

## U-Boot 编译方法

## Revision History

Draft Date	Revision No.	Description
2019/03/26	V1.1	1. 内容更新，添加支持 TL570x-EVM 开发板。
2018/07/01	V1.0	1. 初始版本。

## 目 录

1 前言 .....	错误!未定义书签。
2 U-Boot-2017.01 编译方法 .....	3
2.1 安装 U-Boot 源码 .....	3
2.2 清理 U-Boot .....	5
2.3 编译 U-Boot .....	5
2.4 从 SD 系统启动卡启动 U-Boot .....	6
3 U-Boot 环境变量还原与保存 .....	7
更多帮助 .....	9

## 1 实验说明

- 操作环境：Windows 7/10 64bit；VMware14.1.1；Ubuntu 14.04.3 64bit。
- U-Boot-2017.01 版本适用性

表 1

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

- 清理命令区别说明如下：

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。

- U-Boot 启动过程可以分启动第一阶段和启动第二阶段。编译所得到的 MLO 和 u-boot.img 文件，分别是 U-Boot 启动第一阶段和启动第二阶段镜像文件。

## 2 U-Boot-2017.01 编译方法

### 2.1 安装 U-Boot 源码

Ubuntu 上执行如下命令创建 U-Boot 源码安装目录“/home/tronlong/AM57xx/U-Boot/U-Boot-2017.01”，如下图所示：

```
Host# mkdir -p /home/tronlong/AM57xx/U-Boot/U-Boot-2017.01
```

```
tronlong@tronlong-virtual-machine:~$ mkdir -p /home/tronlong/AM57xx/U-Boot/U-Boot-2017.01
tronlong@tronlong-virtual-machine:~$
```

图 1

U-Boot 源码为光盘“U-Boot\U-Boot-2017.01\src\u-boot-[Git 系列号]-[版本号].tar.gz”，Git 系列号与版本号以实际情况为准。U-Boot 支持特性可以查看“\U-Boot\am57xx-uboot-feature-support.xls”文件。

将 U-Boot 源码压缩文件复制到“/home/tronlong/AM57xx”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 -zxvf u-boot-2017.01-gd3af627-v1.0.tar.gz -C U-Boot/U-Boot-2017.01/
```

```
tronlong@tronlong-virtual-machine:~$ cd /home/tronlong/AM57xx
tronlong@tronlong-virtual-machine:~/AM57xx$ ls u-boot-2017.01-gd3af627-v1.0.tar.gz
u-boot-2017.01-gd3af627-v1.0.tar.gz
tronlong@tronlong-virtual-machine:~/AM57xx$ tar -zxvf u-boot-2017.01-gd3af627-v1.0.tar.gz -C U-Boot/U-Boot-2017.01/
```

图 2

```
tools/scripts/
tools/scripts/define2mk.sed
tools/socfpgaimage.c
tools/tbot/
tools/tbot/README
tools/tbot/README-ToDo
tools/tbot/README.create_a_new_testcase
tools/tbot/README.install
tools/ublimage.c
tools/ublimage.h
tools/ubsha1.c
tools/vybridimage.c
tools/xway-swap-bytes.c
tools/zynqimage.c
tools/zynqmpimage.c
tronlong@tronlong-virtual-machine:~/AM57xx$
```

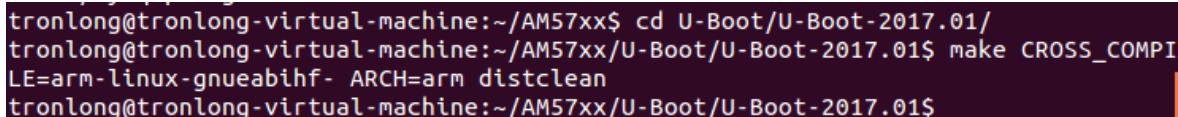
图 3

## 2.2 清理 U-Boot

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 U-Boot 源码安装目录，执行 U-Boot 清理命令。

Host# cd U-Boot/U-Boot-2017.01/

Host# make CROSS\_COMPILE=arm-linux-gnueabi- ARCH=arm distclean



```
tronlong@tronlong-virtual-machine:~/AM57xx$ cd U-Boot/U-Boot-2017.01/
tronlong@tronlong-virtual-machine:~/AM57xx/U-Boot/U-Boot-2017.01$ make CROSS_COMPILE=arm-linux-gnueabi- ARCH=arm distclean
tronlong@tronlong-virtual-machine:~/AM57xx/U-Boot/U-Boot-2017.01$
```

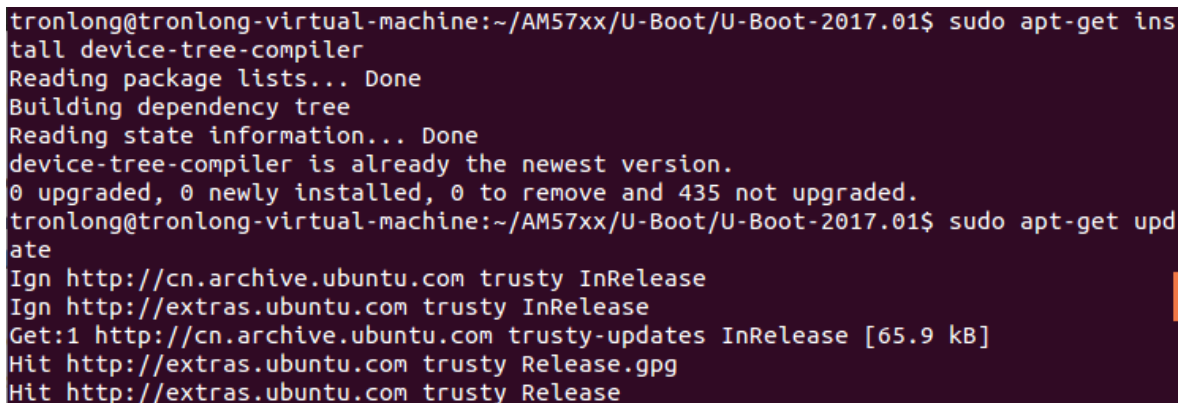
图 4

## 2.3 编译 U-Boot

执行如下指令，安装 device-tree-compiler 工具，此工具用于 U-Boot 的编译。

Host# sudo apt-get install device-tree-compiler

Host# sudo apt-get update



```
tronlong@tronlong-virtual-machine:~/AM57xx/U-Boot/U-Boot-2017.01$ sudo apt-get install device-tree-compiler
Reading package lists... Done
Building dependency tree
Reading state information... Done
device-tree-compiler is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 435 not upgraded.
tronlong@tronlong-virtual-machine:~/AM57xx/U-Boot/U-Boot-2017.01$ sudo apt-get update
Ign http://cn.archive.ubuntu.com trusty InRelease
Ign http://extras.ubuntu.com trusty InRelease
Get:1 http://cn.archive.ubuntu.com trusty-updates InRelease [65.9 kB]
Hit http://extras.ubuntu.com trusty Release.gpg
Hit http://extras.ubuntu.com trusty Release
```

图 5

执行以下命令进行 U-Boot 编译：

Host# make CROSS\_COMPILE=arm-linux-gnueabi- ARCH=arm am57xx\_evm\_defconfig  
all

```
tronlong@tronlong-virtual-machine:~/AM57xx/U-Boot/U-Boot-2017.01$ make CROSS_COMPILE=arm-linux-gnueabihf- ARCH=arm am57xx_evm_defconfig all
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
```

图 6

编译完成后会在 U-Boot 源码目录下生成 U-Boot 镜像文件 MLO、u-boot.img，如下图所示：

```
OBJCOPY spl/u-boot-spl-nodtb.bin
FDTGREP spl/u-boot-spl.dtb
CAT spl/u-boot-spl-dtb.bin
COPY spl/u-boot-spl.bin
MKIMAGE MLO
COPY u-boot.dtb
MKIMAGE u-boot-dtb.img
./scripts/check-config.sh u-boot.cfg \
    ./scripts/config_whitelist.txt . 1>&2
tronlong@tronlong-virtual-machine:~/AM57xx/U-Boot/U-Boot-2017.01$ ls
api      drivers  MAINTAINERS  System.map      u-boot-dtb.bin
arch     dts      Makefile     test            u-boot-dtb.img
board    examples MLO          TISDK-README   u-boot.lds
cmd      fs       net          tools           u-boot.map
common   include  post         u-boot         u-boot-nodtb.bin
config.mk Kbuild   README      u-boot.bin      u-boot.srec
configs  Kconfig  scripts     u-boot.cfg      u-boot.sym
disk     lib      snapshot.commit u-boot.cfg.configs
doc      Licenses spl          u-boot.dtb
tronlong@tronlong-virtual-machine:~/AM57xx/U-Boot/U-Boot-2017.01$
```

图 7

## 2.4 从 SD 系统启动卡启动 U-Boot

将编译生成的 MLO 和 u-boot.img 文件拷贝到 SD 系统启动卡 boot 分区下，替换掉 SD 系统启动卡 boot 分区下对应的文件。开发板断电，将替换好文件的 SD 系统启动卡插入开发板 Micro SD 卡槽，根据底板丝印将开发板的拨码开关拨为 SD/MMC1 模式，此为 Micro SD 卡启动模式。

使用 Micro USB 线连接 PC 和开发板的 USB 转串口调试口，然后将开发板上电，串口终端会有类似以下的启动信息打印，说明 SD 系统启动卡启动 U-Boot 成功，如下图所示：

```
U-Boot SPL 2017.01-00475-g8fc2019 (Mar 07 2019 - 10:05:20)
DRA722-GP ES2.1
Trying to boot from MMC1
reading u-boot.img
reading u-boot.img
reading u-boot.img
reading u-boot.img

U-Boot 2017.01-00475-g8fc2019 (Mar 07 2019 - 10:05:20 +0800)

CPU : DRA722-GP ES2.1
Model: TI AM572x EVM Rev A3
Board: Tronlong am57xx_evm boards
DRAM: 1 GiB
MMC: OMAP SD/MMC: 0, OMAP SD/MMC: 1
i2c_write: error waiting for data ACK (status=0x116)
i2c_write: error waiting for data ACK (status=0x116)
i2c_write: error waiting for data ACK (status=0x116)
SCSI: SATA link 0 timeout.
AHCI 0001.0300 32 slots 1 ports 3 Gbps 0x1 impl SATA mode
flags: 64bit ncq stag pm led clo only pmp pio slum part ccc apst
scanning bus for devices...
Found 0 device(s).
Net: could not get PHY for ethernet@48484000: addr 1
eth0: ethernet@48484000
Hit any key to stop autoboot: 0
=>
```

图 8

### 3 U-Boot 环境变量还原与保存

进入 U-Boot 命令行，执行如下命令，还原为系统默认设置的 U-Boot 环境变量。

U-Boot> env default -a

```
=> env default -a
## Resetting to default environment
=>
```

图 9

执行如下命令，保存更改，如下图所示：

U-Boot> saveenv

```
=> saveenv
Saving Environment to MMC...
Writing to MMC(1)... done
=>
```

图 10



任何 U-Boot 环境变量的修改，如果需要得到保存，都需要执行 saveenv 命令。执行 saveenv 命令后，所有 U-Boot 环境变量将会保存到 eMMC 中。

修改环境变量后，执行 boot 命令，即可使用更改后的 U-Boot 环境变量启动 Linux 系统，如下图所示：

U-Boot> boot

```
Hit any key to stop autoboot: 0
=>
=> boot
switch to partitions #0, OK
mmc0 is current device
SD/MMC found on device 0
reading boot.scr
** Unable to read file boot.scr **
reading uEnv.txt
574 bytes read in 3 ms (186.5 KiB/s)
Loaded env from uEnv.txt
Importing environment from mmc0 ...
switch to partitions #0, OK
mmc0 is current device
SD/MMC found on device 0
5883024 bytes read in 528 ms (10.6 MiB/s)
166472 bytes read in 37 ms (4.3 MiB/s)
## Flattened Device Tree blob at 88000000
   Booting using the fdt blob at 0x88000000
   Loading Device Tree to 8ffd4000, end 8ffffa47 ... OK

Starting kernel ...
```

图 11



## 更多帮助

销售邮箱: [sales@tronlong.com](mailto:sales@tronlong.com)

技术邮箱: [support@tronlong.com](mailto:support@tronlong.com)

创龙总机: 020-8998-6280

技术热线: 020-3893-9734

创龙官网: [www.tronlong.com](http://www.tronlong.com)

技术论坛: [www.51ele.net](http://www.51ele.net)

线上商城: <https://tronlong.taobao.com>