



How to Create/Modify an RAM disk Image

Glomation GESBC-93XX series single board computers use RAM disk as the root file system in it's default configuration. Most likely one would want to create/change the RAM disk to have user applications installed. This Document describes how to mount and edit files in a RAM disk image, unmount the image and recompress it.

System Requirement

In order to mount a RAM disk image, your development system must have MTD support compiled in, and have the Loopback Interface available. Most Linux distribution have these configured in the kernel.

To check the availability of MTD and loopback interface, issue the following command on the development system

```
# cat /proc/mtd
or
# ls -l /dev/mtd*
```

If any information is shown with `/proc/mtd`, and if you see any files named `mtdblock`, `mtd`, etc that means the system has MTD support.

If not please rebuild the Linux kernel with MTD support. The follow instruction shows the basic steps to do that.

Go to the kernel source tree directory on your development system and issue...

```
# make menuconfig
```

The following options must be enabled to use MTD.

```
Memory Technology Devices (MTD) --->
  <*> Memory Technology Device (MTD) support
    [*]   Debugging
        (0)   Debugging verbosity (0 = quiet, 3 = noisy)
  [*]   MTD partitioning support
  <*> Direct char device access to MTD devices
  <*> Caching block device access to MTD devices
RAM/ROM/Flash chip drivers --->
  <*> Detect flash chips by Common Flash Interface (CFI)
probe
  <*> Support for Intel/Sharp flash chips
  <*> Support for AMD/Fujitsu flash chips
Self-contained MTD device drivers --->
  <*> MTD emulation using block device
```

As well as....

```
Block devices --->
  <*> RAM disk support
  (32768) Default RAM disk size (kbytes)
  [*] Initial RAM disk (initrd) support
```

After the kernel has been reconfigured, re-build the kernel, install the new kernel and reboot your development system.

Once you have MTD and RAM disk support on your development system, you are ready to create/modify the RAM disk image.

Create New RAM Disk Image

The Ext2 file system can be created and mounted onto an embedded device using the concept of RAM disk.

```
mke2fs -vm0 /dev/ram 16384
mount -t ext2 /dev/ram /mnt
cd /mnt
cp /bin, /sbin, /etc, /dev ... files in mnt
cd ../
umount /mnt
dd if=/dev/ram bs=1k count=16384 of=ramdisk
gzip -9 ramdisk
```

mke2fs is the utility used to create an ext2 file system -- creating the super block, inodes, inode table, and etc -- on any device.

In the above usage, /dev/ram is the device on which an ext2 file system of 16384 blocks is built. Then the device (/dev/ram) is mounted on a temporary directory named /mnt and all the necessary files are copied. Once these files are copied, the file system is unmounted and the contents of the device (/dev/ram) is dumped into a file (ramdisk), which is the required RAM disk (the Ext2 file system). The RAM disk file is then compressed and ready to be loaded by Redboot.

The above sequence creates a RAM disk of 16 MB and fills it with the necessary file utilities.

Some of the important directories to include in RAM disk are:

/bin -- Holds most of the binaries like init, busybox, shell, file management utilities, and etc.

/dev -- Contains all the device nodes used in the device

/etc -- Contains all the configuration files for the system

/lib -- Contains all the necessary libraries like libc, libdl, and so on

Modify Existing RAM Disk Image

1. First uncompress the ramdisk.gz file.

```
# gunzip ramdisk.gz
```

2. Make a directory to mount the ramdisk image.

```
# mkdir ramdisk_dir
```

3. Mount the ramdisk image as a loop back device

```
# mount -o loop ramdisk ramdisk_dir
```

You can then issue the command 'df' to see the new RAM disk partition on you development system.

4. Edit Files

You can now edit files inside the RAM disk. Depending on how the original RAM disk is created, you may or may not have enough space on the RAM disk to install your application. Refer to **Create New RAM Disk Image** section on how to create a new RAM disk with more storage space.

5. Unmount the ramdisk and compress it.

```
# umount ramdisk_dir
```

```
# gzip -9 ramdisk
```

You will then have a new compressed root file system that you can now load with RedBoot