GT1X Driver Porting Guide

for Android

Rev. 02——July. 10, 2015

====== Disclaimer ======

The information concerning the device in this publication is intended for you only and is subject to change without prior notice. It is your responsibility to ensure its application complies with technical specifications. Shenzhen Huiding Technology Co., Ltd. (hereafter referred to as “GOODIX”) makes no representation or guarantee for this information, express or implied, written or verbal, statutory or otherwise including but not limited to representation or guarantee for its application, quality, performance, merchantability or fitness for a particular purpose. GOODIX shall assume no responsibility for this information and relevant consequences arising out of the use of such information. Without written consent of GOODIX, it is prohibited to use GOODIX products as critical components in any life support system. This document conveys no licenses, implicitly or otherwise, to any intellectual property rights belonging to GOODIX.

Address: Floor 13, Phase B, Tengfei Industrial Building, Futian Free Trade Zone, Shenzhen, China
Tel / Fax：+86-755-33338828 Email: info@goodix.com

Contents

Address: Floor 13, Phase B, Tengfei Industrial Building, Futian Free Trade Zone, Shenzhen, China
Tel / Fax：+86-755-33338828 Email: info@goodix.com

1. Basic Information about the Driver

Chip Models Supported	GT1151,GT1152,GT9286,GT1143,GT1133,GT5663,GT5668,GT5688
I^2C Device Addresses	0x5d, 0x14(7 bits)
I^2C Register Address	16 bits
Debug Tools	Support
Firmware Upgrade	Support
HotKnot™ feature	Support
Gesture wakeup feature	Support
Stylus feature	Support(Both active and passive)
Platforms supported	Samsung,Qualcomm,TI,etc
Quantity of touch panel supported(Sensor IDs)	6

2. Description on Driver Files

Normally, the folder *reference drivers* in the driver reference pack contain the following files whose functions and directions are described below:

2.1 gt1x.c (Required)

File that implements the primary functions of the driver, such as driver mounting, interrupt response and suspend resume etc.

2.2 gt1x.h (Required)

Platform-dependent header file defines some macros and constants, which being dependent on the host chip platforms.

2.3 gt1x_generic.c(Required)

File that implements the platform-independent functions of the driver, the main role is to be interacted with the gt1x chips, such as chip initialization, data acquisition, coordinate reporting etc.

2.4 gt1x_generic.h(Required)

Platform-independent header file defines some macros and constants and forwards declarations of external variables and functions being used in gt1x_generic.c.

Address: Floor 13, Phase B, Tengfei Industrial Building, Futian Free Trade Zone, Shenzhen, China
Tel / Fax：+86-755-33338828 Email: info@goodix.com

2.5 gt1x_update.c (Recommended)

File used to support firmware update, which is not required but strongly recommended, enabling the touch IC to update its firmware to the latest version when necessary.

2.6 gt1x_tools.c (Recommended)

File used to support the *gtp_tools.apk* debug tool which is able to perform analysis, debug and test on the TP after the whole machine is assembled. It is strongly recommended that this file be added to the driver, especially when COB (chip on board) is applied. These tools help a lot in TP debugging on a whole machine.

2.7 gt1x_extents.c (Required conditionally)

File used to support the HotKnot™ and Gesture wakeup features, HotKnot™ is a near field (short range) wireless communication technology using touch panel, and the Gesture wakeup feature makes user can use a custom-defined gesture to wake up the touch panel from sleep mode.

2.8 gt1x_firmware.h (Required conditionally)

Header file used to store the preset array of the firmware which is responsible for header file update. The array is initialized by default to null.

3. Porting the Driver Step by Step

3.1 Copy File

Copy all files in folder *reference driver* to the directory *drivers/input/touchscreen/* in the kernel.

3.2 Modify *Makefile*

In Linux kernel directory *drivers/input/touchscreen/,* edit *Makefile* and add the following items to the file [Note: separate the (.o) files with space]：

```
obj-y += gt1x.o gt1x_generic.o gt1x_update.o gt1x_tools.o gt1x_extents.o
```

3.3 Add Device

If you platform support Device Tree, please refer to the documents in driver package directory dtsi/. If not, find the board-level files which initialize the I^2C bus in the kernel. For example,

Address: Floor 13, Phase B, Tengfei Industrial Building, Futian Free Trade Zone, Shenzhen, China
Tel / Fax：+86-755-33338828 Email: info@goodix.com

real6410, the development board of this driver, is located in file *arch/arm/mach-s3c6410/mach-smdk6410.c.* If the driver of the touch panel needs to be mounted to I²C0, please follow the driver adding instructions below. *0x5d* is the I²C slave address of the GT1X touch ICs. For the specific address, please refer to the corresponding datasheet. "Goodix-TS" is the name of the I²C device driver, which has to comply with the *GTP_I²C_NAME* in the driver reference code.

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

3.4 Modify the Reference Code

Normally, only the content in files *gt1x.h* and *gt1x_generic.h* needs to be modified during the porting process. Edit these two header files and do the porting according to the notes.

3.4.1 STEP1 Replace the Configuration Table (REQUIRED)

In the file *gt1x_generic.h*, replace the content in GTP_CFG_GROUP[X] with the configuration of the TP that employed for this project. The configuration is usually located in the files (with the extension of *cfg or *txt) provided by the TP manufacturer. *gtxxx* is the chip model being used; [X] (Sensor ID) indicates the serial number of the TP manufacturer. If there is only one manufacturer, please place the configuration in the array GTP_CFG_GROUP0 and make sure other arrays are null. For the numbering of the TP manufacturers, please refer to the sensor ID configuration in the datasheet.

```
// TODO: define your own default or for Sensor_ID == 0 config here.
#define CTP_CFG_GROUP0 { \
    0x42,0xE0,0x01,0x20,0x03,0x05,0x14,0x01,0x02,0x08,\
    // ...
}
// TODO: define your config for Sensor_ID == 1 here, if needed
#define GTP_CFG_GROUP1 {\
    }
// TODO: define your config for Sensor_ID == 2 here, if needed
#define GTP_CFG_GROUP2 {\
    }
// TODO: define your config for Sensor_ID == 3 here, if needed
#define GTP_CFG_GROUP3 {\
    }
```


Address: Floor 13, Phase B, Tengfei Industrial Building, Futian Free Trade Zone, Shenzhen, China
Tel / Fax：+86-755-33338828 Email: info@goodix.com

```
// TODO: define your config for Sensor_ID == 4 here, if needed
#define GTP_CFG_GROUP4 {\
    }
```

3.4.2 STEP2 Modify the I/O Definition and I/O Operation (REQUIRED)

Change the definitions of GTP_INT_PORT and GTP_RST_PORT to the corresponding pin definitions. In addition, check whether the statements at the back regarding the I/O operation are applicable to your target platform. If not applicable, change the statements to the applicable ones.

```
//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)
```

3.4.3 STEP3 Configure the User-Defined Parameters (OPTIONAL)

If you want to configure the parameters such as resolution, interrupt triggering mechanism and maximum number of fingers supported, please enable the macro GTP_CUSTOM_CFG in ON/OFF define, and change the parameters according to the instructions below:

Address: Floor 13, Phase B, Tengfei Industrial Building, Futian Free Trade Zone, Shenzhen, China
Tel / Fax：+86-755-33338828 Email: info@goodix.com

```
//*********************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
```

3.4.4 STEP4 Configure the Touch Key (OPTIONAL)

If the TP supports touch key, then you need to do the touch key configuration. Enable GTP_HAVE_TOUCH_KEY in ON/OFF define, and then configure the touch keys according to the instructions below. Adjustment can be made on the functions and sequence of the touch keys in GTP_KEY_TAB as needed.

```
//*********************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
```

3.4.5 Firmware and configuration Update

To employ Firmware Update, you need to enable the macro GTP_AUTO_UPDATE.

Firmware Update can be achieved in two ways:

① Update through the .bin file:

The preset .bin file location of GT1X is */data/_goodix_update_.bin* and */sdcard/_goodix_udpate_.bin;*

② Update through the array of the firmware:

Update by using the array firmware gtp_default_FW in *gt1x_firmware.h*. You need to

Address: Floor 13, Phase B, Tengfei Industrial Building, Futian Free Trade Zone, Shenzhen, China
Tel / Fax：+86-755-33338828 Email: info@goodix.com

enable GTP_AUTO_UDPATE and GTP_HEADER_FW_UPDATE.

The configuration updates location to: */data/_goodix_config_.cfg* and */sdcard/_goodix_config_.cfg.* If way ① is adopted, the driver will search for the .cfg files together with the .bin file. Configuration update will be implemented if the .bin file is found; if way ② is employed, the .cfg files will be searched after the firmware update.

3.4.6 Gesture Wakeup

The macro related to gesture wakeup is GTP_GESTURE_WAKEUP. Fixed gesture and custom gesture are available if above macro is enabled and a configuration that supports gesture is required. You can get a gesture SDK and demo app source code from GOODIX, this would reduce your development time.

4. Appendix

4.1 Sensor ID

If the ICs of the same model from Goodix and the TPs from more than one manufacturer are employed in one project, then the Sensor ID should be configured. The host sends the configuration of the corresponding ID to the IC during initialization and thus TPs from different manufacturers can be distinguished. The configuration method of Sensor ID, usually, is to pull up, pull down or float a certain or several I/O pins during making the layout. The configuration method varies among different chips. Please refer to the respective datasheet for detail.

4.2 Firmware and Configuration

Firmware is a set of programs running in the IC. The firmware is developed for and nonvolatile to an IC model. Configuration is an array that initializes the firmware in the early stage of firmware operation. The host sends the configuration parameters to the IC via I^2C bus after power-on. Then the IC is able to operate properly. Configuration is generated according to TP characteristics such as the structure, process and number of channels. Once these characteristics are modified, configuration should be modified as well.

4.3 Coordinate Reporting Based on SLOT

Address: Floor 13, Phase B, Tengfei Industrial Building, Futian Free Trade Zone, Shenzhen, China
Tel / Fax：+86-755-33338828 Email: info@goodix.com

For configuration of the android 4.0 or later, Application layer, SLOT coordinate reporting is required. If the driver still uses the traditional position reporting method, android Application layer may identify the reported coordinates as the relative coordinates. If this occurs, please enable the macro GTP_ICS_SLOT_REPORT and change the reporting method to SLOT. For detail, please refer to relevant materials regarding report event in the Linux input subsystem and InputReader.cpp of the android Application layer.

4.4 ESD Protection Mechanism

Add a thread in the driver in every 2 seconds to check the operating status of the IC. If the operation is abnormal, then reset the IC. This function is mainly used to prevent the TP from being invalid due to strong ESD. You can decide whether to enable this function depending on the EDS testing result. Note: This function can be applied only if the host is able to control the VDD of the CTP chip and reset the CTP chip via RESET pin.

4.5 Macro Definition

gt1x_generic.h in the driver defines some macros that will be used during debugging in ON/OFF define.0 indicates disabled, 1 indicates enabled. The macros are interpreted below：

GTP_INCELL_PANEL	For incell touch ic
GTP_DRIVER_SEND_CFG	Send configuration to gt1x when initialization
GTP_CUSTOM_CFG	Need modify some configuration of gt1x,such as resolution etc
GTP_CHANGE_X2Y	Exchange x y axis coordinate when reporting coordinate
GTP_WARP_X_ON	Invert x axis coordinate when reporting coordinate
GTP_WARP_Y_ON	Invert y axis coordinate when reporting coordinate
GTP_GESTURE_WAKEUP	Enable gesture wakeup feature
GES_BUFFER_ADDR	Set gesture buffer address
GTP_HOTKNOT	Enable HotKnot™ feature
HOTKNOT_TYPE	HotKnot firmware stored in flash(0) or sending from driver(1)
HOTKNOT_BLOCK_RW	HotKnot read data from gt1x use polling(0) or interrupt(1) mode
GTP_PROXIMITY	Enable proximity feature to instead of an infrared p-sensor
GTP_HAVE_TOUCH_KEY	Enable touch key feature

Address: Floor 13, Phase B, Tengfei Industrial Building, Futian Free Trade Zone, Shenzhen, China
Tel / Fax：+86-755-33338828 Email: info@goodix.com

GTP_WITH_STYLUS	Enable active/passive stylus feature
GTP_HAVE_STYLUS_KEY	Enable stylus key feature, depends on GTP_WITH_STYLUS
GTP_AUTO_UPDATE	Enable firmware upgrade feature
GTP_HEADER_FW_UPDATE	Update from firmware stored in the header file gt1x_firmware.h, depends on GTP_AUTO_UPDATE
GTP_POWER_CTRL_SLEEP	Power on/off when enter suspend/resume
GTP_ICS_SLOT_REPORT	Use protocol B of the input subsystem when reporting coordinate
GTP_CREATE_WR_NODE	Enable debug tools supported feature(such as gtp_tools.apk)
GTP_ESD_PROTECT	Enable esd protection mechanism, create a monitor thread to check some esd errors at regular time intervals(default as 2s)
GTP_CHARGER_SWITCH	Enable charger switching feature to improve anti-interfere ability when connected with charger
GTP_DEBUG_ON	Enable debug log output from the GTP_DEBUG macro
GTP_DEBUG_ARRAY_ON	Enable debug log output from the GTP_DEBUG_ARRAY macro
GTP_DEBUG_FUNC_ON	Enable debug log output from the GTP_DEBUG_FUNC macro
GTP_SMART_COVER	Enable smart cover feature

5. Revision History

Revision	Description	Date
Rev.00	Preliminary Release	2014-09-28
Rev.01	Gt1x Version 1.2	2015-04-20
Rev.02	Gt1x Version 1.4	2015-07-10

Address: Floor 13, Phase B, Tengfei Industrial Building, Futian Free Trade Zone, Shenzhen, China
Tel / Fax：+86-755-33338828 Email: info@goodix.com