

Linux-RT 内核编译方法

Revision History

Draft Date	Revision No.	Description
2019/01/25	V1.0	1. 初始版本。

目 录

1 前言	3
2 Linux-RT-4.9.65 内核编译方法	3
2.1 安装 Linux-RT 内核源码	3
2.2 清理 Linux-RT 内核	4
2.3 配置 Linux-RT 内核	5
2.4 编译设备树文件	7
2.5 编译 Linux-RT 内核	9
3 extra 驱动编译	11
3.1 编译 extra 相关的驱动模块	12
3.2 安装 extra 相关的驱动	17
3.3 重新启动系统	20
更多帮助	21

1 前言

➤ 操作环境

VMware14.1.1;

Ubuntu 14.04.3 64bit

➤ Linux-RT-4.9.65 内核版本适用性说明

表 1

开发板型号	是否支持本实验
TL437x-EasyEVM	支持
TL437x-EVM	支持
TL437Xf-EVM	不支持

➤ 清理命令区别说明

make clean: 删除大多数的编译生成文件，但会保留配置文件。

make mrproper: 删除所有的编译生成文件，同时删除配置文件以及各种备份文件。

make distclean: 删除所有的编译生成文件，同时删除配置文件以及各种备份文件和补丁文件，清除最完整。

删除的文件范围从小到大依次为：make clean < make mrproper < make distclean。

2 Linux-RT-4.9.65 内核编译方法

2.1 安装 Linux-RT 内核源码

打开 Ubuntu，执行如下命令创建 Linux-RT 内核源码安装目录
"/home/tronlong/AM437x/kernel/Linux-RT-4.9.65"，如下图所示：

```
Host# mkdir -p /home/tronlong/AM437x/kernel/Linux-RT-4.9.65
```

```
tronlong@tronlong-virtual-machine: ~  
tronlong@tronlong-virtual-machine:~$ mkdir -p /home/tronlong/AM437x/kernel/Linux-RT-4.9.65  
tronlong@tronlong-virtual-machine:~$
```

图 1

Linux-RT 内核源码为光盘"Linux-RT\kernel\Linux-RT-4.9.65\src\linux-rt-4.9.65-[Git 系列号]-[版本号].tar.gz", Git 系列号与版本号以实际的为准。kernel 支持特性可以查看"\kernel\kernel-feature-support.xls"文件。

将内核源码压缩包复制到"/home/tronlong/AM437x"目录下, 再将其解压至安装目录, 执行命令如下:

```
Host# cd /home/tronlong/AM437x
```

```
Host# tar -xvf linux-rt-4.9.65-gdb43c01-v1.0.tar.gz -C kernel/Linux-RT-4.9.65/
```

```
tronlong@tronlong-virtual-machine:~$ cd /home/tronlong/AM437x  
tronlong@tronlong-virtual-machine:~/AM437x$ ls linux-rt-4.9.65-gdb43c01-v1.0.tar.gz  
linux-rt-4.9.65-gdb43c01-v1.0.tar.gz  
tronlong@tronlong-virtual-machine:~/AM437x$ tar -xvf linux-rt-4.9.65-gdb43c01-v1.0.tar.gz -C kernel/Linux-RT-4.9.65/
```

图 2

2.2 清理 Linux-RT 内核

确保已配置为 V04.03.00.05 版本 Linux-RT Processor-SDK 交叉编译工具链后, 进入 Linux-RT 内核源码安装目录, 执行 Linux-RT 内核清理命令。

```
Host# cd kernel/Linux-RT-4.9.65/
```

```
Host# make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- distclean
```

```
tronlong@tronlong-virtual-machine:~/AM437x$ cd kernel/Linux-RT-4.9.65/  
tronlong@tronlong-virtual-machine:~/AM437x/kernel/Linux-RT-4.9.65$ make ARCH=arm  
CROSS_COMPILE=arm-linux-gnueabi- distclean  
tronlong@tronlong-virtual-machine:~/AM437x/kernel/Linux-RT-4.9.65$
```

图 3

2.3 配置 Linux-RT 内核

在 Linux-RT 内核源码安装目录下，执行如下命令配置 Linux-RT 内核：

```
Host# make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- tisd_k_am437x-evm-rt_defconfig
```

```
tronlong@tronlong-virtual-machine:~/AM437x/kernel/Linux-RT-4.9.65$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- tisd_k_am437x-evm-rt_defconfig
HOSTCC scripts/basic/fixdep
HOSTCC scripts/kconfig/conf.o
SHIPPED scripts/kconfig/zconf.tab.c
SHIPPED scripts/kconfig/zconf.lex.c
SHIPPED scripts/kconfig/zconf.hash.c
HOSTCC scripts/kconfig/zconf.tab.o
HOSTLD scripts/kconfig/conf
#
# configuration written to .config
#
tronlong@tronlong-virtual-machine:~/AM437x/kernel/Linux-RT-4.9.65$
```

图 4

可以通过 menuconfig 命令，启动图形界面修改配置。如果不需要，则可跳过此步骤。

执行 menuconfig 命令前，请先执行如下命令安装图形依赖库：

```
Host# sudo apt-get install libncurses5-dev
```

```
tronlong@tronlong-virtual-machine:~/AM437x/kernel/Linux-RT-4.9.65$ sudo apt-get install libncurses5-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libtinfo-dev
Suggested packages:
  ncurses-doc
The following NEW packages will be installed:
  libncurses5-dev libtinfo-dev
0 upgraded, 2 newly installed, 0 to remove and 529 not upgraded.
Need to get 246 kB of archives.
After this operation, 1,479 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

图 5

输入 Y，等待安装完成。

```
Do you want to continue? [Y/n] Y
Selecting previously unselected package libtinfo-dev:amd64.
(Reading database ... 182734 files and directories currently installed.)
Preparing to unpack .../libtinfo-dev_5.9+20140118-1ubuntu1_amd64.deb ...
Unpacking libtinfo-dev:amd64 (5.9+20140118-1ubuntu1) ...
Selecting previously unselected package libncurses5-dev:amd64.
Preparing to unpack .../libncurses5-dev_5.9+20140118-1ubuntu1_amd64.deb ...
Unpacking libncurses5-dev:amd64 (5.9+20140118-1ubuntu1) ...
Setting up libtinfo-dev:amd64 (5.9+20140118-1ubuntu1) ...
Setting up libncurses5-dev:amd64 (5.9+20140118-1ubuntu1) ...
tronlong@tronlong-virtual-machine:~/AM437x/kernel/Linux-RT-4.9.65$
```

图 6

安装依赖库后，执行如下命令启动 menuconfig 配置界面，如下图所示：

Host# make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- menuconfig

```
tronlong@tronlong-virtual-machine:~/AM437x/kernel/Linux-RT-4.9.65$ make ARCH=arm
CROSS_COMPILE=arm-linux-gnueabi- menuconfig
HOSTCC scripts/kconfig/mconf.o
HOSTCC scripts/kconfig/lxdialog/checklist.o
HOSTCC scripts/kconfig/lxdialog/util.o
HOSTCC scripts/kconfig/lxdialog/inputbox.o
HOSTCC scripts/kconfig/lxdialog/textbox.o
HOSTCC scripts/kconfig/lxdialog/yesno.o
HOSTCC scripts/kconfig/lxdialog/menubox.o
HOSTLD scripts/kconfig/mconf
scripts/kconfig/mconf Kconfig
```

图 7

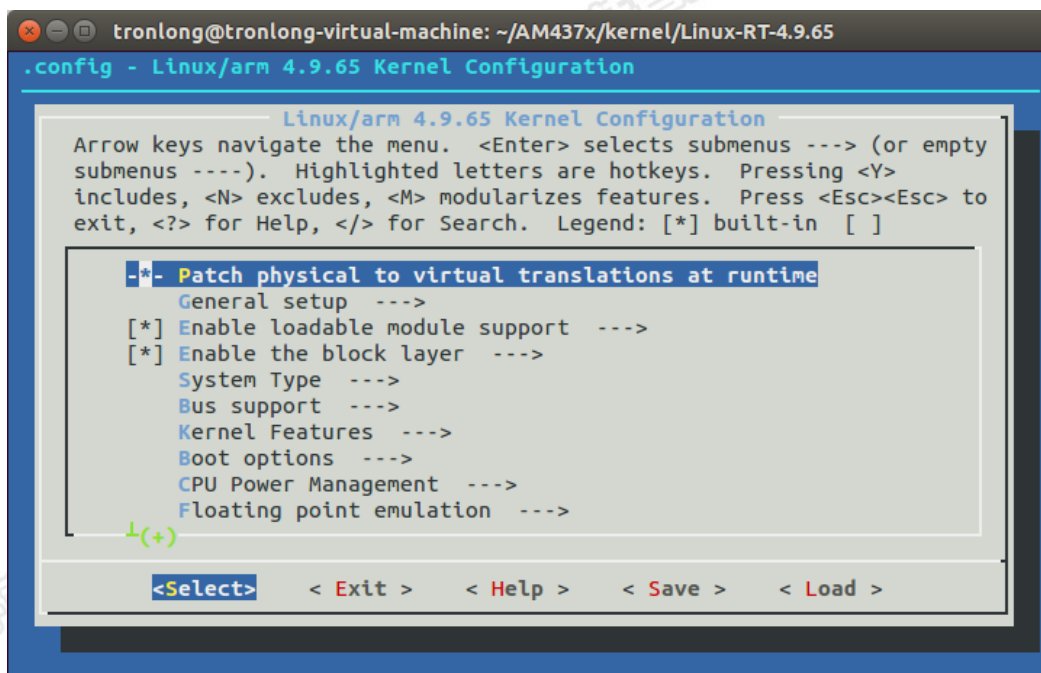


图 8

使用键盘上下键和空格键进行选择，每个选项前都会有一个括号供用户选择，选择项为空表示不选中此选项，" * "表示选中此选项并编入内核，" M "表示选中此选项并编译成模块。修改配置后，点击"< Save >"保存配置，并点击"< Exit >"退出。

2.4 编译设备树文件

设备树文件现加载模式都是采用基础设备树文件与动态设备树文件配合。基础设备树文件主要描述各个开发板的基础硬件设备，例如 LED 灯，按键等；动态设备树文件主要描述各个开发板特有的硬件设备，例如 TL437x-EVM 开发板的音频接口，TL437x-EasyEVM 的 TTL UART1、2、5。

系统上电启动后默认加载基础设备树文件，动态设备树文件需要进入文件系统后手动加载。

➤ 编译基础设备树文件

执行以下命令编译基础设备树文件 tl437x-evm-common.dtb，系统上电启动后，默认自动加载 tl437x-evm-common.dtb。

Host# make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- tl437x-evm-common.dtb

```
tronlong@tronlong-virtual-machine:~/AM437x/kernel/Linux-RT-4.9.65$ make ARCH=arm
CROSS_COMPILE=arm-linux-gnueabi- tl437x-evm-common.dtb
scripts/kconfig/conf --silentoldconfig Kconfig
HOSTCC scripts/basic/bin2c
WRAP arch/arm/include/generated/asm/bitperlong.h
WRAP arch/arm/include/generated/asm/clkdev.h
WRAP arch/arm/include/generated/asm/cputime.h
WRAP arch/arm/include/generated/asm/current.h
WRAP arch/arm/include/generated/asm/early_ioremap.h
WRAP arch/arm/include/generated/asm/emergency-restart.h
WRAP arch/arm/include/generated/asm/errno.h
WRAP arch/arm/include/generated/asm/exec.h
WRAP arch/arm/include/generated/asm/ioctl.h
WRAP arch/arm/include/generated/asm/ipcbuf.h
```

图 9

```
CC      scripts/mod/devicetable-offsets.s
GEN      scripts/mod/devicetable-offsets.h
HOSTCC   scripts/mod/file2alias.o
HOSTCC   scripts/mod/sumversion.o
HOSTLD   scripts/mod/modpost
HOSTCC   scripts/kallsyms
HOSTCC   scripts/conmakehash
HOSTCC   scripts/sortextable
DTC      arch/arm/boot/dts/tl437x-evm-common.dtb
tronlong@tronlong-virtual-machine:~/AM43/x/kernel/Linux-RT-4.9.65$
```

图 10

编译完成后，会在内核源码目录"arch/arm/boot/dts"路径下生成 tl437x-evm-common.dtb 文件，将其拷贝到文件系统分区 rootfs 的"/boot"目录下，开发板上电后将自动加载。

➤ 编译动态设备树文件

AM437x 平台各型号开发板与动态设备树文件.dtb 的对应关系如下表。

表 2

开发板型号	对应.dtb 设备树文件
TL437x-EasyEVM	tl437x-easyevm.dtb
TL437x-EVM	tl437x-evm.dtb

以 TL437x-EVM 开发板为例，编译开发板的动态设备树文件。如需编译 TL437x-EasyEVM 开发板的动态设备树文件，将指令中的关键字"tl437x-evm"改成"tl437x-easyevm"。

执行以下指令，对 TL437x-EVM 开发板设备树文件进行预编译。

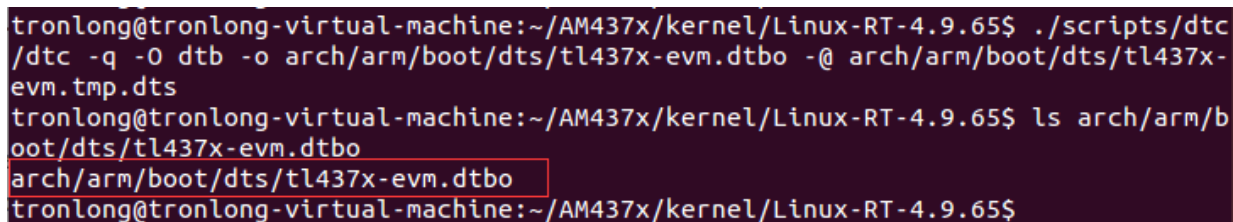
```
Host# cpp -nostdinc -I include -undef -x assembler-with-cpp arch/arm/boot/dts/tl437x-evm.dts > arch/arm/boot/dts/tl437x-evm.tmp.dts
```

```
tronlong@tronlong-virtual-machine:~/AM437x/kernel/Linux-RT-4.9.65$ cpp -nostdinc -I include -undef -x assembler-with-cpp arch/arm/boot/dts/tl437x-evm.dts > arch/arm/boot/dts/tl437x-evm.tmp.dts
tronlong@tronlong-virtual-machine:~/AM437x/kernel/Linux-RT-4.9.65$
```

图 11

执行以下指令，编译 TL437x-EVM 开发板动态设备树文件。

```
Host# ./scripts/dtc/dtc -q -O dtb -o arch/arm/boot/dts/tl437x-evm.dtbo -@  
arch/arm/boot/dts/tl437x-evm.tmp.dts
```



```
tronlong@tronlong-virtual-machine:~/AM437x/kernel/Linux-RT-4.9.65$ ./scripts/dtc  
/dtc -q -O dtb -o arch/arm/boot/dts/tl437x-evm.dtbo -@ arch/arm/boot/dts/tl437x-  
evm.tmp.dts  
tronlong@tronlong-virtual-machine:~/AM437x/kernel/Linux-RT-4.9.65$ ls arch/arm/b  
oot/dts/tl437x-evm.dtbo  
arch/arm/boot/dts/tl437x-evm.dtbo  
tronlong@tronlong-virtual-machine:~/AM437x/kernel/Linux-RT-4.9.65$
```

图 12

编译完成后，会在内核源码目录"arch/arm/boot/dts"路径下生成 tl437x-evm.dtbo 文件，将其拷贝到文件系统分区 rootfs 的"/lib/firmware/"目录下，开发板上电后需要手动设置加载此设备树文件。

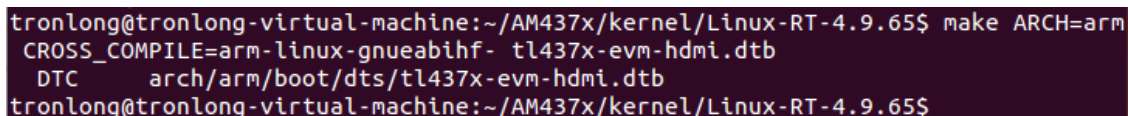
➤ 编译 HDMI 设备树文件

由于 tl437x-evm-common.dtb 设备树文件默认配置的是 LCD 显示屏，如需使用 HDMI 显示屏则需重新编配置设备树文件。

执行以下命令，编译 tl437x-evm-hdmi.dtb 设备树文件。

```
Host# make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- tl437x-evm-hdmi.dtb
```

将编译出来的 tl437x-evm-hdmi.dtb 重命名为 tl437x-evm-common.dtb，替换开发板文件系统 rootfs 分区"/boot"目录下原来的设备树文件，然后 reboot 重启。重启后 HDMI 屏即可正常使用。



```
tronlong@tronlong-virtual-machine:~/AM437x/kernel/Linux-RT-4.9.65$ make ARCH=arm  
CROSS_COMPILE=arm-linux-gnueabi- tl437x-evm-hdmi.dtb  
DTC arch/arm/boot/dts/tl437x-evm-hdmi.dtb  
tronlong@tronlong-virtual-machine:~/AM437x/kernel/Linux-RT-4.9.65$
```

图 13

2.5 编译 Linux-RT 内核

在 Linux-RT 内核源码安装目录下，执行如下命令编译 Linux-RT 内核：

创龙

Host# make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- zImage -j 4

“-j 4”是一个编译选项，告诉操作系统用 4 个线程去编译，加快编译速度。

```
tronlong@tronlong-virtual-machine:~/AM437x/kernel/Linux-RT-4.9.65$ make ARCH=arm
CROSS_COMPILE=arm-linux-gnueabihf- zImage -j 4
CHK      include/config/kernel.release
CHK      include/generated/uapi/linux/version.h
UPD      include/generated/uapi/linux/version.h
UPD      include/config/kernel.release
CHK      include/generated/utsrelease.h
UPD      include/generated/utsrelease.h
GEN      include/generated/mach-types.h
CC      kernel/bounds.s
```

图 14

第一次编译内核耗时较长，大约需要 5min。编译完成如下图所示：

```
AS      arch/arm/boot/compressed/piggy.o
LD      arch/arm/boot/compressed/vmlinux
OBJCOPY arch/arm/boot/zImage
Kernel: arch/arm/boot/zImage is ready
tronlong@tronlong-virtual-machine:~/AM437x/kernel/Linux-RT-4.9.65$
```

图 15

编译完成后，会在 Linux-RT 内核源码安装目录"arch/arm/boot"路径下生成内核镜像文件 zImage。可将编译出来的内核镜像文件，替换开发板文件系统中的内核镜像文件。

使用 SD 系统启动卡启动系统时，如需替换内核镜像，有如下方法：

- 使用新的内核镜像文件，保持文件名与原文件一致，替换 SD 系统启动卡 rootfs 分区 "/boot" 目录下的对应文件。
- 使用新的内核镜像文件，保持文件名与原文件一致，替换 SD 系统启动卡制作文件目录下的对应文件，然后重新制作 SD 系统启动卡。

2.6 编译模块 modules

在 Linux-RT 内核源码安装目录下，执行如下指令编译内核配置中选中的模块：

Host# make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- modules -j 4

```
tronlong@tronlong-virtual-machine:~/AM437x/kernel/Linux-RT-4.9.65$ make ARCH=arm
CROSS_COMPILE=arm-linux-gnueabihf- modules -j 4
CHK include/config/kernel.release
CHK include/generated/uapi/linux/version.h
CHK include/generated/utsrelease.h
CHK include/generated/timeconst.h
CHK include/generated/bounds.h
CHK include/generated/asm-offsets.h
CALL scripts/checksyscalls.sh
```

图 16

将 SD 系统启动卡插入 PC 机，并将其成功挂载到 Ubuntu。执行如下命令，将编译的模块安装到系统卡 rootfs 分区，"/media/tronlong/rootfs/"为 SD 卡文件系统在 Ubuntu 的挂载路径。

Host# sudo make ARCH=arm modules_install INSTALL_MOD_PATH=/media/tronlong/rootfs/

```
tronlong@tronlong-virtual-machine:~/AM437x/kernel/Linux-RT-4.9.65$ sudo make ARCH=arm modules_install INSTALL_MOD_PATH=/media/tronlong/rootfs/
INSTALL arch/arm/crypto/aes-arm-bs.ko
INSTALL arch/arm/crypto/aes-arm-ce.ko
INSTALL arch/arm/crypto/aes-arm.ko
INSTALL arch/arm/crypto/ghash-arm-ce.ko
INSTALL arch/arm/crypto/sha1-arm-ce.ko
INSTALL arch/arm/crypto/sha1-arm-neon.ko
INSTALL arch/arm/crypto/sha1-arm.ko
INSTALL arch/arm/crypto/sha2-arm-ce.ko
INSTALL arch/arm/crypto/sha256-arm.ko
INSTALL arch/arm/crypto/sha512-arm.ko
INSTALL crypto/ablk_helper.ko
```

图 17

```
INSTALL sound/soc/codecs/snd-soc-sti-sas.ko
INSTALL sound/soc/codecs/snd-soc-tlv320aic31xx.ko
INSTALL sound/soc/codecs/snd-soc-tlv320aic3x.ko
INSTALL sound/soc/codecs/snd-soc-wm8978.ko
INSTALL sound/soc/fsl/snd-soc-fsl-sai.ko
INSTALL sound/soc/generic/snd-soc-simple-card-utils.ko
INSTALL sound/soc/generic/snd-soc-simple-card.ko
INSTALL sound/soc/omap/snd-soc-omap-hdmi-audio.ko
INSTALL sound/usb/snd-usb-audio.ko
INSTALL sound/usb/snd-usbmidi-lib.ko
DEPMOD 4.9.65-rt23
tronlong@tronlong-virtual-machine:~/AM437x/kernel/Linux-RT-4.9.65$
```

图 18

3 extra 驱动编译

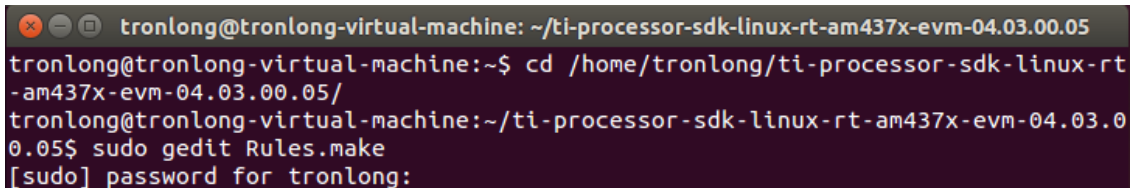
对于重新配置、编译过的内核，其版本号可能和做卡工具中的内核版本号不一致，此时会因为文件系统上的驱动模块和内核版本不一致而导致驱动模块无法安装，从而造成某些功能不正常（主要是显示相关的）。因此编译过的内核，应按照如下步骤操作，避免上述问题出现。

3.1 编译 extra 相关的驱动模块

进入 Linux-RT Processor-SDK 包安装目录（这部分的驱动模块在内核是无法生成的，驱动源码位于 Linux-RT Processor-SDK 包中），执行如下命令打开 Rules.make 文件：

```
Host# cd /home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/
```

```
Host# sudo gedit Rules.make
```



```
tronlong@tronlong-virtual-machine: ~/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05
tronlong@tronlong-virtual-machine:~$ cd /home/tronlong/ti-processor-sdk-linux-rt-
-am437x-evm-04.03.00.05/
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am437x-evm-04.03.0
0.05$ sudo gedit Rules.make
[sudo] password for tronlong:
```

图 19

将打开的 Rules.make 文件按如下修改，如下图所示：

```
DESTDIR=/media/tronlong/rootfs/ //修改为文件系统所在目录
```

```
LINUXKERNEL_INSTALL_DIR=/home/tronlong/AM437x/kernel/Linux-RT-4.9.65/ //修改
为内核所在目录
```

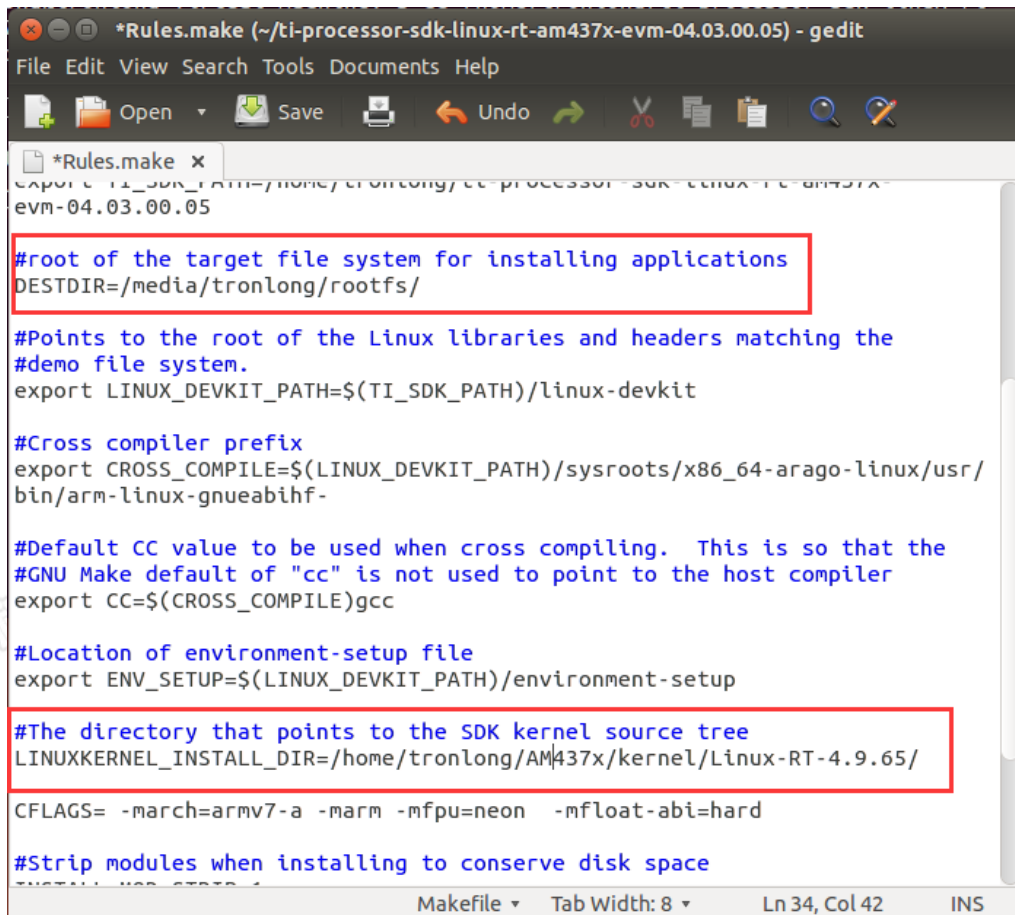


图 20

执行如下命令打开 Makefile 文件，将 cmem-mod、cryptodev、uio-module-driv、ti-sgx-ddk-km 驱动所在行的“linux”字符串删除，这样在重新编译这些驱动时，就不会再次编译内核。修改完成后保存，分别如下图所示：

Host# sudo gedit Makefile

```
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05$ pwd
/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05$ sudo gedit Makefile
```

图 21

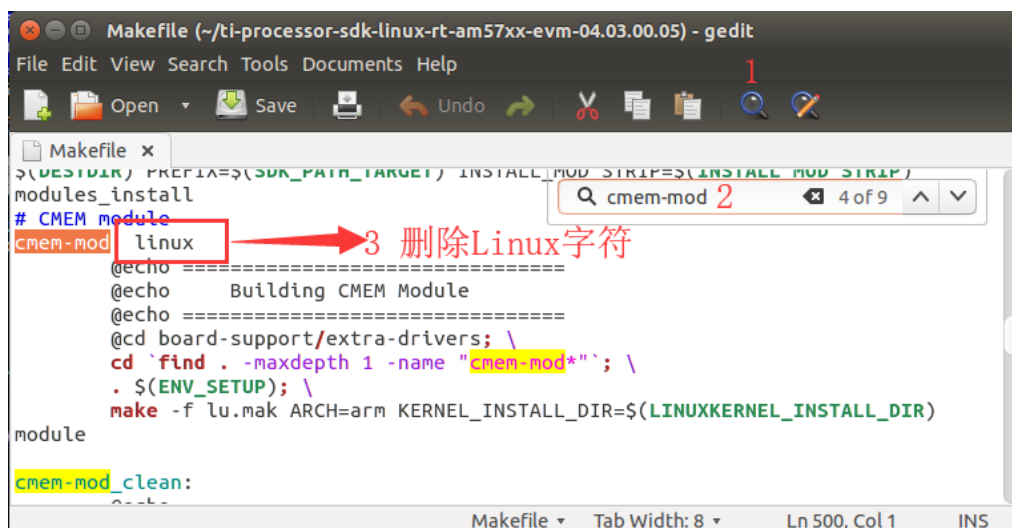


图 22

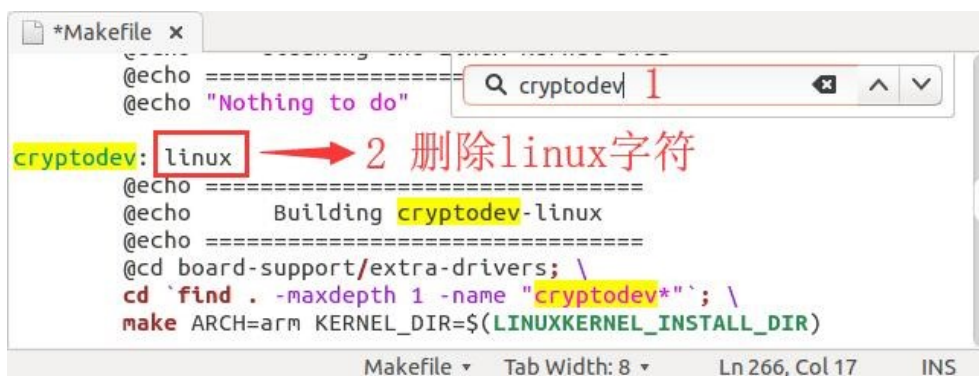


图 23

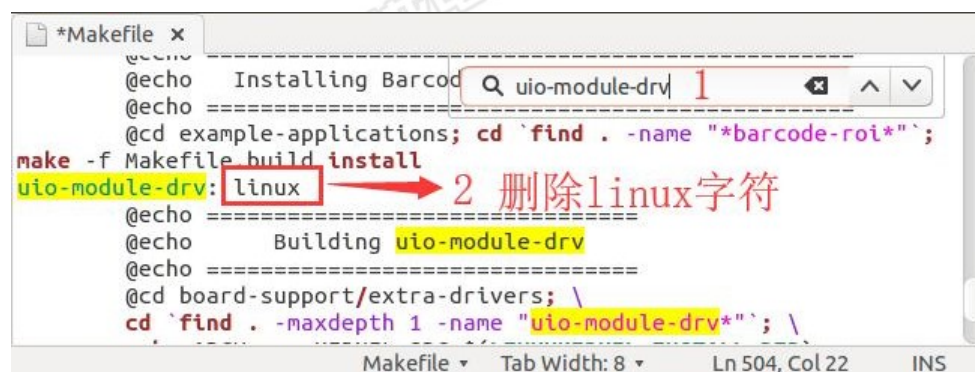


图 24

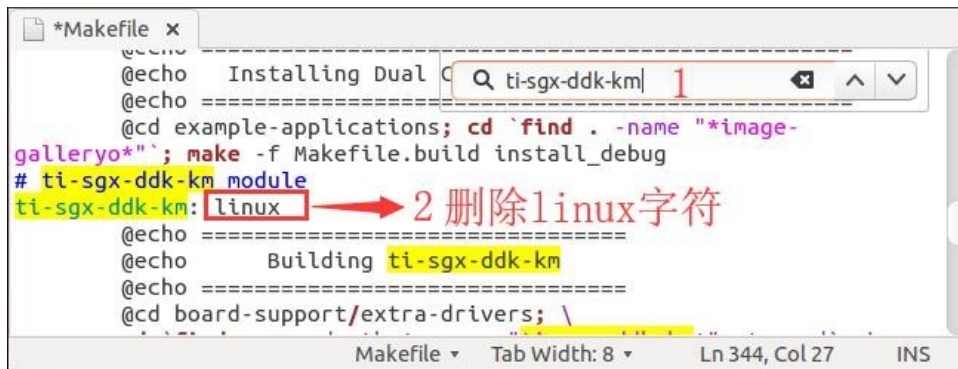


图 25

在 Linux-RT Processor-SDK 包安装目录下，依次执行以下命令编译以上相关的驱动：

Host# sudo make cmem-mod

```
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05$ sudo make cmem-mod
=====
Building CMEM Module
=====
make[1]: Entering directory `/home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/cmem-mod-4.14.01.00+gitAUTOINC+b687f3c365'
make -C src/cmem/module ARCH=arm
make[2]: Entering directory `/home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/cmem-mod-4.14.01.00+gitAUTOINC+b687f3c365/src/cmem/module'
Making module release...
make -C /home/tronlong/AM437x/kernel/Linux-RT-4.9.65/ M=`pwd` ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- \
    EXTRA_CFLAGS="-I/home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/cmem-mod-4.14.01.00+gitAUTOINC+b687f3c365/include" modules
make[3]: Entering directory `/home/tronlong/AM437x/kernel/Linux-RT-4.9.65'
CC [M] /home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/cmem-mod-4.14.01.00+gitAUTOINC+b687f3c365/src/cmem/module/cmemk.o
```

图 26

Host# sudo make cryptodev

```
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05
$ sudo make cryptodev
=====
Building cryptodev-linux
=====
make[1]: Entering directory `/home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/cryptodev-module-1.8'
make -C /home/tronlong/AM437x/kernel/Linux-RT-4.9.65/ SUBDIRS=`pwd` ARCH=arm CROSS_COMPILE=/home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/linux-devkit/sysroots/x86_64-arago-linux/usr/bin/arm-linux-gnueabihf- modules
make[2]: Entering directory `/home/tronlong/AM437x/kernel/Linux-RT-4.9.65'
CC [M] /home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/cryptodev-module-1.8/ioctl.o
CC [M] /home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/cryptodev-module-1.8/main.o
CC [M] /home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/cryptodev-module-1.8/cryptlib.o
CC [M] /home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/cryptodev-module-1.8/authenc.o
CC [M] /home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/cryptodev-module-1.8/zc.o
CC [M] /home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/cryptodev-module-1.8/util.o
LD [M] /home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/cryptodev-module-1.8/cryptodev.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/cryptodev-module-1.8/cryptodev.mod.o
```

图 27

Host# sudo make uio-module-drv

```
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05$ sudo make uio-module-drv
=====
Building uio-module-drv
=====
make[1]: Entering directory `/home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/uio-module-drv-2.2.1.0+gitAUTOINC+bda9260f22'
make -C /home/tronlong/AM437x/kernel/Linux-RT-4.9.65/ M=/home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/uio-module-drv-2.2.1.0+gitAUTOINC+bda9260f22 modules
make[2]: Entering directory `/home/tronlong/AM437x/kernel/Linux-RT-4.9.65'
CC [M] /home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/uio-module-drv-2.2.1.0+gitAUTOINC+bda9260f22/uio_module_drv.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/uio-module-drv-2.2.1.0+gitAUTOINC+bda9260f22/uio_module_drv.mod.o
LD [M] /home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/uio-module-drv-2.2.1.0+gitAUTOINC+bda9260f22/uio_module_drv.ko
make[2]: Leaving directory `/home/tronlong/AM437x/kernel/Linux-RT-4.9.65'
make[1]: Leaving directory `/home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/uio-module-drv-2.2.1.0+gitAUTOINC+bda9260f22'
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05$
```

图 28

Host# sudo make ti-sgx-ddk-km

```
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05$ sudo make ti-sgx-ddk-km
=====
Building ti-sgx-ddk-km
=====
make[1]: Entering directory `/home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/ti-sgx-ddk-km-1.14.3699939/eurasia_km/eurasiacon/build/linux2/omap_linux'
LD      /home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/ti-sgx-ddk-km-1.14.3699939/eurasia_km/eurasiacon/binary2_omap_linux_release/target/kbuild/built-in.o
CC [M]  /home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/ti-sgx-ddk-km-1.14.3699939/eurasia_km/eurasiacon/binary2_omap_linux_release/target/kbuild/services4/srvkm/env/linux/osfunc.o
CC [M]  /home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/ti-sgx-ddk-km-1.14.3699939/eurasia_km/eurasiacon/binary2_omap_linux_release/target/kbuild/services4/srvkm/env/linux/mutibits.o
make[1]: Leaving directory `/home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/ti-sgx-ddk-km-1.14.3699939/eurasia_km/eurasiacon/build/linux2/omap_linux'
```

图 29

3.2 安装 extra 相关的驱动

将 SD 系统启动卡插入 PC 机 USB 端口，并挂载到 Ubuntu。在 Linux-RT Processor-SDK 包安装目录下，依次执行以下命令，将 extra 相关的驱动安装到 SD 启动卡文件系统，默认的安装目录为"/lib/modules/<kernel_release>/extra/"。

Host# sudo make cmem-mod_install

Host# sudo make cryptodev_install

Host# sudo make uio-module-driv_install

Host# sudo make ti-sgx-ddk-km_install

```
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05$ sudo make cmem-mod_install
=====
Installing CMEM Module
=====
make[1]: Entering directory `/home/tronlong/AM437x/kernel/Linux-RT-4.9.65'
INSTALL /home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/cmem-mod-4.14.01.00+gitAUTOINC+b687f3c365/src/cmem/module/cmek.ko
DEPMOD 4.9.65-rt23
make[1]: Leaving directory `/home/tronlong/AM437x/kernel/Linux-RT-4.9.65'
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05$ sudo make cryptodev_install
=====
Installing cryptodev-linux
=====
make[1]: Entering directory `/home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/cryptodev-module-1.8'
make -C /home/tronlong/AM437x/kernel/Linux-RT-4.9.65/ SUBDIRS=`pwd` modules_install
make[2]: Entering directory `/home/tronlong/AM437x/kernel/Linux-RT-4.9.65'
INSTALL /home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/cryptodev-module-1.8/cryptodev.ko
DEPMOD 4.9.65-rt23
make[2]: Leaving directory `/home/tronlong/AM437x/kernel/Linux-RT-4.9.65'
make[1]: Leaving directory `/home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/cryptodev-module-1.8'
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05$
```

图 30


```
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05$ sudo make uio-module-driv_install
=====
Installing uio-module-driv
=====
make[1]: Entering directory `/home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/uio-module-driv-2.2.1.0+gitAUTOINC+bda9260f22'
make -C /home/tronlong/AM437x/kernel/Linux-RT-4.9.65/ M=/home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/uio-module-driv-2.2.1.0+gitAUTOINC+bda9260f22 modules_install
make[2]: Entering directory `/home/tronlong/AM437x/kernel/Linux-RT-4.9.65'
INSTALL /home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/uio-module-driv-2.2.1.0+gitAUTOINC+bda9260f22/uio_module_drv.ko
DEPMOD 4.9.65-rt23
make[2]: Leaving directory `/home/tronlong/AM437x/kernel/Linux-RT-4.9.65'
make[1]: Leaving directory `/home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/uio-module-driv-2.2.1.0+gitAUTOINC+bda9260f22'
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05$ sudo make ti-sgx-ddk-km_install
=====
Installing ti-sgx-ddk-km
=====
make[1]: Entering directory `/home/tronlong/AM437x/kernel/Linux-RT-4.9.65'
INSTALL /home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/ti-sgx-ddk-km-1.14.3699939/eurasia_km/eurasiacon/binary2_omap_linux_release/target/kbuild/bc_example.ko
INSTALL /home/tronlong/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05/board-support/extra-drivers/ti-sgx-ddk-km-1.14.3699939/eurasia_km/eurasiacon/binary2_omap_linux_release/target/kbuild/pvrsrvkm.ko
DEPMOD 4.9.65-rt23
make[1]: Leaving directory `/home/tronlong/AM437x/kernel/Linux-RT-4.9.65'
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05$
```

图 31

执行如下命令，查看驱动是否安装成功到 SD 卡文件系统"/lib/modules/<kernel_release>/extra/"目录下：

Host# ls /media/tronlong/rootfs/lib/modules/4.9.65-rt23/extra/

```
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05$ ls /media/tronlong/rootfs/lib/modules/4.9.65-rt23/extra/
bc_example.ko cmemk.ko cryptodev.ko pvrsrvkm.ko uio_module_drv.ko
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am437x-evm-04.03.00.05$
```

图 32

3.3 重新启动系统

将重新安装 extra 驱动的 SD 系统启动卡插入开发板，上电启动开发板，此时可见 LCD 显示屏正常显示并能进入 Matrix 界面。开发板文件系统下执行如下指令，查看安装的驱动模块：

Host# lsmod

```
root@AM437x-Tronlong:~# lsmod
Module                Size  Used by
sha512_generic        10031  0
sha512_arm            12316  0
sha256_generic        10178  0
sha1_generic          2928  0
sha1_arm_neon         6325  0
sha1_arm              3926  1 sha1_arm_neon
md5                   2219  0
cbc                   2388  0
xhci_plat_hcd         5991  0
xhci_hcd              108430 1 xhci_plat_hcd
pru_rproc             10682  0
pruss_intc            7249  1 pru_rproc
usbcore               198592 2 xhci_plat_hcd,xhci_hcd
dwc3                  70762  0
pruss                 9725  1 pru_rproc
udc_core              19338  1 dwc3
usb_common             4737  3 udc_core,usbcore,dwc3
ti_am335x_adc         6672  0
pm33xx                4348  0
omap_des              11036  0
pvrsrvkm              403455 0
des_generic           17608  1 omap_des
omap_aes_driver       19021  0
pruss_soc_bus         3649  0
omap_sham             21671  0
crypto_engine         6466  2 omap_des,omap_aes_driver
ti_emif_sram          5406  1 pm33xx
dwc3_omap             4673  0
extcon_core           16349  2 dwc3_omap,dwc3
ads7846               12894  0
wkup_m3_ipc           8077  1 pm33xx
wkup_m3_rproc         3670  1
omap_wdt              4719  0
remoteproc            28648 3 pru_rproc,wkup_m3_rproc,wkup_m3_ipc
ti_am335x_tscadc       5875  1 ti_am335x_adc
sch_fq_codel          8911  1
cryptodev             38385  1
cmemk                 35005  0
root@AM437x-Tronlong:~#
```

图 33

更多帮助

销售邮箱: sales@tronlong.com

技术邮箱: support@tronlong.com

创龙总机: 020-8998-6280

技术热线: 020-3893-9734

创龙官网: www.tronlong.com

技术论坛: www.51ele.net

线上商城: <https://tronlong.taobao.com>