

Boot EndeavourOS on Raspberry Pi 4b from USB device

First I need to give credit where credit is due. [XSystem252 has a guide](#) for Rpi 4b USB boot with BTRFS format, full encryption with SSH unlock in the case of a headless SBC. His guide was extremely helpful for the making of this guide. If you are interested in btrfs and/or encryption, give his site a visit.

This guide will Install EndeavourOS and set up the Raspberry Pi 4b for USB boot with ext4 format and no encryption. The important thing for this procedure is the USB 3 device. Be sure the USB 3 device supports booting from the device, and TRIM capabilities. A lot of the cheaper external USB 3 enclosures that require you to install a 2.5 inch SSD don't have either capability. When looking for USB 3 enclosures to house your SSD, look for ones that have UASP support. The UASP standard is pretty loose, especially with TRIM, but it's better than nothing. USB 3 thumb drives usually work, but may not be the fastest flash memory. It is recommended to experiment with a USB 3 thumb drive, then once you are comfortable with the procedure, experiment with external SSDs.

The first thing that needs to be done, is update the EEPROM in the Raspberry Pi 4b. This only needs to be done once, then you are good to go. Raspberry Pi OS will be utilized to do this EEPROM upgrade. It is the official OS for RPi4, and the EEPROM update is now official and in their stable repository. This has to be the safest way to flash the EEPROM, but you still do this at your own risk.

Before the EEPROM update, the device firmware would search the micro SD slot for a valid u-boot file. The new firmware Ver '000138a1' adds the capability to also search the USB ports for a valid u-boot. Ver '000138a1' firmware will search the micro SD slot first, if no valid u-boot is found, it will then search the USB ports for a valid u-boot. After the update, booting from the micro SD still works like before. You are not locked into only booting from USB.

Install Raspberry Pi OS Lite on a 8GB or bigger micro SD card.

In a Operational Linux computer, in a internet browser go to:

<https://www.raspberrypi.org/software/operating-systems/#raspberry-pi-os-32-bit>

and click on the 'Raspberry Pi OS Lite' download button.

After it is downloaded, using a file manager, go to the Downloads directory. Then on the downloaded file, right click, and use an Archive Manager to extract the image file.

Use gnome-disks or your favorite ISO burner to install the image on a 8 GB or larger micro SD card. When finished, insert the micro SD card in the Raspberry Pi 4b and power it up.

When the RPi4 is finished booting and doing a little house cleaning, log in as username ' pi ' and password ' raspberry '. Update the image.

```
$ sudo apt update
$ sudo apt full-upgrade
$ sudo systemctl reboot
```

After reboot, as user ' pi ' edit the following config file using nano or vi

```
$ sudo nano /etc/default/rpi-eeprom-update  
change  
From: FIRMWARE_RELEASE_STATUS="critical"  
To: FIRMWARE_RELEASE_STATUS="stable"  
close file.
```

Update the firmware.

```
$ sudo rpi-eeprom-update -d -a
```

The EEPROM is now updated. Raspberry Pi OS and the micro SD card it is on are no longer needed.

Back in a working Linux Computer, on a 16 GB or larger uSD card, install EndeavourOS and a DE. You can re-use the micro SD card that Raspberry Pi OS was installed on if you want.

Go to

<https://github.com/endeavouros-arm/image-install>

and follow the instructions to install an Archlinux Arm base install on a Raspberry Pi 4b.

After the Archlinux 64 bit base install is complete, insert the micro SD card into the Raspberry Pi 4b and power up the Rpi4.

Go to

<https://github.com/endeavouros-arm/manuals/blob/master/EOS-desktop-instructions.pdf>

Click on 'Download' and the instructions will be displayed in a browser image viewer where the instructions can then be printed or saved as .pdf

You can skip to page 5, follow the instructions to install EndeavourOS with your choice of DE.

You should now be in your Raspberry Pi 4b with a working EndeavourOS with your favorite Desktop Environment on a micro SD card. Now we will start to prepare the OS to be transferred to a USB 3 device. First, install some necessary packages for the USB install. In a terminal window,

```
$ sudo pacman -S --needed dosfstools rsync unzip base-devel uboot-tools mkinitcpio-utils mkinitcpio-netconf
```

Most packages will probably be already installed, hence the --needed

The current Archlinux bootloader does not support USB booting. We need to utilize a release candidate version which does support it. In the future, when the rc bootloader is released as stable, this step will probably not be necessary.

Go to

<https://dwngrit.github.io/>

Enter the following URL

<https://github.com/archlinuxarm/PKGBUILDS/tree/master/alm/uboot-raspberrypi>

Then click download. The uboot-raspberrypi zip archive will be downloaded for you.

In a file manager, create a temporary directory, such as ~/Temp and then move the zip archive to that directory. Use an Archive Manager to extract the files into ~/Temp In a terminal window, cd to the ~/Temp/uboot-raspberrypi directory and as user (please do not use root for this). Use vi or nano

```
$ nano PKGBUILD
```

change

```
pkgver FROM '2020.07' TO ' 2020.10rc2 '
```

change the first md5sum value from

```
86e51eecd15e658ad1df943a0edf622 TO bae5280c7ce49961c3722fa9019535bf
```

Close nano

Make sure you are in the PKGBUILD's directory. Again, as user

```
$ makepkg -s          (this will build the package but NOT install it)
```

Now we change from the old bootloader to the release candidate bootloader

```
$ sudo pacman -R uboot-raspberrypi
```

```
$ sudo pacman -U uboot-raspberrypi-2020.10rc2-2aarch64.pkg.tar.xz
```

```
$ sudo systemctl reboot
```

Change mkinitcpio. Use vi or nano to edit

```
$ sudo nano /etc/mkinitcpio.conf
```

In the MODULES section, add two modules ' pcie_brcmstb ' and ' broadcom '. It should look like this
MODULES=(pcie_brcmstb broadcom) plus any modules that were already there if any.

In the HOOKS section insert ' keyboard ' and ' keymap ' after the autodetect entry.

If ' keyboard ' is already present, move it to the proper location. It should look like this

```
HOOKS=(base udev autodetect keyboard keymap modconf block sleep netconf filesystems fsck)
```

close nano and rebuild initramfs

```
$ sudo mkinitcpio -P    ( may see ' Possibly missing firmware ' warnings, ignore them)
```

All the preparations are complete. Next is to copy our EndeavourOS install to the USB device.

First we need to prepare the USB device by partitioning, and formatting it. There are several ways to accomplish this. I prefer Gparted.

Click on "GParted" tab and select the USB SD READER (ensure the right device is selected)
Note the Device Name of the SD READER, such as /dev/sda. Write this down.

Click on "Device" tab, and create a msdos Partition Table.

If an existing partition is mounted, this will fail. Highlight the partition with the key symbol, right click on it, select "unmount" then try again.

Click on "Partition" tab, then new

Free Space preceding MiB: 4	Create as: Primary Partition
New Size MiB: 250	Partition name:
Free Space following (MiB): XXXX	File System: fat32
Align to: MiB	Label: BOOT

Create a second partition

Free Space preceding MiB: 0	Create as: Primary Partition
New Size MiB: XXXXXX	Partition name:
Free Space following (MiB): 0	File System: ext4
Align to: MiB	Label: ROOT

Apply All Operations

Close GParted

Now we will copy our EndeavourOS install to the USB 3 device.

In a terminal window, create a new folder in your home directory

```
$ mkdir ~/rpi4-copy
```

```
$ cd ~/rpi4-copy
```

```
$ mkdir usb-boot usb-root
```

```
$ lsblk -f (to identify the device name of the USB, such as /dev/sda)
```

```
$ sudo mount /dev/sda1 usb-boot
```

```
$ sudo mount /dev/sda2 usb-root
```

To copy the system to the USB 3 device.

```
$ sudo rsync - -info=progress2 -axHAX /boot/ usb-boot/ ( the slashes are important as is)
```

```
$ sudo rsync - -info=progress2 -axHAX / usb-root/ ( this may take a while)
```

Empty cache

```
$ sudo sync
```

Adjust /etc/fstab Use vi or nano to edit /etc/fstab

```
$ sudo nano usb-root/etc/fstab
```

comment out existing dev/mmcblk1p1 yada yada line

```
# /dev/mmcblk1p1 /boot vfat defaults 0 0
```

create new entry

```
/dev/sda1 /boot vfat defaults 0 0
```

close nano

```
$ cd ~/rpi4-copy
```

```
$ sudo umount usb-boot usb-root
```

Shutdown the system, remove the micro SD card, and boot into the USB device.

Save the micro SD card and use it as a seed. If your first install was a USB 3 thumb drive and you want to experiment with installing on an external SSD. Then put the micro SD card in the Rpi 4, hook up the USB external SSD, and boot. Go to the section where the external device is partitioned and formatted (page 4) and do page four again.