Orange Pi 5 – Simple Overview and Installation with M.2 SSD

crosstalksolutions.com/orange-pi-5-simple-overview-and-installation-with-m-2-ssd

Watch Video At: https://youtu.be/cBqV4QWjolE

The Orange Pi 5 is the latest SBC (single board computer) model from Orange Pi – an open source hardware and software manufacturer based out of China. I have been a long time fan of the Raspberry Pi – a similar SBC product that is now about 3 years old.

However, the issue with the Raspberry Pi's is that chip shortages and global supply chain issues have made them next to impossible to buy – and even when you can get your hands on one, buyers are typically limited to very small quantities of product.

Orange Pi 5 has stepped in to help fill this gap with their latest product, but it's still not without its own issues – we'll cover the pros and cons in this article. And, while the Orange Pi 5 seems to be available via both Amazon and Aliexpress, the Amazon reviews show multiple people who have had their orders cancelled, and the pricing seems to be in flux. I purchased my Orange Pi 5 8GB model in late November, 2022 for $75 + shipping – that same unit is now $90.00 + shipping on Aliexpress. Through Amazon, the 8GB model is $99.99 – so you'll definitely pay a premium to get it quicker. It took about a month for mine to arrive from Aliexpress.

Hardware

Let's start by talking about the hardware – the Orange Pi 5 has some pretty impressive stats. It features a Rockchip RK3588S 8-core 64-bit processor, comes in flavors of 4GB/8GB/16GB/32GB of RAM (though why someone would need an SBC with 32GB of RAM is a mystery to me – if you have a solid use case for an SBC with 32GB of RAM, please comment below), has an integrated ARM Mali-G610 GPU which boasts support for up to 8K / 60fps video, Gigabit Ethernet, HDMI 2.1, and my favorite feature, an M.2 interface for either an NVMe SSD hard drive or a combo WiFi/Bluetooth card.

I won't dig too deep into the specifics of this hardware since if you're a fan of single board computing, you basically know what they can be used for. Most people aren't seriously considering these things as desktop PC replacements, so they don't need to be the most powerful devices in the world.

Additionally, when companies boast about 8K video at 60fps, it also makes me think *'ok… so what?'* because – again – no one is buying an Orange Pi 5 because they need to stream 8K video at 60fps – that's just not realistic…and even if that was your specific need, this probably wouldn't be the best option anyway.

Regardless though – the Orange Pi 5 features a pretty impressive lineup of hardware specs. My favorite thing so far is the integrated M.2 slot on the bottom of the board. I purchased a 128GB M.2 2230 PCIe NVMe drive from Amazon for less than $20 bucks, and it's relatively easy to run the OS off of that SSD so that you aren't relying on a microSD. The SSD is so much faster, and long term, should be a lot more reliable.

Two things that I don't like about the Orange Pi 5 vs. the Raspberry Pi 4 is that first of all, there's no built-in wireless card. If you want WiFi or Bluetooth, you'll need something like this AX210NGW WiFi Card (***note – I have not tried this card with the Orange Pi 5 – but this is what I would buy if I was going to try it out).

The downside to this lack of WiFi is that you have to choose if you want to use the M.2 slot for storage or for wireless capability – though you could probably also use a USB WiFi dongle as well.

Which really brings me to the main issue I've had with this board so far – documentation, or lack thereof.

It's Just a Baby

Yes – this board is so new, there are relatively few articles and videos about how to set it up, which products are compatible with it, which operating systems work best etc. etc. It's getting better every day – for instance, when I first started playing with the board, there was no official support for Armbian for the Orange Pi 5 – then a few days later, they finally added the Orange Pi 5 to their list of supported hardware and created a page to download the OS. I was able to get Armbian running on the Orange Pi 5 using the M.2 SSD using basically these same instructions. Which of course brings us to…

Software

This board is definitely still in its infancy. As I stated earlier, Armbian wasn't available when I first checked, and now it is. Additionally, the official Orange Pi 5 downloadable OS images from their official website are also a bit weird. As of the writing of this article, the 'Orange Pi OS (Droid)' folder is completely empty – so there's no official Orange Pi OS image yet? I have no idea, and I'm not going to spend my life in their forums trying to find answers.

They do however have both an Ubuntu and a Debian image available, so that's what I'm going to focus on in this article. I'm going to walk step by step through downloading the Ubuntu image, flashing it into a microSD card, booting into the Orange Pi 5, and then transferring that image over to the M.2 SSD so that you no longer need the microSD card.

Getting Started

To get started with this project, you're going to need a few things:

Orange Pi 5 – Of course you will have to have an Orange Pi 5 – the one that I'm using is the 8GB of RAM version – it cost me just over $86 bucks shipped from Aliexpress, though the price has since gone up.

microSD card – You'll want a decent microSD card – this typically means a Class 10 card (or better). I'm using a SanDisk Ultra 32GB Class 10 microSD card for this article.

Heat Sinks – These are a good idea to help dissipate the heat from the CPU and other chips on the board.

M.2 SSD – I went for this 128GB KIOXIA 128GB M.2 2230 form factor SSD. Works great on the Orange Pi 5 – you can go for other SSD's with more or less space, but the bottom of the board only supports 2230 or 2242 size SSD's.

PoE Splitter (optional, but useful) – If you want to PoE power your Orange Pi 5, these splitters are great – takes PoE power in and then splits it out to Ethernet + USB-C.

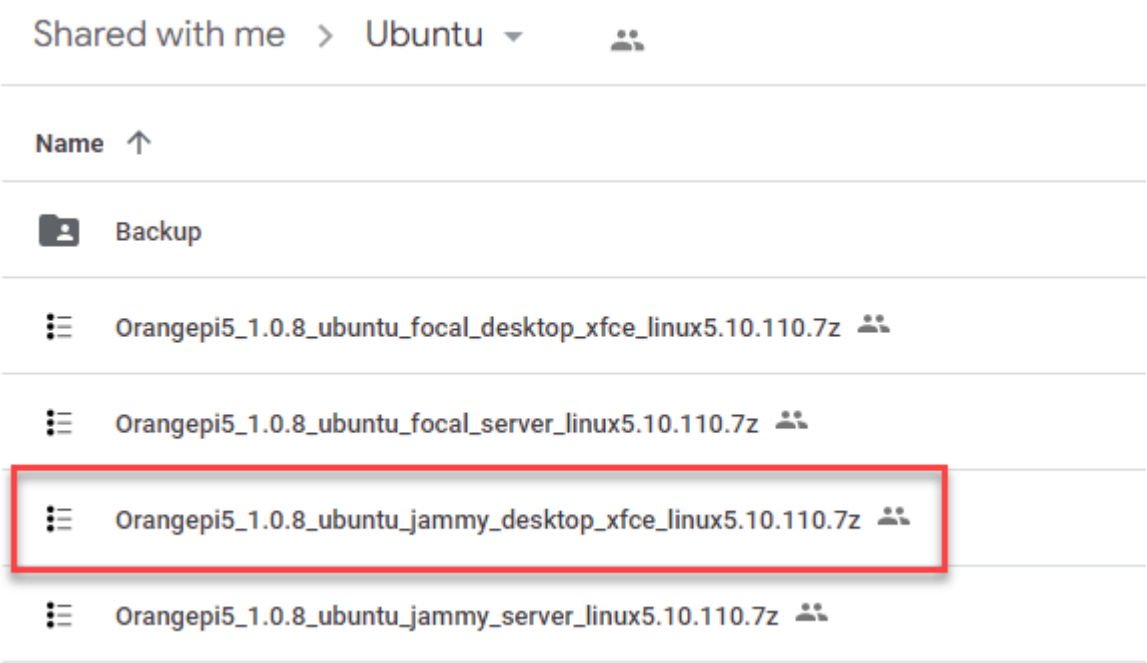
*** HUGE note on the M.2 slot – the Orange Pi 5 does NOT come with a riser screw to secure the M.2, and the hole for that M.2 screw is not threaded – it also has to be TINY – a normal M.2 riser screw that you would get on a standard PC motherboard will be too big. I never really figured out exactly what size screw to use here – after a bunch of searching around, I ended up just finding one in a box of misc. screws and then hot glued it into the board. If anyone DOES know of the correct M.2 screw to purchase that works with this board, please let me know in the comments and I'll update it here.

Prepare the microSD Card

The first thing we need to do here is prepare a microSD card to take the OS image. This tutorial is going to use the Ubuntu Image that can be found here. If that link no longer works, you can just click on Download from the Orange Pi 5 product page and then click into Ubuntu.

*** NOTE: As of the writing of this article, there are 2 versions of Ubuntu in the download folder – 'Jammy' and 'Focal.' The difference is that Jammy is Ubuntu 22.04 and Focal is the older Ubuntu 20.04.

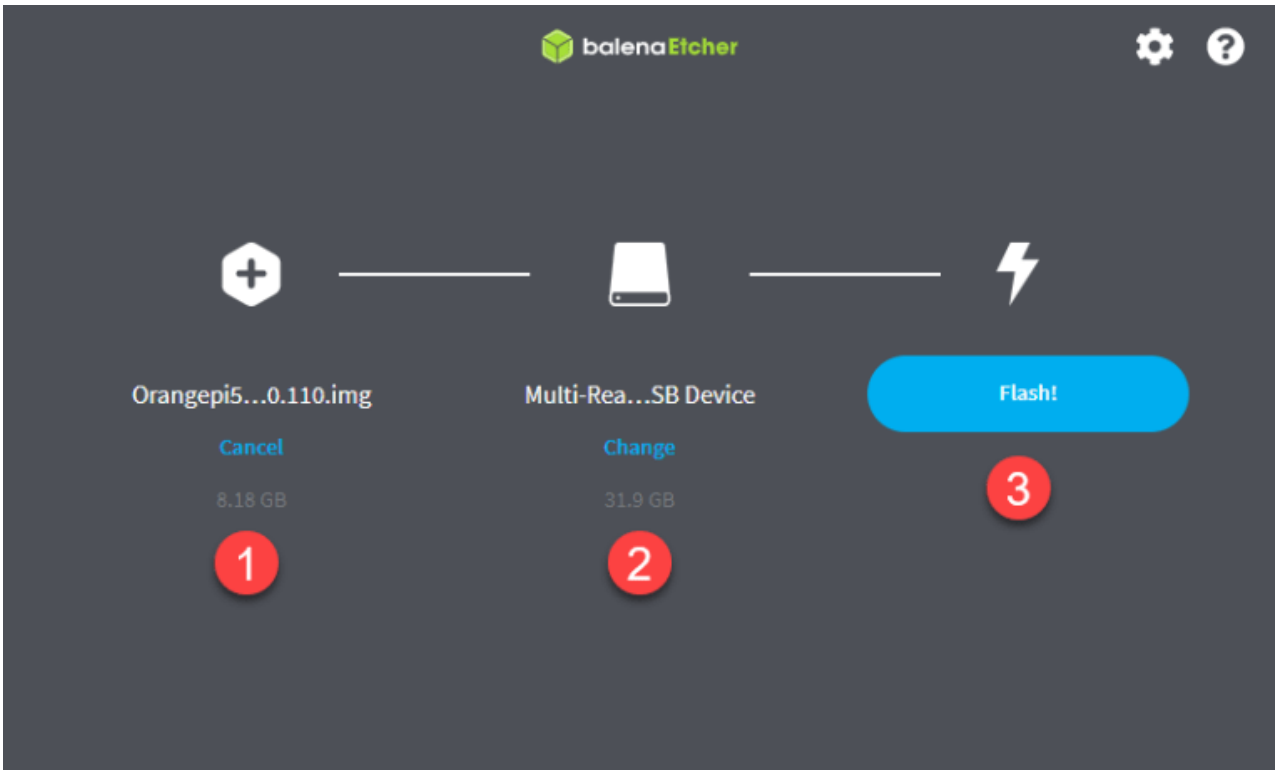
Since we want a desktop version, download the Jammy version with XFCE desktop.



If you need a different OS – or a different version of Ubuntu – or if you're trying a different SBC board. You're on your own. This tutorial is following these specifics – anything else is out of scope, and therefore you'll have to keep Googling to find your answers friend.

Download the 7z file for the Jammy Desktop Ubuntu version and unzip it with something like WinRAR. The unzipped file will be an .img file that can be flashed directly onto the microSD card.

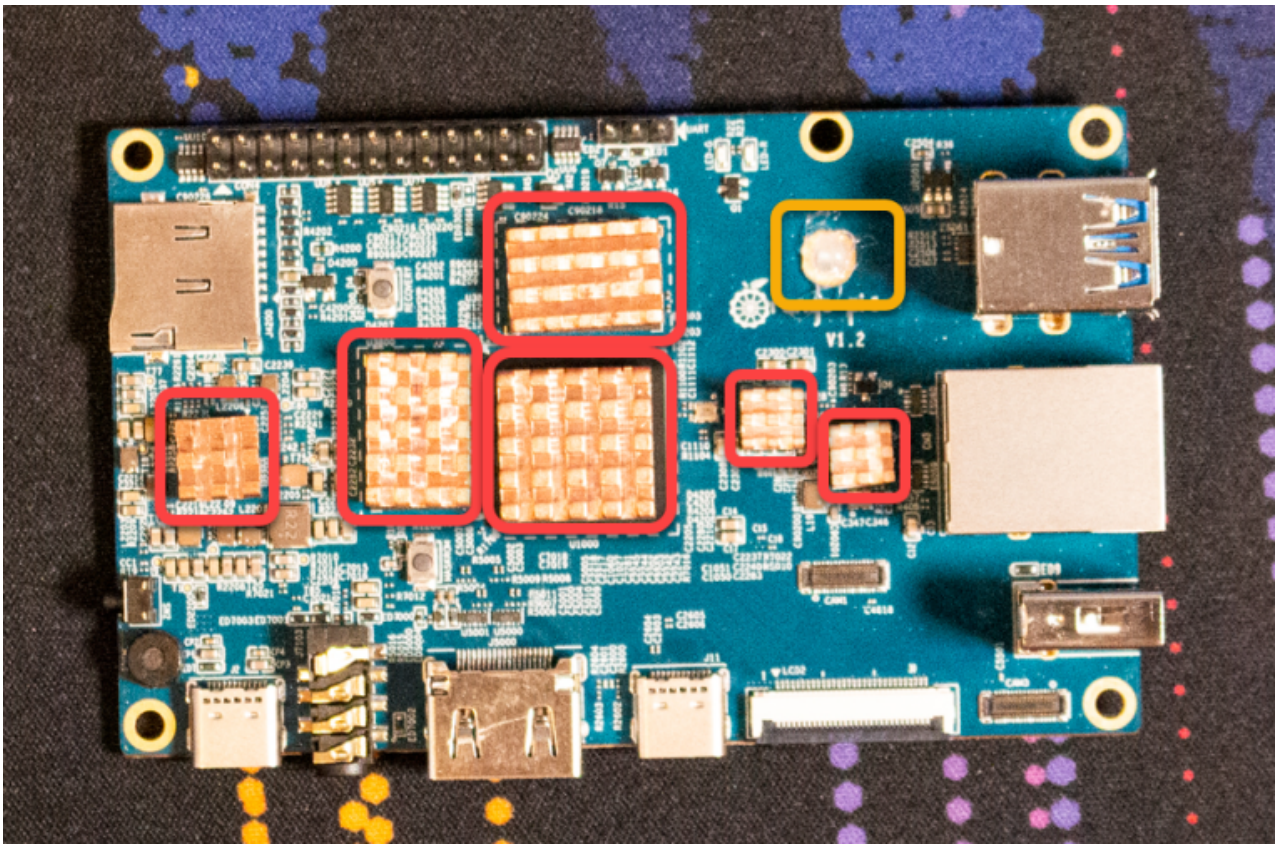
To flash onto the microSD card, we're going to use Balena Etcher. First select 'Flash from File' and choose the Jammy Desktop .img file that was downloaded and unzipped. For target, select your microSD card. Finally click 'Flash' to write the OS image to the microSD card. This takes about 5-6 minutes on my computer, but could take more or less time on yours.



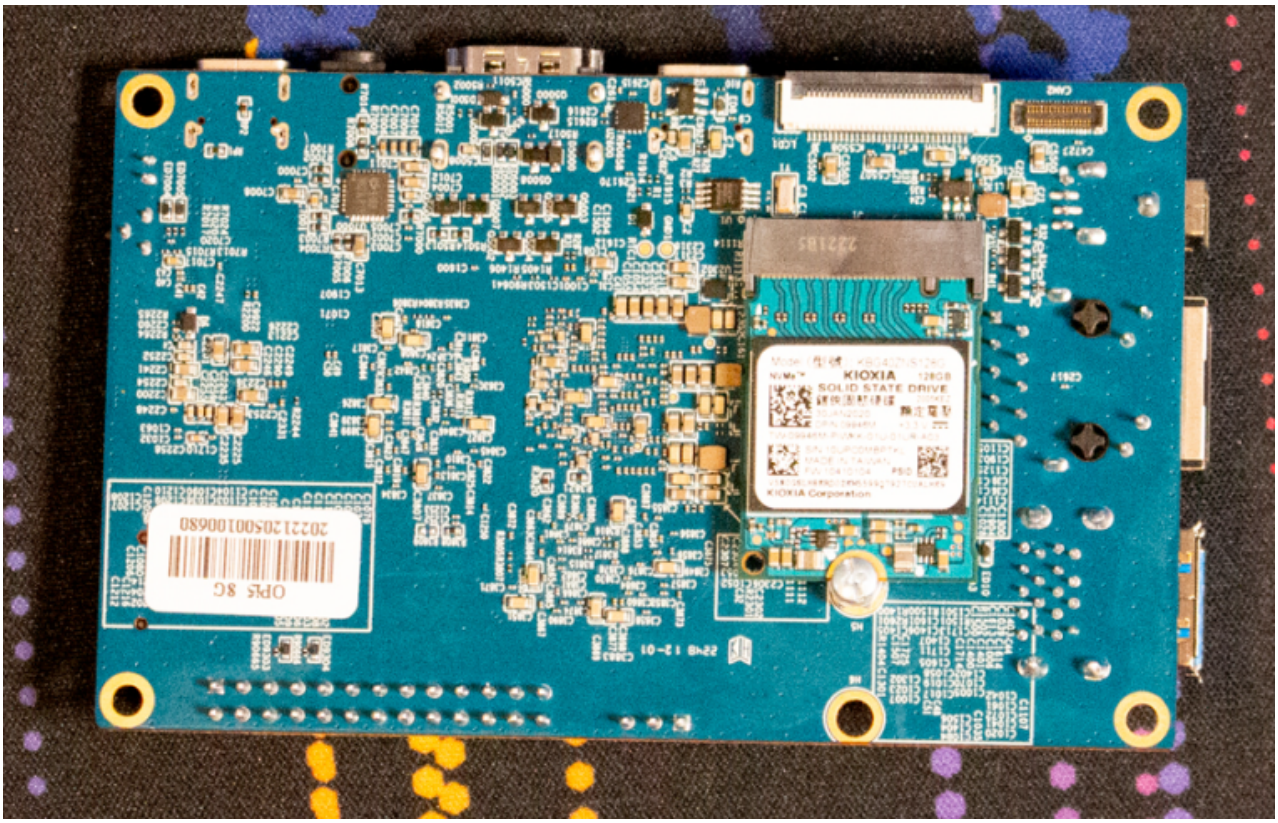
*** NOTE: If you aren't using a fresh new microSD card, you may have to format it. One great way to do so is with SD Card Formatter on Windows.

Prepare the Orange Pi 5

While our microSD card is being flashed, let's prepare the Orange Pi 5. Starting with where to put the heat sinks – I may have gone a bit overboard here, but I essentially just covered up every exposed chip on the top of the board. You can also see in yellow where I hot glued the M.2 screw.



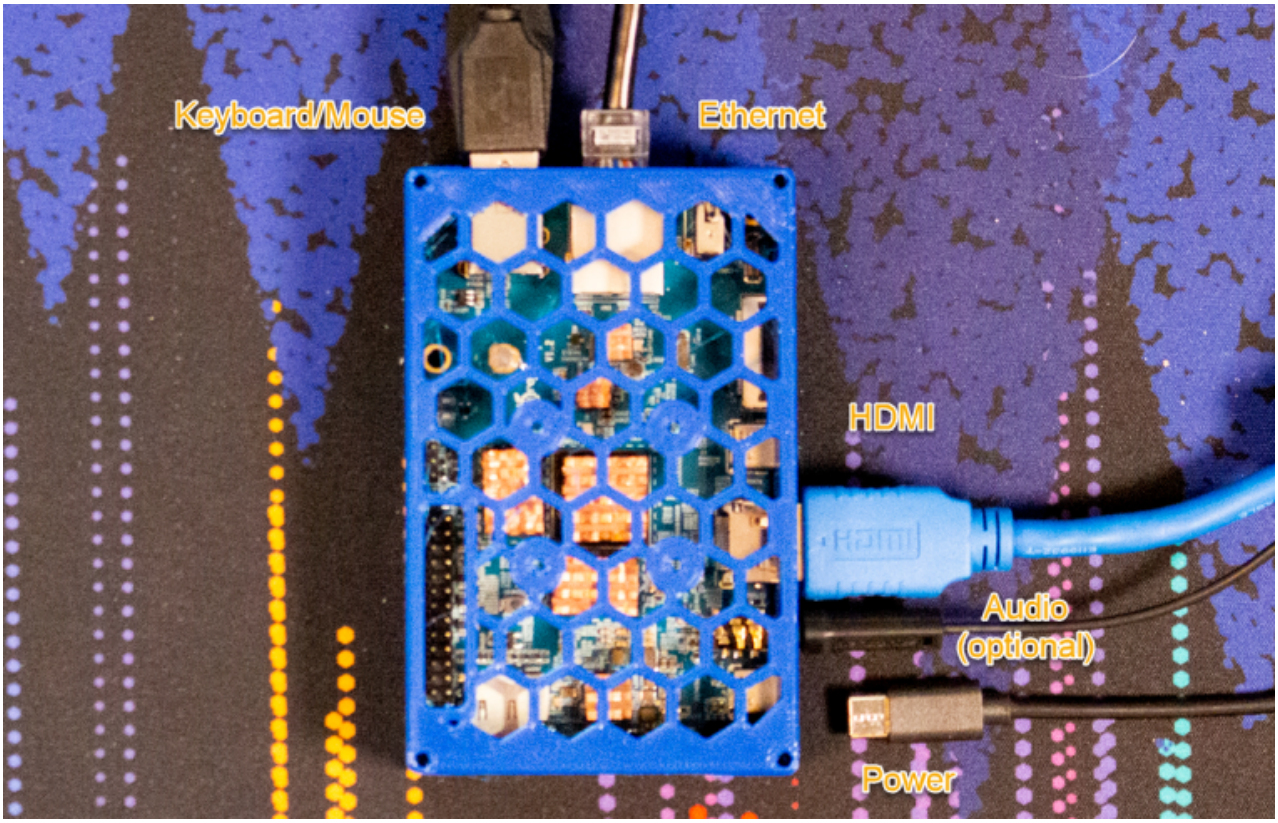
Heat sink installation locations.



Bottom of Orange Pi 5 with M.2 SSD installed.

You'll also want to plug in your devices – you'll probably want a keyboard, mouse, HDMI monitor, and Ethernet cable. You don't need to plug power in just yet – wait for the microSD card to finish.

For a case, I 3D printed this one: https://www.printables.com/model/359018-orange-pi-5-case – it was the 2nd one that I tried – very sturdy, fits the Orange Pi 5 well, and has space for both the M.2 SSD and an optional fan.



Connections – do not plug power in yet!

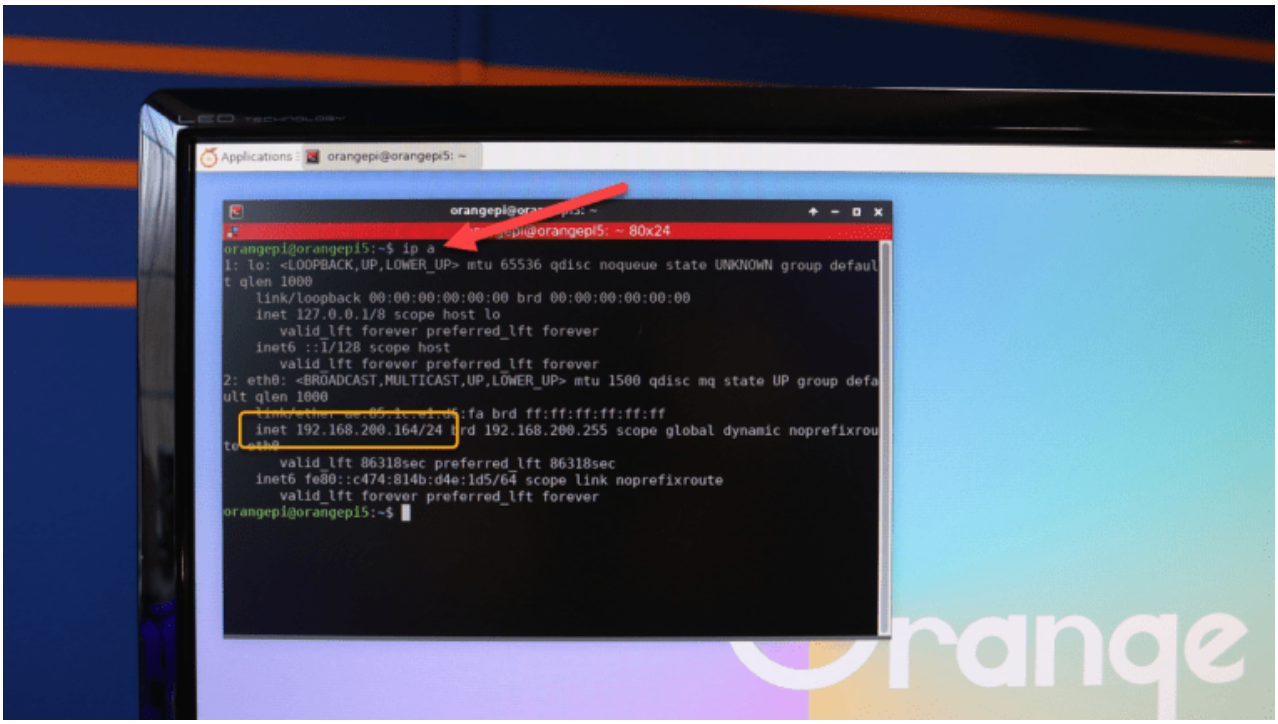
Booting for the First Time

By now Balena Etcher should be done flashing the microSD card, so insert it into the Orange Pi 5 and plug in the power. You should start to see some stuff on the attached monitor.

Once booted, you are automatically logged into the GUI interface. Click on Applications –> Terminal to bring up the Linux CLI and type:

```
ip a
```

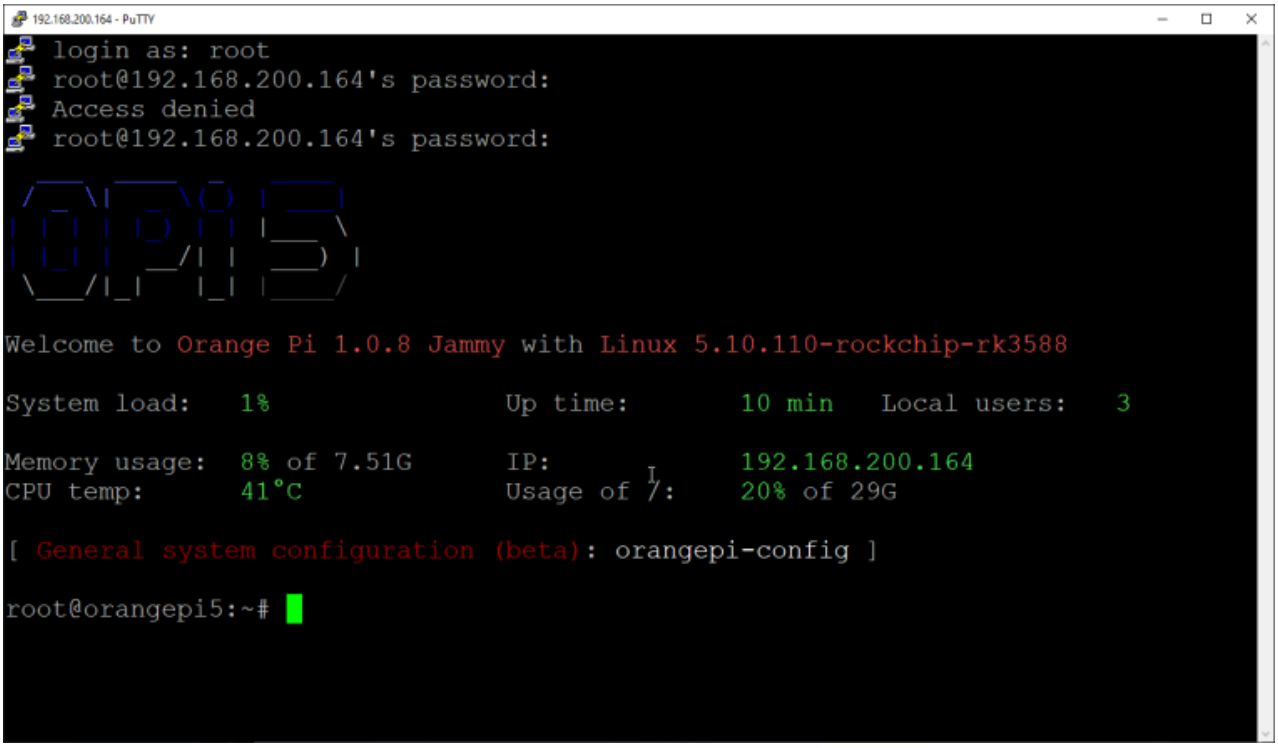
This will show you the IP address that was given to your Orange Pi 5 by your network's DHCP server.



eth0 received 192.168.200.164.24

In my example, I received 192.168.200.164 for the IP address of eth0. You can now SSH to that IP address remotely with a program such as PuTTY to connect remotely to the Linux CLI. The default login is:

User: root
Password: orangepi



Once connected remotely (or from the Terminal window at the console), let's start by making sure everything is up to date:

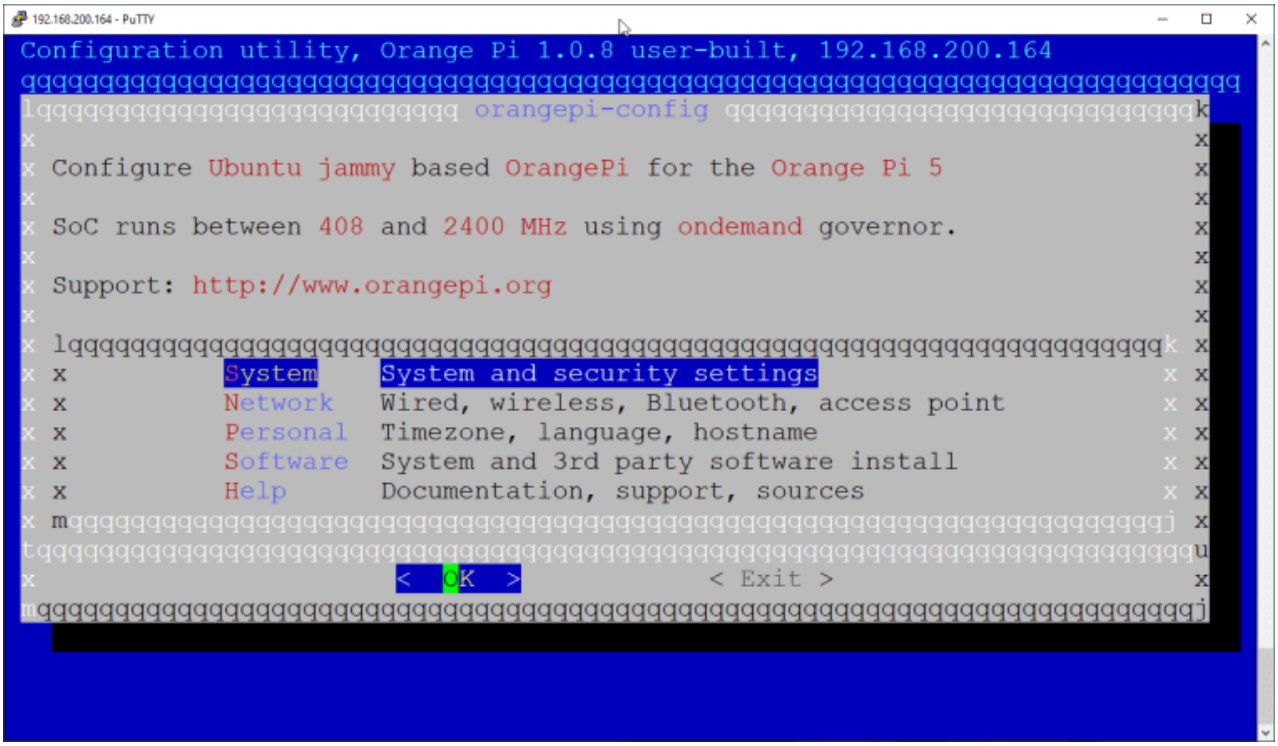
```
sudo apt update
sudo apt upgrade -y
```

For all of the super technical people out there – yes I am aware that you don't need to use sudo when you're logged in root. It's just a force of habit – just go with it and don't be so critical.

Once you're up to date, you're basically done! Congrats – you can now move forward and use your Orange Pi 5 for whatever awesome projects you have in mind.

One thing you may want to do first is run:

```
sudo orangepi-config
```



Running the orangepi-config menu.

This is where you can go through and configure the Orange Pi 5's settings – for instance, choose Personal –> Timezone to set your timezone to your local area.

You may also want to brush through the 'Software' menu – there are a number of great utilities and programs that you can install including Samba for accessing the file system of the Orange Pi 5 from a networked Windows computer (Software –> Softy –> enable Samba).

Booting from the M.2 SSD

As of now, you're still booting the Orange Pi 5 and running the OS off of the microSD card. This isn't always ideal – microSD cards don't live as long as SSD's, and they are much much slower. So let's fix that.

First though, let's take a tour around the various drives that the Orange Pi 5 sees – run:

```
sudo fdisk -l
```

This command shows us all of our various drives – let's take a closer look:

```
Disk /dev/mmcblk1: 29.72 GiB, 31914983424 bytes, 62333952 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
Disklabel type: gpt
Disk identifier: 58FCCC02-859A-5041-ADA7-14802AADF341

Device          Start      End  Sectors  Size Type
/dev/mmcblk1p1  61440   585727   524288  256M Linux extended boot
/dev/mmcblk1p2 585728 61702143 61116416 29.1G Linux filesystem
```

Drive /dev/mmcblk1 is the microSD card.

The disk labeled /dev/mmcblk1 is our microSD card. From this info we can see that it's 29.72GB in size (32GB formatted down), and that it has 2 partitions – /dev/mmcblk1p1 and /dev/mmcblk1p2 – these are the boot and OS partitions.

```
Disk /dev/nvme0n1: 119.24 GiB, 128035676160 bytes, 250069680 sectors
Disk model: KBG40ZNS128G NVMe KIOXIA 128GB
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 58FCCC02-859A-5041-ADA7-14802AADF341

Device          Start       End   Sectors   Size Type
/dev/nvme0n1p1  61440    585727    524288   256M Linux extended boot
/dev/nvme0n1p2 585728 247562239 246976512 117.8G Linux filesystem
```

/dev/nvme0n1 is our M.2 SSD.

We can also see /dev/nvme0n1 – this is the M.2 SSD. So now we know that the Orange Pi 5 can see this drive no problem. If you do not see an NVMe drive when you run the fdisk command, you should probably start by shutting everything down, re-seating the SSD and then booting it back up.

In my case, I have a 128GB SSD which formatted down to 119.24GB and it currently has 2 partitions – /dev/nvme0n1p1 and /dev/nvme0n1p2. YOUR M.2 drive may be empty of partitions if it's brand new – but if you've been messing around with the Orange Pi 5 for a bit, you'll likely have something on there – we're going to address how to take care of that in the next section.

```
Disk /dev/mtdblock0: 16 MiB, 16777216 bytes, 32768 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
Disklabel type: gpt
Disk identifier: 311A2BE6-C13A-4853-A08F-616708BFD0D3

Device           Start   End Sectors  Size Type
/dev/mtdblock0p1    64  7167    7104  3.5M Linux filesystem
/dev/mtdblock0p2  7168  7679     512  256K Linux filesystem
/dev/mtdblock0p3  7680  8063     384  192K Linux filesystem
/dev/mtdblock0p4  8064  8127      64   32K Linux filesystem
/dev/mtdblock0p5  8128  8191      64   32K Linux filesystem
/dev/mtdblock0p6  8192 16383    8192    4M Linux filesystem
/dev/mtdblock0p7 16384 32734   16351    8M Linux filesystem
```

/dev/mtdblock0 is the onboard 16GB flash storage.

The last drive I want to bring your attention to is the onboard 16MB SPI flash drive – yours may or may not have any partitions depending on if it's a brand new setup or not. Before we move forward, let's side-step a bit to clear out any unnecessary partitions.

Clearing Out Pre-existing Partitions (optional)

If you're like me, you reinstall the SBC over and over trying out different configurations and operating systems. But – each time you do that, you're leaving a bit of a mess that needs to be cleaned up before those drives can be used again. If this is your FIRST time booting up the Orange Pi 5 and attempting to install the OS to the M.2 SSD, then you can skip these steps – BUT – when you need to re-install, you'll likely have to clear out these partitions before it works like it did the first time.

Let's clear out the M.2 SSD first – run this command:

```
sudo gdisk /dev/nvme0n1
```

This will bring you to the gdisk menu where we can manage partitions – next to Command press 'p' to print a list of existing partitions on this SSD.

```
Command (? for help): p
Disk /dev/nvme0n1: 250069680 sectors, 119.2 GiB
Model: KBG40ZNS128G NVMe KIOXIA 128GB
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): 58FCCC02-859A-5041-ADA7-14802AADF341
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 2048, last usable sector is 250069646
Partitions will be aligned on 2048-sector boundaries
Total free space is 2566799 sectors (1.2 GiB)

Number  Start (sector)    End (sector)  Size       Code  Name
   1           61440          585727   256.0 MiB   EA00  bootfs
   2          585728       247562239   117.8 GiB   8300

Command (? for help):
```

Existing partitions on my M.2 SSD.

Here we can see 2 partitions – 1 is 256MB and is labeled 'bootfs' (boot file system) – the second is the rest of the drive space on the SSD. Press 'd' followed by 1 to delete the 1st partition. Then press 'd' again to delete the last partition (it will assume you mean the only partition left on the drive, so you don't have to explicitly tell it to delete partition 2).

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

Command (? for help): d
Using 2

Command (? for help):
```

Removing existing partitions from the M.2 SSD.

Now if you type 'p' again you won't see any remaining partitions – the M.2 SSD has been cleared out.

To exit, type 'w' and then 'Y' to confirm that you want to make these changes.

We also need to do the same for the 16MB SPI flash drive – we're going to use that SPI flash drive to eventually tell the Orange Pi to boot from the M.2 SSD instead of the microSD card, but just like before, if this isn't your first rodeo, you have to clear out the old stuff before you can move forward. Type the following:

```
sudo gdisk /dev/mtdblock0
```

You can now manage the partitions on the 16MB SPI flash storage of the Orange Pi 5 – press ‘p’ to take a look at the existing partitions.

```
Command (? for help): p
Disk /dev/mtdblock0: 32768 sectors, 16.0 MiB
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): 311A2BE6-C13A-4853-A08F-616708BFD0D3
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 32734
Partitions will be aligned on 64-sector boundaries
Total free space is 30 sectors (15.0 KiB)

Number  Start (sector)    End (sector)  Size       Code  Name
   1              64            7167   3.5 MiB     8300  idbloader
   2            7168            7679   256.0 KiB   8300  vnvm
   3            7680            8063   192.0 KiB   8300  reserved_space
   4            8064            8127   32.0 KiB    8300  reserved1
   5            8128            8191   32.0 KiB    8300  uboot_env
   6            8192           16383   4.0 MiB     8300  reserved2
   7           16384           32734   8.0 MiB     8300  uboot
```

Existing partitions on the 16MB SPI flash storage.

As you can see there are 7 existing partitions because I have already used this Orange Pi 5 for other operating systems. Let’s clear them out by pressing ‘d’ and then choosing each partition number one at a time. Once they’re all gone, press ‘w’ and confirm with ‘Y’ to save those changes.

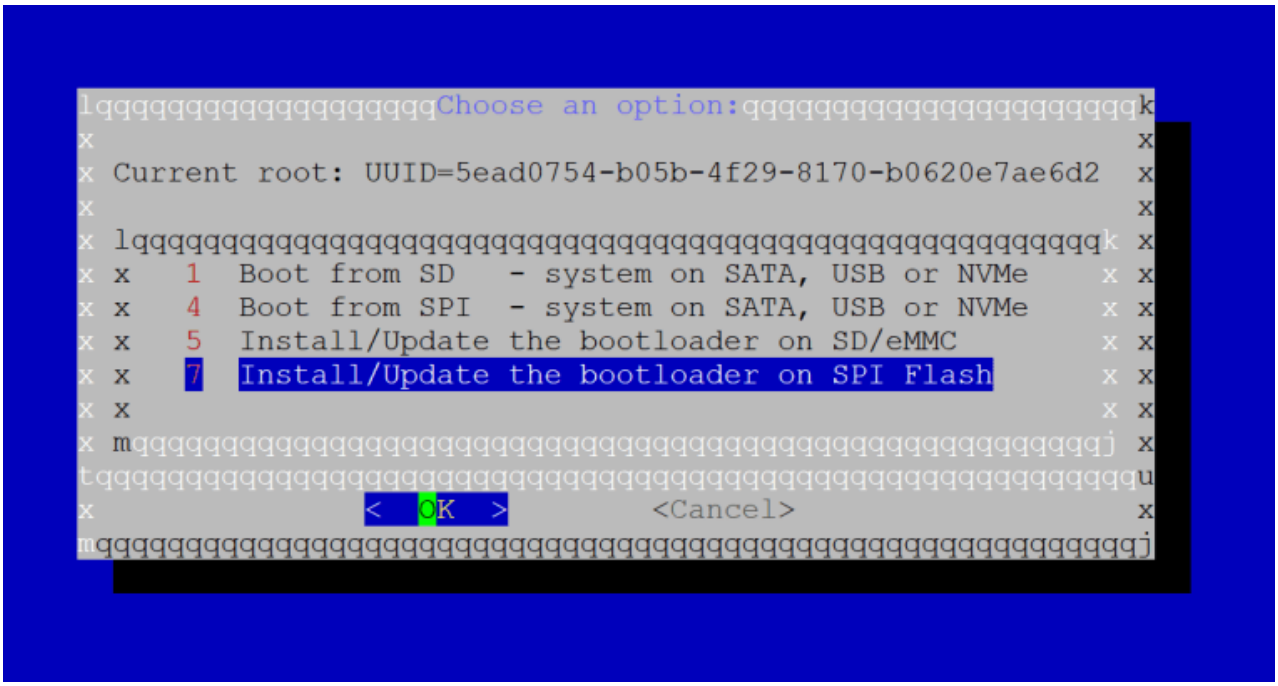
Once that’s done, you should be good to continue with installing the OS to the M.2 SSD so that the Orange Pi 5 can be fully booted without the need for a microSD card at all.

Transferring the OS image over to the M.2 SSD

The first step of this process is to tell the Orange Pi 5 to use its 16MB SPI flash storage for u-boot. To do this, run this command:

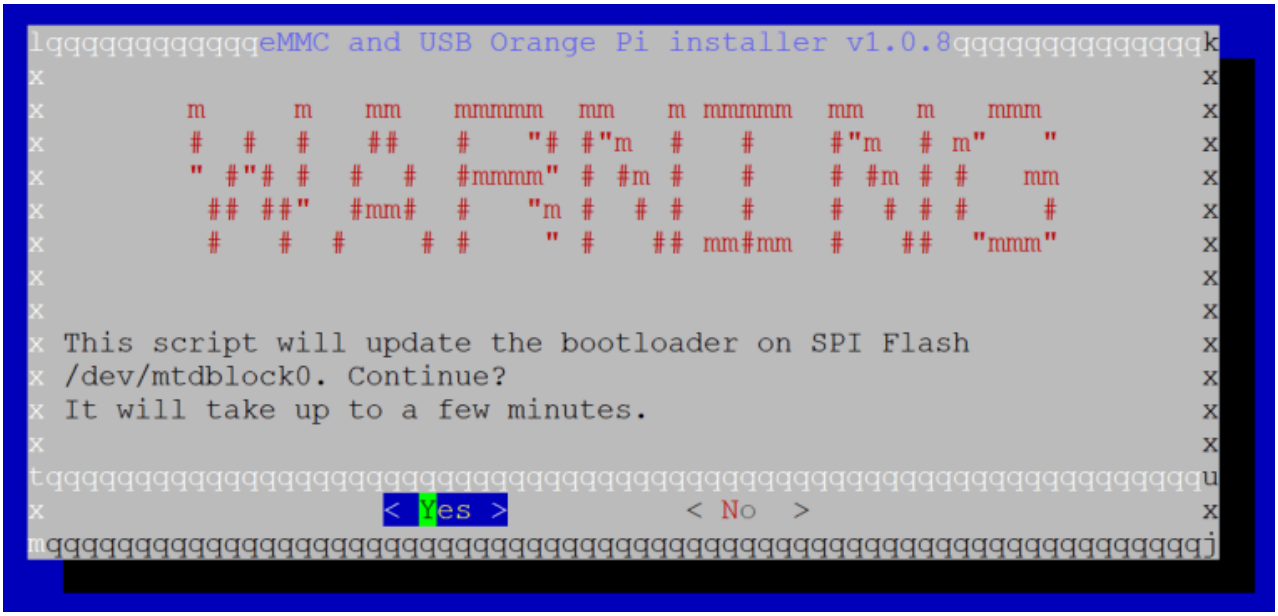
```
sudo nand-sata-install
```

You can also get to this same screen by running ‘sudo orangepi-config’ and choosing System –> Install.



Choose option 7 to install the bootloader on the 16MB SPI flash storage.

Choose option 7 – Install/Update the bootloader on SPI Flash – this process takes a few minutes, and will kick you back to the Linux CLI when complete.



*** NOTE if you get any errors about not enough space on the SPI flash, follow the instructions in the optional blue section above.

If you take a look at the SPI flash drive after this process, you should see a bunch of partitions now:

```
sudo fdisk -l /dev/mtdblock0
```

```
root@orangepi5:~# fdisk -l /dev/mtdblock0
Disk /dev/mtdblock0: 16 MiB, 16777216 bytes, 32768 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
Disklabel type: gpt
Disk identifier: 311A2BE6-C13A-4853-A08F-616708BFD0D3

Device           Start   End Sectors  Size Type
/dev/mtdblock0p1    64  7167    7104  3.5M Linux filesystem
/dev/mtdblock0p2  7168  7679     512  256K Linux filesystem
/dev/mtdblock0p3  7680  8063     384  192K Linux filesystem
/dev/mtdblock0p4  8064  8127      64   32K Linux filesystem
/dev/mtdblock0p5  8128  8191      64   32K Linux filesystem
/dev/mtdblock0p6  8192 16383    8192    4M Linux filesystem
/dev/mtdblock0p7 16384 32734   16351    8M Linux filesystem
root@orangepi5:~#
```

7 partitions on the SPI flash storage drive.

Now that we've prepared the SPI flash drive, we now need to flash an OS image over to the M.2 SSD. This is basically the same concept as what we did earlier using Balena Etcher to flash an OS image to our microSD card. The only problem is that we don't have an OS image that we can use to flash the M.2 SSD! So first we'll need to copy one onto the microSD card remotely.

*** NOTE: In this tutorial, I'm going to use the scp command in Windows PowerShell to push the image over to the Orange Pi 5 – but there are many ways to skin this cat. You could even just copy the image to a USB thumb drive, plug that drive into the Orange Pi 5 and copy it over locally.

Open up Windows PowerShell and navigate to the location of the image file that we unzipped at the beginning of this tutorial. To copy that image file over to the Orange Pi 5, run this command:

```
scp .\Orangepi5_1.0.8_ubuntu_jammy_desktop_xfce_linux5.10.110.img
root@192.168.200.164:/home/orangepi/Downloads
```

*** Note: the command above is on all one line – it formats to 2 lines because of WordPress.

Let's break that command down – scp = Secure Copy Protocol (or Secure CoPy – basically just a fancy copy command). Then we have the file we want to copy over – in our case, it's the Orange Pi Ubuntu OS image file that we extracted earlier in this tutorial. NOTE that your version of OS image file may be different than what I have in this tutorial, so pay special attention to the actual file name.

After that, we're copying as the root user (root@) and the IP address of our Orange Pi – in my case, it's 192.168.200.164, but your IP will most likely be different. Finally, we're indicating where we want to drop that file – we're putting it in /home/orangepi/Downloads. Syntax is important here – so double and triple-check that you have everything correct before pressing ENTER.

Copying that file over should just take a few minutes – once it's done, you can run this command in the Orange Pi 5 CLI to verify that it copied over properly:

```
ls -lah /home/orangepi/Downloads
```

```
root@orangepi5:/home/orangepi/Downloads# ls -lah /home/orangepi/Downloads/
total 7.7G
drwxr-xr-x  2 orangepi orangepi 4.0K Jan  8 16:47 .
drwxr-x--- 16 orangepi orangepi 4.0K Dec 22 03:35 ..
-rw-r--r--  1 root     root     7.7G Jan  8 16:52 Orangepi5_1.0.8_ubuntu_jammy_desktop_xfce_linux5.10.110.img
```

OK – we've got our OS image on the Orange Pi 5 – let's push it to the M.2 SSD! Use this command (all one line):

```
sudo dd bs=1M if=Orangepi5_1.0.8_ubuntu_jammy_desktop_xfce_linux5.10.110.img
of=/dev/nvme0n1 status=progress
```

This process takes 5-6 minutes. Note that if your OS image file version is different than mine, use your own version/image file name.

Once complete – shut down the Orange Pi 5 with this command:

```
sudo shutdown -h now
```

Then pull out the microSD card and press the power button on the Orange Pi 5 to boot it back up. Once it's booted back up, keep in mind that this is basically a completely new OS – you most likely have a new IP address from DHCP, so run Applications –> Terminal to get to the Linux CLI and run 'ip a' to see your new IP.

Once you're reconnected to the Orange Pi 5, you should probably update the OS again with the commands we used towards the beginning of this tutorial. You'll also need to reset any configuration changes you made in orangepi-config.

One Clean-up Item

If you put the microSD card back into the Orange Pi 5 and then reboot, the Orange Pi 5 will be booted back into the OS on the microSD card. This is because the UUID (Universally Unique IDentifier) of the OS image on the microSD card (/dev/mmcblk1p2) is the same as the UUID on the M.2 SSD (/dev/nvme0n1p2) – we can see this by running this command:

```
sudo blkid
```

```
root@orangepi5:~# sudo blkid
/dev/nvme0n1p1: SEC_TYPE="msdos" LABEL_FATBOOT="opi_boot" LABEL="opi_boot" UUID="4CA7-1DD8" BLOCK_SI
/dev/nvme0n1p2: LABEL="opi_root" UUID="5ead0754-b05b-4f29-8170-b0620e7ae6d2" BLOCK_SIZE="4096" TYPE=
/dev/zram0: UUID="b1ba3c73-23f2-4708-8e37-d709401b177a" TYPE="swap"
/dev/zram1: LABEL="log2ram" UUID="c98b292b-f870-4bb6-b0ba-8767103b7fc3" BLOCK_SIZE="4096" TYPE="ext4
/dev/mmcblk1p1: SEC_TYPE="msdos" LABEL_FATBOOT="opi_boot" LABEL="opi_boot" UUID="4CA7-1DD8" BLOCK_SI
/dev/mmcblk1p2: LABEL="opi_root" UUID="5ead0754-b05b-4f29-8170-b0620e7ae6d2" BLOCK_SIZE="4096" TYPE=
/dev/mtdblock0: PTUUID="311a2be6-c13a-4853-a08f-616708bfd0d3" PTTYPE="gpt"
root@orangepi5:~#
```

Same UUID

Same UUID on our M.2 SSD and the microSD card.

To fix this, we can run this command:

```
sudo tune2fs -U random /dev/mmcblk1p2
```

This tells the Orange Pi 5 to change the UUID on the 2nd partition of our microSD card. You can verify that it changed by running the blkid command again:

```
root@orangepi5:~# sudo tune2fs -U random /dev/mmcblk1p2
tune2fs 1.46.5 (30-Dec-2021)
root@orangepi5:~# sudo blkid
/dev/nvme0n1p1: SEC_TYPE="msdos" LABEL_FATBOOT="opi_boot" LABEL="opi_boot" UUID="4CA7-1DD8" BLOCK_SIZE="512" TYPE="vfa
/dev/nvme0n1p2: LABEL="opi_root" UUID="5ead0754-b05b-4f29-8170-b0620e7ae6d2" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="
/dev/zram0: UUID="b1ba3c73-23f2-4708-8e37-d709401b177a" TYPE="swap"
/dev/zram1: LABEL="log2ram" UUID="c98b292b-f870-4bb6-b0ba-8767103b7fc3" BLOCK_SIZE="4096" TYPE="ext4"
/dev/mmcblk1p1: SEC_TYPE="msdos" LABEL_FATBOOT="opi_boot" LABEL="opi_boot" UUID="4CA7-1DD8" BLOCK_SIZE="512" TYPE="vfa
/dev/mmcblk1p2: LABEL="opi_root" UUID="232337c8-c207-409c-aff5-11ad6ced079a" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="
/dev/mtdblock0: PTUUID="311a2be6-c13a-4853-a08f-616708bfd0d3" PTTYPE="gpt"
root@orangepi5:~#
```

Different UUID's - good!

Successfully changed the UUID of the microSD card's 2nd partition.

Now, if you reboot the Orange Pi 5, it won't boot off of this particular microSD card when it's inserted.

Speed Testing!

Let's now check the speed of our OS running on the M.2 SSD. To test – we'll use a command that we copied from pibenchmarks.com (all one line):

```
sudo curl
https://raw.githubusercontent.com/TheRemote/PiBenchmarks/master/Storage.sh | sudo
bash
```

This command downloads and installs the necessary speed testing benchmark software and runs it – it will first detect your hard drive hardware and then run a few tests. Once complete, it asks for a description of the speed test (can be anything) as well as your Alias (leave blank for Anonymous). Entering an Alias every time you run this benchmark will save all of your results under that same Alias on pibenchmarks.com.

In my testing, I got a score of 18,691 which is pretty impressive.

```
Category                    Test                    Result
HDParm                      Disk Read               358.55 MB/s
HDParm                      Cached Disk Read        363.53 MB/s
DD                          Disk Write              215 MB/s
FIO                         4k random read          51328 IOPS (205313 KB/s)
FIO                         4k random write         34829 IOPS (139319 KB/s)
IOZone                      4k read                 55056 KB/s
IOZone                      4k write                94928 KB/s
IOZone                      4k random read          41131 KB/s
IOZone                      4k random write         78084 KB/s

                            Score: 18691

Compare with previous benchmark results at:
https://pibenchmarks.com/
```

Pibenchmarks.com score.

After a few minutes, your score will also appear on pibenchmarks.com under 'Latest.'

Test #65,691 - 1/8/23 7:46 pm

Score: **18,691**

User: **CrosstalkSolutions**

Brand: Kioxia OEM PCIe NVMe SSD

Board: Orange Pi 5

Amazon

Device: Kioxia KBG40ZNS128GNVMeKIOXIA128GB

Class: SSD (PCIe NVMe)

Note: *Orange Pi 5 with 128 M.2 SSD running Ubuntu*

Nice score!

So that's it! Hopefully I have given you enough information and explanations to venture forth on your own trying out different OS's and configurations for your Orange Pi 5.

If you found this tutorial helpful, consider buying me a beer!

Or consider buying any of the awesome items in the Crosstalk Solutions merch store: