

Utiliser USB en mode périphérique (OTG)

Table of Contents

- [Utiliser USB en mode périphérique \(OTG\)](#)
 - [Mode hôte USB](#)
 - [Mode périphérique USB](#)
 - [g_ether Gadget](#)
 - [g_mass_storage - Périphérique de stockage de masse USB](#)
 - [g_serial - Gadget de périphérique série](#)
 - [g_cdc - Ethernet composite + gadget série](#)
 - [g_multi - Ethernet composite + série + gadget de stockage de masse](#)
 - [g_hid - Gadget de périphérique d'interface humaine \(HID\)](#)
 - [g_webcam - Gadget de classe audio et vidéo USB composite](#)
 - [g_ncm - Gadget Ethernet de sous-classe USB CDC NCM](#)
 - [ConfigFs](#)
 - [Création d'un périphérique série CDC ACM](#)
 - [Création d'un périphérique Ethernet CDC ECM :](#)
 - [Création d'un périphérique de stockage de masse USB \(avec 2 LUN de 16 Mo chacun\) :](#)
 - [OpenWrt OTG](#)
 - [Sélection du mode OTG](#)
 - [Utilisation d'un port non OTG en mode appareil](#)

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 pour les cartes que vous souhaitez connecter à un port hôte **USB** et vous comporter comme un « périphérique ».



Les Newport SBC ne prennent pas en charge les modes OTG/Device/Gadget.

Mode hôte USB

En mode hôte permet de se connecter à un périphérique USB. Aucune configuration particulière n'est nécessaire pour cela.

Mode périphérique USB

Le mode périphérique permet la connexion à un hôte USB tel qu'un PC.

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.

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



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 peut s'apparenter 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 passer en mode développeur dans le menu système 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 appareil MTP (Media Transfer Protocol).



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

g_ether Gadget

Le pilote du gadget **g_ether** se comporte comme un dongle USB vers Ethernet. Une fois chargé, le système en mode périphérique ajoutera un périphérique réseau « usb<n> » qui peut être utilisé de la même manière que n'importe quel autre périphérique réseau. Sur le système hôte USB, un périphérique réseau similaire apparaîtra tant qu'un pilote prenant en charge la norme « CDC Ethernet » est disponible.

Ce module peut être construit avec un support supplémentaire :

- **EEM**: CDC Ethernet Emulation Model (EEM) est une norme plus récente dotée d'une interface plus simple qui peut être utilisée par davantage de matériel hôte USB.
- **RNDIS** : la prise en charge de RNDIS est une option supplémentaire (plus gourmande en CPU) qui peut être plus compatible avec les pilotes Windows.
- sur l'appareil cible (carte Gateworks avec contrôleur OTG) :

modprobe g_ether

- L'interface réseau usb0 apparaît sur la cible (à traiter comme n'importe quelle autre interface réseau)
- sur l'appareil hôte (c'est-à-dire PC), un appareil avec VID:PID 0525:a4a2 apparaîtra conformément à la norme Ethernet CDC
- **usb0** apparaît sur l'hôte Linux (en utilisant le pilote cdc_ether)
- les paramètres du module peuvent spécifier le VID, le PID, la version de l'appareil, la chaîne du fabricant, la chaîne du produit, le numéro de série
- les paramètres du module peuvent spécifier l'adresse Ethernet du périphérique et de l'hôte et s'il faut ou non utiliser le mode CDC EEM



Remarques sur l'hôte Linux : Le pilote cdc_ether prend en charge cela et créera un périphérique « usb<n> » sur l'hôte USB.



Remarques sur l'hôte Windows: le pilote g_ether est généralement construit avec la prise en charge RNDIS activée, ce qui le rendra compatible avec les pilotes de Windows7 et versions ultérieures qui apparaîtront dans le gestionnaire de périphériques en tant que « Gadget USB Ethernet/RNDIS » et peuvent être configurés comme n'importe quelle autre interface réseau.

g_mass_storage - Périphérique de stockage de masse USB

Le pilote **g_file_storage** se comporte comme un périphérique de stockage de masse USB tel qu'un disque dur USB ou une clé USB. Vous pouvez décider d'utiliser un « fichier » comme magasin de sauvegarde ou comme périphérique bloc (c'est-à-dire une partition flash ou un disque physique). Le fichier/périphérique est fourni au module via le paramètre de module 'file'.

Si on utilise un « fichier » de stockage de sauvegarde, il faut le créer au préalable avec la taille souhaitée. Par exemple, pour créer un magasin de sauvegarde de 64 Mo :

```
dd bs=1M nombre=64 si=/dev/zéro de=/backing_file
```

Pour l'utiliser comme magasin de sauvegarde :

```
modprobe g_mass_storage file=/backing_file
```

g_serial - Gadget de périphérique série

Le Serial Gadget prend en charge **CDC-ACM** et **CDC-OBEX** qui peuvent interagir avec les hôtes MS-Windows ou Linux en utilisant le pilote « cdc-acm » pour créer une connexion « USB-série ».

- sur l'appareil cible (carte Gateworks avec contrôleur OTG) :

```
modprobe g_serial
```

- sur le périphérique hôte (c'est-à-dire PC), un périphérique USB CDC ACM (VID:PID 0525:a4a7 par défaut) apparaîtra et se comportera comme un périphérique série
- Les paramètres du module peuvent spécifier le VID, le PID, la version de l'appareil, chaîne du fabricant, chaîne du produit, numéro de série
- les paramètres du module peuvent spécifier s'il faut ou non utiliser CDC ACM, CDC OBEX et le nombre de ports à créer



Remarques sur l'hôte Linux: le pilote cdc_acm énumérera ce périphérique comme '/dev/ttyACM<n>'

g_cdc - Ethernet composite + gadget série

Le gadget **g_cdc** prend en charge deux fonctions dans une seule configuration :

- une liaison CDC Ethernet (ECM) (connexion USB vers Ethernet)
- une liaison CDC ACM (port série) (connexion USB vers série)
- sur l'appareil cible (carte Gateworks avec contrôleur OTG) :

modprobe g_cdc

- sur le périphérique hôte (c'est-à-dire PC), un périphérique USB CDC ACM (VID: PID 0525: a4aa) apparaîtra



Remarques sur l'hôte Linux:

- le pilote cdc_acm énumérera ce périphérique comme '/dev/ttyACM<n>'
- le pilote cdc_ether énumérera ce périphérique comme périphérique réseau 'usb<n>'



Remarques sur l'hôte Windows:

- Un appareil CDC Composite Gadget apparaîtra dans le Gestionnaire de périphériques

g_multi - Ethernet composite + série + gadget de stockage de masse

Le gadget g_multi prend en charge plusieurs fonctions dans une seule configuration :

- une liaison CDC Ethernet (ECM)
- une liaison CDC ACM (port série)
- un périphérique de stockage de masse USB
- sur l'appareil cible (carte Gateworks avec contrôleur OTG) :

modprobe g_cdc

- sur le périphérique hôte (c'est-à-dire PC), un périphérique USB CDC ACM (VID:PID 1d6b:0104 par défaut) apparaîtra
- les paramètres du module peuvent spécifier le VID, le PID, la version de l'appareil, la chaîne du fabricant, la chaîne du produit, le numéro de série
- les paramètres du module peuvent spécifier le périphérique Ethernet, l'adresse de l'hôte et le multiplicateur de longueur de file d'attente à grande vitesse



Remarques sur l'hôte Linux:

- le pilote cdc_acm énumérera ce périphérique comme '/dev/ttyACM<n>'
- le pilote cdc_ether énumérera ce périphérique comme un 'usb<n>' périphérique réseau
- le pilote de stockage USB fournira la fonctionnalité de stockage de masse USB



Remarques sur l'hôte Windows:

- Un périphérique Gadget composite multifonction apparaîtra dans le Gestionnaire de périphériques

g_hid - Gadget de périphérique d'interface humaine (HID)

Le pilote du gadget **HID** fournit une émulation générique des périphériques d'interface humaine **USB (HID)**, par exemple des claviers, des souris, des écrans tactiles, etc.

- sur l'appareil cible (carte Gateworks avec contrôleur OTG) :

```
modprobe g_hid
```

- les paramètres du module peuvent spécifier le VID, le PID, la version de l'appareil, la chaîne du fabricant, la chaîne du produit, le numéro de série

g_webcam - Gadget de classe audio et vidéo USB composite

Le pilote du gadget **g_webcam** fournit un périphérique de classe audio et vidéo USB composite.

- sur l'appareil cible (carte Gateworks avec contrôleur OTG) :

```
modprobe g_webcam
```

- sur le périphérique hôte (c'est-à-dire un PC), un périphérique « Linux Foundation Webcam Gadget » (VID:PID 1d6b:0102 par défaut) apparaîtra
- sur le périphérique cible (carte Gateworks), un périphérique `/dev/video<n>` sera créé et disponible en tant que périphérique de sortie **Video4Linux** prenant en charge la vidéo 320/240 YUYV
- les paramètres du module peuvent spécifier le VID, le PID, la version de l'appareil, la chaîne du fabricant, la chaîne du produit, le numéro de série



Remarques sur l'hôte Linux: le pilote `uvcvideo` énumérera le périphérique et créera un périphérique de capture vidéo `/dev/video<n>`



Remarques sur l'hôte Windows:

- Un périphérique USB Composite apparaîtra dans le Gestionnaire de périphériques
- Un périphérique de caméra UVC apparaîtra sous Périphériques d'imagerie dans le gestionnaire de périphériques et sera disponible pour capturer des vidéos.

g_ncm - Gadget Ethernet de sous-classe USB CDC NCM

Le pilote de gadget **g_ncm** fournit une sous-classe **USB CDC NCM**. **NCM** est un protocole avancé pour l'encapsulation Ethernet, permettant le regroupement de plusieurs trames Ethernet en un seul transfert USB avec diverses possibilités d'alignement.

- sur l'appareil cible (carte Gateworks avec contrôleur OTG) :

`modprobe g_ncm`

- sur le périphérique hôte (c'est-à-dire un PC), un périphérique « Linux-USB Ethernet Gadget » (VID:PID 0525:a4a1 par défaut) apparaîtra
- sur le périphérique cible (carte Gateworks), un périphérique réseau USB<n> sera créé
- les paramètres du module peuvent spécifier le VID, le PID, la version de l'appareil, la chaîne du fabricant, la chaîne du produit, le numéro de série
- les paramètres du module peuvent spécifier les adresses Ethernet du périphérique et de l'hôte ainsi que le multiplicateur de longueur de file d'attente utilisé à grande vitesse



Remarques sur l'hôte Linux: le pilote `cdc_ncm` énumérera le périphérique et créera une interface réseau dans `/sys/class/net`



Remarques sur l'hôte Windows: Un appareil NCM Gadget apparaîtra dans le Gestionnaire de périphériques

ConfigFs

La prise en charge de Linux Configfs (**CONFIG_CONFIGFS_FS/**) permet une configuration dynamique complète des périphériques gadgets à partir de l'espace utilisateur, auquel cas on peut créer une configuration unique ou un périphérique composite multi-configuration avec une ou plusieurs des fonctions disponibles dans `drivers/usb/gadget/udc/functions`:

- **usb_f_acm**: CDC Serial (ACM - Modèle de contrôle abstrait)
- **usb_f_ecm**: CDC Ethernet (ECM - Modèle de contrôle de réseau Ethernet)
- **usb_f_eem**: CDC Ethernet (EEM - Modèle d'émulation Ethernet)
- **usb_f_fs**: Système de fichiers
- **usb_f_hid**: Interface HID
- **usb_f_mass_storage**: Classe de stockage de masse USB
- **usb_f_midi**: MIDI
- **usb_f_ncm**: Réseau CDC (NCM - Modèle de contrôle réseau Ethernet)
- **usb_f_obex**: CDC OBEX (modèle d'échange d'objets)
- **usb_f_phonet**: CDC Phonet
- **usb_f_printer**: Fonction d'imprimante
- **usb_f_rndis**: (Spécification de l'interface du pilote réseau distant - Microsoft Ethernet sur USB)
- **usb_f_serial**: Fonction série générique
- **usb_f_subset**: Sous-ensemble CDC (Ethernet sans mécanisme de contrôle - juste un transfert de données brutes)
- **usb_f_uac1**: Classe audio USB
- **usb_f_uac2**: Classe audio USB 2.0
- **usb_f_uvc**: Classe vidéo USB



Tous les modules de noyau ci-dessus peuvent ne pas être disponibles en fonction de la configuration de votre noyau ou de votre BSP.

Création d'un périphérique série CDC ACM

```
# configurations de montage
mount -t configfs none /sys/kernel/config
# charger le module libcomposite
modprobe libcomposite
# créer un gadget
mkdir /sys/kernel/config/usb_gadget/g1
# cd vers son nœud configfs
cd /sys/kernel/config/usb_gadget/g1
# le configurer (vid/pid peut être n'importe quoi si la classe USB est
utilisée pour la compatibilité du pilote)
echo 0xabcd > idVendor
echo 0x1234 > idProduct
# configurer son numéro de série/fabricant/produit
mkdir strings/0x409
echo myserial > strings/0x409/serialnumber
echo mymfg > strings/0x409/manufacturer
echo myproduct > strings/0x409/product
# créer une configuration
mkdir configs/c.1
# configurer avec des attributs si nécessaire
echo 120 > configs/c.1/MaxPower
# s'assurer que la fonction est chargée
modprobe usb_f_acm
# créer la fonction (le nom doit correspondre à un module usb_f_<name> tel
que 'acm')
mkdir functions/acm.0
# associer la fonction à la configuration
ln -s functions/acm.0 configs/c.1
# activer le gadget en le liant à un UDC depuis /sys/class/udc
echo 0000:01:00.0 > UDC
# pour le dissocier : echo "" UDC; sleep 1; rm -rf
/sys/kernel/config/usb_gadget/g1
```

Création d'un périphérique Ethernet CDC ECM :

```
# configurations de montage
mount -t configfs none /sys/kernel/config
# charger le module libcomposite
modprobe libcomposite
# créer un gadget
```



```
mkdir /sys/kernel/config/usb_gadget/g1
# cd vers son nœud configfs
cd /sys/kernel/config/usb_gadget/g1
# le configurer (vid/pid peut être n'importe quoi si la classe USB est
utilisée pour la compatibilité du pilote)
echo 0xabcd > idVendor
echo 0x1234 > idProduct
# configurer son numéro de série/fabricant/produit
mkdir strings/0x409
echo myserial > strings/0x409/serialnumber
echo mymfg > strings/0x409/manufacturer
echo myproduct > strings/0x409/product
# créer une configuration
mkdir configs/c.1
# le configurer avec des attributs si nécessaire
echo 120 > configs/c.1/MaxPower
# s'assurer que la fonction est chargée
modprobe usb_f_ecm
# créer la fonction (le nom doit correspondre à un module usb_f_<name> tel
que 'ecm')
fonctions mkdir/ecm.0
# associer la fonction à la configuration
ln -s fonctions/ecm.0 configs/c.1
# activer le gadget en le liant à un UDC depuis /sys/class/udc
echo 0000:01:00.0 > UDC
# pour le dissocier : echo "" UDC; sleep 1; rm -rf
/sys/kernel/config/usb_gadget/g1
```

Création d'un périphérique de stockage de masse USB (avec 2 LUN de 16 Mo chacun) :

```
# configurations de montage
mount -t configfs none /sys/kernel/config
# charger le module libcomposite
modprobe libcomposite
# créer un gadget
mkdir /sys/kernel/config/usb_gadget/g1
# cd vers son nœud configfs
cd /sys/kernel/config/usb_gadget/g1
# le configurer (vid/pid peut être n'importe quoi si la classe USB est
utilisée pour la compatibilité du pilote)
echo 0xabcd > idVendor
echo 0x1234 > idProduct
# configurer son numéro de série/fabricant/produit
mkdir strings/0x409
echo myserial > strings/0x409/serialnumber
echo mymfg > strings/0x409/manufacturer
echo myproduct > strings/0x409/product
# créer des configurations
```

```
mkdir configs/c.1
mkdir configs/c.2
mkdir configs/c.3
# les configurer avec des attributs si nécessaire
echo 120 > configs/c.1/MaxPower
echo 120 > configs/c.2/MaxPower
echo 120 > configs/c.2/MaxPower
# s'assurer que la fonction est chargée
modprobe usb_f_mass_storage
# créer la fonction (le nom doit correspondre à un module usb_f_<name> tel
que 'acm')
mkdir functions/mass_storage.0
# créer un ou plusieurs magasins de sauvegarde : dans cet exemple, 2 LUN de
16 Mo chacun
dd bs=1M count=16 if=/dev/zero of=/tmp/lun0.img # 16MB
dd bs=1M count=16 if=/dev/zero of=/tmp/lun1.img # 16MB
# associer aux partitions
mkdir functions/mass_storage.0/lun.0
echo /tmp/lun0.img > functions/mass_storage.0/lun.0/file
mkdir functions/mass_storage.0/lun.1
echo /tmp/lun1.img > functions/mass_storage.0/lun.1/file
# associer la fonction à la configuration
ln -s functions/mass_storage.0 configs/c.1
# activer le gadget en le liant à un UDC depuis /sys/class/udc
echo 0000:01:00.0 > UDC
# pour le dissocier : echo "" UDC; sleep 1; rm -rf
/sys/kernel/config/usb_gadget/g1
```

OpenWrt OTG

OpenWrt regroupe plusieurs des modules du noyau Linux ci-dessus sous forme de packages :

- **g_ether** - Modules du noyau → Prise en charge USB → kmod-usb-eth-gadget
- **g_mass_storage** - Modules du noyau → Prise en charge USB → kmod-usb-mass-storage-gadget
- **g_serial** - Modules du noyau → Prise en charge USB → kmod-usb-serial-gadget

La prise en charge des gadgets doit être activée :

- Modules du noyau → Prise en charge USB (kmod-usb-gadget)



Tous ces packages tenteront de charger automatiquement leur module de noyau respectif, donc celui qui apparaît en premier dans l'ordre alphabétique sera chargé. On peut voir ce qui est chargé en regardant les modules actuels :

```
lsmod | grep g_*
```

Sélection du mode OTG

Les contrôleurs hôtes USB OTG sont des contrôleurs « à double rôle » dans le sens où ils peuvent se comporter comme un hôte USB ou un périphérique USB. La décision sur le mode dans lequel il se trouve est généralement contrôlée par l'état de la broche d'identification OTG (**OTG_ID**) qui est mise à la terre sur OTG pour héberger les câbles et laissée flottante sur OTG pour les câbles de périphérique.

Dans certains cas, on peut avoir une carte sans signal **OTG_ID** sur laquelle on souhaite quand même utiliser USB OTG en mode périphérique.

Les contrôleurs à double rôle peuvent être forcés en mode « hôte » ou en mode « périphérique » via la propriété « **dr_mode** » de l'arborescence des périphériques. Par exemple, pour forcer un contrôleur **IMX6 OTG** en mode périphérique, il faut ajouter `dr_mode = "peripheral"` ; au **dt** tel que :

```
&usbotg {
    vbus-supply = <&reg_5p0v>;
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_usbotg>;
    disable-over-current;
    dr_mode = "peripheral";
    status = "okay";
};
```

Cela peut être fait dans le bootloader, voici un exemple pour GW54xx :

```
setenv fixfdt 'fdt addr ${fdt_addr}; fdt resize; fdt set /soc/aips-
bus@21000000/usb@2184000 dr_mode host' # host mode
setenv fixfdt 'fdt addr ${fdt_addr}; fdt resize; fdt set /soc/aips-
bus@21000000/usb@2184000 dr_mode gadget' # gadget mode
saveenv #une fois que vous avez fait votre sélection
```

De plus, certains contrôleurs hôtes tels que le contrôleur Chips and Media utilisé sur l'IMX6 ont des hooks qui leur permettent d'être configurés au moment de l'exécution dans l'espace utilisateur Linux. Par exemple sur les cartes IMX6 :

```
cat /sys/kernel/debug/ci_hdrc.0/role # voir le rôle actuel ; valeur par
défaut dictée par la propriété dr_mode dt ou l'état de la broche OTG_ID si
elle n'est pas définie
echo gadget > /sys/kernel/debug/ci_hdrc.0/role # spécifier le mode du
périphérique
echo host > /sys/kernel/debug/ci_hdrc.0/role # spécifier le mode hôte
```

Utilisation d'un port non OTG en mode appareil

Dans certains cas, on peut utiliser une prise USB Type-A en mode appareil de manière non standard.

Cela peut être utilisé par exemple sur des cartes qui acheminent un contrôleur USB OTG vers une prise Type-A qui n'a pas de signal **OTG_ID**. Dans ce cas, il faut soler **VBUS** pour s'assurer que l'hôte et l'appareil ne le pilotent pas en même temps. Cette opportunité existe sur certaines cartes Ventana (où le contrôleur hôte « usbotg » est activé dans l'arborescence des périphériques mais les signaux sont acheminés vers un connecteur de prise USB Type-A au lieu d'un contrôleur OTG).

Dans ce cas, on peut procéder comme suit :

- utiliser une fiche de type A non standard vers un câble de fiche de type A et isoler le **VBUS** (fil rouge) pour s'assurer que l'hôte et l'appareil ne conduisent pas tous les deux le **VBUS**
- configurer le contrôleur OTG pour le mode appareil (voir ci-dessus)
- Charger un pilote de gadget (voir ci-dessus)

From:

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

Permanent link:

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

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

