Android ALL

GPS porting Guide on the Android

TCCxxx-Android-All

-V1.01E-GPS Porting Guide

Feb. 2014

DISCLAIMER

All information and data contained in this material are without any commitment, are not to be considered as an offer for conclusion of a contract, nor shall they be construed as to create any liability. Any new issue of this material invalidates previous issues. Product availability and delivery are exclusively subject to our respective order confirmation form; the same applies to orders based on development samples delivered. By this publication, Telechips, Inc. does not assume responsibility for patent infringements or other rights of third parties that may result from its use.

Further, Telechips, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of Telechips, Inc.

This product is designed for general purpose, and accordingly customer be responsible for all or any of intellectual property licenses required for actual application. Telechips, Inc. does not provide any indemnification for any intellectual properties owned by third party.

Telechips, Inc. can not ensure that this application is the proper and sufficient one for any other purposes but the one explicitly expressed herein. Telechips, Inc. is not responsible for any special, indirect, incidental or consequential damage or loss whatsoever resulting from the use of this application for other purposes.

COPYRIGHT STATEMENT

Copyright in the material provided by Telechips, Inc. is owned by Telechips unless otherwise noted.

For reproduction or use of Telechips' copyright material, permission should be sought from Telechips. That permission, if given, will be subject to conditions that Telechips' name should be included and interest in the material should be acknowledged when the material is reproduced or quoted, either in whole or in part. You must not copy, adapt, publish, distribute or commercialize any contents contained in the material in any manner without the written permission of Telechips. Trade marks used in Telechips' copyright material are the property of Telechips.

Important Notice

For customers who use licensed Codec ICs and/or licensed codec firmware of mp3:

"Supply of this product does not convey a license nor imply any right to distribute content created with this product in revenue-generating broadcast systems (terrestrial. Satellite, cable and/or other distribution channels), streaming applications(via internet, intranets and/or other networks), other content distribution systems(pay-audio or audio-on-demand applications and the like) or on physical media(compact discs, digital versatile discs, semiconductor chips, hard drives, memory cards and the like). An independent license for such use is required. For details, please visit http://mp3licensing.com" .

For customers who use other firmware of mp3:

"Supply of this product does not convey a license under the relevant intellectual property of Thomson and/or Fraunhofer Gesellschaft nor imply any right to use this product in any finished end user or ready-to-use final product. An independent license for such use is required. For details, please visit http://mp3licensing.com" .

For customers who use Digital Wave DRA solution:

"Supply of this implementation of DRA technology does not convey a license nor imply any right to this implementation in any finished end-user or ready-to-use terminal product. An independent license for such use is required."

For customers who use DTS technology:

"This product made under license to certain U.S. patents and/or foreign counterparts."

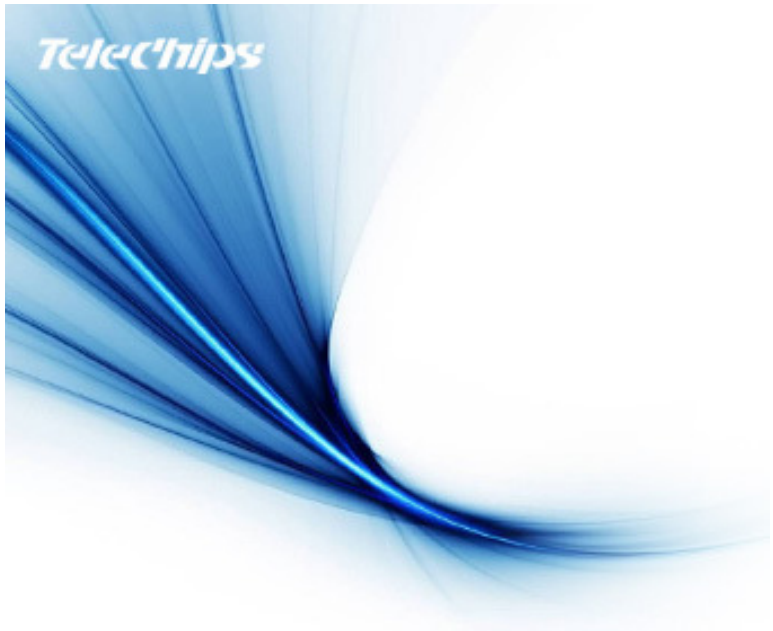
"© 1996 – 2011 DTS, Inc. All rights reserved."

For customers who use Dolby technology:

"Supply of this Implementation of Dolby technology does not convey a license nor imply a right under any patent, or any other industrial or intellectual property right of Dolby Laboratories, to use this Implementation in any finished end-user or ready-to-use final product. It is hereby notified that a license for such use is required from Dolby Laboratories."

For customers who use MS technology:

"This product is subject to certain intellectual property rights of Microsoft and cannot be used or disctributed further without the appropriate license(s) from Microsoft."

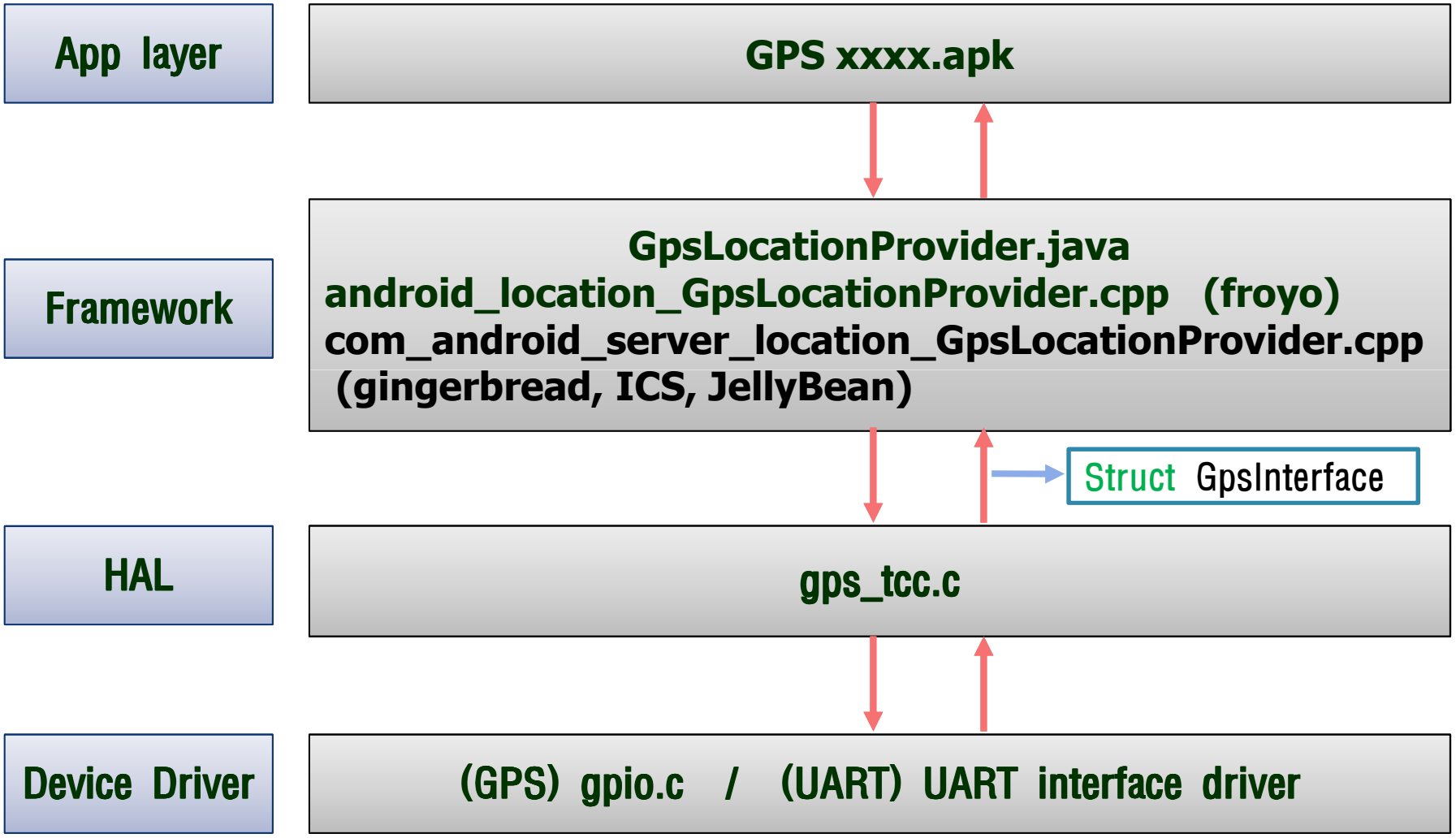


Index

Revision History

Date	Version	Description
2013-03-14	1.00	Initial JellyBean Release
2014-02-13	1.01	Initial Kitkat Beta Release

Structure


App layer
GPS xxxx.apk
Framework
GpsLocationProvider.java
android_location_GpsLocationProvider.cpp (froyo)
com_android_server_location_GpsLocationProvider.cpp
(gingerbread, ICS, JellyBean)
Struct GpsInterface
HAL
gps_tcc.c
Device Driver
(GPS) gpio.c / (UART) UART interface driver

Device Driver

- File paths : android/kernel/drivers/gps/
- gpio.c (gps_gpio.ko)
  - This file have function to control GPIO port which is enable to GPS module.
  - Open() is function to initialization of GPS device driver.
    Please don't enable the power of GPS module to reduce the current at sleep/idle state.
  - the ioctl() function controls the power of GPS module.

HAL

- File paths : android/hardware/telechips/common/libgps/
- gps_tcc.c
  - Refer to gps_qemu.c to porting GPS HAL layer.
  - The gps_stat_init() of this file calls open() to enable GPS device driver. (gps_gpio, $uart_port)

    ex) $uart_port -> ttyTCC3
  - When system stay on idle status, the gps_state_start() and gps_state_stop() call the ioctl() of gpio.C.

HAL

- File paths : android/hardware/libhardware/include
- gps.h
  - This file has defines and structures and it is related to GPS. (callback-related)
  - Refer to gps_tcc.c of HAL layer.
  - Refer to android_location_GpsLocationProvider.cpp(froyo) of Frameworks
    com_android_server_location_GpsLocationProvider.cpp(gingerbread, ICS)

```
/** Represents the standard GPS interface. */
typedef struct {
    int   (*init)( GpsCallbacks* callbacks );
    int   (*start)( void );
    int   (*stop)( void );
    void  (*set_fix_frequency)( int frequency );
    void  (*cleanup)( void );
    int   (*inject_time)(GpsUtcTime time, int64_t timeReference,
                         int uncertainty);
    void  (*delete_aiding_data)(GpsAidingData flags);
    int   (*set_position_mode)(GpsPositionMode mode, int fix_frequency);
    const void* (*get_extension)(const char* name);
} GpsInterface;
```

gps.h

```
static const GpsInterface  tccGpsInterface = {
    tcc_gps_init,
    tcc_gps_start,
    tcc_gps_stop,
    tcc_gps_set_fix_frequency,
    tcc_gps_cleanup,
    tcc_gps_inject_time,
    tcc_gps_delete_aiding_data,
    tcc_gps_set_position_mode,
    tcc_gps_get_extension,
};
```

tcc_gps.c

  - The init(GpsCallbacks *) register the structure of GpsInterface and link with Frameworks and HAL layer.

File path : android/hardware/telechips/common/libgps/**gps_tcc.c**

- Checking opening UART3(ttyTCC3) & GPIO

```
state->fd = open(state->device, O_RDWR | O_NONBLOCK | O_NOCTTY);          // UART3
D("tcc : %s Device Open FDescriptor %d", state->device, state->fd);

if (state->fd < 0) {
    D("tcc : no gps Hardware detected");
    return;
}
```

```
state->fdGps = open("/dev/gps_gpio", O_RDWR);                              // Gps_GPIO
D("tcc : Gps_GPIO Device Open FDescriptor %d",state->fdGps);

if (state->fdGps < 0) {
    D("tcc : Couldn't open gps_gpio");
    return;
}
```

HAL

File path : android/hardware/telechips/common/libgps/**gps_tcc.c**
android/kernel/drivers/gps/**gpio.c**

- Checking GPIO setting of ioctl() in gps_state_start() & gps_state_stop()

```
static void
gps_state_start( GpsState*  s )
{
    char  cmd = CMD_START;
    int   ret;

    do { ret=write( s->control[0], &cmd, 1 ); }
    while (ret < 0 && errno == EINTR);

    if (ret != 1)
        D("%s: could not send CMD_START command: ret=%d: %s",
          __FUNCTION__, ret, strerror(errno));

    ret = ioctl(s->fdGps, 0); // 0 -> On, 1 -> Off              // TCC_GPS
    //GPS_GPIO드라이버의 IOCTL을 호출, GPS모듈을 ON/OFF함
}


static void
gps_state_stop( GpsState*  s )
{
    char  cmd = CMD_STOP;
    int   ret;

    do { ret=write( s->control[0], &cmd, 1 ); }
    while (ret < 0 && errno == EINTR);

    if (ret != 1)
        D("%s: could not send CMD_STOP command: ret=%d: %s",
          __FUNCTION__, ret, strerror(errno));

    ret = ioctl(s->fdGps, 1); // 0 -> On, 1 -> Off              // TCC_GPS
    //GPS_GPIO드라이버의 IOCTL을 호출, GPS모듈을 ON/OFF함
}
```

FAQ-01 Setting of UART

- Referenced baud rate of UART5 : 9600 baud rate (bps), at sirf.

There are 5EA by UART channels in 892x , 893X

but only UART0 ~ 3 can use DMA to transmit data.

The remaining UART channels (4,5) can not use DMA and there is a baud rate limitation. in case of using UART channels (4.5), the transmission data loss can be occurred by overrun of UART while data is transmitted faster than 9600 baud-rate.

If you want to use a GPS or other device which support baud rate faster than 9600, and if there is no unused UART channel which can support DMA transmission,

please refer to "Android-ALL-V1.00K-Uart-User Guide.pdf"