Rootfs notes

Rootfs in FLASH (UBIFS)

The easiest way to generate UBIFS image is by setting BR2_TARGET_ROOTFS_UBIFS to y in the buildroot config. Buildroot will also need the following file system parameters, that can be queried with mtdinfo tool:

- BR2_TARGET_ROOTFS_UBIFS_LEBSIZE: logical erase block (LEB) size (130944 for this device)
- BR2_TARGET_ROOTFS_UBIFS_MINIOSIZE: minimum I/O unit size (1 for this device)
- BR2_TARGET_ROOTFS_UBIFS_MAXLEBCNT: maximum logical eraseblock count (this device has 768, 764 are usable, as reported by ubiattach).

mtdinfo -a -u

```
Count of MTD devices:           4
Present MTD devices:            mtd0, mtd1, mtd2, mtd3
Sysfs interface supported:      yes

mtd0
Name:                           data
Type:                           nor
Eraseblock size:                131072 bytes, 128.0 KiB
Amount of eraseblocks:          768 (100663296 bytes, 96.0 MiB)
Minimum input/output unit size: 1 byte
Sub-page size:                  1 byte
Character device major/minor:   90:0
Bad blocks are allowed:         false
Device is writable:             true
Default UBI VID header offset:  64
Default UBI data offset:        128
Default UBI LEB size:           130944 bytes, 127.9 KiB
Maximum UBI volumes count:      128
```

To format FLASH partition and put UBIFS image into it the following command sequence may be used:

- run ubiformat:

ubiformat /dev/mtd0

```
ubiformat: mtd0 (nor), size 100663296 bytes (96.0 MiB), 768 eraseblocks of
131072 bytes (128.0 KiB), min. I/O size 1 bytes
libscan: scanning eraseblock 767 -- 100 % complete
ubiformat: 768 eraseblocks have valid erase counter, mean value is 2
ubiformat: formatting eraseblock 767 -- 100 % complete
```

- attach formatted device to UBI (this will create /dev/ubi0 device):

ubiattach -p /dev/mtd0

```
 UBI device number 0, total 768 LEBs (100564992 bytes, 95.9 MiB), available
764 LEBs (100041216 bytes, 95.4 MiB), LEB size 130944 bytes (127.9 KiB)
```

- create UBI volumes on formatted device (this will create /dev/ubi0_0 device, AKA ubi0:rootfs):

ubimkvol /dev/ubi0 -m -N rootfs

```
 Set volume size to 100041216
 Volume ID 0, size 764 LEBs (100041216 bytes, 95.4 MiB), LEB size 130944
bytes (127.9 KiB), dynamic, name "rootfs", alignment 1
```

- put UBI rootfs image so that xtensa linux could access it (in this example kernel is booted with NFS root and UBI image goes to /mnt/rootfs.ubifs)
- run ubiupdatevol to copy file system image to the specific UBI volume:

ubiupdatevol /dev/ubi0_0 /mnt/rootfs.ubifs

- now the volume may be mounted:

mount -t ubifs /dev/ubi0_0 /mnt

- to use that volume as rootfs add the following parameters to the kernel command line (note that 0 in the 'ubi.mtd=0' is the number of mtd partition to bind to ubi0, and 'rootfs' in the ubi0:rootfs below is the volume name):

```
ubi.mtd=0 root=ubi0:rootfs rootfstype=ubifs
```

Further reading:

- http://www.linux-mtd.infradead.org/doc/ubifs.html
- http://www.linux-mtd.infradead.org/faq/ubifs.html
- https://www.kernel.org/doc/Documentation/filesystems/ubifs.txt