

## 云端笔记

学习的备忘录

### OPENWRT编译之树莓派4B

📅 2019-11-24 | 📁 Linux

#### 环境配置说明

- 适用硬件: Raspberry Pi 4B
- 编译系统: Debian For WSL
- 参考借鉴: [参数配置](#) | [LuCI配置](#) | [网卡设置](#)

```
1 # 安装编译工具
2 sudo apt-get update
3 sudo apt-get install build-essential asciidoc binutils bzip2 \
4 gawk gettext git libncurses5-dev libz-dev patch unzip zlib1g-dev \
5 lib32gcc1 libc6-dev-i386 subversion flex uglifyjs libssl-dev upx \
6 gcc-multilib p7zip p7zip-full msmtp texinfo libglib2.0-dev xmlto \
7 git-core qemu-utils libelf-dev autoconf automake libtool autopoint \
8 curl wget device-tree-compiler rsync
```

#### 获取编译源码

1. 注意: 不要使用root用户编译, 最好使用梯子来加速源码的下载速度
2. 编译完成后把 bin/targets/brcm27xx/bcm2711/openwrt-bcm27xx-bcm2711-rpi-4-ext4-factory.img.gz解压后的img文件刷入SD卡中即可
3. 可选lean版魔改源码替换官方源码: <https://github.com/coolsnowwolf/lede>

```
1 # 设置代理来加速源码下载, 没有代理可跳过或根据需求设置
2 # export all_proxy="127.0.0.1:7890"
3
4 # 获取openwrt官方源码
5 mkdir openwrt
6 cd openwrt/
7 git clone https://git.openwrt.org/openwrt/openwrt.git ./
8
9 # 以后每次编译前建议执行以下三行命令更新源码
10 git pull
11 ./scripts/feeds update -a
```

```

12 ./scripts/feeds install -a
13
14 make defconfig          # 测试编译环境
15 make menuconfig        # 进入固件配置界面
16 make download -j8 V=s   # 下载所需源码, 请尽量使用梯子
17 make -j1 V=s           # 首次编译推荐用单线程
18
19 # 再次编译前建议使用make clean清理
20 make clean             # 清除bin目录
21 make dirclean          # 清除bin目录和交叉编译工具及工具链目录
22 make distclean         # 清除所有相关的东西, 包括下载的软件包, 配置文件, feed内容等

```

## 可选内核配置

```

1 # 额外配置内核参数, 一般不需要配置
2 # 只能包含到内核而不能作为模块, 因为编译后的模块不会添加到固件内
3 make kernel_menuconfig # 进入内核配置界面
4
5 # 打开KVM支持: Virtualization > Kernel-based Virtual Machine support
6 # 修改的内核参数会保存到openwrt/target/linux/bcm27xx/bcm2711/config-5.4
7 # 和openwrt/target/linux/generic/config-5.4合成最终内核配置文件
8 # openwrt/build_dir/target-aarch64_cortex-a72_musllinux-bcm27xx_bcm2711/linux-5.4.42

```

## 固件编译配置

1. 简单的make menuconfig参数配置, 除必选配置外的其他项可根据需求自行选择

2. 基础配置

```

1 # 必选配置
2 Target System -> Broadcom BCM27xx
3 Subtarget -> BCM2711 boards (64 bit)
4 Target Profile -> Raspberry Pi 4B
5
6 # 镜像参数
7 Target Images -> ext4          # ext4格式的固件可方便地调整分区大小
8 Target Images -> squashfs      # squashfs格式的固件可恢复出厂设置
9 Target Images -> Kernel partition size = 64          # boot分区大小为64M
10 Target Images -> Root filesystem partition size = 512 # root分区大小为512M
11
12 # 可选工具
13 Base system -> block-mount      # 在LuCI界面添加<挂载点>菜单
14 Base system -> blockd          # 自动挂载设备
15 Base system -> wireless-tools  # 无线扩展工具
16 Administration -> htop         # 添加htop命令
17 Firmware -> xxx               # 选择你需要的网卡固件, 默认即可

```

### 3. 内核模块

```

1  # 文件系统
2  Kernel modules -> Filesystems -> kmod-fs-ext4
3  Kernel modules -> Filesystems -> kmod-fs-ntfs
4  Kernel modules -> Filesystems -> kmod-fs-squashfs
5  Kernel modules -> Filesystems -> kmod-fs-vfat
6  Kernel modules -> Filesystems -> kmod-fuse
7
8  # 网卡支持
9  Kernel modules -> Network Devices -> kmod-xxx    # 有线网卡支持, 跟以下几项可根据需求选
10 Kernel modules -> Wireless Drivers -> kmod-rt2800-usb    # 添加Ralink
11 Kernel modules -> USB Support -> kmod-usb-net -> kmod-usb-net-sr9700    # 添加USB2.0
12 Kernel modules -> USB Support -> kmod-usb-net -> kmod-usb-net-rtl8152    # 添加USB2.0
13 Kernel modules -> USB Support -> kmod-usb-net -> kmod-usb-net-asix    # 添加支持亚
14 Kernel modules -> USB Support -> kmod-usb-net -> kmod-usb-net-asix-ax88179    # 添加U
15
16 # USB支持
17 Kernel modules -> USB Support -> kmod-usb-core    # 启用USB支持
18 Kernel modules -> USB Support -> kmod-usb-hid    # USB键鼠支持
19 Kernel modules -> USB Support -> kmod-usb-ohci    # 添加OHCI支持
20 Kernel modules -> USB Support -> kmod-usb-uhci    # 添加UHCI支持
21 Kernel modules -> USB Support -> kmod-usb-storage    # 启用USB存储
22 Kernel modules -> USB Support -> kmod-usb-storage-extras
23 Kernel modules -> USB Support -> kmod-usb-usb2    # 开启USB2支持
24 Kernel modules -> USB Support -> kmod-usb-usb3    # 开启USB3支持

```

### 4. LuCI配置

```

1  # LuCI设置
2  LuCI -> Collections -> luci    # 开启luci
3  LuCI -> Modules -> Translations -> Chinese(zh-cn)    # 中文支持
4  LuCI -> Themes -> luci-theme-material    # 添加主题
5
6  # LuCI应用
7  LuCI -> Applications -> luci-app-aria2    # 下载工具
8  LuCI -> Applications -> luci-app-firewall    # 防火墙
9  LuCI -> Applications -> luci-app-hd-idle    # 硬盘休眠
10 LuCI -> Applications -> luci-app-opkg    # 软件包
11 LuCI -> Applications -> luci-app-qos    # 服务质量
12 LuCI -> Applications -> luci-app-samba4    # 网络共享
13 LuCI -> Applications -> luci-app-frpc    # 内网穿透
14 LuCI -> Applications -> luci-app-shadowsocks-libev    # 翻墙软件
15 LuCI -> Applications -> luci-app-upnp    # UPnP服务
16 LuCI -> Applications -> luci-app-wol    # 网络唤醒
17 .....

```

### 5. 其他工具

```

1  Network -> Download Manager -> ariang      # Aria2管理页面
2  Network -> File Transfer -> Aria2 Configuration -> *** # 选择Aria2支持的功能
3  Network -> File Transfer -> curl           # 添加curl命令
4  Network -> File Transfer -> wget           # 添加wget命令
5  Utilities -> Compression -> bsdtar         # tar打包工具
6  Utilities -> Compression -> gzip          # GZ 压缩套件
7  Utilities -> Compression -> xz-utils      # XZ 压缩套件
8  Utilities -> Compression -> unzip         # zip解压工具
9  Utilities -> Compression -> zip           # zip压缩工具
10 Utilities -> Disc -> fdisk                 # 磁盘分区工具
11 Utilities -> Disc -> lsblk                 # 磁盘查看工具
12 Utilities -> Editors -> vim               # vim编辑器
13 Utilities -> Filesystem -> ntfs-3g        # NTFS读写支持
14 Utilities -> Filesystem -> resize2fs      # 分区大小调整
15 Utilities -> Terminal -> screen           # 添加screen
16 Utilities -> pciutils                      # 添加lspci命令
17 Utilities -> usbutils                     # 添加lsusb命令

```

## 6. IPv6支持

```

1  Global build settings -> Enable IPv6 support in packages # 启用IPv6项
2  Network -> odhcp6c                                         # IPv6客户端
3  Network -> odhcpd-ipv6only                                 # IPv6服务端
4  Network -> Firewall -> ip6tables                         # IPv6防火墙
5  LuCI -> Protocols -> luci-proto-ipv6                     # WebUI支持

```

## 手动配置网卡

1. 用网线连接自己的电脑和树莓派，LEDE默认的IP为静态 192.168.1.1，可以将电脑网卡设置为静态IP：192.168.1.2 来访问树莓派上的LEDE管理界面。
2. 设置2.4G无线：网络 -> 无线 选项卡中点修改，基本设置中设置为Legacy模式，频宽2.4GHz和信道自选（切勿使用自动信道和自动频宽），高级设置中国家代码AU。接口配置中模式：接入点AP，网络：lan，然后设置无线安全加密WPA2-PSK和密码等，其余默认即可。
3. 设置5G无线：网络 -> 无线 选项卡中点修改，基本设置中设置为AC模式，只能使用36信道和频宽为20Mhz或者40Mhz才能用作AP（切勿使用自动信道和自动频宽），高级设置中国家代码AU。接口配置中模式：接入点AP，网络：lan，然后设置无线安全加密WPA2-PSK和密码等，其余默认即可。
4. 设置WAN口 网络 -> 接口 选项卡中添加一个新接口命名为“ wan”，协议根据自己的网络情况自行选择(我的是DHCP客户端)，在 物理设置 中单选接口为eth0，先保存但别应用。设置LAN口 网络 -> 接口 选项卡中编辑lan网卡的物理设置，在无线网络和自定义接口上打上钩填上“eth0.1”，并取消原本eth0前的钩后保存。

## 自动配置网卡

在openwrt目录下新建一个名为files的文件夹，openwrt在编译的时候，会把files文件夹里的文件编译到固件的根目录。

在files/etc/config/中新建两个文件：network，wireless内容如下用来分别配置接口和无线

```
1  # openwrt/files/etc/config/network
2
3  config interface 'loopback'
4      option ifname 'lo'
5      option proto 'static'
6      option ipaddr '127.0.0.1'
7      option netmask '255.0.0.0'
8
9  config globals 'globals'
10     option ula_prefix 'fd38:13ae:ceeb::/48'
11
12 config interface 'lan'
13     option type 'bridge'
14     option proto 'static'
15     option ipaddr '192.168.1.1'
16     option netmask '255.255.255.0'
17     option ip6assign '60'
18     option ifname 'eth0.1'
19     option _orig_ifname 'eth0 wlan0'
20     option _orig_bridge 'true'
21
22 config interface 'wan'
23     option ifname 'eth0'
24     option proto 'dhcp'
25     option _orig_ifname 'erspan0'
26     option _orig_bridge 'false'


1  # openwrt/files/etc/config/wireless
2
3  config wifi-device 'radio0'
4      option type 'mac80211'
5      option channel '36'
6      option hwmode '11a'
7      option path 'platform/soc/fe300000.mmcnr/mmc_host/mmc1/mmc1:0001/mmc1:0001:'
8      option htmode 'VHT40'
9
10 config wifi-iface 'default_radio0'
11     option device 'radio0'
12     option mode 'ap'
13     option encryption 'none'
14     option ssid 'RPI4'
15     option network 'lan'
```

## 其他实用技巧

1. 系统 -> LED配置中：LED0是绿色灯，LED1是红色灯，可以用来当无线收发的指示灯
2. 使用： `cat /sys/class/thermal/thermal_zone0/temp` 查看CPU温度
3. 在Samba中使用root用户除了执行 `smbpasswd -a root` 添加用户外，还需要注释模板中的：  
`#invalid users = root`
4. 使用Win10 WSL环境编译的，编译前建议先将Win10系统变量剔除，否则可能报错：`export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin`
5. 单核编译大概需要4个小时，中间可能会有几次错误，有的错误可跳过对后续编译没有影响
6. 在OpenWRT的 `.config` 文件中内核的选项都是形如 `CONFIG_KERNEL_XXX` 。自己手动添加内核选项时不要忘了在中间添加 `_KERNEL_` 前缀。
7. Goproxy 代理失败报错的解决方法：`export GO111MODULE=on` 和 `export GOPROXY=https://goproxy.cn`

1 go: github.com/armon/go-socks5@v0.0.0-20160902184237-e75332964ef5: Get "https://prox

# rpi

◀ Android平台QT添加MySQL插件

Imagebuilder为树莓派4B构建OPENWRT系统 ▶

© 2016 — 2020  leux

粤ICP备19106113号