

GT800_801 驱动移植说明书

一、驱动基本信息

支持芯片型号	GT800、GT801
I2C 设备地址(7 位)	0x55
I2C 寄存器地址	8 位
支持 APK 工具?	部分支持
支持自动升级?	不支持
固化配置	不支持
支持 Sensor ID 数	3 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. **gt800_801.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. **gt800_801.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. **goodix_tool.c(Recommend)**: 驱动中用于支持 *goodix_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+=gt800_801.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, 0x55 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", 0x55),},
};
```

4. **修改参考代码**: 一般情况下, 移植过程中只需修改 `gt800_801.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, 重点注意 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 {\
    (u8)GTP_REG_CONFIG_DATA,\
    0x0F,0x05,0x05,0x28,0x02,0x14,0x14,0x10,0x28,0xB2,0x02,0x58,0x04,\
    0x00,0x01,0x23,0x45,0x67,0x89,0xAB,0xCD,0xE0,0x00,0x00,0x34,0x2E,\
    0x4D,0xC4,0x20,0x00,0x01,0x80,0x00,0x00,0x00,0x28,0x00,0x32,0x2C,\
    0x01,0xEC,0x00,0x3C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,\
    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： 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“\”，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      10
#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。

4. 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        //开机监测版本号自动升级开关
```

四、版本修订记录

文档版本	修订	日期
V1.0	初次建立	2012-05-01