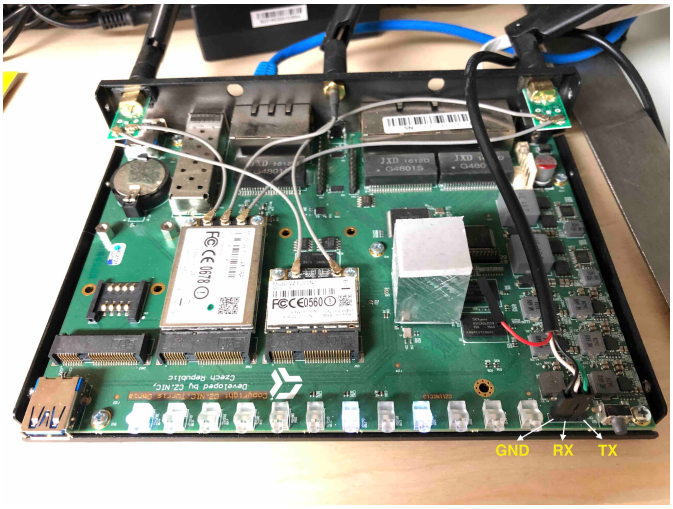
Turris Omnia RTROM01-2G & RTROM01-CE: Flashing Instruction

Hardware Information

Wi-Fi Extender Reference Targets

Serial Port Access

https://doc.turris.cz/doc/en/troubleshooting/serial_link#turris_omnia



Caution: Do **not** connect Vcc!

NOTE: Wouter Cloetens shared links and information on taking serial connection and flashing OpenWRT image

Setting up Bootloader:

Setting default U-boot environment

In serial console, press "Enter" key to get into u-boot prompt

Run following u-boot commands to set default U-boot environment

```
env default -a
saveenv
reset
```

Flashing

The Omnia ships with TurrisOS preinstalled. This is an OpenWrt fork with the older Linux 4.4 kernel. It uses btrfs to manage the eMMC flash.

We do not use it in this mode. The RDK-B port is closer to the mainline OpenWrt port, which uses Linux 4.14, and a traditional partition table.

Reflashing procedure from TurrisOS:

- upgrade through a "medkit" image through the use of a USB stick and pressing the reset button.
- https://openwrt.org/toh/turris_cz.nic/turris_cz.nic_omnia is not entirely accurate, and cannot complete without access to the serial port, due to an error in the factory initialisation of the U-boot environment. Updated procedure:
    - Remove the casing. Connect a TTL serial interface, see "Serial Port Access" above.
    - Format a USB flash drive as **ext2**.
    - Download openwrt-18.06.2-mvebu-cortexa9-turris-omnia-sysupgrade.img.gz and omnia-medkit-openwrt-18.06.2-mvebu-cortexa9-turris-omnia-initramfs.tar.gz from http://downloads.openwrt.org/releases/18.06.2/targets/mvebu/cortexa9/ and copy both files to the root of the flash drive.

- Disconnect other USB devices from the Omnia and connect the flash drive to either USB port. (I have only verified this with the USB port at the front).
- Hold down the reset button (backside, bottom centre) and plug in the power cord. Wait until the fourth LED lights up (green), then release (before the 5th LED lights up). Please click here to see more detail on rescue modes.
- Wait approximately 2 minutes for the Turris Omnia to flash itself with the temporary image, during which LEDs will change multiple times.
- Either use the serial console, or connect a computer to a LAN port (LAN0 to LAN4) of the Turris Omnia with a DHCP client.
  From a computer:

```
ssh root@192.168.1.1
```

  Then, from the ssh session or a serial console:

```
mount /dev/sda1 /mnt
sysupgrade /mnt/openwrt-18.06.2-mvebu-cortexa9-turris-omnia-sysupgrade.img.gz
```

- Wait another minute for the final OpenWrt image to be flashed. The Turris Omnia will reboot itself and you can remove the flash drive.
- After device reboot. You are now running stock OpenWrt 18.06.2.

- [x] Wouter Cloetens to provide an updated medkit image that fixes the U-boot environment, removing the need to open the casing and connect a serial port.

The eMMC disk is now regularly partitioned.

```
root@OpenWrt:~# fdisk /dev/mmcblk0

Welcome to fdisk (util-linux 2.32).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): p
Disk /dev/mmcblk0: 7.3 GiB, 7818182656 bytes, 15269888 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x15344058

Device         Boot Start    End Sectors   Size Id Type
/dev/mmcblk0p1 *     2048  35327   33280  16.3M  c W95 FAT32 (LBA)
/dev/mmcblk0p2      36864 561663  524800 256.3M 83 Linux
```

If 'fdisk' command is not preset

root@OpenWrt:~# opkg update

root@OpenWrt:~# opkg install fdisk

The devicetree database and the Linux kernel are in the FAT32 partition:

```
root@OpenWrt:~# mount /dev/mmcblk0p1 /mnt
root@OpenWrt:~# ls -l /mnt
-rwxr-xr-x    1 root     root         17353 Aug 17  2018 armada-385-turris-omnia.dtb
-rwxr-xr-x    1 root     root       2445072 Aug 17  2018 zImage
```

It is strongly recommend to keep the OpenWrt partition and kernel. If boot fails, it is easy to change the U-boot environment and boot back into OpenWrt (Fail safe OS).

Flashing RDKB image (A Yocto Project based Distro)

Follow instruction from wiki page(https://wiki.rdkcentral.com/display/RDK/Wifi-Extender+Yocto+Build+Instructions) to make yocto's RDK-B image from yocto workspace in your PC.

Boot into openwrt based linux system in turris omnia board. Its downlink interface will have 192.168.1.1 IP address and have it connected to your PC.

Create a new partitions for RDK-B.
Create one primary partition for zImage. Create another extended partition for holding logical partitions for rootfs and /nvram.

```
root@OpenWrt:/# fdisk /dev/mmcblk0

Welcome to fdisk (util-linux 2.28.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): p
Disk /dev/mmcblk0: 7.3 GiB, 7818182656 bytes, 15269888 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x15488508

Device         Boot Start    End Sectors   Size Id Type
/dev/mmcblk0p1 *     2048  35327   33280  16.3M  c W95 FAT32 (LBA)
/dev/mmcblk0p2      36864 561663  524800 256.3M 83 Linux


Command (m for help): n
Partition type
   p   primary (2 primary, 0 extended, 2 free)
   e   extended (container for logical partitions)
Select (default p):
Using default response p.
Partition number (3,4, default 3):
First sector (35328-15269887, default 563200):
Last sector, +sectors or +size{K,M,G,T,P} (563200-15269887, default 15269887): 596479

Created a new partition 3 of type 'Linux' and of size 16.3 MiB.

Command (m for help): n
Partition type
p primary (3 primary, 0 extended, 1 free)
e extended (container for logical partitions)
Select (default e): e

Selected partition 4
First sector (35328-15269887, default 598016):
Last sector, +sectors or +size{K,M,G,T,P} (598016-15269887, default 15269887):

Created a new partition 4 of type 'Extended' and of size 7 GiB.

Command (m for help): n
All primary partitions are in use.
Adding logical partition 5
First sector (600064-15269887, default 600064):
Last sector, +sectors or +size{K,M,G,T,P} (600064-15269887, default 15269887): +512M

Created a new partition 5 of type 'Linux' and of size 512 MiB.

Command (m for help): n
All primary partitions are in use.
Adding logical partition 6
First sector (1650688-15269887, default 1650688):
Last sector, +sectors or +size{K,M,G,T,P} (1650688-15269887, default 15269887): +64M

Created a new partition 6 of type 'Linux' and of size 64 MiB.

All primary partitions are in use.

Command (m for help): n
All primary partitions are in use.
Adding logical partition 7
First sector (1783808-15269887, default 1783808):
```

```
Last sector, +sectors or +size{K,M,G,T,P} (1783808-15269887, default 15269887): +512M

Created a new partition 7 of type 'Linux' and of size 512 MiB.

Command (m for help): p
Disk /dev/mmcblk0: 7.3 GiB, 7818182656 bytes, 15269888 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x15488508

Device      Boot Start End Sectors Size Id Type
/dev/mmcblk0p1 * 2048 35327 33280 16.3M c W95 FAT32 (LBA)
/dev/mmcblk0p2   36864 561663 524800 256.3M 83 Linux
/dev/mmcblk0p3   563200 596479 33280 16.3M 83 Linux
/dev/mmcblk0p4   598016 15269887 14671872 7G 5 Extended
/dev/mmcblk0p5   600064 1648639 1048576 512M 83 Linux
/dev/mmcblk0p6   1650688 1781759 131072 64M 83 Linux
/dev/mmcblk0p7   1783808 2832383 1048576 512M 83 Linux

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Re-reading the partition table failed.: Resource busy

The kernel still uses the old table. The new table will be used at the next reboot or after you run partprobe
(8) or kpartx(8).


root@OpenWrt:/# reboot
```

After reboot, format /dev/mmcblk0p3, /dev/mmcblk0p5, /dev/mmcblk0p6 and /dev/mmcblk0p7 as ext2 partitions.

```
root@OpenWrt:/# mkfs.ext2  /dev/mmcblk0p3
mke2fs 1.44.1 (24-Mar-2018)
/dev/mmcblk0p3 contains a ext2 file system
        last mounted on /mnt on Fri Aug  9 13:56:55 2019
Proceed anyway? (y,N) y
Discarding device blocks: done
Creating filesystem with 16640 1k blocks and 4176 inodes
Filesystem UUID: 1b53ea80-b120-4072-a87e-9e68092ed311
Superblock backups stored on blocks:
        8193

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

root@OpenWrt:/# mkfs.ext2  /dev/mmcblk0p5
mke2fs 1.44.1 (24-Mar-2018)
/dev/mmcblk0p5 contains a ext2 file system
        last mounted on / on Fri Aug  9 13:57:53 2019
Proceed anyway? (y,N) y
Discarding device blocks: done
Creating filesystem with 262145 1k blocks and 65792 inodes
Filesystem UUID: 14219ea9-3584-4d61-adb4-8370dd73b5cc
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729, 204801, 221185

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

root@OpenWrt:/# mkfs.ext2  /dev/mmcblk0p6
mke2fs 1.44.1 (24-Mar-2018)
/dev/mmcblk0p6 contains a ext2 file system
        last mounted on /nvram on Fri Aug  9 13:59:29 2019
Proceed anyway? (y,N) y
Discarding device blocks: done
Creating filesystem with 65536 1k blocks and 16384 inodes
Filesystem UUID: df6ac518-0a1a-432a-a921-d6958307340a
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

root@OpenWrt:/# mkfs.ext2  /dev/mmcblk0p7
mke2fs 1.44.1 (24-Mar-2018)
Discarding device blocks: done
Creating filesystem with 131072 4k blocks and 32768 inodes
Filesystem UUID: 543d4394-4a94-457a-b487-9c3595d3131e
Superblock backups stored on blocks:
        32768, 98304
Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

Copy zImage, dtb and a rootfs tar file from yocto workspace to turris omnia(whose IP is 192.168.1.1).
(in your PC)

```
$ scp <yocto_workspace>build-turris/tmp/deploy/images/turris/zImage root@192.168.1.1:/tmp
$ scp <yocto_workspace>build-turris/tmp/deploy/images/turris/{"zImage-",""}armada-385-turris-omnia.dtb root@192.
168.1.1:/tmp/armada-385-turris-omnia.dtb
$ scp <yocto_workspace>build-turris/tmp/deploy/images/turris/rdkb-generic-broadband-image_default_<image-
timestamp>.rootfs.tar.gz root@192.168.1.1:/tmp/
```

(in turris omnia)

```
root@OpenWrt:~# mount /dev/mmcblk0p3 /mnt
root@OpenWrt:~# mv /tmp/zImage /mnt
root@OpenWrt:~# mv /tmp/armada-385-turris-omnia.dtb /mnt
root@OpenWrt:~# umount /mnt
root@OpenWrt:~# mount /dev/mmcblk0p5 /mnt
root@OpenWrt:~# tar -xzf /tmp/rdkb-generic-broadband-image_default_<image-timestamp>.rootfs.tar.gz -C /mnt
root@OpenWrt:~# umount /mnt
root@OpenWrt:~# reboot
```

Press enter while turris omnia reboots to get u-boot prompt. Then type following u-boot commands for booting yocto based RDKB image instead of openwrt image.

```
=> env set yocto_bootargs earlyprintk console=ttyS0,115200 root=/dev/mmcblk0p5 rootfstype=ext2 rw rootwait
=> env set yocto_mmcload setenv bootargs \"\$yocto_bootargs cfg80211.freg=\$regdomain\"\; ext2load mmc 0:3
0x01000000 zImage\; ext2load mmc 0:3 0x02000000 armada-385-turris-omnia.dtb
=> env set mmcboot run yocto_mmcload \|\| run openwrt_mmcload \|\| run factory_mmcload\; bootz 0x01000000 -
0x02000000
=> saveenv
=> reset
```

Now, Turris Omnia will boot with Yocto based RDKB image.

RDK Firmware(Image) upgrade:

Approach 1:

RDK firmware upgrade with XConf server: Firmware upgrade through XCONF server - Turris-Omnia - User Manual - 2020 - M6

Approach 2(Quick):

Copy zImage, dtb file and rootfs files(**not *dbg* rootfs file**) from PC or VM to /tmp/ directory of Turris Omnia which is currently running RDK Image.

For example

```
scp zImage--4.14.22-r0-turris-20200720105910.bin root@<TurrisOmnia-IP>:/tmp/
scp armada-385-turris-omnia.dtb root@<TurrisOmnia-IP>:/tmp/
scp rdkb-generic-broadband-image_default_20200720105910.rootfs.tar.gz root@<TurrisOmnia-IP>:/tmp/
```

In Turris Omnia, execute */lib/rdk/TurrisFwUpgrade.sh* to flash new RDK image present in /tmp folder

```
sh /lib/rdk/TurrisFwUpgrade.sh
```

Approach 3:

Go back to OpenWrt OS (please refer **Fallback to OpenWrt OS** section below).

Have Ethernet connection from your PC to LAN port of Turris Omnia.

Keep rootfs(*rdkb-generic-broadband-image_default_*.rootfs.tar.gz*), zImage(*zImage--4.14.22-r0-turris-*.bin*) and dtb file in home directory of your PC.

In OpenWrt OS, create a script(*yocto-fw-upgrade.sh)* with following commands with updates as it needs.

```
MYPC_IP=192.168.1.83
USER_NAME=manigandan
scp $USER_NAME@$MYPC_IP:*$1* /tmp/
mount /dev/mmcblk0p3 /mnt
mv /tmp/zImage-* /mnt/zImage
mv /tmp/armada-385-turris-omnia.dtb /mnt/
umount /mnt
mount /dev/mmcblk0p5 /mnt
rm -rf /mnt/*
tar -xzf /tmp/rdkb-* -C /mnt/
umount /mnt
reboot
```

Run the script **from inside OpenWrt OS**.

```
root@OpenWrt:/# ./yocto-fw-upgrade.sh <image-timestamp>
```

Press enter while turris omnia reboots to get into u-boot prompt. Then type following u-boot commands to get back to Yocto based RDKB image.

```
=> env set yocto_bootargs earlyprintk console=ttyS0,115200 root=/dev/mmcblk0p5 rootfstype=ext2 rw rootwait
=> env set yocto_mmcload setenv bootargs \"\$yocto_bootargs cfg80211.freg=\$regdomain\"\; ext2load mmc 0:3
0x01000000 zImage\; ext2load mmc 0:3 0x02000000 armada-385-turris-omnia.dtb
=> env set mmcboot run yocto_mmcload \|\| run openwrt_mmcload \|\| run factory_mmcload\; bootz 0x01000000 -
0x02000000
=> saveenv
=> reset
```

Turris Omnia will now run upgraded version of Yocto based RDK image.

Fallback to OpenWrt OS(Failsafe):

To fallback to openwrt OS, enter following u-boot commands.

```
=> env set mmcboot run openwrt_mmcload \|\| run factory_mmcload\; bootz 0x01000000 - 0x02000000
=> saveenv
=> reset
```