SIM8500_android10_启动速度优化文档

智能模组

芯讯通无线科技(上海)有限公司
上海市长宁区临虹路289号3号楼芯讯通总部大楼
电话：86-21-31575100
技术支持邮箱：support@simcom.com
官网：www.simcom.com

名称：	SIM8500_android10_启动速度优化文档
版本：	1.00
日期：	2022.3.8
状态：	已发布

版权声明

芯讯通无线科技(上海)有限公司
上海市长宁区临虹路289号3号楼芯讯通总部大楼
电话：86-21-31575100
邮箱：simcom@simcom.com
官网：www.simcom.com

了解更多资料，请点击以下链接：
http://cn.simcom.com/download/list-230-cn.html

技术支持，请点击以下链接：
http://cn.simcom.com/ask/index-cn.html 或发送邮件至 support@simcom.com

关于文档

版本历史

版本	日期	作者	备注
1.00	2022.3.8	杨振东	第一版

适用范围

本文档适用于 SIMCom SIM8500 系列。

目录

1 介绍

1.1 本文目的

基于展锐 8500 android 10 进行启动速度优化，主要从两方面进行，一方面是系统侧，另一方面是 kernel，参考此应用文档，开发者可以很快理解并快速开发相关业务。

1.2 参考文档

1.3 术语和缩写

2 Android 系统优化

由于 Android 系统功能和代码量庞大，因此我们需要细分一些小的功能块来进行相关的优化，如优化 Locales、优化资源加载、优化 pkg(优化掉不必要的 pkg 不仅能够减小 ROM 大小而且能够优化开机时间)、优化 java server 等，下面我们主要优化这些功能来进行系统优化，进而达到提高 android 启动速度。

2.1 基础优化脚本

在系统优化时，我们需要定义一个基础的优化 mk 脚本以作为添加其他优化模块的基础入口。
如在 **device/sprd/sharkle/common** 目录下定义 **simcom_cut.mk** 脚本作为基础优化脚本入口，我们需要将此基础脚本引入编译工程中，如下图红色框标注部分在 **device/sprd/sharkle/sl8541e_1h10_go/sl8541e_1h10_go_base.mk** project 工程脚本中引入 **simcom_cut.mk**，**所要引入 simcom_cut.mk 脚本文件的 project 工程脚本文件不同项目可能有所不同，依据实际项目所定**：

```
diff --git a/sl8541e_1h10_go/sl8541e_1h10_go_base.mk b/sl8541e_1h10_go/sl8541e_1h10_go_base.mk
index a19da82..d365151 100644
--- a/sl8541e_1h10_go/sl8541e_1h10_go_base.mk
+++ b/sl8541e_1h10_go/sl8541e_1h10_go_base.mk
@@ -47,7 +47,8 @@ PRODUCT_PROPERTY_OVERRIDES := \
 # Get some sounds
 $(call inherit-product-if-exists, frameworks/base/data/sounds/AudioPackageGo.mk)
 # Get the TTS language packs
-$(call inherit-product-if-exists, external/svox/pico/lang/all_pico_languages.mk)
+#[SIM8500-579] Modify TTS language packs from all_pico_languages.mk to PicoLangEnUsInSystem.mk by wangwenrui 20220221
+$(call inherit-product-if-exists, external/svox/pico/lang/PicoLangEnUsInSystem.mk)
 $(call inherit-product, $(SRC_TARGET_DIR)/product/generic.mk)
 else
 $(call inherit-product, $(SRC_TARGET_DIR)/product/full_base.mk)
@@ -298,3 +299,7 @@ PRODUCT_COPY_FILES += \
 #for uififo
 PRODUCT_PROPERTY_OVERRIDES += \
     sys.use_fifo_ui=1
+
+# [SIM8500-579] Add system cut mk file by zhendong.yang for SIM8501_CUT at 2022-02-21
+$(call inherit-product, $(PLATCOMM)/simcom_cut.mk)
+# [SIM8500-579] Add system cut mk file by zhendong.yang for SIM8501_CUT at 2022-02-21
yangzhendong@dev173:/mnt/sdb1/yangzhendong-sdb1/simcom/SIM8501/device/sprd/sharkle$
```

2.2 Pkg Module 优化

➢ Pkg Module 编译优化；

Pkg Module 编译优化是添加新的编译宏，在编译的过程中将编译宏中所标注的 module 进行排除编译(即不进行编译包含，不生成编译结果)，这样不仅能够减小 rom 的大小，而且能够优化开机时间

✧ **添加编译宏 PRODUCT_SIMCOM_CUT_PACKAGES**

针对需要被排除的 pkg modules 添加自定义的编译宏,编译宏定义在 **build/make/core/product.mk** 文件中，具体如下图；

```
diff --git a/core/product.mk b/core/product.mk
index 9ff0678..cd3eba6 100644
--- a/core/product.mk
+++ b/core/product.mk
@@ -144,6 +144,13 @@ _product_var_list := \
     PRODUCT_MINIMIZE_JAVA_DEBUG_INFO \
     PRODUCT_INTEGER_OVERFLOW_EXCLUDE_PATHS \

+# [SIM8500-579]Add simcom cut var config by zhendong.yang for SIM8501_CUT at 2022-02-21
+_product_var_list += \
+    PRODUCT_SIMCOM_CUT_KEEPLOCALES \
+    PRODUCT_SIMCOM_CUT_PACKAGES \
+
+# [SIM8500-579]Add simcom cut var config by zhendong.yang for SIM8501_CUT at 2022-02-21
+
```

✧ **修改 module 编译脚本**

添加了编译宏后，我们需要在编译应用的地方实现编译排除逻辑(编译不仅针对正常的编译条件,也包含 debug、eng、tests 等编译选项)，具体的逻辑就是在当前的 modules 中排除 PRODUCT_SIMCOM_CUT_PACKAGES 中所包含的 pkg modules, 编译脚本修改在 **build/make/core/main.mk**,具体如下图；

```
diff --git a/core/main.mk b/core/main.mk
index 44ad271..d703c11 100644
--- a/core/main.mk
+++ b/core/main.mk
```

```
 ifdef FULL_BUILD
+
+  # [SIM8500-579]Add simcom cut filter out by zhendong.yang for SIM8501_CUT at 2022-02-21
+  modules_simcom_cut := $(PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_SIMCOM_CUT_PACKAGES)
+  #$(warning "zhendong.yang : cutted modules " = $(modules_simcom_cut))
+  #$(warning "zhendong.yang : origin modules " = $(PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_PACKAGES))
+  ifneq (,$(modules_simcom_cut))
+      #Common
+      modules_simcom_keepcommon := $(filter-out $(foreach cut_debug_item, $(modules_simcom_cut), \
+          $(cut_debug_item)), $(PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_PACKAGES))
+      #Debug
+      modules_simcom_keepdebug := $(filter-out $(foreach cut_debug_item, $(modules_simcom_cut), \
+          $(cut_debug_item)), $(PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_PACKAGES_DEBUG))
+      #ENG
+      modules_simcom_keepeng := $(filter-out $(foreach cut_eng_item, $(modules_simcom_cut), \
+          $(cut_eng_item)), $(PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_PACKAGES_ENG))
+      #TEST
+      modules_simcom_keeptests := $(filter-out $(foreach cut_test_item, $(modules_simcom_cut), \
+          $(cut_test_item)), $(PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_PACKAGES_TESTS))
+
+      $(eval PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_PACKAGES := $(modules_simcom_keepcommon))
+      $(eval PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_PACKAGES_DEBUG := $(modules_simcom_keepdebug))
+      $(eval PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_PACKAGES_ENG := $(modules_simcom_keepeng))
+      $(eval PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_PACKAGES_TESTS := $(modules_simcom_keeptests))
+
+      modules_simcom_keepcommon :=
+      modules_simcom_keepdebug :=
+      modules_simcom_keepeng :=
+      modules_simcom_keeptests :=
+
+      #$(warning "zhendong.yang : PRODUCT_PACKAGES  1111**********" = $(PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_PACKAGES))
+      #$(warning "zhendong.yang : PRODUCT_PACKAGES_DEBUG  2222**********" = $(PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_PACKAGES_DEBU
+      #$(warning "zhendong.yang : PRODUCT_PACKAGES_ENG  3333**********" = $(PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_PACKAGES_ENG))
+      #$(warning "zhendong.yang : PRODUCT_PACKAGES_TESTS  4444**********" = $(PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_PACKAGES_TEST
+  endif
+  # [SIM8500-579]Add simcom cut filter out by zhendong.yang for SIM8501_CUT at 2022-02-21
+
```

```
   # The base list of modules to build for this product is specified
   # by the appropriate product definition file, which was included
   # by product_config.mk.
@@ -812,6 +849,15 @@ ifdef FULL_BUILD
   product_MODULES := $(filter-out $(foreach m, $(product_MODULES), \
       $(EXECUTABLES.$(m).OVERRIDES)), $(product_MODULES))

+  # [SIM8500-579]Add simcom cut filter out by zhendong.yang for SIM8501_CUT at 2022-02-21
+  ifneq (,$(modules_simcom_cut))
+      product_MODULES := $(filter-out $(foreach cut_item, $(modules_simcom_cut), \
+          $(cut_item)), $(product_MODULES))
+      #$(warning "zhendong.yang : ****product_MODULES " = $(product_MODULES))
+  endif
+  modules_simcom_cut :=
+  # [SIM8500-579]Add simcom cut filter out by zhendong.yang for SIM8501_CUT at 2022-02-21
+
   # Resolve the :32 :64 module name
```

➢ 添加需要优化的 Pkg；

在添加新的编译宏并修改编译脚本后，需要应用此编译宏以达到我们所需要的排除编译的效果，就需要定义 PRODUCT_SIMCOM_CUT_PACKAGES 宏的内容并引入。

✧ 定义 pkg 优化脚本

定义一个新的 mk 脚本文件来添加 pkg modules,如在 **device/sprd/sharkle/common** 下面创建 **simcom_cut_packages.mk** 脚本文件，定义格式如下图；

```
1 ############    Cut Package    ################
2 PRODUCT_SIMCOM_CUT_PACKAGES += \
3     QuickCamera \
```

✧ 添加需要优化的 pkg

按照上面的定义格式添加需要被排除的 pkg modules,具体参考如下图；

```
############    Cut Package    #################
PRODUCT_SIMCOM_CUT_PACKAGES += \
    QuickCamera \
    DreamCamera2 \
    Camera2 \
    DreamFMRadio \
    EasterEgg \
    MmsService \
    CtsShimPrivPrebuilt \
    SprdCalendarProvider \
    WallpaperCropper \
    CallFireWall \
    SupportCPVersionDisplay \
    WakeupScreenPlugin \
    DocumentsUI \
    CompanionDeviceManager \
    RingBackTonePlugin \
    SprdBrowser \
    DreamSoundRecorder \
    ContactsDefaultContactAddon \
    DefaultContainerService \
    PacProcessor \
    TouchPalGlobal \
    LovelyFontContainerService \
    SharedStorageBackup \
    InputDevices \
    MusicFX \
    PrintRecommendationService \
    NewGallery2 \
    LovelyFonts \
    PowerSaveModeLauncher \
    BrowserXposed \
    CtsShimPrebuilt \
    SprdBrowserCustomAddon \
    NewMusic \
    WallpaperBackup \
    FdnService \
    CaptivePortalLogin \
    LimitOwnerInfoLength \
    BLEDemo \
```

```
    Bluelet \
    FastScrollBarSupportAddon \
    ShakePhoneToStartRecording \
    FlipToMute \
    BuiltInPrintService \
    DocumentsUIXposed \
    MaxRingingVolumeAndVibrate \
    FadeDownRingtoneToVibrate \
    CalendarProvider \
    SoundRecorder \
    Gallery2 \
    Browser2 \
    ContactsBlackListAddon \
    BookmarkProvider \
    Music \
    EmergencyInfo \
    PrintSpooler \
    SmilPlayer \
    CellBroadcastReceiver \
    OneTimeInitializer \
```

✧ **引用 pkg 优化脚本**

在创建好 **simcom_cut_packages.mk** 脚本文件之后,我们只需要将此脚本文件引入项目编译的基础优化编译脚本文件即可如引入 **simcom_cut.mk**，此后此 pkg 的优化脚本就将会被加入项目编译中并实现编译优化，如下图；

```
# Add cut config for simcom project

SIMCOM_COMMON_DIR := device/sprd/sharkle/common

# Cut Package
$(call inherit-product, $(SIMCOM_COMMON_DIR)/simcom_cut_packages.mk)

# Cut Locales
$(call inherit-product, $(SIMCOM_COMMON_DIR)/simcom_cut_locales.mk)

# Cut java server by prop
$(call inherit-product, $(SIMCOM_COMMON_DIR)/simcom_cut_server.mk)
```

2.3 Locales 优化

➢ Locales 编译优化

Locales 编译优化是添加新的编译宏，定义系统需要保留的编译语言资源，来替换系统默认的使用的编译语言资源，这样来达到减小 rom 和优化开机时间的作用

✧ 添加编译宏 PRODUCT_SIMCOM_CUT_KEEPLOCALES

针对需要保留的语言资源添加自定义的编译宏,编译宏定义在 **build/make/core/product.mk** 文件中，具体如下图；

```
diff --git a/core/product.mk b/core/product.mk
index 9ff0678..cd3eba6 100644
--- a/core/product.mk
+++ b/core/product.mk
@@ -144,6 +144,13 @@ _product_var_list := \
     PRODUCT_MINIMIZE_JAVA_DEBUG_INFO \
     PRODUCT_INTEGER_OVERFLOW_EXCLUDE_PATHS \

+# [SIM8500-579]Add simcom cut var config by zhendong.yang for SIM8501_CUT at 2022-02-21
+_product_var_list += \
+    PRODUCT_SIMCOM_CUT_KEEPLOCALES \
+    PRODUCT_SIMCOM_CUT_PACKAGES \
+
+# [SIM8500-579]Add simcom cut var config by zhendong.yang for SIM8501_CUT at 2022-02-21
+
```

✧ 修改 LOCALES 编译脚本

添加了编译宏定义后，我们需要在编译应用的地方实现编译替换，具体的逻辑就是用当前定义的 **PRODUCT_SIMCOM_CUT_KEEPLOCALES** 来替换系统编译最终的 LOCALES,编译脚本修改在 **build/make/core/product_config.mk**,具体如下图；

```
diff --git a/core/product_config.mk b/core/product_config.mk
index 1c3eea1..a2c5b22 100644
--- a/core/product_config.mk
+++ b/core/product_config.mk
@@ -283,6 +283,15 @@ ifneq (,$(extra_locales))
   extra_locales :=
 endif

+# [SIM8500-579] cut locales by zhendong.yang for SIM8501_CUT at 2022-02-21
+simcom_cut_keeplocales := $(strip $(PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_SIMCOM_CUT_KEEPLOCALES))
+ifneq (,$(simcom_cut_keeplocales))
+  PRODUCT_LOCALES := $(simcom_cut_keeplocales)
+  simcom_cut_keeplocales :=
+endif
+$(warning "zhendong.yang : PRODUCT_LOCALES" = $(PRODUCT_LOCALES))
+# [SIM8500-579] cut locales by zhendong.yang for SIM8501_CUT at 2022-02-21
+
 # Add PRODUCT_LOCALES to PRODUCT_AAPT_CONFIG
 PRODUCT_AAPT_CONFIG := $(strip $(PRODUCT_LOCALES) $(PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_AAPT_CONFIG))
```

➢ 添加 Keep Locales

在添加新的编译宏并修改编译脚本后，需要应用此编译宏以达到我们所需要的替换 Locales 效果，就需要定义 PRODUCT_SIMCOM_CUT_KEEPLOCALES 宏的内容并引入。

✧ 定义 Locales 优化脚本并添加需要保留的 locales

定义一个新的 mk 脚本文件来定义需要保留的 locales,如在 **device/sprd/sharkle/common** 下面创建 **simcom_cut_locales.mk** 脚本文件，定义格式如下图；

```
#############    Cut Locales    ##################
PRODUCT_SIMCOM_CUT_KEEPLOCALES := en_US zh_CN
```

如上图，我们定义了系统需要保留的语言为 en_US zh_CN

✧ 引用 Keep Locales 脚本

在创建好 **simcom_cut_locales.mk** 脚本文件之后,我们只需要将此脚本文件引入项目编译的基础优化编译脚本文件即可如引入 **simcom_cut.mk**，此后 Keep Locales 优化脚本就将会被加入项目编译中并实现编译优化，如下图

```
# Add cut config for simcom project

SIMCOM_COMMON_DIR := device/sprd/sharkle/common

# Cut Package
$(call inherit-product, $(SIMCOM_COMMON_DIR)/simcom_cut_packages.mk)

# Cut Locales
$(call inherit-product, $(SIMCOM_COMMON_DIR)/simcom_cut_locales.mk)

# Cut java server by prop
$(call inherit-product, $(SIMCOM_COMMON_DIR)/simcom_cut_server.mk)
```

✧ **优化系统设置中语言选择列表**

在引用完 Keep Locales 脚本后，系统编译时如图只会保留 en_US zh_CN 两种语言，但是系统设置中语言选项中仍旧会显示除了 en_US zh_CN 外很多其他语言选择，尽快选择后可能不生效，但这严重影响了用户的体验，因此我们需要去掉其中多余的语言选择，我们需要在 device/sprd/sharkle/common 目录下创建 **simcom_cut/overlay/frameworks/base/core/res/res/values/locale_config.xml**,内容具体如下:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Copyright (C) 2015 The Android Open Source Project

     Licensed under the Apache License, Version 2.0 (the "License");
     you may not use this file except in compliance with the License.
     You may obtain a copy of the License at

          http://www.apache.org/licenses/LICENSE-2.0

     Unless required by applicable law or agreed to in writing, software
     distributed under the License is distributed on an "AS IS" BASIS,
     WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
     See the License for the specific language governing permissions and
     limitations under the License.
-->

<resources>

    <string-array translatable="false" name="supported_locales">
        <item>en-US</item> <!-- English (United States) -->
        <item>zh-Hans-CN</item> <!-- Chinese (Simplified Han,China) -->
    </string-array>

</resources>
```

在创建完 overlay 文件 locale_config.xml 后，我**们需要在编译**过程中引入覆盖，就可以替换掉 settings 中的语言列表选项了，如我们修改了 **device/sprd/sharkle/sl8541e_1h10_go/sl8541e_1h10_gofu.mk 文件引入了 overlay 文件目录(备注：不同项目可能引入 overlay 目录的工程 mk 文件也不同)**，具体如下图：

```
diff --git a/sl8541e_1h10_go/sl8541e_1h10_gofu.mk b/sl8541e_1h10_go/sl8541e_1h10_gofu.mk
index ddde60c..f029582 100644
--- a/sl8541e_1h10_go/sl8541e_1h10_gofu.mk
+++ b/sl8541e_1h10_go/sl8541e_1h10_gofu.mk
@@ -48,6 +48,12 @@ TARGET_BOARD_SPEC_CONFIG := device/sprd/sharkle/sl8541e_1h10_go/$(PRODUCT_NAME).

 DEVICE_PACKAGE_OVERLAYS := $(BOARDDIR)/overlay $(PLATDIR)/overlay $(PLATCOMM)/overlay

+# [SIM8500-579] cut locales by zhendong.yang for SIM8501_CUT at 2022-02-21
+ifdef PRODUCT_SIMCOM_CUT_KEEPLOCALES
+DEVICE_PACKAGE_OVERLAYS += $(PLATCOMM)/simcom_cut/overlay
+endif
+# [SIM8500-579] cut locales by zhendong.yang for SIM8501_CUT at 2022-02-21
+
 #Runtime Overlay Packages
```

✧ **特殊 Locales 替换声明**

在引用 **simcom_cut_locales.mk 脚本编译后，一般情况下系统就会按照我们正常所预测的那样保留我们脚本所定义的 locales(如高通平台)，但是展锐平台却不同，我们的替换不生效，那是因为展锐在它的编译脚本中又引入了 PRODUCT_REVISION 宏定义，**如果此宏定义的内容包含 **multi-lang**

选项，那么说明展锐会引用 vendor/sprd/feature-configs/**multi-lang** 目录下的多语言选项配置，如图：

```
# The Spreadtrum Communications Inc. 2015

# This is the top of this features, this feature may
# contains much sub-features, and they may divided
# and take effects by properties / addons / overlays or
# prebuilts

# You can gather them by build vars as follows:

# properties -> FEATURES.PRODUCT_PROPERTY_OVERRIDES +=
# addons/packages/prebuilts -> FEATURES.PRODUCT_PACKAGES +=
# overlay -> FEATURES.PRODUCT_PACKAGE_OVERLAYS +=
#

# Now, let's get it on as follows

FEATURES.PRODUCT_LOCALES := en_US en_AU en_CA en_GB en_IE en_IN en_NZ en_SG en_ZA zh_CN zh_HK zh_TW ar_EG ar_IL fa_IR ru_RU fr_FR sw_TZ th_T
H tr_TR es_ES es_US hi_IN in_ID vi_VN ha_GH my_MM nl_NL uk_UA it_IT pt_PT pt_BR et_EE as_ET ro_RO ms_MY bn_BD tl_PH ceb_PH te_IN ta_IN ur_PK
 am_ET pl_PL da_DK de_DE gu_IN cs_CZ pa_IN sl_SI el_GR hr_HR lv_LV ml_IN mr_IN kn_IN sv_SE nb_NO hu_HU bg_BG sq_AL lt_LT sr_RS sk_SK fi_FI c
a_ES mk_MK eu_ES gl_ES km_KH lo_LA iw_IL ne_NP si_LK bs_BA ka_GE bo_CN ug_CN or_IN kok_IN mai_IN doi_IN mni_IN ks_IN sa_IN sd_IN brx_IN sat_
IN so_SO om_ET ti_ET

FEATURES.PRODUCT_PACKAGE_OVERLAYS += \
    $(PRODUCT_REVISION_COMMON_CONFIG_PATH)/multi-lang/overlay
```

上图展锐定义了 FEATURES.PRODUCT_LOCALES 宏以及对应的 overlay 目录，这个宏在执行 vendor/sprd/build/core/apply_produce_revision.mk 脚本时替换我们修改过的 LOCALES 宏。如图：

```
# To control the oem.prop
ifndef BOARD_OEMIMAGE_PARTITION_SIZE
# The properties has been set into the ADDITIONAL_BUILD_PROPERTIES, so append them.
ADDITIONAL_BUILD_PROPERTIES += $(FEATURES.PRODUCT_PROPERTY_OVERRIDES)
#$(info ADDITIONAL_BUILD_PROPERTIES := $(ADDITIONAL_BUILD_PROPERTIES))
endif

# The Locales will be overwrite completely
ifdef FEATURES.PRODUCT_LOCALES
PRODUCT_AAPT_CONFIG := $(strip $(filter-out $(PRODUCTS.$(INTERNAL_PRODUCT).PRODUCT_LOCALES),$(PRODUCT_AAPT_CONFIG)))
PRODUCT_LOCALES := $(strip $(FEATURES.PRODUCT_LOCALES))
PRODUCT_AAPT_CONFIG := $(strip $(PRODUCT_LOCALES) $(PRODUCT_AAPT_CONFIG))
#$(info PRODUCT_LOCALES := $(PRODUCT_LOCALES))
endif

#$(info )
```

因此我们修改的 LOCALES 可能不会生效，如需要避免这个问题，我们需要在展锐定义的 **PRODUCT_REVISION** 中删掉 **multi-lang** 选项，修改如下图:

```
diff --git a/sl8541e_1h10_go/sl8541e_1h10_gofu_osea.mk b/sl8541e_1h10_go/sl8541e_1h10_gofu_osea.mk
index 1488198..a07ed9e 100644
--- a/sl8541e_1h10_go/sl8541e_1h10_gofu_osea.mk
+++ b/sl8541e_1h10_go/sl8541e_1h10_gofu_osea.mk
@@ -17,7 +17,13 @@
 include device/sprd/sharkle/sl8541e_1h10_go/sl8541e_1h10_gofu.mk
 $(call inherit-product-if-exists, vendor/sprd/modules/faceunlock/faceunlock_device.mk)

+# [SIM8500-579] cut locales by zhendong.yang for SIM8501_CUT at 2022-02-21
+ifdef PRODUCT_SIMCOM_CUT_KEEPLOCALES
+PRODUCT_REVISION := oversea
+else
 PRODUCT_REVISION := oversea multi-lang
+endif
+# [SIM8500-579] cut locales by zhendong.yang for SIM8501_CUT at 2022-02-21
 include $(APPLY_PRODUCT_REVISION)
```

2.4 JAVA SERVER 优化

去掉不需要的 server 也可以达到优化开机时间和减小内存占用效果。

✧ **定义 Server 优化脚本**

定义一个新的 mk 脚本文件来优化 Server 是否启动,如在 **device/sprd/sharkle/common** 下面创建 **simcom_cut_server.mk** 脚本文件，定义格式如下图；

```
############    Cut Java Server    ################
# config prop for cut java server
PRODUCT_PROPERTY_OVERRIDES += \
    config.enable.systemcut=true \
    config.disable_cameraservice=true \
    config.disable_searchmanager=true \
    config.disable_vrmanager=true \
    config.disable_consumerir=true \
    pm.dexopt.disable_bg_dexopt=true \
```

如上图，我们定义了系统优化的 server 属性，config.enable.systemcut 为系统优化公共属性，其他属性如 config.disable_vrmanager 等为对应系统服务所对应的属性，为 android 原始代码定义

✧ **引用 server 优化脚本**

在创建好 **simcom_cut_server.mk** 脚本文件之后,我们只需要将此脚本文件引入项目编译的基础优化编译脚本文件即可如引入 **simcom_cut.mk**，此后优化脚本就将会被加入项目编译中并实现编译优化，如下图

```
# Add cut config for simcom project

SIMCOM_COMMON_DIR := device/sprd/sharkle/common

# Cut Package
$(call inherit-product, $(SIMCOM_COMMON_DIR)/simcom_cut_packages.mk)

# Cut Locales
$(call inherit-product, $(SIMCOM_COMMON_DIR)/simcom_cut_locales.mk)

# Cut java server by prop
$(call inherit-product, $(SIMCOM_COMMON_DIR)/simcom_cut_server.mk)
```

✧ **修改 server 服务启动并修改相关调用**

(1) 引用 android 系统属性优化 server 启动

Android 系统中原本存在一些控制服务是否启动的属性如 config.disable_vrmanager,如果我们配置属性 config.disable_vrmanager 为 true 则表示禁止启动 vr 服务，如 SystemServer.java 中的代码

```
        boolean disableTextServices = SystemProperties.getBoolean("config.disable_textservices",
                false);
        boolean disableConsumerIr = SystemProperties.getBoolean("config.disable_consumerir", false);
        boolean disableVrManager = SystemProperties.getBoolean("config.disable_vrmanager", false);
        boolean disableCameraService = SystemProperties.getBoolean("config.disable_cameraservice",
                false);
        boolean enableLeftyService = SystemProperties.getBoolean("config.enable_lefty", false);
```

```
        if (!disableVrManager) {
            traceBeginAndSlog("StartVrManagerService");
            mSystemServiceManager.startService(VrManagerService.class);
            traceEnd();
        }
```

上图代码显示如果配置 config.disable_vrmanager 为 true,则 VrManagerService 服务将不会被启动，同时我们也应该修改和 VrManagerService 相关类的调用逻辑，如注册 VrManager 的类 SystemServiceRegistry.java 以及 VrManager 相关的调用，具体如下图:

```
diff --git a/core/java/android/app/SystemServiceRegistry.java b/core/java/android/app/SystemServiceRegistry.java
index 6ccccf6..d3c49e4 100755
--- a/core/java/android/app/SystemServiceRegistry.java
+++ b/core/java/android/app/SystemServiceRegistry.java
```

```
@@ -953,6 +959,9 @@ final class SystemServiceRegistry {
         registerService(Context.VR_SERVICE, VrManager.class, new CachedServiceFetcher<VrManager>() {
             @Override
             public VrManager createService(ContextImpl ctx) throws ServiceNotFoundException {
+                if (android.os.SystemProperties.getBoolean("config.disable_vrmanager", false)) {
+                    return new VrManager(null);
+                }
                 IBinder b = ServiceManager.getServiceOrThrow(Context.VR_SERVICE);
                 return new VrManager(IVrManager.Stub.asInterface(b));
             }
```

```
diff --git a/core/java/android/app/VrManager.java b/core/java/android/app/VrManager.java
index 5c6ffa3..d65a96f 100644
--- a/core/java/android/app/VrManager.java
+++ b/core/java/android/app/VrManager.java
@@ -57,6 +57,10 @@ public class VrManager {
         mService = service;
     }

+    private boolean isValidVrService() {
+        return mService != null && !android.os.SystemProperties.getBoolean("config.disable_vrmanager", false);
+    }
+
     /**
      * Registers a callback to be notified of changes to the VR Mode state.
      *
@@ -68,7 +72,9 @@ public class VrManager {
             android.Manifest.permission.ACCESS_VR_STATE
     })
     public void registerVrStateCallback(VrStateCallback callback, @NonNull Handler handler) {
-        if (callback == null || mCallbackMap.containsKey(callback)) {
+        if (callback == null
+            || mCallbackMap.containsKey(callback)
+            || !isValidVrService()) {
             return;
         }
```

```
@@ -98,7 +104,7 @@ public class VrManager {
     })
     public void unregisterVrStateCallback(VrStateCallback callback) {
         CallbackEntry entry = mCallbackMap.remove(callback);
-        if (entry != null) {
+        if (entry != null && isValidVrService()) {
             try {
                 mService.unregisterListener(entry.mStateCallback);
             } catch (RemoteException ignore) {
@@ -122,6 +128,7 @@ public class VrManager {
             android.Manifest.permission.ACCESS_VR_STATE
     })
     public boolean getVrModeEnabled() {
+        if (!isValidVrService()) return false;
         try {
             return mService.getVrModeState();
         } catch (RemoteException e) {
@@ -139,6 +146,7 @@ public class VrManager {
             android.Manifest.permission.ACCESS_VR_STATE
     })
     public boolean getPersistentVrModeEnabled() {
+        if (!isValidVrService()) return false;
         try {
             return mService.getPersistentVrModeEnabled();
         } catch (RemoteException e) {
@@ -157,6 +165,7 @@ public class VrManager {
      */
     @RequiresPermission(android.Manifest.permission.RESTRICTED_VR_ACCESS)
     public void setPersistentVrModeEnabled(boolean enabled) {
+        if (!isValidVrService()) return;
         try {
             mService.setPersistentVrModeEnabled(enabled);
         } catch (RemoteException e) {
```

```
@@ -176,6 +185,7 @@ public class VrManager {
     @RequiresPermission(android.Manifest.permission.RESTRICTED_VR_ACCESS)
     public void setVr2dDisplayProperties(
             Vr2dDisplayProperties vr2dDisplayProp) {
+        if (!isValidVrService()) return;
         try {
             mService.setVr2dDisplayProperties(vr2dDisplayProp);
         } catch (RemoteException e) {
@@ -191,6 +201,7 @@ public class VrManager {
      */
     @RequiresPermission(android.Manifest.permission.RESTRICTED_VR_ACCESS)
     public void setAndBindVrCompositor(ComponentName componentName) {
+        if (!isValidVrService()) return;
         try {
             mService.setAndBindCompositor(
                     (componentName == null) ? null : componentName.flattenToString());
```

(2) 引用 config.enable.systemcut 属性优化 server 启动

config.enable.systemcut 属性为我们自定义的优化 server 启动属性，我们配置属性 config.enable.systemcut 为 true，如果某个 server 在启动服务时添加判断则可控制 server 的启动，如 SystemServer.java 中的代码

```
diff --git a/services/java/com/android/server/SystemServer.java b/services/java/com/android/server/SystemServer.java
index 253e068..73a661f 100644
--- a/services/java/com/android/server/SystemServer.java
+++ b/services/java/com/android/server/SystemServer.java
@@ -721,6 +721,10 @@ public final class SystemServer {
         //Bug 692795 PowerGuruService powerguru --> END
         PowerController powerctl = null;

+        // [SIM8500-579] Cut java server by zhendong.yang for SIM8501_CUT at 2022-02-24
+        boolean enableSystemCut = SystemProperties.getBoolean("config.enable.systemcut", false);
+        // [SIM8500-579] Cut java server by zhendong.yang for SIM8501_CUT at 2022-02-24
+
         boolean disableStorage = SystemProperties.getBoolean("config.disable_storage", false);
         boolean disableBluetooth = SystemProperties.getBoolean("config.disable_bluetooth", false);
         boolean disableLocation = SystemProperties.getBoolean("config.disable_location", false);
```

```
             if (NightDisplayController.isAvailable(context)) {
                 traceBeginAndSlog("StartNightDisplay");
@@ -1365,11 +1385,13 @@ public final class SystemServer {
                     traceEnd();
                 }

-                if (mPackageManager.hasSystemFeature(PackageManager.FEATURE_APP_WIDGETS)
-                        || context.getResources().getBoolean(R.bool.config_enableAppWidgetService)) {
-                    traceBeginAndSlog("StartAppWidgerService");
-                    mSystemServiceManager.startService(APPWIDGET_SERVICE_CLASS);
-                    traceEnd();
+                if (!enableSystemCut) {
+                    if (mPackageManager.hasSystemFeature(PackageManager.FEATURE_APP_WIDGETS)
+                            || context.getResources().getBoolean(R.bool.config_enableAppWidgetService)) {
+                        traceBeginAndSlog("StartAppWidgerService");
+                        mSystemServiceManager.startService(APPWIDGET_SERVICE_CLASS);
+                        traceEnd();
+                    }
                 }

                 // We need to always start this service, regardless of whether the
@@ -1435,7 +1457,7 @@ public final class SystemServer {
             }
             traceEnd();
```

上图代码显示如果配置 config.enable.systemcut 为 true,则 AppWidgetService 服务将不会被启动，同时我们也应该修改和 AppWidgetService 相关类的调用逻辑，如注册 AppWidgetManager 的类 SystemServiceRegistry.java 以及 AppWidgetManager 相关的调用，具体如下图:

```
diff --git a/core/java/android/app/SystemServiceRegistry.java b/core/java/android/app/SystemServiceRegistry.java
index 6ccccf6..d3c49e4 100755
--- a/core/java/android/app/SystemServiceRegistry.java
+++ b/core/java/android/app/SystemServiceRegistry.java
```

```
             }
@@ -761,7 +764,8 @@ final class SystemServiceRegistry {
             @Override
             public FingerprintManager createService(ContextImpl ctx) throws ServiceNotFoundException {
                 final IBinder binder;
-                if (ctx.getApplicationInfo().targetSdkVersion >= Build.VERSION_CODES.O) {
+                if (ctx.getApplicationInfo().targetSdkVersion >= Build.VERSION_CODES.O
+                        && !android.os.SystemProperties.getBoolean("config.enable.systemcut", false)) {
                     binder = ServiceManager.getServiceOrThrow(Context.FINGERPRINT_SERVICE);
                 } else {
                     binder = ServiceManager.getService(Context.FINGERPRINT_SERVICE);
@@ -859,8 +863,10 @@ final class SystemServiceRegistry {
                 new CachedServiceFetcher<AppWidgetManager>() {
             @Override
             public AppWidgetManager createService(ContextImpl ctx) throws ServiceNotFoundException {
-                IBinder b = ServiceManager.getServiceOrThrow(Context.APPWIDGET_SERVICE);
-                return new AppWidgetManager(ctx, IAppWidgetService.Stub.asInterface(b));
+                IBinder b = android.os.SystemProperties.getBoolean("config.enable.systemcut", false)
+                        ? null
+                        : ServiceManager.getServiceOrThrow(Context.APPWIDGET_SERVICE);
+                return new AppWidgetManager(ctx, b != null ? IAppWidgetService.Stub.asInterface(b) : null);
             }});

         registerService(Context.MIDI_SERVICE, MidiManager.class,
```

```
diff --git a/core/java/android/appwidget/AppWidgetManager.java b/core/java/android/appwidget/AppWidgetManager.java
old mode 100644
new mode 100755
index 969b19e..338b861
--- a/core/java/android/appwidget/AppWidgetManager.java
+++ b/core/java/android/appwidget/AppWidgetManager.java
@@ -1144,7 +1144,7 @@ public class AppWidgetManager {
      */
     public boolean isRequestPinAppWidgetSupported() {
         try {
-            return mService.isRequestPinAppWidgetSupported();
+            return mService == null ? false : mService.isRequestPinAppWidgetSupported();
         } catch (RemoteException e) {
             throw e.rethrowFromSystemServer();
         }
@@ -1193,6 +1193,7 @@ public class AppWidgetManager {
     public boolean requestPinAppWidget(@NonNull ComponentName provider,
             @Nullable Bundle extras, @Nullable PendingIntent successCallback) {
         try {
+            if (mService == null) return false;
             return mService.requestPinAppWidget(mPackageName, provider, extras,
                     successCallback == null ? null : successCallback.getIntentSender());
         } catch (RemoteException e) {
```

(3) Server 启动优化说明

如果 Server 被禁止启动，那么相关的调用可能都需要被修改，如对应的 manager 以及调用 manager 和 server 的地方。

初步可优化的 server 如下图

```
【修改描述】：1.裁剪部分java服务(search/vr/ir/UiModeManagerService
              /AppWidgerService/BroadcastRadioService/Print
              /Fingerprint/MmsServiceBroker/VibratorService
              /DreamManagerService/CertBlacklister/TwilightService
              /DockObserver/ThermalObserver/PinnerService

/KeyChainSystemService/EntropyMixer/PruneInstantAppsJobService
              /CountryDetectorService/dexopt)
```

可能不完善，后续可以追加 server 优化。

2.5 系统字体裁剪

✧ **移除 Google 相关字体**

修改“build/target/product/handheld_system.mk”文件，将其中 google-fonts 相关 fonts 删除。

```
diff --git a/target/product/handheld_system.mk b/target/product/handheld_system.mk
old mode 100644
new mode 100755
index 6463a54..9423591
--- a/target/product/handheld_system.mk
+++ b/target/product/handheld_system.mk
@@ -20,11 +20,11 @@
 # does, use base_vendor.mk).
 $(call inherit-product, $(SRC_TARGET_DIR)/product/media_system.mk)
 $(call inherit-product-if-exists, frameworks/base/data/fonts/fonts.mk)
-$(call inherit-product-if-exists, external/google-fonts/dancing-script/fonts.mk)
-$(call inherit-product-if-exists, external/google-fonts/carrois-gothic-sc/fonts.mk)
-$(call inherit-product-if-exists, external/google-fonts/coming-soon/fonts.mk)
-$(call inherit-product-if-exists, external/google-fonts/cutive-mono/fonts.mk)
-$(call inherit-product-if-exists, external/google-fonts/source-sans-pro/fonts.mk)
+#$(call inherit-product-if-exists, external/google-fonts/dancing-script/fonts.mk)
+#$(call inherit-product-if-exists, external/google-fonts/carrois-gothic-sc/fonts.mk)
+#$(call inherit-product-if-exists, external/google-fonts/coming-soon/fonts.mk)
+#$(call inherit-product-if-exists, external/google-fonts/cutive-mono/fonts.mk)
+#$(call inherit-product-if-exists, external/google-fonts/source-sans-pro/fonts.mk)
 $(call inherit-product-if-exists, external/noto-fonts/fonts.mk)
 $(call inherit-product-if-exists, external/roboto-fonts/fonts.mk)
 $(call inherit-product-if-exists, external/hyphenation-patterns/patterns.mk)
```

✧ 仅保留中文及英文系统字体

在“external/noto-fonts/fonts.mk”文件中仅保留“NotoSansCJK-Regular.tcc”字体，其余均删除。

```
diff --git a/fonts.mk b/fonts.mk
old mode 100644
new mode 100755
index fee43e2..2c6ee3d
--- a/fonts.mk
+++ b/fonts.mk
@@ -17,222 +17,224 @@
 # get installed too.

 PRODUCT_PACKAGES := \
-    NotoColorEmoji.ttf \
-    NotoNaskhArabic-Bold.ttf \
-    NotoNaskhArabic-Regular.ttf \
-    NotoNaskhArabicUI-Bold.ttf \
-    NotoNaskhArabicUI-Regular.ttf \
-    NotoSansAdlam-Regular.ttf \
-    NotoSansAhom-Regular.otf \
-    NotoSansAnatolianHieroglyphs-Regular.otf \
-    NotoSansArmenian-Bold.otf \
-    NotoSansArmenian-Medium.otf \
-    NotoSansArmenian-Regular.otf \
-    NotoSansAvestan-Regular.ttf \
-    NotoSansBalinese-Regular.ttf \
-    NotoSansBamum-Regular.ttf \
-    NotoSansBassaVah-Regular.otf \
-    NotoSansBatak-Regular.ttf \
-    NotoSansBengali-Bold.otf \
-    NotoSansBengali-Medium.otf \
-    NotoSansBengali-Regular.otf \
-    NotoSansBengaliUI-Bold.otf \
-    NotoSansBengaliUI-Medium.otf \
-    NotoSansBengaliUI-Regular.otf \
-    NotoSansBhaiksuki-Regular.otf \
-    NotoSansBrahmi-Regular.ttf \
-    NotoSansBuginese-Regular.ttf \
-    NotoSansBuhid-Regular.ttf \
-    NotoSansCanadianAboriginal-Regular.ttf \
-    NotoSansCarian-Regular.ttf \
-    NotoSansChakma-Regular.otf \
-    NotoSansCham-Bold.ttf \
-    NotoSansCham-Regular.ttf \
-    NotoSansCherokee-Regular.ttf \
     NotoSansCJK-Regular.ttc \
-    NotoSansCoptic-Regular.ttf \
-    NotoSansCuneiform-Regular.ttf \
```

在“frameworks/base/data/fonts/fonts.xml”文件中仅保留 NotoSansCJK-Regula 及 Roboto 相关字体的配置，其余均删除。修改后内容如下：

```
<familyset version="22">
    <!-- first font is default -->
    <family name="sans-serif">
        <font weight="100" style="normal">Roboto-Thin.ttf</font>
        <font weight="100" style="italic">Roboto-ThinItalic.ttf</font>
        <font weight="300" style="normal">Roboto-Light.ttf</font>
        <font weight="300" style="italic">Roboto-LightItalic.ttf</font>
        <font weight="400" style="normal">Roboto-Regular.ttf</font>
        <font weight="400" style="italic">Roboto-Italic.ttf</font>
        <font weight="500" style="normal">Roboto-Medium.ttf</font>
        <font weight="500" style="italic">Roboto-MediumItalic.ttf</font>
        <font weight="900" style="normal">Roboto-Black.ttf</font>
        <font weight="900" style="italic">Roboto-BlackItalic.ttf</font>
        <font weight="700" style="normal">Roboto-Bold.ttf</font>
        <font weight="700" style="italic">Roboto-BoldItalic.ttf</font>
    </family>
    <!-- Note that aliases must come after the fonts they reference. -->
    <alias name="sans-serif-thin" to="sans-serif" weight="100" />
    <alias name="sans-serif-light" to="sans-serif" weight="300" />
    <alias name="sans-serif-medium" to="sans-serif" weight="500" />
    <alias name="sans-serif-black" to="sans-serif" weight="900" />
    <alias name="arial" to="sans-serif" />
    <alias name="helvetica" to="sans-serif" />
    <alias name="tahoma" to="sans-serif" />
    <alias name="verdana" to="sans-serif" />
    <family name="sans-serif-condensed">
        <font weight="300" style="normal">RobotoCondensed-Light.ttf</font>
        <font weight="300" style="italic">RobotoCondensed-LightItalic.ttf</font>
        <font weight="400" style="normal">RobotoCondensed-Regular.ttf</font>
        <font weight="400" style="italic">RobotoCondensed-Italic.ttf</font>
        <font weight="500" style="normal">RobotoCondensed-Medium.ttf</font>
        <font weight="500" style="italic">RobotoCondensed-MediumItalic.ttf</font>
        <font weight="700" style="normal">RobotoCondensed-Bold.ttf</font>
        <font weight="700" style="italic">RobotoCondensed-BoldItalic.ttf</font>
    </family>
    <alias name="sans-serif-condensed-light" to="sans-serif-condensed" weight="300" />
    <alias name="sans-serif-condensed-medium" to="sans-serif-condensed" weight="500" />
    <family name="monospace">
        <font weight="400" style="normal">DroidSansMono.ttf</font>
    </family>
    <family lang="zh-Hans">
        <font weight="400" style="normal" index="2">NotoSansCJK-Regular.ttc</font>
    </family>
    <!-- TODO: Add Bopo -->
    <family lang="zh-Hant">
        <font weight="400" style="normal" index="3">NotoSansCJK-Regular.ttc</font>
    </family>
    <family lang="ja">
        <font weight="400" style="normal" index="0">NotoSansCJK-Regular.ttc</font>
    </family>
    <family lang="ko">
        <font weight="400" style="normal" index="1">NotoSansCJK-Regular.ttc</font>
    </family>
</familyset>
```

2.6 系统功优化

如果客户没有需求，可以在“frameworks/base/packages/SettingsProvider/res/values/defaults.xml”中将蓝牙、WiFi、锁屏、网络统计等等功能默认关闭。

```
wangwenrui@dev171:~/work_sapce-2t/SIM8501_A8_20220218/frameworks/base/packages/SettingsProvider$ git
diff --git a/packages/SettingsProvider/res/values/defaults.xml b/packages/SettingsProvider/res/value
old mode 100644
new mode 100755
index 4deb987..00d2965
--- a/packages/SettingsProvider/res/values/defaults.xml
+++ b/packages/SettingsProvider/res/values/defaults.xml
@@ -47,7 +47,7 @@
     -->
     <string name="def_location_providers_allowed" translatable="false">gps</string>
     <bool name="assisted_gps_enabled">true</bool>
-    <bool name="def_netstats_enabled">true</bool>
+    <bool name="def_netstats_enabled">false</bool>
     <bool name="def_usb_mass_storage_enabled">true</bool>
     <bool name="def_wifi_on">false</bool>
     <!-- 0 == never, 1 == only when plugged in, 2 == always -->
@@ -82,7 +82,7 @@
     <string name="def_trusted_sound" translatable="false">/system/media/audio/ui/Trusted.ogg</strin
     <string name="def_wireless_charging_started_sound" translatable="false">/system/media/audio/ui/

-    <bool name="def_lockscreen_disabled">false</bool>
+    <bool name="def_lockscreen_disabled">true</bool>
     <bool name="def_device_provisioned">false</bool>
     <integer name="def_dock_audio_media_enabled">1</integer>
```

如果客户没有需求，可以在“frameworks/base/core/res/res/values/config.xml”中关闭屏幕自动旋转、屏幕保护、多窗口等功能。

```
diff --git a/core/res/res/values/config.xml b/core/res/res/values/config.xml
old mode 100644
new mode 100755
index 7e5d0fb..4db4594
--- a/core/res/res/values/config.xml
+++ b/core/res/res/values/config.xml
@@ -698,7 +698,7 @@
          settings are omitted from the system UI.  In certain situations we may
          still use the accelerometer to determine the orientation, such as when
          docked if the dock is configured to enable the accelerometer. -->
-    <bool name="config_supportAutoRotation">true</bool>
+    <bool name="config_supportAutoRotation">false</bool>

     <!-- If true, the screen can be rotated via the accelerometer in all 4
          rotations as the default behavior. -->
@@ -1881,7 +1881,7 @@
          Consider setting this resource to false or disabling dreams by default when a
          doze component is specified below since dreaming will supercede dozing and
          will prevent the system from entering a low power state until the dream ends. -->
-    <bool name="config_dreamsSupported">true</bool>
+    <bool name="config_dreamsSupported">false</bool>

     <!-- If supported, are dreams enabled? (by default) -->
     <bool name="config_dreamsEnabledByDefault">true</bool>
@@ -2803,10 +2803,10 @@

     <!-- True if the device supports at least one form of multi-window.
          E.g. freeform, split-screen, picture-in-picture. -->
-    <bool name="config_supportsMultiWindow">true</bool>
+    <bool name="config_supportsMultiWindow">false</bool>

     <!-- True if the device supports split screen as a form of multi-window. -->
-    <bool name="config_supportsSplitScreenMultiWindow">true</bool>
+    <bool name="config_supportsSplitScreenMultiWindow">false</bool>

     <!-- True if the device supports running activities on secondary displays. -->
     <bool name="config_supportsMultiDisplay">true</bool>
wangwenrui@dev171:~/work_sapce-2t/SIM8501_A8_20220218/frameworks/base/core/res/res$
```

2.7 编译参数优化

Android 在首次启动和首次安装应用时，需要将字节码翻译成机器码，这样 Android 系统的启动速度将会大大减慢，如果没有预优化，APP 的运行速度也会加上翻译所需要的时间。所以，这个翻译的工作需要转移到编译上面来，也就是所，在编译 APK 文件时，将会预先对 APK 进行翻译的优化，然后再打包到系统里面去，这样 Android 系统在首次启动时，就不再需要花费大量的时间去翻译 APK 的字节码。

可以在“device/sprd/sharkle/sl8541e_1h10_32b/sl8541e_1h10_32b_Natv.mk”文件中添加如下参数实现上述优化目的：

```
diff --git a/sl8541e_1h10_32b/sl8541e_1h10_32b_Natv.mk b/sl8541e_1h10_32b/sl8541e_1h10_32b_Natv.mk
index cbe8dcc..56bd8f4 100755
--- a/sl8541e_1h10_32b/sl8541e_1h10_32b_Natv.mk
+++ b/sl8541e_1h10_32b/sl8541e_1h10_32b_Natv.mk
@@ -199,3 +199,6 @@ PRODUCT_ODMKO_KO_DIRS := \

 PRODUCT_ODMKO_KO_NAMES :=
 endif
+
+WITH_DEXPREOPT := true
+WITH_DEXPREOPT_PIC := true
\ No newline at end of file
```

3 Kernel 优化

3.1 关闭不使用的驱动

SIM8500 的驱动代码在 bsp 路径下，主要为 kernel 和 modules 两个目录，通过串口 log 信息，可以去看到那些外设驱动加载了，通过 error 信息或对比原理图，看那些外设实际上是没有的就可以关闭掉。

3.1.1 关闭 kernel 路径下驱动功能

关闭 kernel 里边的驱动，两种方式可选其一，一种方式是在 dts 里关闭，一种是在 deconfig 里关闭宏控不让驱动参与编译。

（1） dts 里关闭，dts 路径为：bsp/kernel/kernel4.14/arch/arm/boot/dts/sl8541e-1h10_xxx.dts 相关联 dts 和 dtsi 文件，例如，要关闭 i2c4 通道下挂载的充电芯片 fan54015_chg，只需要把该设备节点的状态

设置为 disabled 就可以，如下：

```
&i2c4 {
        status = "okay";
        clock-frequency = <400000>;

        flash_ic: flash-ic@63 {
                compatible = "sprd,flash-ocp8137";
                reg = <0x63>;
                status = "disabled";
                sprd,flash-ic = <8137>;
                sprd,torch = <1>;
                sprd,preflash = <1>;
                sprd,highlight = <1>;
                sprd,torch-level = <128>;
                sprd,preflash-level = <128>;
                sprd,highlight-level = <128>;
                sprd,lvfm-enable = <1>;
                flash-torch-en-gpios = <&ap_gpio 88 0>;
                flash-chip-en-gpios = <&ap_gpio 89 0>;
                flash-en-gpios = <&ap_gpio 76 0>;
                flash-sync-gpios = <&ap_gpio 141 0>;
        };

        fan54015_chg: charger@6a {
                compatible = "fairchild,fan54015_chg";
                reg = <0x6a>;
                status = "disabled";
                phys = <&hsphy>;
                monitored-battery = <&bat>;
                extcon = <&extcon_gpio>;
                vddvbus:otg-vbus {
                        regulator-name = "vddvbus";
                };
        };
};
```

添加状态为 disabled

这样设置后，对应” fairchild,fan54015_chg”的驱动为：bsp/kernel/kernel4.14 /drivers/power/supply/fan54015-charger.c，驱动在加载时匹配名称为”fairchild,fan54015_chg”的设备节点时，匹配不成功就不会加载该驱动。

（2） deconfig 里关闭，deconfig 文件路径为：bsp\kernel\kernel4.14\arch\arm\configs\sprd_sharkle_defconfig，例如，要关闭 usb 转网卡驱动 rtl8152.c，驱动路径为：bsp/kernel/kernel4.14/drivers/netusb/r8152.c，驱动参与编译依赖的宏控在 bsp/kernel/kernel4.14/drivers/netusb/Makefile 中可以看到为 CONFIG_USB_RTL8152：

```
# SPDX-License-Identifier: GPL-2.0
#
# Makefile for USB Network drivers
#

obj-$(CONFIG_USB_CATC)          += catc.o
obj-$(CONFIG_USB_KAWETH)        += kaweth.o
obj-$(CONFIG_USB_PEGASUS)       += pegasus.o
obj-$(CONFIG_USB_RTL8150)       += rtl8150.o
obj-$(CONFIG_USB_RTL8152)       += r8152.o
obj-$(CONFIG_USB_HSO)           += hso.o
obj-$(CONFIG_USB_LAN78XX)       += lan78xx.o
obj-$(CONFIG_USB_NET_AX8817X)   += asix.o
asix-y := asix_devices.o asix_common.o ax88172a.o
obj-$(CONFIG_USB_NET_AX88179_178A)      += ax88179_178a.o
obj-$(CONFIG_USB_NET_CDCETHER)  += cdc_ether.o
obj-$(CONFIG_USB_NET_CDC_EEM)   += cdc_eem.o
obj-$(CONFIG_USB_NET_DM9601)    += dm9601.o
obj-$(CONFIG_USB_NET_SR9700)    += sr9700.o
obj-$(CONFIG_USB_NET_SR9800)    += sr9800.o
obj-$(CONFIG_USB_NET_SMSC75XX)  += smsc75xx.o
obj-$(CONFIG_USB_NET_SMSC95XX)  += smsc95xx.o
obj-$(CONFIG_USB_NET_GL620A)    += gl620a.o
obj-$(CONFIG_USB_NET_NET1080)   += net1080.o
obj-$(CONFIG_USB_NET_PLUSB)     += plusb.o
obj-$(CONFIG_USB_NET_RNDIS_HOST)        += rndis_host.o
```

不让该驱动参与编译的话，在 bsp\kernel\kernel4.14\arch\arm\configs\sprd_sharkle_defconfig 中把该宏控设置为 n 或注释掉：

```
CONFIG_PPP_MULTILINK=y
CONFIG_PPPOE=y
CONFIG_PPTP=y
CONFIG_PPPOL2TP=y
CONFIG_PPP_ASYNC=y
# CONFIG_PPP_SYNC_TTY is not set
# CONFIG_SLIP is not set
CONFIG_SLHC=y
CONFIG_USB_NET_DRIVERS=y
# CONFIG_USB_CATC is not set
# CONFIG_USB_KAWETH is not set
# CONFIG_USB_PEGASUS is not set
# CONFIG_USB_RTL8150 is not set
CONFIG_USB_RTL8152=n          <-- 把y改为n
# CONFIG_USB_LAN78XX is not set
# CONFIG_USB_USBNET is not set
# CONFIG_USB_HSO is not set
# CONFIG_USB_IPHETH is not set
CONFIG_WLAN=y
# CONFIG_WIRELESS_WDS is not set
# CONFIG_WLAN_VENDOR_ADMTEK is not set
# CONFIG_WLAN_VENDOR_ATH is not set
# CONFIG_WLAN_VENDOR_ATMEL is not set
# CONFIG_WLAN_VENDOR_BROADCOM is not set
# CONFIG_WLAN_VENDOR_CISCO is not set
```

3.1.2 关闭 modules 路径下驱动功能

module 下的主要包含 camera、gpu、fingerprint、sensor、tp、wifi、bt、FM 等功能的驱动，是否让参与编译的配置文件为：

bsp\device\sharkle\androidq\sl8541e_1h10_32b\sl8541e_1h10_32b_base\modules.cfg

需要的功能保留，不需要的功能，注释或删除掉对应的.ko 项。

3.2 关闭 debug 信息

SIM8500 的 user 版本和 userdebug 版本，deconfig 文件里的 debug 宏控也是也有差异的，user 版本关闭了很多调试宏控；

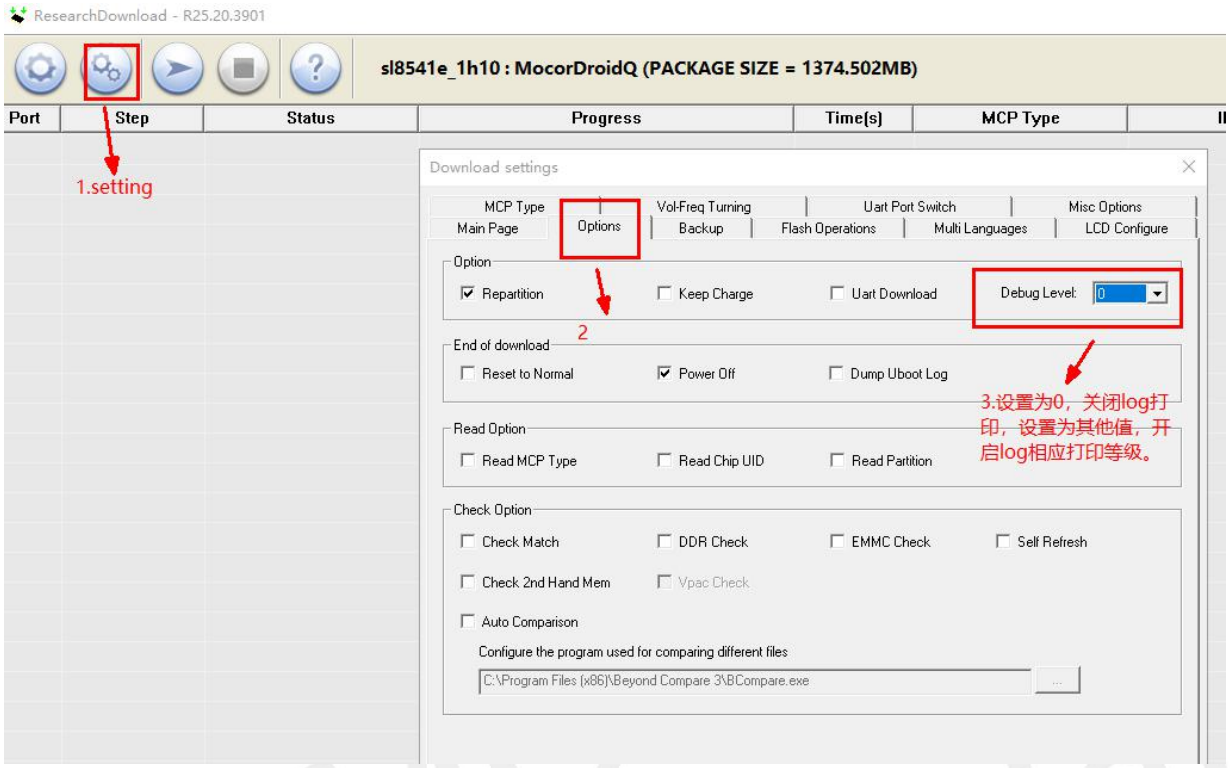
User 宏控和 userdebug 宏控具体差异点在：

bsp/kernel/kernel4.14/sprd-diffconfig/androidq/sharkle/arm/aarch32_user_diff_config 文件里。

如果编译 user 版本，那么，deconfig 里的 debug 宏可以不用操作，编译 userdebug 版本的话，可以参照 aarch32_user_diff_config 文件把 debug 宏关闭。

3.3 关闭串口 log 打印

串口 log 的打印，延长了较多的开机时间，SIM8500 的串口 log 的关闭，需要在下载版本的时候，在下载工具里设置关闭，设置配置如下：



设置后之后，再下载版本，再重新上电开机，就可以看到内核 log 关闭了。

4 总结

开机时间优化，主要几点：

1、主要是根据产品的软硬件需求，把代码里开启的功能，实际不用的，裁剪掉；

2、关闭串口 log 打印；

3、使用 user 版本。

如在具体产品的开机时间优化中，有任何问题和建议，请联系我们，谢谢！