

OpenWRT 23.05.x with MESHdesk packages



Follow these instructions to include the MESHdesk package on devices capable of running OpenWRT version 23.05.x

Minimum Hardware Requirements

- The minimum hardware requirements are:
 - **8M Flash**
 - **64M RAM**
- Although the system can potentially support hardware with less resources, supporting them in 2022 is not practical any more.
- The hardware does not need a radio on it.
- Hardware without radios can be managed using **APdesk**.
- Mediatek and Atheros / Qualcomm SOC devices are supported.
- Other target systems are also supported but have not been thoroughly tested to date.
- If you are not sure if your hardware will work please visit the OpenWrt website and check. They have an ever growing list of supported hardware.
- Next we will look at the steps you need to take to compile OpenWrt with the MESHdesk package included.

Steps In Adding New Hardware

Explaining our modus operandi (MO)

- To introduce a new device we will:
 1. Create an initial development firmware image with all the required packages and files included but with central management **disabled**.
 2. We will then flash a device with this image and make the **device specific** tweaks in order for it to work as expected when managed centrally.
 3. We can then use these tweaked files to build a production image for the device that will include the device specific files.
- This approach allows us to:
 - Minimize the risk of bricking the device (A factory reset will bring it back to a normal running OpenWrt)
 - Allow fine tuning to the device without the need to create and flash a new firmware image each time we make a tweak.

Initial Development Firmware

These steps are **ONCE ONLY** when introducing new hardware.

1. Build OpenWRT with MESHdesk package (MESHdesk disabled).
2. Flash your device.
3. Prepare the **wan_network** file for specific device.
4. Prepare **meshdesk** config file for specific device.
5. Prepare **captive_config.json** file for specific device.



To structure the documentation better we have dedicated pages to discuss these files including their function and how to tweak them for your specific device.

This is a once only process so after you have made the tweaks to the files you can skip this step in future and simply use those device specific files when you need to build new firmware for the same hardware.

Production Firmware

1. Build OpenWRT firmware with the MESHdesk package that contains the device specific files for
 1. wan_network
 2. captive_config.json
 3. meshdesk
2. Flash your device with the production firmware.
 - So without further ado, lets get going with the initial development firmware.
 - We will take a Xiaomi 4A 100M Access Point as a sample unit.
 - You can use the hardware of your choice and simply apply the same principles.

Checkout the MESHdesk code

- Most of the packages one can include when building OpenWrt firmware are either part of the SDK or can be included using the package feeds that is also part of the SDK.
- You can also include packages that is not part of the SDK or part of the feeds.
- The MESHdesk package is such a package.
- The MESHdesk package is hosted on Github as part of a git repository.
- This git repository also includes the other items we will use with our firmware development (the Luci application and some additional files)
- Check out the **openwrt-meshdesk** package from the Github repository.

```
#Do this in the working directory e.g. cd 23.05.2 (on the same level as the openwrt directory)
git clone https://github.com/RADIUSdesk/openwrt-meshdesk.git openwrt-meshdesk
```

- The repository has three main components. Each one is located in a unique folder.
 1. **MESHdesk** - This is the MESHdesk package which will be included by the SDK.
 2. **files** - This is the override structure containing files to override during the build process.
 3. **luci-app-meshdesk** - This is the Luci application used to enable or disable central control of the device.

Copying the three components

- The **MESHdesk** folder needs to be copied under the **package** folder (openwrt/package).

```
#cd to the working directory
cp -R ./openwrt-meshdesk/MESHdesk ./openwrt/package
```

- The **files** folder needs to sit directly under the **openwrt** folder (root level).

```
#cd to the working directory
cp -R ./openwrt-meshdesk/files ./openwrt
```

- The **luci-app-meshdesk** folder needs to be copied under the **feeds/luci/applications** folder.

```
#cd to the working directory
cp -R ./openwrt-meshdesk/luci-app-meshdesk ./openwrt/feeds/luci/applications
```

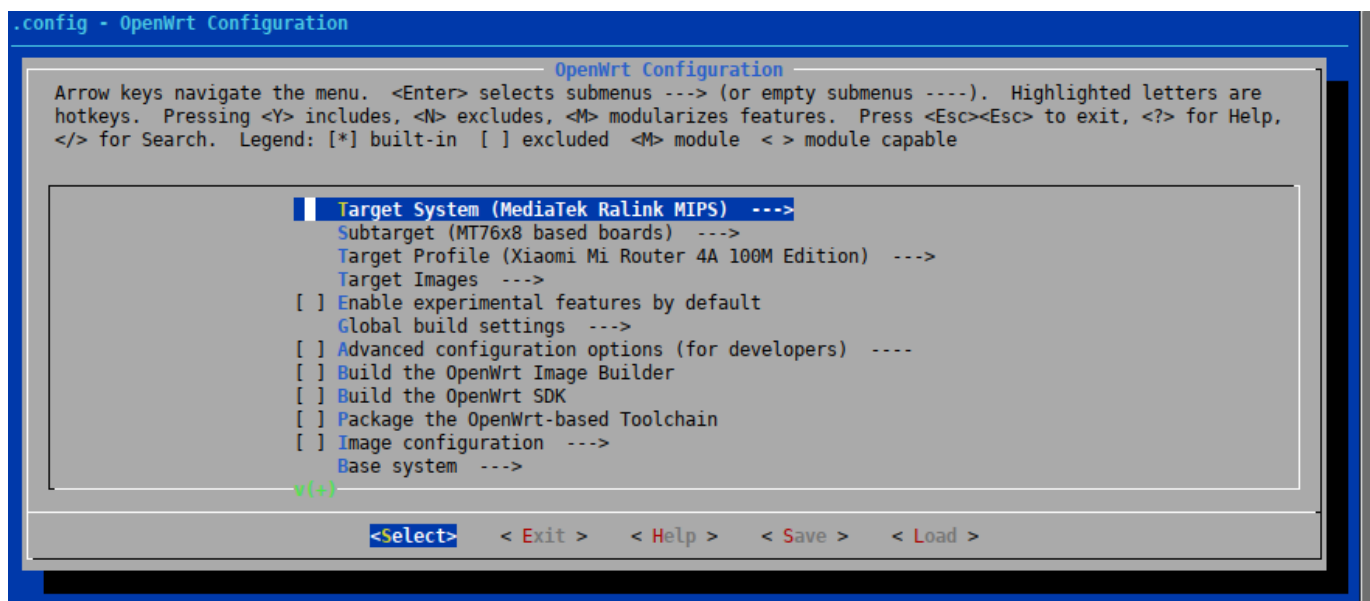
Updating the available packages

- Since we added a Luci application, we need to tell the SDK about it.
- After you copied the packages across issue the following command:

```
#cd to the working directory
cd ./openwrt
scripts/feeds update -i
#Install the package to make it visible
scripts/feeds install luci-app-meshdesk
```

- The result is that the MESHdesk Luci application will be listed as one of the available Luci applications.

Select Packages To Include With Firmware



- The OpenWrt SDK has a cursors interface (much like Midnight Commander for the old school readers 🤖)
- To start this cursors interface up you need to cd from the working directory to the **openwrt** directory.
- Then issue the following:

```
#cd to the working directory
cd openwrt
make menuconfig
```

- Here you will select the hardware architecture of the device you want to build the firmware for

and also its model.

- Also select the following packages when building the firmware.
- When selecting a package there are the options to build it as a module (M) or fully include it (*).
- **Make sure you select with the (*) option to fully include the package.**
- Package names in bold are required.
- The Mosquitto packages are for MQTT support.
- The Batman packages are for mesh support.

Location	Package	Comment
Base system	MESHdesk	
Kernel Modules → Network Support	kmod-batman-adv	Keep the default options
Kernel Modules → Netfilter Extensions	kmod-nft-bridge	Include if you want to block or speed limit users
Languages → Lua	lua-mosquitto	
Languages → Lua	libiwininfo-lua	
Languages → Lua	luasocket	
Libraries	libuci-lua	
Luci → Collections	luci	
Luci → Modules	luci-compat	Needs this modules for our package VERY IMPORTANT
Luci → Applications	luci-app-meshdesk	Luci App to enable and disable central management
Luci → Themes	luci-theme-material	Modern theme that is easy to customize
Luci → Libraries	luci-lib-httpclient	
Luci → Libraries	luci-lib-httpprotoutils	
Luci → Libraries	luci-lib-json	
Luci → Libraries	luci-lib-jsonc	
Network → Captive Portals	coova-chilli	Select OpenSSL as SSL Library . Also select Enable the JSON interface.. and ..Coova miniportal...
Network → File Transfer	curl	
Network → Firewall	iptables-nft	Very important for backward compatibly of iptables
Network → Routing and Redirection	relayd	
Network → WirelessAPD	wpad IEEE 802.1x Auth/Supplicant (built-in full)	Un-select wpad-basic also required for Hotspot 2.0
Network	batctl-full	Un-select batctl-default
Network	mosquitto-client-ssl	Note the CLIENT package
Network	iw-full	Required for Hotspot 2.0

- After you selected these packages you can save the configuration and issue **make** to build the firmware.
- The completed firmware will be located under the **openwrt/bin/target/<architecture>** folder.
- In our case we will use *openwrt/bin/targets/ramips/mt76x8/openwrt-ramips-mt76x8-xiaomi_mi-router-4a-100m-squashfs-sysupgrade.bin*

- The firmware you just built will be standard OpenWrt in effect and you can flash your hardware as with normal OpenWRT then access it on **192.168.1.1**.
 - Username and Password is **root** and **admin** for Luci and ssh.
- The next section will cover the files you have to attend to for the specific hardware tweaks.

Note on Warnings


- When you issue the **make** command these warnings might pop-up.
- They are harmless and can be ignored.

```
WARNING: Makefile 'package/utils/busybox/Makefile' has a dependency on
'libpam', which does not exist
WARNING: Makefile 'package/utils/busybox/Makefile' has a dependency on
'libpam', which does not exist
WARNING: Makefile 'package/utils/busybox/Makefile' has a build dependency on
'libpam', which does not exist
WARNING: Makefile 'package/network/services/lldpd/Makefile' has a dependency
on 'libnetsnp', which does not exist
WARNING: Makefile 'package/utils/policycoreutils/Makefile' has a dependency
on 'libpam', which does not exist
WARNING: Makefile 'package/utils/policycoreutils/Makefile' has a dependency
on 'libpam', which does not exist
WARNING: Makefile 'package/utils/policycoreutils/Makefile' has a build
dependency on 'libpam', which does not exist
make[2]: Entering directory
'/home/system/Documents/fw_sdk/23.05.2/openwrt/scripts/config'
make[2]: 'conf' is up to date.
make[2]: Leaving directory
'/home/system/Documents/fw_sdk/23.05.2/openwrt/scripts/config'
make[1] world
make[2] target/compile
make[3] -C target/linux compile
```

Files to tweak

Use ssh to gain access to the device in order to tweak these files.

- [wan_network](#) file
- [meshdesk](#) file
- [captive_portal.json](#) file
- Once the tweaks are completed we can test everything out.
- Log onto your device using Luci (<http://192.168.1.1>)
- The following image shows how to point the device to the controller using the GUI.



4A-100-GW

Status

System

Services

MESHdesk Settings

Network

Logout

MESHdesk

Controller Details

Central Management

Disabled

Enabled

Protocol

HTTPS

FQDN

cloud.radiusdesk.com

Supply Dummy Value If Not Using DNS System

IP Address

164.160.89.129

Optional If FQDN Fails / Not Used

- Point the device to your controller and reboot it.
- If all goes well it will show up in Unknown Nodes.
- If it is a new hardware type add it to the controller as described here: Hardwares (Again Once Off)

The Production Build

- If everything on the device work as intended you can use those tweaked files to build a final version of the firmware for the specific hardware.
- Copy the files to a temporary folder on the machine where you are building the firmware.
- Use the following as a lookup for the location inside the SDK where the tweaked files need to go.

On Device	On SDK
/etc/MESHdesk/configs/wan_network	openwrt/package/MESHdesk/files/MESHdesk/configs/
/etc/config/meshdesk	openwrt/package/MESHdesk/files/MESHdesk/
/etc/MESHdesk/configs/captive_config.json	openwrt/package/MESHdesk/files/MESHdesk/configs/
/etc/MESHdesk/reporting/report_to_server.lua	openwrt/package/MESHdesk/files/MESHdesk/reporting

- This brings us to the end of the page on how to build MESHdesk firmware for specific hardware.

From:

<http://www.radiusdesk.com/wiki/> - **RADIUSdesk**

Permanent link:

<http://www.radiusdesk.com/wiki/technical/openwrt-meshdesk>

Last update:

2023/12/27 19:20

