

武汉纤效电子科技 openwrt sdk 编译使用说明

第一部分：搭建编译环境（编译在 linux 环境下）

1、在编译 openwrt 固件之前需要搭建好编译环境，安装好如下一些库：

```
sudo apt-get install gcc
sudo apt-get install g++
sudo apt-get install binutils
sudo apt-get install patch
sudo apt-get install bzip2
sudo apt-get install flex
sudo apt-get install make
sudo apt-get install gettext \ pkg-config
sudo apt-get install unzip
sudo apt-get install zlib1g-dev
sudo apt-get install libc6-dev
sudo apt-get install subversion
sudo apt-get install libncurses5-dev
sudo apt-get install gawk \ sharutils
sudo apt-get install curl
sudo apt-get install libxml-parser-perl
sudo apt-get install ocaml-nox
sudo apt-get install build-essential
sudo apt-get install gcc-multilib
sudo apt-get install git-core
sudo apt-get install bison
sudo apt-get install autoconf
sudo apt-get install texinfo
sudo apt-get install ncurses-term
sudo apt-get install asciidoc
sudo apt-get install libz-dev
```

在 ubuntu 下，同时按住 Ctrl + Alt + T，调出终端 输入以上命令安装依赖库，以 gcc 为例如下：



```
zhang@ubuntu:~$ sudo apt-get install gcc
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
gcc 已经是最新版 (4:5.3.1-1ubuntu1)。
下列软件包是自动安装的并且现在不需要了：
  libgif7:i386 libp11-kit-gnome-keyring:i386
  linux-image-extra-4.4.0-21-generic linux-image-extra-4.4.0-38-generic
  wine-gecko2.21 wine-gecko2.21:i386 wine-mono0.0.8
使用 'sudo apt autoremove' 来卸载它(它们)。
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 40 个软件包未被升级。
zhang@ubuntu:~$
```

***注意：**安装时看看打印信息确认库是否安装成功，某些库安装失败会导致后面的编译出错。以上截图显示的 gcc 库已经安装了；如果显示安装失败，请重新安装，直到安装完所有库。搭建好环境是关键所在，一定确保环境 ok。

第二部分：下载 OpenWrt 源码并编译

1、从 openwrt 网站上下载 openwrt 源码： (<https://github.com/openwrt/openwrt>)

```
git clone https://github.com/openwrt/openwrt.git
```

2、下载完成后，将会出现 openwrt/这个目录，如下：

```
zhang@ubuntu:~$ ls
examples.desktop  openwrt15.05-old      unknown      公共的      下载
gittest           openwrt sdk编译使用说明.odt  VirtualBox  VMs        模板      音乐
minicom.log       qx_qsdk               win-share    视频      桌面
openwrt           regdb.txt             zhang       图片
openwrt15.05      trunk                 zp-openwrt  文档
zhang@ubuntu:~$
```

3、进入到 openwrt/目录下作如下操作：

```
cd openwrt/
```

```
./scripts/feeds update -a （确保pc网络畅通，更新扩展）
```

```
./scripts/feeds install -a
```

```
zhang@ubuntu:~$ cd openwrt
zhang@ubuntu:~/openwrt$ ls
BSDmakefile  dl          include    Makefile   scripts
config       docs       key-build  package    target
Config.in    feeds.conf.default  key-build.pub  README     toolchain
core         files      LICENSE    rules.mk   tools
zhang@ubuntu:~/openwrt$ ./scripts/feeds update -a
Updating feed 'packages' from 'https://github.com/openwrt/packages.git;for-15.05'
...
正克隆到 './feeds/packages'...
remote: Counting objects: 2695, done.
remote: Compressing objects: 100% (2271/2271), done.
接收对象中： 39% (1052/2695), 524.00 KiB | 39.00 KiB/s
```

4、输入这个命令后耐心等待更新完所有的扩展，更新完成后可以看到如下打印信息，然后就可以执行下一条命令来安装扩展：

```
Updating feed 'management' from 'https://github.com/openwrt-management/packages.git;for-15.05' ...
正克隆到 './feeds/management'...
remote: Counting objects: 36, done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 36 (delta 5), reused 23 (delta 0), pack-reused 0
展开对象中： 100% (36/36), 完成。
检查连接... 完成。
Create index file './feeds/management.index'
Collecting package info: done
Collecting target info: done
zhang@ubuntu:~/openwrt$ ./scripts/feeds install -a
```

*注意：在使用./scripts/feeds update -a 的时候可能出现错误，检查网络畅通后再多尝试几次。

5、检查编译环境是否搭建完成使用 make defconfig，如下所示：

```
zhang@ubuntu:~/openwrt15.05/source/openwrt$ make defconfig
Checking 'working-make'... ok.
Checking 'case-sensitive-fs'... ok.
Checking 'gcc'... ok.
Checking 'working-gcc'... ok.
Checking 'g++'... ok.
Checking 'working-g++'... ok.
Checking 'ncurses'... ok.
Checking 'zlib'... ok.
Checking 'libssl'... ok.
Checking 'tar'... ok.
Checking 'find'... ok.
Checking 'bash'... ok.
Checking 'patch'... ok.
Checking 'diff'... ok.
Checking 'cp'... ok.
Checking 'seq'... ok.
Checking 'awk'... ok.
Checking 'grep'... ok.
Checking 'getopt'... ok.
Checking 'stat'... ok.
Checking 'md5sum'... ok.
Checking 'unzip'... ok.
Checking 'bzip2'... ok.
Checking 'wget'... ok.
Checking 'perl'... ok.
Checking 'python'... ok.
Checking 'svn'... ok.
Checking 'git'... ok.
Checking 'file'... ok.
Checking 'openssl'... ok.
Checking 'ldconfig-stub'... ok.
Collecting package info: done
Collecting target info: done
#
# configuration written to .config
#
zhang@ubuntu:~/openwrt15.05/source/openwrt$
```

*注意：以上环境搭建 ok;如若环境出现问题会提示缺少哪些库未安装，手动去安装即可。

6、以上没问题可以编译自己的固件了。进入定制界面：

```
zhang@ubuntu:~/openwrt$ ls
BSDmakefile  feeds          LICENSE       staging_dir
config       feeds.conf.default  Makefile     target
Config.in    files         package      tmp
core        include      README       toolchain
fl         key-build   rules.mk     tools
docs       key-build.pub  scripts
zhang@ubuntu:~/openwrt$ make menuconfig
```

使用 make menuconfig 后，如果没有出错会弹出如下界面：

```
.config - OpenWrt Configuration

OpenWrt 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, <?>

Target System (Atheros AR7xxx/AR9xxx) --->
Subtarget (Generic) --->
Target Profile (Default Profile (all drivers))
Target Images --->
Global build settings --->
[ ] Advanced configuration options (for developers
[ ] Build the OpenWrt Image Builder
[ ] Build the OpenWrt SDK
[ ] Package the OpenWrt-based Toolchain
[ ] Image configuration --->
Base system --->
Administration --->
Boot Loaders --->
Development --->
Extra packages ----
Firmware --->
Kernel modules --->
Languages --->
Libraries --->
LuCI --->
Mail --->
Multimedia --->
Network --->
Sound --->
Utilities --->

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

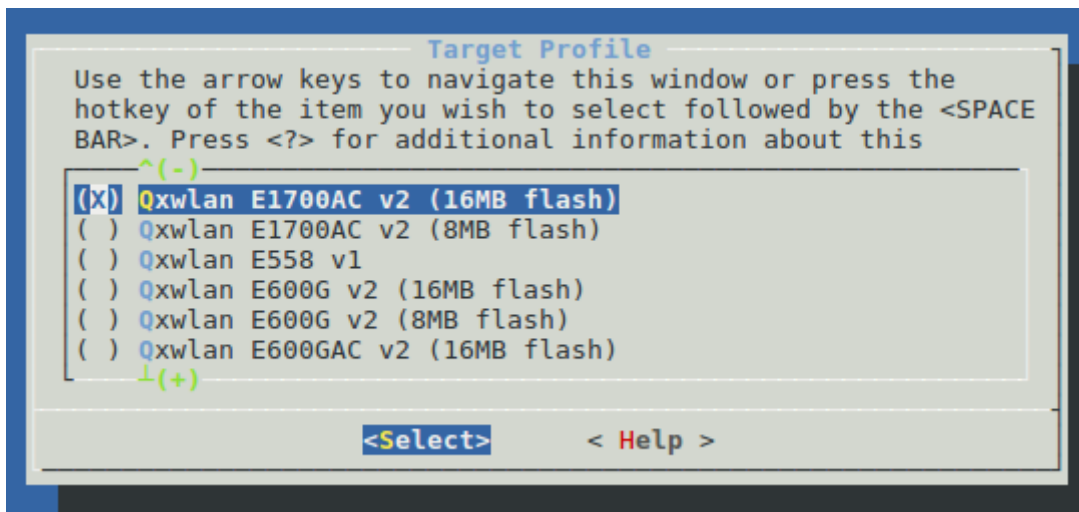
make 作为 openwrt 版本的编译命令，只能在 openwrt 目录执行，进入配置菜单界面，键盘上下是移动光标，左右是选择底部按键，回车是确认，空格是设置选择模式，选项最前面的选择模式有[*]表示编译进固件，[M]表示编译成安装包，[]表示不选择，Esc 是返回上级菜单，按?是帮助，按/是搜索。

7、在此界面中可以选择要编译的固件平台（芯片类型）、型号，还能选择固件中要添加的功能和组件，配置好后保存并退出菜单即可（若无需别的扩展要求，只需选择机型，其他默认即可）。在这里简单示范一下如何选择需要的机型：

键盘上下键移动到 target profile ，回车键进入：

```
Target System (Atheros AR7xxx/AR9xxx) --->
Subtarget (Generic) --->
Target Profile (Default Profile (all drivers)) --->
Target Images --->
Global build settings --->
```

进入之后可以看到很多机型，上下键找到需要的机型，键盘回车就选择好机型了：



如果想添加或者取消某些功能的支持也可以在此配置，这里以 usb 为例：

上下键移动到 kernel modules，再回车键进入里层：



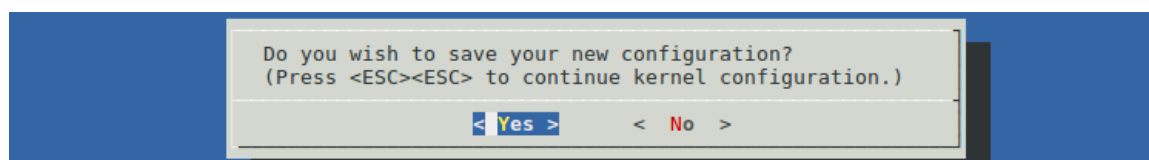
在里层找到 usb support;回车进入里层：



在里层找到如图所示，按下 y 则添加 usb 支持（前面会出现*表示选定）按下 n 则取消次功能：



选定好机型和所需要的模块功能后键盘 Esc 回退到如下界面，选择 yes 回车则保存配置：



8、配置好以后就可以开始编译了：

make V=s (make 是编译命令，V=s 表示输出 debug 信息，V 一定要大写，如果要让 CPU 全速编译，就加上 -j 参数，第一次编译最好不带-j 参数)


```
zhang@ubuntu:~/zhang/new_project/lede_sdk/openwrt$ make V=s
WARNING: Makefile 'package/feeds/packages/aircrack-ng/Makefile' has a dependency
on 'ethtool', which does not exist
WARNING: Makefile 'package/feeds/packages/jool/Makefile' has a dependency on 'et
htool', which does not exist
WARNING: Makefile 'package/feeds/packages/krb5/Makefile' has a dependency on 'li
bss', which does not exist
WARNING: Makefile 'package/feeds/packages/krb5/Makefile' has a dependency on 'li
bcomerr', which does not exist
WARNING: Makefile 'package/feeds/packages/openvswitch/Makefile' has a dependency
on 'kmod-geneve', which does not exist
^Ctarget/Makefile:17: recipe for target '/home/zhang/zhang/new_project/lede_sdk/
openwrt/staging_dir/target-mips_24kc_musl/stamp/.target_prereq' failed
```

第一次编译时间很久（需要下载很多包，可能下载失败报错，重复重新运行 make V=s），耐心等待编译结束。如果按照上面操作进行，且没有错误，在 make 的过程中是不会有问题的，如果出现错误，你需要自己检查以上操作是否全部无误。编译失败会有提示错误信息注意查看打印信息，编译完成后打印如下：

```
Generating index for package ./swconfig_10_ar71xx.ipk
Generating index for package ./uboot-envtools_2014.10-2_ar71xx.ipk
Generating index for package ./ubox_2015-11-22-c086167a0154745c677f8730a336ea9cf7d71031_ar71xx.ipk
Generating index for package ./ubus_2015-05-25-f361bfa5fcb2daadf3b160583ce665024f8d108e_ar71xx.ipk
Generating index for package ./ubusd_2015-05-25-f361bfa5fcb2daadf3b160583ce665024f8d108e_ar71xx.ipk
Generating index for package ./uci_2015-08-27.1-1_ar71xx.ipk
Generating index for package ./usign_2015-05-08-cf8dcdb8a4e874c77f3e9a8e9b643e8c17b19131_ar71xx.ipk
Generating index for package ./wpad-mini_2015-03-25-1_ar71xx.ipk
Signing package index...
make[2]: Leaving directory '/home/zhang/openwrt'
make[1]: Leaving directory '/home/zhang/openwrt'
zhang@ubuntu:~/openwrt$
```

9、编译完成后，在 bin/targets/ar71xx/generic/看到生成的固件：

```
zhang@ubuntu:~/zhang/new_project/lede_sdk/openwrt/bin/targets/ar71xx/generic$ ls
config.seed
openwrt-ar71xx-generic-device-e1700ac-v2-16m.manifest
openwrt-ar71xx-generic-e1700ac-v2-16M-squashfs-sysupgrade.bin
openwrt-ar71xx-generic-root.squashfs
openwrt-ar71xx-generic-uImage-lzma.bin
openwrt-ar71xx-generic-vmlinux.bin
openwrt-ar71xx-generic-vmlinux.elf
openwrt-ar71xx-generic-vmlinux.lzma
openwrt-ar71xx-generic-vmlinux-lzma.elf
packages
sha256sums
zhang@ubuntu:~/zhang/new_project/lede_sdk/openwrt/bin/targets/ar71xx/generic$
```

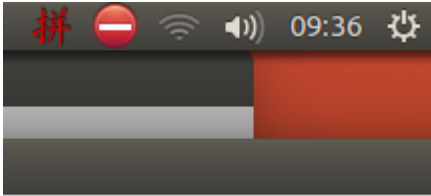
上图中被标记的位置是我们升级所需要的固件，具体升级哪一个看接下来的 openwrt 刷入路由器操作。

第三部分：将 OpenWrt 刷入路由器（E1700AC 为例）

一、ip 地址添加(以 linux 为例，windows 在网络配置设置)

在升级固件之前需要在 pc 上添加 192.168.1.xxx(2-255)网段的 ip 地址，使用此 sdk 编译生成的固件 ip 均为 192.168.1.1 便于后面升级以及测试。具体操作如下：

1、将鼠标移动到 pc 右上角，点击网络连接图标，会看到很多选择，最下面选择“编辑连接”：



2、点击“编辑连接”之后会弹出如下框图，左键“单击有线连接 1”再单击“编辑(E)”（双击“有线连接 1”效果一样）：



3、会再弹出一个框图，点击“IPv4 设置”如下图所示操作（ip 地址 192.168.1.xxx），最后保存：



4、查看 ip 地址是否已经添加成功，终端输入 ifconfig，如下图：

```
zhang@ubuntu:~/openwrt/bin/ar71xx$ ifconfig
enp2s0  Link encap:以太网  硬件地址 74:d4:35:fb:59:3f
        inet 地址:192.168.1.10  广播:192.168.1.255  掩码:255.255.255.0
        inet6 地址: fe80::a846:60e:ee6e:d86c/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
        接收数据包:617  错误:0  丢弃:0  过载:0  帧数:0
        发送数据包:1434  错误:3  丢弃:0  过载:0  载波:0
        碰撞:0  发送队列长度:1000
        接收字节:186018 (186.0 KB)  发送字节:158797 (158.7 KB)
```

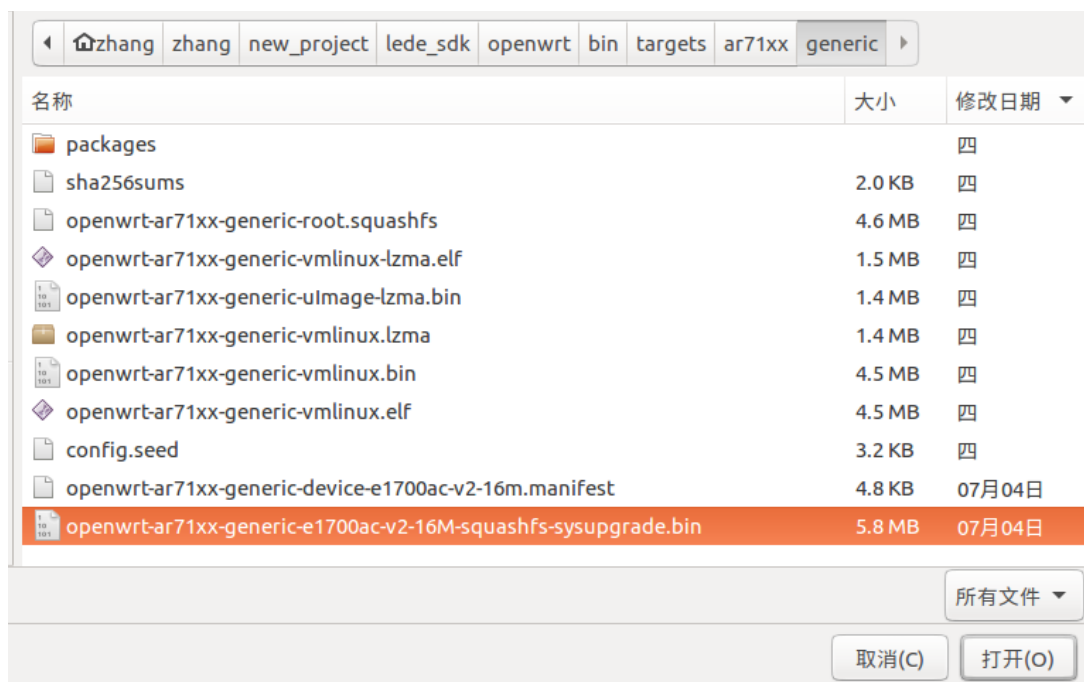
二、升级方法

方法 1：复位键升级固件（e700ac 将拨动开关打到靠近网口侧，升级完立即拨回原位）

1、将路由器与 pc 用网线连接起来，在按下复位的同时给路由器插上电源线;当看到路由器上除了电源指示灯以外，其他灯都在闪烁后松开复位键;然后打开网页输入 192.168.1.1 可以看到升级界面：

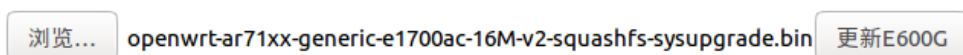


2、点击浏览，找到自己要升级的固件（xxxx-sysupgrade.bin 为要升级的固件），选中后点击打开即可：



3、最后点击更新，然后等待路由器自动重启，升级已完成（路由器有一个 led 灯一直闪烁，等到停止闪烁表示升级启动完成）：

请从你的硬盘上选择合适的固件并单击更新E600G 按钮。



4、点击更新后可以看到如下界面（e700ac 需要将拨动开关打到原位（远离网口侧）），然后等待重启即完成升级：

