PiSO Mounter Setup and Use Guide

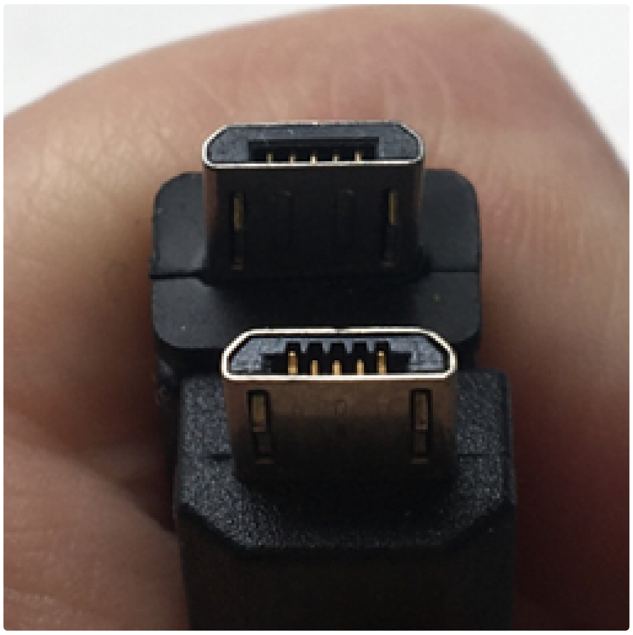
A PiSO mounter is a raspberry pi zero that emulates a usb drive. The purpose of the device is to automate testing involving .iso and .img images (such as memtest86) that must be physically inserted into systems. The device can dynamically load images to be inserted into the intended system and is compatible with the product applications testing framework.

This article details the steps to take to setup and use a PiSO Mounter Device

🔌 Equipment

Need:

- Raspberry Pi Zero W or Zero W 2
  - This is the best model; other models **might** work too.
- Micro-USB B to USB A cable (Must enable data transfer)
  - Top: Transfers Data in **some** cases
  - Bottom: Transfers Data in **all** cases



- Micro SD Card
- TP-Link Deco or equivalent

Optional:

- Setup tools:
  - Keyboard
  - Monitor
- Pi Zero W Case

📘 Setup Guide:

This guide is adapted from:

1. https://github.com/thagrol/Guides/blob/main/mass-storage-gadget.pdf - Connect your Github account
2. Raspberry Pi Zero Wireless – Smart USB Flash Drive [Samba Server] (geekering.com)

Setup

1. Set up TP Link Deco (or any other suitable device) in access point mode.
2. Connect Deco to Lab Network
3. Install Raspberry Pi OS Lite (32-bit) on the SD card. (Make sure to enable SSH, configure Wi-fi before boot to allow for headless setup.)
4. Connect the Pi to the Deco using the created username and password

Configuring Pi as a USB flash drive.

Enable the USB driver.

1. Edit the /boot/config.txt file:

```
sudo nano /boot/config.txt
```

2. and append this line to the bottom of the file:

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

Set a Static IP

1. Find an available static IP here
2. Edit /etc/dhcpcd.conf

```
sudo nano /etc/dhcpcd.conf
```

3. Add the following lines to the bottom of the file.

```
interface [INTERFACE]

static routers=[ROUTER IP]

static domain_name_servers=[DNS IP]

static ip_address=[STATIC IP ADDRESS YOU WANT]/24
```

4. The configuration should resemble the following:

```
interface wlan0

static routers=10.12.8.1

static domain_name_servers= 8.8.8.8

static ip_address=10.12.8.---/24
```

where ‘---’ corresponds to the IP you have chosen from the Vancouver lab network

5. Reboot to see changes

```
sudo reboot
```

Set up Boot Service

1. Enable execute permission on the script files

```
sudo chmod +x ...pisomounter/boot.sh
sudo chmod +x ...pisomounter/stop.sh
```

2. Create a system service files

```
sudo nano /etc/systemd/system/boot.service
```

3. Copy the following into the file

```
[Unit]
Description = Reboot Service
After=network.target

[Service]
WorkingDirectory=/home/asteralabs/pisomounter
ExecStart=/bin/bash /home/asteralabs/pisomounter/boot.sh
ExecStop=/bin/bash /home/asteralabs/pisomounter/shutdown.sh

[Install]
WantedBy=multi-user.target

```

4. Enable the service so it will run on boot

```
sudo systemctl enable myservice
```

💿 Using the Device

Connecting the USB drive

```
$ sudo modprobe g_mass_storage file=/path/to/image cdrom=_ removable=_ stall=_
```

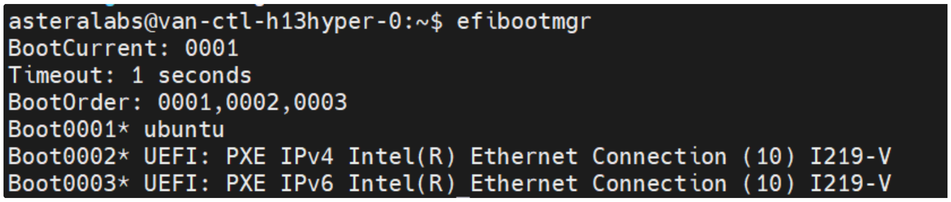
- There are three relevant specifications to consider: cdrom, removable, and stall.
- For booting into images, it is necessary to specify.
  - cdrom=1 removable=0 stall=0
- For inserting into a laptop
  - cdrom=0 removable =1 stall=0

Ejecting the USB drive

```
$ sudo modprobe -r g_mass_storage
```

Manipulating UEFI Boot Order

To change the boot order from the server’s command line, use efibootmgr.



```
asteralabs@van-ctl-h13hyper-0:~$ efibootmgr
BootCurrent: 0001
Timeout: 1 seconds
BootOrder: 0001,0002,0003
Boot0001* ubuntu
Boot0002* UEFI: PXE IPv4 Intel(R) Ethernet Connection (10) I219-V
Boot0003* UEFI: PXE IPv6 Intel(R) Ethernet Connection (10) I219-V
```

efibootmgr menu

- To select an entry to be booted into next:

  -
    ```
    sudo efibootmgr -n XXXX
    ```

where XXXX is the 4-number code after “Boot”