

Ubuntuphone - Tutorial : How to set up mtp, adb and ssh on Meizu MX4 Ubuntu Edition

Auteur : Winael

2015-07-07

Ubuntuphone - Tutorial : How to set up mtp, adb and ssh on Meizu MX4 Ubuntu Edition

One of the major benefits of Ubuntu Phones is the facility to easily take control of it through the abd protocol, or easily initiate SSH sessions into it. Unfortunately, the Meizu MX4 Ubuntu Edition is somewhat 'reluctant'! Fortunately, the 'fix' takes less time to set up than a TV ad.

Requirements

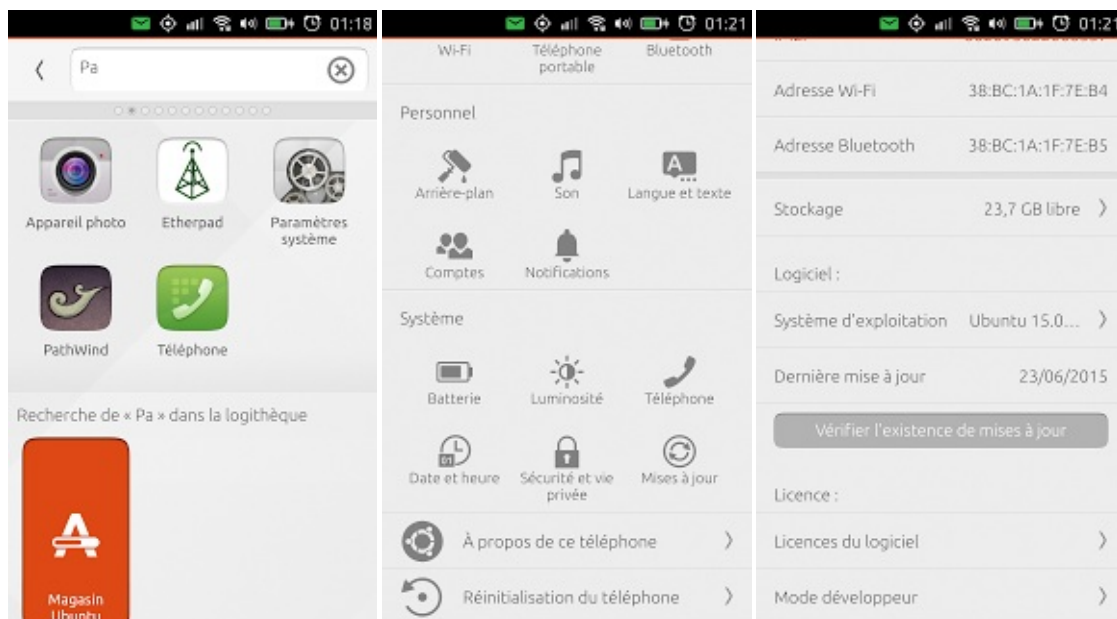
We'll begin by installing all the necessary elements for this tutorial:

```
$ sudo add-apt-repository ppa:ubuntu-sdk-team/ppa
$ sudo add-apt-repository ppa:phablet-team/tools
$ sudo apt-get update
$ sudo apt-get dist-upgrade
$ sudo apt-get install libmtp-common mtp-tools libmtp-dev libmtp-runtime libmtp9 ubuntu-sdk phablet-tools
```

Then we have to activate the debug mode on the phone. Go to:

System settings → About this phone → Developer Mode

and activate the developer mode.



< Mode développeur



En mode développeur, n'importe qui peut lire, modifier ou supprimer tout le contenu de ce téléphone en le connectant à un autre appareil.

Mode développeur



Vous devez avoir défini un code ou une phrase secrète pour utiliser le Mode Développeur

Sécurité du verrouillage



Setting up the MTP:

First, as Insiders, we have found that the Meizu phone was poorly recognized by our favorite OS. We were unable to mount the phone in mass storage mode, and so it's impossible to transfer files into it via USB using our SSH keys, or control it via adb.

In this case, the first thing to do is to interrogate lsusb:

With an unconnected phone:

```
$ lsusb
Bus 002 Device 003: ID 058f:6364 Alcor Micro Corp. AU6477 Card Reader Controller
Bus 002 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 006 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 005 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 005: ID 046d:c00c Logitech, Inc. Optical Wheel Mouse
Bus 001 Device 004: ID 046d:c31d Logitech, Inc. Media Keyboard K200
Bus 001 Device 003: ID 046d:0825 Logitech, Inc. Webcam C270
Bus 001 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 002: ID 03f0:8711 Hewlett-Packard Deskjet 2050 J510
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

With a connected phone:

```
$ lsusb
Bus 002 Device 003: ID 058f:6364 Alcor Micro Corp. AU6477 Card Reader Controller
Bus 002 Device 004: ID 2a45:0c02
Bus 002 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 006 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 005 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 005: ID 046d:c00c Logitech, Inc. Optical Wheel Mouse
Bus 001 Device 004: ID 046d:c31d Logitech, Inc. Media Keyboard K200
Bus 001 Device 003: ID 046d:0825 Logitech, Inc. Webcam C270
Bus 001 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 002: ID 03f0:8711 Hewlett-Packard Deskjet 2050 J510
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

It is reasonable to think that the line Bus 002 Device 004: ID 2a45: 0c02 matches our device.

Let's save the manufacturer's ID and Model ID in the variables VENDORID and MODELID respectively:

```
$ export VENDORID="2a45"
$ export MODELID="0c02"
```

FUSE can permit another user as root, to mount a third party filesystem. This requires that we uncomment the line user_allow_other in the file /etc/fuse.conf:

```
$ sudo sed -i 's/#user_allow_other/user_allow_other/' /etc/fuse.conf
```

Then we have to change the udev rules that support MTP in the file /lib/udev/rules.d/ 69-libmtp.rules order to declare the Meizu:

```
$ sudo bash -c 'cat << EOF >> /etc/udev/rules.d/69-libmtp.rules
# Meizu MX4
ATTR{idVendor}=="2a45", ATTR{idProduct}=="0c02", SYMLINK+="MeizuMX4Ubuntu", ENV{ID_MTP_DEVICE}="1", EN
EOF'
```

Normally, by plugging in your Meizu MX4 Ubuntu Edition, you should see it mounted automatically.

Setting up _{adb}:

It's possible to control the Meizu MX4 Ubuntu Edition through adb (Android Debug Bridge). To set it up, you just have to add the idVendor in the file \$HOME/.android/adb_usb.ini:

```
$ mkdir $HOME/.android
$ echo "0x2a45" >>> $HOME/.android/adb_usb.ini
```

Then we restart the adb service:

```
$ adb kill-server
```

The phone Meizu MX4 Ubuntu Edition is now recognized when we run the command:

```
$ adb devices
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
List of devices attached
75HACL66VJ8X    device
```

To access your device, you just run:

```
$ adb shell
phablet@ubuntu-phablet:~$
```

Setting up SSH:

Being able to manage our phone using the command-line is better. But the best – can we control it wirelessly? Well, it is possible, thanks to WiFi through our local network.

Keys exchange:

To secure the connection from the command-line between mobile devices running workstations Ubuntu, it is recommended to use phablet-shell, one of the tools that come from phablet-tools that can establish a SSH connection between devices. During the first launch, phablet-shell will handle the keys exchange, then connect via SSH on the mobile device

```
$ phablet-shell
/home/winael/.ssh/known_hosts updated.
Original contents retained as /home/winael/.ssh/known_hosts.old
Warning: Permanently added '[localhost]:2222' (RSA) to the list of known hosts.
Welcome to Ubuntu 15.04 (GNU/Linux 3.10.35+ armv7l)
```

```
* Documentation: https://help.ubuntu.com/
Last login: Sun Jun 28 14:44:03 2015 from localhost.localdomain
phablet@ubuntu-phablet:~$
```

Fixing IP adress

Now, it is recommended to fix the IP Address to always use the same address on a specific network.

From the Meizu MX4 Ubuntu Edition mobile device, we list all the current IPs for the phone:

```
$ sudo -i
[sudo] password for phablet:
# ip addr show wlan0
11: wlan0: mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 38:bc:1a:1f:7e:b4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.23/24 brd 192.168.1.255 scope global dynamic wlan0
        valid_lft 82074sec preferred_lft 82074sec
    inet6 fe80::3abc:1aff:fe1f:7eb4/64 scope link
        valid_lft forever preferred_lft forever
```

We must go to the /etc/NetworkManager directory and take a look at the content of our connection settings:

```
# cd /etc/NetworkManager/system-connections
# ls
# cat
```

We edit the connection:

First, we list all the connections setup:

```
# nmcli connection show
NAME                UUID                                  TYPE      DEVICE
Livebox-Navijobard  0c0a0040-ee0b-494c-a297-c5bfc2c1133c 802-11-wireless wlan0
AndroidAP           830dbbc7-8b8e-42be-a41c-9584988731b1 802-11-wireless --
NUMA Welcome        aafc220a-c594-44a4-9095-483fccddb66 802-11-wireless --
Nexus 5 Network     b7801b53-637f-4dde-a250-3479184ab3d7 bluetooth  --
/208150004998871/context1 9e611658-b475-7c4e-954b-edc06b75ab87 gsm        ril_0
NUMA Inside         baf064d2-588c-40aa-a291-1efb8779ec3a 802-11-wireless --
```

- We edit the connections that interester us

```
# nmcli connection edit Livebox-Navijobard
```

```
===| nmcli interactive connection editor |===
```


Editing existing '802-11-wireless' connection: 'Livebox-Navijobard'

Type 'help' or '?' for available commands.

Type 'describe [.]' for detailed property description.

You may edit the following settings: connection, 802-11-wireless (wifi), 802-11-wireless-security (wifi-sec), 802-1x, ip

- We print the actual configuration:

```
nmcli> print
=====
Connection profile details (Livebox-Navijobard)
=====
connection.id:           Livebox-Navijobard
connection.uuid:         0c0a0040-ee0b-494c-a297-c5bfc2c1133c
connection.interface-name: --
connection.type:         802-11-wireless
connection.autoconnect:  yes
connection.timestamp:    1436215756
connection.read-only:    no
connection.permissions:  --
connection.zone:         --
connection.master:       --
connection.slave-type:   --
connection.secondaries:  --
connection.gateway-ping-timeout: 0
-----
802-11-wireless.ssid:    Livebox-Navijobard
802-11-wireless.mode:    infrastructure
802-11-wireless.band:    --
802-11-wireless.channel: 0
802-11-wireless.bssid:   --
802-11-wireless.rate:    0
802-11-wireless.tx-power: 0
802-11-wireless.mac-address: 49:AB:3D:5G:6E:D4
802-11-wireless.cloned-mac-address: --
802-11-wireless.mac-address-blacklist:
802-11-wireless.mtu:     auto
802-11-wireless.seen-bssids: 7B:04:E7:AC:8A:FE,7B:04:E7:AC:8A:FF
802-11-wireless.hidden:   no
802-11-wireless.powersave: --
-----
802-11-wireless-security.key-mgmt: wpa-psk
802-11-wireless-security.wep-tx-keyidx: 0
802-11-wireless-security.auth-alg: open
802-11-wireless-security.proto:
802-11-wireless-security.pairwise:
802-11-wireless-security.group:
802-11-wireless-security.leap-username: --
802-11-wireless-security.wep-key0: --
802-11-wireless-security.wep-key1: --
802-11-wireless-security.wep-key2: --
802-11-wireless-security.wep-key3: --
802-11-wireless-security.wep-key-flags: 0 (none)
802-11-wireless-security.wep-key-type: 0 (unknown)
802-11-wireless-security.psk: --
802-11-wireless-security.psk-flags: 0 (none)
802-11-wireless-security.leap-password: --
802-11-wireless-security.leap-password-flags: 0 (none)
-----
ipv4.method:             auto
ipv4.dns:
ipv4.dns-search:
ipv4.addresses:
ipv4.routes:
```

```

ipv4.ignore-auto-routes:    no
ipv4.ignore-auto-dns:      no
ipv4.dhcp-client-id:       --
ipv4.dhcp-send-hostname:   yes
ipv4.dhcp-hostname:        --
ipv4.never-default:        no
ipv4.may-fail:             yes

```

```

ipv6.method:               auto
ipv6.dns:
ipv6.dns-search:
ipv6.addresses:
ipv6.routes:
ipv6.ignore-auto-routes:   no
ipv6.ignore-auto-dns:      no
ipv6.never-default:        no
ipv6.may-fail:             yes
ipv6.ip6-privacy:          -1 (unknown)
ipv6.dhcp-hostname:        --

```

- We have to set the IPv4 configuration:

```
nmcli> goto ipv4
```

You may edit the following properties: method, dns, dns-search, addresses, address-labels, routes, ignore-auto-routes

- We switch the IPv4 configuration to manual:

```
nmcli ipv4> set method manual
```

- We modify the IP address, the mask, and the gateway:

```
nmcli ipv4> set addresses 192.168.1.101/24 192.168.1.255
```

- We modify the DNS address:

```
nmcli ipv4> set dns 8.8.8.8 8.8.4.4
```

- We exit from the IPv4 configuration:

```
nmcli ipv4> back
```

- We activate the automatic connection:

```
nmcli> set connection.autoconnect TRUE
```

- We print the new settings:

```
nmcli> print
```

- - Save, and exit:

```
nmcli> save
```

```
nmcli> quit
```

We reboot the phone and take a look on its IP:

```
phablet@ubuntu-phablet:~$ ifconfig
```

```

ccmni0  Link encap:Ethernet  HWaddr fe:3f:26:a0:08:56
        inet addr:10.163.45.115  Mask:255.255.255.0
        UP RUNNING NOARP  MTU:1500  Metric:1
        RX packets:134 errors:0 dropped:0 overruns:0 frame:0
        TX packets:140 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:33299 (33.2 KB)  TX bytes:11014 (11.0 KB)

```

```
lo      Link encap:Local Loopback
```

```
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:492 errors:0 dropped:0 overruns:0 frame:0
TX packets:492 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:45396 (45.3 KB) TX bytes:45396 (45.3 KB)
```

```
wlan0 Link encap:Ethernet HWaddr 38:bc:1a:1f:7e:b4
inet addr:192.168.1.101 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fe80::3abc:1aff:fe1f:7eb4/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:8 errors:0 dropped:0 overruns:0 frame:0
TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:718 (718.0 B) TX bytes:692 (692.0 B)
```

The IP address is set up with the address that we fixed.

Now, we have just to set up the DNS resolution of our Internet box in the file `/etc/hosts`

Then we can just connect to the phone with the command:

```
$ ssh phablet@mx4ubuntu
```

You can add a bookmark to `sftp://phablet@mx4ubuntu:/home/phablet` in nautilus to mount your phone using WiFi. It will simplify your life for file transfers.

See you soon for more tips!!!

Thx KennedyM for the corrections :)

Table des Matières

Ubuntuphone - Tutorial : How to set up mtp, adb and ssh on Meizu MX4 Ubuntu Edition	1
Ubuntuphone - Tutorial : How to set up mtp, adb and ssh on Meizu MX4 Ubuntu Edition	2
Requirements	3
Setting up the MTP:	6
Setting up adb:	7
Setting up SSH:	8
Keys exchange:	8
Fixing IP adress	8