

 xiaonui 10月 '18 编辑

OpenWrt is an open source linux distribution for embedded devices,

It has support a lot of hardwares, I got a new device called LiChee Pi Nano these days.

So I just add a few patches and template in OpenWrt v18.06.1, it runs perfect well.

This is the way to compile the image from the source code.

1 Clone the source code

```
cd ~
git clone -b licheepi-nano https://github.com/qinfengling/openwrt.git
```

2 Update the configuration for Lichee Pi Nano

```
cd ~/openwrt
cat > .config << EOF
CONFIG_TARGET_sunxi=v
CONFIG_TARGET_sunxi_arm9=v
CONFIG_TARGET_sunxi_arm9_DEVICE_suniv-f1c100s-licheepi-nano=y
CONFIG_DEFVFI=v
CONFIG_BRCMFMAC_SDIO=y
CONFIG_BRCMFMAC_USB=v
CONFIG_DRTVFR_11AC_SUPPORT=y
CONFIG_DRTVFR_11N_SUPPORT=v
CONFIG_PACKAGE_MAC80211_DEBUGFS=y
CONFIG_PACKAGE_MAC80211_MSHV=v
CONFIG_PACKAGE_brcmfmac-firmware-usb=y
CONFIG_PACKAGE_hostapd-common=y
CONFIG_PACKAGE_iw=v
CONFIG_PACKAGE_iwinfo=v
CONFIG_PACKAGE_kmod-ata-ahci-nlatform=y
CONFIG_PACKAGE_kmod-ata-core=v
CONFIG_PACKAGE_kmod-ata-sunxi=y
CONFIG_PACKAGE_kmod-brcmfmac=v
CONFIG_PACKAGE_kmod-brcmutil=v
CONFIG_PACKAGE_kmod-cf80211=y
CONFIG_PACKAGE_kmod-libahci=v
CONFIG_PACKAGE_kmod-mmc=v
CONFIG_PACKAGE_kmod-nls-base=y
CONFIG_PACKAGE_kmod-ntf-mdio=v
CONFIG_PACKAGE_kmod-rtc-sunxi=v
CONFIG_PACKAGE_kmod-rtl8192c-common=y
CONFIG_PACKAGE_kmod-rtl8192cu=y
CONFIG_PACKAGE_kmod-rtl8xxxu=y
CONFIG_PACKAGE_kmod-rtlwifi=v
CONFIG_PACKAGE_kmod-rtlwifi-usb=y
CONFIG_PACKAGE_kmod-scsi-core=v
CONFIG_PACKAGE_kmod-sun4i-emac=y
CONFIG_PACKAGE_kmod-usb-core=y
CONFIG_PACKAGE_lhiwinfo=v
CONFIG_PACKAGE_rtl8188eu-firmware=v
CONFIG_PACKAGE_rtl8192cu-firmware=y
CONFIG_PACKAGE_swconf=v
CONFIG_PACKAGE_wireless-regdb=y
CONFIG_PACKAGE_wpad-mini=y
CONFIG_SOFT_FLOAT=v
CONFIG_TARGET_OPTIONS=y
EOF
```

3 Generate the image

```
make world
```

TIP: You can also speed up the compile with -j option.

When you finish the compile without any failure, you can find the image (openwrt-sunxi-arm9-suniv-f1c100s-licheepi-nano-ext4-sdcard.img.gz) under `./bin/targets/sunxi/arm9/`,

NOTE: It is an image for TF card, you need to decompress it first, then write it to the TF card.
ths [@qinfengling](#)