**Modify Android to Support Atmel Touch Controller**

Runhong Deng

1. **Scope of this document.**

Atmel provides a document on this topic - ‘Accessing, Building, Configuring, and Debugging Atmel maXTouch Linux Device Drivers’. A great details was given on how to add driver for Atmel touch controller into Linux/Android.

This document serves as a supplement to Atmel’s document. It provides up-to-date details and also provides something is not included in Atmel’s document. Both hardware and software implementation will be covered in this document.

Any Atmel maxTouch chips, like mxt224, mxt768E are supported by a single driver.

1. **Hardware Connection**

I2C\_2 is used from OMAP5 to communicate with Atmel touch controller. The voltage level for this port is 1.8V in Panda5 board. In both mxt224 and mxt768E evaluation boards, the I2C bus is at 2.7V. Modification is necessary in order to make it work at 1.8V. Please refer documents from Atmel on this modification.

maxTouch

I2C

nRST

nCHG

nGPIO\_1976

I2C\_2

GPIO\_199

GPIO\_197

OMAP5

1. **Software Modification**

**The following code is added in board file - board-omap5panda.c**

* 1. **GPIO definition**

#define GPIO\_TSC\_INT 197

#define GPIO\_TSC\_RST 199

* 1. **Platform data definition**

static struct mxt\_platform\_data mxt\_platform\_data = {

.gpio\_reset = GPIO\_TSC\_RST,

.irqflags = IRQF\_TRIGGER\_FALLING,

.cfg\_name = "maxtouch.cfg"

};

* 1. **I2C initialization**

static int \_\_init omap5pandai2c\_init(void)

{

omap\_register\_i2c\_bus(1, 400, NULL, 0);

omap\_register\_i2c\_bus(2, 400, panda5\_i2c\_2\_boardinfo,

ARRAY\_SIZE(panda5\_i2c\_2\_boardinfo)); //added by R.Deng. For TSC

omap\_register\_i2c\_bus(3, 400, NULL, 0);

omap\_register\_i2c\_bus(4, 400, NULL, 0);

omap\_register\_i2c\_bus(5, 400, panda5\_i2c\_5\_boardinfo,

ARRAY\_SIZE(panda5\_i2c\_5\_boardinfo));

return 0;

}

* 1. **GPIO initialization and Interrupt Configuration**

**The following code should be added to function ‘\_\_init omap\_5\_panda\_init’**

printk(KERN\_INFO "Initializing Atmel Touch Controller - Setup GPIO and I2C\n");

err = omap\_mux\_init\_gpio(GPIO\_TSC\_RST, OMAP\_PIN\_OUTPUT | OMAP\_MUX\_MODE6); //reset pin for TSC, R.DENG

err = err + omap\_mux\_init\_gpio(GPIO\_TSC\_INT, OMAP\_PIN\_INPUT | OMAP\_MUX\_MODE6);

if(err){

pr\_err("Require GPIO for TSC = %d", err );

}

err = gpio\_request(GPIO\_TSC\_INT, "ts\_irq");

if (err < 0) {

printk(KERN\_ERR "%s: failed to request GPIO for TSC IRQ"

": %d\n", \_\_func\_\_, err);

return;

}

gpio\_direction\_input(GPIO\_TSC\_INT);

panda5\_i2c\_2\_boardinfo[0].irq = gpio\_to\_irq(GPIO\_TSC\_INT);

gpio\_export(GPIO\_TSC\_INT, true);

printk(KERN\_INFO "Done Initializing Atmel Touch Controller.\n");

1. **Kernel Configuration**

Open file ‘android\_omap\_defconfig’ in folder ‘\kernel\android-3.4\arch\arm\configs’. Change from ‘CONFIG\_TOUCHSCREEN\_ADS7846=y’ to ‘CONFIG\_TOUCHSCREEN\_ATMEL\_MXT =y’.

1. **Multi Touch support**

In order to support multi touch, need to add file ‘Atmel\_maXTouch\_Touchscreen.idc’ into android file system ‘/system/usr/idc’.  
For a touchscreen configuration, this file should minimally contain the text below.  Other params can be added to suit.

# This is an example of an input device configuration file.

# It might be used to describe the characteristics of a built-in touch screen.

# This is an internal device, not an external peripheral attached to the USB

# or Bluetooth bus.

device.internal = 1

# The device should behave as a touch screen, which uses the same orientation

# as the built-in display.

touch.deviceType = touchScreen

touch.orientationAware = 1

# Additional calibration properties...

# etc...

1. **Debug Approach**

Atmel provides great debug tools for both Linux and Windows OS. The following configuration mechanism is used during debug. In this configuration, touch controller connects to Panda5 board with I2C. However, with ADB Driver (provided Google) and ADB Bridge Client (provided by Atmel), the Atmel configuration tool, Object Server, ‘see’ the touch controller in the same way as it connects to PC directly.

A Beagle I2C/SPI Protocol Analyzer is also used to monitor all the I2C activities.

Beagle I2C/SPI Protocol Analyzer

PC

(Windows 7)

ADB Driver

ADB Bridge Client

Object Server

maxTouch

ADB

I2C\_2

USB

Panda5

(Android)

1. **TO-DO List**
   1. **Change CHG interrupt to level trigger, instead of edge trigger.**
   2. **MaxtouchUtility doesn’t work.**
2. **Linux**

**Panda5AJ.1.5 Release Notes**

<http://omapedia.org/wiki/Panda5AJ.1.5_Release_Notes>

**OMAP ADB Driver**

This is for OMAP3 and 4. But it’s verified that it works with OMAP5.

<http://omappedia.org/wiki/Support_Tools#Android_Gadget_ADB:_Windows_PC_Host>

**Atmel maxTouch Linux Driver and toolset**

<https://github.com/atmel-maxtouch>

1. **Revision**

|  |  |  |
| --- | --- | --- |
| Date | Author | Notes |
| 05/09/2013 | R. Deng | Initial release |
| 05/16/2013 | R. Deng | Added To-Do list |