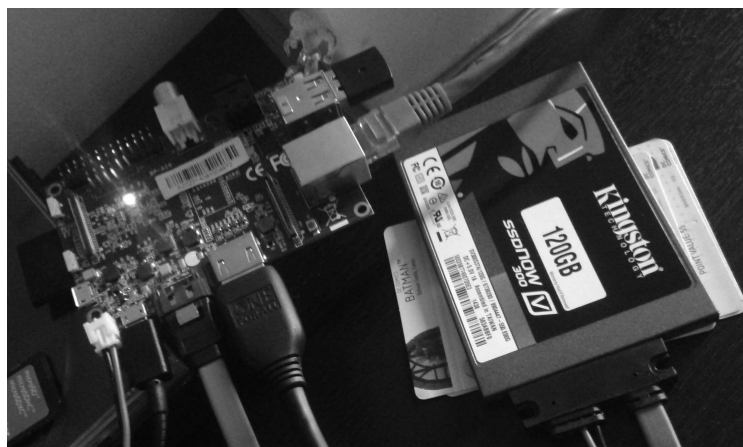


## Banana PI – Set up Lubuntu / Raspbian on a SATA SSD/HD



After a year of using a class10 SD card, I started to get some i/o errors... So decided to start taking advantage of one of the nicest features banana Pi has: SATA port + SATA power supply.

I had no idea of where to begin... So Googled it. Found 3 sites that more or less described how to proceed, but both where for Raspbian only and dead ended me along the way. Hope this guide helps other bananians using Lubuntu to achieve your goal!

I recommend using an SSD if possible, given that it consumes a lot less power and its heat generation is close to none.

Set up Lubuntu / Raspbian on an SD card as described in many other sites ([http://www.lemaker.org/resources/9-39/banana\\_pi\\_quick\\_start\\_guide.html](http://www.lemaker.org/resources/9-39/banana_pi_quick_start_guide.html)). Connect your disk to the Banana Pi's SATA socket, but don't power up the disk.

Open a terminal and run this command to display new messages in one of Linux's log files:

```
tail  
-f /var/log/messages
```

Power up the SATA disk, and look for the new drive in the messages in the terminal. If the drive is auto-mounted when you connect it, unmount it.

### **Partition the disk**

Use this command to start the fdisk partitioning tool:

```
sudo fdisk /dev/sda
```

Type 'p' to list all the partitions on the disk:

```

Command (m for help): p

Disk /dev/sda: 100.0 GB,
100030242816 bytes
255 heads, 63 sectors/track,
12161 cylinders, total
195371568 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
Disk identifier: 0x99b699b6

```

```

Device Boot Start End Blocks
Id System
/dev/sda1 63 96389 48163+ de
Dell Utility
/dev/sda2 * 96390 102494699
51199155 7 HPFS/NTFS/exFAT
/dev/sda3 102494700 195366464
46435882+ 5 Extended
/dev/sda5 102494763 183639014
40572126 83 Linux
/dev/sda6 183639078 195366464
5863693+ 82 Linux swap /
Solaris

```

If there is any partition there, we will have to remove them!

## Remove existing partitions

I'm going to delete all of these partitions, and replace them with one partition that fills the entire disk. Press 'd' to delete a partition, and then enter the number of the partition to be deleted:

```
Command (m for help): d
Partition number (1-6): 1
```

```
Command (m for help): d
Partition number (1-6): 2
```

```
Command (m for help): d
Partition number (1-6): 3
```

## Write the changes to the disk

None of the changes made so far have been written to the disk. We can do this by typing 'w' at the prompt:

```
Command (m for help): w
The partition table has been
altered!
```

```
Calling ioctl() to re-read
partition table.
Syncing disks.
```

Fdisk will automatically quit. Restart fdisk so that you can create a partition, and type 'n' to create a new partition. When prompted for the type, press 'p' for primary, and enter '1' for the partition number. Just press return to use the default values when prompted to enter the first and last sectors.

```
sudo fdisk /dev/sda

Command (m for help): n
Partition type:
p primary (0 primary, 0
extended, 4 free)
e extended
Select (default p): p
Partition number (1-4,
default 1): 1
First sector (2048-195371567,
default 2048):
Using default value 2048
Last sector, +sectors or
+size{K,M,G} (2048-195371567,
default 195371567):
Using default value
195371567
```

Type 'w' to write the changes to the disk again.

### **Format the partition**

Next, format the new partition with the ext4 filesystem:

```

    sudo mke2fs -t ext4 -L
rootfs /dev/sda1

mke2fs 1.42.5 (29-Jul-2012)
Filesystem label=rootfs
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe
width=0 blocks
6111232 inodes, 24421190
blocks
1221059 blocks (5.00%)
reserved for the super user
First data block=0
Maximum filesystem blocks=0
746 block groups
32768 blocks per group, 32768
fragments per group
8192 inodes per group
Superblock backups stored on
blocks:
32768, 98304, 163840, 229376,
294912, 819200, 884736,
1605632, 2654208,
4096000, 7962624, 11239424,
20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Creating journal (32768
blocks): done
Writing superblocks and
filesystem accounting
information: done

```

## Copy the data

Mount the partition, and use rsync to copy the contents of the SD card's root directory to the SATA disk.

```
sudo mount /dev/sda1 /media
sudo rsync -ax / /media
```

So far, we copied our OS to the SATA disc... and now? Would it boot directly from there?

No...Currently it's impossible to boot from a SATA drive without a SD card. You will still need a small SD card for the bootloader. However, it is possible to adjust the bootscript (which is similar to a bootloader) on the SD card so that it points directly to your SATA drive. This way your system will boot the OS right from the SATA drive.

We will have to change the kernel parameters in order to "boot" from the SSD drive

## How to do it in LUBUNTU

```
mkdir /tmp/boot
mount /dev/mmcblk0p1 /tmp/boot
nano /tmp/boot/uEnv.txt
```

Change the last line to:

```
.. root=/dev/sda1 ...
```

Sync to disk and reboot to your hard drive

```
sync
reboot
```

You can test the result with:

```
df -h
```

=====

## How to do it in RASPBIAN

Adjust boot parameters to use the new disk

The last step is to edit a file on the SD card's boot partition which contains boot parameters. The file is `/boot/uEnv.txt` (on the Raspberry Pi the equivalent file is `/boot/cmdline.txt`). It contains a variable called 'root', and it needs to be adjusted to point to the new partition, `/dev/sda1`:

```
sudo nano /boot/uEnv.txt
```

The file should look like this once you've made the changes:

```
bootargs=console=ttyS0,1152
00 console=tty0
disp.screen0_output_mode=EDID
:1280x720p50
hdmi.audio=EDID:0
console=tty1 root=/dev/sda1
rootfstype=ext4
elevator=deadline rootwait
aload_script=fatload mmc 0
0x43000000
bananapi/script.bin;
aload_kernel=fatload mmc 0
0x48000000 bananapi/uImage;
bootm 0x48000000;
uenvcmd=run aload_script
aload_kernel
```

Finally, restart your server. When it starts up again, it should use the hard disk as the root file system.



