GT1X Driver Porting Guide

for MTK platform

(Based on android)

Rev. 03——July. 29, 2016

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

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	MTK Platforms with Kernel3.18
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_tpd.c (Required)

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

2.2 gt1x_tpd_custom.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.

2.9 gt1x_config.h (Recommended)

Header file used to store the preset array of the config which is responsible for header file update. Putting the config info corresponded to your TP

2.10 Kconfig (Required)

Platform-independent source file defines some macros .

3. Porting the Driver Step by Step (Example: MT6797)

3.1 Copy Files

Copy all files from the directory *reference driver* to the directory */kernel-3.18/drivers/input/touchscreen/mediatek/GT1X /*. If the folder *GT1X* does not exist, please create one ,here is *GT1X.*

3.2 Modify *Makefile*

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

In directory *mediatek/[your_project_name]* (Note:[your_project_name] is the name of the project that you are compiling), edit file *Makefile* , add an item and modify according to the instructions below (Note: GT1X must comply with the name of the folder that you used to store the driver source code of the TP):

```
obj-$(CONFIG_TOUCHSCREEN_MTK_GT5668)    +=   GT1X/
```

3.3 Modify *Kconfig*

The macros in gt1x_generic.h (such as GTP_DRIVER_SEND_CFG and so on) are moved in an Kconfig which is put in the driver path.The macros were deleted.If you want to open or close a marco in version1.6,you should make a configuration in the kernel.There are two steps for config the Kconfig.

Step one:In directory *mediatek/[your_project_name]* (Note:[your_project_name] is the name of the project that you are compiling), edit file *Kconfig* .Add an item as below,for example,we want add a GT5668,so we add a item name *TOUCHSCREEN_MTK_GT5668(you can change other names by your project)*,and the Kconfig path is ***drivers/input/touchscreen/mediatek/GT1x/Kconfig***

```
config TOUCHSCREEN_MTK_GT5668
    bool "GT5668 for Mediatek package"
    default n
    help
      Say Y here if you have GT5668 touch panel.
      If unsure, say N.
      To compile this dirver as a module, choose M here: the
      module will be called.
source "drivers/input/touchscreen/mediatek/GT1x/Kconfig"
```

Step two: Execute *make menuconfig* to find your configuration, and press Y to select your project item.Press spacekey in your keyboard to open or close a macros. You will see the macros for the drivers as below

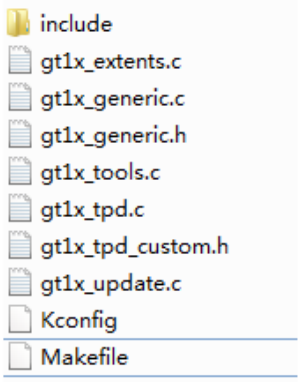
Address: Floor 13, Phase B, Tengfei Industrial Building, Futian Free Trade Zone, Shenzhen, China
Tel / Fax：+86-755-33338828 Email: info@goodix.com

```
[*]   GT5668 for Mediatek package
(firmware1) GTP_FIRMWARE_GOODIX (NEW)
(config1) GTP_CONFIG_GOODIX (NEW)
[ ]     GTP_INCELL_PANEL (NEW)
[*]     GTP_DRIVER_SEND_CFG
[*]     GTP_CUSTOM_CFG
[ ]     GTP_CHANGE_X2Y
[ ]     GTP_WARP_X_ON
[ ]     GTP_WARP_Y_ON
[*]     GTP_GESTURE_WAKEUP
[*]     GTP_HOTKNOT
[ ]     HOTKNOT_TYPE (NEW)
[*]     HOTKNOT_BLOCK_RW
[ ]     GTP_FW_UPDATE_VERIFY (NEW)
[*]     GTP_HAVE_TOUCH_KEY
[ ]     TPD_HAVE_BUTTON (NEW)
[ ]     GTP_PROXIMITY
```

3.4 Modify the custom config and firmware data

In version1.6,the custom config data and firmware data are put in independent files,which are configured in the kernel Kconfig .When compile the kernel codes,the path of configuration table and firmware table should be put in the drivers *Makefile*.

```
(firmware1) GTP_FIRMWARE_GOODIX (NEW)
(config1) GTP_CONFIG_GOODIX (NEW)
[ ]     GTP_INCELL_PANEL (NEW)
[*]     GTP_DRIVER_SEND_CFG
[*]     GTP_CUSTOM_CFG
[ ]     GTP_CHANGE_X2Y
[ ]     GTP_WARP_X_ON
[ ]     GTP_WARP_Y_ON
[*]     GTP_GESTURE_WAKEUP
[*]     GTP_HOTKNOT
[ ]     HOTKNOT_TYPE (NEW)
[*]     HOTKNOT_BLOCK_RW
[ ]     GTP_FW_UPDATE_VERIFY (NEW)
[*]     GTP_HAVE_TOUCH_KEY
[ ]     TPD_HAVE_BUTTON (NEW)
[ ]     GTP_PROXIMITY
[ ]     GTP_WITH_STYLUS (NEW)
[ ]     GTP_HAVE_STYLUS_KEY (NEW)
[*]     GTP_AUTO_UPDATE
[ ]     GTP_HEADER_FW_UPDATE
[*]     GTP_REQUEST_FW_UPDATE
[*]     GTP_CREATE_WR_NODE
[*]     GTP_ESD_PROTECT
[ ]     GTP_CHARGER_SWITCH
[*]     GTP_DEBUG_ARRAY_ON
[ ]     GTP_DEBUG_FUNC_ON
[ ]     GTP_POWER_CTRL_SLEEP
[ ]     GTP_ICS_SLOT_REPORT (NEW)
```




```
ccflags-y += -I$(srctree)/drivers/input/touchscreen/mediatek/GT1x/include/$(CONFIG_GTP_FIRMWARE_GOODIX)/
ccflags-y += -I$(srctree)/drivers/input/touchscreen/mediatek/GT1x/include/$(CONFIG_GTP_CONFIG_GOODIX)/
```

3.4.1Modify the ***Makefile in*** the *GT1X*

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

If you enable the ***GTP_PROXIMITY*** ,you should add the code in the Makefile:

```
ccflags-y += -I$(srctree)/drivers/misc/mediatek/hwmon/include
```

Except that you should open the ***CONFIG_MTK_SENSOR_SUPPORT=y*** in the kernel configuration.

The path of MTK_SENSOR_SUPPORT is as below:

```
CONFIG_MTK_SENSOR_SUPPORT:

sensor config to sensor port sensor feature in project.

Symbol: MTK_SENSOR_SUPPORT [=y]
Type  : boolean
Prompt: MTK_SENSOR_SUPPORT
  Location:
    -> Device Drivers
      -> Misc devices
        -> MediaTek Properitary Configuration (MEDIATEK_SOLUTION [=y])
  Defined at drivers/misc/mediatek/Kconfig:129
  Depends on: MEDIATEK_SOLUTION [=y]
```

3.4.2.Replace the Configuration Table (REQUIRED)

In the file *gt1x_generic.h*, replace the content in CTP_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 GTP_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 {\
    }
// TODO: define your config for Sensor_ID == 4 here, if needed
#define GTP_CFG_GROUP4 {\
    }
```


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 == 5 here, if needed
#define GTP_CFG_GROUP5 {\
    }
```

3.5 Modify the Reference Code

Normally, only the content in files *gt1x_tpd_custom.h* and *gt1x_generic.h* need to be modified during the porting process. Edit the header files and do the porting modification as follows:

3.5.1 STEP1 Specify I2C bus number (REQUIRED)

The i2c bus number is defined by macro TPD_I2C_NUMBER in *gt1x_tpd_custom.h*. Modify macro TPD_I2C_NUMBER with number of the i2c bus that the touch IC connected in hardware.

DMA (Direct Memory Access) enables the I^2C bus to transfer data in a faster rate and transfer more bytes at a time. The macro that supports the I^2C DMA is GTP_SUPPORT_I2C_DMA. The maximum number of bytes to transfer at a time is configured to 255.The macro is GTP_DMA_MAX_TRANSACTION_LENGTH. This macro is disabled by default. If the platform supports I^2C DMA, it is recommended that the macro GTP_SUPPORT_I2C_DMA be enabled. The table below shows whether the i2c bus supported DMA mode in parts of MTK platforms. Please decide whether to enable this macro according to the table below.

MTK platforms	DMA Support
MT6575	×
MT6577	×
MT6572	√
MT6589	√
MT6582	√
MT6592	√
MT6595	√
Other	Check the datasheet

```
#define TPD_I2C_NUMBER          0
#define TPD_SUPPORT_I2C_DMA     1
```

3.5.2 STEP2 Define the I/O Pins and I/O Operation (REQUIRED)

In the file *gt1x_tpd_custom.h*, modify the IO define according to the hardware setting.

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

```
#if defined(CONFIG_MTK_LEGACY)
#define TPD_POWER_SOURCE_CUSTOM MT6328_POWER_LDO_VGP1
#endif
#define GTP_GPIO_AS_INT(pin) tpd_gpio_as_int(pin)
#define GTP_GPIO_OUTPUT(pin, level) tpd_gpio_output(pin, level)
```

3.5.3 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 open the the macro GTP_CUSTOM_CFG in the file *kernel Kconfig*, and change the parameters according to the instructions below:

```
#ifdef   CONFIG_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
```

3.5.4 STEP5 Configure the Touch key (OPTIONAL)

If the TP supports touch key, then you need to configure the touch key. The configuration of the touch key is controlled by two macros: TPD_HAVE_BUTTON in the file gt1x_tpd_custom.h and GTP_HAVE_TOUCH_KEY in the file *Kconfig file*. The difference between the two macros is that, the former indicates that the touch key will be converted into the extended coordinate and then reported to the host with vibration feedback while the latter indicates the touch key will be reported in the form of key value and without vibration feedback. If vibration feedback is needed, modifications should be made to the android Application layer.

```
#ifdef CONFIG_GTP_HAVE_TOUCH_KEY
#define GTP_KEY_TAB
{KEY_BACK, KEY_HOMEPAGE, KEY_MENU, KEY_SEARCH}
#define GTP_MAX_KEY_NUM   4
#endif
```


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

If the macro TPD_HAVE_BUTTON is enabled, the central coordinate of the key should be configured from mtk_tpd.c which are parsed from the device tree.

```
&touch {
        tpd-resolution = <1080 1920>;
        use-tpd-button = <1>;
        tpd-key-num = <3>;
        tpd-key-local= <139 172 158 0>;
        tpd-key-dim-local = <90 883 100 40 230 883 100 40 370 883 100 40 0 0 0 0>;
        tpd-max-touch-num = <5>;
        tpd-filter-enable = <1>;
        tpd-filter-pixel-density = <192>;
        tpd-filter-custom-prameters = <0 0 0 0 0 0 0 0 0 0 0 0>;
        tpd-filter-custom-speed = <0 0 0>;
        pinctrl-names = "default", "state_eint_as_int", "state_eint_output0", "state_eint_output1",
                "state_rst_output0", "state_rst_output1";
        pinctrl-0 = <&ctp_pins_default>;
        pinctrl-1 = <&ctp_pins_eint_as_int>;
        pinctrl-2 = <&ctp_pins_eint_output0>;
        pinctrl-3 = <&ctp_pins_eint_output1>;
        pinctrl-4 = <&ctp_pins_rst_output0>;
        pinctrl-5 = <&ctp_pins_rst_output1>;
        status = "okay";
};
```

3.5.5 Firmware and configuration Update

To employ Firmware Update, you need to enable the macro GTP_AUTO_UPDATE. Firmware Update can be achieved in thress 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 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)Update through the .bin file,with enable the GTP_REQUEST_FW_UPDATE macro.It use the *request_firmware()* to apply the data of firmware.If GTP_REQUEST_FW_UPDATE is enable ,you should put an bin file in the /etc/firmware/.The bin file must be the same as names which is named in the gt1x_update.c.Here the bin file name is *gt1x_fw.BIN*,and you can change to another,but it must be the same as in the phone.

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

```
#ifdef CONFIG_GTP_REQUEST_FW_UPDATE
#include <linux/firmware.h>
#define GT1X_FW_NAME          "gt1x_fw.BIN"
#endif
```

4)Request Firmware from Userspace Patch

```
### Kernel support
Check kernel's configuration and make sure it is properly configured to support this feature.
1. Excute the following command in the root directory of kernel:
```
$ make menuconfig
```
2. Make sure the item `<*>Userspace firmware loading support` is selected.
```
    Device Drivers -->
        Generic Driver Options -->
            <*> Userspace firmware loading support
```
### Driver support
### GT9p firmware
Rename the GT9p firmware to `gt1x_fw.bin`, and put it into one of the following directories in the device:
```
/etc/firmware/
```
```

3.5.6 Gesture Wakeup

The macro related to gesture wakeup is GTP_GESTURE_WAKEUP. Fixed gesture and custom

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

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 ESD Protection Mechanism

Add a thread to the driver. The thread will check the operating status of the IC in every 2 seconds. 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.*

4.4 I^2C Transmission Rate

The transmission rate can be configured via the macro I^2C _MASTER_CLOCK. To optimize the user experience, it is recommended that the I^2C transmission rate be greater than 300 kHz.

4.5 Macro Definition

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

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
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

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

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
Rev.03	GT1x Version 1.6	2016-07-29

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