

# 常见问题

MA35D1

江天文

05/09/2023

Joy of innovation  
**nuvoTon**

# | 在Buildroot外编译内核驱动模块

- 在Buildroot外编译内核驱动模块, 意味着脱离 (out-of-tree) Linux内核源码编译外部 (“external”) 模块.
- Linux内核使用kbuild构件编译模块, kbuild构件一般使用文件**Makefile**或**Kbuild**, 如果两者同时存在, 那么就使用Kbuild.
- 创建Kbuild文件, 假定模块的名字为mod\_xyz

```
obj-m := mod_xyz.o
mod_xyz-y := main.o

ccflags-y := -I$(src)/include
ccflags-y += -I$(src)
```

- 创建mod\_xyz\_defs.h → \$ touch include/mod\_xyz\_defs.h

# | 在Buildroot外编译内核驱动模块

- 创建Makefile

```
ifneq ($(KERNELRELEASE),)
include Kbuild

else
KERNELDIR ?= /lib/modules/`uname -r`/build

default:
$(MAKE) -C $(KERNELDIR) M=$$PWD

clean:
$(MAKE) -C $(KERNELDIR) M=$$PWD clean

endif
```

# | 在Buildroot外编译内核驱动模块

- 驱动主文件main.c
- 驱动源码目录结构

\$(src)

```
|____ Kbuild
|____ Makefile
|____ main.c
|____ include
      |____ mod_xyz_defs.h
```

- 驱动示范代码

[https://github.com/symfund/ma35d1-portal/tree/master/drivers/mod\\_xyz](https://github.com/symfund/ma35d1-portal/tree/master/drivers/mod_xyz)

```
#include <linux/init.h>
#include <linux/module.h>
#include <linux/printk.h>
#include <linux/kernel.h>
#include <linux/utsname.h>

#include "mod_xyz_defs.h"

static int __init mod_xyz_init(void)
{
    printk(KERN_INFO "Loading module mod_xyz...\n");
    pr_alert("%s version %s: %s\n", utsname()->sysname, utsname()->release, utsname()->version);
    return 0;
}

static void __exit mod_xyz_exit(void)
{
    pr_alert("exit mod_xyz driver\n");
}

module_init(mod_xyz_init);
module_exit(mod_xyz_exit);

MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("Linux Kernel Module mod_xyz");
MODULE_AUTHOR("2023 @nuvoton");
```

# | 在Buildroot外编译内核驱动模块

- 开始编译内核模块前, 先设置编译环境

**\$ source \${BR2\_DIR}/output/host/environment-setup**

- 编译

**\$ make**

- 清除

**\$ make clean**

## I 参考

- Linux Kernel Makefiles

<https://www.kernel.org/doc/html/latest/kbuild/makefiles.html>

- Building External Modules

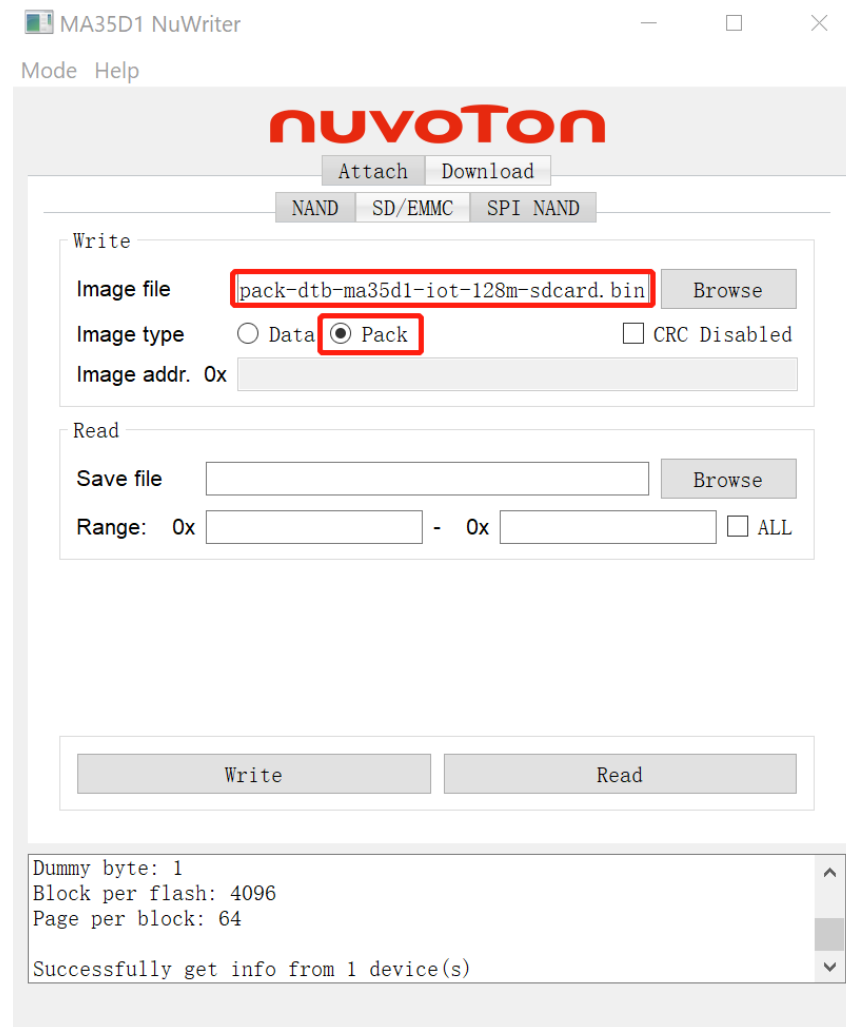
<https://www.kernel.org/doc/html/latest/kbuild/modules.html>

# 在设备树文件中打开或关闭设备后快速烧写设备树

- 在设备树文件中打开或关闭设备后, 不用重编Linux
- 修改设备树文件, 将某个设备打开(okay)或关闭(disabled)

**linux/arch/arm64/boot/dts/nuvoton/ma35d1-iot-128.dts**

- 获取rebuild-dts.sh, 放到  
workspace/scripts/rebuild-dts.sh  
<https://github.com/symfund/ma35d1-portal/blob/master/scripts/rebuild-dts.sh>
- 在Buildroot根目录下执行  
**\$ source workspace/scripts/rebuild-dts.sh**
- 生成Pack数据类型的DTB image文件:  
**pack-dtb-ma35d1-iot-128m-sdcard.bin**
- 在NuWriter中以**Pack**数据类型方式烧写该DTB image文件**pack-dtb-ma35d1-iot-128m-sdcard.bin**

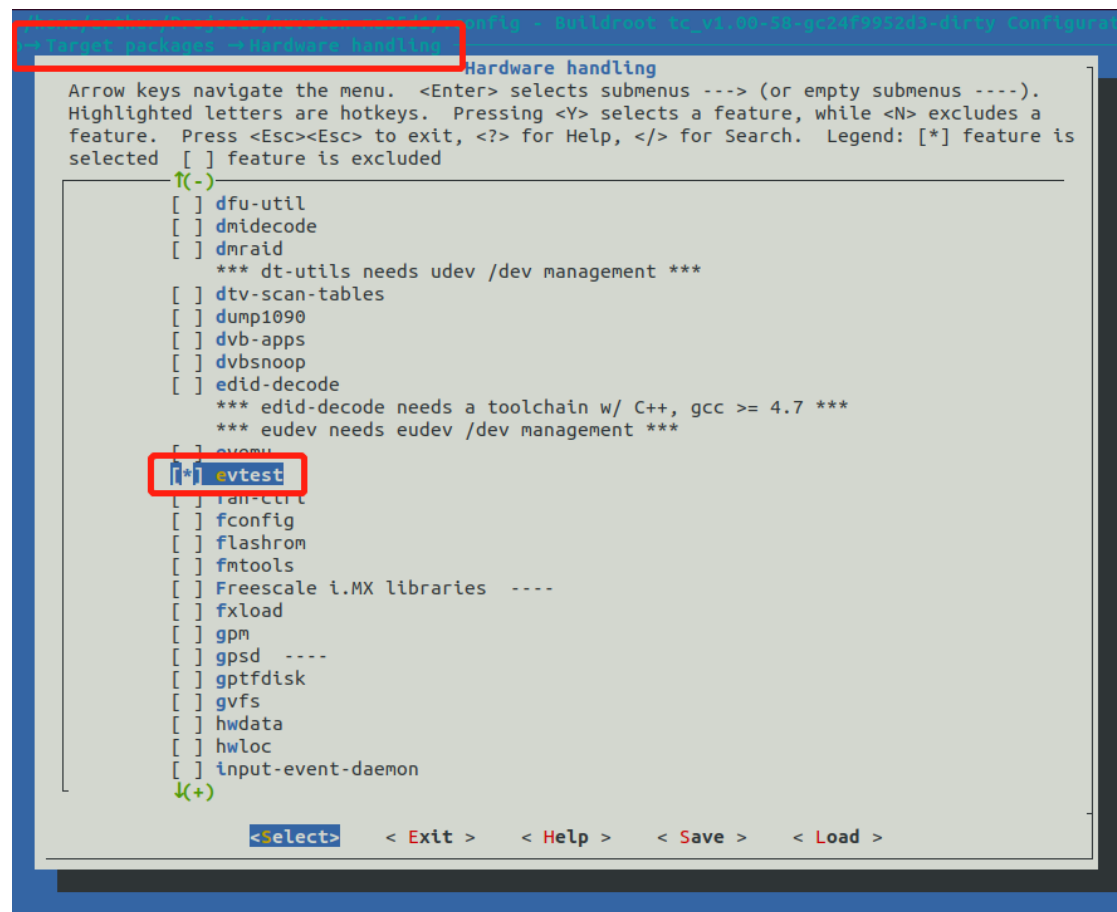


# | 测试触摸驱动 (/dev/input/event\*)

- 测试电容式(capacitive I2C接口)或电阻式(resistive ADC接口)触摸芯片(GT911)驱动时, 可在Buildroot里打开evtest软件包

→Target packages →Hardware handling  
[\*] evtest

- evtest还可以用来测试按键输入





*Joy of innovation*  
**nuvoTon**

谢谢

謝謝

Děkuji

Bedankt

Thank you

Kiitos

Merci

Danke

Grazie

ありがとう

감사합니다

Dziękujemy

Obrigado

Спасибо

Gracias

Teşekkür ederim

Cảm ơn