

7.3 Simplest module for embedded Linux drivers

Section 7.1 introduce a simple Linux program Hello World, which is run in user mode applications, we now introduce a run in kernel mode Hello World program, it is actually a simple driver module.

7.3.1 Hello module source code

Source code description:

Source code directory	/opt/FriendlyARM/mini2440/linux-2.6.32.2/drivers/char
Source code file name	mini2440_hello_module.c
The driver major number	-
Device Name	-
Test program source code directory	-
Test program name	-
Test program executable file name	-
Description: The driver will not load any device to create the dev nodes.	

Source code:

```
#include <linux/kernel.h>
#include <linux/module.h>
static int __init mini2440_hello_module_init(void)
{
    printk("Hello, Mini2440 module is installed !\n");
    return 0;
}
static void __exit mini2440_hello_module_cleanup(void)
{
    printk("Good-bye, Mini2440 module was removed!\n");
}
module_init(mini2440_hello_module_init);
module_exit(mini2440_hello_module_cleanup);
MODULE_LICENSE("GPL");
```

7.3.2 Adding Hello module to the kernel source tree and compile

Generally compiler of the driver module version 2.6 need to add code to the kernel source tree, and do the appropriate configuration, as follows (Note: The following steps are already done in fact, you only need to open the check directly to compiler on it):

Step 1: Edit the configuration file Kconfig, or add options to make it appear in the make menuconfig when open linux-2.6.32.2/drivers/char/Kconfig file, add as shown:

```

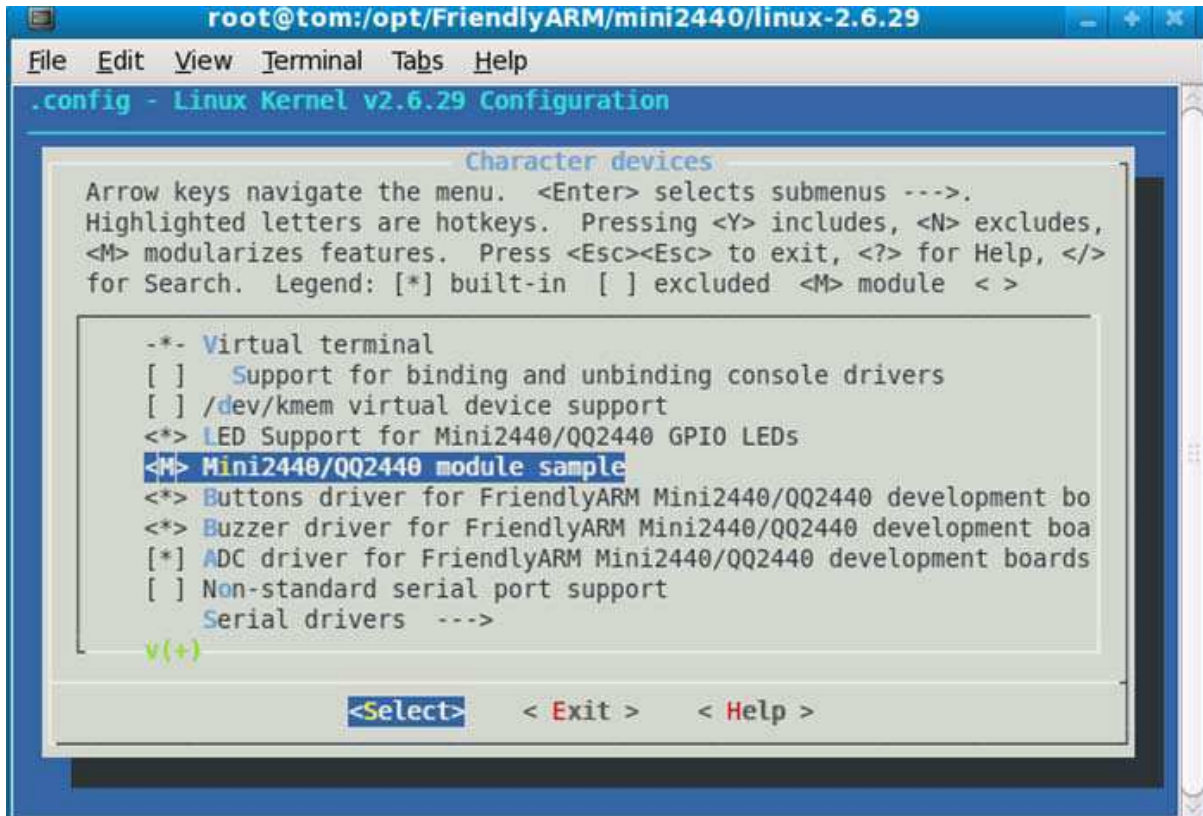
root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29/drivers/char
File Edit View Terminal Tabs Help
/dev/kmem device is rarely used, but can be used for certain
kind of kernel debugging operations.
When in doubt, say "N".

config LEDS_MINI2440
    tristate "LED Support for Mini2440/QQ2440 GPIO LEDs"
    depends on ARCH_S3C2410
    help
        This option enables support for LEDs connected to GPIO lines
        on Mini2440/QQ2440 boards.

config MINI2440_HELLO_MODULE
    tristate "Mini2440/QQ2440 module sample"
    depends on ARCH_S3C2410
    default m if MACH_FRIENDLY_ARM_MINI2440
    help
        Mini2440/QQ2440 module sample.

config MINI2440_BUTTONS
    tristate "Buttons driver for FriendlyARM Mini2440/QQ2440 development boards"
    depends on MACH_FRIENDLY_ARM_MINI2440
    default y if MACH_FRIENDLY_ARM_MINI2440
    
```

Save and exit, this time in the linux-2.6.32.2 directory where you run what you can make menuconfig “Device Drivers > Character devices” to see just add menu options, and press the space bar will select the <M> this means that the option should compiled as a module; then press the space will become <*>, which means should this option compiled into the kernel, we choose <M>, as shown:



Step 2: the previous step, although we can choose when configuring the kernel, but in fact the implementation at this time or can't mini2440_hello_module.c kernel compiled, but also need to in the Makefile in the kernel configuration options and the actual source code linked open linux-2.6.32.2/drivers/char/Makefile, add and save and exit as shown in figure:

```

root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29
File Edit View Terminal Tabs Help
obj-$(CONFIG_IPMI_HANDLER) += ipmi/

obj-$(CONFIG_HANGCHECK_TIMER) += hangcheck-timer.o
obj-$(CONFIG_TCG_TPM) += tpm/

obj-$(CONFIG_PS3_FLASH) += ps3flash.o

obj-$(CONFIG_JS_RTC) += js-rtc.o
js-rtc-y = rtc.o

obj-$(CONFIG_LEDS_MINI2440) += mini2440_leds.o
obj-$(CONFIG_MINI2440_HELLO_MODULE) += mini2440_hello_module.o
obj-$(CONFIG_MINI2440_BUTTONS) += mini2440_buttons.o
obj-$(CONFIG_MINI2440_BUZZER) += mini2440_pwm.o
obj-$(CONFIG_MINI2440_ADC) += mini2440_adc.o

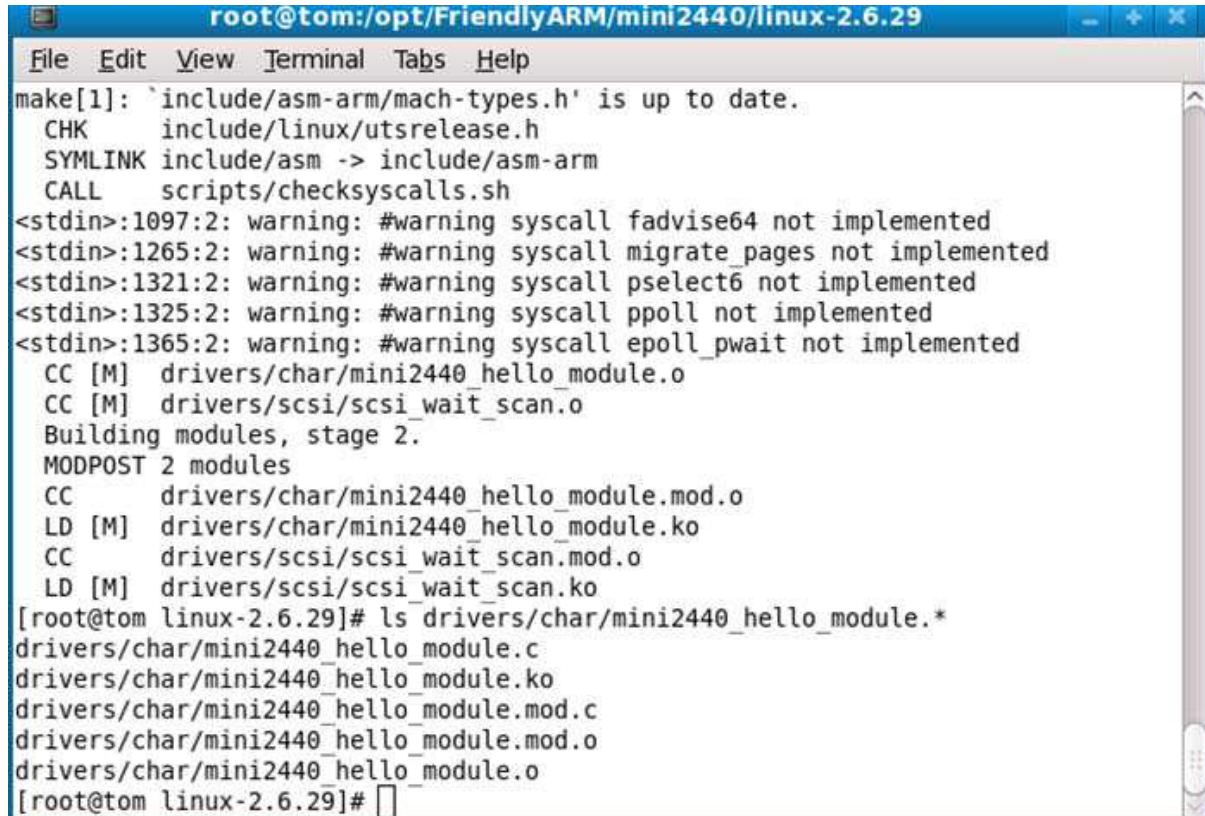
# Files generated that shall be removed upon make clean
clean-files := consolemap_deftbl.c defkeymap.c

quiet_cmd_conmk = CONMK  $@
      cmd_conmk = scripts/conmakehash $< > $@

$(obj)/consolemap_deftbl.c: $(src)/$(FONTMAPFILE)
    
```

Step 3: then back to linux-2.6.32.2 root directory of the location, do make modules, we can produce the required documents mini2440_hello_module.ko a kernel module, as shown:

So far, we have completed the model-driven compilation.



```
root@tom:/opt/FriendlyARM/mini2440/linux-2.6.29
File Edit View Terminal Tabs Help
make[1]: `include/asm-arm/mach-types.h' is up to date.
CHK    include/linux/utsrelease.h
SYMLINK include/asm -> include/asm-arm
CALL   scripts/checksyscalls.sh
<stdin>:1097:2: warning: #warning syscall fadvise64 not implemented
<stdin>:1265:2: warning: #warning syscall migrate_pages not implemented
<stdin>:1321:2: warning: #warning syscall pselect6 not implemented
<stdin>:1325:2: warning: #warning syscall ppoll not implemented
<stdin>:1365:2: warning: #warning syscall epoll_pwait not implemented
CC [M]  drivers/char/mini2440_hello_module.o
CC [M]  drivers/scsi/scsi_wait_scan.o
Building modules, stage 2.
MODPOST 2 modules
CC      drivers/char/mini2440_hello_module.mod.o
LD [M]  drivers/char/mini2440_hello_module.ko
CC      drivers/scsi/scsi_wait_scan.mod.o
LD [M]  drivers/scsi/scsi_wait_scan.ko
[root@tom linux-2.6.29]# ls drivers/char/mini2440_hello_module.*
drivers/char/mini2440_hello_module.c
drivers/char/mini2440_hello_module.ko
drivers/char/mini2440_hello_module.mod.c
drivers/char/mini2440_hello_module.mod.o
drivers/char/mini2440_hello_module.o
[root@tom linux-2.6.29]#
```

7.3.3 Downloading Hello module and install on the development board.

This rz command the compile using the mini2440_hello_module.ko downloaded to the board and move it to /lib/modules/2.6.29.4-FriendlyARM directory and then execute the board are now.

```
#modprobe mini2440_hello_module
```

Can see that the module has been loaded (**note: do not load the module using the modprobe command added "ko" suffix**)

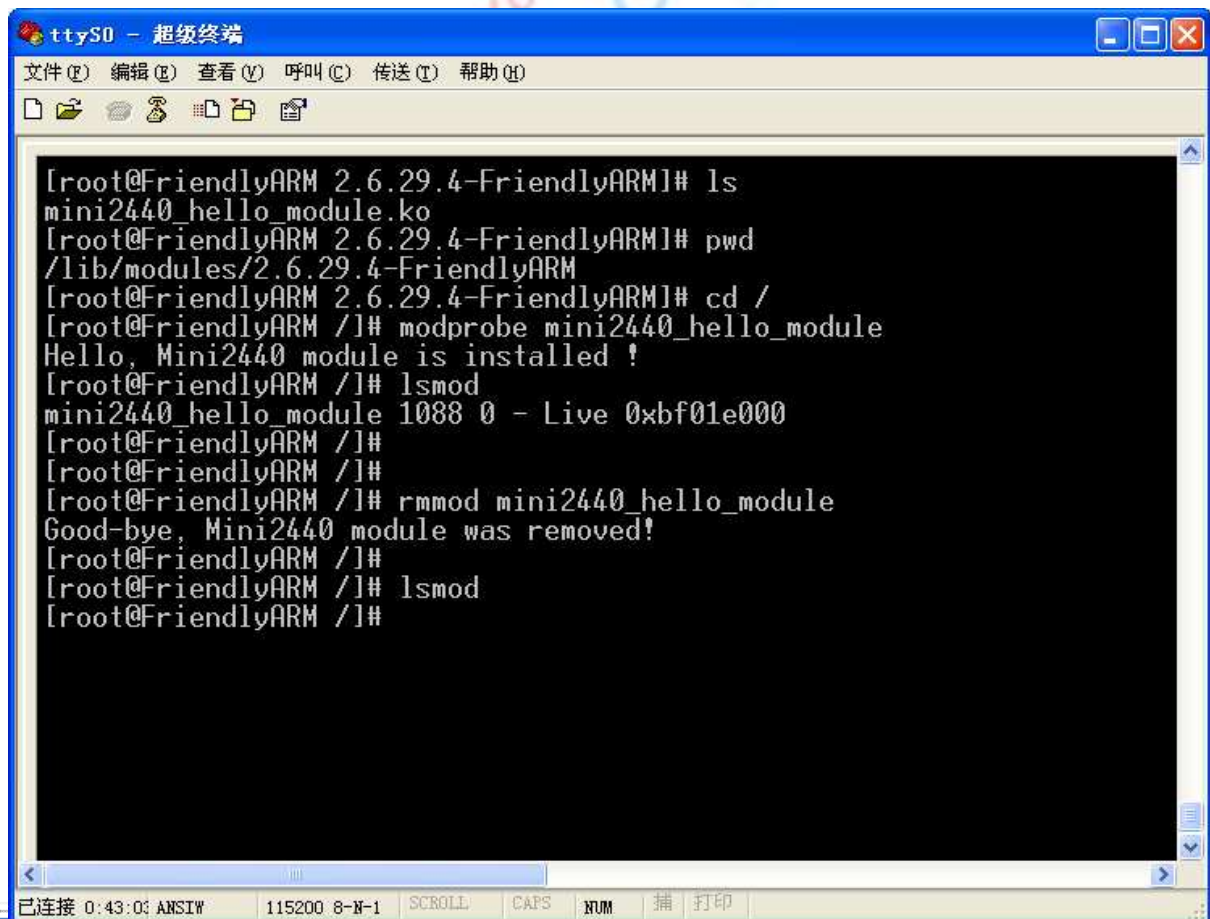
And then execute the following command; you can see that the module is unloaded.

```
#rmmod mini2440_hello_module
```

Note: To be able to properly unload the module, the module must be the into the development board /lib/modules/2.6.29.4-FriendlyARM directory

Also to note: because our kernel updates sometimes, if the kernel version has changed, follow the specific kernel version of the module to re-establish a store directory, This is /lib/modules/2.6.29.4-FriendlyARM

The whole process as below:



```
tttyS0 - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
[root@FriendlyARM 2.6.29.4-FriendlyARM]# ls
mini2440_hello_module.ko
[root@FriendlyARM 2.6.29.4-FriendlyARM]# pwd
/lib/modules/2.6.29.4-FriendlyARM
[root@FriendlyARM 2.6.29.4-FriendlyARM]# cd /
[root@FriendlyARM /]# modprobe mini2440_hello_module
Hello, Mini2440 module is installed !
[root@FriendlyARM /]# lsmod
mini2440_hello_module 1088 0 - Live 0xbf01e000
[root@FriendlyARM /]#
[root@FriendlyARM /]#
[root@FriendlyARM /]# rmmod mini2440_hello_module
Good-bye, Mini2440 module was removed!
[root@FriendlyARM /]#
[root@FriendlyARM /]# lsmod
[root@FriendlyARM /]#
```

已连接 0:43:00 ANSIW 115200 8-N-1 SCROLL CAPS NUM 捕 打印