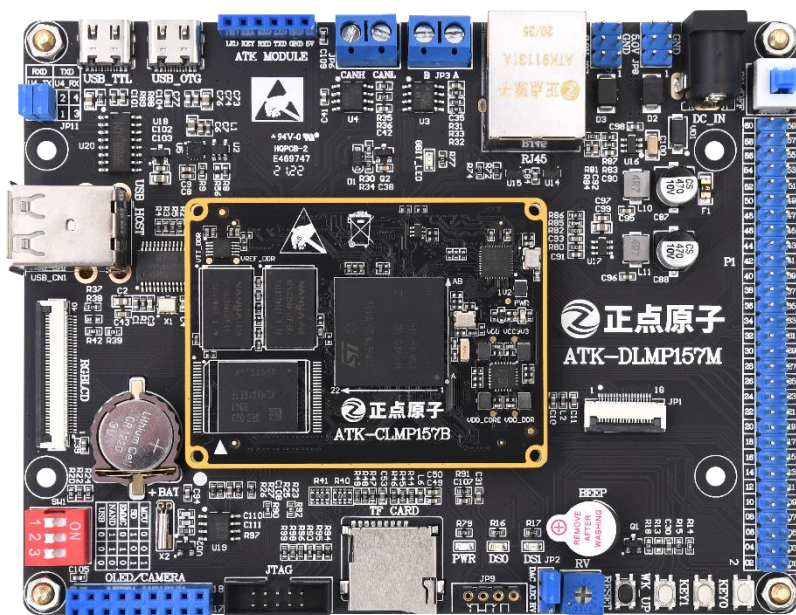
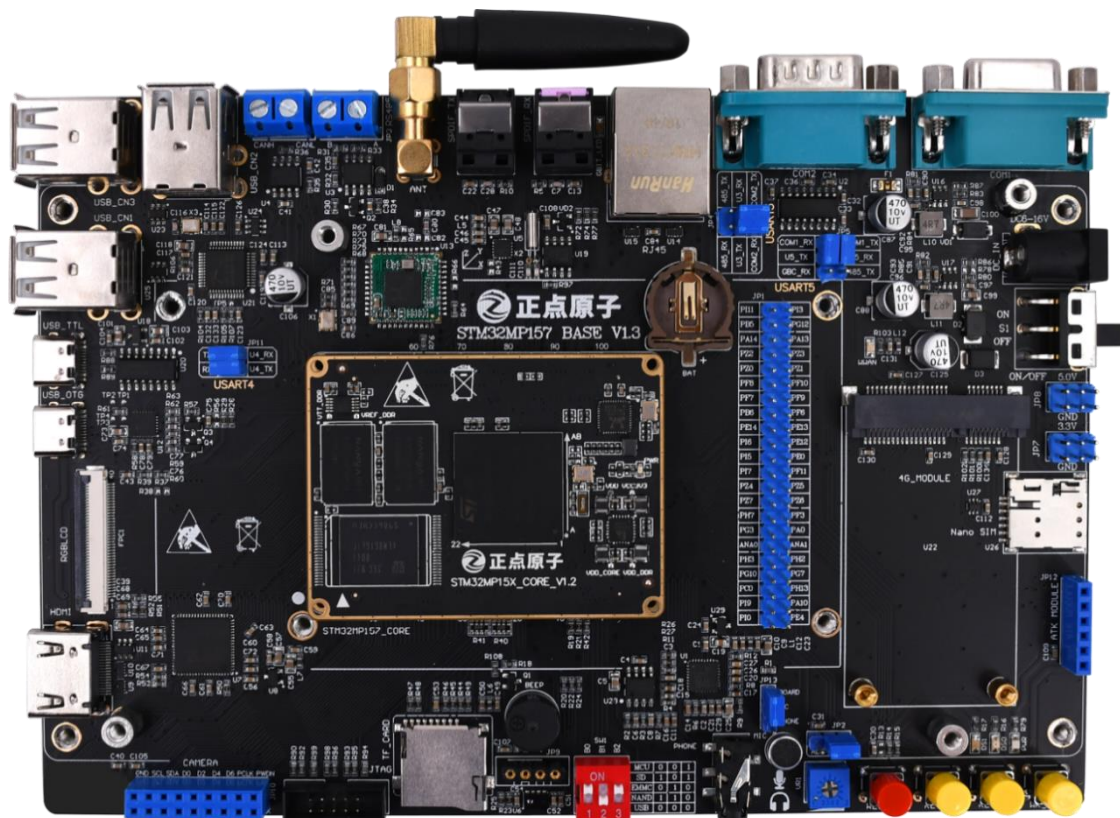


STM32MP157 出厂系统 logo

修改参考手册 V1.2





正点原子公司名称：广州市星翼电子科技有限公司

原子哥在线教学平台：www.yuanzige.com

开源电子网 / 论坛：<http://www.openedv.com/forum.php>

正点原子淘宝店铺：<https://openedv.taobao.com>

正点原子官方网站：www.alientek.com

正点原子 B 站视频：<https://space.bilibili.com/394620890>

电话：020-38271790 传真：020-36773971

请关注正点原子公众号，资料发布更新我们会通知。

请下载原子哥 APP，数千讲视频免费学习，更快更流畅。



扫码关注正点原子公众号



扫码下载“原子哥”APP

文档更新说明

版本	版本更新说明	负责人	校审	发布日期
V1.0	初稿:	正点原子 linux 团队	正点原子 linux 团队	2021.1.26
V1.1	1、添加封面 MP157 MINI 图片	正点原子 linux 团队	正点原子 linux 团队	2021.7.28
V1.2	1、添加第一章 psplash 修改前提说明, V1.2 版本出厂系统不需要修改 psplash	正点原子 linux 团队	正点原子 linux 团队	2021.8.6

目录

前言	5
第一章 修改 psplash.....	6
1.1 图片要求.....	6
1.2 准备文件.....	6
1.3 搭建编译环境.....	6
1.4 编译	7
1.5 验证效果.....	8
第二章 修改内核的 logo	8
2.1 图片要求.....	8
2.2 准备文件.....	9
2.3 修改内核 logo	9
2.4 编译出厂内核.....	10
2.5 验证	11



前言

文档目的: 修改正点原子出厂的 psplash 和内核的 logo。

文档基于的操作环境:

主机: Windows10

虚拟机: Ubuntu18.04

开发板系统: 出厂系统

文档使用到的 gcc 编译器和交叉编译器:

用户能力基础: 熟悉编译出厂内核源码、搭建 TFTP 网络环境

文档笔误反馈: QQ: 2765961790

第一章 修改 psplash

前提说明: 为了方便用户使用, 不需要修改太多 logo, 在 V1.2 版本的出厂系统上去掉了 psplash, 如果是 V1.2 之后的版本 (包括 V1.2), 就可以不用修改这个 psplash, 只修改内核 logo 就行。V1.1 版本的可以参考本文档修改 psplash。

在出厂系统上执行 `cat /etc/version` 指令可以查看当前系统版本号。

```
root@ATK-MP157:~# cat /etc/version
文件系统版本
版本: v1.0 ~ v1.1
无记录
版本: v1.2
日期: 2021年7月23日
修改内容:
1. 统一Qt版本为5.12.9
2. 其他优化, 详细请看快速体验手册v1.4.
root@ATK-MP157:~#
```

图 1.1-1 出厂系统为 V1.2 版本

1.1 图片要求

图片像素不能超过 RGB 屏幕像素, 最终生成的文件大小最好也不超过 100K。可以先用原子提供的图片做一遍测试。

1.2 准备文件

网盘路径: 开发板光盘 A-基础资料\1、程序源码\12、psplash 源码

在 Ubuntu 中新建一个名为 psplash-drm 的文件夹, 将 psplash-drm.tar.bz2 拷贝到 psplash-drm 路径下进行解压。

解压 psplash-drm.tar.bz2

```
tar vxvf psplash-drm.tar.bz2
```

```
alientek@ubuntu18:~$ mkdir psplash-drm
alientek@ubuntu18:~$ cd psplash-drm/
alientek@ubuntu18:~/psplash-drm$
alientek@ubuntu18:~/psplash-drm$
alientek@ubuntu18:~/psplash-drm$ ls
psplash-drm.tar.bz2
alientek@ubuntu18:~/psplash-drm$ tar vxvf psplash-drm.tar.bz2
./
./README.txt
./image_header.h
./alientek_428x214_24bits.png
./psplash-drm-quit
./basic_splash_drm.c
./alientek_428x214_24bits_rotation.png
./plymouth
./psplash-drm-start.service
./Makefile
alientek@ubuntu18:~/psplash-drm$ ls
alientek_428x214_24bits.png    basic_splash_drm.c  Makefile  psplash-drm-quit  psplash-drm.tar.bz2
alientek_428x214_24bits_rotation.png  image_header.h    plymouth  psplash-drm-start.service  README.txt
```

图 1.2-1 解压 psplash-drm 压缩包

1.3 搭建编译环境

解压完后, 执行 `cat` 指令查看 README.txt 帮助文件, 可以看到需要安装 libgtk2.0-dev 库。执行以下指令安装此库。

```
sudo apt-get install libgtk2.0-dev
```


要使用出厂系统交叉编译器才能编译出能在出厂系统上执行的 psplash-drm 文件。因此要先安装出厂系统专门的交叉编译器,具体安装方法可以参考《【正点原子】STM32MP157 快速体验》。

网盘路径: [开发板光盘 A-基础资料\5、开发工具\1、交叉编译器\st-example-image-qtwayland-openstlinux-weston-stm32mp1-x86_64-toolchain-3.1-snapshot.sh](#)

将上述的 sh 脚本文件拷贝到 Ubuntu, 执行以下指令赋予可执行权限。

```
chmod +x st-example-image-qtwayland-openstlinux-weston-stm32mp1-x86_64-toolchain-3.1-snapshot.sh
```

然后执行此脚本文件直接安装,安装过程中会询问安装的目录,我们默认即可,按 Enter 继续。

```
./st-example-image-qtwayland-openstlinux-weston-stm32mp1-x86_64-toolchain-3.1-snapshot.sh
```

安装完成后,安装的交叉编译工具链都会安装在/opt/目录下。

```
ls /opt/st/
```

1.4 编译

回到 psplash 目录下, **使能环境变量**后才能编译(注意切换终端时需要重新使能这个环境变量才能编译)

```
source /opt/st/stm32mp1/3.1-snapshot/environment-setup-cortexa7t2hf-neon-vfpv4-ostl-linux-gnueabi
```

执行 display 指令可以查看我们的 logo 图片。

```
display alientek_428x214_24bits.png
```



图 1.4-1 使用 display 指令查看图片

根据 README.txt 帮助文件的提示,依次执行以下指令。

```
make generate_header
```

```
make psplash
```

```
alientek@ubuntu18:~/psplash-drm$ make generate_header
alientek@ubuntu18:~/psplash-drm$ make psplash
arm-ostl-linux-gnueabi-gcc -mthumb -mfpu=neon-vfpv4 -mfloat-abi=hard -mcpu=cortex-a7 --sysroot=/opt/st/stm32mp1/3.1-snapshot/sysroots/cortexa7t2hf-neon-vfpv4-ostl-linux-gnueabi -O2 -pipe -g -feliminate-unused-debug-types -Wl, -O1 -Wl,--hash-style=gnu -Wl,--as-needed -o psplash-drm basic_splash_drm.c -I. `pkg-config --cflags --libs libdrm` -Wall -Os
```

图 1.4-2 编译 psplash

编译成功后会生成 psplash-drm 文件。

```
alientek@ubuntu18:~/psplash-drm$ ls
alientek_428x214_24bits.png  image_header.h  psplash-drm  psplash-drm.tar.bz2
alientek_428x214_24bits_rotation.png  Makefile  psplash-drm-quit  README.txt
basic_splash_drm.c  plymouth  psplash-drm-start.service
```

图 1.4-3 编译得到的 psplash-drm 文件

1.5 验证效果

在开发板文件系统的/usr/bin 目录下有一个 psplash-drm 文件, 我们将刚刚在虚拟机中编译得到的 psplash-drm 文件拷贝到开发板的/usr/bin 目录下替换掉原本的 psplash-drm, 然后赋予 psplash-drm 文件权限, 然后直接执行 psplash-drm 指令即可执行。

```
chmod 777 psplash-drm  
psplash-drm
```

```
root@ATK-stm32mp1:/usr/bin# chmod 777 psplash-drm  
root@ATK-stm32mp1:/usr/bin# psplash-drm  
using card '/dev/dri/card0'  
mode for connector 32 is 1024x600  
exiting  
root@ATK-stm32mp1:/usr/bin#
```

图 1.5-1 执行 psplash-drm

此时注意看开发板屏幕, 会出现我们已经修改好的 logo。



图 1.5-2 修改效果

第二章 修改内核的 logo

本章需要用户提前编译好出厂内核源码, 如果没编译过, 请先按照《【正点原子】STM32MP1 嵌入式 Linux 驱动开发指南》第 17.1 Linux 内核初次编译小节的内容编译出厂内核源码。

2.1 图片要求

图片像素不能超过 RGB 屏幕像素, 最终生成的文件大小最好也不超过 100K。可以先用原子提供的图片做一遍测试。

2.2 准备文件

网盘路径: [开发板光盘 A-基础资料\1、程序源码\12、psplash 源码\alientek_428x214_24bits.png](#)

2.3 修改内核 logo

在编译好的出厂内核源码的 `drivers/video/logo` 目录下可以看到正点原子的 logo。如下。

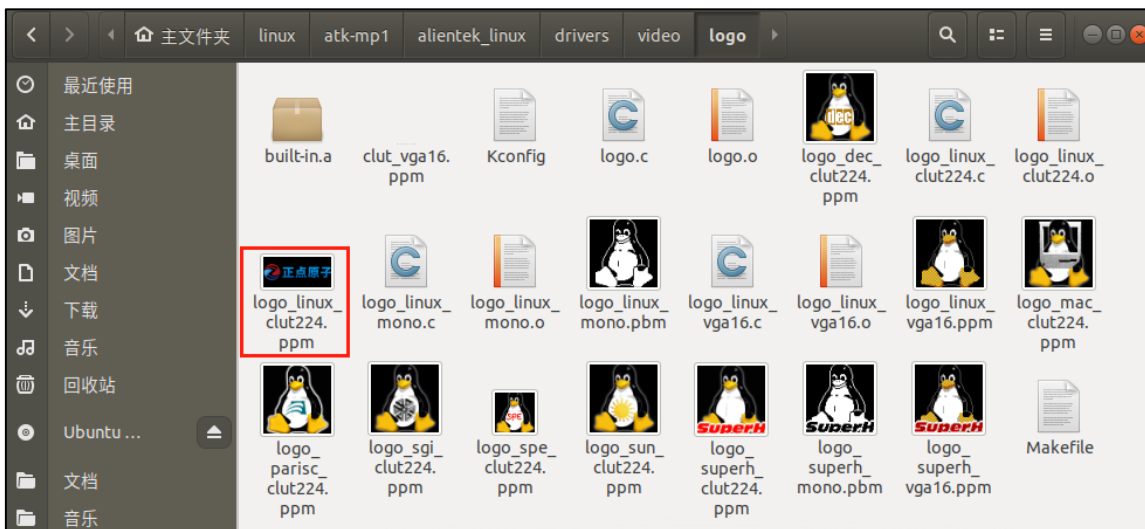


图 2.3-1 正点原子的内核 logo

首先将 `logo_linux_clut224.ppm` 文件重命名为 `logo_linux_clut224_V1.ppm`。

```
mv logo_linux_clut224.ppm logo_linux_clut224_V1.ppm
```

安装图片转换工具 `netpbm`。

```
sudo apt-get install netpbm
```

将网盘中的 `alientek_428x214_24bits.png` 文件拷贝到出厂内核源码的 `drivers/video/logo` 目录下。

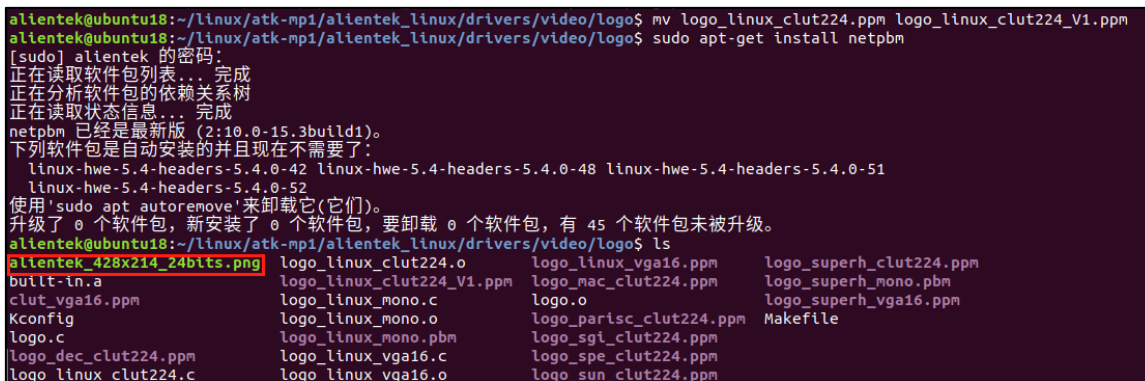


图 2.3-2 准备工作

使用 `netpbm` 工具转换图片, 执行以下指令。

```
pngtopnm alientek_428x214_24bits.png > alientek_428x214_24bits.pnm
```

```
pnmquant 224 alientek_428x214_24bits.pnm > alientek_428x214_24bits_224.pnm
```

```
pnmtoplainpnm alientek_428x214_24bits_224.pnm > logo_linux_clut224.ppm
```

```
alientek@ubuntu18:~/linux/atk-mp1/alientek_linux/drivers/video/logo$ pngtopnm alientek_428x214_24bits.png > alientek_428x214_24bits.pnm
alientek@ubuntu18:~/linux/atk-mp1/alientek_linux/drivers/video/logo$ pnmquant 224 alientek_428x214_24bits.pnm > alientek_428x214_24bits_224.pnm
pnmcolormap: making histogram...
pnmcolormap: 2429 colors found
pnmcolormap: choosing 224 colors...
pnmrenmap: 222 colors found in colormap
alientek@ubuntu18:~/linux/atk-mp1/alientek_linux/drivers/video/logo$ pnmtoplainpnm alientek_428x214_24bits_224.pnm > logo_linux_clut224.ppm
```

图 2.3-3 转换图片格式

分析:

```
pngtopnm alientek_428x214_24bits.png > alientek_428x214_24bits.pnm
```

这一步是将准备好的 alientek_428x214_24bits.png 转换成 alientek_428x214_24bits.pnm, 即转换成 pnm 格式。

```
pnmquant 224 alientek_428x214_24bits.pnm > alientek_428x214_24bits_224.pnm
```

这一步是将 alientek_428x214_24bits.pnm 图片的最大颜色数量改为 224。注意, 这里前后名字不能一样, 可以理解为要另存为。否则会报错。

```
pnmtoplainpnm alientek_428x214_24bits_224.pnm > logo_linux_clut224.ppm
```

这一步是将 pnm 格式图片转换成 ppm 格式, 顺便将其重命名为 logo_linux_clut224.ppm, 我们是在内核源码的 drivers/video/logo 目录下直接进行的, 所以此步就直接替换内核原来的 logo (之前的 logo_linux_clut224.ppm 我们已经重命名了 logo_linux_clut224_V1.ppm)

如果不是在 drivers/video/logo 目录下进行的转换, 需要将转换好的 logo_linux_clut224.ppm 拷贝到 drivers/video/logo 目录下替换掉原先的 logo_linux_clut224.ppm。

2.4 编译出厂内核

修改完后, 返回到内核源码的顶层目录, 执行 stm32mp157d_atk.sh 编译脚本。之前没有创建这个脚本的话需要自行创建。

```
#!/bin/sh
```

```
make ARCH=arm CROSS_COMPILE=arm-none-linux-gnueabi- distclean
```

```
make ARCH=arm CROSS_COMPILE=arm-none-linux-gnueabi- stm32mp1_atk_defconfig
```

```
make ARCH=arm CROSS_COMPILE=arm-none-linux-gnueabi- menuconfig
```

```
make ARCH=arm CROSS_COMPILE=arm-none-linux-gnueabi- uImage dtbs LOADADDR=0XC2000040 -j16
```

然后执行编译。

```
./stm32mp157d_atk.sh
```

```
alientek@ubuntu18:~/linux/atk-mp1/alientek_linux$ ./stm32mp157d_atk.sh
#
# No change to .config
#
scripts/kconfig/mconf Kconfig

*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.
```

图 2.4-1 编译内核源码

编译完成会得到 uImage 和 stm32mp157d-atk.dtb。

```
Load Address: c2000040
Entry Point:  c2000040
Kernel: arch/arm/boot/uImage is ready
alientek@ubuntu18:~/linux/atk-mp1/alientek_linux$ ls arch/arm/boot/uImage -l
-rw-r--r-- 1 alientek alientek 8281608 1月  5 10:40 arch/arm/boot/uImage
alientek@ubuntu18:~/linux/atk-mp1/alientek_linux$ ls arch/arm/boot/dts/stm32mp157d-atk.dtb -l
-rw-r--r-- 1 alientek alientek 74594 1月  4 11:38 arch/arm/boot/dts/stm32mp157d-atk.dtb
```

图 2.4-2 编译得到的文件

2.5 验证

我们可以先通过 TFTP 网络挂载的方法来体验下我们修改好的内核。如果没做过 TFTP 挂载, 需要先看《【正点原子】STM32MP157 网络环境 TFTP&NFS 搭建手册》的相关内容。将编译得到的内核和设备树文件拷贝到 TFTP 工作目录。

```
cp arch/arm/boot/uImage ~/linux/tftp/
```

```
cp arch/arm/boot/dts/stm32mp157d-atk.dtb ~/linux/tftp/
```

```
alientek@ubuntu18:~/linux/atk-mp1/alientek_linux$ cp arch/arm/boot/uImage ~/linux/tftp/
alientek@ubuntu18:~/linux/atk-mp1/alientek_linux$ cp arch/arm/boot/dts/stm32mp157d-atk.dtb ~/linux/tftp/
alientek@ubuntu18:~/linux/atk-mp1/alientek_linux$
alientek@ubuntu18:~/linux/atk-mp1/alientek_linux$ ls ~/linux/tftp/
stm32mp157d-atk.dtb  test.c  uImage
```

图 2.5-1 拷贝内核和设备树文件到 TFTP 工作目录

设置开发板相关 IP 信息后, 使用 tftp 挂载并启动修改好的内核和设备树。

```
setenv bootcmd 'tftp c2000000 uImage;tftp c4000000 stm32mp157d-atk.dtb;bootm c2000000 - c4000000'
```

```
saveenv
```

```
boot
```

```
Hit any key to stop autoboot: 0
STM32MP> setenv ipaddr 192.168.1.7
STM32MP> setenv gatewayip 192.168.1.1
STM32MP> setenv netmask 255.255.255.0
STM32MP> setenv serverip 192.168.1.208
STM32MP> saveenv
Saving Environment to MMC... Writing to MMC(1)... OK
STM32MP> ping 192.168.1.208
ethernet@5800a000 Waiting for PHY auto negotiation to complete..... done
Using ethernet@5800a000 device
host 192.168.1.208 is alive
STM32MP> setenv bootcmd 'tftp c2000000 uImage;tftp c4000000 stm32mp157d-atk.dtb;bootm c2000000 - c4000000'
STM32MP> saveenv
Saving Environment to MMC... Writing to redundant MMC(1)... OK
STM32MP> boot
Using ethernet@5800a000 device
TFTP from server 192.168.1.208; our IP address is 192.168.1.7
Filename 'uImage'.
Load address: 0xc2000000
Loading: #####
#####
```

图 2.5-2 TFTP 启动

启动时, 开发板屏幕效果如图所示。



图 2.5-3 显示效果

可以看到,开发板启动时,屏幕上会有我们刚刚修改的 logo 显示。但是还有一些内核打印信息出现在开发板屏幕上,而且开发板会重启。这是因为我们没有加载文件系统导致的,我们这里只是验证下是否修改成功,后面可以通过加载文件系统或者烧录内核、设备树到开发板上来解决。