

第 18 章	驱动开发之字符设备驱动程序编译测试 .....	2
18.1	创建 Makefile 文件 .....	2
18.2	配置编译驱动程序 .....	3
18.3	动态的加载和卸载内核驱动模块软件包 .....	5

[www.hi-wooya.cc](http://www.hi-wooya.cc)

## 第 18 章 驱动开发之字符设备驱动程序编译测试

本章目标

- 掌握如何编译自己的内核驱动模块
- 掌握如何动态的加载和卸载内核驱动模块软件包

### 18.1 创建 Makefile 文件

在上一章中，我们讲了很多关于 Makefile 的语法知识，但是并没有举例讲解如何应用。接下来，我们就来利用前面的知识，写出一个自己的 Makefile 来。

首先创建一个 f403tech\_drv 文件夹，然后进入该文件夹，再创建一个 src 文件夹，将第 16 章写的驱动程序 f403tech\_drv.c 拷贝到 src 目录。然后在 src 目录下创建一个 Makefile 文件，内容如下。

```
obj-m += f403tech_drv.o
```

然后再回到 f403tech\_drv 文件夹，接着再创建一个 Makefile 文件，内容如下。

```
#
# Copyright (C) 2008-2012 OpenWrt.org
#
# This is free software, licensed under the GNU General Public License v2.
# See /LICENSE for more information.
#

include $(TOPDIR)/rules.mk
include $(INCLUDE_DIR)/kernel.mk

PKG_NAME:=f403tech_drv
PKG_RELEASE:=1

include $(INCLUDE_DIR)/package.mk

define KernelPackage/f403tech_drv
    SUBMENU:=Other modules
```

```
# DEPENDS:=@!LINUX_3_3
TITLE:=Motor driver
FILES:=$(PKG_BUILD_DIR)/f403tech_drv.ko
# AUTOLOAD=$(call AutoLoad,30,f403tech_drv,1)
KCONFIG:=
endif

define KernelPackage/f403tech_drv/description
This is a f403tech_drv drivers
endif

MAKE_OPTS:= \
    ARCH="$(LINUX_KARCH)" \
    CROSS_COMPILE="$(TARGET_CROSS)" \
    SUBDIRS="$(PKG_BUILD_DIR)"

define Build/Prepare
    mkdir -p $(PKG_BUILD_DIR)
    $(CP) ./src/* $(PKG_BUILD_DIR)/
endif

define Build/Compile
    $(MAKE) -C "$(LINUX_DIR)" \
        $(MAKE_OPTS) \
        modules
endif

$(eval $(call KernelPackage,f403tech_drv))
```

至于该 Makefile 文件中每行是什么意思，请查阅第 17 章，这里不再赘述。

## 18.2 配置编译驱动程序

接下来，我们就来配置编译驱动程序。首先将 f403tech\_drv 文件夹传到 OpenWrt 源码的 /home/openwrt/7620/openwrt/barrier\_breaker/package/kernel 目录下。然后进入 OpenWrt 源码的顶层目录，执行 make menuconfig。

```
# cd /home/openwrt/7620/openwrt/barrier_breaker
# make menuconfig
```

在弹出的菜单界面里，首先进入 Kernel modules 选项。

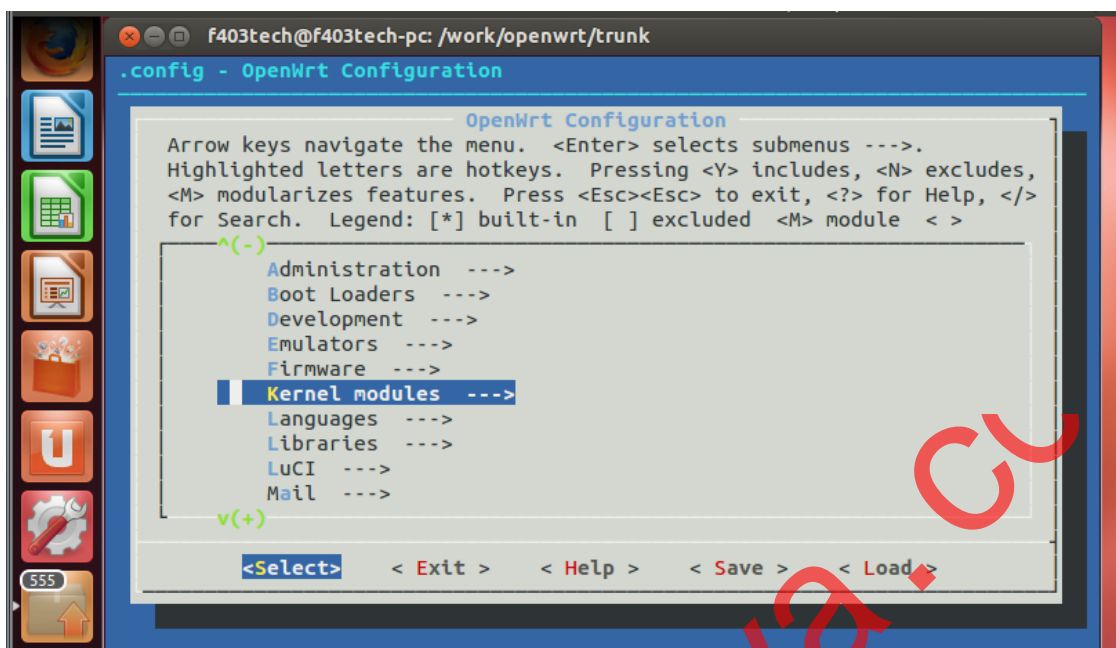


图 1

紧接着进入 Other modules 选项。

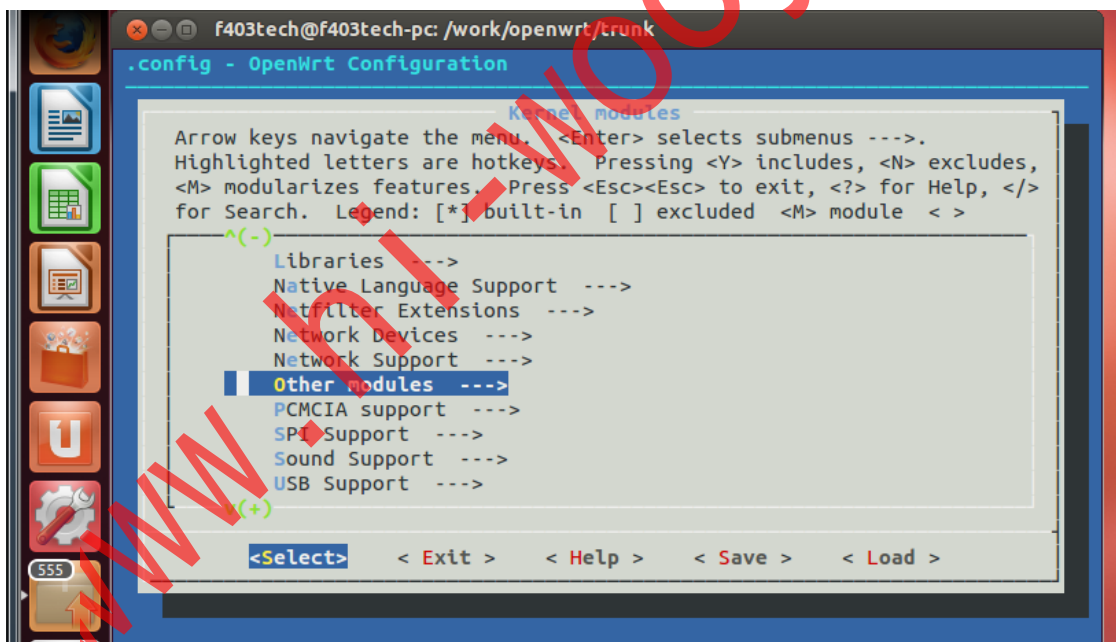


图 2

然后我们就能看到我们的驱动程序的选项了 kmod-f403tech\_drv, 将它配置成 y, 即\*。

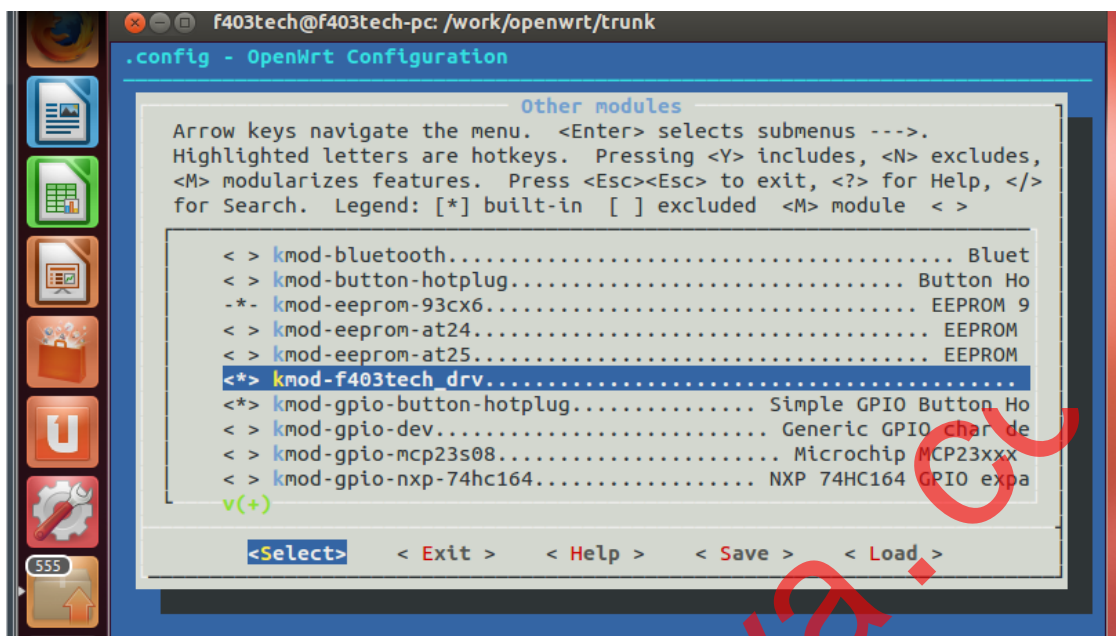


图 3

最后，退出保存。然后再执行编译命令。

```
# make package/kernel/f403tech_drv/compile V=99
```

编译完成以后，就能在 barrier\_breaker/bin/ramips/packages/base 目录下看到我们的内核模块驱动程序的软件包。

```
f403tech@f403tech-pc:/work/openwrt/trunk/bin/ramips/packages$ ls *f403*
kmod-f403tech_drv_3.10.36-1_ramips_24kec.ipk
f403tech@f403tech-pc:/work/openwrt/trunk/bin/ramips/packages$
```

图 4

### 18.3 动态的加载和卸载内核驱动模块软件包

通过前面的努力，我们终于得到了我们自己的内核驱动模块软件包了。接下来就来使用它。首先使用第 8 章中的方法，将 kmod-f403tech\_drv\_3.10.36-1\_ramips\_24kec.ipk 软件包传到开发板上面。

然后使用 opkg install 命令来安装软件包。

```
# opkg install kmod-f403tech_drv_3.10.36-1_ramips_24kec.ipk
```

使用 opkg list 命令来查看已经安装了哪些软件包。这里是否执行该命令都无所谓。

接下来进入我们的 /lib/modules/3.10.36 目录，就能看到我们的驱动程序模块了。

```
# cd lib/modules/3.10.36/
```

```
root@openwrt:/lib/modules/3.10.36#
root@openwrt:/lib/modules/3.10.36# ls
arc4.ko          ipt_REJECT.ko      rt2800lib.ko
asix.ko          iptable_filter.ko  rt2800mmio.ko
cdc-acm.ko       iptable_mangle.ko  rt2800soc.ko
cdc_ether.ko     iptable_nat.ko     rt2x00lib.ko
cfg80211.ko      iptable_raw.ko     rt2x00mmio.ko
compat.ko        ipv6.ko            rt2x00soc.ko
crc-ccitt.ko     leds-gpio.ko       slhc.ko
crc-itu-t.ko     mac80211.ko        usb-common.ko
crypto_blkcipher.ko mii.ko             usb_wwan.ko
eeprom_93cx6.ko  nf_conntrack.ko    usbcore.ko
ehci-hcd.ko      nf_conntrack_ftp.ko usbnet.ko
ehci-platform.ko nf_conntrack_ipv4.ko usbserial.ko
f403tech_drv.ko  nf_conntrack_ipv6.ko x_tables.ko
gpio_button_hotplug.ko nf_conntrack_irc.ko xt_CT.ko
ip6_tables.ko    nf_defrag_ipv4.ko  xt_LOG.ko
ip6t_REJECT.ko   nf_defrag_ipv6.ko  xt_REDIRECT.ko
ip6t_ah.ko       nf_nat.ko          xt_TCPMSS.ko
ip6t_eui64.ko    nf_nat_ftp.ko      xt_comment.ko
ip6t_frag.ko     nf_nat_ipv4.ko     xt_conntrack.ko
ip6t_hbh.ko      nf_nat_irc.ko      xt_limit.ko
ip6t_ipv6header.ko nls_base.ko        xt_mac.ko
ip6t_mh.ko       ohci-hcd.ko        xt_mark.ko
ip6t_rt.ko       option.ko           xt_multiport.ko
ip6table_filter.ko ppp_async.ko        xt_nat.ko
ip6table_mangle.ko ppp_generic.ko      xt_state.ko
ip6table_raw.ko  pppoe.ko            xt_tcpudp.ko
ip_tables.ko     pppox.ko            xt_time.ko
ipt_MASQUERADE.ko rndis_host.ko
root@openwrt:/lib/modules/3.10.36#
```

图 6

接下来就通过 insmod 命令来装载驱动模块。

```
# insmod f403tech_drv.ko
```

当我们一执行该命令，串口终端上面就会打印出。

```
[ 458.870000] f403tech_drv_init:Hello F403Tech
```

这行打印信息，正是我们在驱动程序的入口函数中添加的调试信息。

如果我们想卸载驱动模块，则使用 rmmod 命令。

```
# rmmod f403tech_drv
```

当执行完这条命令后，就成功的卸载了该驱动程序。我们也能看到如下打印。

```
[ 564.800000] f403tech_drv_exit:Hello F403Tech
```

这条打印，是在我们的驱动的出口函数中添加的调试信息。

注意：

- 1). 该教程为我司(<https://wy-wulian.taobao.com/>)原创教程，版权所有；
- 2). 该教程会不断更新、不断深入，详情请咨询我司客服；
- 3). 针对该教程，我们还有 QQ 群和论坛，专门负责技术答疑，详情请咨询我司客服。