GT9XX Driver Porting Guide

for Android

Rev. 00——Sep. 24, 2014

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	GT911 GT9110 GT9110P GT913 GT915 GT918 GT927 GT928 GT960 GT968 GT910 GT960F GT950 GT968F GT9158 GT967 GT9150 GT963 GT9271 GT917D
I^2C Device Addresses (7 bits)	0x5d, 0x14
I^2C Register Address	16 bits
APK Tools /ADB Tools	Support
Auto-Update	Header file (that contains the firmware); the .bin file.
Quantity of Sensor IDs Supported	6

2. Description on Driver Files

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

2.1 gt9xx.c (Required)

File that implements the primary functions of the driver, such as mounting, position reporting and suspend resume.

2.2 gt9xx.h (Required)

Header file of the driver, which defines some macros and constants and forwards declarations of external variables and functions.

2.3 gt9xx_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.

2.4 gt9xx_firmware.h (Recommended)

Header file used to store the preseted array of the firmware which is responsible for the header file update. The array is initialized by default to null. If the GT9XXF-compatible mode needs to be enabled（Set `GTP_COMPATIBLE_MODE` to 1），you must replace the *gt9xx_firmware.h* in the driver with file of the same name in folder `GT9XXF` beneath GT9XXF Firmware Headers.

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 goodix_tool.c (Recommended)

File used to support the *gtp_tools.apk* tool and ADB tool which are 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.

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

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

3.3 Add Device

Find the board-level files which initialize the I^{2}C bus in the kernel. For example, 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 I^{2}C0, please follow the driver adding instructions below. *0x5d* is the I^{2}C slave address of the GT9XX touch ICs. For the specific address, please refer to the corresponding datasheet. "Goodix-TS" is the name of the I^{2}C device driver, which has to comply with the *GTP_I^{2}C_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 file *gt9xx.h* needs to be modified during the porting process. Open the header file *gt9xx.h* and do the porting according to the notes. Special attention should be paid to the modification of `TODO part.`

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

3.4.1 STEP1 Replace the Configuration Table (REQUIRED)

Replace the content in CTP_CFG_GROUP with the configuration of the TP that you employed for this project. The configuration is usually located in the files (with the extension of *cfg or *txt) provided by the TP manufacturer.

```
// TODO: define your own default or for Sensor_ID == 0 config here.
#define CTP_CFG_GROUP1 { \
    0x42,0xE0,0x01,0x20,0x03,0x05,0x14,0x01,0x02,0x08,\
    // ...
}
// TODO: define your config for Sensor_ID == 1 here, if needed
#define CTP_CFG_GROUP2 {\
    }
// TODO: define your config for Sensor_ID == 2 here, if needed
#define CTP_CFG_GROUP3 {\
    }
// TODO: define your config for Sensor_ID == 3 here, if needed
#define CTP_CFG_GROUP4 {\
    }
// TODO: define your config for Sensor_ID == 4 here, if needed
#define CTP_CFG_GROUP5 {\
    }
// TODO: define your config for Sensor_ID == 5 here, if needed
#define CTP_CFG_GROUP6 {\
    }
```

Cautions：

（1）If the Sensor ID is not configured（refer to Section 4 Appendix for details）, please make sure the configuration is placed in the macro CTP_CFG_GROUP1 and the other groups are null. After the replacement, a back slash "\"should be added at the end of each line;

（2）If the write register *GTP_REG_CONFIG_DATA* is configured in the first line of the configuration macro, then start the configuration replacement from the second line.

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.

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

```
//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)
```

Caution: The INT and RST pins should be initialized to input floating. (floating: neither pull up nor pull down)

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:

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

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

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 STEP5 Add *#include* File (OPTIONAL)

Add the #include file that is required by your target platform at the beginning of the header file. This step is optional.

3.4.6 GT9XXF Compatibility

GT9XXF series currently consists of GT910，GT912，GT950，GT960F and GT968F. Some modifications have been made to the driver of GT9XX so as to be compatible with GT9XXF. The corresponding macro is GTP_COMPATIBLE_MODE. If this macro needs to be enabled, please replace the *gt9xx_firmware.h* in folder GT9XXF beneath GT9XXF Firmware Headers with the gt9xx_firmware.h in the driver. Besides, make sure the GTP_DRIVER_SEND_CFG and GTP_ESD_PROTECT are enabled.

Recommended settings：

```
#define  GTP_COMPATIBLE_MODE  1
#define  GTP_DRIVER_SEND_CFG  1
#define  GTP_ESD_PROTECT        1
```

If screen-off is achieved by power down with small system (designed to obtain faster response) employed, the following macros need to be enabled：

```
#define  GTP_COMPATIBLE_MODE  1
#define  GTP_DRIVER_SEND_CFG  1
#define  GTP_ESD_PROTECT        1
#define  GTP_POWER_CTRL_SLEEP 1
#define  GTP_FL_LITTLE_SYSTEM 1
```


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

3.4.7 Auto-Update

To employ Auto-Update, you need to enable the macro GTP_AUTO_UDPATE.

Auto-Update can be achieved in two ways:

① Update through the .bin file:

The preseted file location of GT9XX is */data/_goodix_update_.bin* and */sdcard/_goodix_udpate_.bin;*

That of GT9XXF is */data/_fl_update_.bin* and */sdcard/_fl_update_.bin*。

② Update through the array of the firmware:

Update by using the array of the firmware *gtp_default_FW* in *gt9xx_firmware.h*. You need to enable GTP_AUTO_UDPATE and GTP_HEADER_FW_UPDATE. GT9XXF does not support update in this way.

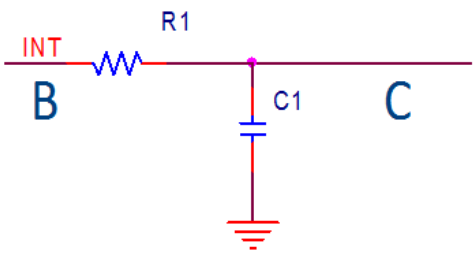
For ICs do not belong to GT9XXF series，if Auto-Update function is needed, you should enable the macro GTP_AUTO_UPDATE_CFG simultaneously.

Preset the configuration Auto-Update 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.

3.4.8 Gesture/Slide Wakeup

The macro related to gesture/slide wakeup is GTP_GESTURE_WAKEUP. To provide this function, the following circuit should be added (for detailed schematic, please refer to Datasheet):

Connect the RC circuit to the INT pin in serial. R: 680Ω, C: 680p. See diagram below:



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

Connect B to the INT pin of GT9XX, C to the INT pin of the host; pull-up resistor cannot be connected to the INT pin of the host.

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 IC 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 Configuration Version Number and Stationary Configuration

The first byte in the configuration of GT9XX is the version number. Only when the version number of the configuration being sent is later or equals to that stored in the chip, can the configuration be accepted by GT9XX and be valid. If the configuration being sent is rejected by the IC when debugging, please check the configuration version number in the chip first and see if it meets the requirement mentioned above. Configure the version number of the IC configuration to 0x5A(90) or greater, and then the host will not send the configuration in the driver. By doing so, the configuration can be fixed. Otherwise, the driver will reset the version number to 0x41(65).

4.4 Position 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 Application layer, SLOT position 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.

4.5 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.

4.6 Macro Definition

gt9xx.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：

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

```
✧#define GTP_DEBUG_ON          // Macro used to print log; outputs log when
                                 enabled.
✧#define GTP_DEBUG_ARRAY_ON    //Macro used to print array
✧#define GTP_DEBUG_FUNC_ON     // Macro used to track the
                                 function invoking process
✧#define GTP_CUSTOM_CFG        // Macro used to support User-defined
                                 configuration; enable when user wants to
                                 modify certain parameters
✧#define GTP_HAVE_TOUCH_KEY    // Macro used to support touch key;
                                 enable when the TP supports touch key
✧#define GTP_AUTO_UPDATE       // Firmware Update by using .bin file
✧#define GTP_HEADER_FW_UPDATE  // Firmware Update by using firmware in
                                   gt9xx_firmware.h (needs to enable
                                   GTP_AUTO_UPDATE)
✧#define GTP_AUTO_UPDATE_CFG   // Auto-Update by searching for .cfg file
                                 (needs to enable GTP_AUTO_UPDATE)
✧#define GTP_ESD_PROTECT       // ESD protection mechanism on/off
✧#define GTP_ICS_SLOT_REPORT   // android 4.0 or later, position reporting
                                   based on slot protocol
✧#define GTP_GESTURE_WAKEUP     // Gesture wakeup
✧#define GTP_COMPATIBLE_MODE    // Compatible with GT9XXF
✧#define GTP_LITTLE_SYSTEM      //GT9XXF with small system, need to enable
                                   POWER_CTRL_SLEEP simultaneously
✧#define GTP_WITH_PEN           // Supports stylus
✧#define GTP_PEN_HAVE_BUTTON    // Supports Active stylus with button
```

5. Revision History

Revision	Description	Date
Rev.00	Preliminary Release	2014-09-23

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