# I2C驱动编写及测试

**【实验内容】**

编写一个基于i2c总线的温度传感器的驱动

**【实验目的】**

掌握I2C总线的操作方法

**【实验平台】**

主机：Ubuntu 10.10

目标机：FS\_S5PC100

目标机内核版本：2.6.35

交叉编译器版本：arm-none-linux-gnueabi-gcc-4.5.1

**【实验步骤】**

**注意：在实验过程中"$"后的操作在主机上，"#"后的操作在开发板上**

修改gedit drivers/i2c/busses/Kconfig

修改

config I2C\_S3C2410

tristate "S3C2410 I2C Driver"

depends on ARCH\_S3C2410 || ARCH\_S3C64XX

help

Say Y here to include support for I2C controller in the

Samsung S3C2410 based System-on-Chip devices.

为：

config I2C\_S3C2410

tristate "S3C2410 I2C Driver"

depends on ARCH\_S3C2410 || ARCH\_S3C64XX || ARCH\_S5PC100

help

Say Y here to include support for I2C controller in the

Samsung S3C2410 based System-on-Chip devices.

内核配置并重新编译内核

$ make menuconfig

Device Drivers --->

<\*> I2C support --->

<\*> I2C device interface

I2C Hardware Bus support --->

<\*> S3C2410 I2C Driver

$ make zImage

一、用户模式驱动编写

1. 拷贝模块到linux下

将实验代码中的s5pc100\_temp\_app拷贝到/home/linux目录下并进入s5pc100\_temp\_app目录。

1. 编译模块

$ make

1. 编译应用程序

$ arm-none-linux-gnueabi-gcc temp\_app\_mode.c –o temp\_app\_mode

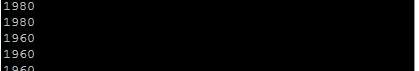
1. 拷贝应用程序到目标板上

$ temp\_app\_mode /source/rootfs

1. 测试

# ./temp\_app\_mode

用手按住温度传感器会发现值会改变



这个值是16进制数，前两位代表度，后两位是小数点后的值，大家可以自己转换一下

二、使用内核i2c子系统构建驱动

1. 修改平台代码并重新编译内核

修改arch/arm/mach-s5pc100/mach-smdkc100.c

修改：

static struct i2c\_board\_info i2c\_devs0[] \_\_initdata = {

};

为：

static struct i2c\_board\_info i2c\_devs0[] \_\_initdata = {

{

I2C\_BOARD\_INFO("lm75", 0x48),

},

};

$ make zImage

1. 拷贝模块到linux下

将实验代码中的s5pc100\_temp拷贝到/home/linux目录下并进入s5pc100\_temp目录。

1. 编译模块

$ make

1. 编译应用程序

$ arm-none-linux-gnueabi-gcc test.c –o test

1. 拷贝驱动及应用程序到目标板上

$ cp s5pc100\_temp.ko test /source/rootfs

1. 启动开发板后加载模块

# insmod s5pc100\_temp.ko

1. 创建设备节点

# mknod /dev/temp c 250 0

1. 测试

# ./test

