



SiS I2C Touch Driver Porting Guide

Rev. 0.01

July. 22, 2011

This specification is subject to change without notice. Silicon Integrated Systems Corporation assumes no responsibility for any errors contained herein.

Copyright by Silicon Integrated Systems Corp., all rights reserved.



Revision History

Date	Rev.	Description
July. 22, 2011	0.01	Initial Release

SiS CONFIDENTIAL

Contents

1. INTRODUCTION	3
2. SOURCE CODE OVERVIEW	4
2.1. The Source Code Tree Structure	4
2.2. Source Files	4
2.2.1. Makefile	4
2.2.2. Kconfig	4
2.2.3. I2C Touch Driver	5
2.2.4. System Call Definition files	5
3. MODIFYING MAIN PROGRAM FOR DIFFERENT OPERATING SYSTEMS.....	6
3.1. I2C addresses issue	6
3.2. Platform Issue	6
3.3. Interrupt	6
4. PATCHING THE KERNEL WITH SIS I2C TOUCH DRIVER.....	7
4.1. Add System Call Definitions.....	7
4.2. Compiling and Enclosing Driver in Kernel	9

1. Introduction

This document explains how to integrate sis touch driver into Android systems.

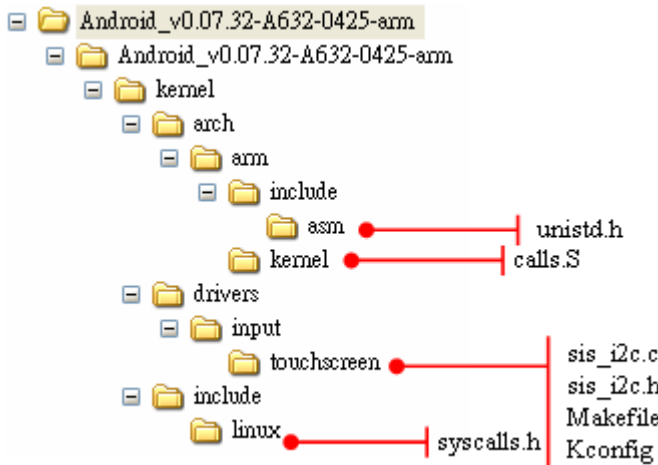
We provide a simple method to integrate and build SiS touch driver within different version of Linux kernel. Additionally, SiS I2C touch driver also uses system calls to requests a service from Linux kernel. How to define system calls will be described in the chapter, “Patching the Kernel with SiS I2C Touch Driver”.

This document contains three parts: source code overview, modifying main program for different operating systems, patching the kernel with SIS I2C touch driver. In the first part, we describe the usage of each file in the source code folder. In the second part, we explain an example about how to define system calls in Linux. On the other hand, the proper system call depends on platform.

2. Source Code Overview

2.1. The Source Code Tree Structure

A example of source code tree structure is as follows:



2.2. Source Files

2.2.1. Makefile

Makefile is located in “driver source dir/kernel/drivers/input/touchscreen”.

Use this Makefile to compile SiS I2C touch driver. Our SiS I2C touch driver can be either enclosed in Linux kernel or compiled independently as a kernel module. Its usage will be described in the chapter, “Patching the Kernel with SiS I2C Touch Driver”.

2.2.2. Kconfig

Kconfig is located in “driver source dir/kernel/drivers/input/touchscreen”.

This file is used only when enclosing SiS I2C touch driver in Linux kernel. Its usage will be described in the chapter, “Patching the Kernel with SiS I2C Touch Driver”.

2.2.3. I2C Touch Driver

Touch drivers are located in “driver source dir/kernel/drivers/input/touchscreen”.

The main program is “sis_i2c.c”, with a header file, “sis_i2c.h”. To compile main program for different operating systems will be described in “Modifying main program for different operating systems”.

2.2.4. System Call Definition files

SiS I2C touch driver uses system calls to requests a service from Linux kernel. How to define system calls will be described in the chapter, “Patching the Kernel with SiS I2C Touch Driver”.

System calls are implemented in SiS Touch driver. We should check

- implementation exists
- syscall table modified
- kernel build successfully

3. Modifying main program for different operating systems

3.1. I2C addresses issue

We generally use 0x05 as our I2C slave address. For kernel version beyond 2.6.35, address 0x04-0x07 are reserved for high speed mode master code (defined in **Kernel/drivers/i2c/i2c-core.c**). We can change SIS_SLAVE_ADDR from 0x05 to other address for the touch driver. Also we should change SIS_SLV_ADDRESS to the new address in firmware parameter settings correspondently.

```
#define SIS_SLAVE_ADDR 0x05
```

3.2. Platform Issue

If porting SiS I2C touch driver to x86 system, use SMBUS protocol to transmit data, and if porting touch driver to embedded systems, use I2C protocol to transmit data. In **sis_i2c.h**, the keyword **_SMBUS_INTERFACE** is undefined for setting I2C protocol as the default definition.

```
//#define _SMBUS_INTERFACE
```

3.3. Interrupt

For different platform, each user should configure GPIO pin for interrupt function first. For example, we use GPIO pin number 133 as our GPIO interrupt pin. Follow below function to configure GPIO pin and assign interrupt service routine to GPIO pin 133. And be sure to define the keyword **_I2C_INT_ENABLE**.

```
#define _I2C_INT_ENABLE  
gpio_request(133," gpio_133" );  
gpio_direction_input(133);  
client->irq = gpio_to_irq(133);  
request_irq(client->irq, sis_ts_irq_handler, IRQF_TRIGGER_FALLING, client->name, ts);
```

4. Patching the Kernel with SiS I2C Touch Driver

4.1. Add System Call Definitions

We provide three system calls for AP/Tools to access the firmware by touch driver. To define system calls, add definitions in kernel source code:

1. Kernel/arch/arm/include/asm/unistd.h

Define three system calls “__NR_sis_I2C_stop”, “__NR_sis_I2C_start”, and “__NR_sis_I2C_IO”, follows the last defined “__NR...” in your kernel source

Add number after the last defined number of system call. The last defined number of system call depends on Linux kernel version. In following example, the last defined number of system call is 365. Then we can progressively increase our system call number after number 365 (i.e. 366, 367, and 368).

```
#define __NR_recvmmsg      (__NR_SYSCALL_BASE+365)
/* SIS */
#define __NR_sis_I2C_stop  (__NR_SYSCALL_BASE+366)
#define __NR_sis_I2C_start (__NR_SYSCALL_BASE+367)
#define __NR_sis_I2C_IO    (__NR_SYSCALL_BASE+368)
```

2. Kernel/arch/arm/kernel/calls.S

Add function calls, “sys_sis_I2C_stop”, “sys_sis_I2C_start” and “sys_sis_I2C_IO” after the last system call.

```
/* 365 */      CALL(sys_recvmmsg)
/* 366 */      CALL(sys_sis_I2C_stop) /* SIS */
               CALL(sys_sis_I2C_start) /* SIS */
               CALL(sys_sis_I2C_IO) /* SIS */
```

3. Kernel/include/linux/syscalls.h

Declare system calls in Kernel/include/linux/syscalls.h.

```
/* SIS */  
asmlinkage long sys_sis_I2C_stop( void );  
asmlinkage long sys_sis_I2C_start( void );  
asmlinkage long sys_sis_I2C_IO( unsigned char* data, int size );
```

4.2. Compiling and Enclosing Driver in Kernel

First copy scripts listed below and paste it into the bottom of kernel/drivers/input/touchscreen/Kconfig (before endif).

```
config TOUCHSCREEN_SIS_I2C
    tristate "SiS 9200 family I2C touchscreen driver"
    depends on I2C
    help
        This enables support for SiS 9200 family over I2C based touchscreens.

config FW_SUPPORT_POWERMODE
    default y
    bool "SiS FW support power mode"
    depends on TOUCHSCREEN_SIS_I2C
    help
        This enables support power mode provided by SiS firmware
```

Then copy scripts listed below and paste it into kernel/drivers/input/touchscreen/Makefile.

```
ifdef CONFIG_TOUCHSCREEN_SIS_I2C
    obj-$(CONFIG_TOUCHSCREEN_SIS_I2C) += sis_i2c.o
endif
```

Last, execute “make menuconfig”.

```
$ make menuconfig
```

[*]Device Drivers →

```
.config - Linux Kernel v2.6.32 Configuration

Linux Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >

^(-)
  Bus support --->
  Kernel Features --->
  Boot options --->
  CPU Power Management --->
  Floating point emulation --->
  Userspace binary formats --->
  Power management options --->
  [*] Networking support --->
  Device Drivers --->
  CBUS support --->
v(+)

<Select>  < Exit >  < Help >
```

Input device support →

```
.config - Linux Kernel v2.6.32 Configuration

Device Drivers
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >

^(-)
  [ ] Misc devices --->
  < > ATA/ATAPI/MFM/RLL support (DEPRECATED) --->
  SCSI device support --->
  < > Serial ATA and Parallel ATA drivers --->
  [ ] Multiple devices driver support (RAID and LVM) --->
  [*] Network device support --->
  [ ] ISDN support --->
  < > Telephony support --->
  Input device support --->
  Character devices --->
v(+)

<Select>  < Exit >  < Help >
```

Touchscreens →

```
.config - Linux Kernel v2.6.32 Configuration

                                Input device support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >

^(-)
<*>  Event interface
< >  Event debugging
< >  Reset key
*** Input Device Drivers ***
[*]   Keyboards --->
[*]   Mice --->
[ ]   Joysticks/Gamepads --->
[ ]   Tablets --->
[*]   Touchscreens --->
[*]   Miscellaneous devices --->
v(+)

<Select>  < Exit >  < Help >
```

[*]SiS 9200 family I2C touchscreen driver

```
.config - Linux Kernel v2.6.32 Configuration

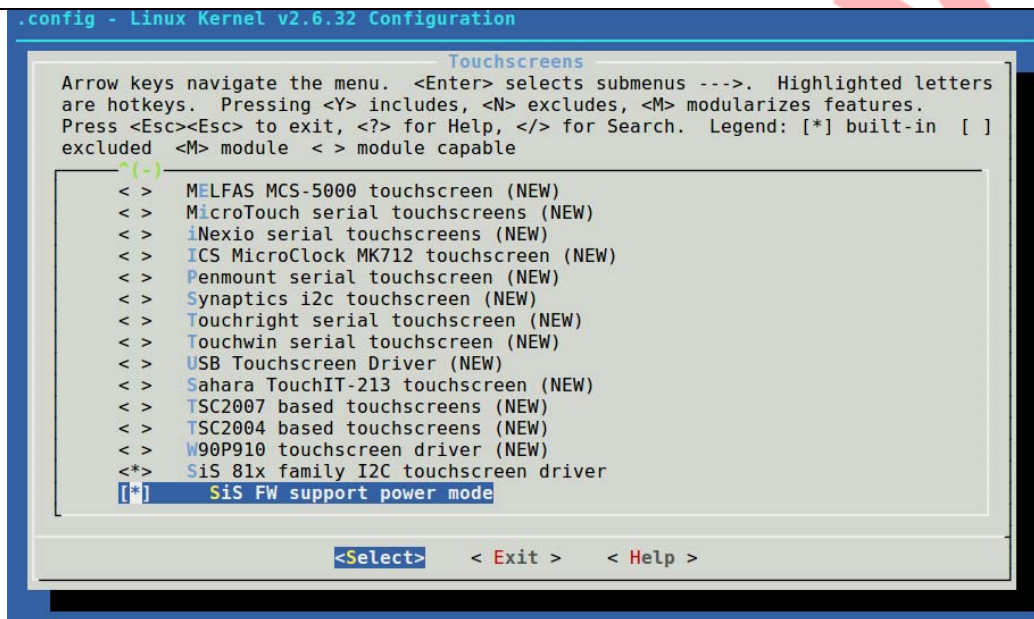
                                Touchscreens
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >

^(-)
< >  Penmount serial touchscreen
< >  Synaptics i2c touchscreen
< >  Touchright serial touchscreen
< >  Touchwin serial touchscreen
< >  USB Touchscreen Driver
< >  Sahara TouchIT-213 touchscreen
< >  TSC2007 based touchscreens
< >  TSC2004 based touchscreens
< >  W90P910 touchscreen driver
[*]  SiS 81x family I2C touchscreen driver

<Select>  < Exit >  < Help >
```

[*]SiS FW support power mode

SiS I2C touch driver supports power saving mode commands. To use power saving mode commands, power saving mode should be enabled in firmware setting. For detail, please refer to segment “3.1.10. Power Mode Command” in “Porting Guide Touch Panel Controller Programming Guide I2C Interface for 9200 series”.



```
.config - Linux Kernel v2.6.32 Configuration

Touchscreens
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters
are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features.
Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]
excluded <M> module <> module capable

[*] (-)
<> MELFAS MCS-5000 touchscreen (NEW)
<> MicroTouch serial touchscreens (NEW)
<> iNexio serial touchscreens (NEW)
<> ICS MicroClock MK712 touchscreen (NEW)
<> Penmount serial touchscreen (NEW)
<> Synaptics i2c touchscreen (NEW)
<> Touchright serial touchscreen (NEW)
<> Touchwin serial touchscreen (NEW)
<> USB Touchscreen Driver (NEW)
<> Sahara TouchIT-213 touchscreen (NEW)
<> TSC2007 based touchscreens (NEW)
<> TSC2004 based touchscreens (NEW)
<> W90P910 touchscreen driver (NEW)
<*> SiS 81x family I2C touchscreen driver
[*] SiS FW support power mode

<Select> < Exit > < Help >
```

Note that we need to compile I2C bus driver. For example, we use Intel chipset for X86 system.

[*]Device Drivers →

```
.config - Linux Kernel v2.6.35 Configuration

Linux Kernel Configuration

Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >

  General setup --->
  [*] Enable loadable module support --->
  *- Enable the block layer --->
    Processor type and features --->
    Power management and ACPI options --->
    Bus options (PCI etc.) --->
    Executable file formats / Emulations --->
  [*] Networking support --->
  Device Drivers --->
  Firmware Drivers --->

v(+)
```

I2C support →

```
.config - Linux Kernel v2.6.35 Configuration

Device Drivers

Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >

^(-)
< > I2O device support --->
[ ] Macintosh device drivers --->
[*] Network device support --->
[ ] ISDN support --->
< > Telephony support --->
  Input device support --->
  Character devices --->
  [*] I2C support --->
  [*] SPI support --->
  PPS support --->

v(+)
```


I2C Hardware Bus Support➔

```
.config - Linux Kernel v2.6.35 Configuration

                                I2C support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >

--- I2C support
[*] Enable compatibility bits for old user-space
<*> I2C device interface
[*] Autoselect pertinent helper modules
||| I2C Hardware Bus support --->
[ ] I2C Core debugging messages
[ ] I2C Algorithm debugging messages
[ ] I2C Bus debugging messages

<Select> < Exit > < Help >
```

[*]Intel 82801(ICH/PCH)

```
.config - Linux Kernel v2.6.35 Configuration

                                I2C Hardware Bus support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >

*** PC SMBus host controller drivers ***
<M> ALI 1535
<M> ALI 1563
<M> ALI 15x3
<M> AMD 756/766/768/8111 and nVidia nForce
<M> SMBus multiplexing on the Tyan S4882
<M> AMD 8111
<+> Intel 82801 (ICH/PCH)
<M> Intel SCH SMBus 1.0
<M> Intel PIIX4 and compatible (ATI/AMD/Serverworks/Broadcom/SMSC)
v(+)

<Select> < Exit > < Help >
```