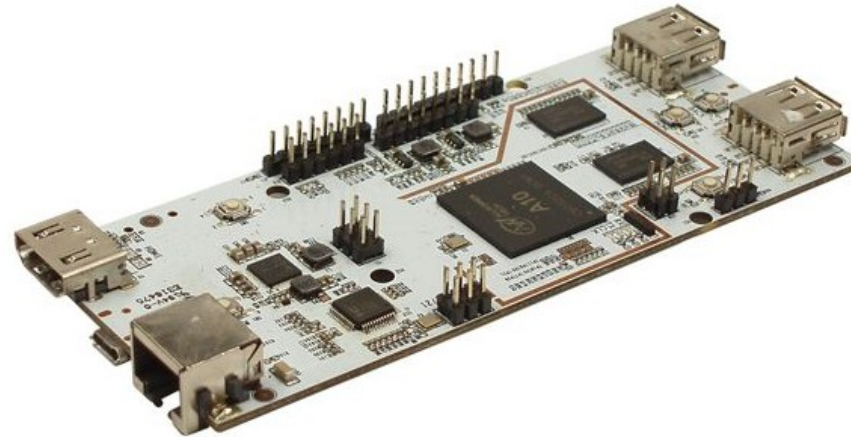


Le pcDuino : Apprendre à utiliser l'accès distant par VNC

Par X. HINAULT – Juin 2013 – www.mon-club-elec.fr

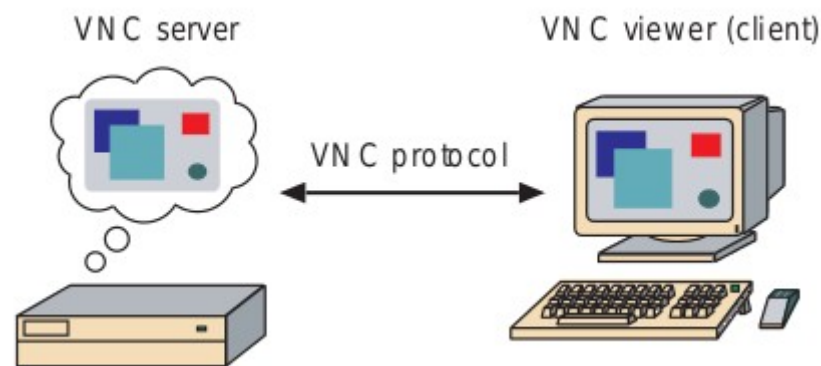


Introduction

- Une fois que vous avez activé l'accès au bureau distant du pcDuino (voir « Personnalisation du système de base »), il va être possible de prendre le contrôle à distance du pcDuino et de travailler dessus comme si vous y étiez à partir de votre bureau, le pcDuino pouvant se trouver dans votre cave, votre grenier, votre jardin ou sur un robot mobile !
- L'intérêt de cette façon de procéder est multiple :
 - vous disposez du retour vidéo si vous utilisez une webcam
 - vous pouvez vérifier que le programme (de serveur ou autre) en cours d'exécution sur le pcDuino fonctionne correctement
 - vous pouvez programmer le pcDuino à distance !
 - vous pouvez visualiser le résultat de mesures à distance,
 - etc..
- En un mot, vous pouvez faire tout ce que vous feriez si vous étiez face à votre PC, sauf que ça se passera à distance : cool non ?

Principe d'accès distant au bureau du pcDuino

Le pcDuino est une unité centrale de PC classique de taille réduite, mais sans écran, ni clavier, ni souris. Pour en prendre le contrôle, on va accéder à distance, via un réseau, au bureau graphique du pcDuino et y travailler comme si on se trouvait face au pcDuino. L'intérêt de procéder ainsi est de gérer le pcDuino à distance et même sans fil depuis son poste fixe. Ceci est très intéressant dans le cas d'un robot mobile, d'un système de mesure ou de capture vidéo d'accès difficile ou peu pratique, etc...



Principe de l'accès distant par VNC.

Le protocole VNC :

- est opensource
- est indépendant de la plateforme, ainsi on peut accéder au pcDuino depuis un MAC, Windows, Linux et même Android.
- Est peu sécurisé, ce qui est sans importance sur un réseau local dédié. Mais si on souhaite plus de sécurité, on pourra tunneller la connexion VNC par SSH (voir le tuto dédié à SSH pour le détail de la procédure)
- Supporte le retour vidéo !

Pré-requis

Si vous avez bien suivi la procédure décrite dans le tuto « personnalisation du système de base », le pcDuino est configuré en serveur VNC actif. Nous allons donc passer à l'action... Si ce n'est pas le cas, veuillez vous y reporter.

Le réseau utilisé

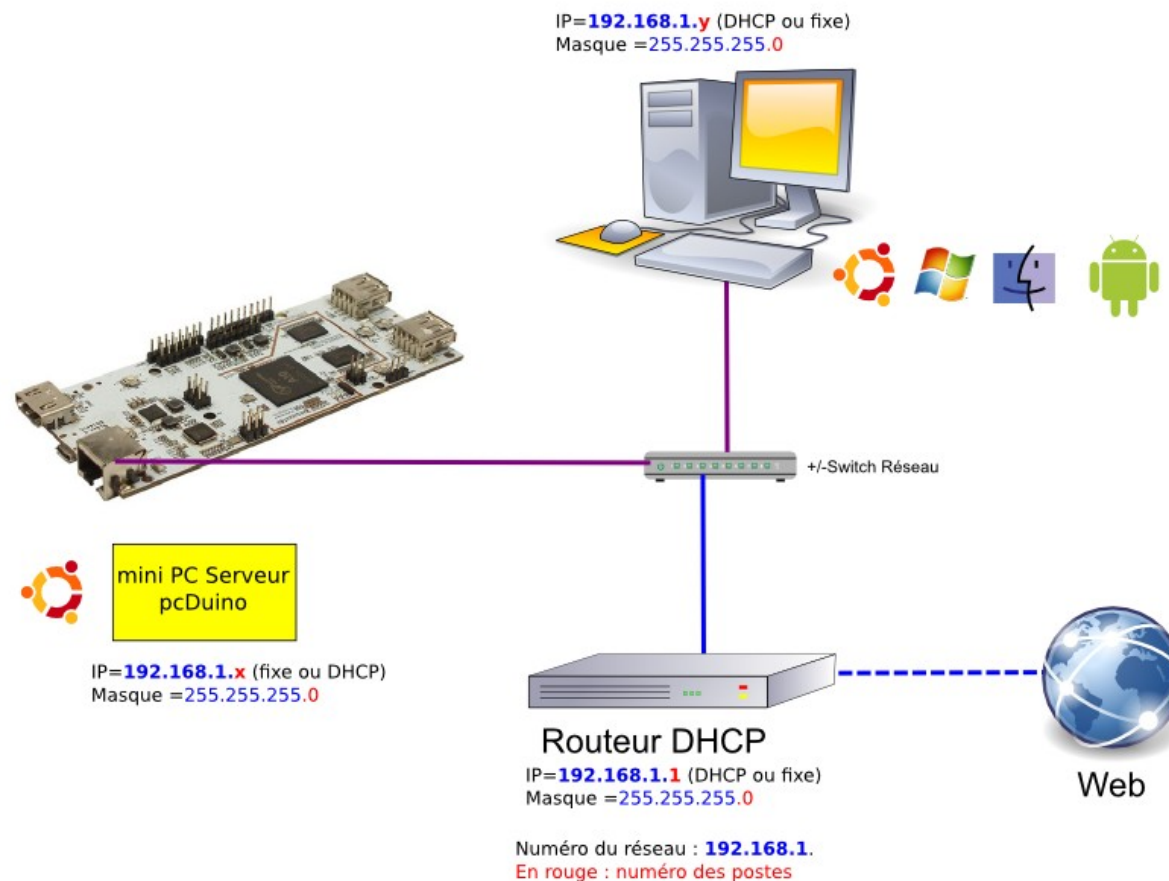
On considère ici que l'accès distant au bureau du pcDuino se fait par VNC (Virtual Network Computing) sur un réseau local disposant d'un routeur. Le réseau doit donc disposer au minimum :

- du **routeur du réseau** : soit une Box internet classique, soit un routeur dédié, wifi et/ou ethernet selon le mode de connexion utilisé.

- du **pcDuino** connecté au routeur soit en wifi, soit en ethernet filaire, (pour une première utilisation, utiliser une connexion ethernet filaire)
- du **poste fixe** (sous Gnu/Linux, Windows ou Mac Os X voire Android) permettant d'accéder au bureau du pcDuino par VNC.

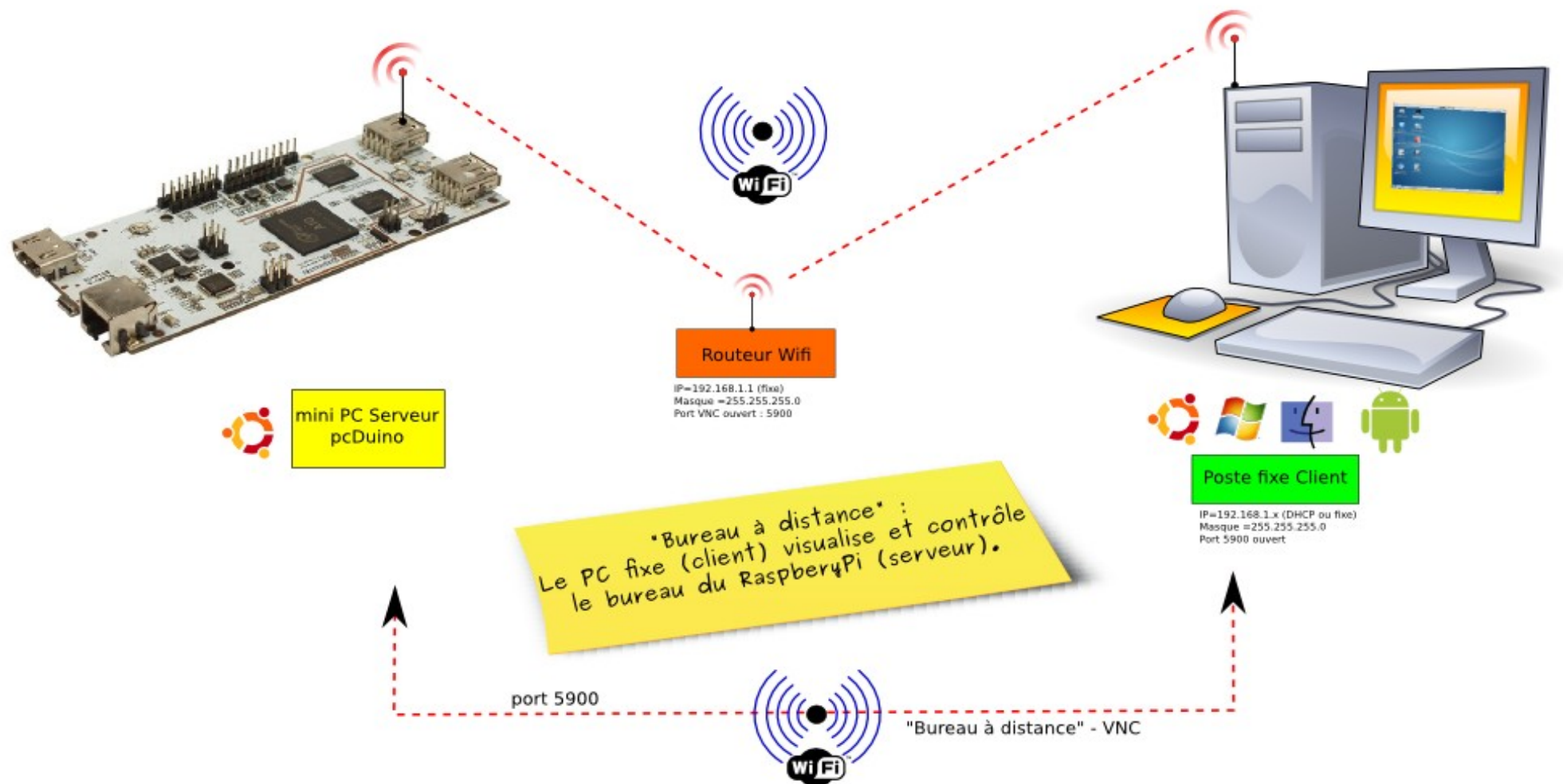
En mode filaire

- Voici le réseau de base que nous allons utiliser à la fois pour accéder au pcDuino à distance :



En mode « sans fil »

- Voici le réseau de vous pourrez utiliser au final à la fois pour accéder au pcDuino et le contrôler, le programmer, sans fil et à distance : à vous le retour vidéo sans fil, le contrôle de dispositifs ou les mesures analogiques sans fil, en toute simplicité ou presque !



IMPORTANT

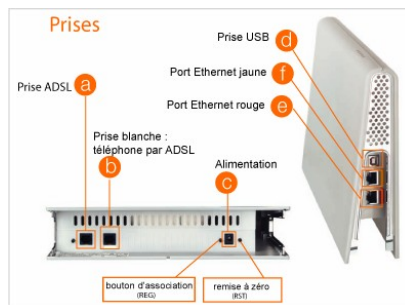
Les ports **22**(ssh), **111** (rpcbind), **2049** (nfs), **5800** (vnc-http) et **5900** (vnc) doivent être ouverts sur le routeur utilisé.

Les éléments du réseau local avec pcDuino que nous allons utiliser

Bien, à présent, trêve de « blabla », passons aux choses concrètes. Voyons à quoi va ressembler notre réseau.

Le premier élément du réseau : le routeur (ou une box).

Je me place ici dans le cas de figure courant de nos jours où vous disposez d'une box internet qui assure la double fonction de routeur et de modem. Les box intègrent la fonction DHCP d'attribution automatique des adresses IP. Votre box est le premier élément de votre réseau.



Si vous n'avez pas de box, vous devrez au moins disposer d'un routeur ethernet qui supporte la fonction DHCP, ce qui est le cas de la plupart des routeurs récents. Sur votre routeur, vous devez disposer d'au moins une prise ethernet RJ45, et idéalement 2.

+/- Elément complémentaire du routeur : le switch Ethernet

Si vous ne disposez que d'une prise Ethernet sur le routeur (c'est parfois la cas sur les box internet), vous devrez également utiliser un switch réseau qui est une sorte de multi-prises RJ45.



Le second élément de votre réseau : le poste fixe où vous travaillez disposant d'une carte réseau

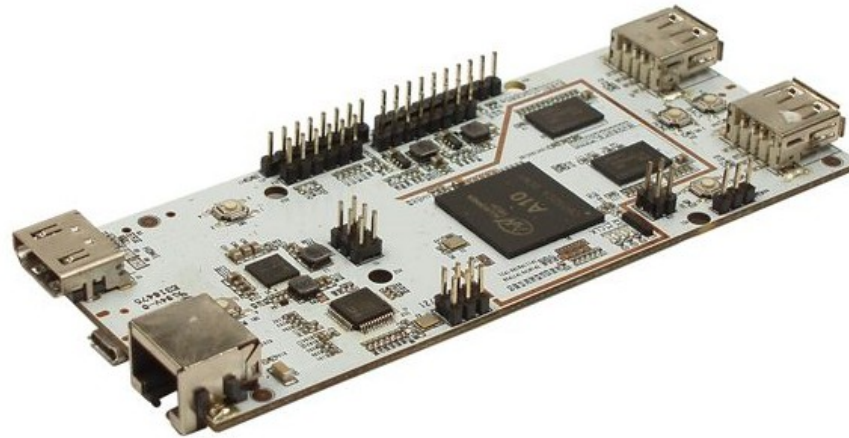
Le second élément du réseau est le poste fixe où vous travaillez. Ce poste doit disposer d'une interface Ethernet filaire (ou carte réseau),

L'autre possibilité est que le poste fixe se connecte au routeur par wifi, mais ce cas de figure n'est pas idéal ici car cela pourrait entraîner certains problèmes de connexion. Donc dans un premier temps au moins, utiliser de préférence une connexion Ethernet pour votre réseau.



Le troisième élément de votre réseau : le pcDuino opérationnel et « personnalisé »

Le troisième élément du réseau est évidemment le pcDuino + câble ethernet (impératif pour une première connexion) ou d'une clé USB wifi (l'autoconnexion sera fonctionnelle uniquement après une première connexion par ethernet)



Les adresses IP sur le réseau local utilisé

On considère ici que l'attribution des adresses sur le réseau se fait automatiquement par DHCP. **Le pcDuino est configuré par défaut pour fonctionner en DHCP.** Typiquement, sur un réseau avec une box internet, les adresses IP des différents postes seront :

- **192.168.1.1** pour le routeur,
- **192.168.1.x** pour le poste fixe, (attribuée automatiquement par le routeur DHCP)
- **192.168.1.y** pour le pcDuino, (attribuée automatiquement par le routeur DHCP)

où les 3 premiers groupes de chiffres représentent le numéro du réseau (**192.168.1**) et les chiffres **1**, **x** et **y** les numéros de chaque poste sur le réseau, le routeur

ayant typiquement toujours le n°1. Les numéros x et y sont attribués automatiquement par le routeur DHCP aux autres postes au moment de leur connexion. Pour être complet, on en déduit que le masque du réseau est **255.255.255.0**.

Truc à connaître :

Si on utilise à la fois des IP fixes et des IP attribuées automatiquement par DHCP sur le même réseau (si si, on peut !), cela est possible si les adresses IP fixes utilisées sont en dehors de la plage d'adresses utilisée par le routeur DHCP. A noter qu'il est également possible sur certains routeurs ou box d'attribuer une adresse IP précise à un matériel donné (défini par son adresse MAC). Pour savoir quelle plage d'adresse votre routeur utilise, s'y connecter depuis un navigateur avec l'adresse **192.168.1.1**

Si vous êtes vraiment perdus avec ces notions, vous pouvez consulter mon tuto Arduino dédié à la présentation du réseau qui est très détaillé. C'est ici :

http://www.mon-club-elec.fr/pmwiki_mon_club_elec/pmwiki.php?n=MAIN.ATELIERSReseauIntroReseauLocal

Découvrir les adresses IP utilisées sur le réseau local.

Une fois que le réseau est matériellement constitué et que tous les éléments sont sous tension, il devient possible de connaître les adresses IP de l'ensemble des postes du réseau. Cette procédure peut sembler « lourde » la première fois, mais on s'y habitue vite et ce n'est pas si sorcier que ça !

Déterminer l'adresse IP du poste fixe :

Tout d'abord, sur le poste fixe, il faut obtenir l'adresse IP qui lui a été attribuée automatiquement par le routeur DHCP. **Sur un poste Gnu/Linux**, ouvrir un Terminal et saisir la ligne de commande :

```
$ ifconfig
```

Une manipulation comparable permet de connaître l'adresse IP de son poste sous Windows et Mac. Voir notamment les liens suivants :

<http://assistance.orange.fr/connaitre-votre-adresse-ip-sur-mac-os-43.php> : menu **Pomme**, ouvrez **Préférences Système > Réseau > Airport > TCP/IP**

<http://assistance.orange.fr/connaitre-votre-adresse-ip-sur-windows-42.php> : "**Démarrer**", sélectionnez "**Exécuter...**". Dans le champ "**Ouvrir**" saisissez "**cmd**". Puis **OK**. Dans la console obtenue, taper l'instruction **ipconfig**

On obtient quelque chose de la forme :

```
eth0  Link encap:Ethernet  HWaddr xx:xx:xx:xx:xx:xx
inet adr:192.168.1.10  Bcast:192.168.1.255  Masque:255.255.255.0
adr inet6: fe80::221:85ff:fe3f:705b/64  Scope:Lien
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
Packets reçus:109 erreurs:0 :0 overruns:0 frame:0
TX packets:81 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 lg file transmission:1000
Octets reçus:15129 (15.1 KB) Octets transmis:16402 (16.4 KB)
Interruption:43 Adresse de base:0xc000
```

```
lo    Link encap:Boucle locale
inet adr:127.0.0.1  Masque:255.0.0.0
adr inet6: ::1/128 Scope:Hôte
UP LOOPBACK RUNNING MTU:16436 Metric:1
Packets reçus:2071 erreurs:0 :0 overruns:0 frame:0
TX packets:2071 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 lg file transmission:0
Octets reçus:91741 (91.7 KB) Octets transmis:91741 (91.7 KB)

ra0   Link encap:Ethernet  HWaddr xx:xx:xx:xx:xx:xx
inet adr:192.168.3.106 Bcast:192.168.3.255  Masque:255.255.255.0
adr inet6: fe80::9afc:11ff:febf:24c1/64 Scope:Lien
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
Packets reçus:64118 erreurs:0 :0 overruns:0 frame:0
TX packets:14347 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 lg file transmission:1000
Octets reçus:22328447 (22.3 MB) Octets transmis:1252998 (1.2 MB)
```

Sur ce poste, il y a 3 interfaces réseaux identifiées chacune par les lettres de la première colonne :

- l'interface eth0 correspondant à un réseau ethernet
- l'interface lo correspondant au réseau local sur le poste fixe lui-même,
- l'interface ra0 correspondant à un réseau wifi

L'adresse IP de chaque interface est de la forme xxx.xxx.xxx.xxx et est précédée de « inet adr : », qui ici nous donne :

- **192.168.1.10** pour l'interface réseau ethernet eth0
- 127.0.0.1 pour l'interface lo du réseau local du poste fixe
- 192.168.3.106 pour l'interface wifi ra0

Par ailleurs, remarquer que le masque réseau est bien de la forme 255.255.255.0 pour les 3 interfaces, donc seul le dernier chiffre indique le numéro du poste fixe sur chaque réseau.

Ainsi, le poste fixe est identifié avec :

- le numéro 10 sur le réseau **192.168.1.** auquel est connecté l'interface eth0
- le numéro **106** sur le réseau 192.168.3. auquel est connecté l'interface ra0

Dans votre cas, le numéro de réseau peut être différent, de la forme 192.168.0 ou 192.168.3 etc..
Idem pour votre poste fixe qui pourra avoir n'importe quel numéro entre 2 et 255.

Tester le réseau : Faire un « ping » vers le routeur depuis le poste fixe

Le principe

- Pour savoir si un poste est présent sur un réseau ou tester si le réseau fonctionne, le plus simple est de faire ce que l'on appelle un « ping » : on envoie des données vers le poste en question qui les renvoie si il est présent. C'est un peu comme si on envoie le ballon au routeur et qu'il nous le renvoie...

Procédure sous Gnu/Linux

- Sur un système Gnu/Linux (Xubuntu ou autre), les choses sont relativement simples :
 - ouvrir un terminal
 - saisir tout simplement :
 - la commande ping
 - suivi du paramètre -c 5 qui indique le nombre de test à faire
 - suivi de l'adresse à tester
- Ainsi, pour faire un ping vers le routeur depuis le poste fixe, on fera :

```
$ ping -c 5 192.168.1.1
```

- Si tout va bien, vous devez obtenir une série de messages indiquant les délais de réponse du routeur.

Procédure Mac et Windows

- La procédure est comparable sous Mac (Mac Os est un système Unix...) : ouvrir un terminal (applications > terminal) et saisissez la même commande.
- Sous Windows : "Démarrer", sélectionnez "Exécuter...". Dans le champ "Ouvrir" saisissez "cmd". Puis **OK**. Dans la console obtenue, taper l'instruction **ping 192.168.1.1**

A tester

Si votre routeur est connecté à internet, vous pouvez tester simplement votre connexion internet en faisant un ping vers google en faisant :

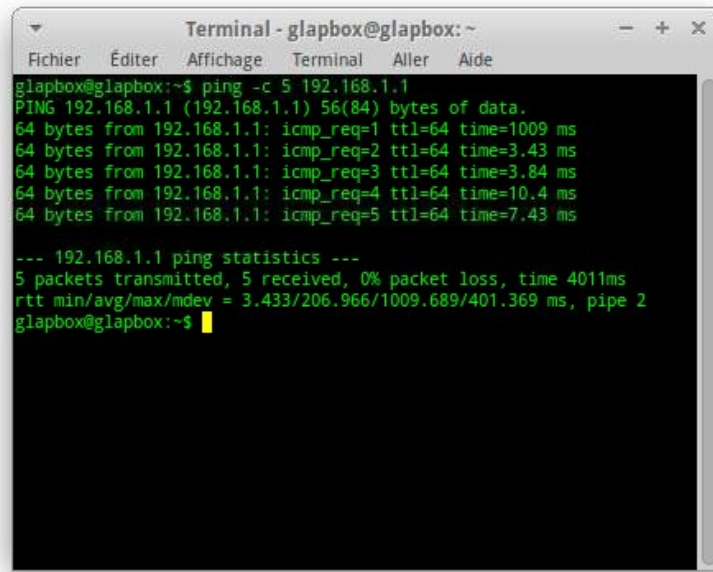
```
ping -c 5 www.google.fr
```

Vous devez obtenir une réponse avec les délais de réponse, ce qui vous confirme que votre connexion est opérationnelle.

A retenir

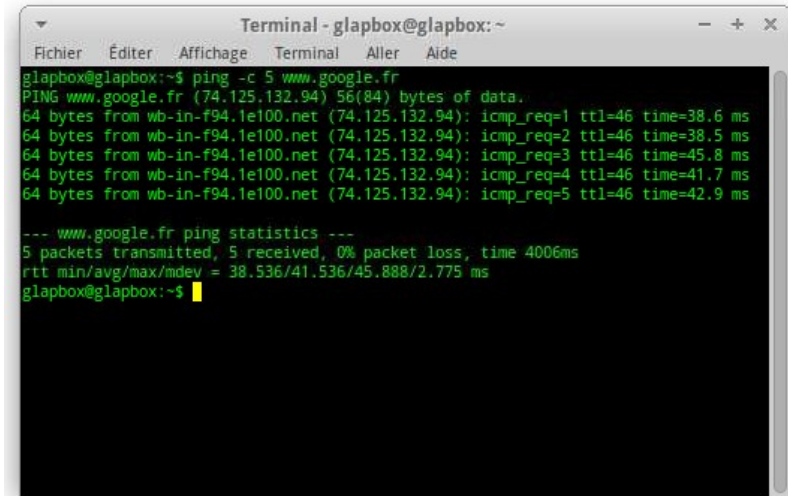
Par la suite, si votre réseau local rencontre un problème ou ne semble pas fonctionner, revenez à la base : faites un « ping » du poste fixe vers le routeur ou votre shield ethernet pour être sûr que tout fonctionne.

Ayez le réflexe « ping » !



```
Terminal - glapbox@glapbox: ~
Fichier Éditer Affichage Terminal Aller Aide
glapbox@glapbox:~$ ping -c 5 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data:
64 bytes from 192.168.1.1: icmp_req=1 ttl=64 time=1009 ms
64 bytes from 192.168.1.1: icmp_req=2 ttl=64 time=3.43 ms
64 bytes from 192.168.1.1: icmp_req=3 ttl=64 time=3.84 ms
64 bytes from 192.168.1.1: icmp_req=4 ttl=64 time=10.4 ms
64 bytes from 192.168.1.1: icmp_req=5 ttl=64 time=7.43 ms

--- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4011ms
rtt min/avg/max/mdev = 3.433/206.966/1009.689/401.369 ms, pipe 2
glapbox@glapbox:~$
```



```
Terminal - glapbox@glapbox: ~
Fichier Éditer Affichage Terminal Aller Aide
glapbox@glapbox:~$ ping -c 5 www.google.fr
PING www.google.fr (74.125.132.94) 56(84) bytes of data:
64 bytes from wb-in-f94.1e100.net (74.125.132.94): icmp_req=1 ttl=46 time=38.6 ms
64 bytes from wb-in-f94.1e100.net (74.125.132.94): icmp_req=2 ttl=46 time=38.5 ms
64 bytes from wb-in-f94.1e100.net (74.125.132.94): icmp_req=3 ttl=46 time=45.8 ms
64 bytes from wb-in-f94.1e100.net (74.125.132.94): icmp_req=4 ttl=46 time=41.7 ms
64 bytes from wb-in-f94.1e100.net (74.125.132.94): icmp_req=5 ttl=46 time=42.9 ms

--- www.google.fr ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 38.536/41.536/45.888/2.775 ms
glapbox@glapbox:~$
```

Scanner le réseau local et obtenir l'adresse IP du pcDuino

A ce stade, nous sommes un peu à l'aveugle sur le réseau :

- nous savons que nous sommes sur le réseau... « Ouah, l'exploit !!! .. Mais en fait, je sais que j'existe puisque je suis moi... »
- nous savons que le routeur est présent et répond, avec le n°1
- reste à savoir qui d'autre est présent, et notamment si le pcDuino est bien présent... et avec quel numéro... C'est un peu le jeu « colin-maillard » du réseau...

Une fois que l'on connaît le numéro du réseau et l'adresse du poste fixe, pour connaître l'adresse du pcDuino, le plus simple est de « scanner » le réseau à partir du

poste fixe. Ceci se fait sous Gnu/Linux, toujours sur le poste fixe, à l'aide d'un petit logiciel appelé **nmap** qui existe aussi en version graphique, appelé Zenmap. Pour la petite histoire, c'est ce logiciel qui est utilisé dans Matrix et autres films de « pirates informatiques », mais rassurez-vous, vous ne faites rien d'illégal en l'utilisant !



Donc, commencer par installer nmap, soit par Synaptic, soit en ligne de commande :

```
$ sudo apt-get install nmap
```

Une fois fait, on lance un « scan » du réseau où est connecté le pcDuino. Ici, nous considérons que le pcDuino est connecté sur le réseau **192.168.1.** à une adresse que nous cherchons à connaître. Afin de chercher tous les postes (numéro entre 1 et 255), et donc le pcDuino, connecté sur le réseau 192.168.1., on va donc saisir simplement la ligne de commande :

```
$ sudo nmap 192.168.1.1-255
```

On obtient quelque chose comme ça :

```
Starting Nmap 5.21 ( http://nmap.org ) at 2012-07-20 12:38 CEST
Nmap scan report for HSIB.home (192.168.1.1)
Host is up (0.016s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
23/tcp    open  telnet
80/tcp    open  http
443/tcp   open  https
992/tcp   open  telnets
8443/tcp  open  https-alt
MAC Address: xx:xx:xx:xx:xx:xx

Nmap scan report for xavier.home (192.168.1.10)
Host is up (0.0000070s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
```

```
80/tcp open http
111/tcp open rpcbind
2049/tcp open nfs

Nmap scan report for new-host-3.home (192.168.1.12)
Host is up (0.00021s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
111/tcp    open  rpcbind
2049/tcp   open  nfs
5800/tcp   open  vnc-http
5900/tcp   open  vnc
MAC Address: xx:xx:xx:xx:xx:xx
```

Ici, on retrouve :

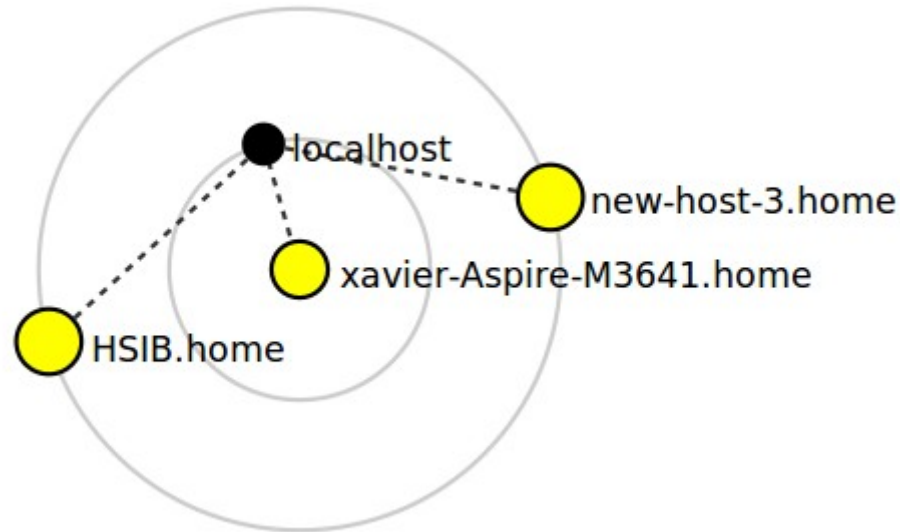
- le routeur avec l'adresse **192.168.1.1**
- le poste fixe avec l'adresse déjà obtenue précédemment : **192.168.1.10**
- et le 3ème poste qui est ici le pcDuino avec l'adresse : **192.168.1.12**

Si il y a « du monde » sur le réseau local, faire un premier « scan » avec le pcDuino éteint puis un second « scan » avec le pcDuino allumé : le poste en plus sur le réseau est le pcDuino.

Après une première connexion, en notant l'adresse MAC du pcDuino, on pourra facilement le reconnaître lors des connexions suivantes. On peut le reconnaître également à l'aide des ports ouverts...

Noter que nmap fournit également les ports ouverts pour chaque poste : on voit ici que sur le pcDuino, les ports 22 (sécurisation ssh), 111 et 2049 (partage de fichier nfs), 5900 (accès vnc) sont ouverts sur le pcDuino.

Pour info, Zenmap, l'implémentation graphique de nmap, fournit une visualisation graphique du réseau, ce qui est aussi sympa (onglet topologie) :



Se connecter par VNC au pcDuino

Une fois que l'on dispose de l'adresse IP du pcDuino, les choses sont simples. Il suffit de lancer le client VNC sur le poste fixe et de se connecter à l'adresse IP du pcDuino.

Côté pcDuino, il n'y a rien à faire : il est déjà paramétré en serveur VNC si vous avez bien suivi la procédure de personnalisation. Pour le détail de l'activation du serveur VNC sur le pcDuino, voir au besoin le document « Personnalisation du système de base ».

Sous Ubuntu, un client VNC léger et rapide est **xtightvncviewer**. Il y a d'autres alternatives, notamment graphiques, mais elles sont plus lentes à mon goût.

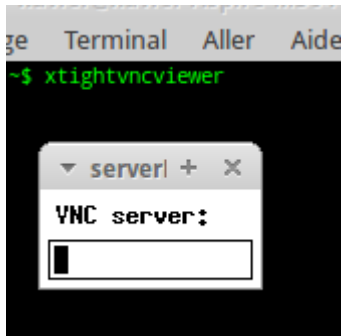
On commence par installer sur le poste fixe le programme `xtightvncviewer` si ce n'est déjà fait, soit avec Synaptic, soit en ligne de commande :

```
$ sudo apt-get install xtightvncviewer
```

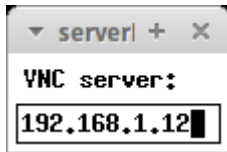
Ensuite, en ligne de commande, on lance le programme avec la commande :

```
$ xtightvncviewer
```

On obtient alors une petite fenêtre où il faut saisir l'adresse du pcDuino :



Saisir alors l'adresse IP du pcDuino et valider :



On obtient alors une seconde fenêtre (ça peut prendre quelques secondes, notamment par wifi..) où il faut saisir un mot de passe (celui que vous avez paramétré en configurant le pcDuino) puis valider. Noter qu'il est possible de modifier ce mot de passe voire même de le supprimer (mais ça devient risqué..).



Et là, MAGIE ! Vous devez voir après quelques instants le bureau de votre pcDuino : ça y est, vous pouvez travailler sur la pcDuino comme si vous étiez sur votre PC !



Stopper la connexion VNC

Pour mettre fin à la connexion VNC, c'est simple : il suffit de fermer la fenêtre VNC ! Par contre, votre pcDuino n'est pas éteint pour autant et continue de fonctionner dans l'état où vous l'avez quittée. Vous pourrez dès lors vous reconnecter de la même façon à tout moment.

Pour éteindre également le pcDuino, le faire avant de se déconnecter.

Optimisation de l'accès VNC

Améliorer la fluidité de la connexion VNC

Pour améliorer la fluidité de la connexion VNC, il est possible très simplement de le faire avec **xtightvncviewer** à l'aide de la commande suivante qui va fixer le taux de compression à utiliser :

```
$ xtightvncviewer -compresslevel 3
```

La valeur 3 donne en pratique de bons résultats sans trop dégrader la qualité de l'image.

Les autres possibilités pour améliorer la fluidité sont :

- réduire la taille de « l'écran » du pcDuino en 800x600 si ce n'est déjà le cas.
- Utiliser une connexion VNC non sécurisée par SSH (la sécurisation SSH ralentit la vitesse de transmission du réseau : entre rapidité et sécurité, il faut choisir)
- Utiliser une connexion wifi norme N à 300 Mb/s (contre 54 Mb/s pour la norme g classique).

Evaluer la bande passante du réseau lors de l'accès VNC

Pour avoir une idée précise de ce qui se passe sur le réseau lors de l'accès VNC, il est utile d'avoir une visualisation de la bande passante. On peut faire cela aussi bien sur le poste fixe que sur le pcDuino lui-même.

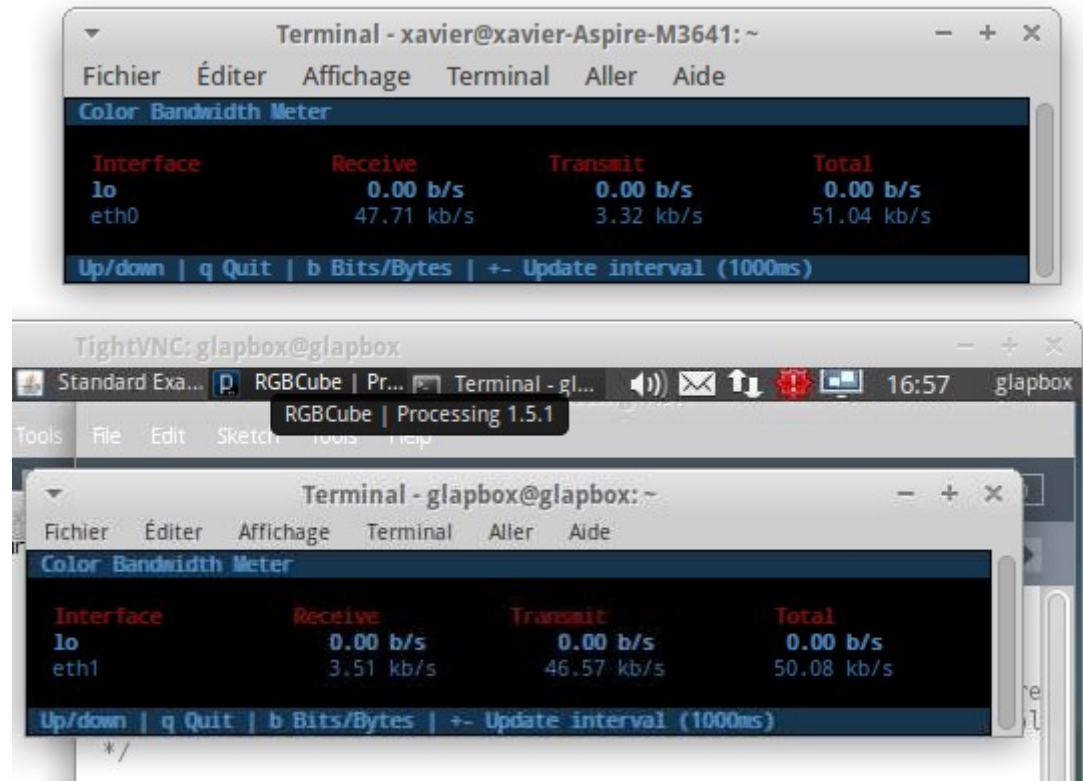
Sous Ubuntu, un utilitaire pratique qui fait cela est le programme en ligne de commande **cbm**. On commence par l'installer si ce n'est déjà fait soit sur le poste fixe, soit sur le pcDuino :

```
$ sudo apt-get install cbm
```

Ensuite, on lance le programme avec la commande :

```
$ cbm
```

On obtient un affichage de la bande passante en bits/secondes ou Bytes/secondes (appuyer sur B pour basculer d'une unité vers l'autre) pour chaque interface réseau. Voici un exemple simultané de **cbm** sur le poste fixe et le pcDuino :



On constate logiquement :

- que le débit global est sensiblement le même sur les 2 postes
- que le débit en émission d'un côté correspond au débit en réception de l'autre côté
- que VNC utilise une communication bi-directionnelle

Truc d'utilisation : réduire la fenêtre Terminal où s'exécute cbm et la mettre « toujours au premier plan » (clic droit sur la barre de fenêtre). Comme ça on visualise la bande passante en temps réel.

A titre indicatif, voici le chiffre de débit obtenu avec **xtightvncviewer** sans compression pour RGBCube dans Processing :

```

Terminal - xavier@xavier-Aspire-M3641: ~
Fichier Éditer Affichage Terminal Aller Aide
Color Bandwidth Meter
Interface      Receive      Transmit     Total
lo             0.00 b/s    0.00 b/s    0.00 b/s
eth0          16.97 Mb/s  531.36 kb/s 17.50 Mb/s
Up/down | q Quit | b Bits/Bytes | +- Update interval (1000ms)

```

A titre indicatif, voici le chiffre de débit obtenu avec **xtightvncviewer** avec compression 3 pour un exemple de programme dynamique (ce qui divise le débit par 2 !):

```

Terminal - xavier@xavier-Aspire-M3641: ~
Fichier Éditer Affichage Terminal Aller Aide
Color Bandwidth Meter
Interface      Receive      Transmit     Total
lo             0.00 b/s    0.00 b/s    0.00 b/s
eth0           5.66 Mb/s   255.37 kb/s 5.91 Mb/s
Up/down | q Quit | b Bits/Bytes | +- Update interval (1000ms)

```

Toujours sur un réseau en wifi N 300Mbps, voici le débit obtenu pour le même exemple de programme dynamique avec compression 3 :

```

Terminal - xavier@xavier-Aspire-M3641: ~
Fichier Éditer Affichage Terminal Aller Aide
Color Bandwidth Meter
Interface      Receive      Transmit     Total
lo             0.00 b/s    0.00 b/s    0.00 b/s
eth0           0.00 b/s    0.00 b/s    0.00 b/s
ra0           8.16 Mb/s   328.17 kb/s 8.49 Mb/s
Up/down | q Quit | b Bits/Bytes | +- Update interval (1000ms)

```

On obtient un gain de près de 50 % de la bande passante par rapport au réseau ethernet filaire.

Il y a d'autres programmes qui permettent cela : tout est question de goût.

Et aussi :

Sécuriser l'accès VNC par SSH

Noter qu'il est également possible de se connecter à la pcDuino en VNC de façon sécurisée avec tunnellation SSH : la procédure est décrite dans le document dédié au SSH. Cependant, cette sécurisation entraîne un ralentissement de la connexion et n'est utile qu'en cas de réseau ouvert. Sur un réseau local fermé dédié, et lorsqu'on recherche la vitesse de connexion en priorité, cette sécurisation n'est pas indispensable.

Activation d'une auto-connexion du pcDuino en wifi

Lors de la première connexion au pcDuino, il est indispensable de se connecter en Ethernet (filaire), seule façon d'avoir un accès automatique au réseau local.

Une fois cette première connexion faite, on se connectera au réseau wifi manuellement. (Voir tuto dédié)

Au prochain démarrage, la connexion wifi devrait être disponible automatiquement si le réseau est disponible. Si ce n'est pas le cas, reconnecter en filaire ethernet et vérifier ce qui ne va pas.

Dépannage de la connexion réseau du pcDuino

- Pour tout problème en wifi avec le pcDuino, avoir le réflexe de se connecter en réseau filaire Ethernet, solution de dépannage.
- Si on n'arrive pas également en Ethernet, on pourra essayer de simplement connecter un clavier au pcDuino et faire un appui sur entrée : ceci relance le démarrage lorsqu'il se bloque suite à un arrêt brutal par exemple.
- Si malgré tout, on n'obtient aucune connexion automatique, **il faut connecter un écran sur la sortie VGA + souris + clavier pour vérifier ce qui se passe... mais ceci ne doit quasiment jamais arriver à l'usage.**

Trucs pratiques d'utilisation de l'accès VNC à connaître

- Il est possible, par VNC, de copier/coller directement du texte depuis le poste fixe vers le pcDuino. Ainsi, par exemple, une ligne de commande dans un document sur le poste fixe pourra être copiée puis collée ensuite dans la fenêtre VNC. Ceci est très pratique. La même chose fonctionnera pour copier/coller un code Python, etc... (**à condition d'ouvrir le code depuis un éditeur texte simple, comme Gedit et pas depuis Geany, le copier/coller n'ayant pas l'air de fonctionner directement par VNC !**) offrant une souplesse de développement maximale. L'inverse ne semble pas fonctionner et il vaudra mieux se rabattre sur le partage de fichier NFS pour accéder aux fichiers voulus en lecture.