



# NanoPi R4S

## URLs

### Hardware

- [https://www.friendlyelec.com/index.php?route=product/product&product\\_id=284](https://www.friendlyelec.com/index.php?route=product/product&product_id=284)
- [https://wiki.friendlyelec.com/wiki/index.php/NanoPi\\_R4S#Essentials\\_You\\_Need](https://wiki.friendlyelec.com/wiki/index.php/NanoPi_R4S#Essentials_You_Need)

### FriendlyWRT

- <https://wiki.friendlyelec.com/wiki/index.php/FriendlyWrt>

## Install OS

- Download: <http://download.friendlyelec.com/NanoPiR4S>
- File: rk3399-sd-friendlywrt-23.05-docker-20231031.img.gz

## Flash Utility

- dd
- win32diskimager
- [balenaEtcher](#)

## First Boot

- Account: root / password (empty password in some versions)
- Web:
  - <http://friendlywrt/>
  - <http://192.168.2.1/>

## Hardware SPEC.

- SoC: Rockchip RK3399

- CPU: big.LITTLE?Dual-Core Cortex-A72(up to 2.0GHz) + Quad-Core Cortex-A53(up to 1.5GHz)
- GPU: Mali-T864 GPU?supports OpenGL ES1.1/2.0/3.0/3.1, OpenCL, DX11, and AFBC
- VPU: 4K VP9 and 4K 10bits H265/H264 60fps decoding, Dual VOP, etc
- PMU: RK808-D PMIC, cooperated with independent DC/DC, enabling DVFS, software power-down, RTC wake-up, system sleep mode
- RAM: 1GB DDR3/4GB LPDDR4
- Flash: no Onboard eMMC
- Ethernet: one Native Gigabit Ethernet, and one PCIe Gigabit Ethernet
- USB: two USB 3.0 Type-A ports
- Pin header extension interface
  - 2x5-pin header: SPI x 1, I2C x 1
  - 4-pin header: USB 2.0
- microSD Slot x 1
- Debug: one Debug UART, 3 Pin 2.54mm header, 3V level, 1500000bps
- LEDs: 1 x power LED and 3 x GPIO Controlled LED (SYS, LAN, WAN)
- others:
  - 2 Pin 1.27/1.25mm RTC battery input connector
  - one User Button
  - one 5V Fan connector
- Power supply: DC 5V/3A, via USB-C connector or Pin header
- PCB: 8 Layer, 66 mm x 66 mm
- Temperature measuring range: 0? to 80?

# OpenWRT One

## URLs

- [\[OpenWrt Wiki\] OpenWrt One](#)
- [\[OpenWrt Wiki\] Quick start guide for OpenWrt installation](#)
- [Banana Pi OpenWrt One Router | BananaPi Docs](#)

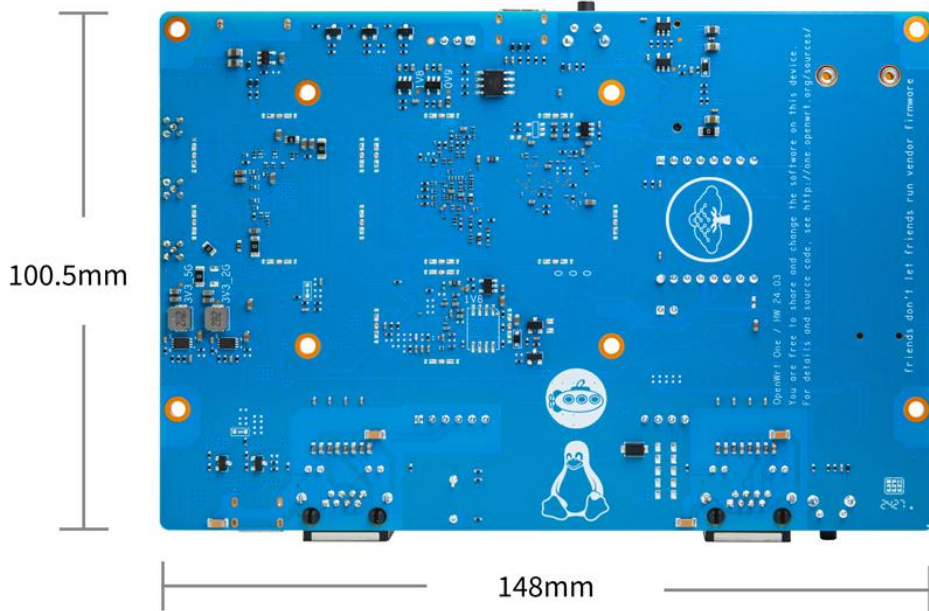
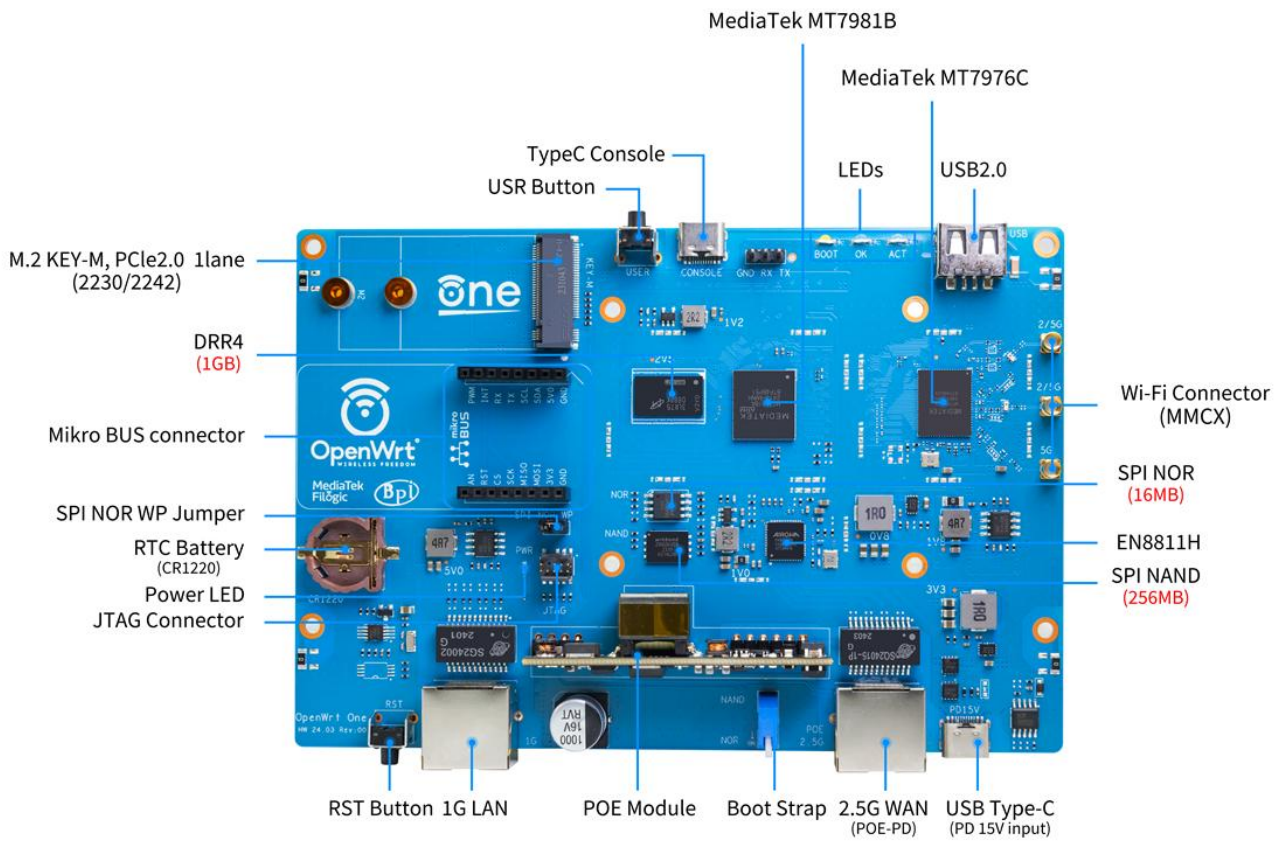
## Hardware

### Specification

SoC	MediaTek MT7981B (Filogic 820) dual-core Cortex-A53 processor @ 1.3 GHz
System Memory	1GB DDR4
Storage	<ul style="list-style-type: none"><li>• 256 MB SPI NAND flash for U-boot and Linux</li><li>• 16 MB SPI NOR flash for write-protected (by default) recovery bootloader (reflashing can be enabled with a jumper)</li><li>• Two types of flash devices are used to make the board almost unbrickable</li><li>• M.2 2242/2230 socket for NVMe SSD (PCIe gen 2 x1)</li></ul>
Networking	<ul style="list-style-type: none"><li>• 1 x 2.5GbE RJ45 port</li><li>• 1 x Gigabit Ethernet RJ45 port</li><li>• Dual-band WiFi 6 via MediaTek MT7976C (2x2 2.4 GHz + 3x3/2x2 + zero-wait DFS 5Ghz)</li><li>• 3x MMCX antenna connectors</li></ul>
USB	<ul style="list-style-type: none"><li>• 1x USB 2.0 Type-A host port</li><li>• USB Type-C (device, console) port using Holtek HT42B534-2 UART to USB chip</li></ul>
Expansion	MikroBUS socket for expansion modules
Debugging	Console via USB-C port or 3-pin header, 10-pin JTAG/SWD header for main SoC
RTC	support RTC onboard

Misc	<ul style="list-style-type: none"><li>• Reset and User buttons</li><li>• Boot select switch: NAND (regular) or NOR (recovery)</li><li>• 2x PWM LEDs, 2x Ethernet LED (GPIO driven)</li><li>• EM6324 External hardware watchdog</li><li>• NXP PCF8563TS (I2C) RTC with battery backup holder for CR1220 coin-cell</li></ul>
Power Supply	<ul style="list-style-type: none"><li>• 15V USB-PD on USB-C port</li><li>• Optional 802.3at/af PoE via RT5040 module</li></ul>
Dimensions	148 x 100.5 mm compatible with Banana Pi BPI-R4 case design
Certifications	FCC/EC/RoHS compliance

## Interface



Installation

Firmware Download

- Download1: [https://firmware-selector.openwrt.org/?version=24.10.1&target=mediatek%2Ffilogic&id=openwrt\\_one](https://firmware-selector.openwrt.org/?version=24.10.1&target=mediatek%2Ffilogic&id=openwrt_one)
- Download2: <https://downloads.openwrt.org/releases/>

## First Power up

?????????Openwrt One?????????????????????OpenWrt??????

1. ????????????? NAND/NOR ?????? NAND
2. ??????? 192.168.1.1 ??????? 1G ???
3. ?????????????????????????????
4. ??????? 192.168.1.1 ??? LuCI GUI????????????? `ssh root@192.168.1.1`?

## FAQ

### Why are there are 2 different flash chips?

The idea is to make the device (almost!) unbrickable and very easy to recover.

- NAND will hold the main loader (U-Boot) and the Linux image and will be the default boot device
- NOR will be write-protected by default (with WP jumper available on the board) and will hold a recovery bootloader (and other essential data, like Wi-Fi calibration)
- a dedicated boot select switch will allow changing between NOR and NAND

### What will the M.2 slot be used for?

We will use M.2 with M-key for NVMe storage. There is a work-in-progress patch to make PCIe work inside the U-Boot bootloader. This will allow booting other Linux distributions such as Debian and Alpine directly from NVMe.

### Why is there no USB 3.x host port on the device?

The USB 3.x and PCIe buses are shared in the selected SoC silicon, hence only a single High-Speed USB port is available

### What is the purpose of the console USB-C port?

Holtek UART to USB bridge with CDC-ACM support on USB-C makes the device ultra easy to communicate with. No extra hardware or drivers will be required. Android for example has CDC-ACM support enabled by default.

## What MAC OUI will the device have?

We plan to register an OUI block for OpenWrt which can also be used for other vendor extensions such as Wi-Fi beacon IEs.

## What is the purpose of the mikroBUS connector?

mikroBUS was chosen as we wanted to make the hardware extendable. There are dedicated pins for UART, SPI, I2C buses and RST/INT signals. The standard uses regular 2.54 mm pitch connectors (you can use available mikroBUS modules or just connect to it something else, with 2.54 mm jumper cables).

## Why have the RTC on board instead of a mikroBUS module?

We believe there are many things a Wi-Fi (or networking in general) device should have on-board by default. Always having a correct time on the device is crucial in many applications, like VPN, DNSSEC, ...

# Upgrade Firmware

## Download

??????????

- Download1: [https://firmware-selector.openwrt.org/?version=24.10.1&target=mediatek%2Ffilogic&id=openwrt\\_one](https://firmware-selector.openwrt.org/?version=24.10.1&target=mediatek%2Ffilogic&id=openwrt_one)
- Download2: <https://downloads.openwrt.org/releases/>

## Using LuCI GUI

- Firmware ???sysupgrade
- Upgrade: LuCI Web ? System ? Backup / Flash Firmware ? Flash new firmware image

## Using CLI

- Firmware ???sysupgrade
- sysupgrade ??????<https://openwrt.org/docs/techref/sysupgrade>

## Command

```
# example downloading the OpenWrt 15.05 upgrade image for a TP-LINK TL-WR1043ND ver. 1.x
router
cd /tmp
wget http://downloads.openwrt.org/chaos_calmer/15.05/ar71xx/generic/openwrt-15.05-ar71xx-
generic-tl-wr1043nd-v1-squashfs-sysupgrade.bin

# check the integrity of the image file via md5sums (older images)
wget http://downloads.openwrt.org/chaos_calmer/15.05/ar71xx/generic/md5sums
md5sum -c md5sums 2> /dev/null | grep OK

# check the integrity of the image file via sha256sums
wget http://downloads.openwrt.org/chaos_calmer/15.05/ar71xx/generic/sha256sums
sha256sum -c sha256sums 2> /dev/null | grep OK

# the desired result is that the downloaded firmware filename is listed with "OK" afterwards
```

```
#####  
# Initiate sysupgrade with your desired options  
# by default ( no -n ) settings are kept  
#####  
sysupgrade -v /tmp/openwrt-15.05-ar71xx-generic-tl-wr1043nd-v1-squashfs-sysupgrade.bin
```

## Using USB drive

### OpenWRT One

1. prepare a FAT32 formatted USB drive that contains the `sysupgrade.itb` file from either the SNAPSHOT or Release repositories
2. remove power
3. insert the USB stick in the Type A USB Port.
4. make sure **NAND boot** switch is selected.
5. press and hold the button on the back side labeled **Reset**
6. power up the device. Release the **Reset** button as soon as all LEDS turn off.
7. wait for the middle LED to go green.

The device will boot from NAND and the bootloader will reflash the kernel and root filesystem on the NAND.

# Network Hardening

## Solutions

- [banIP](#)
- [Fail2Ban](#)

# VPN

## PPTP

- [PPTP Server](#)

## PPTP in LAN not working

LAN ?????????????? PPTP VPN Server

???OpenWRT ?????? PPTP ?????? `kmod-nf-nathelper-extra`?

```
opkg update
opkg install kmod-nf-nathelper-extra
```

????????????????

## OpenVPN

- [\[OpenWrt Wiki\] OpenVPN](#)
- YT: [OpenWRT - VPN into your Home network using OpenVPN | Roadwarrior - YouTube](#)

## OpenVPN Server

Preparation

```
# Install packages
opkg update
opkg install openvpn-openssl openvpn-easy-rsa luci-app-openvpn luci-i18n-openvpn-zh-tw
```

## Generate PKI (Public Key Infrastructure)

```
# Configuration parameters
cat << EOF > /etc/profile.d/50-openvpn-easy-rsa.sh
export EASYRSA_PKI="/etc/openvpn/pki"
export EASYRSA_TEMP_DIR=${EASYRSA_TEMP_DIR:-${TMPDIR:-/tmp/}}
export EASYRSA_CERT_EXPIRE="3650"
```

```

export EASYRSA_BATCH="1"
EOF
. /etc/profile.d/50-openvpn-easy-rsa.sh

# Remove and re-initialize PKI directory
easyrsa init-pki

# Generate DH parameters
easyrsa gen-dh

# Create a new CA
easyrsa build-ca nopass

# Generate server keys and certificate
easyrsa build-server-full server nopass
openvpn --genkey tls-crypt-v2-server ${EASYRSA_PKI}/server.pem

# Generate client keys and certificate
easyrsa build-client-full client nopass
openvpn --tls-crypt-v2 ${EASYRSA_PKI}/server.pem \
--genkey tls-crypt-v2-client ${EASYRSA_PKI}/client.pem

```

## OpenVPN Service Configuration

1. LuCI UI ? VPN ? OpenVPN ? Delete : custom\_config/sample\_server/sample\_client
2. LuCI UI ? VPN ? OpenVPN ? Add : Template based configuration
  - Name : ovpnServer
  - Template : Server configuration for a routed multi-client VPN
3. LuCI UI ? VPN ? OpenVPN ? Edit : ovpnServer
  - server : 10.9.8.0 255.255.255.0 ( ??? tun ????)
  - ca : /etc/openvpn/pki/ca.crt
  - dh : /etc/openvpn/pki/dh.pem
  - cert : /etc/openvpn/pki/issued/server.crt
  - key : /etc/openvpn/pki/private/server.key
  - port : 1194
  - proto : UDP
  - dev\_type : tun
  - client\_to\_client : check
4. LuCI UI ? VPN ? OpenVPN ? Edit : ovpnServer (**Advanced configuration**)
  1. Cryptography
    - tls\_crypt\_v2 : /etc/openvpn/pki/server.pem
  2. Networking
    - persist\_tun : check

- persist\_key : check
- topology : subnet

### 3. VPN

- client\_to\_client : check
- duplicate\_cn : check
- push : route 192.168.8.0 255.255.255.0 (??? LAN ??)
- push : redirect-gateway

## Firewall Configuration

### 1. LuCI UI ? Network ? Firewall ? Traffic Rules ? Add:

- Name : Allow-OpenVPN
- Protocol : UDP
- Source zone : wan/wan6
- Destination zone : Device (input)
- Destination port : 1194
- Action: accept

### 2. LuCI UI ? Network ? Firewall ? General Settings ? Edit: lan ? Advances Settings

- Covered devices : tun0

## Generate client configuration file

```

VPN_CONF="/etc/openvpn/client.ovpn"
VPN_SERV="192.168.0.12"
VPN_PORT="1194"
VPN_PROTO="udp"
VPN_TC="$(cat /etc/openvpn/pki/server.pem)"
VPN_KEY="$(cat /etc/openvpn/pki/private/server.key)"
VPN_CERT="$(openssl x509 -in /etc/openvpn/pki/issued/server.crt)"
VPN_CA="$(openssl x509 -in /etc/openvpn/pki/ca.crt)"
cat << EOF > ${VPN_CONF}
remote ${VPN_SERV} ${VPN_PORT} ${VPN_PROTO}
dev tun
nobind
client
auth-nocache
remote-cert-tls server
<tls-crypt-v2>
${VPN_TC}
</tls-crypt-v2>
<key>
${VPN_KEY}

```

```
</key>
<cert>
${VPN_CERT}
</cert>
<ca>
${VPN_CA}
</ca>
EOF
```

## Wireguard

- [\[OpenWrt Wiki\] WireGuard](#)
- YT: [Configuring Wireguard on OpenWRT - Step by Step Guides - YouTube](#)
- YT: [WireGuard - How to Install and Configure WireGuard VPN Client on Ubuntu | Debian | LinuxMint - YouTube](#)

### Preparation

```
opkg update
opkg install wireguard-tools kmod-wireguard luci-proto-wireguard qrencode
reboot
```

### Create Wireguard Interface

1. LuCI ? Network ? Interfaces ? Add new interface
  - Name : Wireguard
  - Protocol : Wireguard VPN
2. LuCI ? Network ? Interfaces ? Wireguard ? General Settings
  - Generate new key pair
  - Listen Port : 51820
  - IP Addresses : 10.9.7.1/24
3. LuCI ? Network ? Interfaces ? Wireguard ? Advanced Settings
  1. Use custom DNS servers : 8.8.8.8
4. Save & Apply
5. LuCI ? Network ? Interfaces ? Devices ? Configure: Wireguard
  1. Save
6. Save & Apply

### Configure Firewall

1. LuCI ? Network ? Firewall ? Add zone
  - Name :
  - Input/Output/Forward : Accept
  - Masquerading : check
  - MSS Clamping : check
  - Covered networks : lan/Wireguard
  - Allow forward to destination zones : wan/wan6
  - Allow forward from source zones : lan
  - Save
2. Save & Apply

## Configure Port Forwarding

1. LuCI ? Network ? Firewall ? Port Forwards ? Add
  - Name : Wireguard
  - Restrict to address family : automatic
  - Protocol : TCP/UDP
  - Source zone : wan/wan6
  - External port : 51820
  - Destination zone : lan
  - Internal IP address : 10.9.7.1
  - Internal port : 51820
  - Save
2. Save & Apply

## Configure Peer Settings

1. LuCI ? Network ? Interfaces ? Edit: Wireguard ? Peers ? Add peer
  - Description : My Linux Fedora
  - Generate new key pair
  - Allowed IPs : 10.9.7.2/32
  - Save ? Save
2. Save & Apply
3. LuCI ? Network ? Interfaces ? Wireguard ? Restart
4. LuCI ? Network ? Interfaces ? Edit: Wireguard ? Peers ? Edit: My Linux Fedora ?  
Generate Configuration
  - DNS Servers : 8.8.8.8

# Wireless

## Enable Wi-Fi

- [\[OpenWrt Wiki\] Enabling a Wi-Fi access point on OpenWrt](#)

LuCI Web ? Network ? Wireless ? Edit :

- General Setup:
  - Country Code: TW
  - ESSID: OpenWRT
- Wireless Security:
  - Encryption: WPA2-PSK or WPA3-SAE
  - Key: <your-wireless-password>
  - Save
- Save & Apply
- Enable

# Additional Software

## Network Monitor

### nlbwmon

```
opkg install luci-app-nlbwmon luci-i18n-nlbwmon-zh-tw
```

## Netdata

1. Add the custom feed: [fantastic-packages](#)
2. Run `opkg install luci-app-netdata`

### ?????

```
opkg install luci-i18n-base-zh-tw
```

## Themes

### Argon

1. Download \*.ipk: <https://github.com/jerrykuku/luci-theme-argon>
2. Install: `opkg install luci-theme-argon_2.3.2-r20250207_all.ipk`

## DDNS

### Duck DNS

- [Duck DNS](#)
- YT: [OpenWRT : Create your own VPN Server with OpenVPN - YouTube](#)

Install the packages

```
opkg install ddns-scripts luci-app-ddns
```

### NoIP

- [Free Dynamic DNS - Managed DNS - Managed Email - Domain Registration - No-IP](#)
- YT: [OpenWRT - VPN into your Home network using OpenVPN | Roadwarrior - YouTube](#)

Install the packages

```
opkg install ddns-scripts ddns-scripts-noip luci-app-ddns
```

# Quick Start

## DHCP

### Enable DHCP Server

LuCI Web ? Network ? Interfaces ? Edit: lan ? DHCP Server ? **General Setup**

- Ignore interface : ??????/? ?????
- Start : 100?IP ??? \*.100 ?????
- Limit : IP ??????

LuCI Web ? Network ? DHCP and DNS ? General

- Allocate IPs sequentially : ????????? IP?

### Disable IPv6 for LAN

LuCI Web ? Network ? Interfaces ? Edit: lan ? DHCP Server ? IPv6 Settings

- RA-Service : disabled
- DHCPv6-Service : disabled
- NDP-Proxy : disabled

## DHCP Options

???? IP ???? DNS ??? Gateway ?????? tag ?????????????????? DNS ? Gateway ???

LuCI Web ? Network ? Interfaces ? Edit: lan ? DHCP Server ? Advanced Settings ? DHCP-Options

- 6 ?? DNS :
- 3 ?? Gateway :
- ??tag ipphone ?????????? Gateway :
- ????? tag sensor ?????????? Gateway:

? IP??????? tag

LuCI Web ? Network ? DHCP and DNS ? Static Leases



# Custom Feeds Source

## fantastic-packages

- README: <https://github.com/fantastic-packages/packages/tree/gh-pages#readme>
- <https://fantastic-packages.github.io/packages/releases/>

/etc/opkg/customfeeds.conf :

- <major.minor version> : 24.10??????
- <package arch> : aarch64\_cortex-a53??????

```
# fantastic-packages Packages
# URL: https://github.com/fantastic-packages/packages/tree/gh-pages#readme
src/gz fantastic_packages_luci https://fantastic-
packages.github.io/packages/releases/<major.minor version>/packages/<package arch>/luci
src/gz fantastic_packages_packages https://fantastic-
packages.github.io/packages/releases/<major.minor version>/packages/<package arch>/packages
src/gz fantastic_packages_special https://fantastic-
packages.github.io/packages/releases/<major.minor version>/packages/<package arch>/special
```

## Add usign pub-keys to opkg

- Download `https://fantastic-packages.github.io/packages/releases/<major.minor version>/<KEY-ID>.pub`
- Put to `/etc/opkg/keys/<key-id>`, note filename must be lowercase

```
KEYID=<KEY-ID>
mkdir /etc/opkg/keys 2>/dev/null
curl -sSL -o /etc/opkg/keys/${KEYID,,} "https://fantastic-
packages.github.io/packages/releases/<major.minor version>/${KEYID}.pub"
```

# Ad Blocking

????????????????

## AdGuard Home

- [\[OpenWrt Wiki\] AdGuard Home](#)
- GitHub: <https://github.com/AdguardTeam/AdGuardHome>
- [AdGuard Home | Network-wide software for any OS: Windows, macOS, Linux](#)
- [???????openwrt??AdGuardHome?????\\_adguardhome????-CSDN??](#)

## Adblock-lean

- <https://github.com/lynxthecat/adblock-lean>
- Forum: [Adblock-lean: set up adblock using dnsmasq blocklist](#)