Linux-RT 内核编译方法

Revision History

Draft Date	Revision No.	Description
2019/03/21	V1.1	1. 内容更新，兼容 TL570x-EVM 平台； 2. 修改“编译设备树文件”章节。
2018/07/01	V1.0	1. 初始版本。

目　　录

1 实验说明

■ 操作环境：Windows 7/10 64bit；VMware14.1.1；Ubuntu 14.04.3 64bit。

默认情况下，广州创龙为出厂的开发板都提供了一套完整的 Linux 系统，用户在项目评估阶段可直接使用创龙提供的 Linux 系统进行评估。到了项目的实际开发过程中，如需根据项目情况对 Linux 内核进行修改和配置，可参照本文档操作。

Linux-RT-4.9.65 内核版本适用性说明如下：

表 1

开发板型号	是否支持本实验
TL570x-EVM	支持
TL5728-EasyEVM	支持
TL5728-IDK	支持
TL5728F-EVM	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。

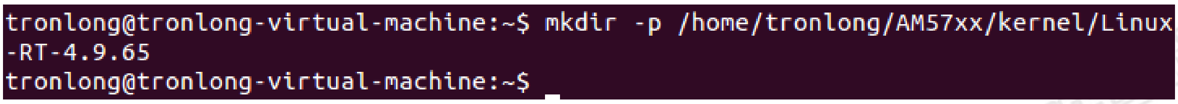
删除的文件范围从小到大依次为：make clean < make mrproper < make distclean。

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，在 Ubuntu 下执行如下命令新建 Linux-RT 内核源码安装目录"/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65"，如下图所示：

```
Host#    mkdir -p /home/tronlong/AM57xx/kernel/Linux-RT-4.9.65
```



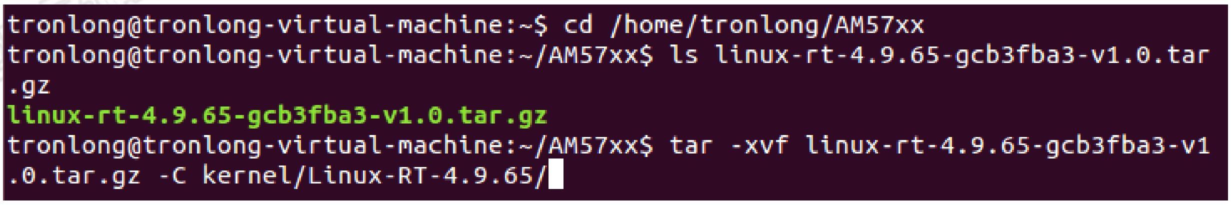
```
tronlong@tronlong-virtual-machine:~$ mkdir -p /home/tronlong/AM57xx/kernel/Linux
-RT-4.9.65
tronlong@tronlong-virtual-machine:~$ 
```

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。将其复制到"/home/tronlong/AM57xx"工作目录下，再将其解压至安装目录，执行命令如下：

Host#　cd /home/tronlong/AM57xx

Host#　tar -xvf linux-rt-4.9.65-gcb3fba3-v1.0.tar.gz -C kernel/Linux-RT-4.9.65/



```
tronlong@tronlong-virtual-machine:~$ cd /home/tronlong/AM57xx
tronlong@tronlong-virtual-machine:~/AM57xx$ ls linux-rt-4.9.65-gcb3fba3-v1.0.tar
.gz
linux-rt-4.9.65-gcb3fba3-v1.0.tar.gz
tronlong@tronlong-virtual-machine:~/AM57xx$ tar -xvf linux-rt-4.9.65-gcb3fba3-v1
.0.tar.gz -C kernel/Linux-RT-4.9.65/
```

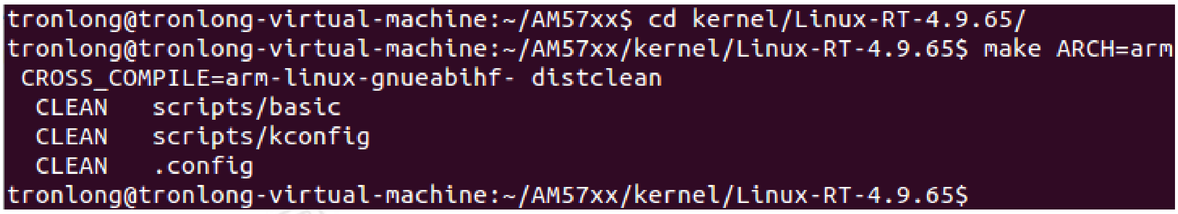
图 2

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-gnueabihf- distclean



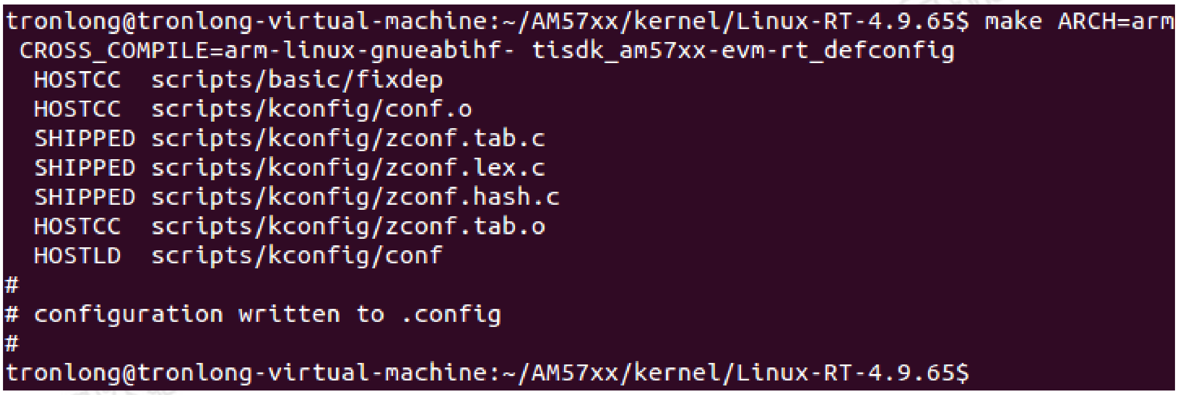
```
tronlong@tronlong-virtual-machine:~/AM57xx$ cd kernel/Linux-RT-4.9.65/
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$ make ARCH=arm
 CROSS_COMPILE=arm-linux-gnueabihf- distclean
  CLEAN   scripts/basic
  CLEAN   scripts/kconfig
  CLEAN   .config
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$
```

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-gnueabihf- tisdk_am57xx-evm-rt_defconfig

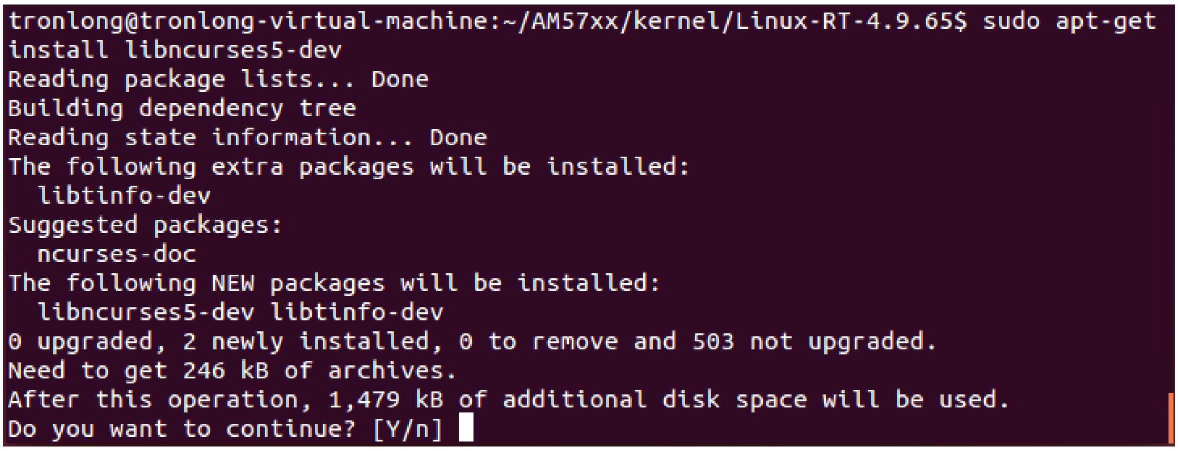


```
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$ make ARCH=arm
 CROSS_COMPILE=arm-linux-gnueabihf- tisdk_am57xx-evm-rt_defconfig
  HOSTCC  scripts/basic/fixdep
  HOSTCC  scripts/kconfig/conf.o
  SHIPPED scripts/kconfig/zconf.tab.c
  SHIPPED scripts/kconfig/zconf.lex.c
  SHIPPED scripts/kconfig/zconf.hash.c
  HOSTCC  scripts/kconfig/zconf.tab.o
  HOSTLD  scripts/kconfig/conf
#
# configuration written to .config
#
tronlong@tronlong-virtual-machine:~/AM57xx/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：

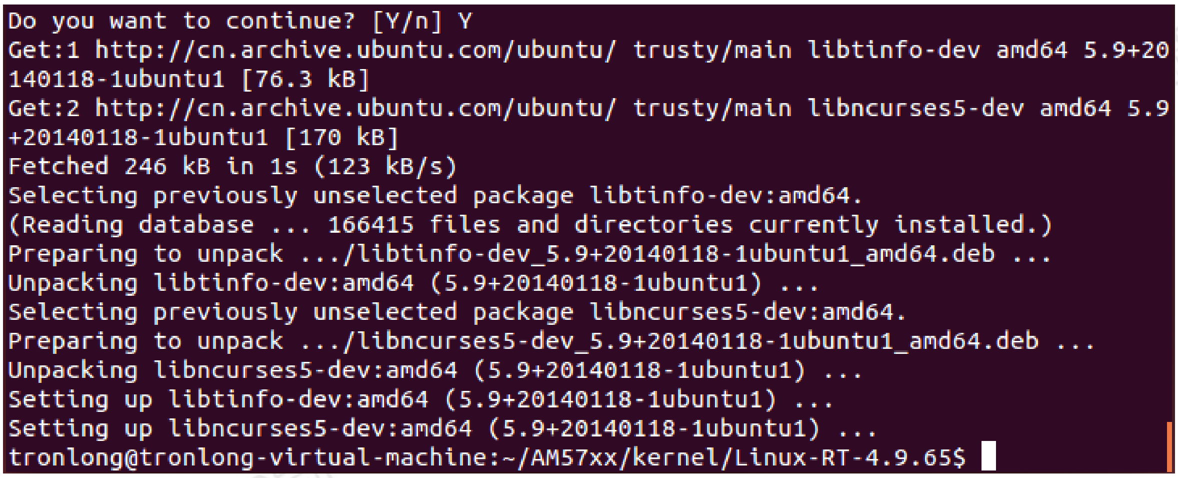
Host# sudo apt-get install libncurses5-dev



```
tronlong@tronlong-virtual-machine:~/AM57xx/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 503 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。

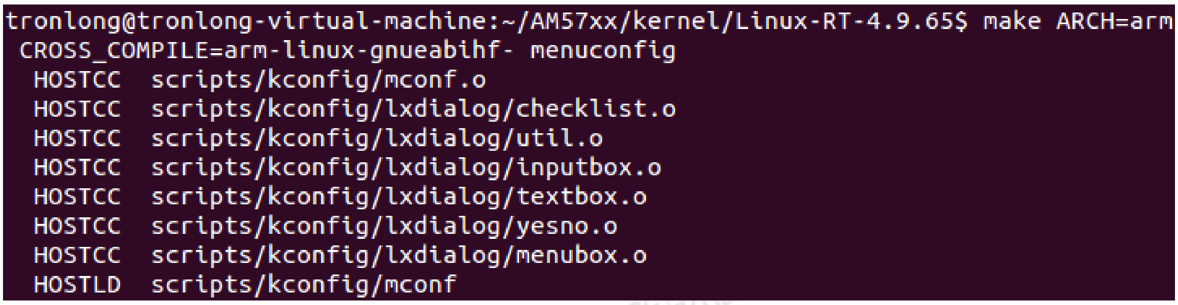


```
Do you want to continue? [Y/n] Y
Get:1 http://cn.archive.ubuntu.com/ubuntu/ trusty/main libtinfo-dev amd64 5.9+20
140118-1ubuntu1 [76.3 kB]
Get:2 http://cn.archive.ubuntu.com/ubuntu/ trusty/main libncurses5-dev amd64 5.9
+20140118-1ubuntu1 [170 kB]
Fetched 246 kB in 1s (123 kB/s)
Selecting previously unselected package libtinfo-dev:amd64.
(Reading database ... 166415 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:~/AM57xx/kernel/Linux-RT-4.9.65$
```

图 6

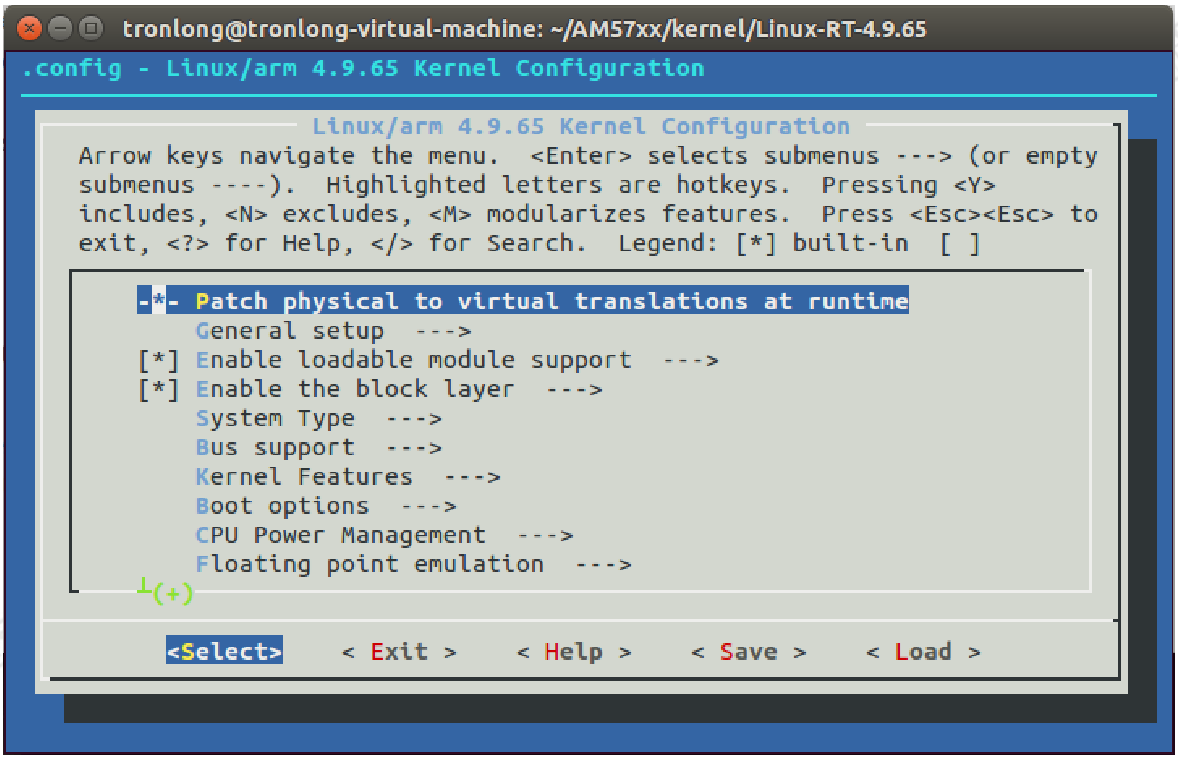
安装依赖库后，执行如下命令启动 menuconfig 配置界面，如下图所示：

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



```
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$ make ARCH=arm
 CROSS_COMPILE=arm-linux-gnueabihf- 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
```

图 7



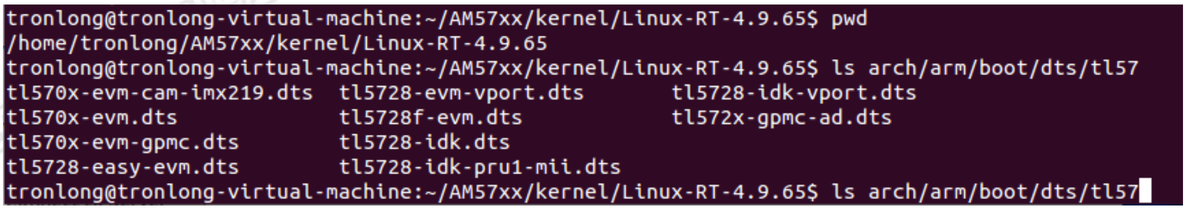
```
tronlong@tronlong-virtual-machine: ~/AM57xx/kernel/Linux-RT-4.9.65
.config - Linux/arm 4.9.65 Kernel Configuration
                    Linux/arm 4.9.65 Kernel Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty
  submenus ----).  Highlighted letters are hotkeys.  Pressing <Y>
  includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to
  exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ]

        -*- Patch physical to virtual translations at runtime
            General setup  --->
        [*] Enable loadable module support  --->
        [*] Enable the block layer  --->
            System Type  --->
            Bus support  --->
            Kernel Features  --->
            Boot options  --->
            CPU Power Management  --->
            Floating point emulation  --->
        └(+)

      <Select>    < Exit >    < Help >    < Save >    < Load >
```

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

由创龙提供的 Linux 系统设备树源文件位于 Linux 内核源码"arch/arm/boot/dts/"目录下，以 **tl57****命名，包括了基础设备树文件和动态设备树文件。AM57x 平台不同型号开发板都有对应的基础设备树文件，该文件主要描述开发板的基础硬件设备，例如 LED、按键等通用外设，系统上电启动时会自动加载。动态设备树文件中主要描述开发板特定的设备接口，例如 tl572x-gpmc-ad.dts 为 AM5728 的 GPMC 接口设备树源文件，进行 GPMC 实验时，需要在文件系统下进行加载。



```
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$ pwd
/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$ ls arch/arm/boot/dts/tl57
tl570x-evm-cam-imx219.dts  tl5728-evm-vport.dts       tl5728-idk-vport.dts
tl570x-evm.dts             tl5728f-evm.dts            tl572x-gpmc-ad.dts
tl570x-evm-gpmc.dts        tl5728-idk.dts
tl5728-easy-evm.dts        tl5728-idk-pru1-mii.dts
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$ ls arch/arm/boot/dts/tl57
```

图 9

表 2

设备树源文件	注释说明
tl570x-evm.dts	TL570x-EVM 基础设备树文件
tl5728-easy-evm.dts	TL5728-EasyEVM 基础设备树文件
tl5728-idk.dts tl5728-easy-evm.dts	TL5728-IDK 基础设备树文件
tl5728f-evm.dts	TL5728F-EVM 基础设备树文件
tl570x-evm-cam-imx219.dts tl5728-evm-vport.dts tl5728-idk-vport.dts tl572x-gpmc-ad.dts tl570x-evm-gpmc.dts 等	（特定功能）动态设备树文件

➢ 编译基础设备树文件

在 Linux-RT 内核源码安装目录下，分别执行如下命令，编译生成对应平台的二进制基础设备树文件：

■ TL570x-EVM 开发板

```
Host#    make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- tl570x-evm.dtb
```

■ TL5728-EasyEVM 开发板

```
Host#    make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- tl5728-easy-evm.dtb
```

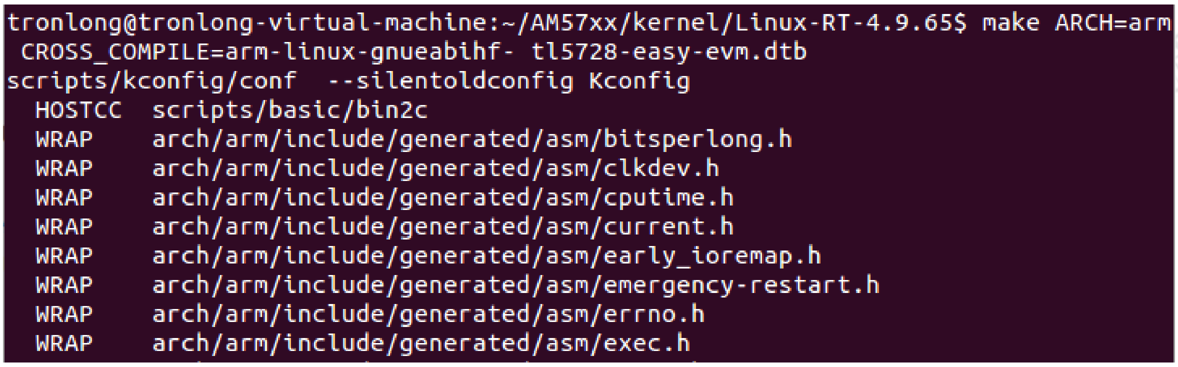
■ TL5728-IDK 开发板

```
Host#    make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- tl5728-idk.dtb
Host#    make  ARCH=arm  CROSS_COMPILE=arm-linux-gnueabihf-  tl5728-idk-pru1-mii.dtb
```

■ TL5728F-EVM 开发板

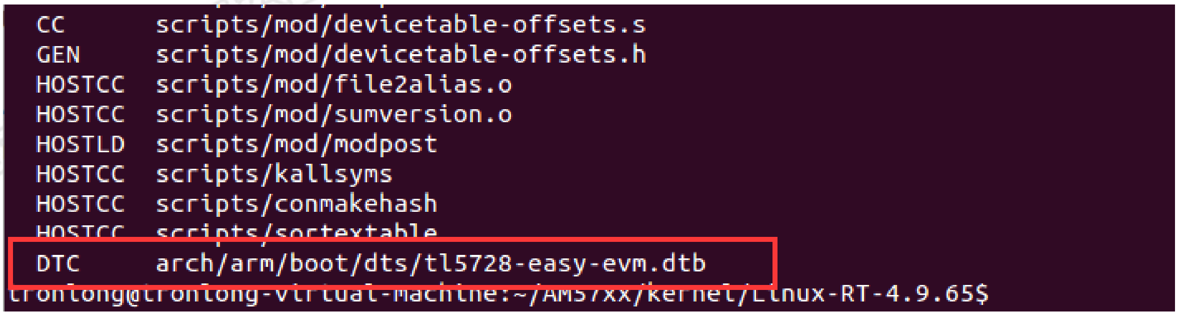
```
Host#    make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- tl5728f-evm.dtb
```

以编译 TL5728-EasyEVM 平台基础设备树文件为例，执行指令如下图所示：



```
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$ make ARCH=arm
 CROSS_COMPILE=arm-linux-gnueabihf- tl5728-easy-evm.dtb
scripts/kconfig/conf  --silentoldconfig Kconfig
  HOSTCC  scripts/basic/bin2c
  WRAP    arch/arm/include/generated/asm/bitsperlong.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
```

图 10



```
  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/tl5728-easy-evm.dtb
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$
```

图 11

编译完成后，会在 Linux-RT 内核源码目录"arch/arm/boot/dts"路径下，生成对应平台的基础设备树文件。如需替换基础设备树，只需将编译生成的新基础设备树文件替换 SD 系统启动卡 rootfs 分区"/boot"目录下的对应文件即可。

➢ **编译动态设备树文件**

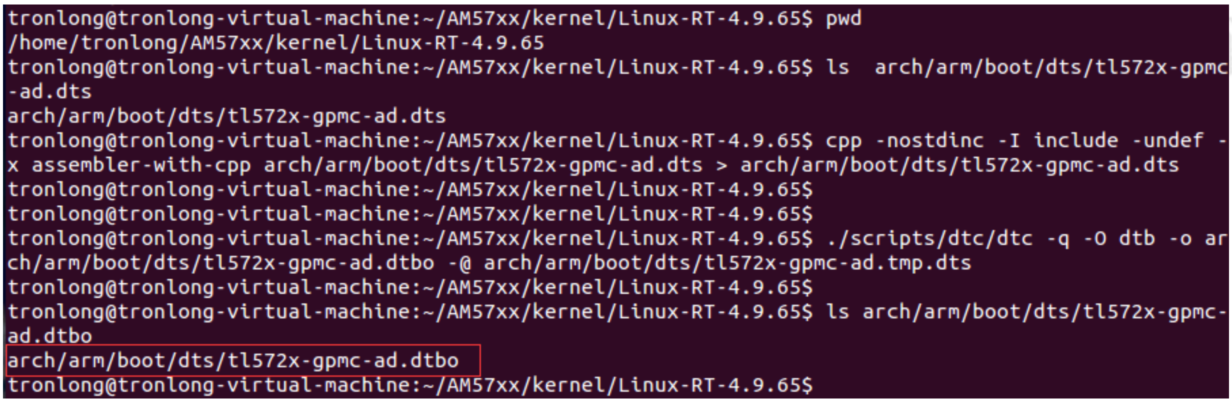
进行 AM5728 的 GPMC 通信测试实验时，需要用到 tl572x-gpmc-ad.dts 动态设备树源文件。这里以 tl572x-gpmc-ad.dts 动态设备树源文件为例，演示编译动态设备树源文件的方法，如需编译其他动态设备树源文件，替换指令中设备树源文件名即可。

执行以下指令，对动态设备树文件进行预编译。

Host# cpp -nostdinc -I include -undef -x assembler-with-cpp arch/arm/boot/dts/**tl572x-gpmc-ad.dts** > arch/arm/boot/dts/**tl572x-gpmc-ad.tmp.dts**

执行以下指令，使用 DTC 编译器编译动态设备树源文件。

Host# ./scripts/dtc/dtc -q -O dtb -o arch/arm/boot/dts/**tl572x-gpmc-ad.dtbo** -@ arch/arm/boot/dts/**tl572x-gpmc-ad.tmp.dts**



```
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$ pwd
/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$ ls  arch/arm/boot/dts/tl572x-gpmc
-ad.dts
arch/arm/boot/dts/tl572x-gpmc-ad.dts
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$ cpp -nostdinc -I include -undef -
x assembler-with-cpp arch/arm/boot/dts/tl572x-gpmc-ad.dts > arch/arm/boot/dts/tl572x-gpmc-ad.dts
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$ ./scripts/dtc/dtc -q -O dtb -o ar
ch/arm/boot/dts/tl572x-gpmc-ad.dtbo -@ arch/arm/boot/dts/tl572x-gpmc-ad.tmp.dts
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$ ls arch/arm/boot/dts/tl572x-gpmc-
ad.dtbo
arch/arm/boot/dts/tl572x-gpmc-ad.dtbo
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$
```

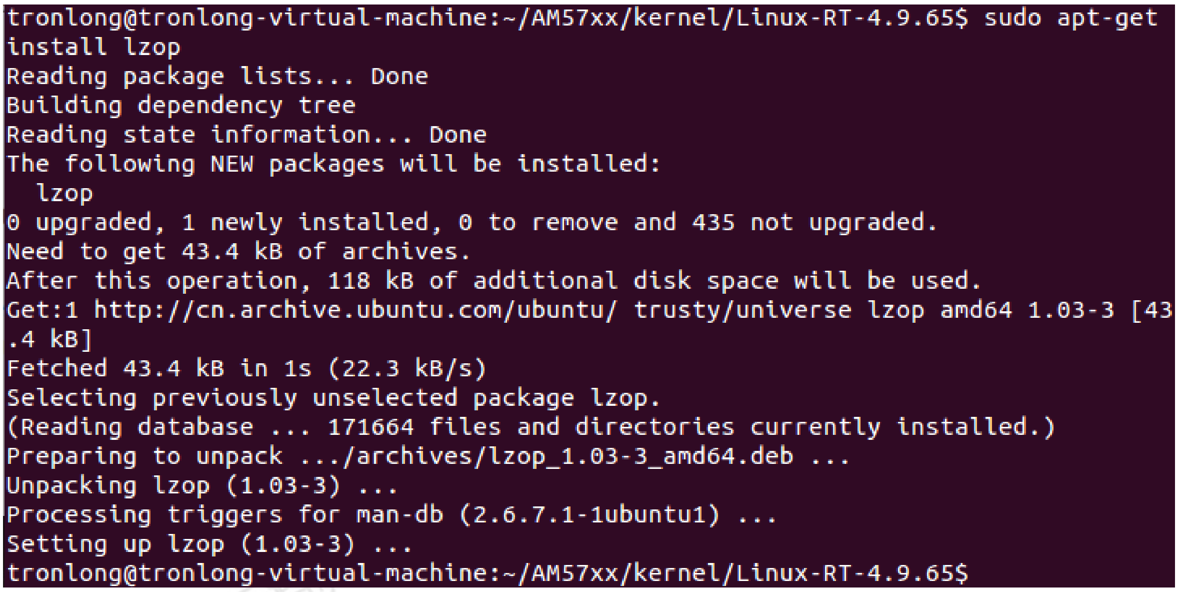
图 12

编译完成后，会在内核源码目录"arch/arm/boot/dts"路径下生成***.dtbo 设备树二进制文件。参照《Linux-RT SD 系统启动卡制作及系统固化》文档，将其拷贝到 SD 系统启动卡 rootfs 分区的"/lib/firmware/"目录下，在文件系统下进行加载。

2.5 编译 Linux-RT 内核

执行如下指令安装 lzop 压缩工具，此工具在编译 Linux-RT 内核时需要用到：

Host# sudo apt-get install lzop



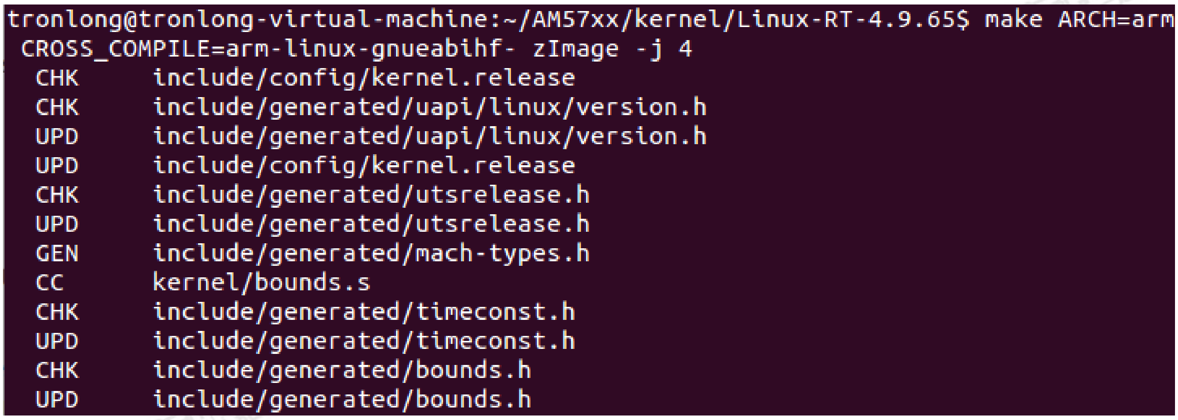
```
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$ sudo apt-get
install lzop
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  lzop
0 upgraded, 1 newly installed, 0 to remove and 435 not upgraded.
Need to get 43.4 kB of archives.
After this operation, 118 kB of additional disk space will be used.
Get:1 http://cn.archive.ubuntu.com/ubuntu/ trusty/universe lzop amd64 1.03-3 [43
.4 kB]
Fetched 43.4 kB in 1s (22.3 kB/s)
Selecting previously unselected package lzop.
(Reading database ... 171664 files and directories currently installed.)
Preparing to unpack .../archives/lzop_1.03-3_amd64.deb ...
Unpacking lzop (1.03-3) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Setting up lzop (1.03-3) ...
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$
```

图 13

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

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

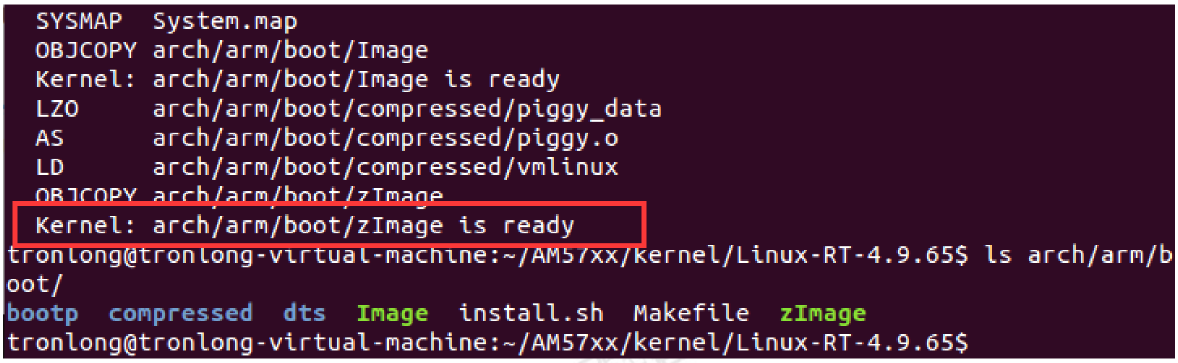
"-j 4"是一个编译选项，告诉操作系统用 4 个线程去编译，加快编译速度。



```
tronlong@tronlong-virtual-machine:~/AM57xx/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
  CHK     include/generated/timeconst.h
  UPD     include/generated/timeconst.h
  CHK     include/generated/bounds.h
  UPD     include/generated/bounds.h
```

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：



```
  SYSMAP  System.map
  OBJCOPY arch/arm/boot/Image
  Kernel: arch/arm/boot/Image is ready
  LZO     arch/arm/boot/compressed/piggy_data
  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:~/AM57xx/kernel/Linux-RT-4.9.65$ ls arch/arm/b
oot/
bootp  compressed  dts  Image  install.sh  Makefile  zImage
tronlong@tronlong-virtual-machine:~/AM57xx/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。

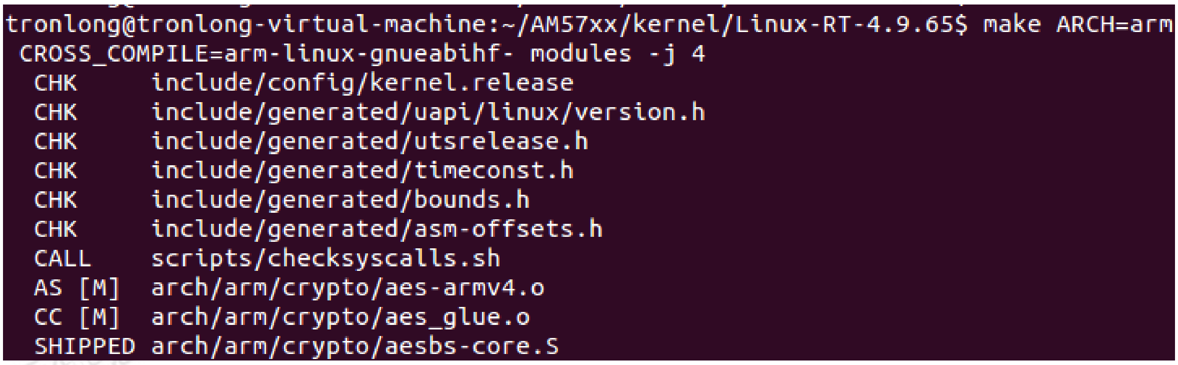
使用 SD 系统启动卡启动系统时，如需替换内核镜像，有如下方法：

- 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



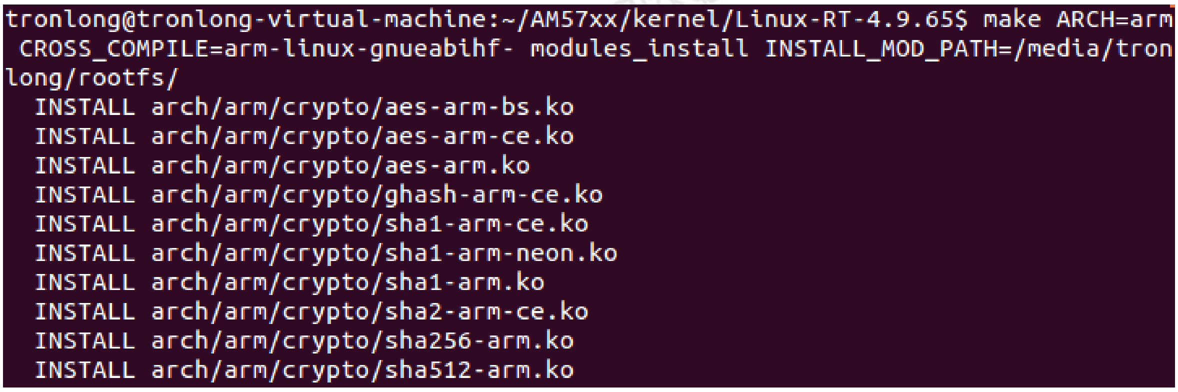
```
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$ make ARCH=arm
 CROSS_COMPILE=arm-linux-gnueabihf- modules -j 4
  CHK     include/config/kernel.release
  CHK     include/generated/uapi/linux/version.h
  CHK     include/generated/utsrelease.h
  CHK     include/generated/timeconst.h
  CHK     include/generated/bounds.h
  CHK     include/generated/asm-offsets.h
  CALL    scripts/checksyscalls.sh
  AS [M]  arch/arm/crypto/aes-armv4.o
  CC [M]  arch/arm/crypto/aes_glue.o
  SHIPPED arch/arm/crypto/aesbs-core.S
```

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：

Host#　make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- modules_install INSTALL_MOD_PATH=/media/tronlong/rootfs/

"/media/tronlong/rootfs/"为 SD 卡文件系统在 Ubuntu 的挂载路径。



```
tronlong@tronlong-virtual-machine:~/AM57xx/kernel/Linux-RT-4.9.65$ make ARCH=arm
 CROSS_COMPILE=arm-linux-gnueabihf- modules_install INSTALL_MOD_PATH=/media/tron
long/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
```

图 17

3 extra 驱动编译

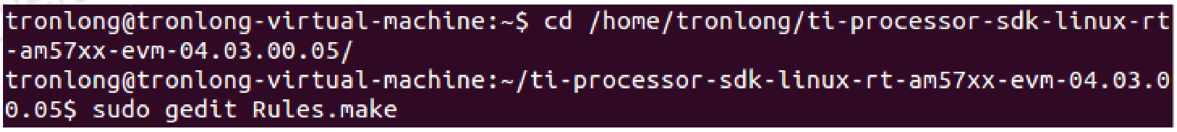
对于重新配置、编译过的内核，其版本号可能和做卡工具中的内核版本号不一致，此时会因为文件系统上的驱动模块和内核版本不一致而导致驱动模块无法安装，从而造成某些功能不正常（主要是显示相关的）。因此编译过的内核，应按照如下步骤操作，避免上述问题出现。

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-am57xx-evm-04.03.00.05/

Host#　sudo gedit Rules.make



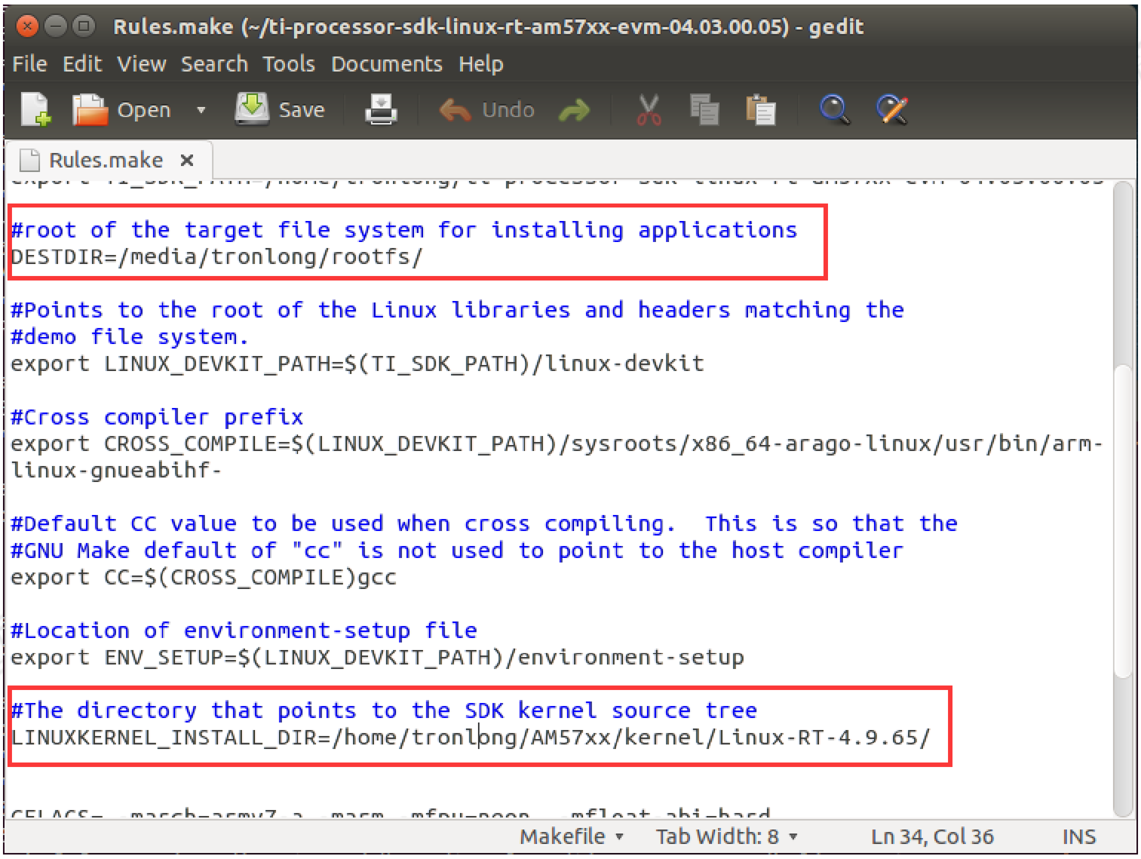
```
tronlong@tronlong-virtual-machine:~$ cd /home/tronlong/ti-processor-sdk-linux-rt
-am57xx-evm-04.03.00.05/
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$ sudo gedit Rules.make
```

图 18

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/AM57xx/kernel/Linux-RT-4.9.65/　//修改为内核所在目录

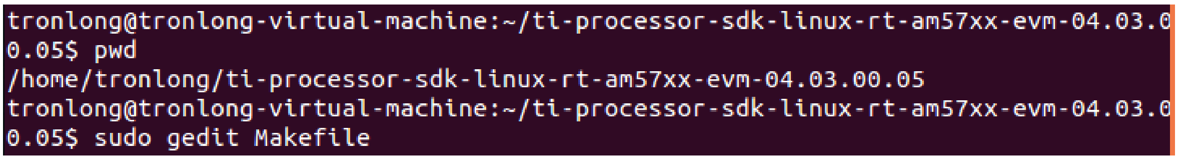

Rules.make (~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
Rules.make
Makefile
Tab Width: 8
Ln 34, Col 36
INS

```
export TI_SDK_PATH=/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05

#root of the target file system for installing applications
DESTDIR=/media/tronlong/rootfs/

#Points to the root of the Linux libraries and headers matching the
#demo file system.
export LINUX_DEVKIT_PATH=$(TI_SDK_PATH)/linux-devkit

#Cross compiler prefix
export CROSS_COMPILE=$(LINUX_DEVKIT_PATH)/sysroots/x86_64-arago-linux/usr/bin/arm-linux-gnueabihf-

#Default CC value to be used when cross compiling.  This is so that the
#GNU Make default of "cc" is not used to point to the host compiler
export CC=$(CROSS_COMPILE)gcc

#Location of environment-setup file
export ENV_SETUP=$(LINUX_DEVKIT_PATH)/environment-setup

#The directory that points to the SDK kernel source tree
LINUXKERNEL_INSTALL_DIR=/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65/


CFLAGS= -march=armv7-a -marm -mfpu=neon -mfloat-abi=hard
```

图 19

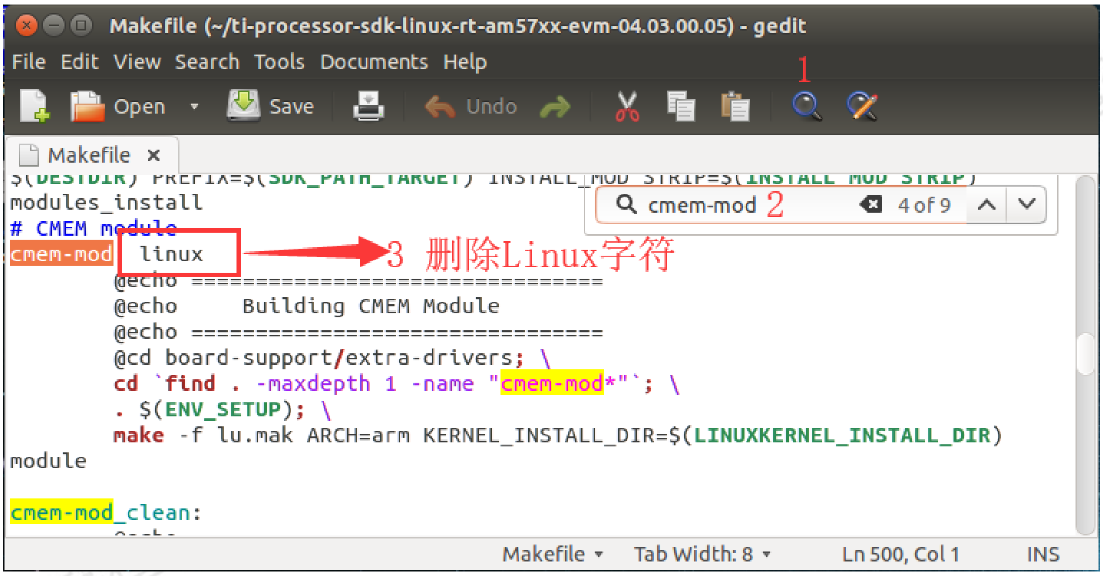
执行如下命令打开 Makefile 文件，将 cmem-mod、cryptodev、gdbserverproxy-module-drv、uio-module-drv、debugss-module-drv、ti-sgx-ddk-km 驱动所在行的"linux"字符串删除，这样在重新编译这些驱动时，就不会再次编译内核。修改完成后保存，分别如下图所示：

Host# sudo gedit Makefile



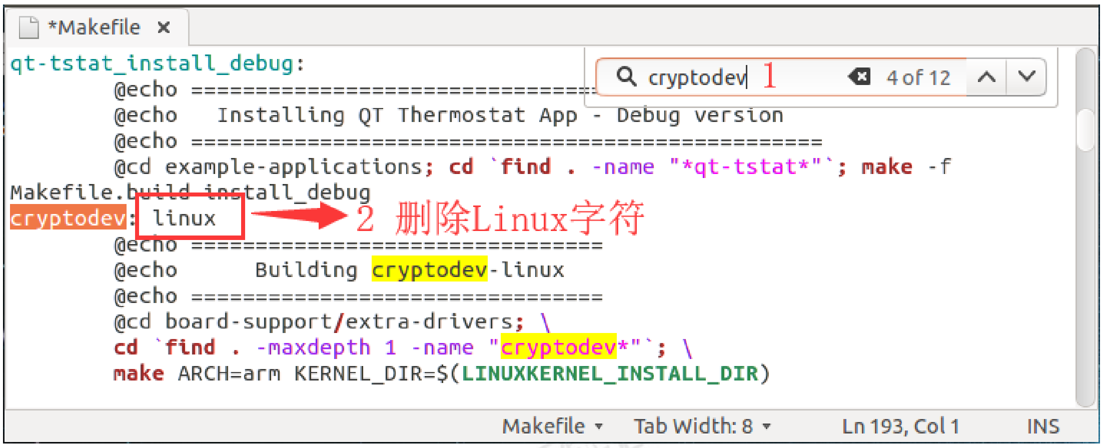
```
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05$ pwd
/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05$ sudo gedit Makefile
```

图 20


Makefile (~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
1
Makefile
cmem-mod 2
4 of 9
3 删除Linux字符
Makefile
Tab Width: 8
Ln 500, Col 1
INS

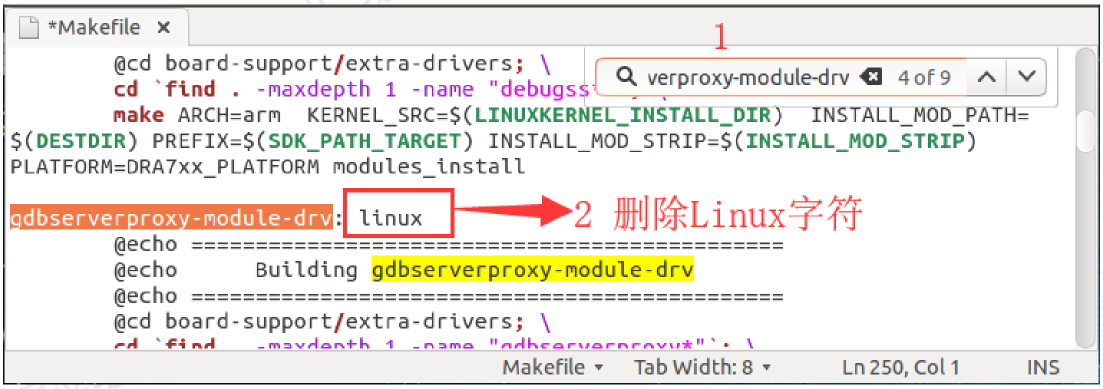
```
$(DESTDIR) PREFIX=$(SDK_PATH_TARGET) INSTALL_MOD_STRIP=$(INSTALL_MOD_STRIP)
modules_install
# CMEM module
cmem-mod  linux
        @echo =================================
        @echo      Building CMEM Module
        @echo =================================
        @cd board-support/extra-drivers; \
        cd `find . -maxdepth 1 -name "cmem-mod*"`; \
        . $(ENV_SETUP); \
        make -f lu.mak ARCH=arm KERNEL_INSTALL_DIR=$(LINUXKERNEL_INSTALL_DIR)
module

cmem-mod_clean:
```

图 21



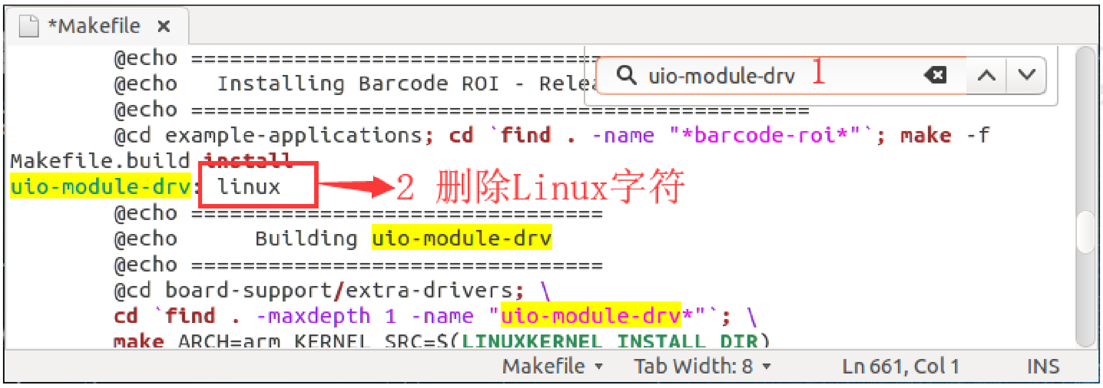
```
qt-tstat_install_debug:
        @echo =================================
        @echo    Installing QT Thermostat App - Debug version
        @echo ==================================================
        @cd example-applications; cd `find . -name "*qt-tstat*"`; make -f
Makefile.build install_debug
cryptodev: linux
        @echo =================================
        @echo      Building cryptodev-linux
        @echo =================================
        @cd board-support/extra-drivers; \
        cd `find . -maxdepth 1 -name "cryptodev*"`; \
        make ARCH=arm KERNEL_DIR=$(LINUXKERNEL_INSTALL_DIR)
```

图 22



```
        @cd board-support/extra-drivers; \
        cd `find . -maxdepth 1 -name "debugss
        make ARCH=arm  KERNEL_SRC=$(LINUXKERNEL_INSTALL_DIR)  INSTALL_MOD_PATH=
$(DESTDIR) PREFIX=$(SDK_PATH_TARGET) INSTALL_MOD_STRIP=$(INSTALL_MOD_STRIP)
PLATFORM=DRA7xx_PLATFORM modules_install

gdbserverproxy-module-drv: linux
        @echo ==============================================
        @echo      Building gdbserverproxy-module-drv
        @echo ==============================================
        @cd board-support/extra-drivers; \
        cd `find . -maxdepth 1 -name "gdbserverproxy*"`; \
```

图 23

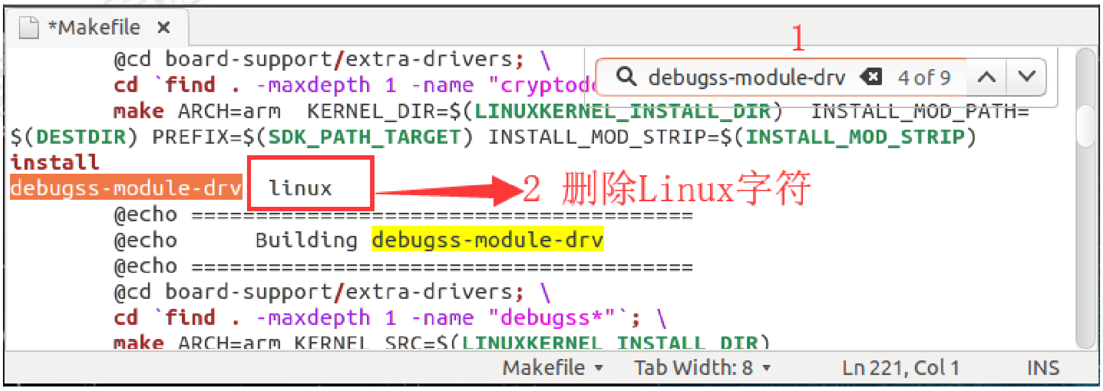


```
	@echo ================================
	@echo    Installing Barcode ROI - Relea
	@echo ================================
	@cd example-applications; cd `find . -name "*barcode-roi*"`; make -f
Makefile.build install
uio-module-drv: linux
	@echo ================================
	@echo      Building uio-module-drv
	@echo ================================
	@cd board-support/extra-drivers; \
	cd `find . -maxdepth 1 -name "uio-module-drv*"`; \
	make ARCH=arm KERNEL_SRC=$(LINUXKERNEL_INSTALL_DIR)
```

1

2 删除Linux字符

图 24

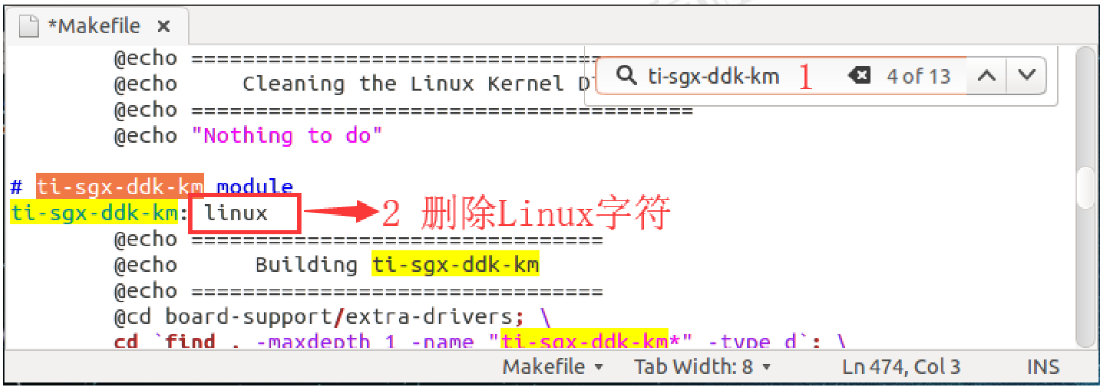


```
	@cd board-support/extra-drivers; \
	cd `find . -maxdepth 1 -name "cryptod
	make ARCH=arm  KERNEL_DIR=$(LINUXKERNEL_INSTALL_DIR)  INSTALL_MOD_PATH=
$(DESTDIR) PREFIX=$(SDK_PATH_TARGET) INSTALL_MOD_STRIP=$(INSTALL_MOD_STRIP)
install
debugss-module-drv: linux
	@echo ======================================
	@echo      Building debugss-module-drv
	@echo ======================================
	@cd board-support/extra-drivers; \
	cd `find . -maxdepth 1 -name "debugss*"`; \
	make ARCH=arm KERNEL_SRC=$(LINUXKERNEL_INSTALL_DIR)
```

1

2 删除Linux字符

图 25



```
	@echo ================================
	@echo     Cleaning the Linux Kernel D
	@echo ======================================
	@echo "Nothing to do"

# ti-sgx-ddk-km module
ti-sgx-ddk-km: linux
	@echo ================================
	@echo      Building ti-sgx-ddk-km
	@echo ================================
	@cd board-support/extra-drivers; \
	cd `find . -maxdepth 1 -name "ti-sgx-ddk-km*" -type d`; \
```

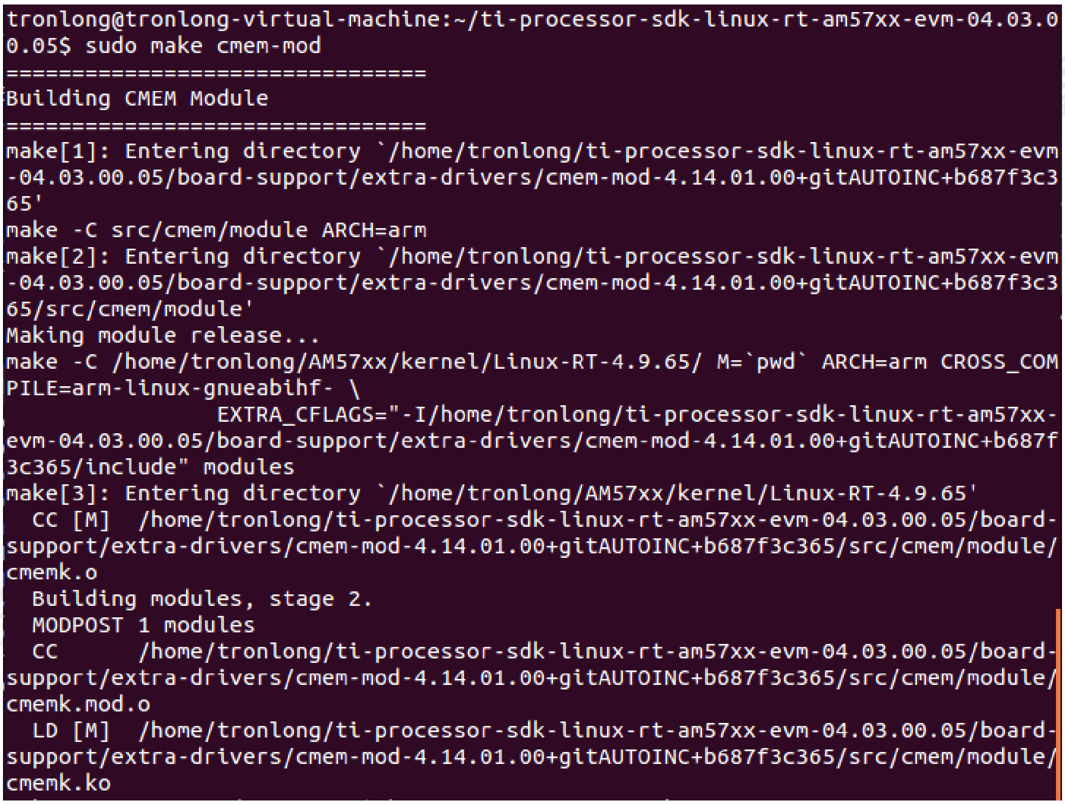
1

2 删除Linux字符

图 26

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：

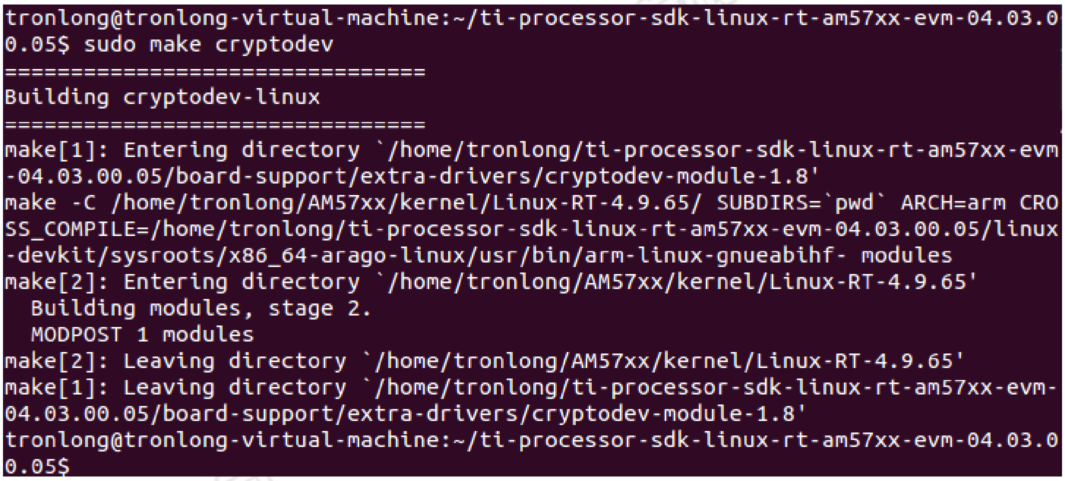
Host# sudo make cmem-mod



```
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$ sudo make cmem-mod
================================
Building CMEM Module
================================
make[1]: Entering directory `/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm
-04.03.00.05/board-support/extra-drivers/cmem-mod-4.14.01.00+gitAUTOINC+b687f3c3
65'
make -C src/cmem/module ARCH=arm
make[2]: Entering directory `/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm
-04.03.00.05/board-support/extra-drivers/cmem-mod-4.14.01.00+gitAUTOINC+b687f3c3
65/src/cmem/module'
Making module release...
make -C /home/tronlong/AM57xx/kernel/Linux-RT-4.9.65/ M=`pwd` ARCH=arm CROSS_COM
PILE=arm-linux-gnueabihf- \
                EXTRA_CFLAGS="-I/home/tronlong/ti-processor-sdk-linux-rt-am57xx-
evm-04.03.00.05/board-support/extra-drivers/cmem-mod-4.14.01.00+gitAUTOINC+b687f
3c365/include" modules
make[3]: Entering directory `/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65'
  CC [M]  /home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-
support/extra-drivers/cmem-mod-4.14.01.00+gitAUTOINC+b687f3c365/src/cmem/module/
cmemk.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-
support/extra-drivers/cmem-mod-4.14.01.00+gitAUTOINC+b687f3c365/src/cmem/module/
cmemk.mod.o
  LD [M]  /home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-
support/extra-drivers/cmem-mod-4.14.01.00+gitAUTOINC+b687f3c365/src/cmem/module/
cmemk.ko
```

图 27

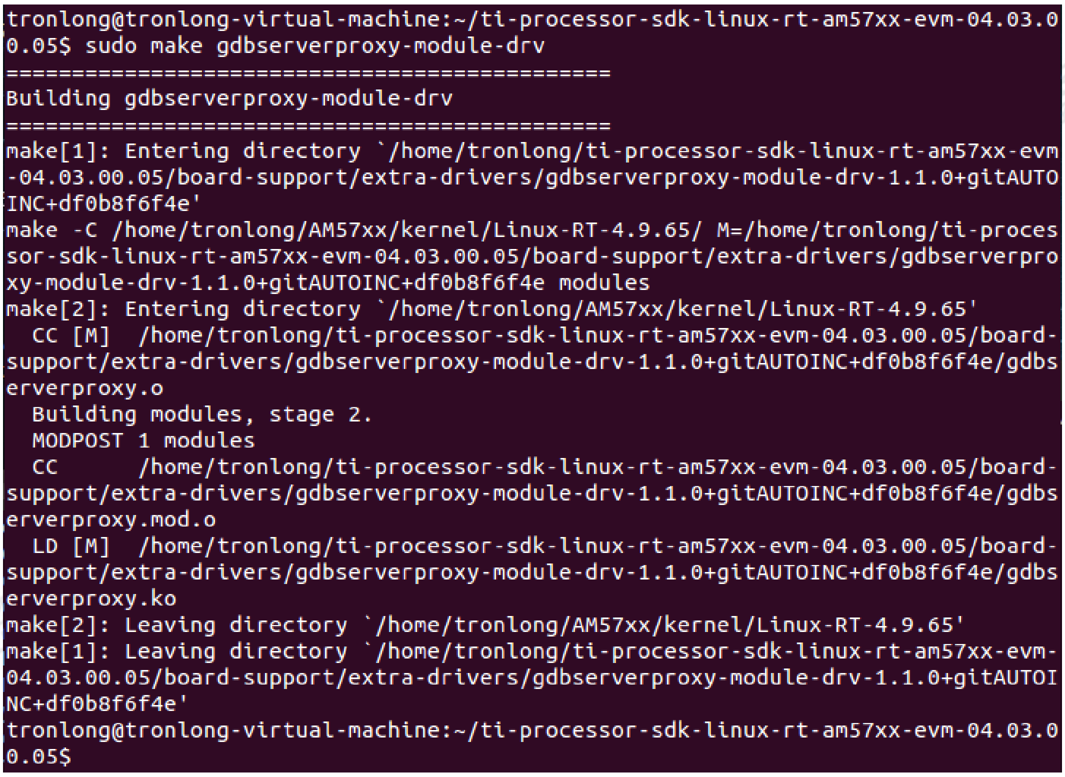
Host# sudo make cryptodev



```
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$ sudo make cryptodev
================================
Building cryptodev-linux
================================
make[1]: Entering directory `/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm
-04.03.00.05/board-support/extra-drivers/cryptodev-module-1.8'
make -C /home/tronlong/AM57xx/kernel/Linux-RT-4.9.65/ SUBDIRS=`pwd` ARCH=arm CRO
SS_COMPILE=/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/linux
-devkit/sysroots/x86_64-arago-linux/usr/bin/arm-linux-gnueabihf- modules
make[2]: Entering directory `/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65'
  Building modules, stage 2.
  MODPOST 1 modules
make[2]: Leaving directory `/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65'
make[1]: Leaving directory `/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-
04.03.00.05/board-support/extra-drivers/cryptodev-module-1.8'
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$
```

图 28

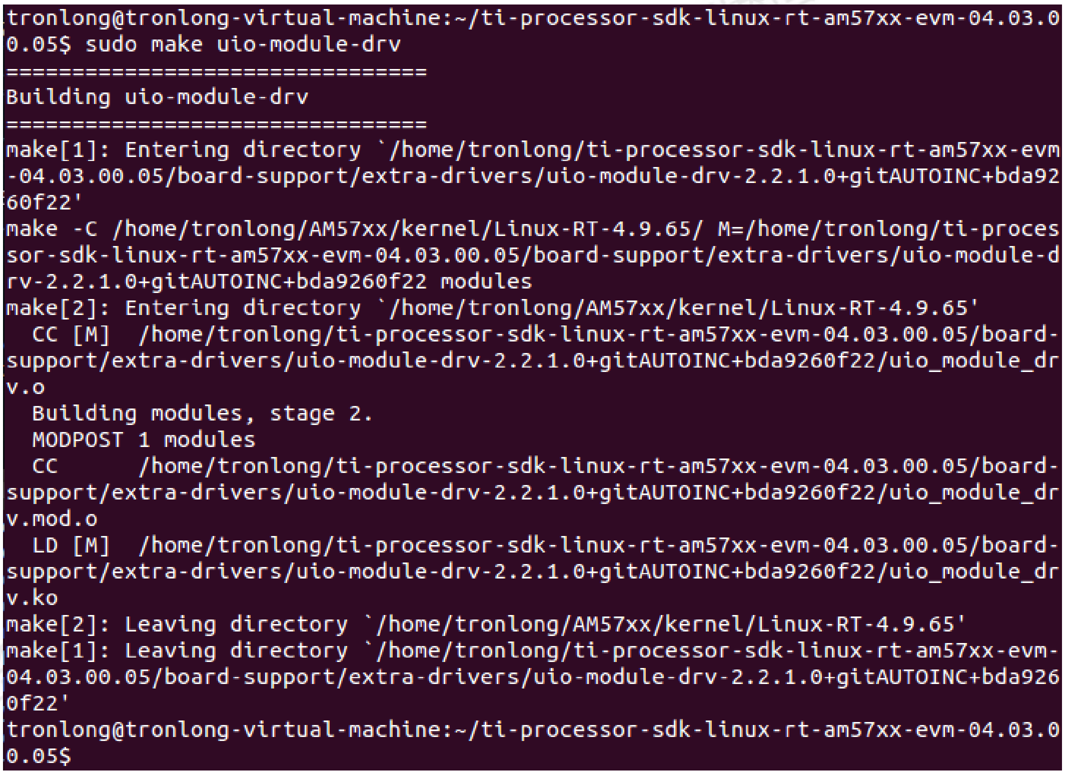
Host# sudo make gdbserverproxy-module-drv



```
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$ sudo make gdbserverproxy-module-drv
==============================================
Building gdbserverproxy-module-drv
==============================================
make[1]: Entering directory `/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm
-04.03.00.05/board-support/extra-drivers/gdbserverproxy-module-drv-1.1.0+gitAUTO
INC+df0b8f6f4e'
make -C /home/tronlong/AM57xx/kernel/Linux-RT-4.9.65/ M=/home/tronlong/ti-proces
sor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-support/extra-drivers/gdbserverpro
xy-module-drv-1.1.0+gitAUTOINC+df0b8f6f4e modules
make[2]: Entering directory `/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65'
  CC [M]  /home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-
support/extra-drivers/gdbserverproxy-module-drv-1.1.0+gitAUTOINC+df0b8f6f4e/gdbs
erverproxy.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-
support/extra-drivers/gdbserverproxy-module-drv-1.1.0+gitAUTOINC+df0b8f6f4e/gdbs
erverproxy.mod.o
  LD [M]  /home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-
support/extra-drivers/gdbserverproxy-module-drv-1.1.0+gitAUTOINC+df0b8f6f4e/gdbs
erverproxy.ko
make[2]: Leaving directory `/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65'
make[1]: Leaving directory `/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-
04.03.00.05/board-support/extra-drivers/gdbserverproxy-module-drv-1.1.0+gitAUTOI
NC+df0b8f6f4e'
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$
```

图 29

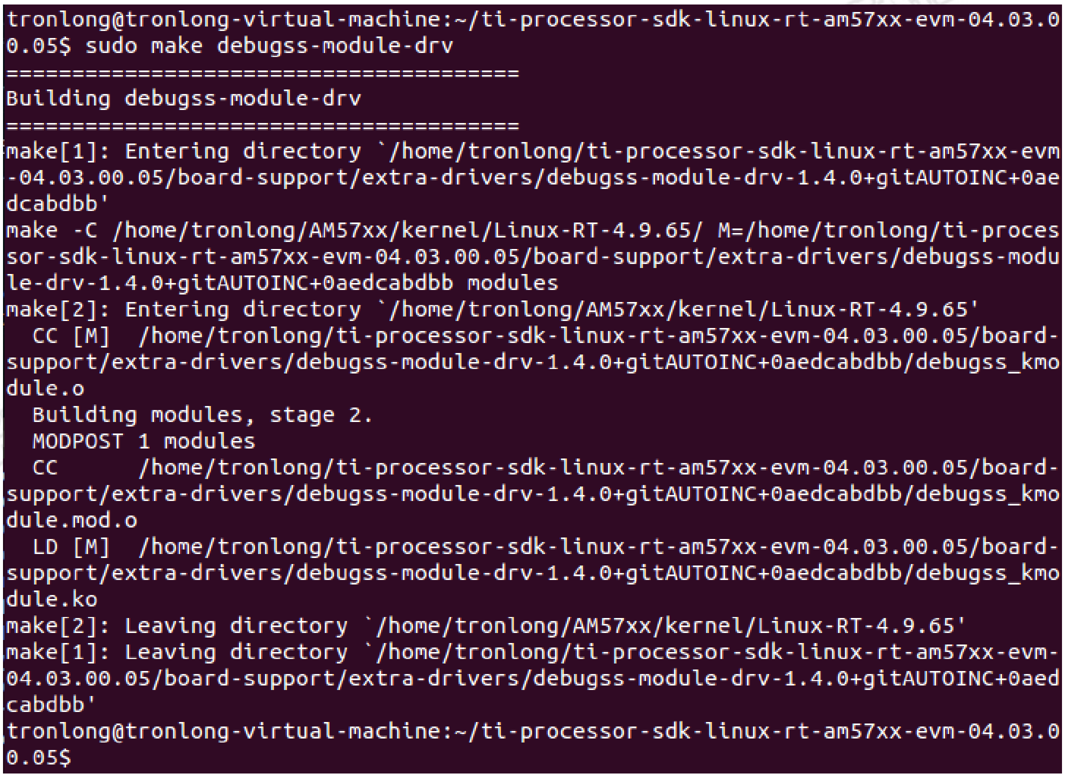
Host# sudo make uio-module-drv



```
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$ sudo make uio-module-drv
================================
Building uio-module-drv
================================
make[1]: Entering directory `/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm
-04.03.00.05/board-support/extra-drivers/uio-module-drv-2.2.1.0+gitAUTOINC+bda92
60f22'
make -C /home/tronlong/AM57xx/kernel/Linux-RT-4.9.65/ M=/home/tronlong/ti-proces
sor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-support/extra-drivers/uio-module-d
rv-2.2.1.0+gitAUTOINC+bda9260f22 modules
make[2]: Entering directory `/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65'
  CC [M]  /home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-
support/extra-drivers/uio-module-drv-2.2.1.0+gitAUTOINC+bda9260f22/uio_module_dr
v.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-
support/extra-drivers/uio-module-drv-2.2.1.0+gitAUTOINC+bda9260f22/uio_module_dr
v.mod.o
  LD [M]  /home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-
support/extra-drivers/uio-module-drv-2.2.1.0+gitAUTOINC+bda9260f22/uio_module_dr
v.ko
make[2]: Leaving directory `/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65'
make[1]: Leaving directory `/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-
04.03.00.05/board-support/extra-drivers/uio-module-drv-2.2.1.0+gitAUTOINC+bda926
0f22'
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$
```

图 30

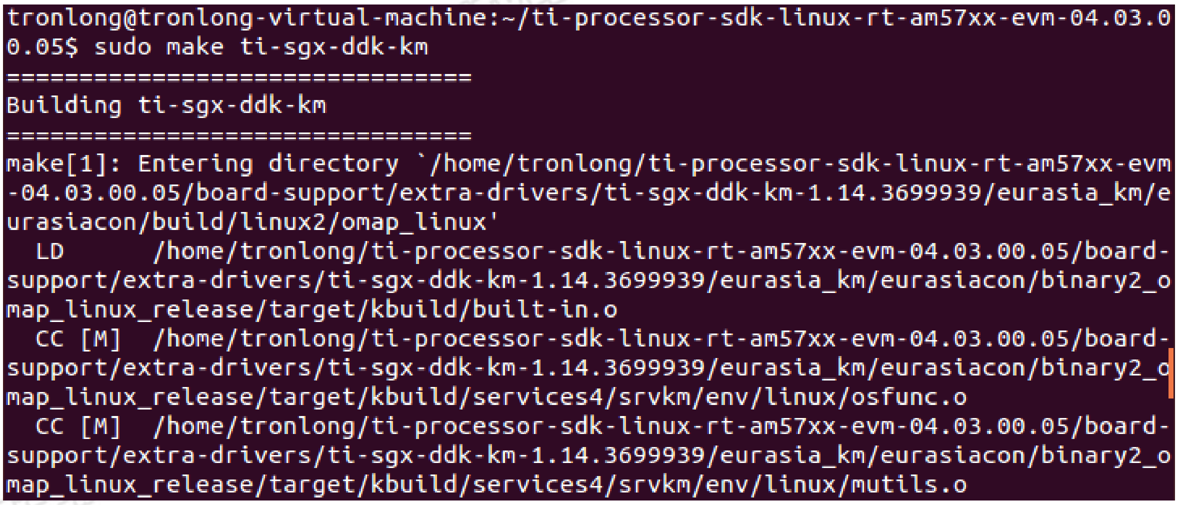
Host# sudo make debugss-module-drv



```
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$ sudo make debugss-module-drv
=======================================
Building debugss-module-drv
=======================================
make[1]: Entering directory `/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm
-04.03.00.05/board-support/extra-drivers/debugss-module-drv-1.4.0+gitAUTOINC+0ae
dcabdbb'
make -C /home/tronlong/AM57xx/kernel/Linux-RT-4.9.65/ M=/home/tronlong/ti-proces
sor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-support/extra-drivers/debugss-modu
le-drv-1.4.0+gitAUTOINC+0aedcabdbb modules
make[2]: Entering directory `/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65'
  CC [M]  /home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-
support/extra-drivers/debugss-module-drv-1.4.0+gitAUTOINC+0aedcabdbb/debugss_kmo
dule.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-
support/extra-drivers/debugss-module-drv-1.4.0+gitAUTOINC+0aedcabdbb/debugss_kmo
dule.mod.o
  LD [M]  /home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-
support/extra-drivers/debugss-module-drv-1.4.0+gitAUTOINC+0aedcabdbb/debugss_kmo
dule.ko
make[2]: Leaving directory `/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65'
make[1]: Leaving directory `/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-
04.03.00.05/board-support/extra-drivers/debugss-module-drv-1.4.0+gitAUTOINC+0aed
cabdbb'
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$
```

图 31

Host# sudo make ti-sgx-ddk-km



```
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$ sudo make ti-sgx-ddk-km
================================
Building ti-sgx-ddk-km
================================
make[1]: Entering directory `/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm
-04.03.00.05/board-support/extra-drivers/ti-sgx-ddk-km-1.14.3699939/eurasia_km/e
urasiacon/build/linux2/omap_linux'
  LD      /home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-
support/extra-drivers/ti-sgx-ddk-km-1.14.3699939/eurasia_km/eurasiacon/binary2_o
map_linux_release/target/kbuild/built-in.o
  CC [M]  /home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-
support/extra-drivers/ti-sgx-ddk-km-1.14.3699939/eurasia_km/eurasiacon/binary2_o
map_linux_release/target/kbuild/services4/srvkm/env/linux/osfunc.o
  CC [M]  /home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-
support/extra-drivers/ti-sgx-ddk-km-1.14.3699939/eurasia_km/eurasiacon/binary2_o
map_linux_release/target/kbuild/services4/srvkm/env/linux/mutils.o
```

图 32

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

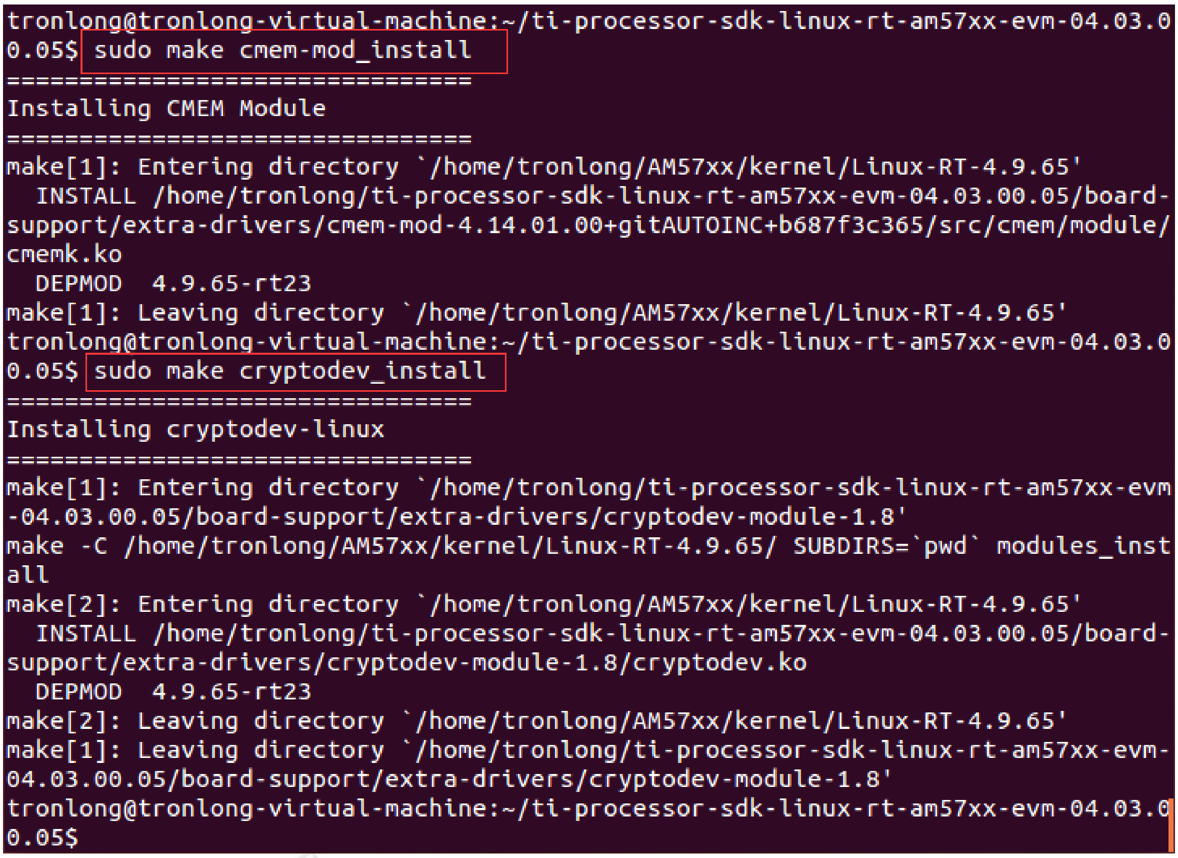
Host# sudo make cryptodev_install

Host# sudo make gdbserverproxy-module-drv_install

Host# sudo make uio-module-drv_install

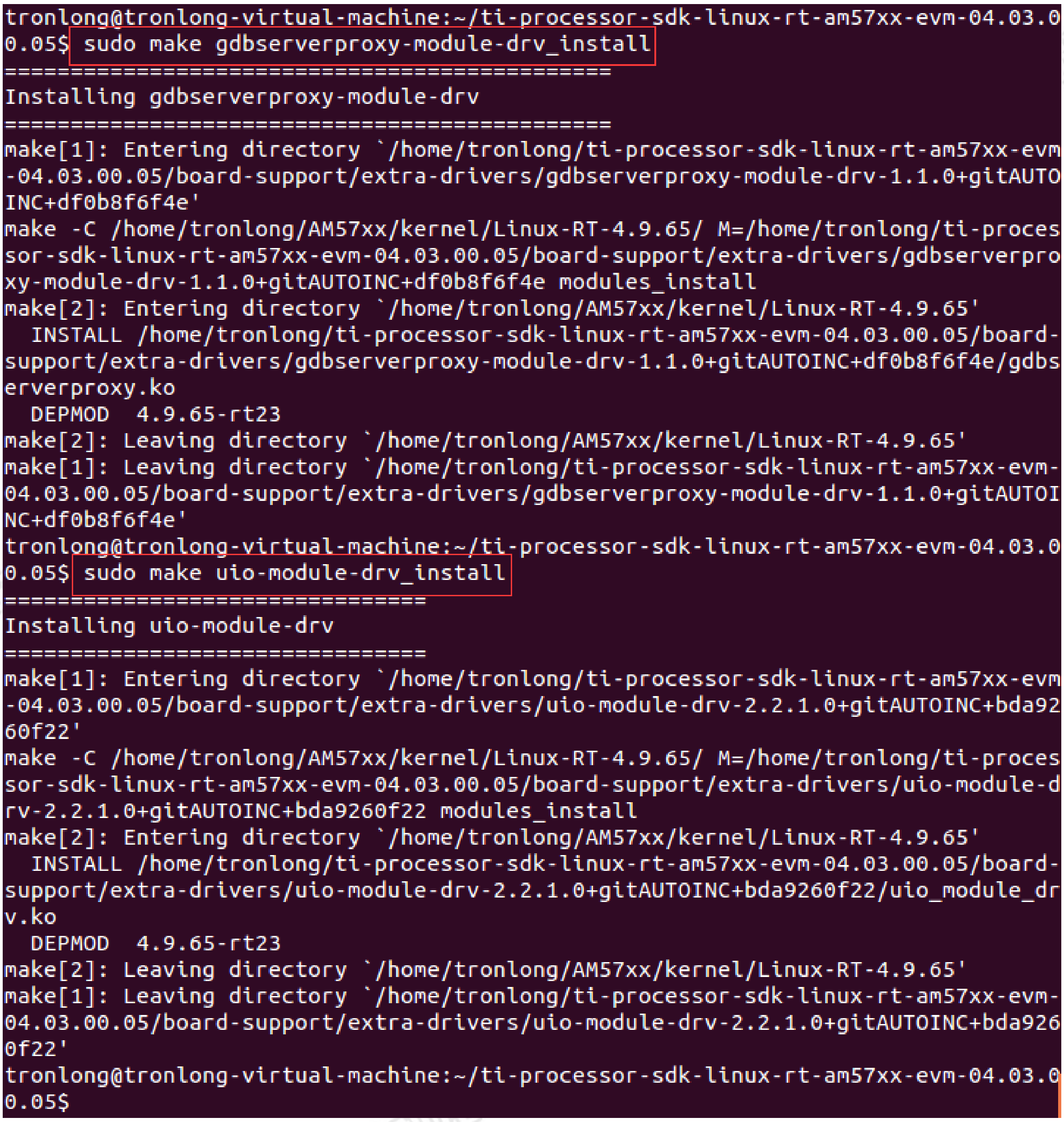
Host# sudo make debugss-module-drv_install

Host# sudo make ti-sgx-ddk-km_install



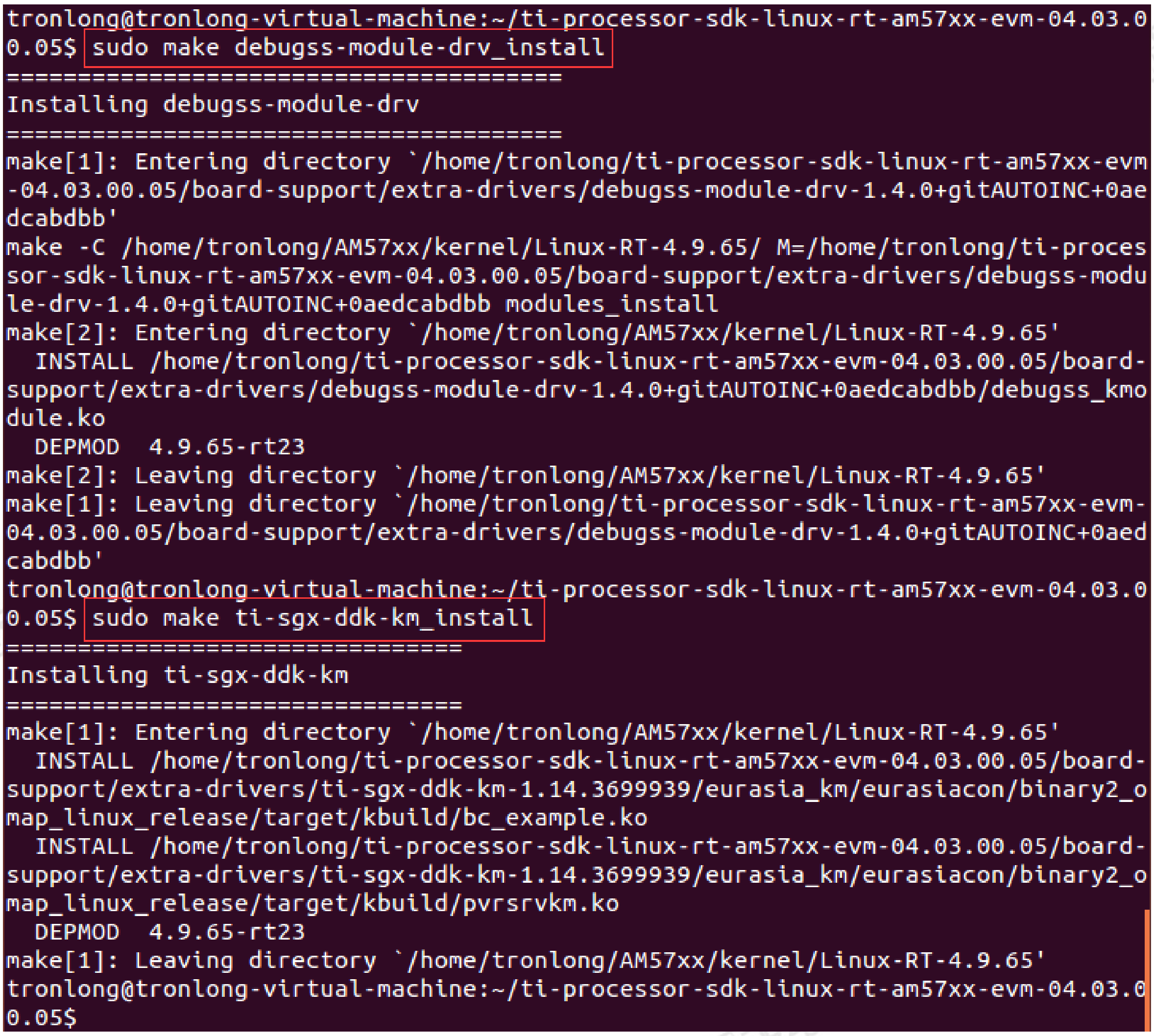
```
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$ sudo make cmem-mod_install
================================
Installing CMEM Module
================================
make[1]: Entering directory `/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65'
  INSTALL /home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-
support/extra-drivers/cmem-mod-4.14.01.00+gitAUTOINC+b687f3c365/src/cmem/module/
cmemk.ko
  DEPMOD  4.9.65-rt23
make[1]: Leaving directory `/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65'
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$ sudo make cryptodev_install
================================
Installing cryptodev-linux
================================
make[1]: Entering directory `/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm
-04.03.00.05/board-support/extra-drivers/cryptodev-module-1.8'
make -C /home/tronlong/AM57xx/kernel/Linux-RT-4.9.65/ SUBDIRS=`pwd` modules_inst
all
make[2]: Entering directory `/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65'
  INSTALL /home/tronlong/ti-processor-sdk-linux-rt-am57xx-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/AM57xx/kernel/Linux-RT-4.9.65'
make[1]: Leaving directory `/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-
04.03.00.05/board-support/extra-drivers/cryptodev-module-1.8'
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$
```

图 33



```
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$ sudo make gdbserverproxy-module-drv_install
==============================================
Installing gdbserverproxy-module-drv
==============================================
make[1]: Entering directory `/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm
-04.03.00.05/board-support/extra-drivers/gdbserverproxy-module-drv-1.1.0+gitAUTO
INC+df0b8f6f4e'
make -C /home/tronlong/AM57xx/kernel/Linux-RT-4.9.65/ M=/home/tronlong/ti-proces
sor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-support/extra-drivers/gdbserverpro
xy-module-drv-1.1.0+gitAUTOINC+df0b8f6f4e modules_install
make[2]: Entering directory `/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65'
  INSTALL /home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-
support/extra-drivers/gdbserverproxy-module-drv-1.1.0+gitAUTOINC+df0b8f6f4e/gdbs
erverproxy.ko
  DEPMOD  4.9.65-rt23
make[2]: Leaving directory `/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65'
make[1]: Leaving directory `/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-
04.03.00.05/board-support/extra-drivers/gdbserverproxy-module-drv-1.1.0+gitAUTOI
NC+df0b8f6f4e'
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$ sudo make uio-module-drv_install
================================
Installing uio-module-drv
================================
make[1]: Entering directory `/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm
-04.03.00.05/board-support/extra-drivers/uio-module-drv-2.2.1.0+gitAUTOINC+bda92
60f22'
make -C /home/tronlong/AM57xx/kernel/Linux-RT-4.9.65/ M=/home/tronlong/ti-proces
sor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-support/extra-drivers/uio-module-d
rv-2.2.1.0+gitAUTOINC+bda9260f22 modules_install
make[2]: Entering directory `/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65'
  INSTALL /home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-
support/extra-drivers/uio-module-drv-2.2.1.0+gitAUTOINC+bda9260f22/uio_module_dr
v.ko
  DEPMOD  4.9.65-rt23
make[2]: Leaving directory `/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65'
make[1]: Leaving directory `/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-
04.03.00.05/board-support/extra-drivers/uio-module-drv-2.2.1.0+gitAUTOINC+bda926
0f22'
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$
```

图 34

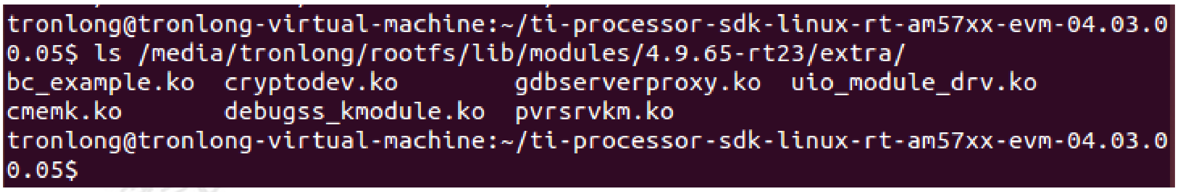


```
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$ sudo make debugss-module-drv_install
=======================================
Installing debugss-module-drv
=======================================
make[1]: Entering directory `/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm
-04.03.00.05/board-support/extra-drivers/debugss-module-drv-1.4.0+gitAUTOINC+0ae
dcabdbb'
make -C /home/tronlong/AM57xx/kernel/Linux-RT-4.9.65/ M=/home/tronlong/ti-proces
sor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-support/extra-drivers/debugss-modu
le-drv-1.4.0+gitAUTOINC+0aedcabdbb modules_install
make[2]: Entering directory `/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65'
  INSTALL /home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-
support/extra-drivers/debugss-module-drv-1.4.0+gitAUTOINC+0aedcabdbb/debugss_kmo
dule.ko
  DEPMOD  4.9.65-rt23
make[2]: Leaving directory `/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65'
make[1]: Leaving directory `/home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-
04.03.00.05/board-support/extra-drivers/debugss-module-drv-1.4.0+gitAUTOINC+0aed
cabdbb'
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$ sudo make ti-sgx-ddk-km_install
================================
Installing ti-sgx-ddk-km
================================
make[1]: Entering directory `/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65'
  INSTALL /home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-
support/extra-drivers/ti-sgx-ddk-km-1.14.3699939/eurasia_km/eurasiacon/binary2_o
map_linux_release/target/kbuild/bc_example.ko
  INSTALL /home/tronlong/ti-processor-sdk-linux-rt-am57xx-evm-04.03.00.05/board-
support/extra-drivers/ti-sgx-ddk-km-1.14.3699939/eurasia_km/eurasiacon/binary2_o
map_linux_release/target/kbuild/pvrsrvkm.ko
  DEPMOD  4.9.65-rt23
make[1]: Leaving directory `/home/tronlong/AM57xx/kernel/Linux-RT-4.9.65'
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$
```

图 35

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/



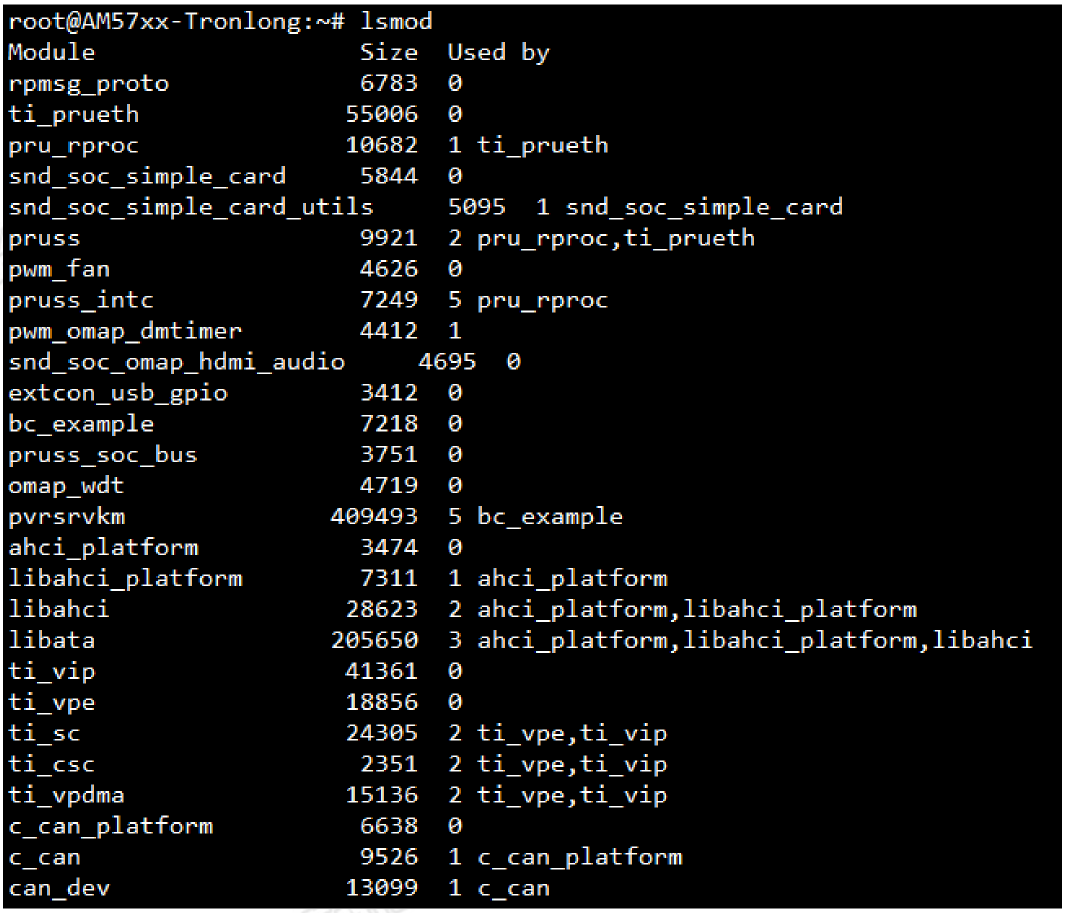
```
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$ ls /media/tronlong/rootfs/lib/modules/4.9.65-rt23/extra/
bc_example.ko  cryptodev.ko       gdbserverproxy.ko  uio_module_drv.ko
cmemk.ko       debugss_kmodule.ko  pvrsrvkm.ko
tronlong@tronlong-virtual-machine:~/ti-processor-sdk-linux-rt-am57xx-evm-04.03.0
0.05$
```

图 36

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@AM57xx-Tronlong:~# lsmod
Module                  Size  Used by
rpmsg_proto             6783  0
ti_prueth              55006  0
pru_rproc              10682  1 ti_prueth
snd_soc_simple_card     5844  0
snd_soc_simple_card_utils     5095  1 snd_soc_simple_card
pruss                   9921  2 pru_rproc,ti_prueth
pwm_fan                 4626  0
pruss_intc              7249  5 pru_rproc
pwm_omap_dmtimer        4412  1
snd_soc_omap_hdmi_audio     4695  0
extcon_usb_gpio         3412  0
bc_example              7218  0
pruss_soc_bus           3751  0
omap_wdt                4719  0
pvrsrvkm              409493  5 bc_example
ahci_platform           3474  0
libahci_platform        7311  1 ahci_platform
libahci                28623  2 ahci_platform,libahci_platform
libata                205650  3 ahci_platform,libahci_platform,libahci
ti_vip                 41361  0
ti_vpe                 18856  0
ti_sc                  24305  2 ti_vpe,ti_vip
ti_csc                  2351  2 ti_vpe,ti_vip
ti_vpdma               15136  2 ti_vpe,ti_vip
c_can_platform          6638  0
c_can                   9526  1 c_can_platform
can_dev                13099  1 c_can
```

图 37

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