

## iTOP-4412 以模块的方式编译驱动

大家好，本章节我们将向大家讲解如何在 linux 下实现以模块的方式加载内核驱动。我们以内核里面蜂鸣器的驱动为例来讲解。

1) 首先打开内核的源码，如下图所示：

```
root@ubuntu:/home/brower/iTop4412_Kernel_3.0#
root@ubuntu:/home/brower/iTop4412_Kernel_3.0# ls
arch          config_for_ubuntu  drivers  Kbuild      Makefile     README      tools
binary        config_for_ubuntu_hdmi  firmware Kconfig      mm           REPORTING-BUGS  usr
block         COPYING            fs       kernel      modem.patch  samples        virt
config_for_android  CREDITS           include  kernel_readme.txt  Module.symvers  scripts
config_for_android_2M  crypto           init     lib          net           security
config_for_linux      Documentation      ipc      MAINTAINERS  pull_log.bat  sound
root@ubuntu:/home/brower/iTop4412_Kernel_3.0#
```

2) 使用命令 “cd drivers/char/”，进入到蜂鸣器驱动所在的目录，如下图所示：

```
root@ubuntu:/home/brower/iTop4412_Kernel_3.0# cd drivers/char/
root@ubuntu:/home/brower/iTop4412_Kernel_3.0/drivers/char#
root@ubuntu:/home/brower/iTop4412_Kernel_3.0/drivers/char# ls
agp          dsp56k.c          hw_random          max485_ctl.c      nvram.c           raw.c             tb0219.c
apm-emulation.c  dtlk.c           i8k.c             mbcs.c            nwbutton.c        rtc.c             tlc1k.c
applicom.c      efirtc.c          ipmi              mbcs.h            nwbutton.h        s3c_mem.c         toshiba.c
applicom.h      exynos_mem.c      itop4412_adc.c    mem.c             nwflash.c         s3c_mem.h         tpm
bfin-otp.c      generic_nvram.c   itop4412_buzzer.c  misc.c            pc8736x_gpio.c    sec.h             ttyprintk.c
briq_panel.c    genrtc.c          itop4412_leds.c   mmtimer.c         pcmcia             scx200_gpio.c     uv_mmtimer.c
bsr.c           gps.c             itop4412_relay.c  msm_smd_pkt.c     ppdev.c           snsc.c            viotape.c
dec_tty.c       gps.h             Kconfig           mspec.c           ps3flash.c        snsc_event.c      virtio_console.c
ds1302.c        hangcheck-timer.c lp.c              mwave             ramoops.c          snsc.h            xilinx_hwicap
ds1620.c        hpet.c           Makefile          nsc_gpio.c        random.c           sonypi.c
```

3) 然后使用命令 “vi Kconfig” 打开当前目录下的内核配置文件，如下图所示：

```
root@ubuntu:/home/brower/iTop4412_Kernel_3.0/drivers/char#
root@ubuntu:/home/brower/iTop4412_Kernel_3.0/drivers/char# vi Kconfig

#
# Character device configuration
#

menu "Character devices"

source "drivers/tty/Kconfig"

config DEVMEM
    bool "Memory device driver"
    default y
    help
```

4) 然后找到 “config BUZZER\_CTL” 所在的位置，如下图所示：

```
config BUZZER_CTL
    bool "Enable BUZZER config"
    default n
    help
        Enable BUZZER config
```

5) 然后把 "bool "Enable BUZZER config" "一行改成" tristate "Enable BUZZER config" ", 如下图所示 :

```
config BUZZER_CTL
    tristate "Enable BUZZER config"
    default n
    help
        Enable BUZZER config
```

6) 然后保存并退出, 如下图所示 :

```
Kconfig: 690L, 23469C written
root@ubuntu:/home/brosver/iTop4412_Kernel_3.0/drivers/char#
root@ubuntu:/home/brosver/iTop4412_Kernel_3.0/drivers/char#
```

7) 然后回到内核源码的根目录下, 如下图所示 :

```
root@ubuntu:/home/brosver/iTop4412_Kernel_3.0#
root@ubuntu:/home/brosver/iTop4412_Kernel_3.0#
root@ubuntu:/home/brosver/iTop4412_Kernel_3.0#
```

8) 然后输入命令" make menuconfig "配置内核, 如下图所示 :

```

root@ubuntu:/home/browser/iTop4412_Kernel_3.0# make menuconfig
scripts/kconfig/mconf Kconfig

.config - Linux/arm 3.0.15 Kernel Configuration

----- Linux/arm 3.0.15 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 <> module capable

[*] Patch physical to virtual translations at runtime (EXPERIMENTAL)
    General setup --->
[*] Enable loadable module support --->
[*] Enable the block layer --->
    System Type --->
[ ] FIQ Mode Serial Debugger
    Bus support --->
    Kernel Features --->
    Boot options --->
    CPU Power Management --->
    Floating point emulation --->
    Userspace binary formats --->
    Power management options --->
v(+)

```

9) 选择“ Device Drivers ”->“Character devices”->“Enable BUZZER config”，如下图所示：

```

----- Character devices -----
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

^(-)
<> Siemens R3964 line discipline
<> RAW driver (/dev/raw/rawN)
<> TPM Hardware Support --->
<> DCC tty driver
<> Log panic/oops to a RAM buffer
[*] Support for /dev/s3c-mem
[*] Support for /dev/exynos-mem
[*] Enable GPS PM
[*] Enable MAX485 pin config
[*] Enable LEDS config
<M> Enable BUZZER config
[*] ADC driver for iTOP4412
[*] Enable RELAY config

<Select> < Exit > < Help >

```

10) 然后把“ Enable BUZZER config ”左边的“ \* ”改成“ M ”，如下图所示：

```

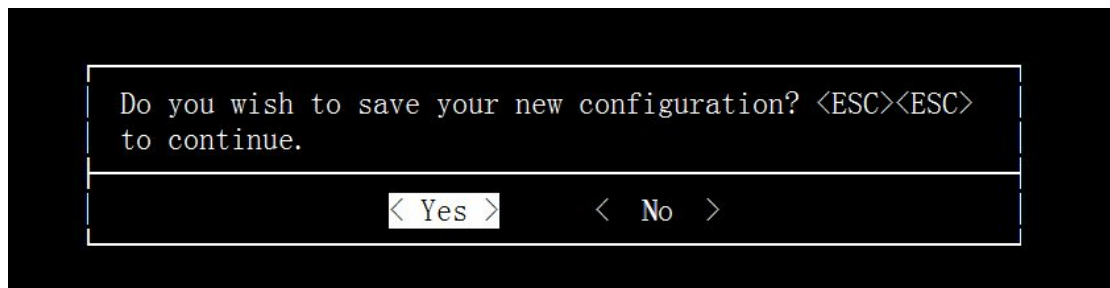
----- Character devices -----
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing
<Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for
Search. Legend: [*] built-in [ ] excluded <M> module <> module capable

^(-)
<> Siemens R3964 line discipline
<> RAW driver (/dev/raw/rawN)
<> TPM Hardware Support --->
<> DCC tty driver
<> Log panic/oops to a RAM buffer
[*] Support for /dev/s3c-mem
[*] Support for /dev/exynos-mem
[*] Enable GPS PM
[*] Enable MAX485 pin config
[*] Enable LEDS config
<M> Enable BUZZER config
[*] ADC driver for iTOP4412
[*] Enable RELAY config

<Select> < Exit > < Help >

```

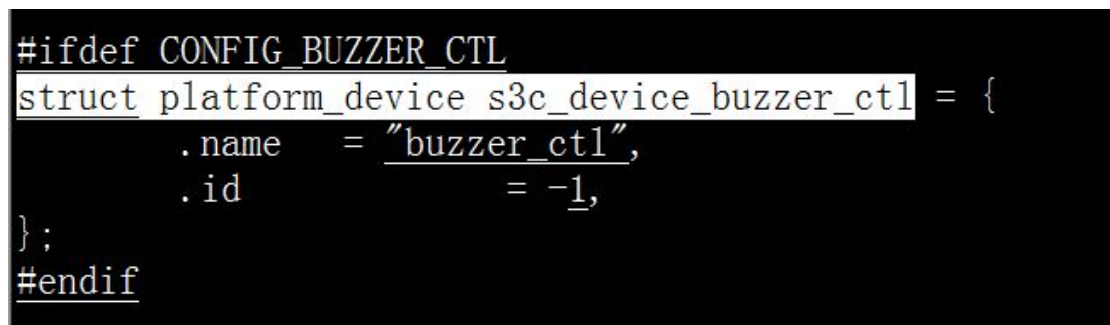
11) 然后保存并退出配置界面，如下图：



12) 然后使用命令“ vi arch/arm/mach-exynos/mach-itop4412.c ”打开 mach-itop4412.c，如下图所示：



13) 然后找到“ struct platform\_device s3c\_device\_buzzer\_ctl ”一行，如下图所示：



14) 把这一行前面的“ #ifdef CONFIG\_BUZZER\_CTL ”改成“ #if defined(CONFIG\_BUZZER\_CTL) || defined(CONFIG\_BUZZER\_CTL\_MODULE) ”，如下图所示：



```
#if defined(CONFIG_BUZZER_CTL) || defined(CONFIG_BUZZER_CTL_MODULE)
struct platform_device s3c_device_buzzer_ctl = {
    .name    = "buzzer_ctl",
    .id      = -1,
};
#endif
```

15) 然后找到“ &s3c\_device\_buzzer\_ctl ”一行，如下图所示：

```
#ifdef CONFIG_BUZZER_CTL
    &s3c_device_buzzer_ctl,
#endif
```

16) 把这一行前面的“ #ifdef CONFIG\_BUZZER\_CTL ”改成“ #if defined(CONFIG\_BUZZER\_CTL) || defined(CONFIG\_BUZZER\_CTL\_MODULE) ”，如下图所示：

```
#if defined(CONFIG_BUZZER_CTL) || defined(CONFIG_BUZZER_CTL_MODULE)
    &s3c_device_buzzer_ctl,
#endif
```

17) 然后保存并退出，返回到 linux 内核源码的根目录下，如下图所示：

```
root@ubuntu:/home/brosner/iTop4412_Kernel_3.0#
root@ubuntu:/home/brosner/iTop4412_Kernel_3.0#
root@ubuntu:/home/brosner/iTop4412_Kernel_3.0#
```

18) 然后输入命令“ make ”，开始编译内核，如下图所示：

```
root@ubuntu:/home/brosner/iTop4412_Kernel_3.0#
root@ubuntu:/home/brosner/iTop4412_Kernel_3.0#
root@ubuntu:/home/brosner/iTop4412_Kernel_3.0# make
```

19) 编译完成后会在内核的“ arch/arm/boot ”目录下生成镜像文件“ zImage ”，如下图所示：

```
root@ubuntu:/home/brosner/iTop4412_Kernel_3.0/arch/arm/boot#
root@ubuntu:/home/brosner/iTop4412_Kernel_3.0/arch/arm/boot# ls
bootp compressed Image install.sh Makefile zImage
root@ubuntu:/home/brosner/iTop4412_Kernel_3.0/arch/arm/boot#
```

20) 在内核的“ drivers/char ”目录下生成了蜂鸣器的驱动模块“ itop4412\_buzzer.ko ”，如下图所示：

```

root@ubuntu:/home/brosver/iTop4412_Kernel_3.0/drivers/char# ls
agp                exynos_mem.o      itop4412_buzzer.mod.c  mem.o             pcmcia            snsc.h
apm-emulation.c   generic_nvram.c   itop4412_buzzer.mod.o  misc.c            ppdev.c           sonypi
applicom.c         genrtc.c          itop4412_buzzer.o     misc.o            ps3flash.c        tb0219
applicom.h         gps.c             itop4412_leds.c       mmtimer.c         ramoops.c         tlc1k
bfin-otp.c         gps.h             itop4412_leds.o       modules.builtin   random.c          toshib
briq_panel.c      gps.o             itop4412_relay.c      modules.order     random.o          tpm
bsr.c              hangcheck-timer.c itop4412_relay.o      msm_smd_pkt.c     raw.c             ttypri
built-in.o         hpet.c            Kconfig               mspec.c           rtc.c             uv_mmt
dcc_tty.c          hw_random         lp.c                  mwave             s3c_mem.c         viotap
ds1302.c           i8k.c             Makefile              nsc_gpio.c        s3c_mem.h         virtio
ds1620.c           ipmi               max485_ctl.c          nvram.c           s3c_mem.o         xilinx
dsp56k.c           itop4412_adc.c    max485_ctl.o          nwbutton.c        scc.h             scx200_gpio.c
dtlk.c             itop4412_adc.o    mbcs.c                nwbutton.h        scsc.c            snsc.c
efirtc.c           itop4412_buzzer.c mbcs.h                nwflash.c         snsc_event.c
exynos_mem.c       itop4412_buzzer.ko mem.c                  pc8736x_gpio.c
root@ubuntu:/home/brosver/iTop4412_Kernel_3.0/drivers/char#

```

21) 下一步我们烧写生成的 zImage 到开发板上，然后开发板启动进入到 android 系统。

22) 然后通过 adb 把蜂鸣器的驱动模块传到开发板的"/data"目录下，如下图所示：

```

D:\bsp\4412_DUT\USB_fastboot_tool\platform-tools>
D:\bsp\4412_DUT\USB_fastboot_tool\platform-tools>adb.exe push itop4412_buzzer.ko
/data
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
903 KB/s (101732 bytes in 0.110s)
D:\bsp\4412_DUT\USB_fastboot_tool\platform-tools>_

```

23) 然后再串口输入命令“ cd /data ”，进入到开发板的"/data"目录，如下图所示：

```

root@android:/ #
root@android:/ # cd /data
root@android:/data #
root@android:/data #
root@android:/data #

```

24) 然后输入“ ls ”命令，可以看到通过 adb 传过来的蜂鸣器驱动模块“ itop4412\_buzzer.ko ”，如下图所示：

```
root@android:/ # cd /data/  
root@android:/data #  
root@android:/data #  
root@android:/data # ls  
app  
app-private  
backup  
bluetooth  
dalvik-cache  
data  
dontpanic  
drm  
itop4412_buzzer.ko  
local  
lost+found  
misc  
property  
resource-cache  
system  
tombstones  
user  
wifi  
root@android:/data #
```

25) 接着在串口输入命令“ insmod itop4412\_buzzer.ko ”, 加载蜂鸣器的驱动, 如下图所示:

```
root@android:/data # insmod itop4412_buzzer.ko  
[ 414.652986] buzzer_ctl initialized  
root@android:/data #  
root@android:/data #
```

26) 通过上图可以看到蜂鸣器的驱动已经加载到内核里面了, 接着输入命令“cd /dev”, 如下图所示:

```
root@android:/data # cd /dev
root@android:/dev # ll
crwxrwxrwx root    root      10,  61 2012-01-01 05:20 AGPS
crw-rw---- system system    10, 243 2012-01-01 05:20 HPD
crwxrwxrwx root    root      10,  58 2012-01-01 05:20 adc
crw-rw-r-- system radio     10,  52 2012-01-01 05:20 alarm
crw-rw---- adb     adb       10,  56 2012-01-01 05:20 android_adb
crw-rw-rw- root    root      10,  62 2012-01-01 05:20 ashmem
crw-rw-rw- root    root      10,  51 2012-01-01 05:20 binder
drwxr-xr-x root    root           2012-01-01 05:20 block
drwxr-xr-x root    root           2012-01-01 05:20 bus
crw----- root    root      10,  42 2012-01-01 05:26 buzzer_ctl
drwxr-xr-x root    root           2012-01-01 05:21 com.qihoo.permmgr.shellservice
crw----- root    root       5,   1 2012-01-01 05:27 console
```

27) 通过上面的图片, 可以看到生成了 `buzzer_ctl` 蜂鸣器的节点了, 我们需要输入 `chmod 777 buzzer_ctl`

命令来修改下 `buzzer_ctl` 的权限, 如下图所示:

```
root@android:/dev #
root@android:/dev # chmod 777 buzzer_ctl
root@android:/dev #
root@android:/dev #
```

28) 然后我们可以使用 android 自带的蜂鸣器小程序来控制蜂鸣器了, 至此以模块的方式加载驱动就完成了。

上面我们的驱动是放在了内核源码的目录下来实现的编译成驱动模块, 很多时候我们都是拿到一个驱动源码, 不需要把他放到内核源码里面, 而是直接把他编译成驱动模块, 下面我们来讲下实现方法, 我们还是以蜂鸣器的驱动为例来讲解。

1) 因为开发板带的内核默认是把蜂鸣器直接编译到内核里面了, 所以我们要重复前边的步骤“1”到步骤“19”, 完成这些步骤以后, 把生成的镜像“zImage”烧写到开发板。

2) 然后在虚拟机的目录下用命令 `mkdir` 建立文件夹“module”, 如下图所示:

```
root@ubuntu:/home/brosver#
root@ubuntu:/home/brosver#
root@ubuntu:/home/brosver# mkdir module
```

3) 然后用命令 `cd` 进入到建立的“module”文件夹, 如下图所示:



```
root@ubuntu:/home/brosver#  
root@ubuntu:/home/brosver# cd module/  
root@ubuntu:/home/brosver/module#  
root@ubuntu:/home/brosver/module#
```

4) 拷贝内核里面的蜂鸣器驱动到 module 文件夹，如下图所示：

```
root@ubuntu:/home/brosver/module#  
root@ubuntu:/home/brosver/module# cp /home/brosver/iTop4412_Kernel_3.0/drivers/char/itop4412_buzzer.c ./  
root@ubuntu:/home/brosver/module#
```

5) 然后在 module 文件夹建立文件 “Makefile”，如下图所示：

```
root@ubuntu:/home/brosver/module#  
root@ubuntu:/home/brosver/module# vi Makefile
```

6) 然后在“Makefile”文件输入下面的信息：

```
obj-m += itop4412_buzzer.o
```

```
KDIR := /home/brosver/iTop4412_Kernel_3.0
```

```
PWD = $(shell pwd)
```

```
all:
```

```
make -C $(KDIR) M=$(PWD) modules
```

```
clean:
```

```
rm -rf *.o
```

其中的 “KDIR := /home/brosver/iTop4412\_Kernel\_3.0”是指定我们内核源码所在的目录，我这里内核源码

是放在了” /home/broswer/ “目录下面，需要根据自己的存放位置来修改这个地方

7) 然后保存并退出 Makefile，在终端输入” make “命令，编译蜂鸣器驱动，如下图所示：

```
root@ubuntu:/home/broswer/module# make
make -C /home/broswer/iTop4412_Kernel_3.0 M=/home/broswer/module modules
make[1]: Entering directory `/home/broswer/iTop4412_Kernel_3.0'
  CC [M] /home/broswer/module/itop4412_buzzer.o
/home/broswer/module/itop4412_buzzer.c:90: warning: initialization from incompatible pointer type
Building modules, stage 2.
MODPOST 1 modules
  CC      /home/broswer/module/itop4412_buzzer.mod.o
  LD [M]  /home/broswer/module/itop4412_buzzer.ko
make[1]: Leaving directory `/home/broswer/iTop4412_Kernel_3.0'
root@ubuntu:/home/broswer/module#
```

8) 编译完成后，可以看下在” module “文件夹下面生成了” itop4412\_buzzer.ko “，如下图所示：

```
root@ubuntu:/home/broswer/module#
root@ubuntu:/home/broswer/module# ls
itop4412_buzzer.c  itop4412_buzzer.mod.c  itop4412_buzzer.o  modules.order
itop4412_buzzer.ko  itop4412_buzzer.mod.o  Makefile           Module.symvers
root@ubuntu:/home/broswer/module#
```

9) 然后可以参照前面的步骤” 22 “来测试我们编译的蜂鸣器驱动了。