

GT9XX 驱动移植说明书

一、驱动基本信息

支持芯片型号	GT918、GT968、GT913、GT913P、GT927、GT927P、GT9110、GT9110P
I2C 设备地址(7 位)	0x5d、0x14（7bits 地址）
I2C 寄存器地址	16 位
支持 APK 工具?	支持
支持自动升级?	支持搜寻 bin 文件自动升级
固化配置	支持
支持 Sensor ID 数	6 个

二、驱动文件说明

一般情况下，驱动参考资料包的 *reference drivers* 文件夹下面包含以下几个文件，下面对每个文件的功能和使用方法进行说明：

1. **gt9xx.c(Required)**: 驱动主功能文件，用来实现驱动的挂载、读取上报坐标、休眠唤醒处理等触摸屏驱动的基本功能。
2. **gt9xx.h(Required)**: 驱动头文件，包含驱动中要用到的一些宏和常量的定义、外部变量和函数的声明等。
3. **gt9xx_update.c(Recommend)**: 驱动用于支持固件升级的文件，对于触摸屏驱动来说，该文件不是必需的，但是强烈推荐在驱动中增加该功能，以便于您使用的触控 IC 在必要时升级为最新版本的固件。
4. **goodix_tool.c(Recommend)**: 驱动中用于支持 *gtp_tools.apk* 工具的文件，该工具可以在装成整机后再 Android 上层对触控 IC 进行测试、调试、检测等功能，强烈推荐在驱动中增加此功能，特别是使用 COB（触控 IC 直接 layout 在主板上）模式的 TP 时，此工具能极大的方便整机上的 TP 调试。

二、驱动移植 STEP_BY_STEP

1. **复制文件**: 将 *reference driver* 文件夹中的所有文件复制到 kernel 的 *drivers/input/touchscreen/* 目录下。

2. **修改 Makefile:** 在 `drivers/input/touchscreen/` 目录下, 打开 `Makefile` 文件, 并在文件中增加以下条目 (注意不同的(.o)文件之间用空格分开):。

```
obj-y+=gt9xx.o gt9xx_update.o goodix_tool.o
```

3. **添加设备:** 找到 kernel 中初始化 I2C 总线的板级文件, 如本驱动的开发平台 real6410 开发板是位于 `arch/arm/mach-s3c6410/ mach-smdk6410.c` 文件中, 如需要将触摸屏驱动挂载 I2C0 总线上, 则按以下方法添加 TP 的 i2c 设备驱动即可, 0x5d 为该型号触控 IC 的 i2c 从设备地址, 具体为多少需参阅该型号芯片的 datasheet, Goodx-TS 为 i2c 设备驱动名, 必须与驱动参考代码中的 GTP_I2C_NAME 保持相同。

```
static struct i2c_board_info i2c_devs0[] __initdata =
{
    { I2C_BOARD_INFO("Goodix-TS", 0x5d), },
};
```

4. **修改参考代码:** 一般情况下, 移植过程中只需修改 `gt9xx.h` 文件中的内容即可, 打开该头文件, 按照注释中的提示移植, 重点关注 TODO part 的修改即可。

- (1) **STEP1 替换配置信息表 (REQUIRED):** 将对应于您正在使用 TP 的配置信息 (一般为 TP 厂提供的(*cfg 或*txt)文件里面的内容), 替换 CTP_CFG_GROUP 中的内容。

```
//STEP_1(REQUIRED):Change config table.
#define CTP_CFG_GROUP1 { \
    0x42, 0xE0, 0x01, 0x20, 0x03, 0x05, 0x14, 0x01, 0x02, 0x08, 0x19, 0x00, 0x50, \
    0x28, 0x03, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, \
    0x00, 0x90, 0x00, 0x0A, 0x48, 0x00, 0xF3, 0x0D, 0x00, 0x00, 0x00, 0x9A, 0x02, \
    0x2D, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x32, \
    0x96, 0x94, 0x85, 0x42, 0x08, 0x00, 0x00, 0xC1, 0x11, 0x1A, 0xB3, 0x15, 0x1C, \
    0xE6, 0x1B, 0x1C, 0x0B, 0x1E, 0x1E, 0x8D, 0x20, 0x21, 0x00, 0x00, 0x10, 0x28, \
    0x48, 0x00, 0x5F, 0x50, 0x30, 0xFF, 0xFF, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, \
    0x00, 0x1C, 0x96, 0x64, 0x0F, 0x00, 0x00, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, \
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0x0A, 0x0C, 0x0E, 0x10, \
    0x12, 0x14, 0x16, 0x18, 0x1A, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, \
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, \
    0x02, 0x04, 0x05, 0x06, 0x08, 0x0A, 0x0C, 0x1D, 0x1E, 0x1F, 0x20, 0x22, 0x24, \
```

```

0x28,0x29,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,\
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,\
0x00,0x00,0x00,0x00,0x00,0x14,0x1E,0x28,0x28,0x32,0x3C,0x3C,0x3C,\
0x50,0x50,0x50,0x32,0x50,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,\
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,\
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xDB,0x01\
}
//TODO puts your group2 config info here,if need.
#define CTP_CFG_GROUP2 {\
}
//TODO puts your group3 config info here,if need.
#define CTP_CFG_GROUP3 {\
}

```

注意事项:

- (1) 如果没有设置 Sensor ID (详见附录)，请务必将配置信息宏定义在 CTP_CFG_GROUP1，并保持其他几组为空，替换完成后，需要在每行后面增加宏定义的连接符“\”；
- (2) 如果实际使用的 sensor ID 数多于参考驱动中的 3 组，请参照这三组完成其他组的配置通过 sensor ID 来区分；
- (3) 如果配置宏的第一行设置了配置信息的写入寄存器 GTP_REG_CONFIG_DATA，则请将配置从第二行开始替换。

- (2) STEP2 修改 IO 定义和 IO 操作方式(REQUIRED)：将 GTP_INT_PORT 和 GTP_RST_PORT 的定义修改为对应于该项目的引脚定义，另外还需检查后面几个关于 IO 操作的语句是否适用于您正在使用的平台，如果不是，则需修改成相应的操作方式。

```

//STEP_2(REQUIRED):Change I/O define & I/O operation mode.
#define GTP_INT_PORT S3C64XX_GPN(15)
#define GTP_RST_PORT S3C64XX_GPL(10)
#define GTP_INT_IRQ gpio_to_irq(GTP_INT_PORT)
.....
#define GTP_GPIO_AS_INPUT(pin) do{\
    gpio_direction_input(pin);\
    s3c_gpio_setpull(pin, S3C_GPIO_PULL_NONE);\
}while(0)

```

注意事项：中断脚和复位脚应初始化为悬浮输入态。（悬浮：既不上拉，也不下拉）。

- (3) STEP3 客户自定义参数 (OPTIONAL)：如果您需要自己指定分辨率、中断触发方式、支持的最多 TOUCH 数等参数，请在 ON/OFF define 中打开 GTP_CUSTOM_CFG 宏，并参照以下修改参数。

```
//*****PART1:ON/OFF define*****
#define GTP_CUSTOM_CFG 1
//*****PART2:TODO define*****
.....
//STEP_3(optional):Custom set some config by custom,if need.
#if GTP_CUSTOM_CFG
    #define GTP_MAX_WIDTH 800
    #define GTP_MAX_HEIGHT 480
    #define GTP_MAX_TOUCH 5
    #define GTP_INT_TRIGGER 0
#else
    #define GTP_MAX_WIDTH 4096
    #define GTP_MAX_HEIGHT 4096
    #define GTP_MAX_TOUCH 5
    #define GTP_INT_TRIGGER 1
#endif
```

- (4) STEP4 配置触摸按键 (OPTIONAL)：如果您正在使用的 TP 带有触摸按键，则需要配置触摸按键，先在 ON/OFF define 中打开 GTP_HAVE_TOUCH_KEY 开关，然后再参照以下设置按键，按键的功能和顺序请在 GTP_KEY_TAB 中按需调整。

```
//*****PART1:ON/OFF define*****
#define GTP_HAVE_TOUCH_KEY 1
//*****PART2:TODO define*****
.....
//STEP_4(optional):If this project have touch key,Set touch key config.
#if GTP_HAVE_TOUCH_KEY
    #define GTP_KEY_TAB {KEY_MENU, KEY_HOME, KEY_SEND}
#endif
```

- (5) STEP5 增加包含文件 (OPTIONAL)：在该头文件的前面增加对应您使用平台所必须的#include

包含文件，这个步骤也是可选的，根据您的编译的情况按需加入。

三、附录

- Sensor ID:** 如果同一个项目中，使用几家 TP 厂的 TP，并且都使用 GOODIX 的同一款 IC，则可以对触控 IC 设置 Sensor ID，主机在初始化的时候发送相应 ID 的配置信息，从而区分不同厂家的 TP。Sensor ID 的设置方法一般是 layout 时对 IC 的某一个或者几个 IO 口进行上拉、下拉或者悬空等设置，每款芯片的设置方法有所差异，具体请参照各 IC 的 datasheet。
- IC 固件和配置信息:** 固件是 IC 内部运行的程序，固件是针对一款 IC 的，而配置信息则是在固件运行的前期对固件进行初始化的一个数组，主机上电后通过 I2C 发送给 IC，IC 才能正常运行，配置信息是针对一款 TP 的，TP 的结构、工艺、通道数等大部分修改都需要通过修改配置信息来适应。
- 固化配置:** 指芯片将收到的配置信息保存在 FLASH 或其他非易失性存储器里，从而主机不必上电时每次都发送配置信息，一般多芯片组合方案（如 GT8105 等）具有该功能，GT82X 系列可以通过命令实现固化配置的功能。固化配置可以给调试带来便利，但是不便于量产，建议在调试具有固化配置的 IC 时，可以根据需要适时使用固化配置功能，关闭发配置，但是调试完毕必须打开发配置功能。
- SLOT 报点方式:** 有些 android4.0 系统上层配置必须采用 SLOT 方式报点，此时若驱动依旧采取传统的报点方式，android 上层可能会将上报的坐标识别成相对坐标，如果出现这种现象，请将 GTP_ICS_SLOT_REPORT 宏打开，将报点方式切换到 SLOT 方式即可。详细内容请参考 linux 输入子系统和 android 上层 InputReader.cpp 中关于上报事件的相关资料。
- ESD 防护机制:** 是指在驱动中增加一个线程，来查询 IC 的工作状态，如果发现工作异常，则复位 IC，主要用于较强 ESD 条件下的避免 TP 失效，您可以根据 ESD 测试结果来决定是否打开该功能。
注意：该功能使用的前提是 CTP 芯片的 VDD 可由主机控制开关或主机可以通过 RESET 控制 CTP 芯片复位。
- 固件升级功能:** GT9XX 驱动支持两种固件升级方式：**a) 开机自动搜寻 bin 文件升级**，即驱动加载后，将创建一个线程，定时（默认间隔 3s）限定次数（默认搜寻 50 次）搜寻驱动中指定路径中的升级 bin 文件（默认路径为/data/goodix/_goodix_update_.bin），如果搜索到该 bin 文件，驱动检查文件合法性，如果满足要求，则自动升级该固件；**b) APK 浏览 bin 文件升级**，此功能需安装 gtp_tools.apk，并将需要升级的 bin 文件拷入 sdcard 或者其他用户权限的目录，运行 apk 软件后，进入第四功能项“Firmware Update”，并确保“Firmware Update”已经勾选，通过“update file”一栏浏览到需要升

级的 bin 文件后，点击“update”按钮进行升级。

7. **配置信息版本号：**GT9XX 配置信息的第一个数据为配置信息版本号，只有发送的配置信息的版本号大于或等于芯片中保存的配置版本号时，发送的配置信息才会被 GT9XX 接受并生效，如果调试过程中发现配置信息发不下去，请首先读出芯片中的配置信息版本号，看是否满足要求。

8. **宏开关定义：**驱动中在 ON/OFF define 部分定义了一些宏开关，以便在调试的过程中使用，0 表示关闭该功能，1 表示打开功能，各开关的释义如下：

```
✧ #define GTP_DEBUG_ON           //调试信息开关，打开则输出调试信息
✧ #define GTP_DEBUG_ARRAY_ON     //调试数组开关，用于调试时打印一片内存的内容
✧ #define GTP_DEBUG_FUNC_ON      //调试函数开关，用于跟踪函数调用流程
✧ #define GTP_CUSTOM_CFG         //客户定制配置开关，用于客户自行修改某些参数
✧ #define GTP_DRIVER_SEND_CFG    //发送配置开关，仅固化配置了的 IC 可以关闭此功能，
    否则必须打开。
✧ #define GTP_HAVE_TOUCH_KEY     //触摸按键开关，仅带有触摸按键的 TP 需要打开
✧ #define GTP_POWER_CTRL_SLEEP   //如果用电源或者 RESET 控制 SLEEP，请设置为 1，否
    则请设置为 0
✧ #define GTP_AUTO_UPDATE        //固件升级开关
✧ #define GTP_ESD_PROTECT        //ESD 防护机制开关
✧ #define GTP_IC3_SLOT_REPORT    //android4.0 配置成 slot 方式报点
```

四、版本修订记录

文档版本	修订	日期
V1.0	初次建立	2012-08-31