标动态显示开关默认关闭，需用户手动开启，如图 4.7 所示。

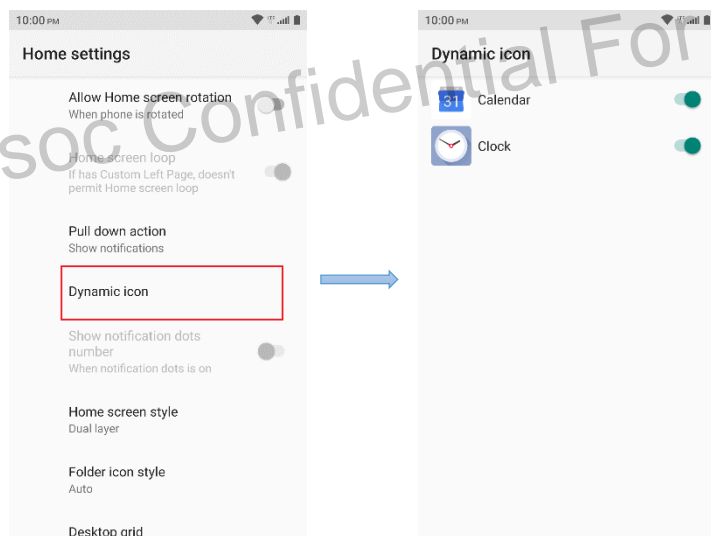


图 4.7 动态图标设置界面

➤ Feature 配置

本功能代码路径：

/packages/apps/Launcher3/src/com/sprd/ext/dynamicicon/

✧ 默认开启主屏幕设置中的动态日历、动态时钟设置开关

/packages/apps/Launcher3/res/values/config_ext.xml

```
<bool name="dynamic_calendar_default_state">false</bool>
```



```
<bool name="dynamic_clock_default_state">>false</bool>
```



注意：

动态图标会使得设备功耗增加，故强烈建议保持现有设计，即默认关闭动态图标设置开关。

✧ 关闭动态图标功能

修改文件：/vendor/sprd/generic/misc/launchercfg/LauncherFeatures.mk

ro.launcher.dynamic //动态图标功能总开关，false 表示时钟和日历图标均无法动态显示

ro.launcher.dynamic.clock //动态时钟，false 表示时钟图标不支持动态显示

ro.launcher.dynamic.calendar //动态日历，false 表示日历图标不支持动态显示

4.5 循环滑动

➤ Feature 简介

待机界面可以向左、向右循环切换页面。值得注意的是，此功能与 Google Feed 功能互斥，即当开启 Google Feed 后，不允许再开启循环滑动。

➤ Feature 开关

本 feature 通过如下 prop 属性控制，可根据项目需求使用该开关开启或者关闭本 feature。

ro.launcher.cyclescroll

➤ Feature 默认状态

本功能所有版本中默认开启，Google Feed 开启时本功能默认关闭。

➤ Feature 配置

本功能的主要代码在如下目录：

Launcher3\src\com\sprd\ext\cyclescroll

如图 4.8 所示，循环滑动功能在主屏幕设置菜单中的开关，开关打开后该功能才会生效。

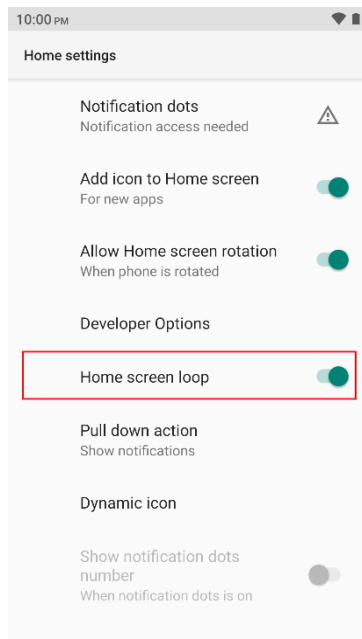


图 4.8 循环滑动设置开关

- ◇ 可以通过修改以下配置项，默认打开循环滑动开关。

Launcher3\res\values\config_ext.xml

```
<bool name="default_circle_slide">true</bool>
```



注意：

Google Feed 仅在 GMS No-Go 版本，即 SearchLauncherQRef 中存在，可在“主屏幕设置”中开启或者关闭。目前所有版本中此 Feature 默认开启，若出于内存优化的需要，可以关闭此功能。

4.6 下拉手势

➤ Feature 简介

用户可在待机界面下拉手势，在“主屏幕设置”中“下拉手势”可以选择如下几项动作：

- ◇ 无操作
- ◇ 打开通知栏
- ◇ 打开搜索界面

➤ Feature 开关

本 feature 通过如下 prop 属性控制，可根据项目需求使用该开关开启或者关闭本 feature。

ro.launcher.gesture

➤ Feature 默认状态

本功能所有版本中默认开启。

➤ Feature 配置

本功能的主要代码在如下目录：

Launcher3\src\com\sprd\ext\gestures\GesturesController.java

✧ 下拉手势默认值 Launcher3/res/values/config_ext.xml

```
<string name="default_pull_down_value" translatable="false">notification</string>
```



注意：

default_pull_down_value 的值请使用 Launcher3/res/values/arrays_ext.xml 中 pull_down_action_values 的值。

本功能与 Android 10 原生下拉状态栏功能存在冲突，因此软件中做了互斥。如需使用原生下拉状态栏功能，请在项目中关闭本 Feature

4.7 文件夹图标样式

➤ Feature 简介

如图 4.9 所示文件夹图标样式有“环形样式”和“网格样式”两种，当打开开发者模式后，用户可

以在主屏幕设置中动态调整文件夹图标的显示风格。



图 4.9 文件夹样式

➤ Feature 开关

本 feature 通过如下 prop 属性控制，可根据项目需求使用该开关开启或者关闭本 feature。

ro.launcher.foldericonmode

➤ Feature 默认状态

默认自适应桌面样式功能，单层桌面使用网格样式，双层桌面使用环形样式。

➤ Feature 配置

本功能的主要代码在如下目录

Launcher3/src/com/sprd/ext/folder

功能自定义调整：

Launcher3/res/values/config_ext.xml

```
<bool name="show_folder_icon_model_settings">true</bool>
<string name="default_folder_icon_model" translatable="false">auto</string>
<integer name="grid_folder_app_icon_size_percentage">26</integer>
<integer name="grid_folder_icon_rows">3</integer>
<integer name="grid_folder_icon_columns">3</integer>
<integer name="grid_folder_page_rows">4</integer>
<integer name="grid_folder_page_columns">3</integer>
<integer name="grid_folder_page_alpha">80</integer>
```

名称	含义
show_folder_icon_model_settings	主屏幕设置中显示文件夹样式菜单
default_folder_icon_model	默认文件夹样式，默认值请使用 packages/apps/Launcher3/res/xml/arrays_ext.xml 中 folder_icon_model_values 的 value 值
grid_folder_app_icon_size_percentage	九宫格文件夹图标缩略图中应用图标的缩放值
grid_folder_icon_rows	九宫格文件夹图标缩略图中图标的行数
grid_folder_icon_columns	九宫格文件夹图标缩略图中图标的列数
grid_folder_page_rows	九宫格文件夹展开界面的图标行数
grid_folder_page_columns	九宫格文件夹展开界面的图标列数
grid_folder_page_alpha	九宫格文件夹展开界面的背景透明度（0 ~ 255）

4.8 Hotseat 图标自适应

➤ Feature 简介

该功能在 Hotseat 图标个数发生变化后，可以自动平均分配 Hotseat 空间，重新对 Hotseat 布局。

如图 4.10 所示。

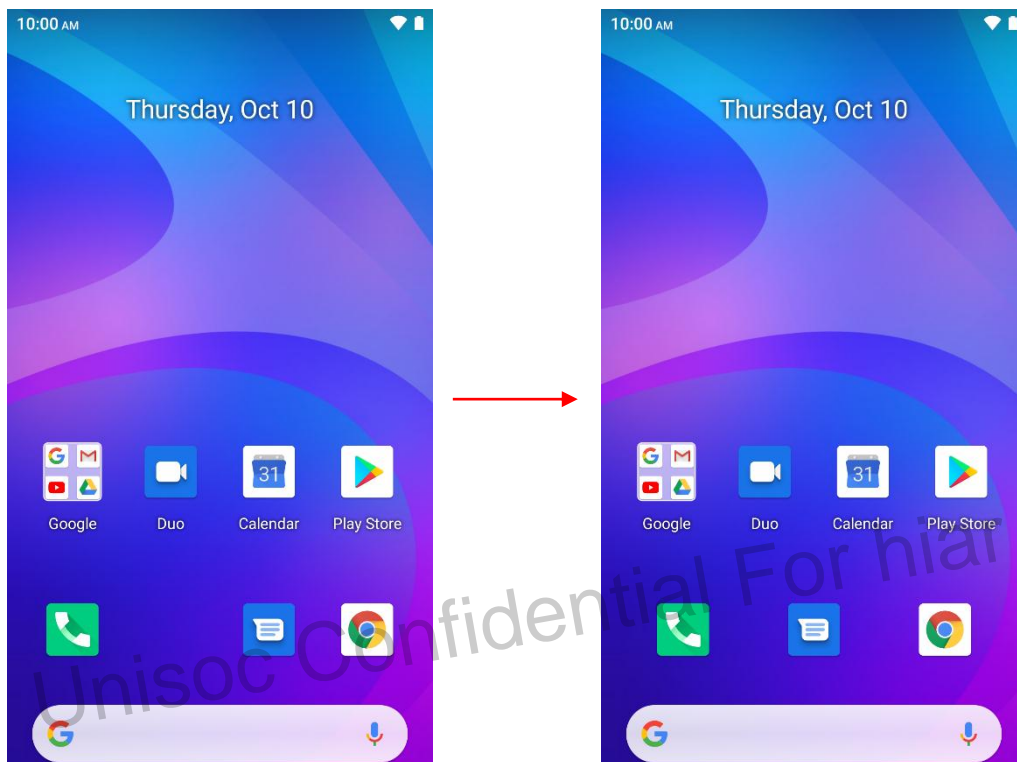


图 4.10 Hotseat 图标自适应

➤ Feature 开关

本 feature 通过如下 prop 属性控制，可根据项目需求使用该开关开启或者关闭本 feature。

ro.launcher.hs.adaptive

➤ Feature 默认状态

默认状态与 4.2 章节的桌面网格功能的状态一致。

➤ Feature 配置

本功能的主要代码在如下目录：

Launcher3/src/com/sprd/ext/grid/HotseatController.java

4.9 最近任务扩展功能

4.9.1 任务锁定

➤ Feature 简介

如图 4.11 所示，该功能允许用户通过拖动某个任务来实现对其锁定，再次拖动解锁。

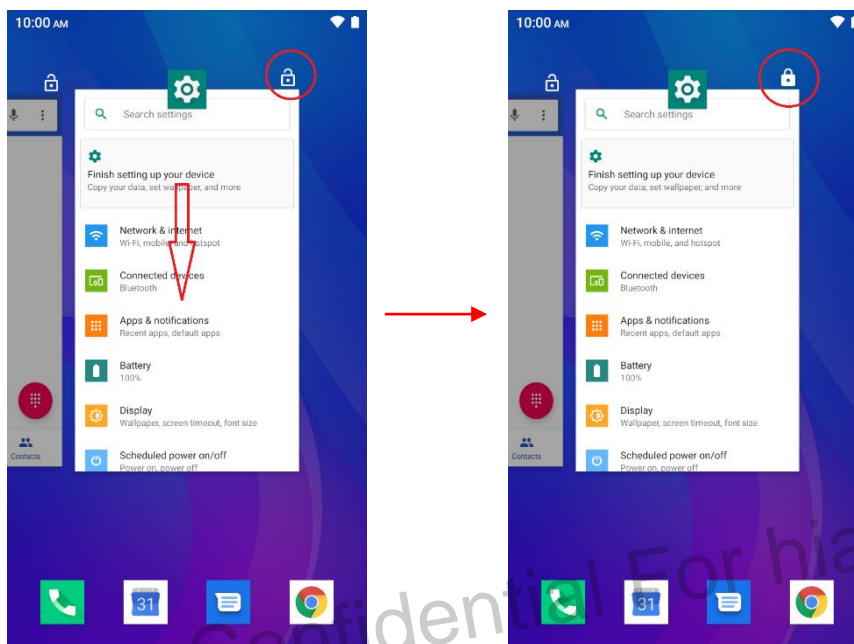


图 4.11 任务锁定和解锁

➤ Feature 开关

本 feature 通过如下 prop 属性控制，可根据项目需求使用该开关开启或者关闭本 feature。

ro.launcher.tasklock

➤ Feature 默认状态

本功能所有版本中默认开启。

➤ Feature 配置

本功能的主要代码在如下目录：

Launcher3\quickstep\src\com\sprd\ext\lockicon

若需默认关闭功能，请配置 ro.launcher.tasklock 的值为 false 即可。

4.9.2 显示内存信息

➤ Feature 简介

该功能在最近任务界面下方显示可用内存和最大内存信息。如图 4.12 所示。

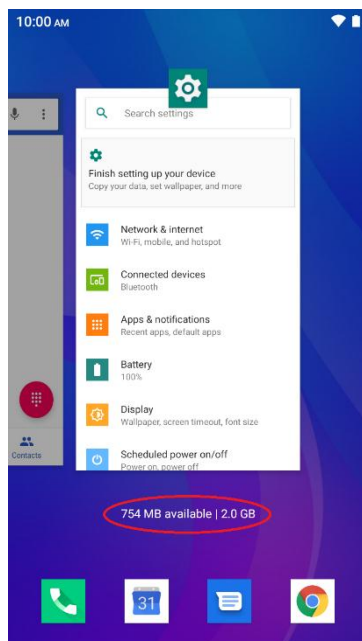


图 4.12 显示内存信息

➤ Feature 开关

本 feature 通过如下 prop 属性控制，可根据项目需求使用该开关开启或者关闭本 feature。

ro.launcher.showmeminfo

➤ Feature 默认状态

本功能所有版本中默认开启。

➤ Feature 配置

本功能的主要代码在如下目录：

Launcher3\quickstep\src\com\sprd\ext\meminfo

若需默认关闭功能，请配置 ro.launcher.showmeminfo 的值为 false 即可。

4.9.3 内存清理提示语

➤ Feature 简介

该功能在点击全部清除后，会提示释放了多少内存和剩余可用内存信息。如图 4.13 所示为操作步骤效果图。

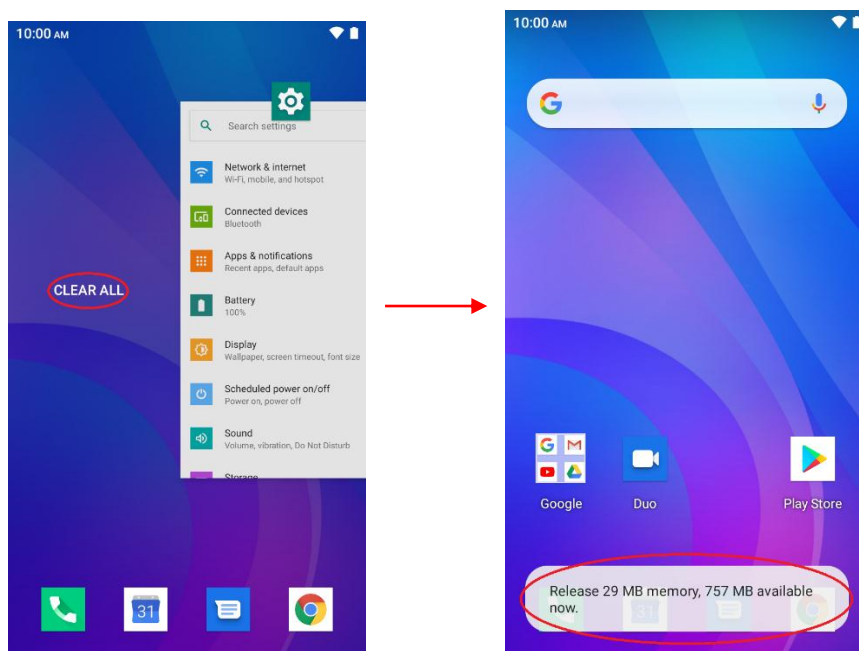


图 4.13 内存清理提示语

➤ Feature 开关

本 feature 通过如下 prop 属性控制，可根据项目需求使用该开关开启或者关闭本 feature。

ro.launcher.showclearmem

➤ Feature 默认状态

本功能所有版本中默认开启。

➤ Feature 配置

本功能的主要代码在如下目录：

Launcher3\quickstep\src\com\sprd\ext\meminfo

若需默认关闭功能，请配置 ro.launcher.showclearmem 的值为 false 即可。

4.9.4 清除按钮显示在底部

➤ Feature 简介

Quickstep 形态，全部清除按钮显示在任务列表最边上，需要滑动列表才可以显示，此功能将全部清除按钮移到列表下方，修改后效果如图 4.14 所示：

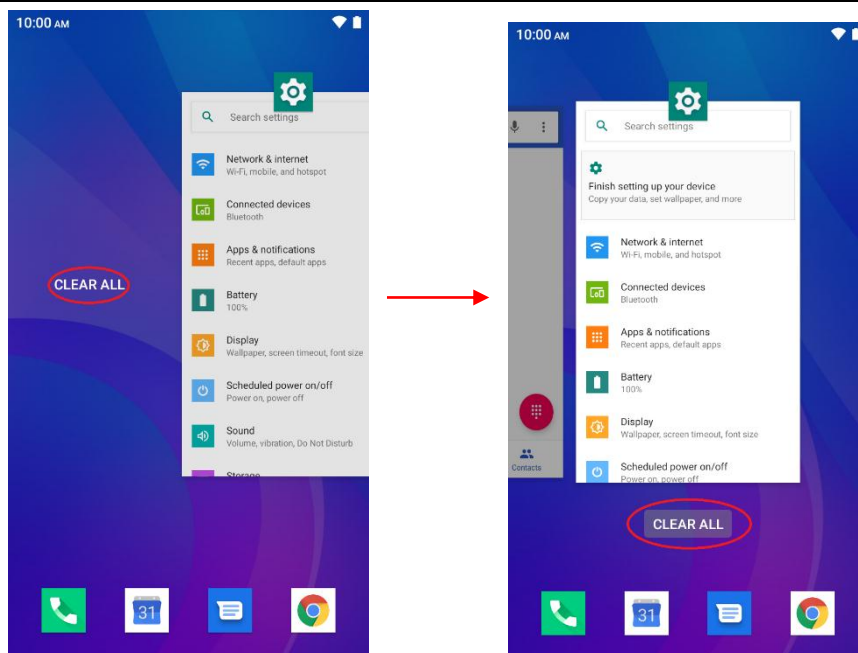


图 4.14 全部清除按钮移到底部

➤ Feature 开关

本 feature 通过如下 prop 属性控制，可根据项目需求使用该开关开启或者关闭本 feature。

ro.launcher.clearallonbottom

➤ Feature 默认状态

本功能所有版本中默认开启。

➤ Feature 配置

本功能的主要代码在如下目录：

Launcher3\quickstep\src\com\sprd\ext\clearall

若需默认关闭功能，请配置 ro.launcher.clearallonbottom 的值为 false 即可

4.10 禁止横屏

➤ Feature 简介

本功能开启后将不再允许待机旋转，“主屏幕设置”中的“允许旋转”开关也会隐藏。

➤ Feature 开关

本 feature 通过如下 prop 属性控制，可根据项目需求使用该开关开启或者关闭本 feature。

ro.launcher.disable.rotation

➤ Feature 默认状态

本 Feature 在 Android 10 No-Go 版本中默认关闭，Android 10-Go 版本中默认开启。



说明：

虽然 Recents 也是在 Launcher 的 Activity 中，但是由于此界面的特殊性，我们允许此界面进行横屏操作。具体代码路径：Launcher3\src\com\android\launcher3\states\RotationHelper.java

Unisoc Confidential For hiar