



# Android 15 GPS 开发指南

版本号: 1.6

发布日期: 2025.02.18

## 版本历史

版本号	日期	制/修订人	内容描述
1.0	2023.06.10	AWA0989	初始版本
1.1	2023.07.24	AWA1586	新增 GPS 模组
1.2	2023.11.10	AWA1586	修改部分描述
1.3	2024.06.20	AWA1586	新增 GPS 模组
1.5	2025.02.10	AWA0989	适配 Android15, 删除无需配置的 selinux 相关内容
1.6	2025.02.18	AWA0989	更正 AT6558R 错误的波特率配置描述

# 目 录

<b>1 前言</b>	<b>1</b>
1.1 编写目的	1
1.2 适用范围	1
1.3 相关人员	1
1.4 相关术语介绍	1
1.5 注意事项	1
<b>2 内核驱动配置</b>	<b>2</b>
2.1 公共驱动	2
2.1.1 sunxi-rf 驱动	2
<b>3 DTS 配置</b>	<b>3</b>
3.1 RF-KILL 供电	3
3.2 GPS 通信接口 UART	4
<b>4 Android 配置</b>	<b>5</b>
4.1 init.gnss.rc	5
4.2 config.mk	5
4.3 GNSS HAL 实现	6
4.3.1 service	6
4.3.2 hals	8
<b>5 开发验证</b>	<b>9</b>

# 1 前言

## 1.1 编写目的

本文旨在介绍 Android 系统上 GPS 模组配置方法，让 GPS 模块开发和使用人员可以根据文档完成 GPS 的常规配置工作，解决常见问题。

## 1.2 适用范围

本文档适用于 Android GPS 开发。

## 1.3 相关人员

GPS 相关的 HAL，SERVICE 及驱动开发人员。

## 1.4 相关术语介绍

Table: GPS 相关术语介绍

术语	解释说明
SUNXI	Allwinner 一系类 SOC 硬件平台
GPS	Global Positioning System 全球定位系统

## 1.5 注意事项

GPS 相关配置文件及 hal 服务均属于 Vendor。对于 GRF SDK，请修改 Vendor SDK 下的相关内容。

## 2 内核驱动配置

### 2.1 公共驱动

公共驱动所有厂家模组都必须配置。具体编译进内核或是编译成模块，需要根据系统需求定义。一般的，在支持 GKI（Generic Kernel Image）的平台上必须配置成模块，其他平台可选。

#### 2.1.1 sunxi-rf 驱动

此驱动为电控制接口的公共驱动，所有模组都将配置（**模组为常供电连接方式除外**）。

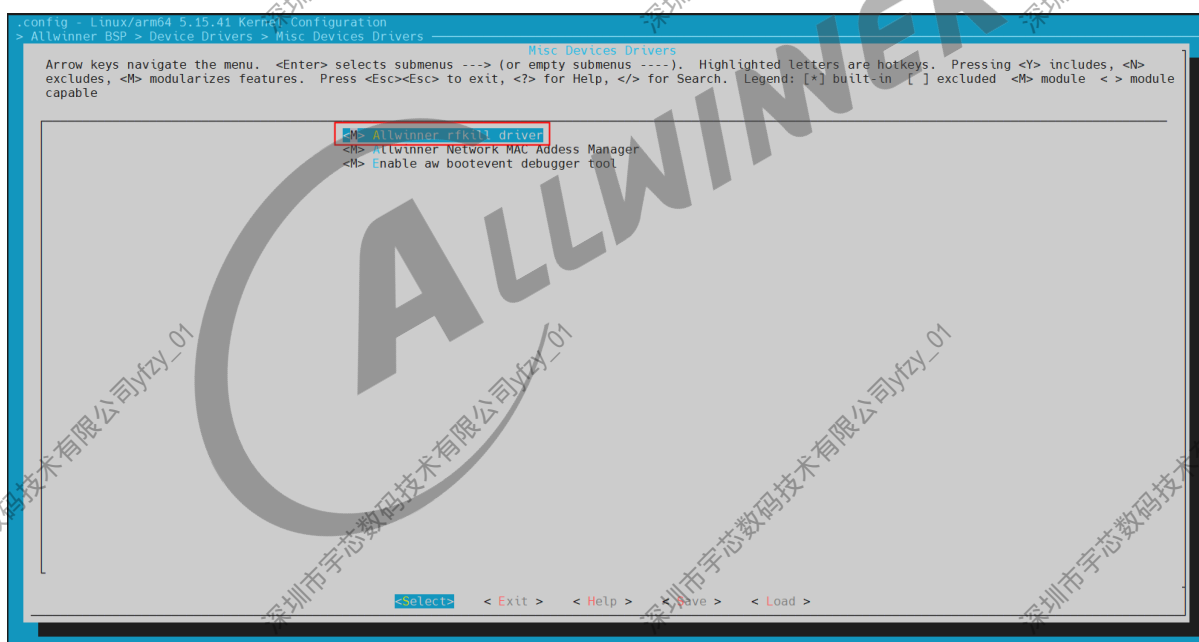


图 2-1: sunxi-rf 驱动配置

```

xxx@GZExdroid04:${SDK}/longan/bsp/drivers/misc/sunxi-rf$ tree
├── internal.h
├── Kconfig
├── Makefile
├── sunxi-gnss.c # gps上电控制代码
├── sunxi-modem.c
├── sunxi-rfskill.c # 主模块
└── sunxi-rfskill.h

```

## 3 DTS 配置

**注意**，若模组为常供电连接方式，则无需配置此部分内容。

文件路径：\${SDK}/longan/device/config/chips/{IC}/configs/{BOARD}/linux-{LINUX-VERSION}/board.dts

### 3.1 RF-KILL 供电

RF-KILL 上电控制相关参考配置如下：

```
&rfskill {
    compatible = "allwinner,sunxi-rfskill";
    chip_en;
    power_en = <&rpio PL 7 GPIO_ACTIVE_HIGH>;
    pinctrl-0;
    pinctrl-names;
    status = "okay";
    /* ... other drivers */
    gnss {
        compatible = "allwinner,sunxi-gnss";
        gnss_regon = <&pico PB 2 GPIO_ACTIVE_HIGH>;
        gnss_rst = <&pico PB 3 GPIO_ACTIVE_LOW>;
    };
};
```

表 3-1: RF-KILL 配置项说明

属性路径	配置项	值含义
rfskill	chip_en	模组使能引脚，硬件未使用时则无需配置
rfskill	power_en	模组外部的电源开关控制引脚，硬件未使用则无需配置
rfskill	pinctrl-0	SOC 特殊引脚配置，如 DCXO pin 复用等，由平台定义
rfskill	pinctrl-names	同上
rfskill	status	表示使用 rfskill 驱动
rfskill/wlan	...	(Wi-Fi 相关配置，GPS 无需关注)
rfskill/gnss	compatible	固定值，请勿修改
rfskill/gnss	gnss_regon	GPS 芯片/上电使能控制引脚
rfskill/bt	gnss_rst	GPS 芯片 reset 控制引脚

**⚠ 注意**

以上所有项必须参看原理图进行配置，修改配置与原理图实际使用的资源保持一致。

## 3.2 GPS 通信接口 UART

下文都以 UART2 为 GPS 为通信接口来描述，若是 GPS 选用其他 uart 接口，可类比处理。

UART 参考配置如下：

```
&uart2 {
    pinctrl-names = "default", "sleep";
    pinctrl-0 = <&uart2_pins_a>;
    pinctrl-1 = <&uart2_pins_b>;
    uart2_type = <2>;
    status = "okay";
};
```

表 3-2: BT-LPM 配置项说明

属性路径	配置项	值含义
uart2	compatible	固定值，请勿修改
uart2	pinctrl-0	uart active 时的 pin 配置，可根据实际改 uart2_pins_a 节点的值
uart2	pinctrl-1	uart sleep 时的 pin 配置，可根据实际改 uart2_pins_b 节点的值
uart2	uart2_type	标识使用 2 线 uart，无硬件流控
uart2	status	表示使能 uart2 驱动

## 4 Android配置

### 4.1 init.gnss.rc

文件路径： android/device/softwinner/{DEVICE}/{PRODUCT}/gnss/init.gnss.rc

#### 说明

此文件配置 GPS 使用的 uart 端口文件权限，及模块上电配置。由于用了系统属性获取配置，因此不同方案一般无差异，无需修改：

```
on early-boot
  chmod 0660 ${persist.vendor.gnss_port}
  chown gps system ${persist.vendor.gnss_port}
  write /sys/devices/virtual/misc/sunxi-rfkill/gnss/state 1
```

### 4.2 config.mk

文件路径： android/device/softwinner/{DEVICE}/{PRODUCT}/gnss/config.mk

#### 说明

此文件配置系统编译的 packages，以及要拷贝到固件的文件。

文件内容如下：

```
LOCAL_MODULE_PATH := $(shell dirname $(lastword $(MAKEFILE_LIST)))
PRODUCT_COPY_FILES += \
  frameworks/native/data/etc/android.hardware.location.gps.xml:${TARGET_COPY_OUT_VENDOR}/etc/permissions/
  android.hardware.location.gps.xml \
  $(LOCAL_MODULE_PATH)/init.gnss.rc:${TARGET_COPY_OUT_VENDOR}/etc/init/init.gnss.rc

CONFIG_AW_GNSS_PORT := /dev/ttyAS2
CONFIG_AW_GNSS_BAUD := 115200 # TD1030: 9600, TD1050: 115200, CC1161W: 115200
CONFIG_AW_GNSS_DEVICE := bdstar

PRODUCT_VENDOR_PROPERTIES += persist.vendor.gnss_baud=$(CONFIG_AW_GNSS_BAUD)
PRODUCT_VENDOR_PROPERTIES += persist.vendor.gnss_port=$(CONFIG_AW_GNSS_PORT)
PRODUCT_VENDOR_PROPERTIES += ro.hardware.gps=$(CONFIG_AW_GNSS_DEVICE)

PRODUCT_PACKAGES += android.hardware.gnss-aidl-service.aw
PRODUCT_PACKAGES += gps.$(CONFIG_AW_GNSS_DEVICE)
```

其中，CONFIG\_AW\_GNSS\_PORT 为使用的 uart 设备节点，请依据实际硬件连接进行修改；CONFIG\_AW\_GNSS\_BAUD 为 GPS 芯片使用的波特率；CONFIG\_AW\_GNSS\_DEVICE 为当前 GPS 使用的 hal 名称。

目前支持的芯片厂商、芯片型号、波特率与 hal 对应关系如下：

表 4-1: GPS 不同芯片配置

芯片厂商	芯片型号	波特率	hal 名称
techtotop	TD1030	9600	techtotop
techtotop	TD1050	115200	techtotop
icofchina	AT6558R	9600	icofchina
bdstar	CC1161W	115200	bdstar
allystar	HD8040	115200	allystar

## 4.3 GNSS HAL 实现

Android13 开始，Google 要求厂商使用 AIDL-HAL，请不要随意切换回 HIDL 版本。AW GNSS HAL 存放在\${SDK}/hardware/aw/gnss 路径下，包含两个重要的路径：

```
build@Exdroid:${SDK}/hardware/aw/gnss$ tree -L 1
├── hals
└── service
```

### 4.3.1 service

interface，定义 framework 与 vendor hal 通信时的接口。由 Android 定义其标准 API，aw 做了具体的实现，厂商无需改动。

service 要求厂商提供的具体接口包含在 GpsCallbacks 结构体内，除了 acquire\_wakelock\_cb、release\_wakelock\_cb，均需要厂商实现。接口主要描述如下：

```
/**
 * Callback with location information. Can only be called from a thread created
 * by create_thread_cb.
 */
typedef void (* gps_location_callback)(GpsLocation* location);

/**
 * Callback with status information. Can only be called from a thread created by
 * create_thread_cb.
 */
typedef void (* gps_status_callback)(GpsStatus* status);

/**
 * Legacy callback with SV status information.
 * Can only be called from a thread created by create_thread_cb.
 *
 * This callback is deprecated, and will be removed in the next release. Use
 * gnss_sv_status_callback() instead.
 */
```

```
typedef void (* gps_sv_status_callback)(GpsSvStatus* sv_info);  
/**  
 * Callback for reporting NMEA sentences. Can only be called from a thread  
 * created by create_thread_cb.  
 */  
typedef void (* gps_nmea_callback)(GpsUtcTime timestamp, const char* nmea, int length);  
/**  
 * Callback to inform framework of the GPS engine's capabilities. Capability  
 * parameter is a bit field of GPS_CAPABILITY_* flags.  
 */  
typedef void (* gps_set_capabilities)(uint32_t capabilities);  
/**  
 * Callback utility for acquiring the GPS wakelock. This can be used to prevent  
 * the CPU from suspending while handling GPS events.  
 */  
typedef void (* gps_acquire_wakelock)();  
/** Callback utility for releasing the GPS wakelock. */  
typedef void (* gps_release_wakelock)();  
/**  
 * Callback for creating a thread that can call into the Java framework code.  
 * This must be used to create any threads that report events up to the  
 * framework.  
 */  
typedef pthread_t (* gps_create_thread)(const char* name, void (*start)(void*), void* arg);  
/** Callback for requesting NTP time */  
typedef void (* gps_request_utc_time)();  
/**  
 * Callback to inform framework of the engine's hardware version information.  
 */  
typedef void (*gnss_set_system_info)(const GnssSystemInfo* info);  
/**  
 * Callback with SV status information.  
 * Can only be called from a thread created by create_thread_cb.  
 */  
typedef void (* gnss_sv_status_callback)(GnssSvStatus* sv_info);  
/** New GPS callback structure. */  
typedef struct {  
    /** set to sizeof(GpsCallbacks) */  
    size_t size;  
    gps_location_callback location_cb;  
    gps_status_callback status_cb;  
    gps_sv_status_callback sv_status_cb;  
    gps_nmea_callback nmea_cb;  
    gps_set_capabilities set_capabilities_cb;  
    gps_acquire_wakelock acquire_wakelock_cb;  
    gps_release_wakelock release_wakelock_cb;  
    gps_create_thread create_thread_cb;  
    gps_request_utc_time request_utc_time_cb;  
  
    gnss_set_system_info set_system_info_cb;  
    gnss_sv_status_callback gnss_sv_status_cb;
```

```
} GpsCallbacks;
```

### 4.3.2 hals

vendor hal，由 gps 厂商提供。下面一个子目录代表一个厂商的 vendor hal，具体厂商、芯片、hal 对应关系见前述章节。



## 5 开发验证

编译固件烧录在 gps 的样机中，若是适配好 gps 的系统，在系统起来后，对应的服务会起来，如下

```
console:/ # ps -A | grep gnss
system    309    1 2339584 10496 binder_wait_for_work 0 S android.hardware.gnss-aidl-service.aw
console:/ #
```

根据硬件连接图连接天线和连接网络（方便地图更新），把天线收发器置于室外，安装 GpsTest apk，即可查看到星历图及定位信息。

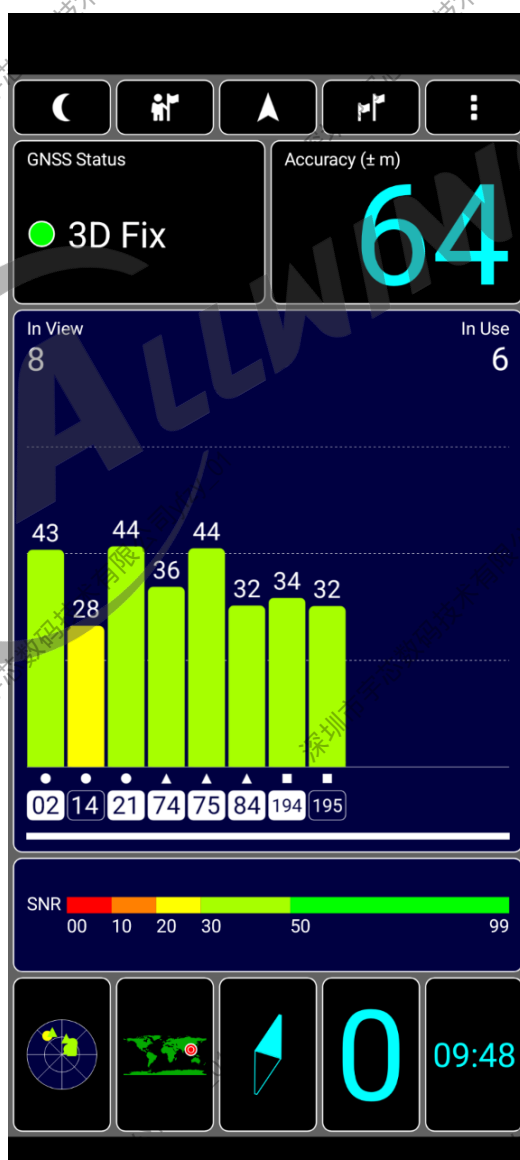


图 5-1: gps\_location

 **注意**

若无法定位，可先排查测试环境是否空旷和天线是否连接好，如模组数据存在问题，请联系 GPS 厂商处理。

若需查看 nmea 原始数据，可直接 cat 对应的 uart 设备节点，如：

```
console:/ # cat /dev/ttyAS2
```






## 著作权声明

版权所有 ©2025 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。