



# **Part 13**

-

# **Activating USB**

*Version: 2021-01-13*

## Connecting an external storage device

You can connect your external hard disk, SSD or USB stick to any of the USB ports on the Raspberry Pi and mount the file system to access the data stored on it.

By default, your Raspberry Pi automatically mounts some of the popular file systems such as FAT, NTFS, and HFS+ at the `/media/pi/<HARD-DRIVE-LABEL>` location.

To set up your storage device so that it always mounts to a specific location of your choice, you must mount it manually.

### Mounting a storage device manually

You can mount your storage device at a specific folder location. It is conventional to do this within the `/mnt` folder, for example `/mnt/mydisk`. Note that the folder must be empty.

- Plug the storage device into a USB port on the Raspberry Pi
- List all the disk partitions on the Pi using the following command

```
sudo lsblk -o UUID,NAME,FSTYPE,SIZE,MOUNTPOINT,LABEL,MODEL
```

The Raspberry Pi uses mount points `/` and `/boot`. Your storage device will show up in this list, along with any other connected storage

- Use the `SIZE`, `LABEL`, and `MODEL` columns to identify the name of the disk partition that points to your storage device. For example, `sda1`
- The `FSTYPE` column contains the filesystem type. If your storage device uses an `exFAT` file system, install the `exFAT` driver

```
sudo apt update
sudo apt -y install exfat-fuse
```

- If your storage device uses an `NTFS` file system, you will have read-only access to it. If you want to write to the device, you can install the `ntfs-3g` driver

```
sudo apt update
sudo apt -y install ntfs-3g
```

- Run the following command to get the location of the disk partition

```
sudo blkid
```

For example, `/dev/sda1`

- Create a target folder to be the mount point of the storage device. The mount point name used in this case is `mydisk`. You can specify a name of your choice

```
sudo mkdir /mnt/mydisk
```

- Mount the storage device at the mount point you created

```
sudo mount /dev/sda1 /mnt/mydisk
```

- Verify that the storage device is mounted successfully by listing the contents

```
ls /mnt/mydisk
```

## Setting up automatic USB disk mounting

You can modify the `fstab` file to define the location where the storage device will be automatically mounted when the Raspberry Pi starts up. In the `fstab` file, the disk partition is identified by the universally unique identifier (UUID).

- Get the UUID of the disk partition

```
sudo blkid
```

- Find the disk partition from the list and note the UUID. For example, 5C24-1453
- Open the `fstab` file using a command line editor such as `nano`

```
sudo nano /etc/fstab
```

- Add the following line in the `fstab` file

```
UUID=5C24-1453 /mnt/mydisk fstype defaults,auto,users,rw,nofail 0 0
```

Replace `fstype` with the type of your file system, which you found above, for example: `ntfs`

- If the filesystem type is `FAT` or `NTFS`, add `,umask=000` immediately after `nofail` - this will allow all users full read/write access to every file on the storage device

```
UUID=5C24-1453 /mnt/mydisk fstype defaults,auto,users,rw,nofail,umask=000  
0 0
```

Now that you have set an entry in `fstab`, you can start up your Raspberry Pi with or without the storage device attached. Before you unplug the device you must either shut down the Pi, or manually unmount it using the steps in 'Unmounting a storage device' below.

**Note:** if you do not have the storage device attached when the Pi starts, the Pi will take an extra 90 seconds to start up. You can shorten this by adding `,x-systemd.device-timeout=30` immediately after `nofail`. This will change the timeout to 30 seconds, meaning the system will only wait 30 seconds before giving up trying to mount the disk.

```
UUID=5C24-1453 /mnt/mydisk fstype defaults,auto,users,rw,nofail,x-  
systemd.device-timeout=30 0 0
```

## Unmounting a storage device

When the Raspberry Pi shuts down, the system takes care of unmounting the storage device so that it is safe to unplug it. If you want to manually unmount a device, you can use the following command

```
sudo umount /mnt/mydisk
```

If no error was displayed, you can now safely unplug the device. If you receive an error that the 'target is busy', this means that the storage device was not unmounted. The 'target is busy' message means there are files on the storage device that are in use by a program. To close the files, use the following procedure

- Close any program which has open files on the storage device
- If you have a terminal open, make sure that you are not in the folder where the storage device is mounted, or in a sub-folder of it
- If you are still unable to unmount the storage device, you can use the `lsof` tool to check which program has files open on the device. You need to first install `lsof` using `apt`

```
sudo apt update  
sudo apt -y install lsof
```

To use `lsof`

```
lsof /mnt/mydisk
```

## Installing Python USB module

To be able to control your USB device you will need the Python USB Libraries. The PyUSB Libraries have been written in collaboration by a team of 'opensource' volunteers. To install it, type

```
sudo pip install pyusb  
sudo pip3 install pyusb
```