ILITEK
I Love Innovation

Linux I2C
Touch Device Driver

Version: V0_0_1_4
Document: ILITEK_LINUX_I2C_DRIVER.pdf

ILI TECHNOLOGY CORP.
8F, No.1, Taiyuan 2nd-St., Jhubei City, Hsinchu County 302, Taiwan, R.O.C.
Tel.886-3-5600099; Fax.886-3-5600055
http://www.ilitek.com

目錄

Revision History

Version No.	Date	Page	Description
0.0.1	2011.3.7	All	Firstly release
0.0.2	2011.5.12	3	Modified driver file name.
0.0.3	2011.9.30	3	Modified version id
0.0.4	2012.11.26	3	Method of adding idc files
0.0.5	2017.7.14	15	Modified driver structure
0.0.0.6	2017.9.12	13	1. Support the dts on MTK platform. 2. For the medium and large inch add debug message command. 3. Check the hex file before upgrading the FW. 4. Add the mach function for Intel platform
0.0.0.7	2019.5.7	14	1. Remove ILI2120 function 2. Add glove mode control.
0.0.0.8	2019.8.22	19	1. Remove glove mode control. 2. Add switch modes function
0.0.1.0	2020.10.15	20	1. Added the chrome os define 2. Added the support for the Lego series IC description 3. Added V6 upgrade 4. Added the setting of low power 5. Added the length setting of raw data 6. Added the length setting for firmware upgrade.
0.0.1.2	2022.5.6	All	1. Sync to template of Chinese version 2. Update description of FW upgrade on boot 3. Update description of gesture wakeup
0.0.1.3	2022.11.23	All	1. Update description of gesture wakeup 2. Update description of FW upgrade/OTA 3. Update description of ESD protection 4. Update description of Low Power Mode 5. Update description of Firmware Mode
0.0.1.4	2023.6.14	5,6,7	1. Stop using ILITEK_KERNEL_DRIVER flag 2. Add macro notes

1. Introduction

A. Brief Intro.

This document introduces how ILITEK_LINUX_I2C_DRIVER works on different platforms, the porting guidance, and some configuration description. This driver integrates the supports on several platforms such as Qualcomm, MTK, Rockchip, which are all defined in ilitek_ts.h. Users can choose one of them by replacing the macro's name depending on their platform as below.

```
00065: #define ILITEK_PLAT_QCOM                    1
00066: #define ILITEK_PLAT_MTK                     2
00067: #define ILITEK_PLAT_ROCKCHIP                3
00068: #define ILITEK_PLAT_ALLWIN                  4
00069: #define ILITEK_PLAT_AMLOGIC                 5
00070:
00071: #define ILITEK_PLAT                         ILITEK_PLAT_QCOM
00072:
```

Supported TP IC	**Protocol V3 ICs:** 2312、2315、2510、2511、2712 **Protocol V6 ICs:** 2520、2521、2322、2323、2316、2326、2130、2131、2132、2900、2901、2910、2911、2531、2532
I2C slave address（7 bits）	0x41
Boot firmware upgrade (OTA)	Supported
Supported platforms	Qcom **(default)**、Rockchip、MTK、Allwinner、Amlogic

B. File Description

- ilitek_ts.h:
  Most of useful macros/settings for some functions or features are included.
- ilitek_platform_init.c:
  It includes board info., driver and platform-related initialization flow.
- ilitek_main.c:
  Implementation of probe main functions, ISR handling, event report, suspend and resume handling, etc.
- ilitek_update.c:
  Implementation of the firmware update flow.
- ilitek_tool.c:
  Implementations of file nodes handling under procfs/ sysfs and device driver registration.
- ilitek_protocol.c:
  Includes the TP IC command protocol handling.
- ilitek_report.c:
  touch/stylus report data handling/decoding flow
- ilitek_crypto.c:
  data encrypt/decrypt related flow

C. Porting Guide

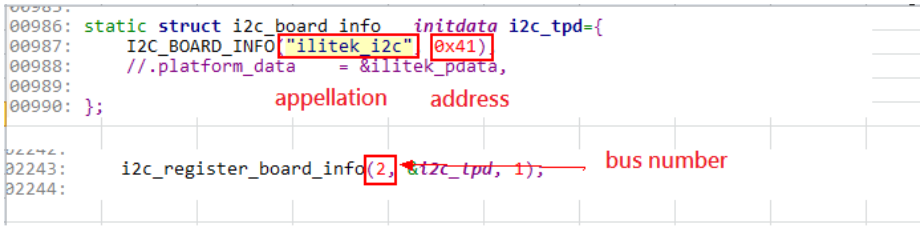
1. Copy the directory of driver source code into /drivers/input/touchscreen/. It might be different based on each platform setting. Normally we put driver into above path.
2. Modify /drivers/input/touchscreen/Makefile, and add below string.

   obj-y += <ILITEK driver directory name>/

3. Setup I2C device to your platform, please find below different ways to setup.

   ✓ **Use Board configuration:**

   The borad file is usually placed as *"linux/arch/arm/mach-xxx/board-xxx.c".* The figure below shows how we did configuration with i2c bus in kernel on customer's platform.



   ✓ **Use Device tree (DTS):**

   DTS file which normally located at *"linux/arch/arm/boot/dts/xxx-xxx.dts".* Users must know which I2C bus TP IC is connected to. Also, INT/Reset gpio nodes should be ready, then add below setting under specific I2C bus node.

```
ilitek@41 {
        compatible = "tchip,ilitek";
        reg = <0x41>;
        interrupt-parent = <&msm_gpio>;
        interrupts = <13 0x0 >;
        vdd-supply = <&pm8916_l17>;
        vcc_i2c-supply = <&pm8916_l6>;
        ilitek,irq-gpio = <&msm_gpio 13 0x0>;
        ilitek,reset-gpio = <&msm_gpio 12 0x0>;
        ilitek,vbus = "vcc_i2c";
        ilitek,vdd = "vdd";
        ilitek,name = "ilitek_i2c";
};
```

2. Description of Macros/Settings

✓ **Driver Version Information**

#define DRIVER_VERSION_0	5
#define DRIVER_VERSION_1	9
#define DRIVER_VERSION_2	2
#define DRIVER_VERSION_3	0
#define CUSTOMER_H_ID	0
#define CUSTOMER_L_ID	0
#define TEST_VERSION	0

✓ **Platform Setting:**

The below macros are confirmed to the driver which of initialization functions should be executed. Set ILITEK_PLAT_QCOM as default.

#define ILITEK_PLAT_QCOM	1
#define ILITEK_PLAT_MTK	2
#define ILITEK_PLAT_ROCKCHIP	3
#define ILITEK_PLAT_ALLWIN	4
#define ILITEK_PLAT_AMLOGIC	5
#define **ILITEK_PLAT**	ILITEK_PLAT_QCOM

✓ #define ILITEK_TOOL

Created for ILITEK tool, **default enabled.**

✓ #define ILITEK_TUNING_MESSAGE

Created for dumping ISR data to userspace tool, **default enabled.**

✓ #define ILITEK_ESD_CHECK_ENABLE 0

Created for enabling ESD protection, **default disabled as 0.** Set to 1 to enable.

(default disabled, please check with ILITEK if you need to enable it)

✓ #define ILITEK_TOUCH_PROTOCOL_B

Report touch event with protocol B, **default disabled.**

✓ #define ILITEK_REPORT_PRESSURE

Report the pressure event, **default disabled.**

- ✓ #define MTK_UNDTS

  Created for MTK platform without using device tree initialization.

- ✓ #define ILITEK_USE_MTK_INPUT_DEV

  For MTK platform only, use tpd->dev to register input device, **default enabled.**

- ✓ #define ILITEK_USE_LCM_RESOLUTION

  Using user-defined LCM resolution, **default disabled**. For MTK platform with ILITEK_USE_MTK_INPUT_DEV, this macro would be enabled. LCM resolution value should be set as below.

  (default disabled, please check with ILITEK if you need to enable it)

  - ✓ #define TOUCH_SCREEN_X_MAX (1080) //LCD_WIDTH
  - ✓ #define TOUCH_SCREEN_Y_MAX (1920) //LCD_HEIGHT

- ✓ #define ILITEK_ROTATE_FLAG 0

  Rotate axis of X and Y for point report, **default disabled.** Enable for non-zero value.

  (default disabled, please check with ILITEK if you need to enable it)

- ✓ #define ILITEK_REVERT_X 0

  Rotate axis of X in opposite way, **default disabled.** Enable for non-zero value.

  (default disabled, please check with ILITEK if you need to enable it)

- ✓ #define ILITEK_REVERT_Y 0

  Rotate axis of Y in opposite way, **default disabled.** Enable for non-zero value.

  (default disabled, please check with ILITEK if you need to enable it)

- ✓ #define ILITEK_ENABLE_REGULATOR_POWER_ON

  Power the TP IC up by regulator. Driver could control its output voltage and switch power on/off in system suspend/resume, **default enabled.**

- ✓ #define ILITEK_GET_GPIO_NUM

  Get reset/irq gpio number from device tree seting, **default enabled.** If it is disabled, user should set ILITEK_RESET_GPIO and ILITEK_IRQ_GPIO accordingly.

✓ #define ILITEK_GESTURE_DEFAULT Gesture_**Disable**

Set gesture wakeup, please replace BOLD text with below options.

```
#define ILITEK_GESTURE_TYPES          \
    X(Disable,      0, "disable")     \   Disable gesture wakeup
    X(Single_Click, 1, "single-click") \  Single touch wakeup
    X(Double_Click, 2, "double-click")    Double touch wakeup
```

Double touch wakeup parameters are listed below.

```
#define DOUBLE_CLICK_DISTANCE                1000
    Distance between first and second touch
#define DOUBLE_CLICK_ONE_CLICK_USED_TIME     800
    Total time of first touch, unit in ms.
#define DOUBLE_CLICK_NO_TOUCH_TIME           1000
    Time interval between first and second touch, unit in ms.
```

✓ #define ILITEK_LOW_POWER_DEFAULT Low_Power_**Sleep**

Power saving mode when system suspend, please replace BOLD text with below options.

(Idle mode is used with gesture wakeup enabled, please check with ILITEK that your firmware do support IDLE mode)

```
#define ILITEK_LOW_POWER_TYPES    \
    X(Sleep,    0, "sleep")       \   Using ILITEK power saving mode
    X(Idle,     1, "idle")        \   Using ILITEK idle mode, need enable gesture wakeup
    X(PowerOff, 2, "poweroff")        Power off TP IC via regulator.
```

3. Description of Some APIs

- ✓ int ilitek_power_on(bool status)

  ILITEK_ENABLE_REGULATOR_POWER_ON should be enabled, and regulator control is implemented in this API.

- ✓ void ilitek_reset(int delay_ms)

  It's the API for TP IC hardware reser. "delay_ms" means delay time after reset (unit in ms), which should according to each TP IC spec.

- ✓ int api_update_ts_info(void *handle)

  It's the API for reading TP IC information, such as firmware version, resolution, module name and firmware mode, etc.

- ✓ int ilitek_request_irq(void)

  It's the API for ISR registeration. Please be aware of INT gpio should be set and acquired correctly.

- ✓ ISR_FUNC(ilitek_i2c_isr)

  It's the main API for interrupt handling, which includes touch point handling and input event report.

- ✓ void ilitek_suspend(void) / void ilitek_resume(void)

  It's the API for system suspend/resume and handle device sleep/wakeup related setting.

- ✓ int ilitek_upgrade_firmware(char *filename)

  It's the main API for firmware upgrade, and .bin/.hex fw file are supported.

4. Function Description

A. Firmware Update on System Boot (OTA)

✓ **Pre-Processing**

Driver's Makefile should includes flag definition as below, then check according to different file format listed below.

ccflags-y += -DILITEK_BOOT_UPDATE

- **.ili file**

  "ilitek_fw.h" and .ili file should be placed under the same path of this driver, those file should be provided by ILITEK or other agents.

- **.bin or .hex file**

  Please make sure kernel config "CONFIG_FW_LOADER" is enabled, and put .bin or .hex file into correct path. Please refer to "/driver/base/firmware_class.c" file and check the setting of "fw_path" variable.

  ※ Refer to "/drivers/base/firmware_loader/main.c" since kernel v4.17.

✓ **FW Upgrade Condition Checking**

If TP IC is in bootloader mode, firmware update would be executed forcely, otherwise, please find below for different situations.

- **.ili file**

  Compare 8 byte firmware version from high byte to low byte. If version of .ili file is higher that current TP IC, fw upgrade would be perform. Please check "decode_ili(…)" API for more detail.

- **.bin or .hex file**

  FW upgrade forcely.

B. Gesture Wake Up

✓ When enable this function, please make sure TP is powered on when system suspend.

✓ **Pre-Processing**

Please set ILITEK_GESTURE_DEFAULT macro accordingly. Currently single/double touch wake up is provided as below.

✓ **User could change gesture wakeup options while system running.**

- Check current mode: cat /sys/bus/i2c/devices/<device name>/gesture

```
rpi4:/data/local/tmp # cat /sys/bus/i2c/devices/1-0041-1/gesture
[disable] single-click double-click
```

- Disable: echo disable > /sys/bus/i2c/devices/<device name>/gesture
- Single touch: echo single-click > /sys/bus/i2c/devices/<device name>/gesture
- Double touch: echo double-click > /sys/bus/i2c/devices/<device name>/gesture

C. Firmware Mode Switch: Please check current firmware is supported with ILITEK and agents.

- List supported modes and show current mode: cat /proc/ilitek/func_mode

```
rpi4:/data/local/tmp # cat /proc/ilitek/func_mode
function mode: [0] 1 2
```

- Switch to firmware mode 0: echo 0 > /proc/ilitek/func_mode
- Switch to firmware mode 1: echo 1 > /proc/ilitek/func_mode
- Switch to firmware mode 2: echo 2 > /proc/ilitek/func_mode

D. ESD Detection

✓ **Description**

It would implement by delayed workqueue, and would check IC's protocol version every time interval. If anomaly happens, hardware reset would be performed. User could modify "ts->esd_delay" for setting time interval.

✓ **Pre-Processing**

- To enable it, ILITEK_ESD_CHECK_ENABLE should be set as 1, which default 0 as disabled. Also, User could enable it manually as below.
- Check current state: cat /sys/bus/i2c/devices/<device name>/esd_check

```
rpi4:/data/local/tmp # cat /sys/bus/i2c/devices/1-0041-1/esd_check
enable [disable]
```

- Enable ESD check: echo enable > /sys/bus/i2c/devices/<device name>/esd_check
- Disable ESD check: echo disable > /sys/bus/i2c/devices/<device name>/esd_check

E. Firmware Upgrade Manually

✓ **Pre-Processing**

- Please check kernel config "CONFIG_FW_LOADER" is enabled.
- Place .hex or .bin fw file to specific path, check "/driver/base/firmware_class.c" and setting of "fw_path" variable.
  (Refer to "/drivers/base/firmware_loader/main.c" since kernel v4.17.)
- User could enable specific path via command (if path "/data/local/tmp" is required)
  echo –n "/data/local/tmp" > /sys/module/firmware_class/parameters/path
- Please name the fw file as ilitek.hex or ilitek.bin, softlink is also feasible as below.
  ln -s /data/local/tmp/<your fw filename.hex> ilitek.hex

```
rpi4:/data/local/tmp # ln -s /data/local/tmp/ILI2901_7.80.0.0_0.0.0.2.hex ilitek.hex
```

✓ **Fimrware upgrade command**

- Via procfs: cat /proc/ilitek/update_fw
- Via sysfs: cat /sys/bus/i2c/devices/<device name>/update_fw
- It would show current firmware version once upgrade sccessfuly as below.

```
rpi4:/data/local/tmp # cat /proc/ilitek/update_fw
upgrade success, fw version: 07-00-00-00-00-00-00-00
```

F. System Suspend/Resume Setting

- ✓ User could modify default mode by setting ILITEK_LOW_POWER_DEFAULT macro. Also, user could change to other mode manually as below.
- ✓ Show current mode: cat /sys/bus/i2c/devices/<device name>/low_power

```
rpi4:/data/local/tmp # cat /sys/bus/i2c/devices/1-0041-1/low_power
[sleep] idle poweroff
```

- ✓ Set sleep mode: echo sleep > /sys/bus/i2c/devices/<device name>/low_power
- ✓ Set idle mode: echo idle > /sys/bus/i2c/devices/<device name>/low_power

(Idle mode should comes w/ gesture wakeup. please check FW is supported w/ ILITEK)

G. Check Firmware Version

- ✓ Command via procfs: cat /proc/ilitek/firmware_version
- ✓ Command via sysfs: cat /sys/bus/i2c/devices/<device name>/firmware_version

```
rpi4:/data/local/tmp # cat /proc/ilitek/firmware_version
fw-version-tag: [07-00-00-00-00-00-00-00]
```

H. Check Module Name

- ✓ Command via sysfs: cat /sys/bus/i2c/devices/<device name>/product_id

```
rpi4:/data/local/tmp # cat /sys/bus/i2c/devices/1-0041-1/product_id
module-name-tag: [ILI2510000000000]
```

I. Console

- ✓ Command format: <write bytes length> <read bytes length> <cmd %x-%x-%x….> , which cmd should be format in %x-%x-%x….
- ✓ It would show writing/reading bytes in kernel log as below.
  - Write only command: echo 3 0 f0-0-0 > /proc/ilitek/console

```
[13927.228790] [ILITEK][MSG][ilitek_console_write:500] [write]: f0-00-00, len: 3
```

  - Write then read command: echo 1 3 42 > /proc/ilitek/console

```
[13938.241406] [ILITEK][MSG][ilitek_console_write:500] [write]: 42, len: 1
[13938.242916] [ILITEK][MSG][ilitek_console_write:506] [read]: 06-00-04, len: 3
```

J. Enable Debug Log

- ✓ Enable debug log: echo dbg_debug > /proc/ilitek_ctrl
- ✓ Disable debug log: echo dbg_info > /proc/ilitek_ctrl
- ✓ Debug log shows while reporting touch event, please find “[ILITEK]” in kernel log.

5. Get Key Info. Flow Guide

✓ Flow to get Key info. is different for V3 and V6 protocol IC listed below. Detailed flow would be introduced with console function.

V3	For IC: ILI2312/ 2315/ 2510/ 2511/ 2712
V6	For IC: ILI2316/ 2326/ 2130/ 2131/ 2132/ 2520/ 2521/ 2322/ 2323/ 2900/ 2901/ 2910/ 2911/ 2531/ 2532

✓ **V3 Protocol IC:**

1. Firstly, check amout of FW key by panel info. cmd: 0x20. Write 1 byte 0x20, then read 15 bytes data. FW key count data is at the **9th byte**.
   - echo 1 15 20 > /proc/ilitek/console

```
[write]: 20, len: 1
[read]: 00-20-00-20-1a-0d-0a-0a-0a-ff-ff-ff-ff-ff-01, len: 15
```

2. Then, read key info. by cmd: 0x22. Write 1 byte 0x22, then read 29 bytes data. **If amount of FW keys larger than 5, it needs to read 25 bytes again.**
   - echo 1 29 22 > /proc/ilitek/console **&& echo 0 25 0 > /proc/ilitek/console**

```
[write]: 22, len: 1
[read]: 01-00-01-00-72-02-bc-20-64-73-05-78-20-64-66-08-34-20-64-74-0a-f0-20-64-02-0d-ac-20-64, len: 29
wlen: 0, rlen: 25, command: 0
[read]: 02-10-68-20-64-02-13-24-20-64-02-15-e0-20-64-02-18-9c-20-64-02-1b-58-20-64, len: 25
```

- The first 29 bytes and additional 25 bytes data format list as below.

First 29 bytes	0	X Length of Key Area	X Length of Key Area **(15:8)**
	1		X Length of Key Area **(7:0)**
	2	Y Length of Key Area	Y Length of Key Area **(15:8)**
	3		Y Length of Key Area **(7:0)**
	4	Key 0	Key ID
	5		X start position **(15:8)**
	6		X start position **(7:0)**
	7		Y start position **(15:8)**
	8		Y start position **(7:0)**
	9~13	Key 1	Format refers to the Key 0
	14~18	Key 2	Format refers to the Key 0
	19~23	Key 3	Format refers to the Key 0
	24~28	Key 4	Format refers to the Key 0
Additional 25 bytes	0~4	Key 5	Format refers to the Key 0
	5~9	Key 6	Format refers to the Key 0
	10~14	Key 7	Format refers to the Key 0
	15~19	Key 8	Format refers to the Key 0
	20~24	Key 9	Format refers to the Key 0

✓ **V6 Protocol IC:**

1. Firstly, check amout of FW key by panel info. cmd: 0x20. Write 1 byte 0x20, then read 15 bytes data. FW key count data is at the **10th byte**.
   - echo 1 15 20 > /proc/ilitek/console

```
wlen: 1, rlen: 15, command: 20
[write]: 20, len: 1
[read]: 00-40-00-40-1a-00-0d-00-0a-04-01-01-00-01-05, len: 15
```

2. Then, read key info. by cmd: 0x22. Write 1 byte 0x22, then read **(5 + 5 * key amount) bytes** data (ex. amount of key is 4, then read 5 + 5 * 4 = 25 bytes data).
   - echo 1 25 22 > /proc/ilitek/console

```
wlen: 1, rlen: 25, command: 22
[write]: 22, len: 1
[read]: 02-00-01-00-01-74-b0-04-64-40-72-60-09-64-40-73-10-0e-64-40-66-c0-12-64-40, len: 25
```

- Data format lists below.

0	Key Mode	1: HW key, 2: HSW key, 3: Virtual key, 其他值: Disable
1	X Length of Key Area	X Length of Key Area **(7:0)**
2		X Length of Key Area **(15:8)**
3	Y Length of Key Area	Y Length of Key Area **(7:0)**
4		Y Length of Key Area **(15:8)**
5	Key 0	Key ID
6		X start position **(7:0)**
7		X start position **(15:8)**
8		Y start position **(7:0)**
9		Y start position **(15:8)**
10~14	Key 1	Format refers to the Key 0
…	…	…
5 + 5 * N	Key N	Format refers to the Key 0

6. Troubleshooting

A. Could not get into driver's probe function

- For using board file for device initialization:

  Please make sure ILITEK_TS_NAME and registered I2C device is the same.

- For using device tree for device initialization:

  Please make sure compatible string in driver and dts node is the same.

- Please make sure i2c device and related file nodes are created by kernel.

  Please check node "<i2c-bus-number>-0041" in "/sys/bus/i2c/devices" as below

```
ls /sys/bus/i2c/devices/
0-001b 0-0041 4-0022 4-0041-1 i2c-1  i2c-3 i2c-9
0-0040 1-001c 4-0041 i2c-0    i2c-10 i2c-4
```

B. I2C communication anomaly or TP IC information anomaly

1. Please check I2C bus number and device address are correct.
2. Then check TP IC power voltage and I2C bus voltage level are correct
3. User could monitor TP IC gpio "SDA/SCL/IRQ/Reset" via Logic Analyzer or Oscilloscope.

C. Touch point report issue

1. Touch works, but issue about point mapping.
   - Need to swap X and Y axis

     Please change ILITEK_ROTATE_FLAG from 0 to 1 or 1 to 0.
   - Need to reverse in X or Y axis

     Please change ILITEK_REVERT_X or ILITEK_REVERT_Y from 0 to 1 or 1 to 0.
   - Need to specify LCM resolution, please enable ILITEK_USE_LCM_RESOLUTION and modifiy TOUCH_SCREEN_X_MAX and TOUCH_SCREEN_Y_MAX accordingly.

2. No touch response
   - Please make sure ISR and IRQ number is correctly registered.
   - Please check system receive TP INT singal via command below and INT count is increasing while touching screen.

```
rk3399_firefly_mid:/data/local/tmp #  cat /proc/interrupts | grep ilitek
110:        776          0          0          0          0          0     gpio1   7 Edge      ilitek_touch_irq
```

   - Enable driver's debug log. Make sure INT signal is handled by driver, and data in debug log is correct.
   - Please monitor SDA/SCA/INT signals by Logic Analyzer or Oscilloscope, and check with spec.

3. Interrupt triggering issue
   - For Lego series TP ICs, please make sure ISR is registered as RISING EDGE TRIGGER.
   - For other series TP ICs, please make sure ISR is registered as FAILING EDGE TRIGGER.