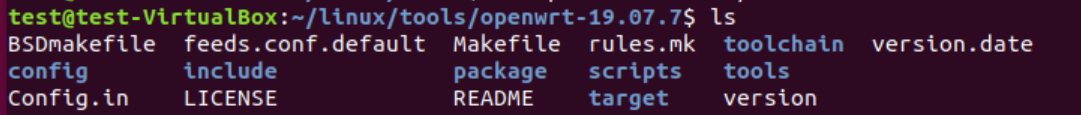
**（一）编译openwrt**



sudo apt update

sudo apt install build-essential asciidoc binutils bzip2 gawk gettext git \

libncurses5-dev libz-dev patch python python3 unzip zlib1g-dev libssl-dev \

wget curl subversion flex uglifyjs node-uglify gcc-multilib g++-multilib

sudo apt install qemu-utils upx

python –version（Python 2.7.x）

没有的话执行：sudo apt install python-minimal

make distclean

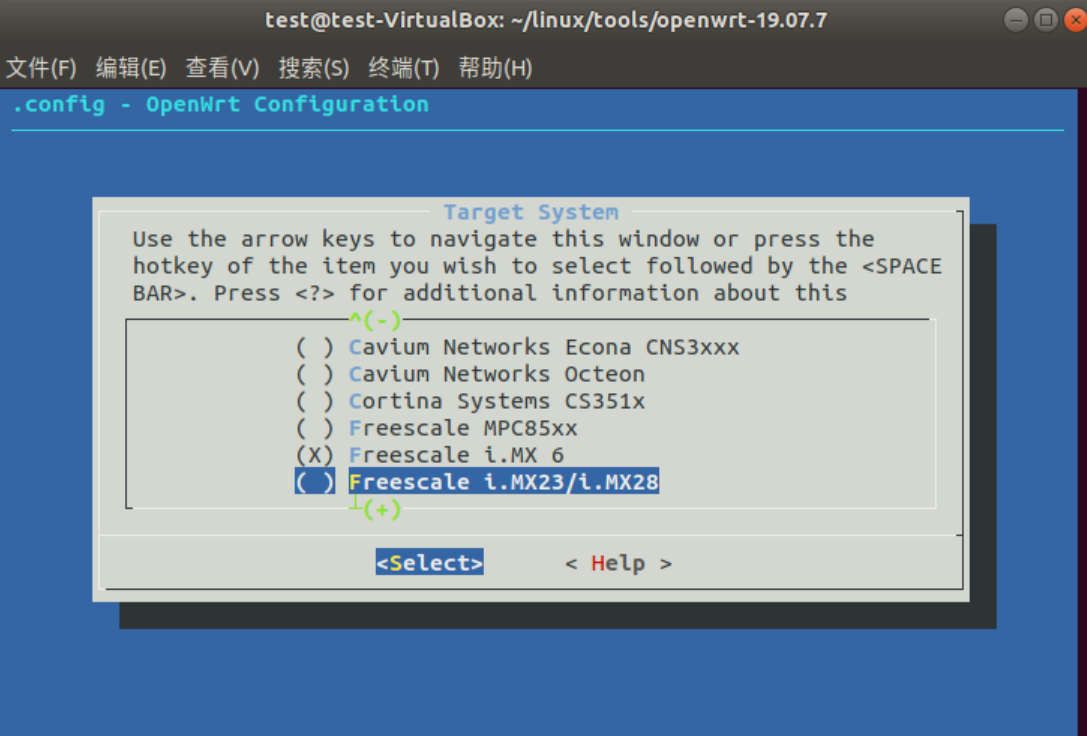
./scripts/feeds update -a

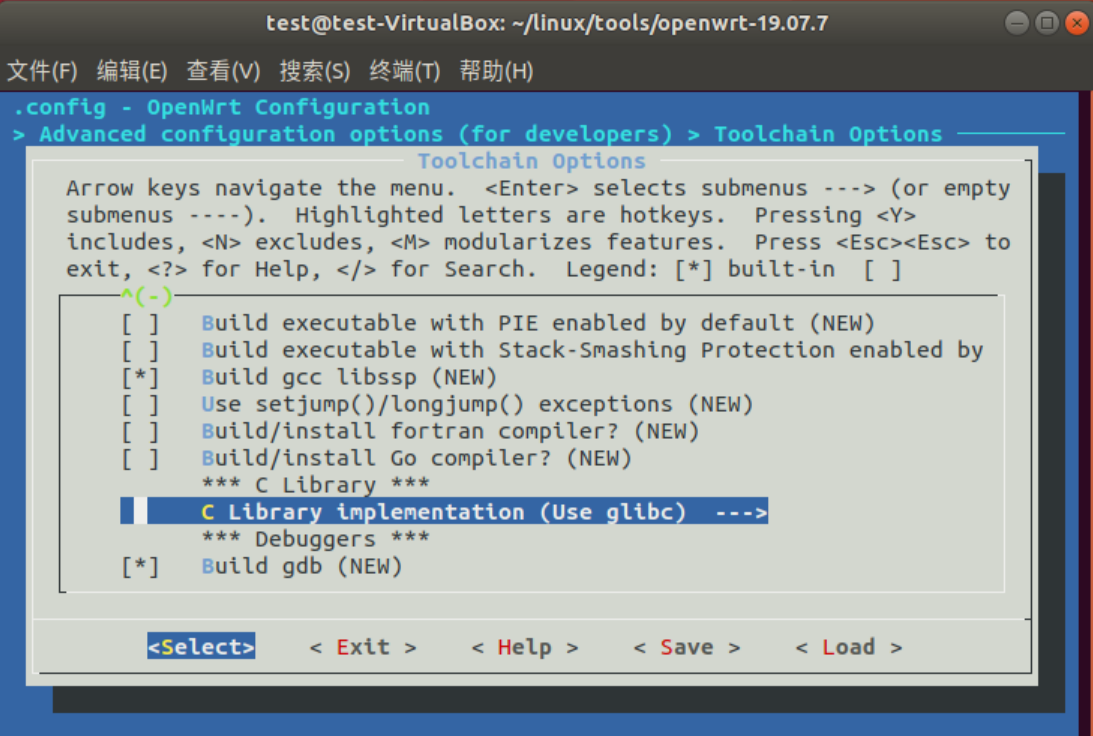
./scripts/feeds install -a

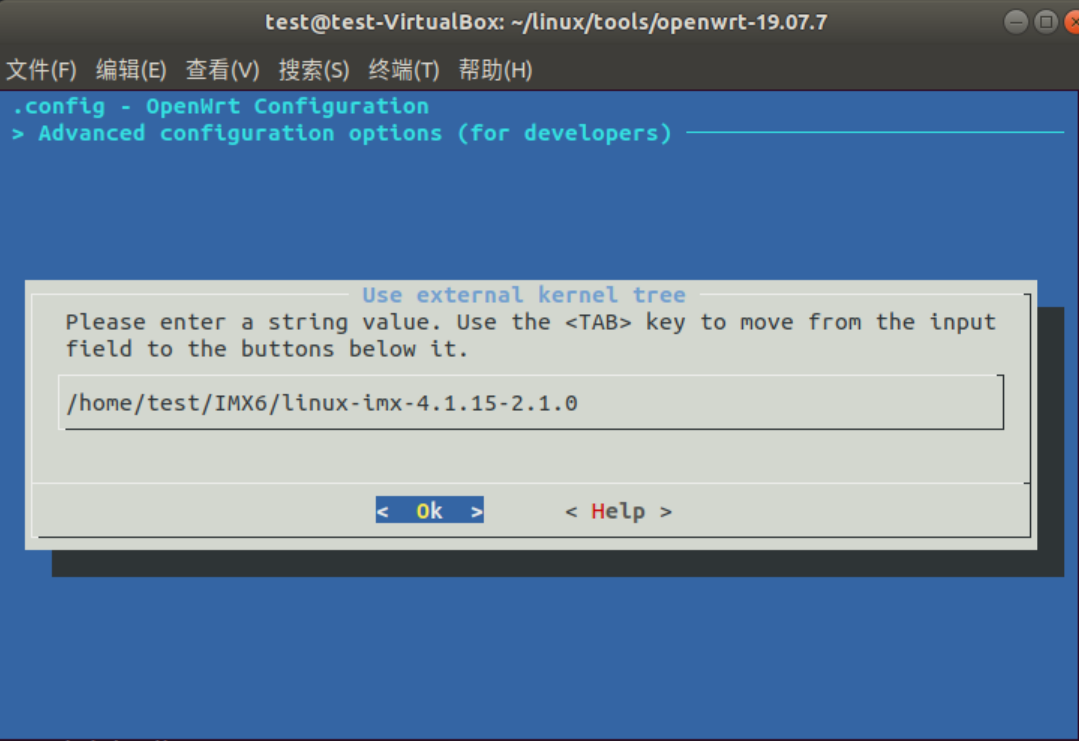
make defconfig

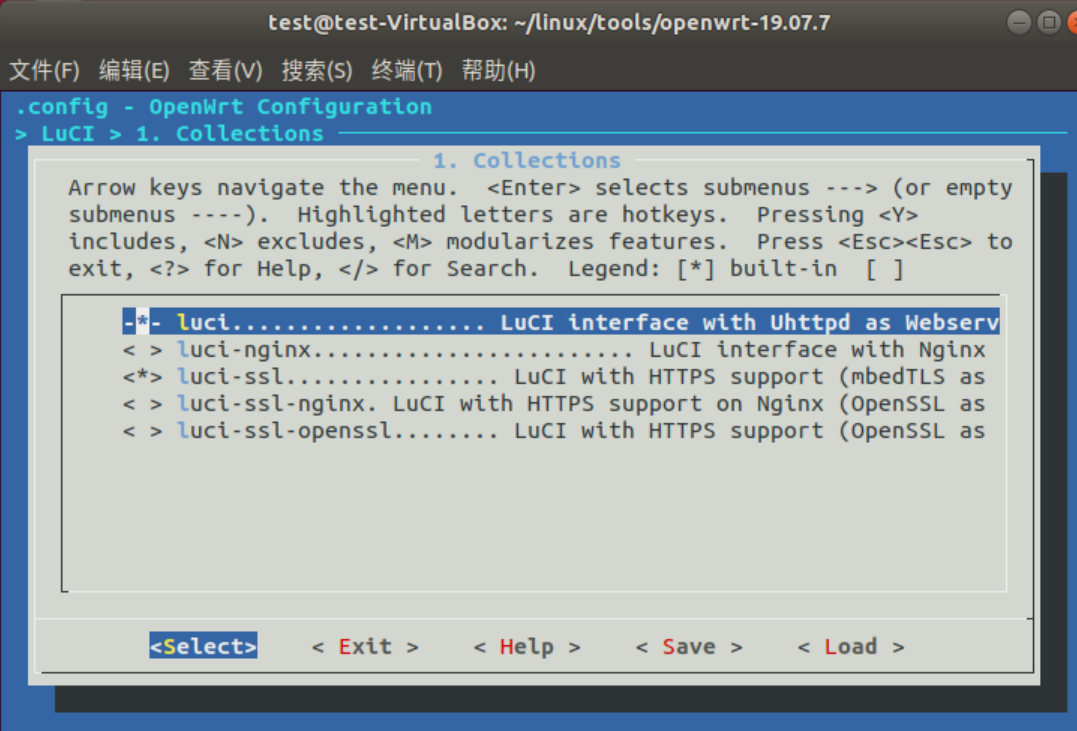
make prereq V=sc

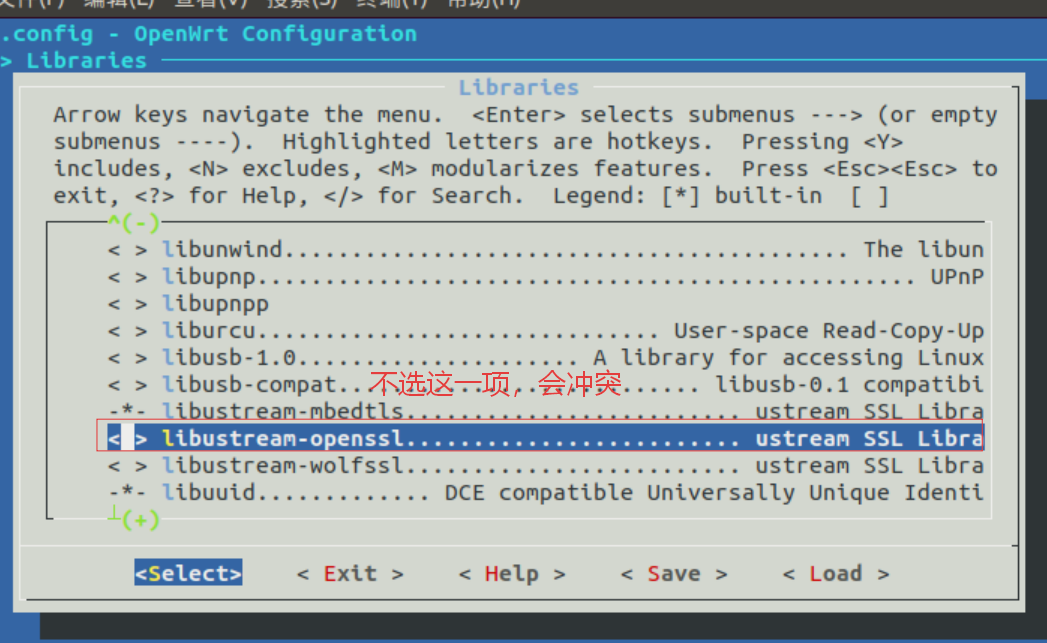
配置：



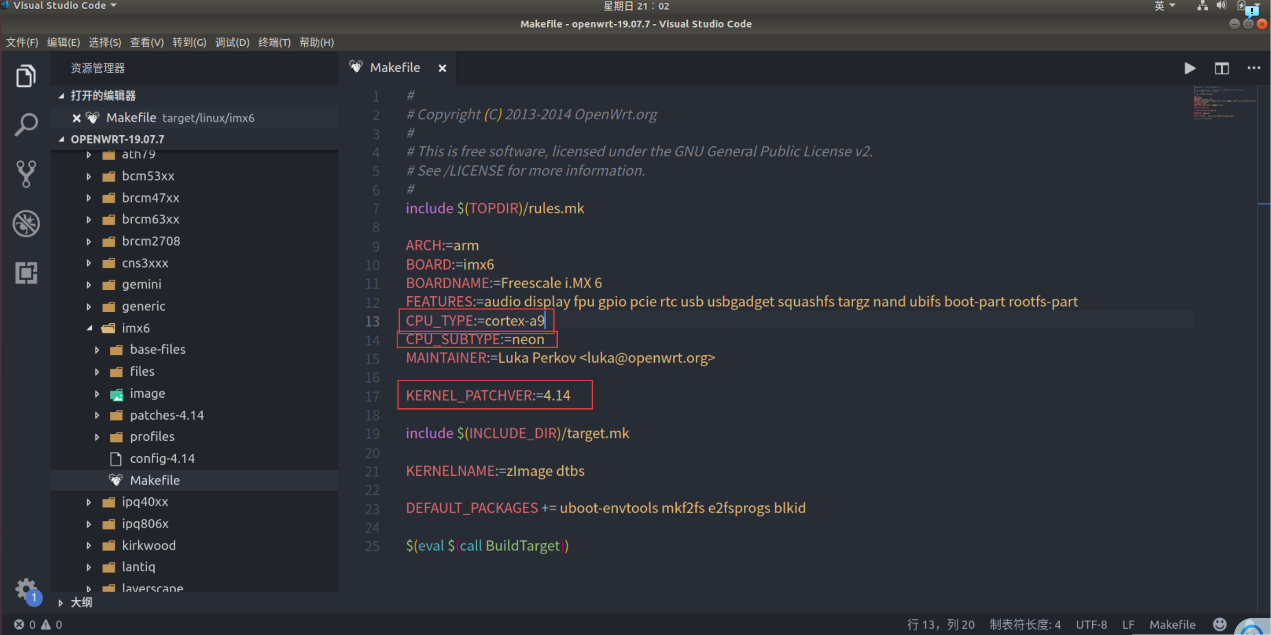




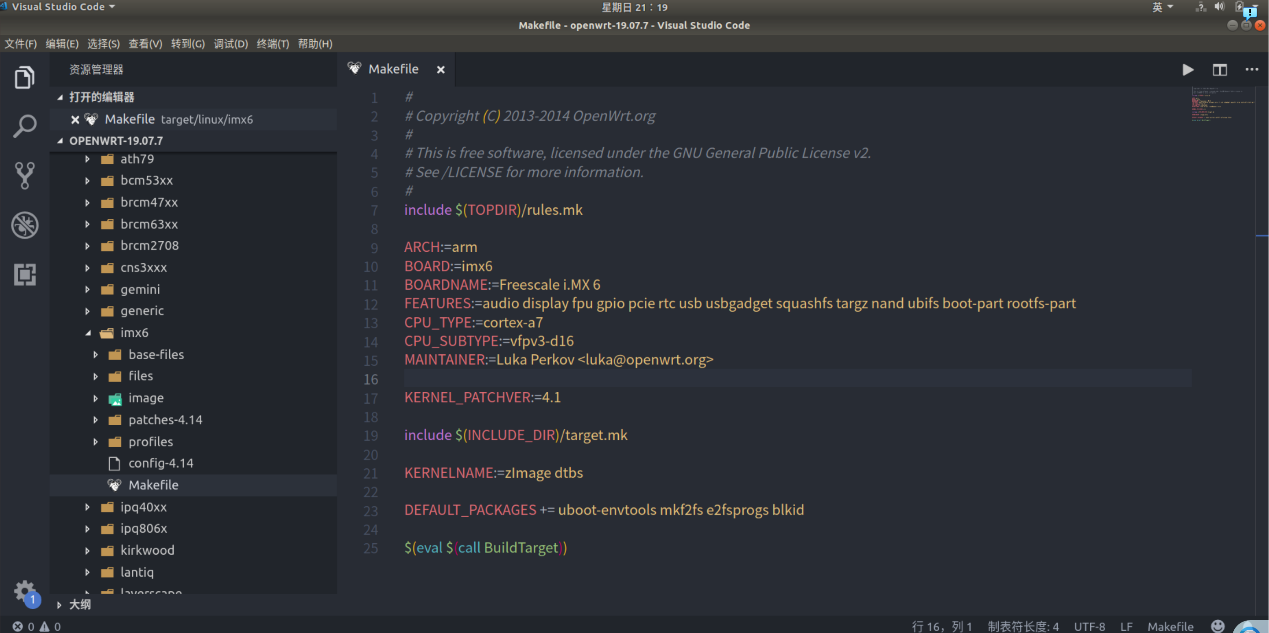




将linux-imx-4.1.15-2.1.0/arch/arm/configs/下对应开发板的内容复制到openwrt-19.07.7/target/linux/imx6/config-4.1



修改为如下内容：



根据报错提示进行修改：

sudo apt install build-essential clang flex bison gawk libncurses5-dev libncursesw5-dev zlib1g-dev libssl-dev gettext unzip file python3 rsync wget

参考文章：<https://blog.csdn.net/qq_40614144/article/details/105538483>

需要先编译一边自己的内核，再编译openwrt。

先使用网络挂载的方式进行调试，最后使用烧录的方式下载。需要的文件：zImage、根文件、设备树，uboot使用原开发板的。

env default -a;saveenvsetenv ipaddr 192.168.211.100setenv ethaddr 00:04:9f:04:d2:35setenv gatewayip 192.168.211.1setenv netmask 255.255.255.0setenv serverip 192.168.211.50saveenv

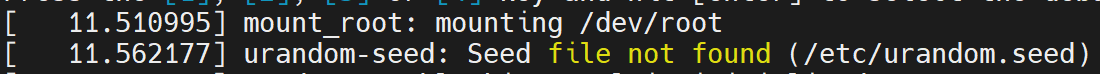
这里之后重启

setenv bootcmd 'tftp 0x80800000 zImage; tftp 0x83000000 openwrt-imx6-imx6ull-14x14-emmc-4.3-800x480-c.dtb; setenv bootargs console=ttymxc0,115200 root=/dev/nfs nfsroot=192.168.211.50:/home/test/linux/nfs/openwrt/rootfs,v3,tcp rw ip=192.168.211.100:192.168.211.50:192.168.211.1:255.255.255.0::eth0:off rootwait; bootz 0x80800000 - 0x83000000'

saveenv

run bootcmd

在内核这部分会有两个日志问题：



对应的解决：

（1）mount\_root

rootfs中的/sbin/mount\_root：

先备份

mv /home/test/linux/nfs/openwrt/rootfs/sbin/mount\_root \ /home/test/linux/nfs/openwrt/rootfs/sbin/mount\_root.orig

之后在该目录中创建被备份的文件，并添加如下内容：

#!/bin/shecho ">>> skip mount\_root (NFS rootfs mode)" > /dev/consoleexit 0

最后，这是一个可执行文件

Chmod +x mount\_root

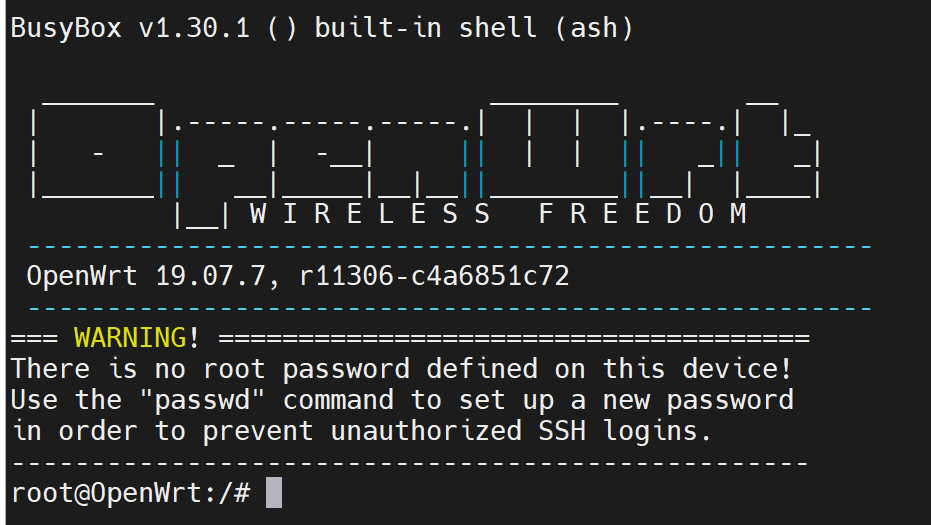
（2）urandom-seed

在rootfs中的/etc/目录下创建urandom.seed，因为在编译生成的时候，文件只能读。添加如下内容

dd if=/dev/urandom of=/home/test/linux/nfs/openwrt/rootfs/etc/urandom.seed bs=1 count=512

sudo chown root:root etc/urandom.seedsudo chmod 644 etc/urandom.seed

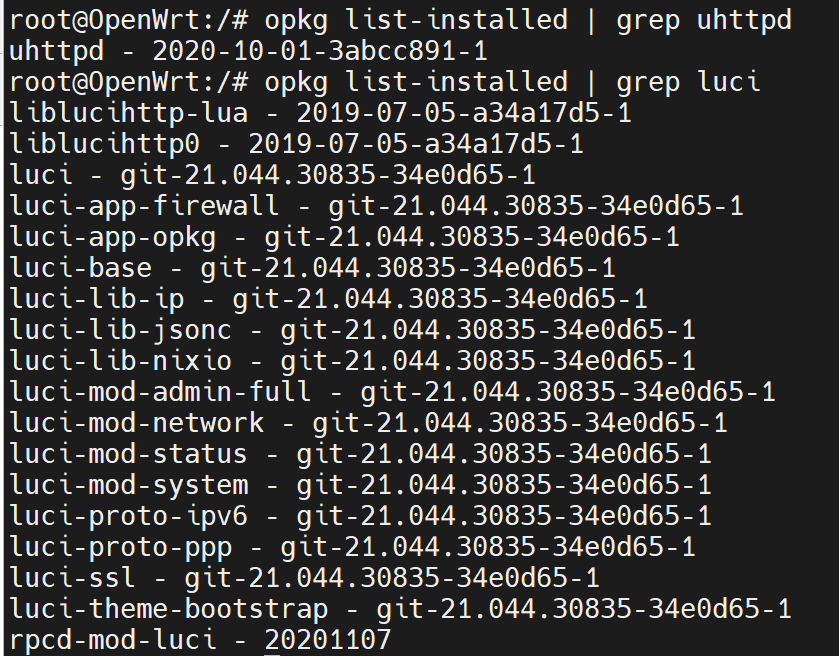
进入系统后便是这样的：



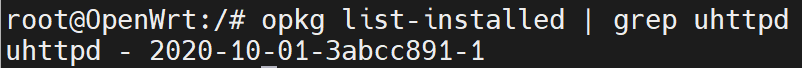
需要ping通才能访问对应的网址

检查luci和uhttpd是否安装：正常情况下会有输出

opkg list-installed | grep luci



opkg list-installed | grep uhttpd

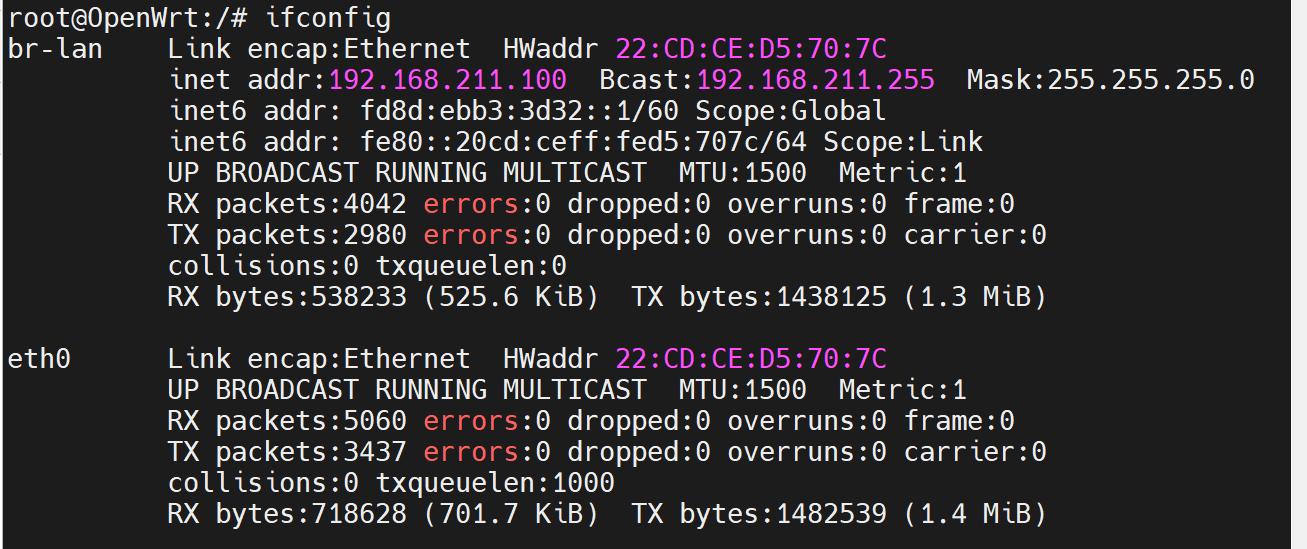


为保证在同一网段下，修改IP：

uci set network.lan.ipaddr='192.168.211.100'

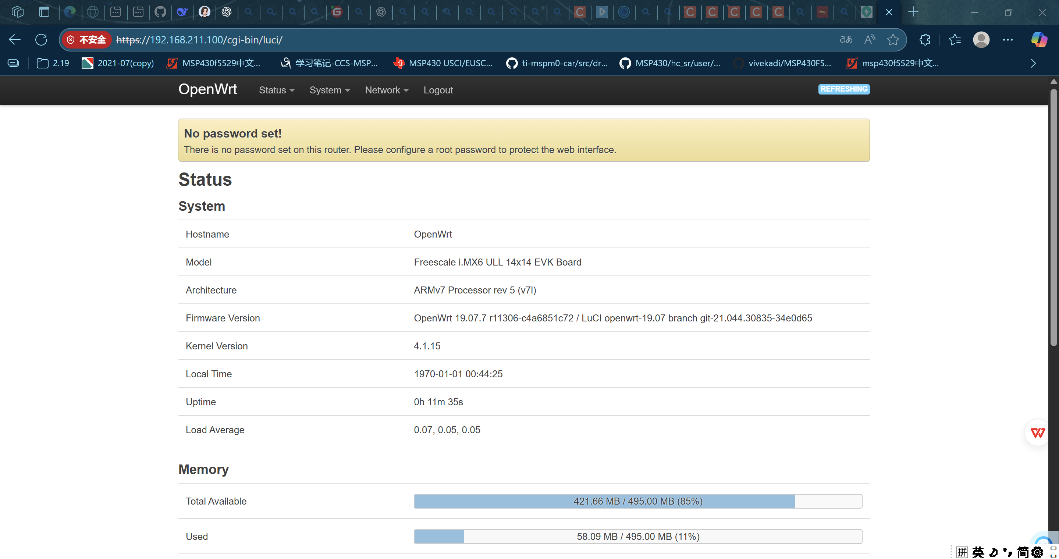
uci commit network

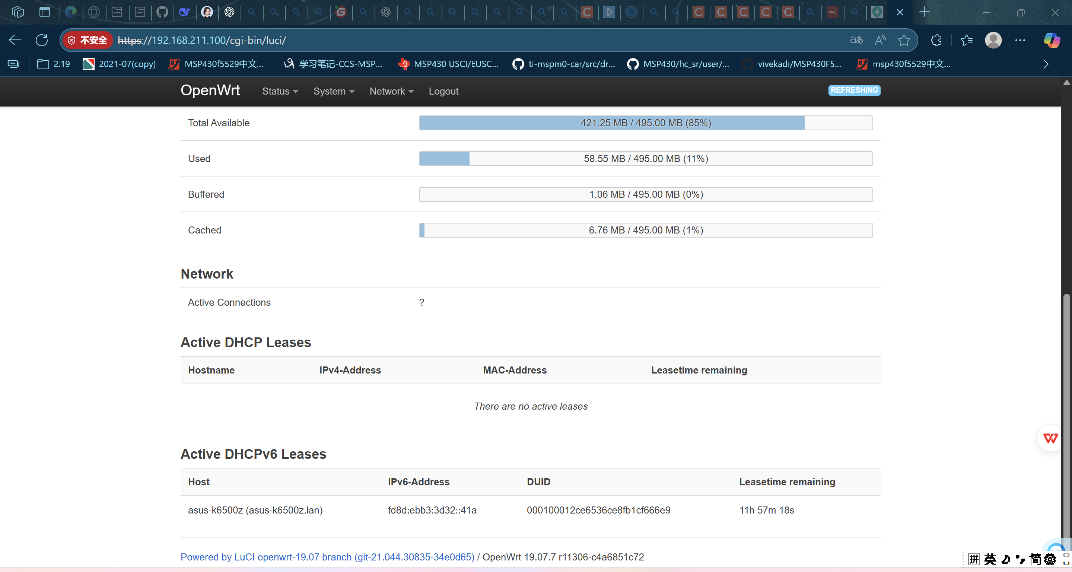
/etc/init.d/network restart



PC端进入管理页：

http://192.168.211.100





**（二）上网设备**

1. 光纤

* 运营商提供的传输介质。
* 家里拉进来的光纤负责把信号从运营商机房送到你家。
* 光纤本身传输的是光信号，不能直接给电脑或手机用。

2. 光猫（又叫“光纤猫”、“光纤调制解调器”）

* 光纤 → 电信号的转换器。
* 运营商给你的光纤先接入光猫，由它把光信号转为以太网电信号。
* 光猫一般会负责 拨号（PPPoE）、认证账号密码，让你能接入运营商的网络。
* 有的光猫带路由功能（叫“光猫路由一体机”），有的只有桥接功能，需要后面再接路由器。

3. 带宽

* 运营商给你开通的网络速率，比如 100M、300M、1000M。
* 光猫和运营商后台协商后，就会限制你能使用的最大速率。
* 带宽决定了你家里网速的“上限”。

4. 路由器

* 光猫出来的网线接到路由器，由路由器负责：

分配 IP（DHCP），让家里多台设备同时上网。

无线 WiFi 信号发射。

NAT 转换、防火墙等网络管理功能。

* 如果光猫本身就带路由功能，也可以不开额外的路由器，但一般家用路由器性能更好。

5. 机顶盒

简单说就是利用运营商专线看电视。

* 常见的是 IPTV 机顶盒。
* 光猫通常有一个单独的 IPTV 专用口，接机顶盒。
* 机顶盒通过专网或 VLAN 与运营商电视平台通信，播放直播/点播电视。
* 它虽然也“用网”，但和你上网用的带宽是两条逻辑通道，互不影响。

总结下来，上网需要的步骤：

安装宽带—>连接路由器—>配置路由器

**（三）预备理论**

参考网页：

<https://lovecodeboy.blog.csdn.net/article/details/130802381>

<https://blog.csdn.net/i826056899/article/details/148685052>

[树莓派+Openwrt连接校园网，打破校园网设备限制\_openwrt 校园网-CSDN博客](https://blog.csdn.net/m0_72845244/article/details/137870117)

**（1）openwrt简介**

OpenWrt 是一个开源、可定制的路由器操作系统，具备强大的网络配置能力、软件包管理、友好的 Web 界面、广泛的硬件支持以及良好的安全性与社区生态。

Openwrt核心组件

内核：底层的系统支持和硬件驱动；

UCI：提供统一的配置接口，便于管理系统配置和参数；

LuCI：Web界面和管理平台，提供直观的图形界面和自定义拓展。

1.开源与可定制

* 完全开源，自由修改和扩展。
* 用户可根据需求裁剪系统，只安装所需功能。

2.软件包管理（opkg）

* 内置包管理器，可像 PC 系统一样安装/卸载/更新软件包。

3.强大的网络配置能力

* 支持 VLAN、QoS、端口转发、防火墙、VPN 等高级功能。
* 可灵活配置路由和交换。

4.LuCI Web 管理界面

* 提供直观的网页界面，简化日常配置和管理。
* 同时支持命令行（SSH）进行精细化控制。

5.广泛的硬件支持

* 兼容多种路由器和嵌入式设备。

**（2）常见网络设置相关概念**

1. **IP 分配**

* **静态 IP**：手动为设备指定固定 IP 地址。
* **动态 IP（DHCP 客户端）**：自动从上游路由器或运营商获取 IP。
* **DHCP 服务器**：由路由器为内网设备分配 IP。

2. **DHCP（动态主机配置协议）**

* 负责给内网设备自动分配 IP 地址、子网掩码、网关和 DNS。
* 可以自定义分配范围，甚至为指定设备绑定固定 IP。

3. **端口映射（Port Forwarding / NAT）**

* 把公网的某个端口映射到内网设备上，例如把外部 8080 → 内部服务器 80。
* 常用于远程访问 NAS、摄像头、家庭服务器等。

4. **VLAN（虚拟局域网）**

* 在同一物理交换机/路由器上划分逻辑隔离的网络。
* 比如：家庭网（192.168.1.x）和 IoT 设备网（192.168.2.x）互相隔离，提升安全性。

5. **无线参数（Wi-Fi 设置）**

* SSID（网络名）、信道（Channel）、加密方式（WPA2/WPA3）、发射功率等。
* 可以设置多 SSID（如家庭网/客人网）、隐藏 SSID 等。

**（3）常见网络角色/模式**

1. **主路由（Router Mode）**

* 直接拨号上网（PPPoE 或 DHCP），负责 NAT、防火墙、DHCP 分配。

2. **旁路由（Bypass Router）**

* 主路由负责拨号，旁路由只接管部分流量（如科学上网、广告过滤）。
* 优点：低风险、兼容原有网络结构。

3. **中继（Repeater/Relay Mode）**

* 无线桥接，将一个 Wi-Fi 网络扩展为更大覆盖范围。

4. **AP（Access Point，无线接入点）**

* 把有线网络转换成无线网络，不做 NAT，一般只提供 Wi-Fi 接入。

**（4）其他常见网络功能**

 **QoS（服务质量）/ SQM（智能队列管理）**

* 控制带宽分配，避免单个设备/应用占满网络。

 **DDNS（动态域名解析）**

* 给家庭公网 IP 绑定域名，方便远程访问。

 **VPN（虚拟专用网络）**

* 支持 OpenVPN、WireGuard 等，实现远程访问或隐私保护。

 **防火墙（Firewall）**

* 基于 iptables/nftables，可定义内外网访问规则。

 **Mesh 网络**

* 多个路由器组成无线自组网，实现无缝 Wi-Fi 覆盖。

 **多 WAN / 负载均衡 / Failover**

* 支持多条宽带同时接入，实现流量分担或主备切换。

**（四）配置路由器**

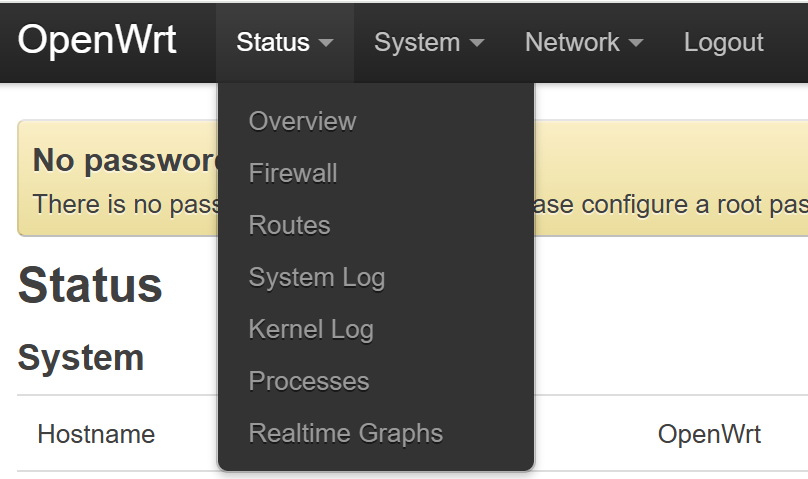
由于是基于imx6ull开发板实现路由器，不像一般的路由器那样有天线，所以不具备WIFI功能，但openwrt系统本身是可配置的。同时，如果在能上网的条件下，操作会很顺利且全面些。

**（1）Web管理界面介绍**

**一、Status**

系统状态：显示路由器运行时间、负载、内存和交换空间使用情况。

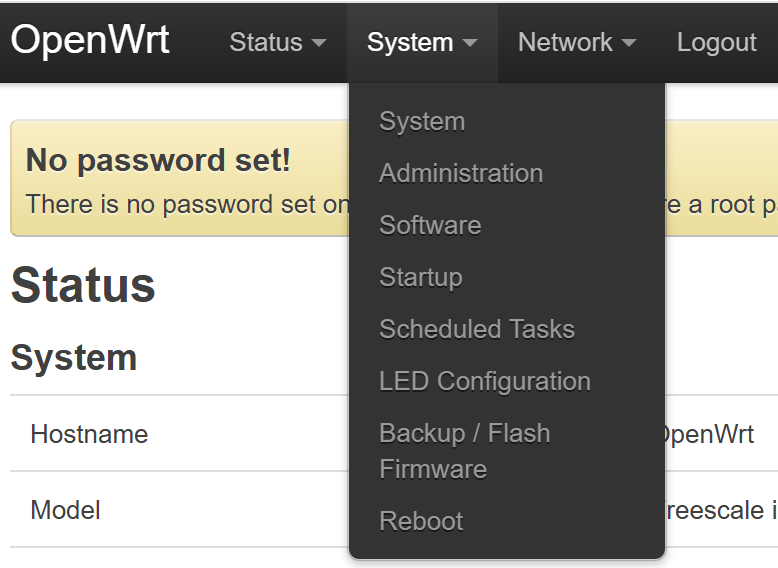
* **Overview（概览）**：展示路由器的整体运行状态，如 CPU 使用率、内存占用、已运行时间等系统关键信息，以及网络连接的基本情况。
* **Firewall（防火墙）**：查看防火墙的规则配置和运行状态，了解网络流量的过滤情况，可辅助排查网络安全相关问题。
* **Routes（路由）**：显示当前系统的路由表，能看到数据包在网络中传输的路径指向，有助于分析网络路由是否正常。
* **System Log（系统日志）**：记录系统运行过程中的各类事件和消息，像系统启动、服务运行、错误提示等，用于排查系统故障。
* **Kernel Log（内核日志）**：专注于内核层面的日志信息，内核相关的加载、驱动运行、异常等情况会在此记录，对深入分析系统底层问题有帮助。
* 所有系统程序和内核运行产生的消息都会在这里显示。如果某个功能不正常，首先来这两处找错误信息。
* **Processes（进程）**：列出当前系统中运行的所有进程，包括进程的 ID、占用资源情况等，可用于查看系统资源使用和进程运行状态。
* **Realtime Graphs（实时图表）**：以图形化的方式实时展示系统资源（如 CPU、内存、网络流量等）的使用情况，直观呈现系统负载变化。



**二、System**

系统：配置路由器本身的核心区域，涉及管理、软件和维护等系统基本设置，比如设置主机名、时区等系统标识和时间相关参数。

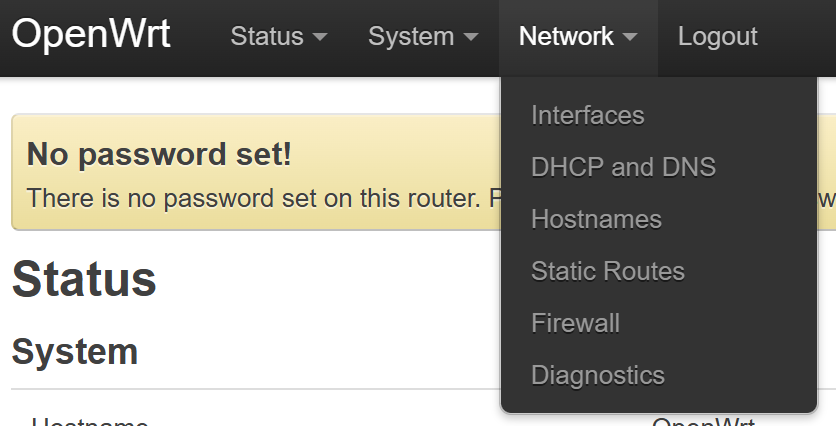
* **Administration（管理）**：主要用于管理路由器的访问权限，可设置 root 密码，还能配置 SSH 等远程访问方式的权限等。
* **Software（软件）**：管理 OpenWrt 系统的软件包，可安装、卸载、更新软件包，扩展路由器的功能。例如，如果你想安装内网穿透、广告屏蔽、异地组网等高级功能，都在这里操作。
* **Startup（启动项）**：配置系统启动时自动运行的服务和脚本，可设置服务的启动顺序等，定制系统启动后的初始运行状态。普通用户一般不需要改动。
* **Scheduled Tasks（计划任务）**：设置定时执行的任务，通过类似Linux cron 的机制，让系统在指定时间自动执行特定命令或脚本，实现自动化操作。
* **LED Configuration（LED 配置）**：对路由器上的 LED 指示灯进行配置，可设置不同状态（如网络连接、数据传输等）下 LED 灯的亮灭、颜色等，方便通过指示灯了解设备状态。
* **Backup / Flash Firmware（备份 / 刷写固件）**：
* Backup: 生成当前系统所有设置的备份文件（.tar 格式）。在进行任何重大修改前，一定要先在这里备份！
* Flash: 用于升级固件或完全重新刷写系统。
* **Reboot（重启）**：执行路由器的重启操作，当系统配置变更需要生效或排查一些简单故障时，可通过此功能重启设备。



**三、Network**

网络：配置所有网络功能的地方，大部分基础设置都在这里。

* **Interfaces（接口）**： 核心中的核心。管理所有网络接口，如 LAN (局域网)、WAN (广域网/互联网口)、Wi-Fi等。
* 可以在这里设置 LAN 的 IP 地址段、DHCP 服务器。
* 设置 WAN 的上网方式，如 PPPoE (ADSL拨号)、DHCP客户端(光猫拨号)、静态IP等。
* **DHCP and DNS（DHCP 和 DNS）**：配置路由器内置的 DHCP 服务器和 DNS 转发器。可以设置静态地址分配（为特定设备固定IP）、自定义DNS服务器（如使用 114.114.114.114 或 Google 的 8.8.8.8）。
* **Hostnames（主机名）**：查看和绑定客户端主机名与IP地址。管理局域网内设备的主机名与 IP 地址的映射关系，方便通过主机名访问设备，增强网络的可识别性。
* **Static Routes（静态路由）**：手动添加静态路由规则，指定特定目标网络的数据包传输路径，在复杂网络环境中优化网络传输，一般家庭用户用不到。
* **Firewall（防火墙）**：这里的防火墙配置更侧重于网络层面的规则设置，可创建、修改防火墙规则，配置网络流量过滤规则（端口转发、流量限制等），控制网络流量的进出，保障网络安全。
* **Diagnostics（诊断）**： 提供网络诊断工具，如 Ping、Traceroute、Nslookup，用于测试网络连通性。



**（2）OpenWrt 配置与管理核心要点**

1.如何通过命令行来配置和管理

opkg（管理工具）安装、管理软件包；

uci 管理openwrt配置文件；

ifconfig 网络配置；

logread/dmesg 查看日志。

2.LuCI Web界面配置——基本配置

[[OpenWrt Wiki] 用户指南](https://openwrt.org/zh/docs/guide-user/start)

“Restore”主要是恢复配置，而不是恢复固件版本。

3.网络基础配置

端口转发和NAT（网络地址转换）：用于内部设备与外部网络之间的通信；

VLAN（虚拟局域网）：网络隔离、带宽控制等；

防火墙：iptables/nftables管理防火墙；

需要理解网络流量的过滤和路由规则。

4.通过opkg安装软件包，扩展路由器

需要知道如何安装、更新、删除软件包；

固件更新：如何刷机、恢复出厂设置、备份、还原配置等。

5.常见问题和故障排查

6.命令行UCI的配置和LuCI Web界面的配置是等价的。

7.应用开发：将一个程序的源代码通过openwrt提供的专用工具链（Toolchain），打包成openwrt系统可识别和安装的.ipk格式软件，可传到路由器上。安装和管理扩展包（软件包）为其中的一部分。