

RPI en mode périphérique USB

Table of Contents

- [RPI en mode périphérique USB](#)
 - [Pilotes de gadget](#)
 - [Gadget de stockage de masse](#)
 - [Configuration du magasin de sauvegarde](#)
 - [Configurer le zéro](#)
 - [Utilisation du module libcomposite](#)

Certains appareils disposent de contrôleurs de périphériques USB ou de contrôleurs à double rôle qui peuvent être utilisés en mode hôte ou en mode périphérique.

Cette page vise à documenter comment utiliser et configurer les périphériques gadgets USB OTG sous Linux lorsque l'on souhaite qu'un SBC connecté à un port hôte USB se comporte comme un « périphérique ».

Lorsqu'il est utilisé dans ce mode, l'appareil doit avoir un « pilote de gadget » chargé qui implémente la personnalité du type d'appareil souhaité.

Pilotes de gadget

Il existe plusieurs pilotes de gadgets Linux dans le noyau Linux actuel. Ceux-ci peuvent être trouvés dans le menu Pilotes de périphérique → Prise en charge USB → Prise en charge des gadgets USB :

- **g_zero (CONFIG_USB_ZERO)**
- **g_audio (CONFIG_USB_AUDIO)**
- **g_ether (CONFIG_USB_ETH)**: implémente un périphérique Ethernet « Communication Device Class » (CDC) pour créer une connexion réseau « USB vers Ethernet ».
- **g_ncm (CONFIG_USB_G_NCM)**: implémente la norme de sous-classe USB CDC NCM. NCM est un protocole avancé pour l'encapsulation Ethernet et permet de regrouper plusieurs trames Ethernet en un seul transfert USB.
- **g_mass_storage (CONFIG_USB_MASS_STORAGE)**: agit comme un pilote de disque de stockage de masse USB. Son référentiel de stockage peut utiliser un fichier normal ou un périphérique bloc spécifié comme paramètre de module ou option sysfs.
- **g_serial (CONFIG_USB_G_SERIAL)**: se comporte comme un périphérique série ACM pour créer une connexion « USB vers série » qui peut être utilisée pour interagir avec les hôtes MS-Windows ou avec le pilote Linux-USB « cdc-acm ».
- **g_midi (CONFIG_USB_MIDI_GADGET)**: agit comme un périphérique audio USB avec une entrée MIDI et une sortie MIDI.
- **g_printer (CONFIG_USB_G_PRINTER)**: canalise les données entre l'hôte USB et un programme en espace utilisateur pilotant le moteur d'impression.
- **g_cdc (CONFIG_USB_CDC_COMPOSITE)**: fournit deux fonctions dans une configuration : une liaison CDC Ethernet (ECM) et une liaison CDC ACM (port série)
- **g_acm_ms (CONFIG_USB_G_ACM_MS)**: fournit deux fonctions dans une configuration : un

périphérique de stockage de masse USB et une liaison CDC ACM (port série)

- **g_multi (CONFIG_USB_G_MULTI):** Un gadget composite multifonction pouvant fournir des interfaces Ethernet (RNDIS et/ou CDC), de stockage de masse et de liaison série ACM.
- **g_hid (CONFIG_USB_G_HID):** Un gadget HID (Human Interface Device) qui fournit une interface générique pour des éléments tels que les claviers, les souris et les écrans tactiles.
- **g_webcam:** Un périphérique webcam De plus, la prise en charge de Linux Configurationfs (CONFIG_CONFIGFS_FS) permet une configuration dynamique complète des périphériques gadgets à partir de l'espace utilisateur, auquel cas vous pouvez créer une configuration unique ou un périphérique composite multi-configuration avec une ou plusieurs des fonctions disponibles dans `drivers/usb/gadget/udc/functions`.



Un seul pilote de gadget (personnalité du périphérique) peut être chargé à la fois, mais certains pilotes de gadget « composites » se comportent comme des « périphériques composites », ce qui signifie qu'ils ont plusieurs points de terminaison par périphérique USB. Cela semblera familier si on pense au fonctionnement d'un téléphone intelligent moderne. Par exemple un téléphone Android: lorsqu'il est branché sur un PC hôte via micro-USB OTG, il se comportera comme un périphérique de stockage (MTP), mais si on souhaite qu'il se comporte comme un périphérique de débogage série, il faut aller dans le menu développeur et sélectionner cette option. Les smartphones modernes ne se comportent plus comme des périphériques de « stockage de masse USB », car ce protocole ne permet pas au système d'exploitation de l'appareil d'accéder au système de fichiers en même temps que le PC hôte. Au lieu de cela, ces appareils agissent comme un MTP (Media Transfer Protocol).



L'ID du fournisseur (VID) et l'ID de l'appareil (PID) présentés à l'hôte USB sont configurables

Gadget de stockage de masse

Le gadget de stockage de masse Présente un lecteur, une partition ou un fichier image à l'hôte USB.

Le gadget de stockage de masse ne peut pas être utilisé avec :

1. Tout modèle PI doté de plus d'un port USB intégré (sauf 4B).
2. Un hub USB en aval du zero.
3. Tout modèle Pi à port USB unique qui démarre à partir d'un périphérique USB5.

Le 4B ne prend en charge que l'exécution du gadget de stockage de masse sur son port USB C, mais il existe des problèmes potentiels avec ceci :

1. Si l'hôte USB ne peut pas fournir suffisamment de courant pour le 4B, il devra disposer du sien. Ceci peut être réalisé en l'alimentant via l'en-tête GPIO ou un câble Y approprié.
2. Lors de l'utilisation d'un bloc d'alimentation séparé pour le 4B et l'hôte USB, il existe un risque de rétroalimentation. l'hôte. Utiliser un câble ou un adaptateur USB modifié avec la ligne +ve

coupée pour éviter cela.

3. Bien que le connecteur soit USB C, le 4B fournit uniquement une connexion USB 2 et ne prend en charge l'USB PD.
4. Les câbles « marqués E » sont connus pour causer des problèmes avec les premières révisions 4B.



Pour disposer d'un stockage connecté via USB disposant d'un accès en écriture simultané et sécurisé des deux côtés de la liaison USB, utiliser le gadget Ethernet et exécuter un serveur samba sur le zéro.

Le gadget de stockage de masse fournit un périphérique de bloc de bas niveau à l'hôte USB. L'hôte accède au magasin de sauvegarde en demandant des lectures/écritures de N octets/blocs commençant à l'adresse A. Le système d'exploitation sur l'hôte USB traite ensuite ces données pour transmettre le fichier¹⁾ demandé à l'application qui en a besoin.

Le gadget de stockage de masse ne sait pas quels fichiers et répertoires sont consultés. Il n'a également aucune connaissance (et aucun moyen de savoir) quand un accès à un fichier donné est terminé.

L'écriture est sûre si seul le côté écriture a le magasin de support monté. Ce n'est pas sûr si les deux côtés l'ont montés et autorisent l'écriture et il n'est pas non plus sûr si un côté l'a monté en lecture/écriture et que les autres en lecture seule.

Les OS modernes utilisent à la fois la mise en cache en lecture et en écriture.

Les caches activés sur l'hôte USB et le zéro ne sont pas et ne peuvent pas être synchronisés.

Tandis que le gadget de stockage de masse sait quels blocs/octets ont été modifiés par l'hôte USB, il ne sait pas et ne peut pas savoir de quels fichiers ou répertoires ils font partie. Il n'a pas non plus connaissance si toutes les données à écrire ont été envoyées ou s'il y en a d'autres à venir.



Le gadget de stockage de masse ignore les modifications apportées par le zéro.

L'hôte USB n'est au courant d'aucune modification apportée par le zéro, le système d'exploitation sur le zéro n'est au courant d'aucun changement. modifications apportées par l'hôte USB.

En raison de ce qui précède, écrire d'un côté ou des deux côtés alors que l'autre est monté entraînera les problèmes comprenant :

1. Les fichiers supprimés d'un côté étant toujours présents de l'autre.
2. Les fichiers ajoutés d'un côté ne sont pas affichés de l'autre.
3. Les fichiers déplacés ou renommés d'un côté affichant toujours l'ancien nom ou l'emplacement de l'autre.
4. Le contenu du fichier est écrasé.
5. Corruption générale des fichiers ou du système de fichiers.



Le zéro est toujours un ordinateur entièrement Linux. Couper l'alimentation (par exemple en le débranchant du port hôte USB) sans effectuer au préalable un arrêt propre, cela entraînera une perte ou une corruption des données, à la fois au sein du magasin de sauvegarde et dans le stockage du système d'exploitation zéro.

Configuration du magasin de sauvegarde

Le magasin de sauvegarde du gadget de stockage de masse peut être l'un des suivants :

- Un volume entier, par ex. `/dev/sda`
- Une partition, par ex. `/dev/sda1`
- Un fichier, qui sera traité comme un périphérique bloc, par ex. `/srv/backing_store`

Ça ne peut pas être:

1. Un répertoire.
2. Un partage réseau monté à partir d'un autre ordinateur.

Utilisation d'un volume entier

L'hôte USB a accès à tout ce qui se trouve sur le périphérique, y compris son secteur de démarrage et ses tables de partition. S'il est exporté vers l'hôte USB en lecture/écriture, l'hôte peut détruire de manière triviale l'intégralité du contenu de l'appareil.

L'utilisation d'un périphérique entier avec un zéro n'est pas recommandée car cela donne à l'hôte USB l'accès au démarrage et les partitions racine, ce qui n'est pas conseillé dans la plupart des cas.

Utilisation d'une partition

L'utilisation d'une partition restreint l'accès de l'hôte USB à la seule partition spécifiée. Il n'a pas accès à d'autres parties du périphérique, y compris la table de partition et le secteur de démarrage du périphérique.

Puisqu'il n'y a pas d'accès à la table de partition du périphérique, l'hôte USB peut la considérer comme une table de partition nue, lecteur non initialisé. Le formatage de la partition via le système d'exploitation sur le zéro ne résoudra pas ce problème, les données exposées à l'hôte USB n'aura toujours pas de table de partition.

L'initialisation et le formatage de la partition à partir de l'hôte USB permettent d'écrire les données nécessaires sur la partition (table de partition, etc.) mais cela donne un lecteur avec deux tables de partition – une à l'emplacement normal et une au début d'une de ses partitions. Certains systèmes d'exploitation ne gèrent pas bien cela lors de la lecture directe plutôt que via le gadget de stockage de masse.

Il est préférable de le faire lorsque le système d'exploitation de la carte SD n'est pas en cours

d'exécution. Un deuxième ordinateur Linux (ou une deuxième carte SD avec Raspberry Pi OS installé) et un lecteur de carte USB seront nécessaires.

Les instructions détaillées dépendent des outils logiciels utilisés. Les grandes étapes sont les suivantes :

1. S'assurer que l'on travaille sur le bon appareil.
2. Réduire la taille de la deuxième partition pour créer la quantité d'espace libre requise après celle-ci prend fin.
3. Créer une nouvelle partition principale dans l'espace libre. Le laisser non formaté.
4. Enregistrer les modifications sur le disque.

Utilisation d'un fichier

L'utilisation d'un fichier restreint l'accès de l'hôte USB uniquement à ce fichier et à son contenu. Un fichier peut être d'une taille arbitraire ²⁾, dans un emplacement arbitraire et avoir un nom arbitraire. Les fichiers d'image disque (.img, .iso, etc.) peuvent être transmis tels quels au gadget de stockage de masse, mais à l'hôte USB doit comprendre les informations de partition et les systèmes de fichiers qu'elle contient pour qu'ils soient utiles.

Il existe trois méthodes ³⁾ principales pour créer un magasin de sauvegarde basé sur des fichiers :

1. `fallocate`
2. `dd`
3. `mkfs` (ou `mke2fs`, `mkdosfs`, etc.)

fallocate est rapide mais n'est pas pris en charge par tous les systèmes de fichiers.

dd est lent et pris en charge par tous les systèmes de fichiers. Le contenu du fichier dépend de ce qui a été utilisé comme if dans **dd**.

fallocate et **dd** produisent tous deux des fichiers non formatés, c'est-à-dire qu'ils ne contiennent pas de système de fichiers utilisable.

Avec certains types de systèmes de fichiers, **mkfs** peut créer le fichier et le système de fichiers qu'il contient en une seule opération. **mkfs** ne crée pas de secteurs de démarrage ni de tables de partition. De nombreux hôtes USB verront le magasin de sauvegarde comme non initialisé et non formaté. ⁴⁾

Pour créer un magasin de sauvegarde basé sur un fichier de 1 Go dans le répertoire actuel :

1. Utilisation de **fallocate** : `fallocate -l 1g fallocate.img`
2. Utilisation de **dd**: `dd if=/dev/zero of=dd.img bs=1M count=1K` **bs** est la taille du bloc/tampon, **count** est le nombre de blocs souhaités.
3. Utilisation de **mkfs**: `mkfs -t vfat -C mkfs.img 1048576` La taille est spécifiée en blocs. La taille des blocs peut varier selon les systèmes de fichiers. Certains permettent que ce soit spécifié.

Partitionnement et formatage du magasin de sauvegarde

Si le magasin de sauvegarde a été créé avec mkfs, cette étape peut être ignorée sauf si on veut contenir une table de partition.

La méthode la plus simple et la moins susceptible de provoquer des complications avec les hôtes USB consiste à passer le fichier de stockage de sauvegarde brut sur le gadget de stockage de masse, puis initialiser, partitionner et formater à partir de l'hôte.

Pour initialiser, partitionner et formater le magasin de sauvegarde sur le zéro :

1. Attacher le magasin de sauvegarde à un périphérique de boucle : `sudo losetup --show -fP /path/to/backing-store` 2. Noter le chemin renvoyé, par ex. `/dev/loop0`
2. Initialiser, partitionner et formater le magasin de sauvegarde à l'aide des outils Linux normaux (fdisk, gparted, etc.) sur le périphérique de boucle créé à l'étape précédente. Plusieurs partitions peuvent être utilisées mais tous les systèmes d'exploitation et hôtes USB ne le prennent pas en charge. Pour une utilisation multiplateforme, le meilleur choix de système de fichiers est actuellement FAT32.
3. Détacher le magasin de sauvegarde du périphérique de boucle : `sudo losetup -d /path/to/backing-store`

Rendre un magasin de sauvegarde basé sur des fichiers plus sécurisé

Cela s'applique uniquement à l'accès d'un utilisateur connecté au zéro et suppose que le magasin de sauvegarde est un fichier sur un système de fichiers natif Linux⁵⁾. Cela n'a aucun impact sur l'accès au contenu du backing store une fois qu'il a été monté sur le zéro ou l'hôte USB.



Les autorisations de fichier par défaut permettent à n'importe quel utilisateur de lire un fichier, à son propriétaire d'y écrire et à son propriétaire (ou root) pour le supprimer. Ce n'est probablement pas souhaitable, surtout si son propriétaire est un utilisateur normal. Les étapes suivantes peuvent améliorer ceci :

1. Modifier le fichier pour qu'il appartienne à root et qu'il ait un groupe de root : `sudo chown root:root /path/to/backing-store`
2. Modifier les autorisations pour que seul root y ait accès : `sudo chmod og-rwx /path/to/backing-store`
3. Rendre le fichier immuable⁶⁾ : `sudo chattr +i /path/to/backing-store`

Configurer le zéro

Cela peut être fait soit en se connectant au zéro, soit en montant la carte SD dans un lecteur connecté à l'hôte USB. Tous les chemins ci-dessous supposent une connexion au zéro, et doivent être ajustés lorsqu'on utilise cette méthode.

1. Sauvegarder le fichier `config.txt`: `sudo cp /boot/config.txt /boot/config.bak`
2. Ajouter ce qui suit à la fin de `/boot/config.txt`

```
[all]
```

```
dtoverlay=dwc2,dr_mode=peripheral
```

Au prochain démarrage, le système d'exploitation utilisera le pilote dwc2 dans le mode correct pour prendre en charge le fonctionnement en tant que gadget clé USB.

, `dr_mode=peripheral` est facultatif sur zéro, zeroW(H) et le 4B16. Il est obligatoire sur A, A+ et 3A+ car ils sont câblés en mode hôte et la broche d'identification n'est pas présente sur leurs connecteurs USB A.

1. Appliquer les étapes de la section 1
2. Ouvrir la **crontab** de root : `sudo crontab -e`
3. Ajouter ce qui suit à la fin du fichier : `@reboot /sbin/modprobe g_mass_storage file=/path/to/backing-store` Lorsqu'on utilise une partition, il est possible d'utiliser LABEL, PARTUUID, etc. en utilisant l'un des liens symboliques trouvés dans `/dev/disk/by-*` plutôt que `/dev/mmcblk0p`
4. Enregistrer et fermer.
5. Redémarrer

Cela donnera un gadget de stockage de masse avec les options par défaut : `rw=y`, `amovable=n`, `cdrom=n`.

- **cdrom=b[,b...]** Spécifie que le magasin de sauvegarde est présenté à l'hôte USB en émulant un CD-ROM. Lecteur ROM. `b` peut être « n », « N » ou « 0 » pour non ; « y », « Y » ou « 1 » pour oui. La valeur par défaut est non.
- **file=nom de fichier[,nom de fichier...]** Chemin d'accès au fichier ou au périphérique de bloc utilisé pour le magasin de sauvegarde. Plusieurs valeurs peuvent être spécifiées jusqu'au maximum par défaut de huit ⁷⁾. Si `amovable` n'est pas activé, un fichier doit être fourni. Chaque fichier sera présenté à l'hôte USB en tant que périphérique de stockage de masse distinct.
- **amovable=b[,b...]** Spécifie si un magasin de sauvegarde est présenté à l'hôte USB en tant que périphérique avec média amovible. `b` peut être « n », « N » ou « 0 » pour non ; « y », « Y » ou « 1 » pour oui. La valeur par défaut est non.
- **ro=b[,b...]** Spécifie si un magasin de sauvegarde est présenté à l'hôte USB en tant que périphérique en lecture seule. `b` peut être « n », « N » ou « 0 » pour non ; « y », « Y » ou « 1 » pour oui. La valeur par défaut pour les périphériques CD-ROM émulés est oui, pour tous les autres, non. Si le magasin de sauvegarde ne peut pas être ouvert pour l'écriture, le paramètre `ro` redeviendra sur `ro`.

D'autres paramètres peuvent être laissés à leurs valeurs par défaut et ne pas être spécifiquement inclus.

Utilisation du module libcomposite

1. Appliquer les étapes de la section 1
2. Devenir root : `sudo -je`
3. Charger **libcomposite** : `modprobe libcomposite`
4. Créer un répertoire pour la configuration du gadget : `mkdir -p /sys/kernel/config/usb_gadget/mygadget`
5. Passer dans ce répertoire : `cd /sys/kernel/config/usb_gadget/mygadget`
6. Définir les identifiants du gadget :
`echo 0x1d6b > idVendor`
`echo 0x0104 > idProduct`

- ```

echo 0x0100 > bcdDevice
echo 0x0200 > bcdUSB
7. Configurer les chaînes de texte :
mkdir -p chaînes/0x409
echo "1234567890" > strings/0x409/serialnumber
echo "me" > strings/0x409/manufacturer
echo "My USB Device" > strings/0x409/product
8. Configuration initiale de l'appareil :
mkdir -p configs/c.1/strings/0x409
echo "Config 1 : Stockage de masse" >
configs/c.1/strings/0x409/configuration
echo 250 > configs/c.1/MaxPower
9. Configurer le gadget de stockage de masse :
mkdir -p fonctions/mass_storage.usb0
echo 0 > fonctions/mass_storage.usb0/lun.0/cdrom
echo 0 > fonctions/mass_storage.usb0/lun.0/ro
echo /chemin/vers/backing-store > fonctions/mass_storage.usb0/lun.0/file
ln -s fonctions/mass_storage.usb0 configs/c.1/
10. Pour définir d'autre magasins, répéter les trois commandes d'echo ci-dessus en ajustant lun.0 (à
lun.1, lun.2 etc.)
11. Terminer8) : ls /sys/class/udc > UDC
12. Quitter le shell racine.

```

Pour automatiser cela, on peut le mettre dans un script<sup>9)</sup>



```

#!/bin/bash
modprobe libcomposite
mkdir -p /sys/kernel/config/usb_gadget/mygadget
cd /sys/kernel/config/usb_gadget/mygadget
echo 0x1d6b > idVendor
echo 0x0104 > idProduct
echo 0x0100 > bcdDevice
echo 0x0200 > bcdUSB
mkdir -p strings/0x409
echo "1234567890" > strings/0x409/serialnumber
echo "me" > strings/0x409/manufacturer
echo "My USB Device" > strings/0x409/product
mkdir -p configs/c.1/strings/0x409
echo "Config 1: Mass Storage" >
configs/c.1/strings/0x409/configuration
echo 250 > configs/c.1/MaxPower
mkdir -p fonctions/mass_storage.usb0
echo 0 > fonctions/mass_storage.usb0/lun.0/cdrom
echo 0 > fonctions/mass_storage.usb0/lun.0/ro
echo /path/to/backing-store >
fonctions/mass_storage.usb0/lun.0/file
ls /sys/class/udc > UDC

```

1. Enregistrer le script



2. Donner l'autorisation d'exécution: `chmod a+x /chemin/vers/script`
3. On peut l'appeler depuis la crontab de root, /etc/rc.local ou en tant que service systemd. </WRAP>

1)

Ou liste de répertoires, table de partition, secteur de démarrage, etc.

2)

Sous réserve de la taille du système de fichiers ou de la partition qui le contient et des limitations de ce système de fichiers.

3)

À l'exclusion de la prise d'image d'un lecteur/partition/système de fichiers existant.

4)

La plupart des hôtes USB Linux pourront le monter en utilisant le nœud de périphérique pour le lecteur, par ex. /dev/sda.

5)

Celui qui prend entièrement en charge le propriétaire, le groupe, les autorisations et les attributs Linux, par exemple ext4.

6)

Cela évite toute suppression accidentelle

7)

Le neuvième fichier et les fichiers suivants seront ignorés en silence.

8)

Une configuration supplémentaire du gadget doit être effectuée avant cette étape.

9)

Sauf les deux premiers et le dernier.

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=prive:rpi-usb-gadget>

Last update: **2025/02/19 10:59**

