

- 1.) Download the firmware file and partition image from Debians Server

<https://d-i.debian.org/daily-images/arm64/daily/netboot/SD-card-images/>

https://d-i.debian.org/daily-images/arm64/daily/netboot/SD-card-images/firmware.pine64_plus.img.gz
<https://d-i.debian.org/daily-images/arm64/daily/netboot/SD-card-images/partition.img.gz>

- 2.) Merge the firmware file and partition image into bootable Debian image

```
zcat firmware.none.img.gz partition.img.gz > h64_debian.img
```

- 3.) Flash bootable Debian image onto SD-Card

```
lsblk                                     find device name of your SD-Card
sudo dd if=/dev/zero of=/dev/sdX bs=446 count=32770    wipe the Boot Sector of your SD-Card
sudo dd if=h64_debian.img of=/dev/sdX bs=4M conv=fsync  flash Debian image to your SD-Card
```

replace **X** with the device letter of your SD-Card

- 4.) Prepare the eMMC-Module for the Pine64 H64B SBC (16GB eMMC module (30310400 sectors))

```
sudo fdisk /dev/sdX
```

type **o** this will clear out any partitions on the drive
type **p** to list partitions, there should be no partitions left
type **n**, then **p** for primary, **1** for the first partition on the drive,
32768 for the first sector, and **1056767** for the last sector, then type **a**
then type **n**, then **p** for primary, **2** for the second partition on the drive,
1056768 for the first sector, and **28213246** for the last sector, then
type **n**, then **p** for primary, **3** for the third partition on the drive,
28213247 for the first sector, and **30310399** for the last sector,
then type **t**, and **3** for the third partition, and **82** for the Hex Code,
then write the partition table and exit by typing **w**

(steps above create 500M for **/boot**, 12.9GiB for **/** and 1GiB for **swap**)

We will format the newly created partitions later with the Debian Installer.

The 500 MiB partition format as **ext2** and set Mount point to **/boot** and set Label to **boot**.

The 12.9 GiB partition format as **ext4** and set Mount point to **/** and set Label to **root**.

The 1 GiB partition format as **swap**.

- 5.) Install Cross Compiler for building U-Boot on our x86_64 Debian Host

```
sudo apt install device-tree-compiler build-essential libssl-dev python3-dev bison
sudo apt install flex libssl-dev swig gcc-aarch64-linux-gnu gcc-arm-none-eabi
sudo apt install gcc make bc git
```

- 6.) Build U-Boot on our x86_64 Debian Host

```
cd /home/youruser/assets
```

```
git clone https://github.com/ARM-software/arm-trusted-firmware
```

```
cd arm-trusted-firmware
git tag
```

remember last stable (v2.8.0)

```
git checkout v2.8.0
make CROSS_COMPILE=aarch64-linux-gnu- PLAT=sun50i_h6 bl31
cd ..
```

```
git clone git://git.denx.de/u-boot.git
```

```
cd u-boot
git tag                                     remember last stable (v2022.10)
git checkout v2022.10
ln -s /home/youruser/assets/arm-trusted-firmware/build/sun50i_h6/release/bl31.bin bl31.bin
make CROSS_COMPILE=aarch64-linux-gnu- BL31=bl31.bin pine_h64_defconfig
make -j4 CROSS_COMPILE=aarch64-linux-gnu- BL31=bl31.bin

cp -r /home/youruser/assets/u-boot/u-boot-sunxi-with-spl.bin /home/youruser/assets/
cd ..
```

8.) Flash U-Boot (Bootloader) onto the SD-Card for the Pine64 H64B SBC

```
lsblk                                     find device name of your SD-Card
sudo dd if=u-boot-sunxi-with-spl.bin of=/dev/sdX bs=1024 seek=8 conv=notrunc
                                         replace X with the device letter of your SD-Card
                                         once finished, unmount the SD-CARD
```

9.) Flash U-Boot (Bootloader) onto the eMMC-Module for the Pine64 H64B SBC

```
lsblk                                     find device name of your eMMC
sudo dd if=u-boot-sunxi-with-spl.bin of=/dev/sdX bs=1024 seek=8 conv=notrunc
                                         replace X with the device letter of your eMMC-Module
                                         once finished, unmount the eMMC-Module
```

10.) Install the eMMC-Module onto your Pine64 H64B SBC, insert the SD-Card, connect HDMI, Mouse and Keyboard, USB to serial (UART) adapter, USB to Ethernet adapter and power it up, now follow the Debian Installer. (build in Ethernet, WiFi and USB3 do not work during installation)

```
sudo screen /dev/ttyUSB0 115200          connects you to the serial output of the Pine64 H64B
                                         CTRL a k exits screen
```



EXT Connector:

Pin 1	→ TX
Pin 3	→ RX
Pin 5	→ Ground

Select the USB Ethernet Adapter as primary network device for the installation.

Select **Manual** partitioning and format the already created partitions as mentioned in Step 4.

In "Software selection" select only **SSH server** and **standard system utilities**

Ignore the "No boot loader installed" warning and **<Continue>**

At the "Finished the installation" prompt select **<Go Back>** and from the "Debian Installer main menu" select **Execute a self** and **<Continue>** and now continue with step 11.

11.) Create essential but yet missing files

chroot target

DTB file handling

mkdir /boot/dtbs

nano /etc/kernel/postinst.d/copy-dtbs create as below

```
#!/bin/sh
```

```
set -e
version="$1"
```

```
echo Copying current dtb files to /boot/dtbs....
cp -a /usr/lib/linux-image-${version}/. /boot/dtbs/
```

chmod +x /etc/kernel/postinst.d/copy-dtbs

/etc/kernel/postinst.d/copy-dtbs `uname -r`

Bootloader configuration

mkdir /boot/extlinux

nano /boot/extlinux/extlinux.conf create as below

```
TIMEOUT 2
DEFAULT debian
```

```
LABEL debian
    MENU LABEL Debian
    KERNEL /vmlinuz
    INITRD /initrd.img
    FDT /dtbs/allwinner/sun50i-h6-pine-h64-model-b.dtb
    APPEND console=ttyS0,115200 console=tty1 root=LABEL=root rw rootwait
```

12.) Hide kernel messages during boot

nano /etc/sysctl.conf amend as below

```
# Uncomment the following to stop low-level messages on console
kernel.printk = 3 4 1 3
```

- 13.) Set primary network interface back to internal Ethernet Port

`nano /etc/network/interfaces` amend as below

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
```

```
source /etc/network/interfaces.d/*
```

```
# The loopback network interface
auto lo
iface lo inet loopback
```

```
# The primary network interface
auto end0
allow-hotplug end0
iface end0 inet dhcp
```

- 14.) Return back to the Debian Installer

`exit`

`exit`

From the "Debian Installer main menu" select **Finish the installation** and **<continue>** and now continue with step 15.

- 15.) Once installation is finished, add missing WiFi/Bluetooth firmware for RTL8723BS Chipset

`nano /etc/apt/sources.list` amend as below

```
# deb http://deb.debian.org/debian/ bookworm main
```

```
deb http://deb.debian.org/debian/ bookworm main contrib non-free non-free-firmware
deb-src http://deb.debian.org/debian/ bookworm main contrib non-free non-free-firmware
```

```
deb http://deb.debian.org/debian-security bookworm-security main contrib non-free non-free-firmware
deb-src http://deb.debian.org/debian-security bookworm-security main contrib non-free non-free-firmware
```

```
deb http://deb.debian.org/debian/ bookworm-updates main contrib non-free non-free-firmware
deb-src http://deb.debian.org/debian/ bookworm-updates main contrib non-free non-free-firmware
```

```
apt update
apt install firmware-realtek
```

- 16.) Perform system update, remove unnecessary packages, enable filesystem check and enable sudo

```
apt update
apt upgrade
apt full-upgrade
apt autoremove
apt autoclean
```

```
tune2fs -c 1 /dev/mmcblkXp1
tune2fs -c 1 /dev/mmcblkXp2
```

use **lsblk** to find correct block device

```
apt install sudo
```

```
adduser youruser sudo
```

Done, enjoy your setup.