



Search...



Store ▾

[Blog](#)[Community](#)[Devices](#)[Documentation](#)[Contests](#)[Podcast](#)[🏠 Documentation / Pinebook Pro / Software / Installing Arch Linux ARM](#)[Edit page](#)

Installing Arch Linux ARM

These instructions can be followed to install Arch Linux ARM on an SD Card, USB Flash Drive, eMMC, or even NVMe if your U-Boot supports it (example Tow-Boot on SPI).

Commands to be run as a normal user are prefixed with `$`, commands to be run as root (or with `sudo`) are prefixed with `#`. The target device is assumed to be `/dev/sdb`, adjust accordingly.

Partitioning

Flashing U-Boot

IMPORTANT

While any build of U-Boot for the Pinebook Pro can be used, this tutorial uses [Tow-Boot](#). The process of installing Tow-Boot is different from any other U-Boot build, so large parts of the partitioning section will need to be changed if you want to use something else. If you already have Tow-Boot installed via SPI, you can skip this step. Use `fdisk` to create a blank GPT partition table. `/boot` will be partition 1, and `/` will be partition 2.

Download and extract the latest release of Tow-Boot for the Pinebook Pro from <https://github.com/Tow-Boot/Tow-Boot/releases>.

```
$ wget https://github.com/Tow-Boot/Tow-Boot/releases/download/release-  
$ tar xf pine64-pinebookPro-2021.10-004.tar.xz
```

Flash Tow-Boot to `/dev/sdb` (replace this with the device you actually intend to use).

```
# dd if=pine64-pinebookPro-2021.10-004/shared.disk-image.img of=/dev/sd
```

This creates the partition table for the device, with the first partition serving to protect Tow-Boot. Do not move or write to this partition.

Creating the partitions

Use `fdisk` to add partitions to `/dev/sdb`.

```
# fdisk /dev/sdb
```

Create the `/boot` partition.

- Type `n` to create a new partition.
- Press enter for partition number two.
- Press enter for the default start sector.
- Type `+512M` to make the new partition with 512 MB.

Mark the `/boot` partition bootable.

- Type `x` to enter expert mode.
- Type `A` to mark a partition bootable.
- Type `2` to select partition two.
- Type `r` to exit expert mode.

Create the root partition.

- Type `n` to create a new partition.
- Press enter for partition number three.
- Press enter for the default start sector.
- Press enter to fill the rest of the device.

Write the changes to disk.

- Type `w` to write the changes and exit.

Formatting the partitions

Format the `/boot` partition as a filesystem supported by your U-Boot. ext4 is recommended:

```
# mkfs.ext4 /dev/sdb2
```

Format the root partition as any filesystem supported by Arch Linux ARM. btrfs for example:

```
# mkfs.btrfs /dev/sdb3
```

Installing the root filesystem

Mounting the partitions

```
# mount /dev/sdb3 /mnt
# mkdir /mnt/boot
# mount /dev/sdb2 /mnt/boot
```

Downloading and verifying the rootfs tarball

Download the tarball and its PGP signature.

```
$ wget http://os.archlinuxarm.org/os/ArchLinuxARM-aarch64-latest.tar.g
```

Import the Arch Linux ARM signing key.

```
$ gpg --keyserver keyserver.ubuntu.com --recv-keys 68B3537F39A313B3E57.
```

Verify the tarball's authenticity.

```
$ gpg --verify ArchLinuxARM-aarch64-latest.tar.gz.sig
```

Verifying the authenticity of the tarball protects you in two ways:

1. Makes sure the tarball came directly from Arch Linux ARM and was not tampered with
2. Prevents you from using a corrupt tarball (for example from an interrupted download)

Extracting and configuring the root filesystem

Extracting the root filesystem

```
# bsdtar -xpf ArchLinuxARM-aarch64-latest.tar.gz -C /mnt
```

Editing fstab

Find the partitions' UUIDs with `blkid`.

```
# blkid /dev/sdb3 /dev/sdb2
```

Example output:

```
/dev/sdb3: UUID="c1ec9712-5c64-46da-852c-9d665416e8a6" UUID_SUB="90e5b1  
/dev/sdb2: UUID="21bbff3f-b82e-416e-93c8-e6d44c3daf82" BLOCK_SIZE="409
```

Add the following lines to **/mnt/etc/fstab**, substituting the example UUIDs with those you received from **blkid**.

```
UUID=c1ec9712-5c64-46da-852c-9d665416e8a6 /      btrfs defaults 0 1  
UUID=21bbff3f-b82e-416e-93c8-e6d44c3daf82 /boot ext4  defaults 0 2
```

Creating extlinux.conf

Create a file **/mnt/boot/extlinux/extlinux.conf** with the following contents, replacing the example UUID with the one for **/dev/sdb3** from **blkid**.

```
DEFAULT arch  
MENU TITLE Boot Menu  
PROMPT 0  
TIMEOUT 50  
  
LABEL arch  
MENU LABEL Arch Linux ARM  
LINUX /Image  
INITRD /initramfs-linux.img  
FDT /dtbs/rockchip/rk3399-pinebook-pro.dtb  
APPEND root=UUID=c1ec9712-5c64-46da-852c-9d665416e8a6 rw  
  
LABEL arch-fallback  
MENU LABEL Arch Linux ARM with fallback initramfs  
LINUX /Image  
INITRD /initramfs-linux-fallback.img  
FDT /dtbs/rockchip/rk3399-pinebook-pro.dtb  
APPEND root=UUID=c1ec9712-5c64-46da-852c-9d665416e8a6 rw
```

Booting and finishing setup

Boot into Arch Linux ARM and log in as **root** with password **root**.

Initialize the pacman keyring.

```
# pacman-key --init  
# pacman-key --populate archlinuxarm
```

For security, change the default passwords for root and the default user **alarm**.

```
# passwd  
# passwd alarm
```

You have now installed Arch Linux ARM on your PineBook Pro.