

在昉·星光 2 上编译和安装 OpenWrt

本手册为用户提供在赛昉科技昉·星光 2 上编译和安装 OpenWrt 的方法。

下载

1. 下载代码（建议在 ubuntu20 或以上的版本编译）：

```
git clone https://git.openwrt.org/openwrt/staging/wigori.git
```

2. 输入以下命令，进入 wigori 目录：

```
cd wigori/
```

3. 输入以下命令，获取镜像：

```
git checkout riscv-jh71x0-202308b
```

配置

1. 输入以下命令进行基本配置。

```
make menuconfig
```

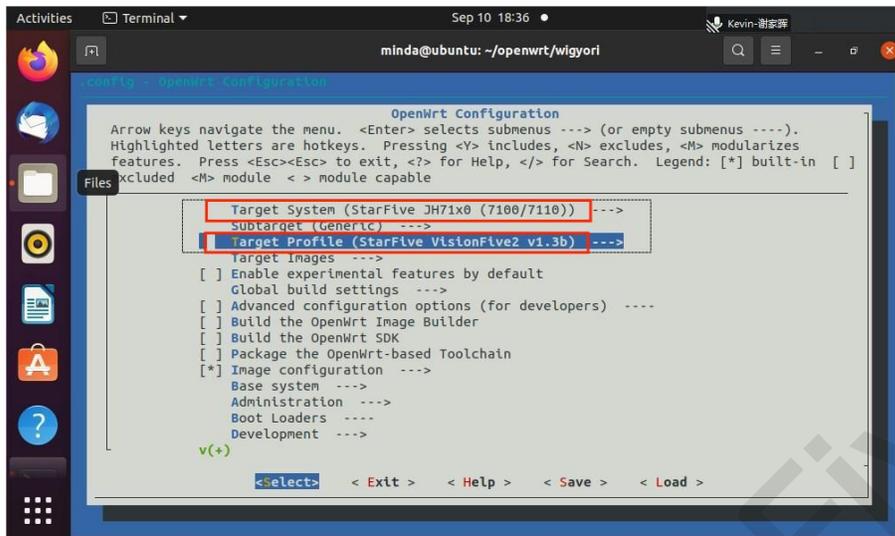
在配置菜单栏选项中，根据以下步骤选择对应的选项并进入：

- a. Target System/Profile 配置：

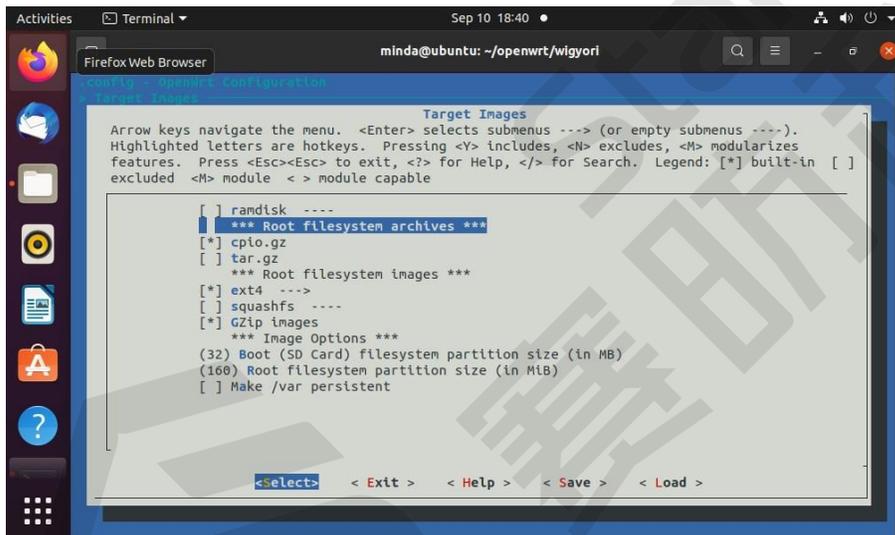
-> Target System (StarFive JH71x0 (7100/7110))

-> Target Profile (StarFive VisionFive 2 v1.2a) 或 Target Profile (StarFive VisionFive 2 v1.3b)

注：昉·星光 2 目前有两个版本，v1.2a 和 v1.3b。



b. Target Image 配置:



2. luci 网页配置。

a. 执行以下命令默认安装 luci:

```
./scripts/feeds update packages luci
```

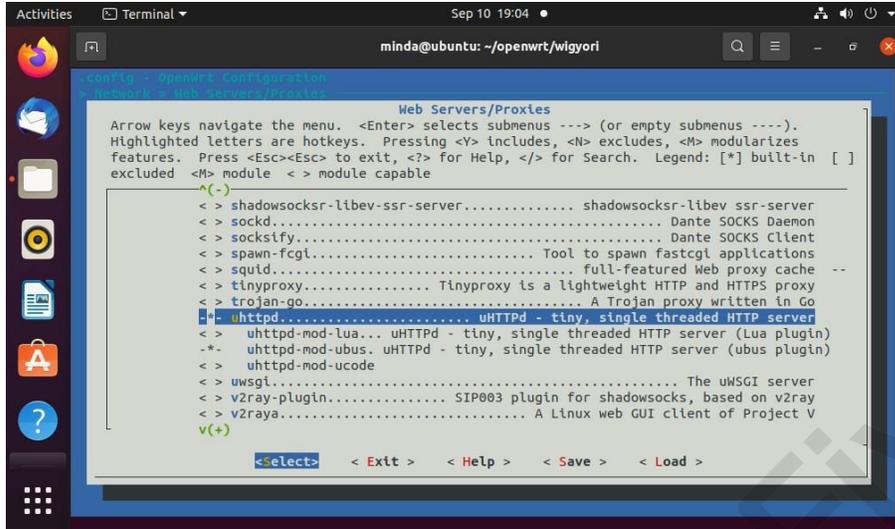
```
./scripts/feeds install -a -p luci
```

b. 输入以下命令进行 luci 网页配置:

```
make menuconfig
```

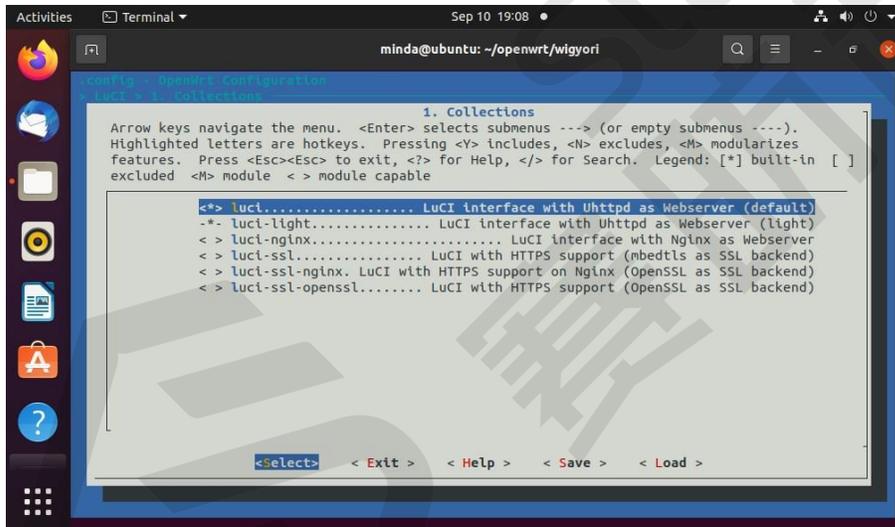
c. 在配置菜单栏按照以下顺序选择 uhttpd:

```
-> Network -> Web Servers/Proxies -> <*> uhttpd
```

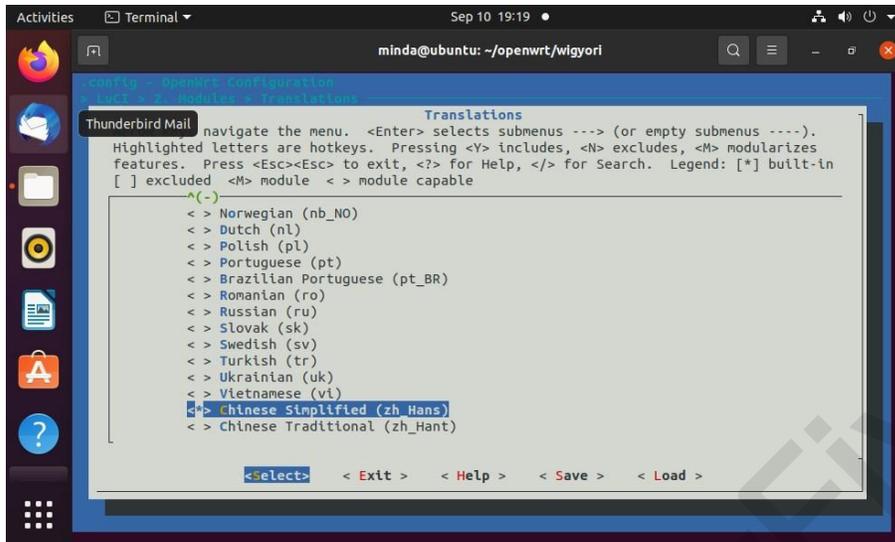


d. 根据以下顺序选择选项进行 Luci 配置：

- -> LuCI -> 1. Collections -> <*> luci



- -> LuCI -> 2. Modules -> Translations -> <*> Chinese Simplified (zh_Hans)



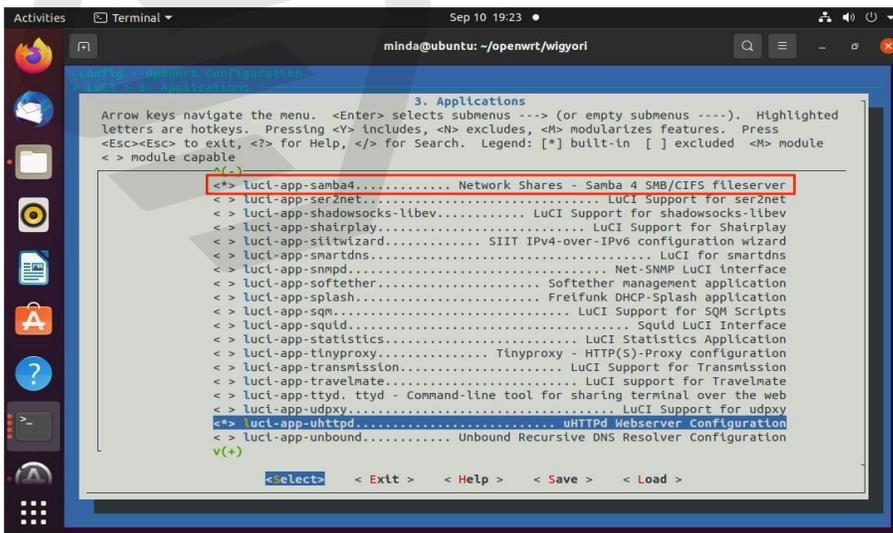
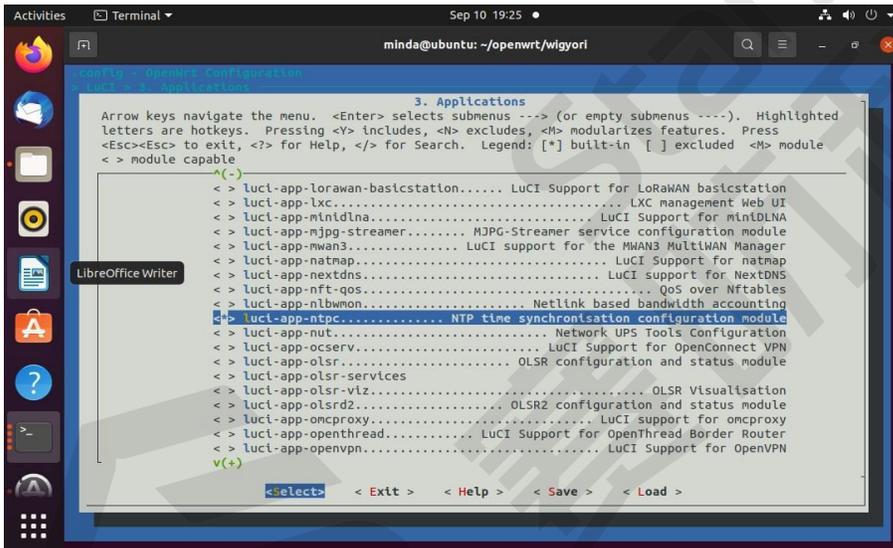
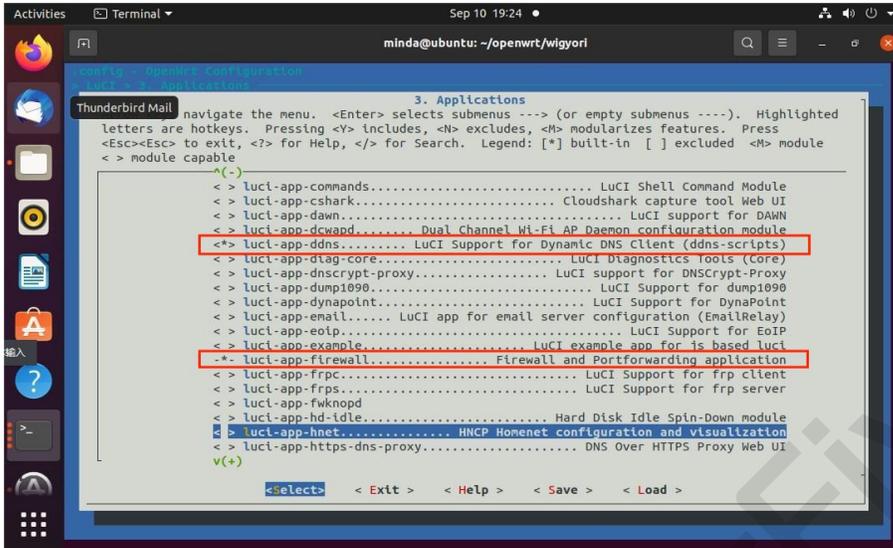
- -> LuCI -> 3. Applications -> <* > luci-app-ddns

 - <* > luci-app-firewall

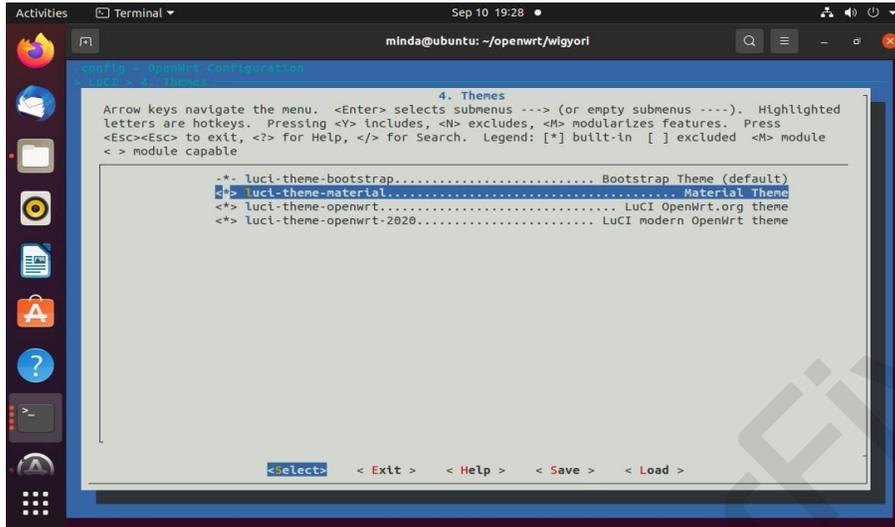
 - <* > luci-app-ntpc

 - <* > luci-app-samba4

 - <* > luci-app-uhttpd



- -> LuCI -> 4. Themes -> 全选



3. 无线配置。

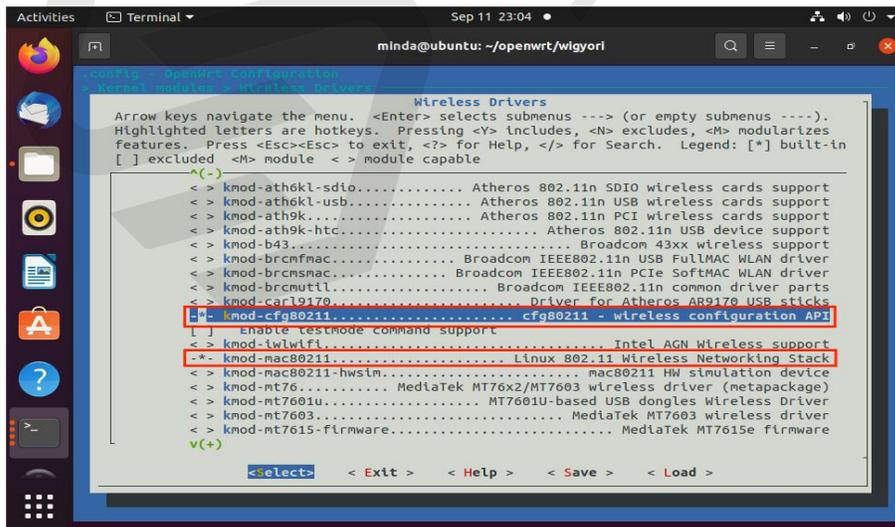
由于昉·星光 2 上没有 WiFi 模块，需要购置[无线 WiFi 模组](#)。OpenWrt 支持数十种 WiFi 驱动，这里选用较易购置的 RTL8821AE PCI 接口的驱动。昉·星光 2 上有 m.2 的 M key 的接口，需要再接 nvme m.2 的 M key 转 a/e key 的接口板才能接上 RTL8821AE 模组。

在配置菜单栏按照如下顺序进行无线配置：

a. -> Kernel modules -> Wireless Driver -> <*> kmod-cfg80211

<*> kmod-mac80211

<*> kmod-rtl8821ae



```

Activities Terminal Sep 10 19:42 mnda@ubuntu: ~/openwrt/wlgyori
OpenWRT - OpenWRT Configuration
Kernel modules: Wireless Drivers
Wireless Drivers
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, <?> for Help, </> for Search. Legend: [*] built-in
[ ] excluded <M> module <> module capable
v(+)
```

- <> kmod-mwl8k..... Driver for Marvell TOPDOG 802.11 Wireless cards
- <> kmod-owl-loader.... Owl loader for initializing Atheros PCI(e) Wlfi chips
- <> kmod-rs191x..... Redpine Signals Inc 91x WLAN driver support
- <> kmod-rs191x-sdio..... Redpine Signals SDIO bus support
- <> kmod-rs191x-usb..... Redpine Signals USB bus support
- <> kmod-rt2800-pci..... Ralink Drivers for RT2x00 cards (RT2800 PCI)
- <> kmod-rt2800-usb..... Ralink Drivers for RT2x00 cards (RT2870 USB)
- <> kmod-rt2x00-lib..... Ralink Drivers for RT2x00 cards (LIB)
- <> kmod-rtl8192ce..... Realtek RTL8192CE/RTL8188CE support
- <> kmod-rtl8192cu..... Realtek RTL8192CU/RTL8188CU support
- <> kmod-rtl8192de..... Realtek RTL8192DE/RTL8188DE support
- <> kmod-rtl8192se..... Realtek RTL8192SE/RTL8191SE support
- <> kmod-rtl8723bs.. Realtek RTL8723BS SDIO Wireless LAN NIC driver (staging)
- <> kmod-rtl8812au-ct. Driver for Realtek 8812 AU devices comfast 912-ac_etc
- <M> kmod-rtl8821ae..... Realtek RTL8821AE support
- <> kmod-rtl8xxxu..... alternative Realtek RTL8XXU support
- [] Realtek wireless debugging
- <> kmod-rtw88..... Realtek RTL8822BE/RTL8822CE/RTL8723DE

```

v(+)
```

- b. -> Network -> WirelessAPD -> <*> wpad-basic-mbedtls
- <*> hostapd-common
 - <*> wpa-cli
 - <*> hostapd-utils

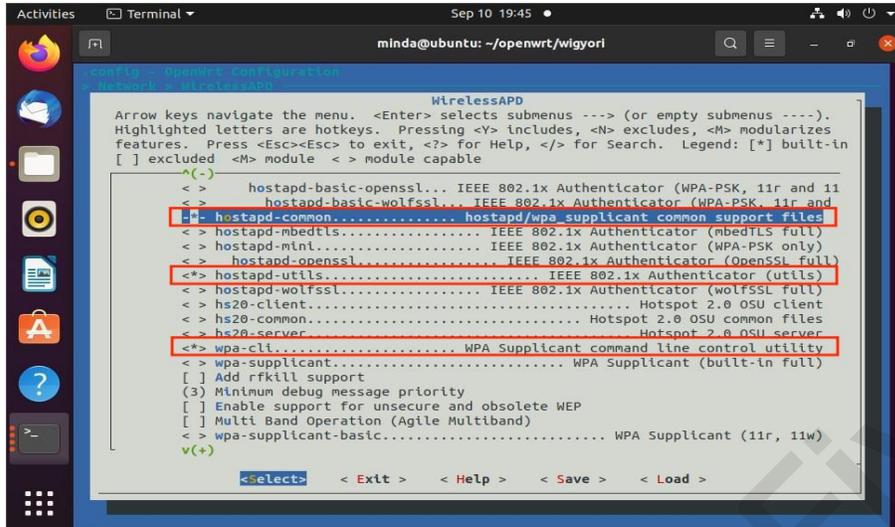
```

Activities Terminal Sep 10 19:46 mnda@ubuntu: ~/openwrt/wlgyori
OpenWRT - OpenWRT Configuration
Kernel modules: Wireless APD
Wireless APD
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, <?> for Help, </> for Search. Legend: [*] built-in
[ ] excluded <M> module <> module capable
v(+)
```

- <> wpa-supplciant-mbedtls..... WPA Supplciant (mbedtls full)
- <> wpa-supplciant-mesh-mbedtls..... WPA Supplciant (mbedtls, 11s, SAE)
- <> wpa-supplciant-mesh-openssl..... WPA Supplciant (openssl, 11s, SAE)
- <> wpa-supplciant-mesh-wolfssl..... WPA Supplciant (wolfssl, 11s, SAE)
- <> wpa-supplciant-mini..... WPA Supplciant (minimal)
- <> wpa-supplciant-openssl..... WPA Supplciant (openssl full)
- <> wpa-supplciant-p2p..... WPA Supplciant (Wi-Fi P2P support)
- <> wpa-supplciant-wolfssl..... WPA Supplciant (wolfssl full)
- <> wpad..... IEEE 802.1x Auth/Supplciant (built-in full)
- <> wpad-basic..... IEEE 802.1x Auth/Supplciant (WPA-PSK, 11r, 11w)
- <M> wpad-basic-mbedtls..... IEEE 802.1x Auth/Supplciant (mbedtls, 11r, 11w)
- <> wpad-basic-openssl..... IEEE 802.1x Auth/Supplciant (openssl, 11r, 11w)
- <> wpad-basic-wolfssl..... IEEE 802.1x Auth/Supplciant (wolfssl, 11r, 11w)
- <> wpad-mbedtls..... IEEE 802.1x Auth/Supplciant (mbedtls)
- <> wpad-mesh-mbedtls..... IEEE 802.1x Auth/Supplciant (mbedtls, 11s, SAE)
- <> wpad-mesh-openssl..... IEEE 802.1x Auth/Supplciant (openssl, 11s, SAE)
- <> wpad-mesh-wolfssl..... IEEE 802.1x Auth/Supplciant (wolfssl, 11s, SA)
- <> wpad-mini..... IEEE 802.1x Auth/Supplciant (WPA-PSK only)

```

v(+)
```



4. 执行以下命令，下载需要的软件包。

```
./scripts/feeds update -a  
./scripts/feeds install -a  
make download V=s
```

注：下载时间可能较长，请耐心等待。下载过程中如出现 `download fail` 报错，表示软件包没有下载完整，请再次执行下载命令，直到没有分出现 `download fail` 为止。

5. 编译。

```
make -j8
```

注：编译过程可能需要 2 个小时。

6. 生成 SD 卡镜像。

```
bin/targets/jh71x0/generic/  
openwrt-jh71x0-generic-visionfive2-v1.3b-ext4-sdcard.img.gz
```

7. 解压该 SD 卡镜像文件。

```
gunzip openwrt-jh71x0-generic-visionfive2-v1.3b-ext4-sdcard.img.gz
```

8. 将镜像刷写到 SD Card。

```
dd if=openwrt-jh71x0-generic-visionfive2-v1.3b-ext4-sdcard.img of=/dev/sdX bs=1M  
oflag=direct
```

注：

- Windows 系统可用 balenaEtcher 软件烧写工具。
- 由于镜像不包括 SPL 和 U-Boot，所以 Nor Flash 需要已刷写 [SPL 和 U-Boot](#)，并且选择 QSPI Nor Flash 模式启动。
- 使用命令 `sudo fdisk -l` 查询，根据您的设备替换 X 的值。例如，您的设备为 `/dev/sdb`，则 X 的值应为 b。

9. 启动 OpenWrt。

10. 增加 passwall 功能。

如果需要增加 passwall 功能，可以按以下步骤，passwall 功能不在默认的功能里面，需要修改代码和下载包支持。

a. 在 wigori 目录下：

```
echo "src-git passwall_packages https://github.com/xiaorouji/openwrt-passwall-packages.git;main" >> "feeds.conf.default"
echo "src-git passwall https://github.com/xiaorouji/openwrt-passwall.git;main" >> "feeds.conf.default"
```

b. 修改 `include/target.mk`：

```
diff --git a/include/target.mk b/include/target.mk
index b5e3e7ff6f..f65e127ecf 100644
--- a/include/target.mk
+++ b/include/target.mk
@@ -53,7 +53,7 @@ DEFAULT_PACKAGES.nas:=\
     mdadm
 # For router targets
 DEFAULT_PACKAGES.router:=\
-     dnsmasq \
+     dnsmasq-full \
```

c. 下载并安装 passwall 包：

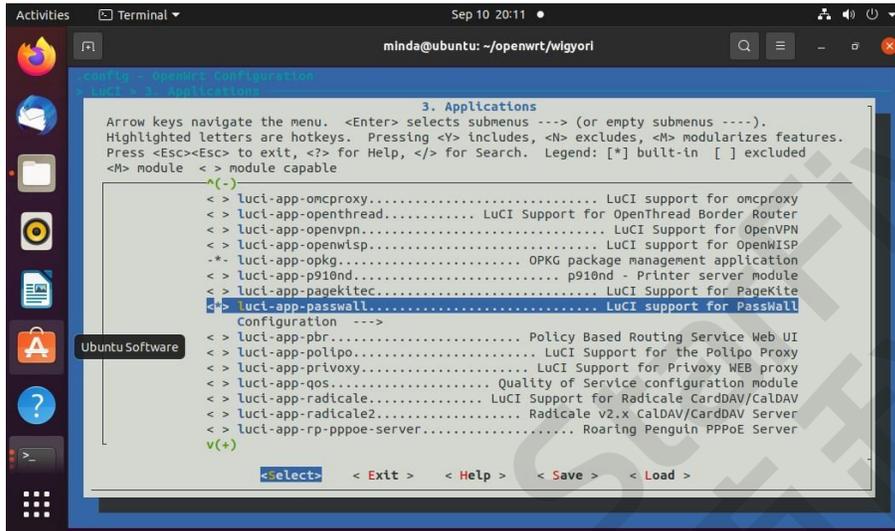
```
./scripts/feeds update -a
./scripts/feeds install -a
./scripts/feeds install -a -f -p PWpackages
./scripts/feeds install luci-app-passwall
```

d. 配置 passwall 和取消配置 dnsmasq:

```
make menuconfig
```

- 配置 passwall:

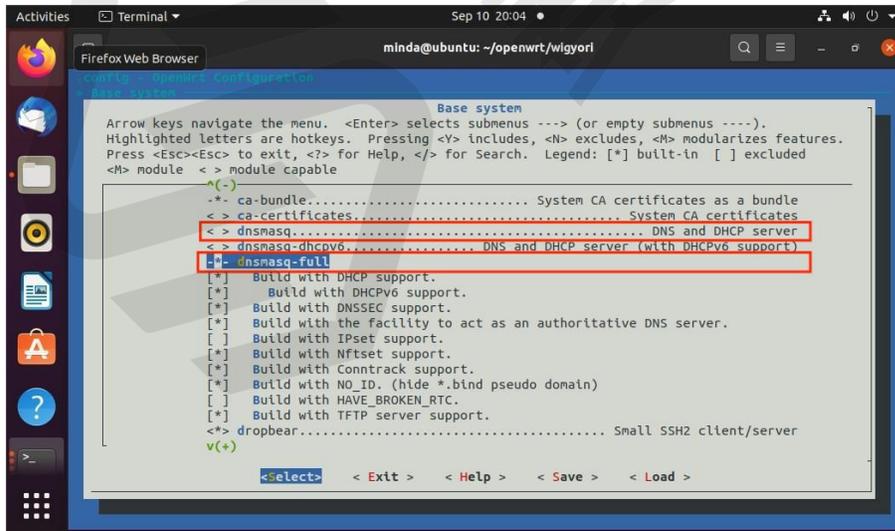
-> OpenWrt Configuration ->LuCI -> 3. Application -> luci-app-passwall



- 取消配置 dnsmasq:

-> OpenWrt Configuration -> Base system -> <>dnsmasq

-> OpenWrt Configuration -> Base system -> -*- dnsmasq-full



e. 更新下载软件包:

```
make download V=s
```

f. 编译:

方法同[第 5 步](#)。