

Introduction au réseau local et apprendre à monter un réseau local avec Arduino.



Ateliers Arduino

par X. HINAULT

www.mon-club-elec.fr



Tous droits réservés – 2012.

Ce document légèrement payant est soumis au droit d'auteur et est réservé à l'usage personnel.

Afin d'encourager la production de supports didactiques de qualité, ce document est légèrement payant.

La licence d'utilisation est attribuée pour un usage personnel uniquement, dans le cercle familial. Mise en ligne et diffusion non autorisées.

Si vous n'êtes pas le détenteur de la licence attribuée pour l'usage de ce document, soyez sympa, merci d'acheter votre exemplaire personnel ici : <https://monclubelec.dpdcart.com/>

Pour tout problème lié à l'utilisation de ce document, veuillez envoyer une copie ici : support@mon-club-elec.fr

Pour obtenir tout autres types de licence d'utilisation (enseignement, commercial, etc...), veuillez contacter l'auteur ici : support@mon-club-elec.fr

Vous avez constaté une erreur ? une coquille ? N'hésitez pas à nous le signaler à cette adresse : support@mon-club-elec.fr

Truc d'utilisation : visualiser ce document en mode diaporama dans le visionneur PDF. Navigation avec les flèches HAUT / BAS ou la souris.

En mode fenêtre, activer le panneau latéral vous facilitera la navigation dans le document. Bonne lecture !

Lancer également le logiciel Arduino et connecter votre carte Arduino afin de pouvoir tester au fur et à mesure les codes d'exemples !

1. Intro

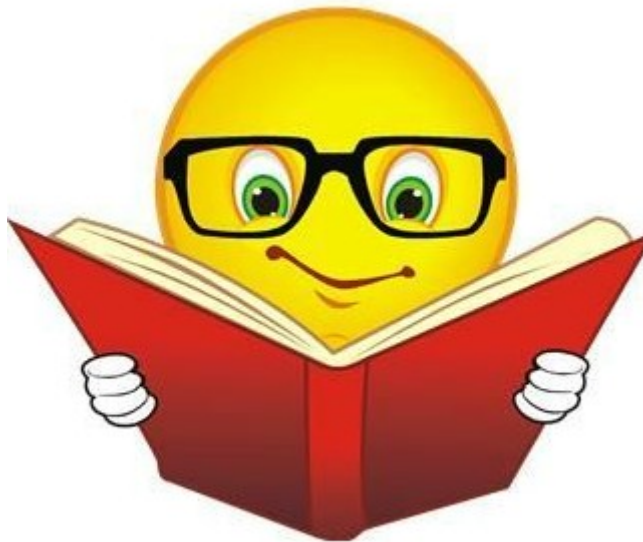
L'objectif ici est :

- de comprendre le principe d'un réseau local, le concept d'adresse IP
- d'apprendre à constituer et monter un réseau local de base
- d'apprendre à récupérer les informations importantes d'un réseau local
- d'apprendre à tester le bon fonctionnement du réseau
- de découvrir la librairie Ethernet
- d'apprendre à écrire le programme de base permettant d'utiliser Arduino sur un réseau local

... afin d'être en mesure d'utiliser le couple shield Ethernet + carte Arduino sur un réseau local.

Remarque

Dans cet atelier les notions abordées vont être nombreuses, notamment en ce qui concerne le réseau local. Le sujet est passionnant mais aussi potentiellement déroutant pour le néophyte. Je vais tenter de vous présenter tout ça le plus simplement possible, mais soyez prévenus : il va falloir prendre son temps pour tout bien comprendre, surtout si les réseaux sont nouveaux pour vous. La bonne nouvelle : à la fin de cet atelier, vous serez capable d'utiliser Arduino sur un réseau Ethernet (par fils) local ! Une fois que vous saurez le faire en mode Ethernet filaire, vous pourrez facilement transposer en wifi (sans fil).

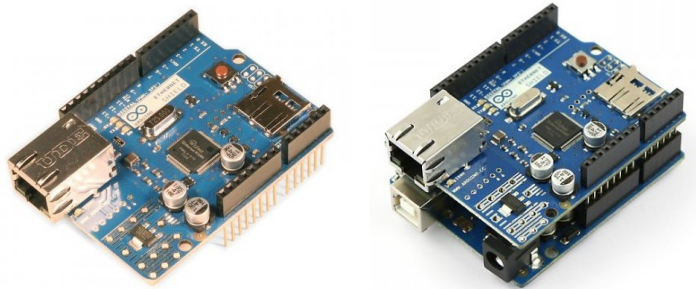


Prêt ? C'est parti !

2. Matériel utile ou nécessaire pour cet atelier

Pour cet atelier vous aurez besoin :

+/- D'une carte d'extension (shield) Ethernet (pas indispensable dans une première approche)



La carte d'extension (ou shield) ethernet Arduino est une carte électronique enfichable broche à broche sur la carte Arduino et qui permet d'utiliser Arduino sur un réseau ethernet local voire même sur internet.

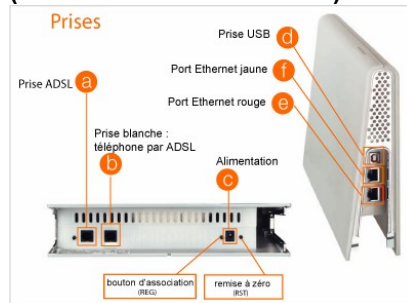
Ce shield utilise la **communication SPI** (broches 13,12,11, et 10 +/- 4) pour communiquer avec Arduino.

Ce shield intègre également un emplacement pour **carte mémoire SD** pour des stockage de données ou de pages HTML locales.

Ne pas confondre ce shield avec la carte UNO Ethernet qui est une variante d'une carte UNO avec ethernet intégré.

disponible chez : <http://snootlab.com> | 33€ environ

D'un routeur Ethernet (ou d'une « box » internet)



Le routeur est un élément central du réseau qui permet de réaliser simplement un réseau local avec plusieurs postes. Ce routeur devra être de type Ethernet (réseau par fil) : si votre routeur supporte aussi le wifi, tant mieux, mais ça ne vous servira à rien ici. Votre routeur devra disposer de la fonction d'attribution automatique des adresses (ou DHCP), ce qui est le cas dans la grande majorité des cas.

A noter qu'une box internet est un routeur Ethernet (associé à un modem ADSL) et pourra ici être utilisée.

Ce routeur devra disposer d'au moins une prise réseau libre RJ45.

+/- d'un switch Ethernet (si le routeur n'a pas au moins 2 prises Ethernet libres)



Si votre routeur ne dispose que d'une seule prise RJ45, il faudra probablement que vous utilisiez également un switch réseau qui est une sorte de « mult prises » RJ45.

Bien qu'il ne soit pas toujours indispensable, je vous conseille fortement de disposer d'un switch car ce n'est pas cher (on en trouve à 10€) et ça vous permettra d'ajouter facilement des postes sur votre réseau.

3. Matériel spécifique nécessaire pour cet atelier (suite)

D'un ou plusieurs câbles réseau dit « RJ45 »



Ou

Pour connecter les éléments du réseau Ethernet entre eux, vous devrez disposer d'au moins 2 câbles réseaux RJ45 (modèle classique, pas « croisé ») :

- 1 pour connecter votre PC au routeur
- 1 pour connecter le shield Ethernet au routeur

A moins que vous ayez l'intention de mettre votre carte Arduino loin de votre poste fixe, vous pouvez utiliser des câbles courts de 1m par exemple.

Noter qu'il existe des câbles RJ45 de petite longueur sur petit enrouleur :

Purchased by Franck Ourion, franck.ourion@univ-lorraine.fr #6280170

Atelier Arduino : Introduction au réseau local et apprendre à monter un réseau local avec Arduino.

pratiques pour réduire l'encombrement !

Conseil d'ami : ne pas hésiter à avoir quelques câbles ethernet d'avance sous le coude...

+/- de 2 blocs CPL (seulement si vous souhaitez déployer le réseau Ethernet via le réseau électrique 220V)



Les blocs CPL (technologie à courant porteur) permettent assez facilement de déployer un réseau Ethernet sur le circuit 220V domestique, avec une portée de 200m sans difficulté.

Vous aurez besoin de cet équipement si vous souhaitez créer un réseau entre Arduino + shield Ethernet et votre poste fixe dans des pièces différentes par exemple.

Cet équipement un peu plus coûteux (compter 40€ pour un bloc de qualité) n'est pas indispensable dans une première approche. Mais sachez que ça existe.

A titre indicatif : j'utilise et je conseille les blocs Delovo AvPlus 200, qui disposent d'une prise terre en façade, sont faciles à utiliser, sont robustes au quotidien et sont livrés avec un utilitaire Linux pour la configuration.

Et d'un ou plusieurs postes fixes (PC, Mac, Netbook,...) disposant d'une carte Ethernet ou wifi !



Je pense que c'est évident, mais je préfère quand même le dire... Vous avez besoin d'un poste fixe disposant d'une carte réseau Ethernet. Celui où vous lisez cette page et avec lequel vous programmez votre carte Arduino devrait faire l'affaire.

Votre poste peut-être sous Windows, Mac OsX ou Gnu/Linux, peu importe. Vous pouvez utiliser indifféremment un PC de bureau, un netbook ou un portable.

4. Un réseau local : ... c'est un peu comme une équipe de foot !

C'est quoi un réseau informatique ?

- Un réseau informatique est un ensemble d'ordinateurs ou équipements électroniques qui sont connectés entre eux et qui vont communiquer entre-eux.
- Pour dire les choses encore plus simplement, un réseau, c'est une « équipe » d'ordinateurs.

Dans une équipe de foot :

- Tous les joueurs qui ont la même couleur de maillot appartiennent à la même équipe : tous les bleus sont de la même équipe, tous les blancs d'une autre équipe, etc...
- Au sein de l'équipe, chaque joueur est identifié par son numéro : il n'y a qu'un seul joueur ayant un numéro donné.



Au sein d'un réseau local... :

- Tous les ordinateurs d'un même réseau local (= les joueurs de la même équipe) ont le même numéro de réseau (= la couleur du maillot). C'est grâce à ce numéro que des ordinateurs connectés entre eux savent qu'ils appartiennent au même réseau, à la même « équipe ».
- Chaque ordinateur d'un réseau local a un numéro qui permet de l'identifier (= le numéro sur le maillot) et il n'y a qu'un seul ordinateur du réseau qui a un numéro donné.

Notion d'adresse IP : le maillot coloré avec son numéro

- Evidemment, dans les faits, on ne met pas un « maillot » en tissu sur son ordinateur... mais on lui attribue un « maillot numérique » que l'on appelle l'adresse IP.
- Une adresse IP est composée de 4 valeurs séparées par des points. Chaque valeur correspond en fait à un octet (ensemble de 8 bits) et peut donc prendre une valeur entre 0 et 255. Quelques exemples d'adresse ip :
 - 192.168.3.4
 - 145.237.1.25
- En fait, l'adresse IP contient 2 informations essentielles :
 - le **numéro du réseau** auquel appartient la machine (c'est l'équivalent de la « couleur du maillot » de l'équipe)
 - le **numéro de la machine** sur le réseau (c'est l'équivalent du numéro sur le maillot)



Si vous avez compris ça, vous avez compris un élément essentiel !

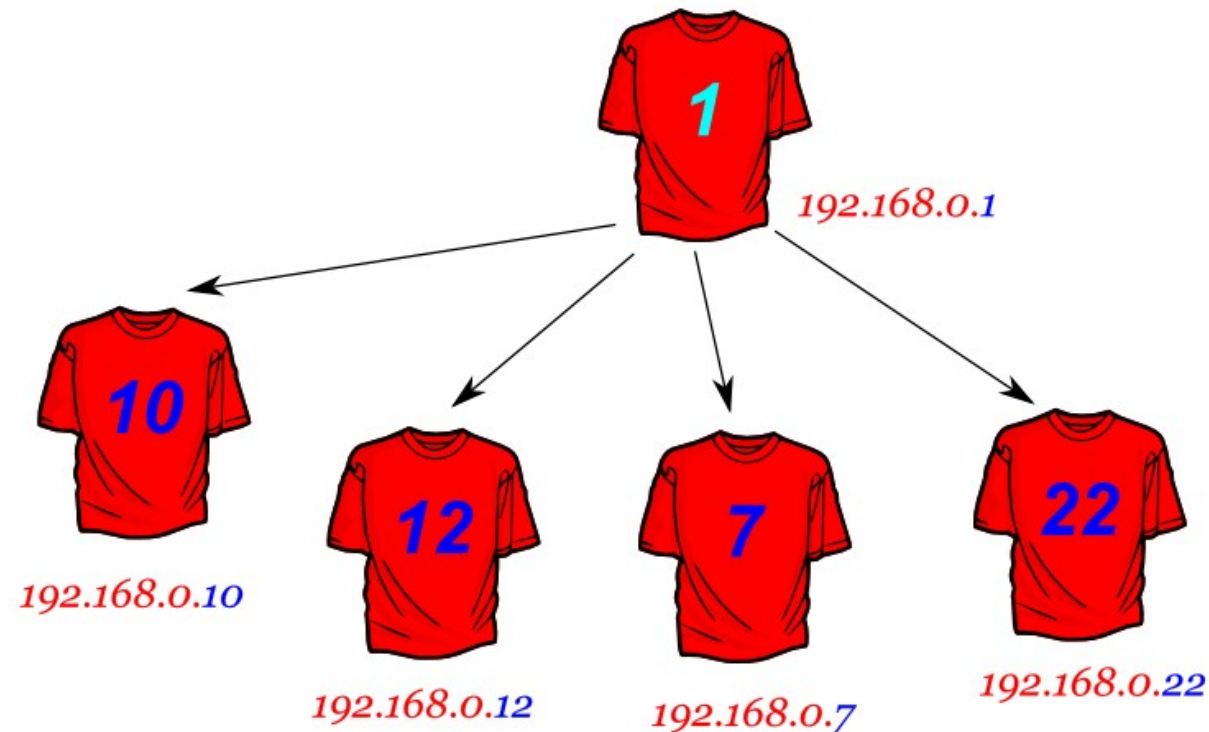
5. Au sein du réseau local : le n°1, un joueur un peu particulier !

Le n°1 : un joueur un peu particulier

- Au sein d'une équipe de foot, le n°1 a un rôle particulier : c'est lui le gardien.
- Au sein de notre « équipe » du réseau local, **le n°1 va aussi avoir un rôle particulier : c'est le routeur.**
- Le routeur, qui assure le rôle de serveur « DHCP », va avoir 2 fonctions essentielles :
 - il va **fixer le numéro du réseau**, autrement dit la couleur du maillot.
 - il va **donner à chaque joueur de l'équipe son numéro sur le réseau** (=dans l'équipe)
- Le nombre de « maillots » que le routeur va pouvoir distribuer est fixé d'avance : on appelle ça la plage des adresses IP. Par exemple, le routeur pourra donner les numéro 10 à 50 ou 100 à 149.

En pratique, ce que fait le routeur

- En pratique, le « routeur » assure le rôle de « serveur DHCP » et va donner à chaque poste du réseau (à chaque joueur) son adresse IP, autrement dit le maillot de couleur avec son numéro dessus.
- Le routeur fixe également la répartition des chiffres « couleur du maillot » et « numéro » au sein de l'adresse IP à l'aide du masque réseau. Généralement, le masque de réseau vaut 255.255.255.0.



6. Note technique : comment savoir quels numéros de l'adresse IP représentent la « couleur du maillot » ?

Structure de l'adresse ip.

- Il faut bien comprendre que :
 - un ou plusieurs des premiers groupes de chiffres représentent le **numéro du réseau**, (= la couleur du maillot)
 - un ou plusieurs des derniers groupes de chiffres représentent le **numéro de la machine** sur le réseau (le numéro sur le maillot). La répartition est fixée par un autre groupe de 4 nombres : le masque de sous-réseau (voir ci-dessous).

Exemples

- Par exemple, dans l'adresse ip **192.168.1.5** :
 - 192.168.1** (les 3 premiers groupes de chiffres) correspond au **numéro du réseau** auquel appartient la machine
 - .5** (le dernier groupe de chiffres) correspond au **numéro de la machine** sur le réseau **192.168.1**
- Autre exemple, dans l'adresse ip **145.227.1.5** :
 - 145.227** correspond au numéro du réseau auquel appartient la machine
 - .1.5** correspond au numéro de la machine sur le réseau **145.227**

Comment se répartissent le numéro du réseau et le numéro de poste au sein de l'adresse IP ?

- Pour savoir quels groupes de chiffres représentent le **numéro de réseau** et quels groupes de chiffres représentent le **numéro de poste** sur le réseau, on utilise le **masque de sous-réseau**.
- Au sein de l'adresse IP, la répartition des groupes de chiffres correspondant au **numéro du réseau** et ceux correspondant au **numéro de la machine** sur le réseau n'est pas figée d'un réseau à l'autre. Parmi le groupe de 4 nombres de l'adresse ip, pour savoir ce qui correspond au **numéro du réseau** et ce qui correspond au **numéro de la machine** sur le réseau, on utilise un autre groupe de 4 nombres appelé "**masque de sous-réseau**".
- Le principe du masque, pour faire simple est le suivant : **lorsque un nombre du masque est à 255, le nombre correspondant de l'adresse ip fait partie du numéro de réseau, lorsqu'un nombre du masque est à 0, le nombre correspondant de l'adresse ip fait partie du numéro de la machine.** (En fait, le masque sert à réaliser une opération logique bit à bit avec l'adresse ip)
- Par exemple :
 - l'adresse ip = **192.168.1.5**
 - le masque de sous-réseau est : **255.255.255.0**
 - donc le numéro de réseau est **192.168.1** et le numéro de la machine est **.5**
- Autre exemple : si maintenant, on a :
 - l'adresse ip = **192.168.1.5**
 - le masque de sous-réseau est : **255.255.0.0**
 - alors le numéro de réseau est **192.168** et le numéro de la machine est **.1.5**

Un point important (mais logique) : tous les postes d'un même réseau doivent avoir et ont le même masque réseau.

Qui fixe le masque réseau ?

- Le masque réseau est fixé par la fonction « serveur DHCP » du routeur (= le joueur n°1) dans le cas du mode automatique. Voir ci-dessous

Quel est l'utilité d'utiliser le masque réseau ?

- Certains pourront faire remarquer que l'on aurait pu figer la répartition et ne pas utiliser de masque de sous-réseau. Mais le masque de sous-réseau offre une grande souplesse dans la configuration du réseau :
 - Si le masque vaut **255.255.0.0**, alors il peut y avoir 65535 machines sur le réseau (mais le nombre de réseaux possibles est limité à 65535)
 - Si le masque vaut **255.255.255.0**, alors il peut y avoir 255 machines sur le réseau (et le nombre de réseaux possibles est de plus de 16 millions...)

7. Exercices d'entraînement

- Exercice 1 :
 - l'adresse ip = **192.168.3.102**
 - le masque de sous-réseau est : **255.255.255.0**
 - donc le numéro de réseau est **192.168.3** et le numéro de la machine est **.102**
- Exercice 2 :
 - l'adresse ip = **192.168.10.53**
 - le masque de sous-réseau est : **255.255.0.0**
 - alors le numéro de réseau est **192.168** et le numéro de la machine est **.10.53**
- Exercice 3 :
 - soit un réseau local ayant le masque de sous-réseau est : **255.255.255.0**
 - parmi les postes suivants, lesquels font partie du réseau ? Lequel est le routeur ?
 - **192.168.1.53**
 - **192.168.1.49**
 - **192.168.2.51**
 - **192.168.1.1**
 - **192.168.3.52**
 - **192.168.1.12**

8. Technique : Réseau : Comprendre les adresses IP : Synthèse

SYNTHESE :

Prenez le temps de bien comprendre le principe de l'adresse IP, sa structure, etc...

Retenez que :

l'adresse ip est constituée de 4 groupes de 3 chiffres

Ces 4 groupes de 3 chiffres définissent le numéro du réseau et le numéro du poste sur le réseau

la répartition de la signification des groupes de 3 chiffres est fixée par le masque réseau lui-même constitué de 4 groupes de 3 chiffres

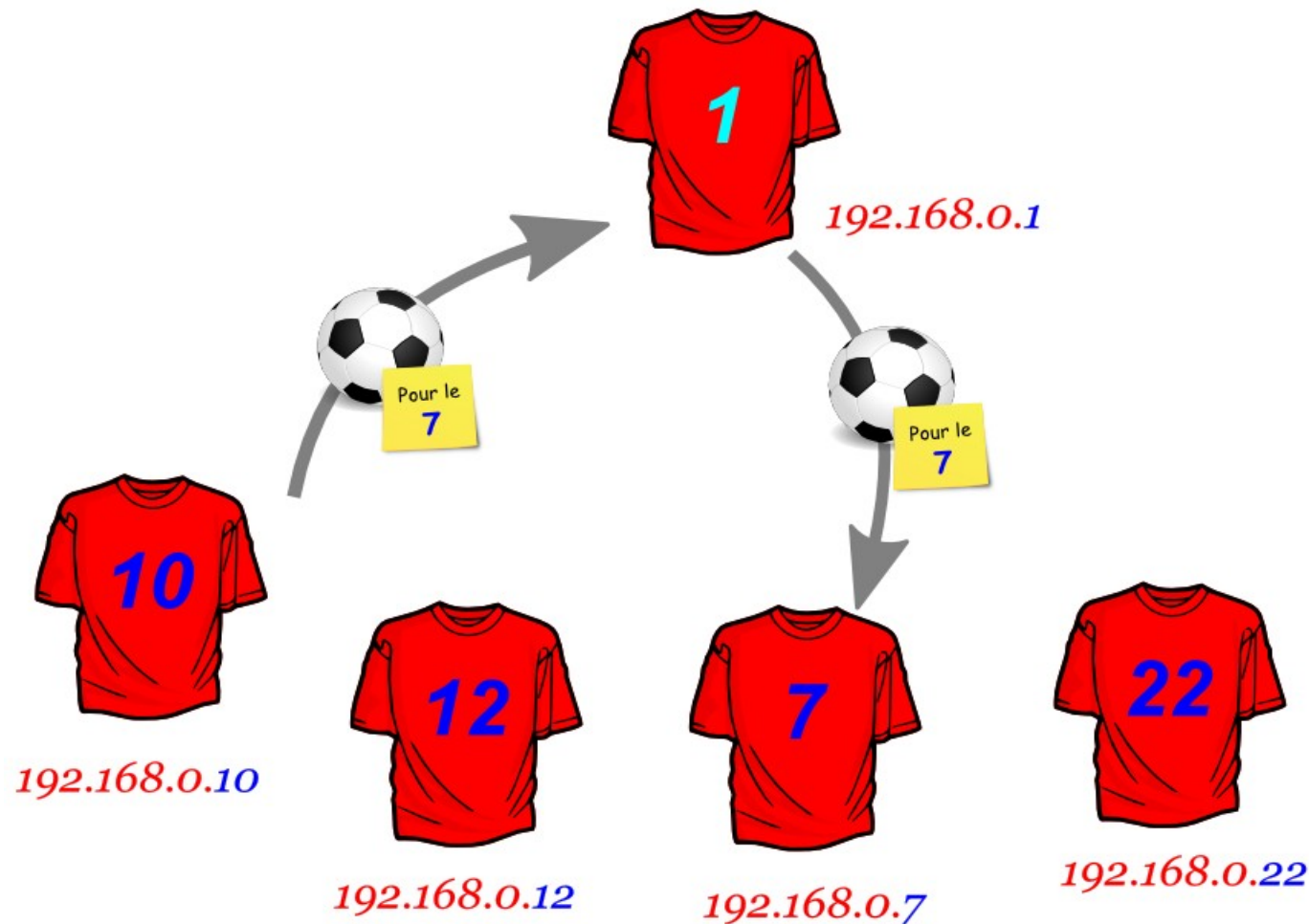
Les groupes de chiffres de l'adresse ip ayant un **masque valant 255** correspondent au **numéro de réseau**

Les groupes de chiffres de l'adresse ip ayant un **masque valant 0** correspondent au **numéro du poste**

9. Fonctionnement du réseau

Règles du « jeu »

- Sur le réseau, **seul le n°1 (= le routeur) voit les autres joueurs et sait qui est présent ou non**
- Lorsqu'un joueur veut envoyer quelque chose (le ballon) à un autre joueur, il « écrit sur le ballon » le numéro du destinataire, et **il est obligé de faire une passe au n°1 qui va renvoyer ensuite le « ballon » vers le joueur de destination** (ceci se fait automatiquement : le n°1 connaît bien son job !).
- Le n°1 est donc le « maître du jeu » : il assure la transmission entre les joueurs :
 - soit en tant que commutateur / switch tant que les messages s'échangent sur le réseau local
 - soit en tant que routeur si la connexion se fait vers un réseau extérieur au réseau local

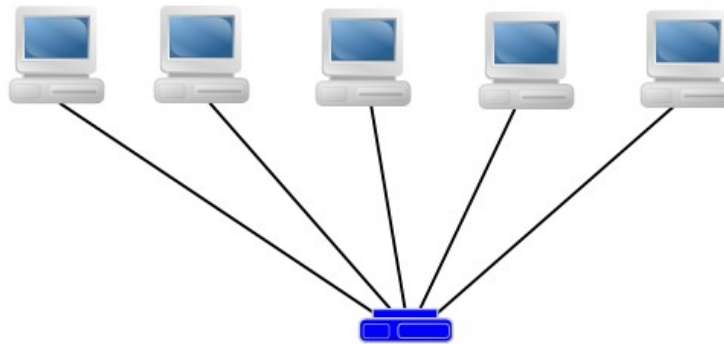


10. Techniquement : structure d'un vrai réseau et matériel nécessaire

Structure générale

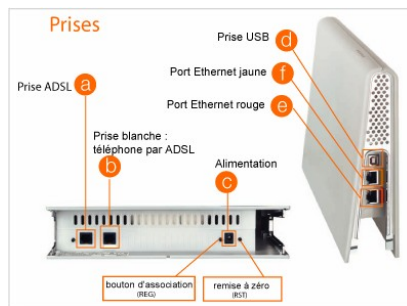
Techniquement, un réseau local type va avoir la même structure quelque soit la technique de connexion utilisée (filaire ou wifi) :

- l'un des postes va être le « meneur du jeu » : on l'appelle le routeur (en bleu sur le schéma). C'est lui qui :
 - fixe le numéro du réseau (couleur du maillot)
 - attribue les numéros individuels des postes (les numéros des joueurs), (rôle de serveur DHCP)
 - voit tous les postes
 - et distribue les messages (rôle de commutateur / switch sur le réseau local et/ou rôle de routeur si connexion au réseau extérieur)
- les joueurs sont l'ensemble des postes du réseau qui sont connectés au routeur.
- Les numéros des joueurs et du meneur sont appelés « adresse IP » et sont attribués par le routeur (mode automatique dit « DHCP »).



Le matériel de base nécessaire

- Pour constituer un réseau local filaire (le plus simple au début), on a donc besoin :
 - d'un **routeur** (ethernet ou wifi ou mixte) ou d'une box (qui est un routeur + modem)
 - de **plusieurs câbles RJ-45** pour connecter les éléments entre eux
 - d'un ou plusieurs **postes** à connecter sur le réseau
 - +/- d'un « multiprise » réseau, appelé switch, si le routeur n'a pas assez de connecteurs RJ-45 (Attention : un switch n'est pas un routeur... mais le routeur est un switch sur le réseau local !)



Purchased by Franck Ourion, franck.ourion@univ-lorraine.fr #6280170

Atelier Arduino : Introduction au réseau local et apprendre à monter un réseau local avec Arduino.

11. Synthèse : Un peu de vocabulaire pour avoir les idées claires

Avant de poursuivre, voici quelques définitions pour vous aider à avoir les idées claires :

LE RESEAU

Un Réseau :

Un ensemble (une équipe) de postes informatiques / électroniques reliés entre eux et capables de communiquer entre eux.

Réseau local :

Réseau constitué par les postes d'un même réseau (dans la maison ou le bureau). Le réseau local « pur » n'utilise pas d'accès vers l'extérieur, mais seulement une connexion locale de type Ethernet ou wifi.

Réseau internet ou web :

Réseau qui relie par le réseau téléphonique, et/ou fibre optique, des postes individuels et des réseaux locaux entre eux.

PRINCIPE DE FONCTIONNEMENT DE BASE DU RESEAU

Adresse IP :

Le numéro qui est attribué à chaque poste sur le réseau.

DHCP :

Fonction du routeur qui permet d'attribuer automatiquement les numéros (ou adresses IP) aux postes du réseau (les joueurs ...). Retenir que le routeur utilise une plage d'adresses prédéfinies, propre à chaque routeur.

IP fixe :

Se dit d'un poste du réseau dont l'adresse IP est fixée manuellement au lieu d'être attribuée par le routeur. Possible avec un routeur DHCP si l'adresse fixe est en dehors de la table DHCP.

LES MODES DE CONNEXION

Ethernet :

Protocole de communication réseau. Par abus de langage, on appelle « réseau ethernet » un réseau où les postes sont reliés entre eux par fil à l'aide d'un câble spécial appelé RJ45.

Wifi :

Réseau où les postes sont reliés entre eux sans fil, par ondes radios.

CPL :

Technologie qui permet d'utiliser le réseau électrique 220V pour connecter entre eux des éléments d'un réseau Ethernet ou wifi. Très pratique et efficace !

Si vous ne retenez pas tout pour le moment, ce n'est pas très grave. Les choses vont s'éclaircir progressivement

LES ELEMENTS DU RESEAU

Routeur :

En tant que tel, un routeur est un appareil permettant à des postes de réseaux différents de communiquer entre eux. Par « abus de langage », on appelle « routeur » sur un réseau local un appareil qui assure simultanément :

- le rôle de **serveur DHCP** : fixe le numéro du réseau et le numéro de chaque poste sur le réseau.
- Le rôle de « **commutateur réseau** » ou switch pour les échanges sur le réseau local
- Le rôle de **routeur** pour les échanges avec des réseaux extérieurs.
- Le rôle de **modem** pour communiquer sur le réseau téléphonique

Noter qu'un routeur peut fonctionner soit en ethernet (fil) soit wifi (sans fil) soit les 2 (cas le plus courant des box internet actuelles)

A part : Modem :

Appareil permettant de transmettre des données informatiques à l'aide du réseau téléphonique. C'est la fonction « modem » qui permet de se connecter à internet. La technologie utilisée actuellement par les modems est dite « ADSL ». Ici, cette fonction modem du routeur ne sera pas indispensable.

Box internet

Appareil électronique et informatique qui permet de se connecter à internet et à créer un réseau local domestique. Bien comprendre qu'une box est à la fois un modem (connexion à internet par le réseau téléphonique) et un routeur (qui permet aux ordinateurs d'un même réseau de communiquer entre eux).

Carte Réseau (Ethernet) :

Carte d'interface électronique intégrée à un ordinateur ou un dispositif et qui permet de communiquer sur un réseau filaire Ethernet (local). Le shield Ethernet est une mini carte Réseau.

Carte wifi :

Carte d'interface électronique intégrée à un ordinateur ou un dispositif et qui permet de communiquer sur un réseau wifi sans fil (local).

FONCTIONNEMENT DU RESEAU LOCAL

Serveur :

Se dit d'un poste du réseau qui répond à des requêtes en provenance d'un client. Un site internet par exemple est en fait un serveur réseau.

Client :

Se dit d'un poste du réseau qui envoie des requêtes (des demandes) à un serveur. Un navigateur sur un poste fixe constitue un client réseau.

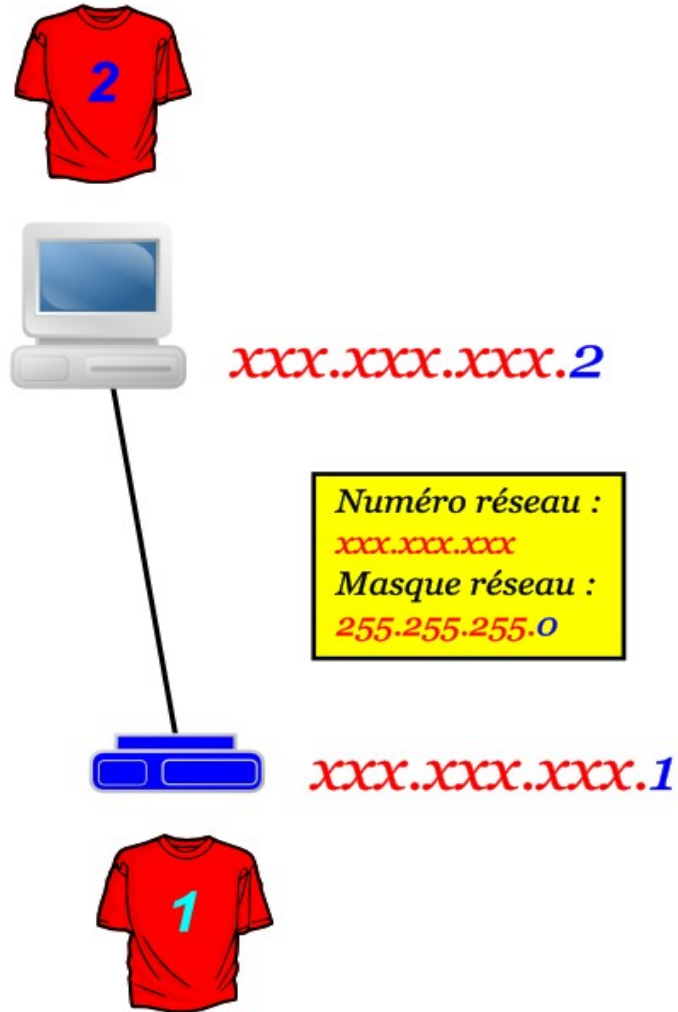
Protocoles TCP/IP et UDP :

Les protocoles TCP/IP et UDP correspondent à la façon dont les postes d'un réseau communiquent entre eux : TCP/IP est le protocole de l'internet, UDP un protocole plus simple.

12. Monter le réseau local minimum

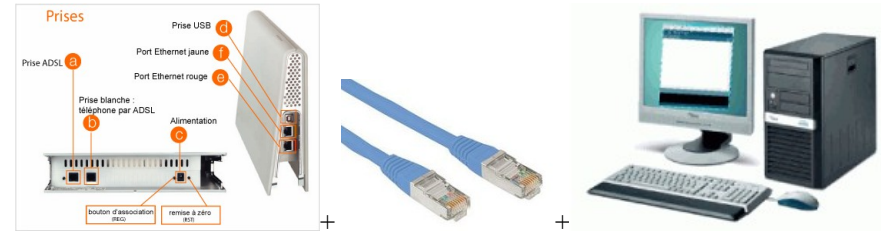
Le réseau minimum associe :

- un routeur (ou une box)
- un poste fixe disposant d'une carte réseau avec une prise ethernet
- un câble ethernet reliant le routeur et le poste fixe

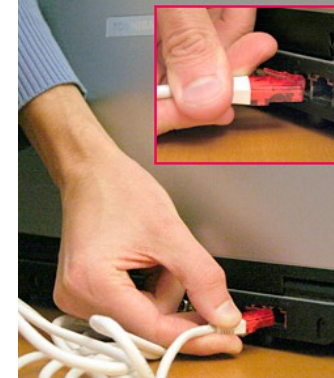


Montage du réseau minimum

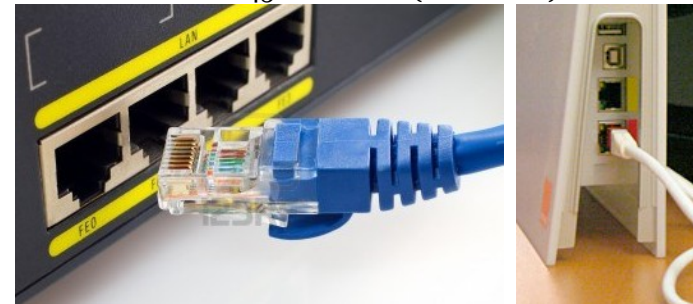
- Vous avez besoin de :



- Connecter le câble RJ 45 à la carte réseau du poste fixe :



- Connecter le câble RJ45 au routeur (ou à la box) :



Attention : certaines box ont une prise Ethernet dédiée pour la télévision : ne pas l'utiliser pour le réseau. Voir la doc de votre Box au besoin.

**C'est facile : quasiment impossible de se tromper !
Une fois fait, allumer votre box puis votre PC.**

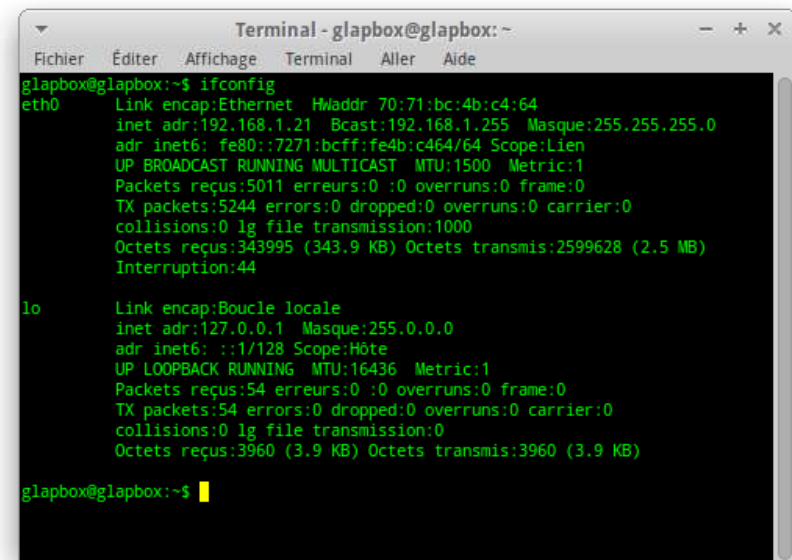
13. Tester et explorer le réseau : mise sous tension et récupération de l'adresse IP du poste fixe

Mise sous tension du réseau

- Cette fois, on va entrer (enfin!) dans le vif du sujet... Une fois le réseau monté comme indiqué, mettre sous tension le routeur puis le poste fixe.
- Normalement, lors de la mise sous tension du réseau, si tout s'est bien passé :
 - le routeur, qui est théoriquement configuré en mode automatique (ou DHCP) par défaut, a fixé automatiquement le **numéro du réseau** et le **masque réseau**
 - il s'est attribué le **numéro 1** et a donc une adresse IP de la forme **yyy.yyy.yyy.1**, typiquement **192.168.0.x**
 - et a attribué à votre poste fixe (normalement également configuré en mode DHCP par défaut) une adresse IP de la forme **yyy.yyy.yyy.x**, typiquement **192.168.0.x**
- A ce stade, si on le connectait au réseau, le couple carte **Arduino + shield Ethernet** n'est pas en mesure de communiquer avec le réseau car la carte Arduino n'a pas encore été programmée pour utiliser le réseau. Mais encore un peu de patience... ça ne va pas tarder.

Obtenir l'adresse IP du poste fixe et analyse des paramètres du réseau

- La première chose que l'on peut faire pour vérifier que le réseau est bien constitué, c'est de **récupérer l'adresse IP du poste fixe**. La manipulation n'est pas très compliquée et peu se faire en ligne de commande.
- Sur un système Gnu/Linux (Xubuntu ou autre), les choses sont relativement simples :
 - ouvrir un Terminal et saisir la commande **ifconfig**.
 - On obtient la liste des interfaces réseau du système : normalement l'interface locale appelée lo et l'interface ethernet appelée eth0.
 - L'adresse IP de votre poste sur le réseau est indiquée après les lettres **inet adr** : et le masque du réseau après **Masque** :
 - Si il n'y a aucun chiffre après inet adr :, vérifier que votre réseau ethernet est connecté (à l'aide du Network Manager du système).
- 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/connaître-votre-adresse-ip-sur-mac-os-43.php> : menu **Pomme**, ouvrez **Préférences Système** > **Réseau** > **Airport** > **TCP/IP**
 - <http://assistance.orange.fr/connaître-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**



```
glapbox@glapbox:~$ ifconfig
eth0      Link encap:Ethernet  Hwaddr 70:71:bc:4b:c4:64
          inet adr:192.168.1.21  Bcast:192.168.1.255  Masque:255.255.255.0
          adr inet6: fe80::7271:bcff:fe4b:c464/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:5011 erreurs:0 :0 overruns:0 frame:0
          TX packets:5244 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:343995 (343.9 KB) Octets transmis:2599628 (2.5 MB)
          Interruption:44

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:54 erreurs:0 :0 overruns:0 frame:0
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          Octets reçus:3960 (3.9 KB) Octets transmis:3960 (3.9 KB)

glapbox@glapbox:~$
```

Analyser l'adresse IP obtenue

- Dans l'exemple ici, on obtient l'adresse IP **192.168.1.21** et le masque est **255.255.255.0** donc l'adresse IP a la structure **192.168.1.21** avec :
 - **192.168.1** : le numéro du réseau local
 - **21** : le numéro du poste fixe sur le réseau

A ce stade, vous avez 3 infos importantes : **le numéro de votre poste, le numéro du réseau et le masque réseau**.
Vous êtes également sûr que votre réseau fonctionne, en tout cas, il en a l'air... cool non ?

14. 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.
- "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

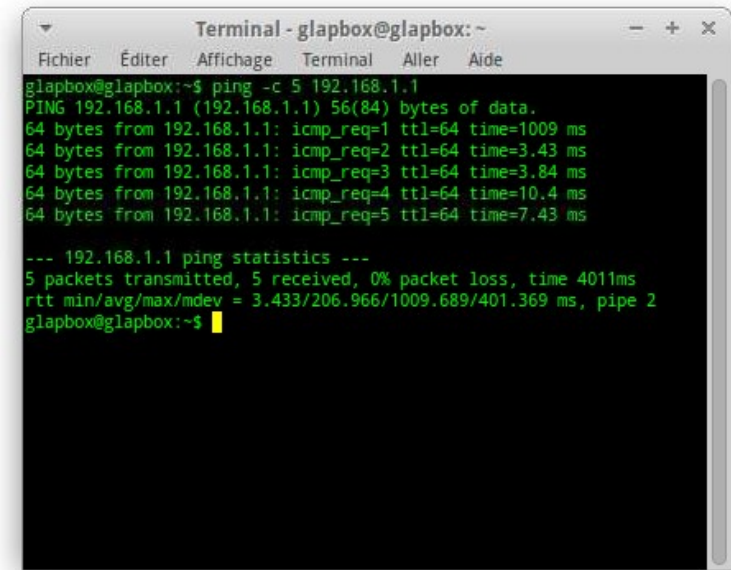
Si votre routeur est connecté à internet, vous pouvez tester simplement votre connexion internet en faisant un ping vers google en faisant :
`ping -c 5 www.google.fr`

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

A retenir

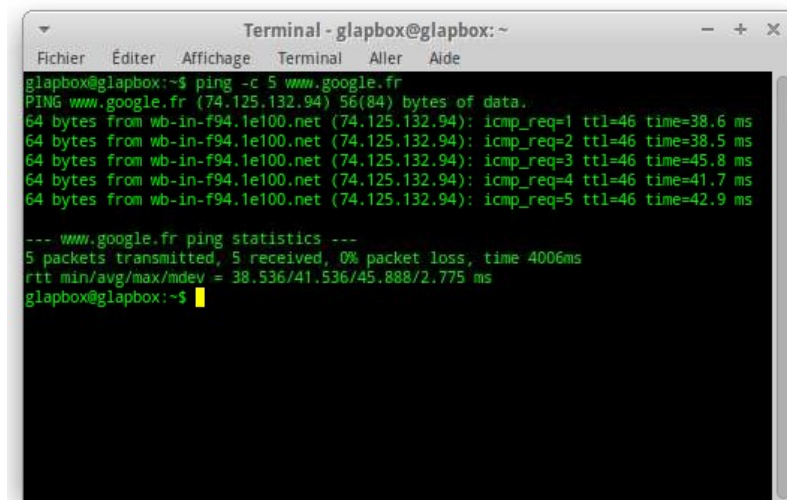
Par la suite, si votre réseau local rencontre un problème ou ne semble pas fonctionner, revenez à la base : faites un « ping » du poste fixe vers le routeur ou votre shield ethernet pour être sûr que tout fonctionne.

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

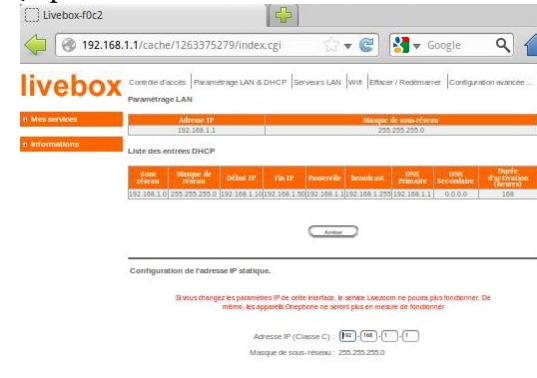
--- www.google.fr ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 38.536/41.536/45.888/2.775 ms
glapbox@glapbox:~$
```


15. Tester le réseau : Se connecter à l'interface du routeur et connaître la plage des adresses DHCP

- Comme nous l'avons vu précédemment, le routeur (=le « meneur du jeu ») dispose d'une plage de numéros utilisables pour les postes du réseau : c'est ce que l'on appelle la **plage DHCP** (dans l'image de l'équipe de foot, cela correspond au « maillot numérotés » que le meneur de jeu (le n°1) attribue aux nouveaux arrivants).
- Pour la suite, nous aurons besoin de connaître la plage DHCP du routeur. Pour ce faire, il va être nécessaire de se connecter à l'interface du routeur, ce qui au passage permet de vérifier que le routeur fonctionne bien.
- On procède comme pour se connecter à internet :
 - ouvrez votre navigateur Firefox ou autre,
 - et saisissez l'adresse **yyy.yyy.yyy.1** (le poste 1 correspond au routeur) soit dans mon exemple l'adresse **92.168.1.1**
- Vous devez alors obtenir une page vous demandant un login et un mot de passe. Par défaut, c'est souvent admin/admin ou admin/password mais cela dépend de votre routeur. Regarder la documentation au besoin. C'est parfois indiqué sur une étiquette sur le routeur.



- Une fois connecté, vous obtenez l'interface de paramétrage de votre routeur (aspect et structure variable selon les modèles de routeur) :



- Rechercher alors une rubrique DHCP ou équivalent et retrouver un champ ou une rubrique indiquant les numéros utilisés par le routeur DHCP :

Serveur DHCP

Sous-réseau : 192.168.1.0

Masque : 255.255.255.0

Début de la plage des
adresses IP : 192.168.1.10

Fin de la plage des
adresses IP : 192.168.1.50

- Comme on peut le voir ici, le routeur DHCP va utiliser les numéros compris entre 10 et 50 pour les postes qui vont se connecter au réseau. Cette plage de valeur peut varier d'un routeur à l'autre, parfois c'est 10-49, ou bien 100-150, etc... **Noter la plage utilisée par votre routeur.**

A ce stade, vous connaissez les paramètres essentiels de votre réseau :

l'adresse IP de votre poste, son numéro sur le réseau, le numéro du réseau, le masque de sous réseau (ifconfig)

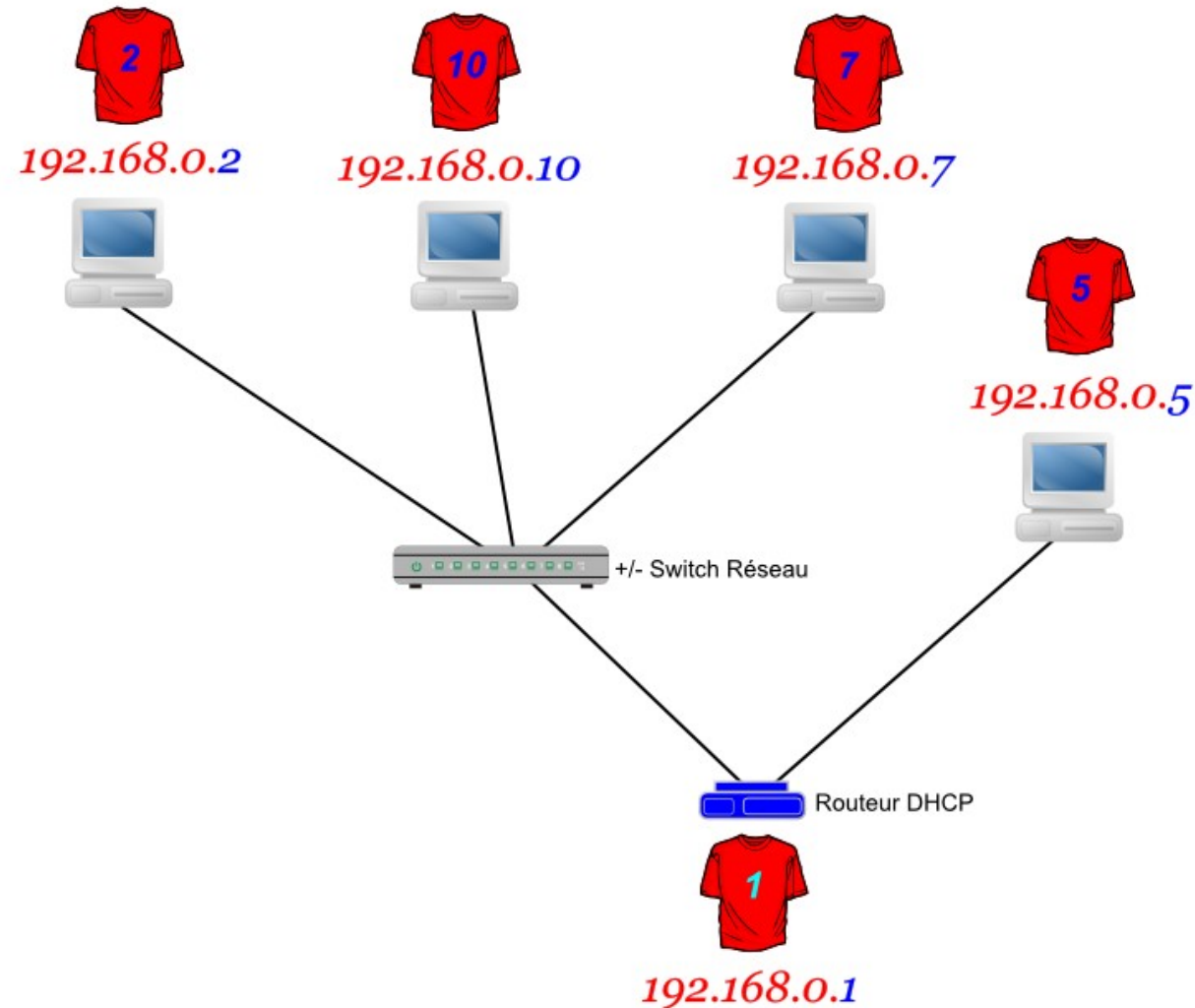
vous savez que votre réseau fonctionne et vous avez les délais de réponse (ping)

vous connaissez également la liste des adresses DHCP de votre routeur (connexion à l'adresse du routeur)

Cool non ? ... et pas si sorcier que ça !

16. Monter un réseau multi-postes : utiliser un « switch » ethernet si besoin...

- A présent, voyons comment monter un réseau local un petit peu plus sophistiqué, avec plusieurs postes.
- La seule différence ici, c'est l'utilisation d'un « switch » réseau si le routeur n'a pas assez de prises ethernet RJ45 : le switch réseau va jouer le rôle de « multiprise » ethernet, tout simplement. Le « switch » n'a pas d'adresse IP, ni de numéro : c'est seulement un « tube » ethernet.
- Pour le reste, c'est identique : on relie tous les postes au routeur soit directement soit via le switch : rien de très sorcier.



Une fois le réseau monté et mis sous tension, on pourra vérifier sur chaque poste l'adresse IP et faire des « ping » de poste à poste : à vous de jouer !

17. Truc avancé : connaître tous les postes présents sur un réseau local à partir d'un poste fixe connecté au réseau !



- Imaginez que vous soyez connecté à un réseau local dont vous ne connaissez pas la structure ou les adresses IP des postes présents : vous connaissez seulement votre adresse IP... Mais qui est présent sur le réseau ? Dans de nombreuses situations, c'est intéressant de le savoir. Sous Gnu/Linux, il existe un utilitaire qui permet de le savoir : le logiciel **nmap** (zenmap en mode graphique). Ce logiciel est d'ailleurs utilisé dans plusieurs films, dont Matrix. Cette fois vous entrez dans la cour des grands... !!
- Le principe est simple : on « scanne le réseau » en indiquant la plage d'adresses à tester :
 - dans un terminal, sous Gnu/Linux, saisir la commande :
 - **sudo nmap 192.168.1.1-255** (pour avoir également la liste des ports ouverts)
 - **sudo nmap 192.168.1.1-255 -sP** (pour seulement avoir la liste des postes présents)
 - de cette façon, on obtiendra l'ensemble des postes présents sur le réseau **192.168.1**

```
glapbox@glapbox:~$ sudo nmap 192.168.1.1-255 -sP
Starting Nmap 5.21 ( http://nmap.org ) at 2012-08-31 15:23 CEST
Nmap scan report for 3.home (192.168.1.1)
Host is up (0.011s latency).
MAC Address: (Sagem Communication)
Nmap scan report for -desktop.home (192.168.1.13)
Host is up (0.0095s latency).
MAC Address: (Intel Corporate)
Nmap scan report for xavier-.home (192.168.1.14)
Host is up (0.000079s latency).
MAC Address: (Micro-star Int'l Co.)
Nmap scan report for desktop.home (192.168.1.18)
Host is up (0.0079s latency).
MAC Address: (Unknown)
Nmap scan report for new-host-4.home (192.168.1.21)
Host is up.
Nmap done: 255 IP addresses (5 hosts up) scanned in 5.61 seconds
glapbox@glapbox:~$
```

- Sur ce réseau **192.168.1.**, on retrouve **5 postes présents** ayant les **numéros 1 (le routeur), 13, 14, 18 et 21**. Pratique non ?

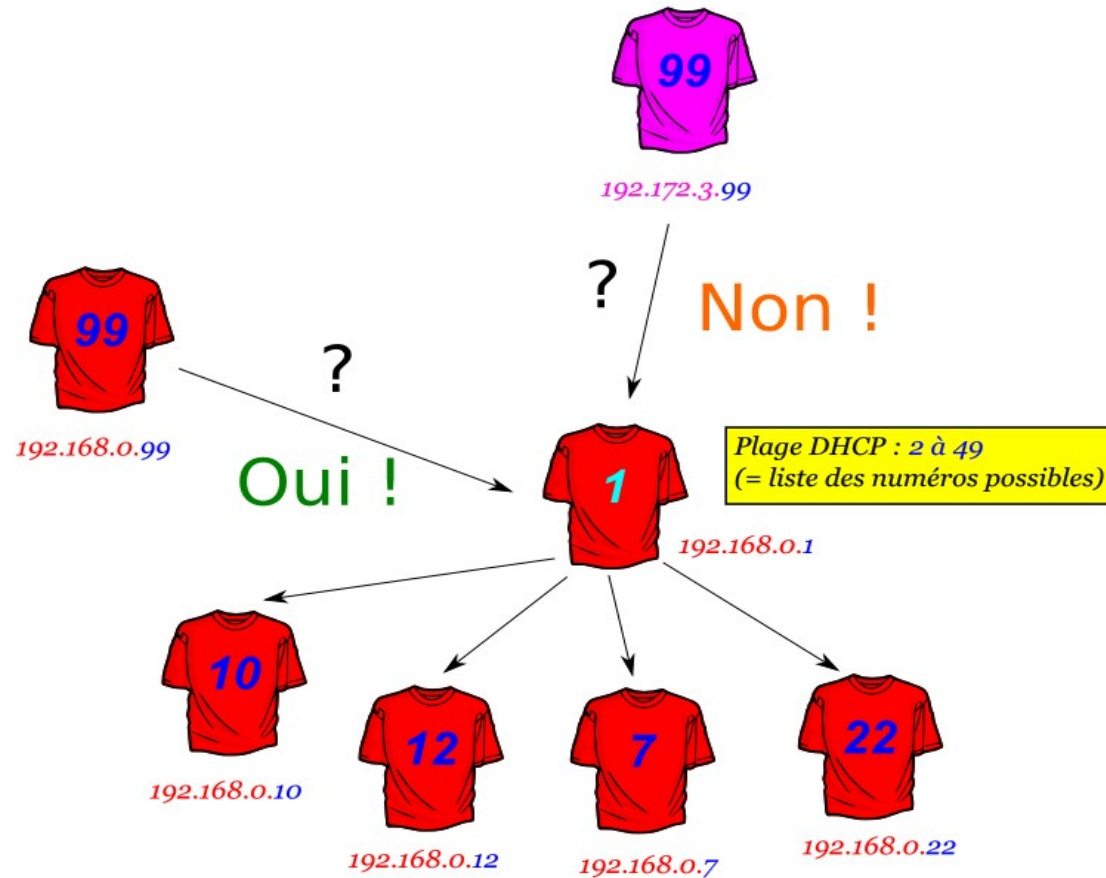
18. Les IP statiques : « allez, on fait tout à la main... ! »

- Jusqu'à présent, je n'ai parlé que de la situation où le routeur s'occupe de donner à chaque poste du réseau son adresse IP : autrement dit, le n°1 distribue les maillots avec les numéros aux joueurs. Ce mode de fonctionnement s'appelle DHCP (pour Dynamic Host Configuration Protocol) et est le mode activé par défaut sur la plupart des ordinateurs et routeurs modernes.
- Mais il est possible de faire autrement : on peut tout fixer « à la main » ! On appelle ça les adresse IP statiques : dans ce cas, chaque poste définit son numéro, le numéro du réseau et le masque de réseau. Pareil pour le routeur qui est configuré manuellement. Son rôle alors consiste à seulement faire transiter les messages entre les postes... mais « il ne distribue pas les maillots » dans ce cas.
- En pratique, c'est plus compliqué... et quand on débute avec les réseaux, c'est un coup à passer des heures à chercher ce qui ne va pas... !! Pour que ça fonctionne, il faut que tous les postes aient le même numéro de réseau, le même masque, et des numéros de postes uniques... logique, mais pas forcément simple... Tout changement sur le réseau peut également entraîne des perturbations...
- Donc, croyez-moi, préférez le DHCP (adressage automatique) surtout qu'on trouve des routeurs à pas cher... !

Je sais qu'il y en a qui préfèrent utiliser les IP fixes... mais c'est quand moins simple pour le néophyte !

19. Un cas à part : « le gars qui se pointe avec son maillot déjà sur lui »...

- On peut parfois se retrouver dans une situation où l'on dispose d'un réseau avec adressage IP automatique par DHCP (autrement dit le routeur distribue les maillots à tous les joueurs...) et il y a un poste avec une IP fixe sur ce réseau... Je vous en parle, car la situation va se présenter avec Arduino. En effet, la fonction DHCP demande une certaine sophistication du poste et dans certains cas, cela n'est tout simplement pas possible ou pas souhaitable...
- Donc, si on se résume, on est dans la situation du match qui a commencé avec déjà une équipe constituée, et il y a un « gars qui se pointe avec son maillot » (=il a fixé lui-même son adresse IP) ! Est-ce qu'il a le droit de jouer avec le reste de l'équipe ??? Alors.... à votre avis ??
- Réfléchissez bien : la réponse est oui si :
 - il a la « même couleur de maillot »
 - ET si il a **un numéro qui n'est pas dans la liste de ceux que le routeur peut utiliser** (çà, c'est l'info du jour, même pour certains connaisseurs...)
- Techniquement, çà veut dire : même masque, même numéro de réseau et un numéro en dehors de la plage DHCP du routeur. C'est le secret pour faire un réseau mixte utilisant des adresses IP fixes et automatiques (DHCP)

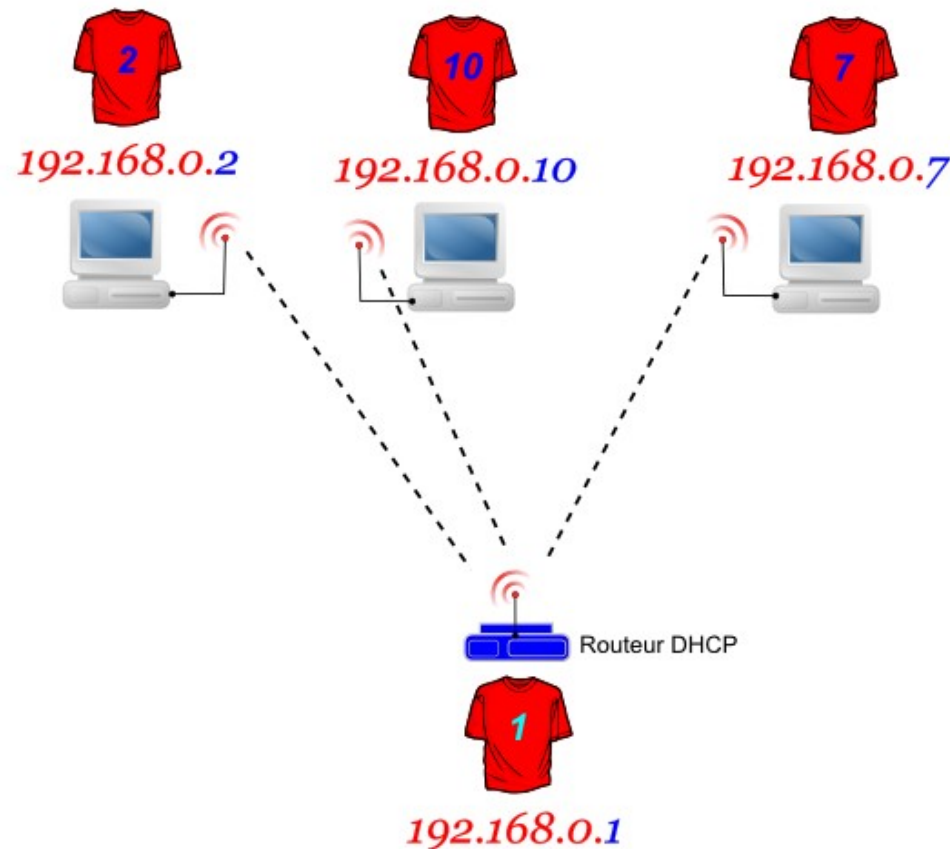


Utiliser une adresse IP fixe est intéressant quand on veut pouvoir se connecter à un élément précis du réseau sans avoir à chercher son IP par un scan réseau

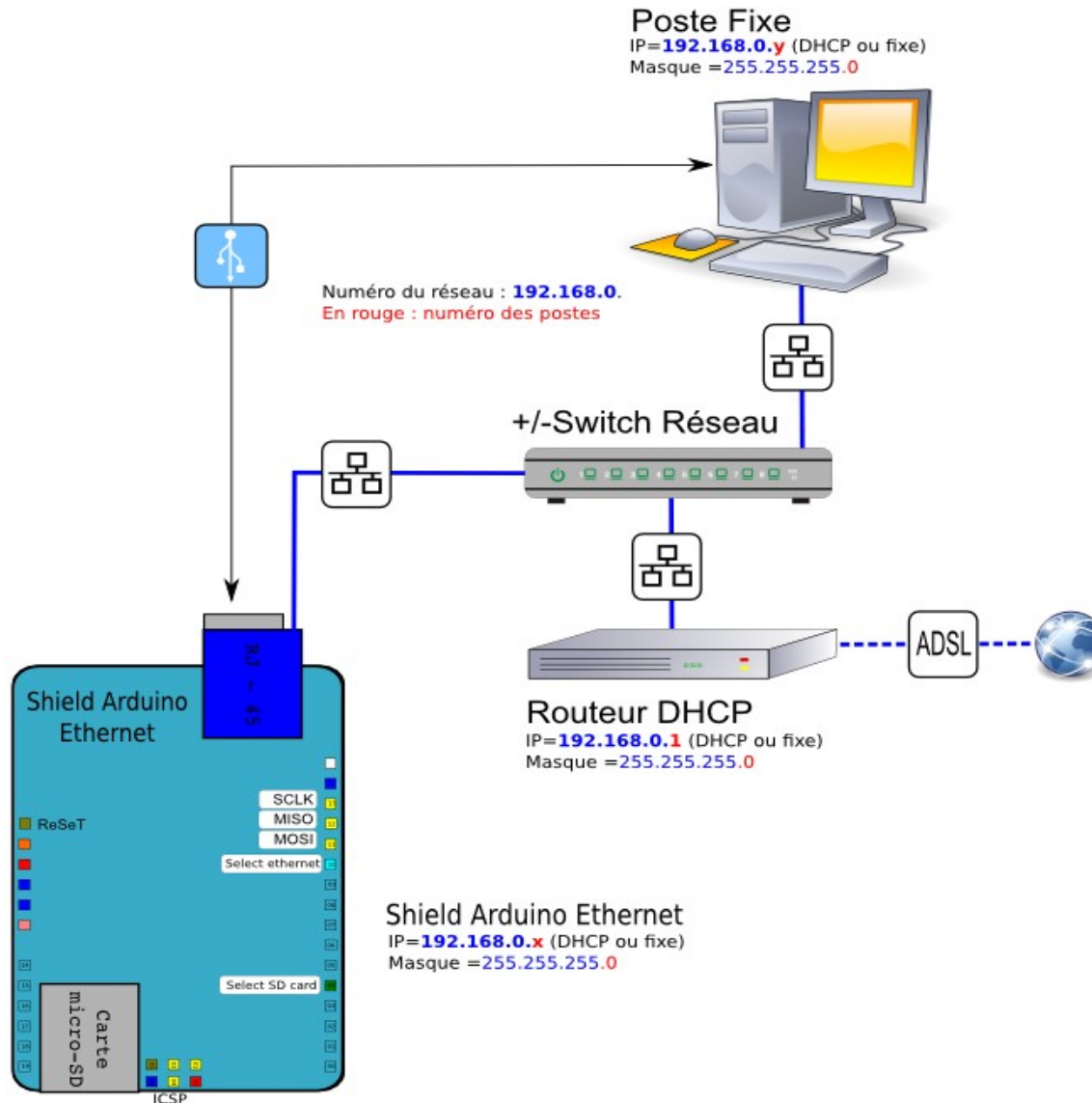
20. Et en wifi ???

- La seule chose qui change avec le wifi, c'est la façon de connecter les postes entre eux : on utilise les ondes radio au lieu d'un câble ethernet.
- Il faut évidemment un routeur wifi (c'est le cas des box actuelles) et une clé ou une carte wifi sur les postes wifi ...
- Sinon, pour le reste, c'est pareil sauf qu'il faut paramétrer la connexion wifi (protection numérique par clé WEP) :
 - dans un premier temps, vous pouvez paramétrer le routeur sans sécurité WEP ce qui facilite la connexion et accélère également le débit. C'est jouable si votre routeur n'est pas connecté à internet et que vous faites un petit réseau local.
 - Sinon, sur chaque poste, lors de la première connexion au réseau, vous devrez entrer une clé de sécurité dite wep ou wpa ou autre...

Une fois le réseau wifi constitué, il est possible de faire un ping, un scan réseau, etc...



21. Monter un réseau local avec Arduino : La structure du réseau que nous allons réaliser



Notre réseau utilisant Arduino va être constitué au minimum :

- d'un **routeur ethernet** (ou d'une box) fonctionnant en mode DHCP (=attribution automatique des adresses) avec au moins 1 prise ethernet RJ45 libre
- +/- d'un **switch réseau** (=«multiprise » réseau) si le routeur ne dispose que d'une prise ethernet RJ45
- d'un **poste fixe**, le pc sur lequel vous travaillez, connecté au routeur directement au routeur ou sur le switch avec un câble ethernet RJ45
- d'un **couple « carte Arduino + shield Ethernet »** connecté également directement au routeur ou sur le switch avec un câble ethernet RJ45

Dans un premier temps, le routeur n'a pas besoin d'être connecté à Internet.

Si il y a plus d'éléments sur votre réseau, cela n'a aucune importance, mais dans un premier temps, mieux vaut faire simple.

Remarquer que le couple « Arduino/shield Ethernet) est connecté au PC fixe de 2 façons :

- par USB d'une part
- et par ethernet d'autre part

Ceci est très pratique en phase de test et de mise au point, mais une fois la programmation terminée, on pourra bien sûr déconnecter le câble USB.

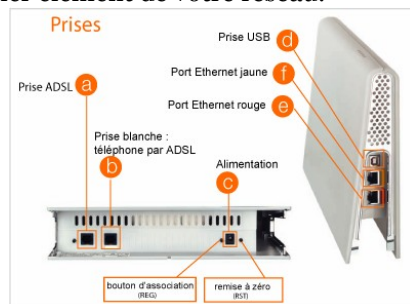
La signification des numéros (adresses IP) indiqués sur ce schéma seront expliqués par la suite.

22. Les éléments du réseau local avec Arduino 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.

Noter que si vous utilisez un routeur ou une box non connecté à internet, ce n'est pas grave : vous n'aurez pas besoin de l'accès à internet pour la majorité des programmes que je vous propose.

Sur votre routeur, vous devez disposer d'au moins une prise ethernet RJ45, et idéalement 2.

+/- Élément complémentaire du routeur : le switch Ethernet

Si vous ne disposez que d'une prise Ethernet sur le routeur (c'est parfois le 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.

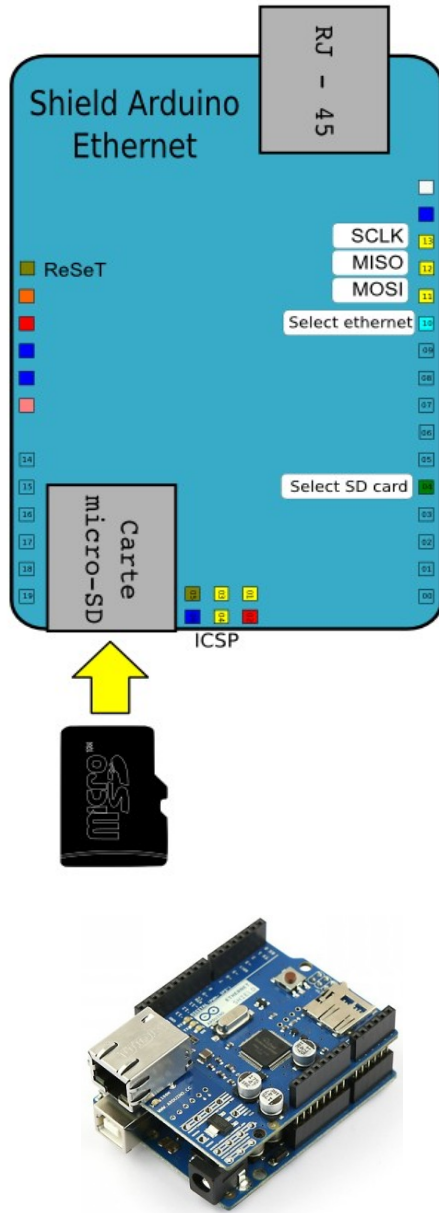


Le troisième élément de votre réseau : le couple « carte Arduino + shield Ethernet »

Le troisième élément du réseau est bien évidemment le shield Ethernet qui sera tout simplement enfiché broche à broche sur la carte Arduino



23. Le shield Ethernet : description et principe d'utilisation



Description

- Le module Ethernet Arduino permet à une carte Arduino de se connecter à un réseau Ethernet filaire ou même à internet.
- Ce module intègre également un emplacement pour une carte mémoire micro-SD, pouvant stocker plusieurs Go de données !
- Ce module est basé sur le circuit intégré [Wiznet W5100](#). Le Wiznet W5100 fournit une pile réseau (IP) capable à la fois de **TCP et UDP**. Il supporte jusqu'à quatre connexions simultanées.
- Il suffit d'utiliser la **librairie Ethernet** pour écrire des programmes qui se connectent à un réseau ou à internet en utilisant ce module.

Brochage

- Ce shield communique avec la carte Arduino en utilisant une communication SPI (voir atelier dédié pour plus de détails). Cette communication utilise les 3 broches suivantes 11 (MOSI), 12 (MISO) et 13 (CLK). *Noter que la connexion de ces broches se fait par le connecteur ICSP de la carte Arduino.*
- La sélection de l'étage utilisé (carte SD ou Ethernet) se fait à l'aide de 2 broches de sélection :
 - la broche 10 pour sélectionner l'étage Ethernet
 - la broche 4 pour sélectionner la carte mémoire SD

La gestion des broches sera assurée automatiquement par les bibliothèques.

- Toutes les autres broches de la carte Arduino restent disponibles pour d'autres utilisations.

Principe d'utilisation

- Le module ethernet se connecte « broche à broche » sur une carte Arduino grâce à ses longues broches qui dépassent du circuit imprimé. On dit que le shield est « stackable » !
- Ainsi le brochage de la carte Arduino n'est pas modifié et permet d'enficher un autre module par dessus et laisse l'accès aux broches de la carte Arduino.

24. Monter le réseau utilisant le shield Ethernet Arduino

- A faire hors tension dans la mesure du possible... Connecter le câble RJ 45 à la carte réseau du poste fixe :



- Connecter le câble RJ45 au routeur (ou à la box) :



- Mettez le shield Ethernet en place sur la carte Arduino :



- Puis connecter le shield Arduino au routeur à l'aide d'un câble RJ-45 :



- Connecter enfin le câble USB entre l'ordinateur et la carte Arduino.



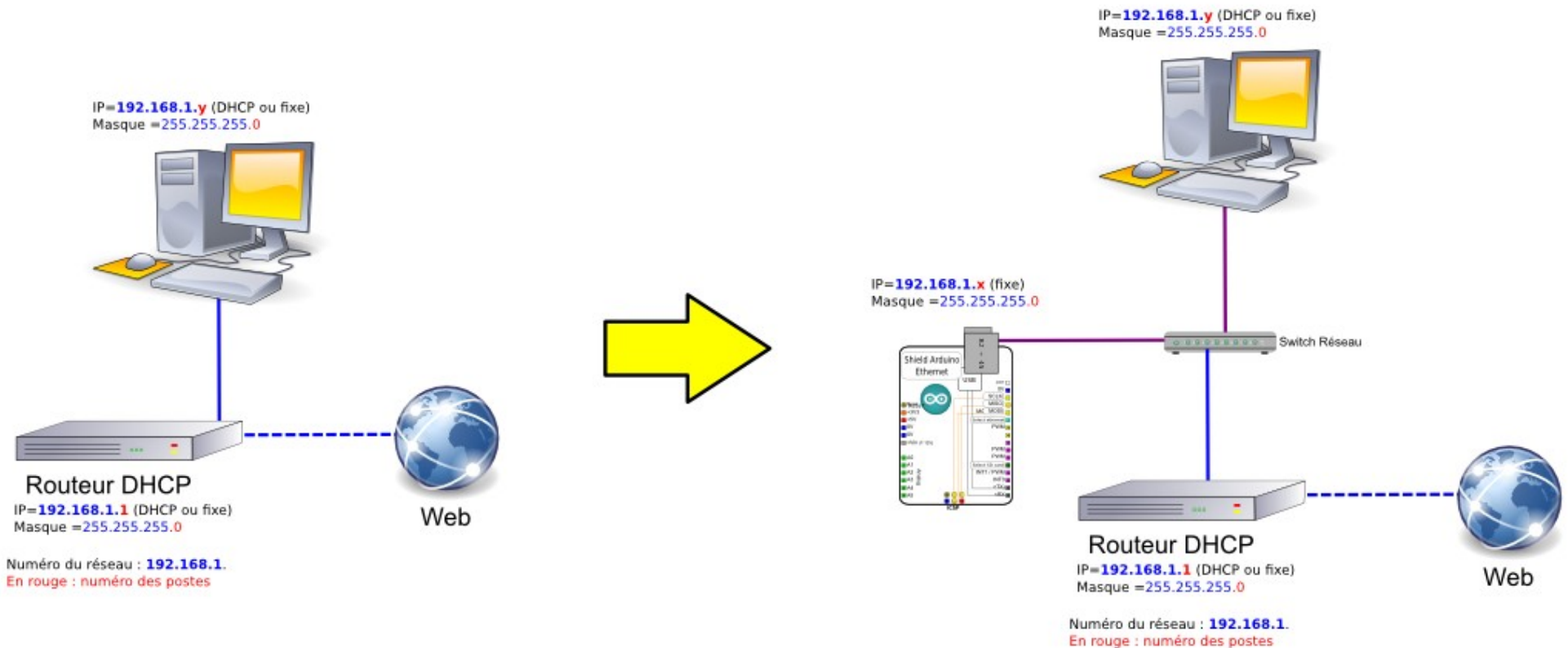
Dans notre cas, on connecte le couple Arduino + shield Ethernet à la fois en USB et en Ethernet au poste fixe, ceci uniquement à des fins didactiques et pour faciliter les développements.

En situation réelle, évidemment, le couple Arduino + shield Ethernet pourra être utilisé à distance du poste fixe, pourvu qu'il soit connecté au réseau Ethernet.

25. Monter le réseau utilisant le shield Ethernet Arduino sur un réseau avec « box » existant

Remarque :

Si votre poste fixe est déjà connecté à votre box internet, la manip' à réaliser est simple : il suffit de débrancher le câble ethernet de votre poste fixe et de le brancher sur le switch réseau. Ensuite, connecter un câble entre le switch réseau et votre PC. Puis un second câble entre le switch réseau et le shield Ethernet enfiché sur la carte Arduino. C'est tout.



26. Rappel : Langage Arduino : Introduction aux bibliothèques

C'est quoi une bibliothèque Arduino ?

Le langage Arduino comporte de nombreuses instructions comme vous avez pu le constater, une quarantaine en tout. Ces instructions sont intégrées dans ce que l'on appelle le « noyau » ou « cœur » (core en Anglais) du langage Arduino. Ces instructions sont « générales » et servent souvent.

Le langage Arduino peut cependant être étendu à la demande avec des instructions dédiées à certaines applications particulières : afin de ne pas surcharger inutilement le « cœur », ces instructions spécifiques ont été intégrées dans des « paquets d'instructions » appelés bibliothèques.

Comment ça marche ?

Par exemple, si on utilise un afficheur LCD, un servomoteur ou encore si l'on utilise un shield ethernet (réseau), on va intégrer dans notre programme la bibliothèque dédiée correspondante.

Principe général d'utilisation

Pour intégrer une bibliothèque dans un programme Arduino, c'est très simple : il suffit d'ajouter en début de programme une ligne de la forme :

```
#include <nombibliothèque.h> // bibliothèque pour servomoteur
```

ATTENTION : l'instruction include est un peu particulière : la ligne commence par un # et il n'y a pas de point virgule de fin de ligne !

Ensuite, dans le code, au niveau de l'entête déclarative, là où vous déclarez vos variables, il va falloir déclarer un objet (une sorte de super variable) représentant la bibliothèque. Cet objet est en fait une instance (= un exemplaire) d'une Classe (=le moule) qui regroupe les fonctions de la bibliothèque. On a :

```
ClasseObjet monObjet; // déclare un objet
```

Généralement ensuite :

- au niveau de la fonction **setup()**, on initialise l'objet avec les paramètres voulus
- au niveau de la fonction **draw()**, on appelle les fonctions de la bibliothèque sous la forme que vous connaissez déjà :

```
monobjet.fonction( param, param, ..);
```

Rappel : pour utiliser une fonction d'une classe du langage Arduino, on utilise le nom de la classe + un point + le nom de la fonction.

Il peut exister des variantes selon les bibliothèques, mais grosso-modo, ça fonctionne de cette façon pour la plupart des bibliothèques Arduino.

Vous avez déjà utilisé une bibliothèque !

Si vous êtes attentifs à tout ce qu'on a déjà vu, vous me direz que ça ressemble étrangement à l'utilisation de la classe **Serial...** et vous aurez raison ! En fait, la classe Serial est une bibliothèque qui est intégrée implicitement lorsque vous lancez Arduino : c'est pour ça que vous n'avez pas besoin d'utiliser **#include** pour l'utiliser.

Les bibliothèques standards Arduino

Les bibliothèques Arduino disponibles sont nombreuses, et disposent chacune de quelques fonctions à plusieurs dizaines... ce qui étend considérablement la puissance du langage Arduino et qui en fait aussi tout son intérêt. Voici la liste des bibliothèques standards du langage Arduino (**le logiciel donne la liste...**) :

- [La bibliothèque Serial](#) - pour les communications séries entre la carte Arduino et l'ordinateur ou d'autres composants
- [La bibliothèque LCD](#) - pour l'utilisation et le contrôle d'un afficheur LCD alpha-numérique standard.
- [La bibliothèque Servo](#) - pour contrôler les servomoteurs.
- [La bibliothèque Stepper](#) - pour contrôler les moteurs pas à pas (nécessite une interface de commande)
- [La bibliothèque Ethernet](#) - pour se connecter à Internet en utilisant le module Arduino Ethernet
- [La bibliothèque EEPROM](#) - référence - pour lire et écrire dans la mémoire EEPROM non volatile.
- [La bibliothèque SD](#) - référence - pour utiliser une carte mémoire SD (utiliser des fichiers, stocker des données, ...)
- [La bibliothèque SoftwareSerial \(Série Logicielle\)](#) - référence - pour communication série logicielle sur n'importe quelles broches de la carte Arduino
- [La bibliothèque Wire / I2C](#) - référence - Interface "deux fils" (TWI/I2C) pour envoyer et recevoir des données sur un réseau de modules ou capteurs.
- [La bibliothèque SPI \(Serial Peripheral Interface\)](#) - pour communication série avec des modules externes supportant le protocole SPI
- [Firmata](#) - pour communiquer avec des applications sur l'ordinateur utilisant un protocole série standard.

En jaune les plus utiles. Impressionnant non ? On les étudiera pas à pas...

Les bibliothèques de la communauté

A côté de ces bibliothèques standards, il existe toute une série de bibliothèques proposées par les uns et les autres et qui concernent des matériels spécifiques, ou autre. Par exemple :

- [La bibliothèque Keypad](#) - pour l'utilisation des claviers matriciels. (**hors référence**).

Faites un tour ici pour voir ce qui existe : <http://arduino.cc/playground/Main/LibraryList>

27. La bibliothèque Ethernet

Présentation

Purchased by Franck Ourion, franck.ourion@univ-lorraine.fr #6280170

Atelier Arduino : Introduction au réseau local et apprendre à monter un réseau local avec Arduino.

Exemple du principe d'utilisation en mode Client

- Cette librairie Arduino permet à une carte Arduino de communiquer sur un réseau ethernet local ou même internet en utilisant un shield Ethernet.

IMPORTANT : Utiliser la dernière version Arduino 1.01 qui corrige quelques bugs existant de la librairie Ethernet.

Inclusion

La librairie s'intègre dans un programme avec la ligne (pas de ; !!) :

```
#include <Ethernet.h> // inclut la librairie Ethernet
```

Noter que cette librairie nécessite également l'utilisation de la librairie de communication série SPI que l'on inclura avec la ligne :

```
#include <SPI.h> // inclut la librairie SPI
```

Les classes de la librairie

A la différence de plusieurs autres librairie qui n'intègrent qu'une seule classe (une classe est un ensemble de fonctions regroupées), la librairie Ethernet intègre plusieurs classes :

- la classe **Ethernet** qui correspond aux fonctions générales de configuration matérielle du shield Ethernet,
- la classe **IPAddress** qui représente une adresse IP
- la classe **EthernetServer** qui correspond aux fonctions utiles lorsque le shield Ethernet est configuré en serveur réseau (protocole TCP)
- la classe **EthernetClient** qui correspond aux fonctions utiles lorsque le shield Ethernet est configuré en client réseau (protocole TCP)
- la classe **EthernetUDP** qui permet une communication allégée sur réseau (protocole UDP)

Le constructeur de la classe existe sous plusieurs formes.

Les fonctions de la librairie

Chaque classe dispose de plusieurs fonctions associées :

Classe Ethernet (configuration matérielle du shield Ethernet)

- | begin() | localIP() | maintain()

Classe EthernetServer (serveur TCP)

- | begin() | available() | write() | print() | println()

Classe EthernetClient (client TCP)

- | connected() | connect() | write() | print() | println() | available() | read() | flush() | stop()

Classe EthernetUDP (communication UDP)

- | begin() | read() | write() | beginPacket() | endPacket() | parsePacket() | available() | remoteIP() | remotePort()

La structure type d'un programme utilisant un shield Ethernet va être la suivante :

Au niveau de l'entête déclarative

- Inclusion des librairies **SPI** et **Ethernet**
- Définition d'un tableau de **byte** correspondant à l'adresse MAC du shield
- Définition de l'adresse IP du serveur auquel se connecter
- Déclaration d'un objet **EthernetClient**

Au niveau de la fonction setup()

- Initialisation de la librairie Ethernet avec l'instruction **Ethernet.begin(mac,ipLocal)**. Bien comprendre que la classe Ethernet configure le shield Ethernet au niveau matériel.

Au niveau de la fonction loop()

- A ce niveau on utilisera selon les besoins toutes les fonctions utiles de la librairie :
 - pour tester la connexion
 - pour envoyer la requête vers le serveur
 - pour lire les données en provenance du serveur,
 - etc...

Dans les pages qui suivent nous allons détailler les différents cas de figure, pas à pas, dans le cadre d'un réseau utilisé avec le protocole dit TCP. La librairie Ethernet Udp qui permet d'utiliser le protocole UDP, protocole simplifié pour l'envoi de texte et de petites quantités données sur le réseau, est abordée dans un autre atelier.



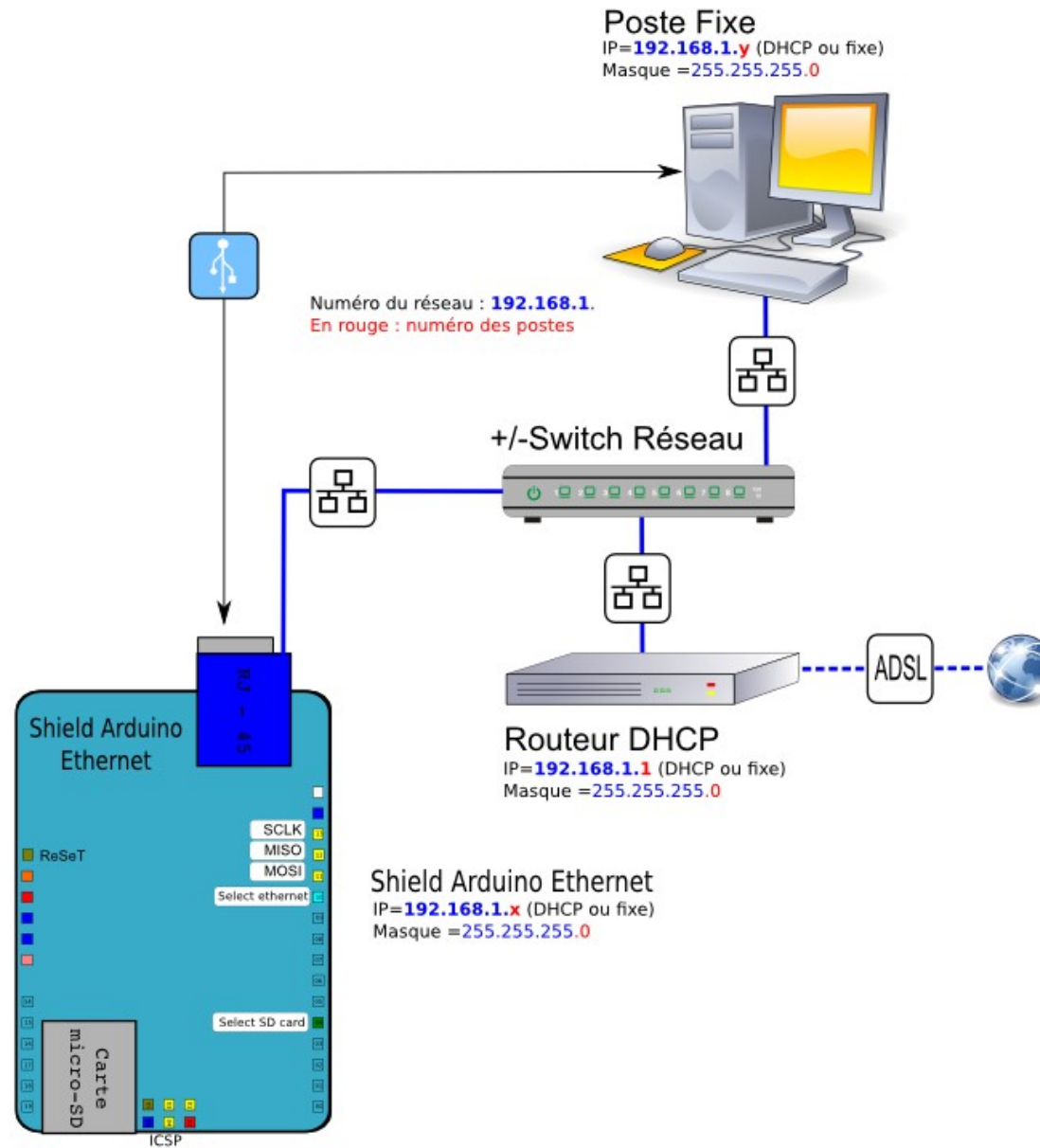
28. Faire un simple « ping » vers le shield Ethernet à partir d'un poste fixe : le réseau

- Enfin, nous allons passer à l'action et coder notre premier programme « réseau » pour Arduino . Prêt ? C'est parti.... Voici donc le réseau que nous allons

Purchased by Franck Ourion, franck.ourion@univ-lorraine.fr #6280170

Atelier Arduino : Introduction au réseau local et apprendre à monter un réseau local avec Arduino.

utiliser :



29. Faire un simple « ping » vers le shield Ethernet à partir d'un poste fixe : le programme

Ce qu'on va faire ici...

- Ce programme va être le « programme minimum » pour utiliser le shield Ethernet en place sur la carte Arduino au sein d'un réseau local. On va ici se contenter de configurer l'adresse IP fixe du shield ethernet et ensuite faire un « ping » vers le shield à partir du poste fixe. Prêt ? ta.. ta.. ta... c'est parti !

Je rappelle une nouvelle fois qu'il est nécessaire d'utiliser la version **Arduino 1.01** (ou suivante) avec les codes qui suivent.

Entête déclarative

Inclusion des bibliothèques utiles

- On commence par inclure les bibliothèques
 - SPI** qui permet au shield Ethernet de communiquer avec la carte Arduino
 - et la bibliothèque **Ethernet** qui comporte toutes les fonctions nécessaires pour la communication du shield Ethernet sur le réseau Ethernet local.

Variables et objets utiles

- On déclare ensuite :
 - un tableau de **byte** correspondant à l'adresse MAC du shield ethernet .

L'adresse MAC est un groupe de nombre hexadécimaux permettant de donner un identifiant unique à chaque matériel utilisé sur un réseau.

- un ou plusieurs objets **IPAddress** correspondant aux différentes adresses IP de configuration utilisée. Ici, nous ne définirons que l'adresse IP locale du shield Ethernet.

ATTENTION :

il faut utiliser une adresse hors de la plage d'adresses du routeur DHCP pour connaître la plage d'adresse du routeur :
s'y connecter depuis un navigateur à l'adresse xxx.xxx.xxx.1
par exemple : sur livebox : plage adresses DHCP entre .10 et .50
=> on peut utiliser .100 pour le shield ethernet

```
// --- Inclusion des bibliothèques ---
#include <SPI.h> // bibliothèque SPI - obligatoire avec bibliothèque Ethernet
#include <Ethernet.h> // bibliothèque Ethernet

// --- Déclaration des constantes utiles ---
// --- Déclaration des constantes des broches E/S numériques ---

// --- Déclaration des constantes des broches analogiques ---

// --- Déclaration des variables globales ---

//---- l'adresse mac = identifiant unique du shield
// à fixer arbitrairement ou en utilisant l'adresse imprimée sur l'étiquette du shield
byte mac[] = { 0x90, 0xA2, 0xDA, 0x00, 0x1A, 0x71 };

//----- l'adresse IP fixe à utiliser pour le shield Ethernet ---
IPAddress ipLocal(192,168,1,100); // l'adresse IP locale du shield Ethernet
// ATTENTION : il faut utiliser une adresse hors de la plage d'adresses du routeur DHCP
// pour connaître la plage d'adresse du routeur : s'y connecter depuis un navigateur à l'adresse xxx.xxx.xxx.1
// par exemple : sur livebox : plage adresses DHCP entre .10 et .50 => on peut utiliser .100 pour le shield ethernet

//----- l'adresse IP du serveur DNS à utiliser ---
//IPAddress serveurDNS(8,8,8,8); // l'adresse du serveur DNS - Google DNS : 8.8.8.8

//----- l'adresse de la passerelle réseau ---
// la passerelle est le poste connecté à internet - typiquement le routeur à l'adresse xxx.xxx.xxx.1
//IPAddress passerelle(192,168,1,1); // l'adresse du poste de connexion à internet ou de la box

//----- masque de sous réseau
//IPAddress masque(255, 255, 255, 0 ); // idem masque sous-réseau des autres postes du réseau et de la box : 255.255.255.0 par défaut
```

Fonction **setup()**

Initialisation série

- On initialise la connexion série

Initialisation du **shield Ethernet**

- On initialise le module Ethernet avec la fonction **Ethernet.begin()**. Bien comprendre que cette fonction initialise simplement le shield Ethernet d'un point de vue matériel. A ce stade, il n'est configuré ni en serveur, ni en client.

Remarquer les différentes possibilités d'initialiser le module avec la fonction **begin()**, notamment la forme DHCP, mais qui utilise 6K de flash supplémentaire. Moi, je vous conseille la forme IP fixe simple.

Affichage de l'adresse IP du **shield Ethernet**

- On affiche l'adresse IP attribuée au module. Remarquer que l'instruction **print** supporte l'objet **IPAddress**.

```
void setup() { // debut de la fonction setup()

// --- ici instructions à exécuter 1 seule fois au démarrage du programm
e ---

// ----- Initialisation fonctionnalités utilisées -----

Serial.begin(115200); // Initialise connexion Série

//---- initialise la connexion Ethernet avec l'adresse MAC du module Eth
ernet, l'adresse IP Locale
//---- +/- l'adresse IP du serveurDNS , l'adresse IP de la passerelle i
nternet et le masque du réseau local

//Ethernet.begin(mac); // forme pour attribution automatique DHCP - util
ise plus de mémoire Flash (env + 6Ko)
Ethernet.begin(mac, ipLocal); // forme conseillée pour fixer IP fixe
locale
//Ethernet.begin(mac, ipLocal, serverDNS, passerelle, masque); // forme
complète

delay(1000); // donne le temps à la carte Ethernet de s'initialiser

Serial.print("L'adresse IP du shield Ethernet est :" );

Serial.println(Ethernet.localIP());

/* Equivalent détaillé
Serial.print(Ethernet.localIP()[0]), Serial.print(".");
Serial.print(Ethernet.localIP()[1]), Serial.print(".");
Serial.print(Ethernet.localIP()[2]), Serial.print(".");
Serial.print(Ethernet.localIP()[3]);
*/

/*
Serial.print(ipLocal[0]), Serial.print(".");
Serial.print(ipLocal[1]), Serial.print(".");
Serial.print(ipLocal[2]), Serial.print(".");
Serial.print(ipLocal[3]);
*/

} // fin de la fonction setup()
```

Fonction **loop()**

- Est laissée vide.

```
void loop() { // debut de la fonction loop()

while(1); // stop loop

} // fin de la fonction loop()
```

Fonctionnement du programme

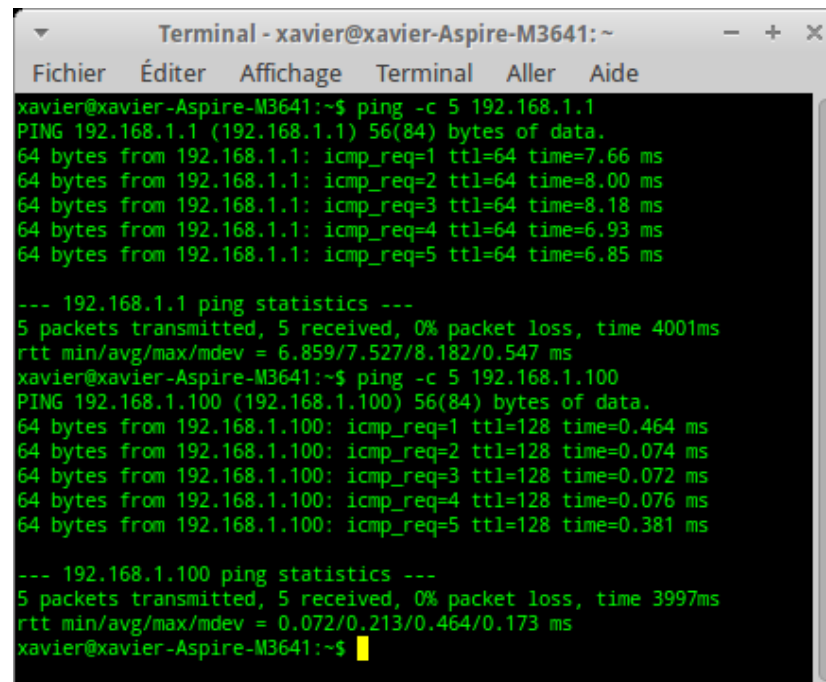
- Une fois la carte Arduino programmée : ouvrir une console système
- Sous Ubuntu (Gnu/Linux) : saisir la commande suivante (cette commande réalise un ping avec 5 envois vers l'adresse indiquée, ici le routeur)

```
ping -c 5 192.168.1.1
```

- ensuite saisir la commande (cette commande réalise un ping avec 5 envois vers l'adresse indiquée, ici le shield Ethernet)

```
ping -c 5 192.168.1.100
```

- Vous devez obtenir quelque chose comme ça dans votre console (ici, remarquer la réponse obtenue du routeur puis du shield Ethernet)



```
Terminal - xavier@xavier-Aspire-M3641: ~
Fichier  Éditer  Affichage  Terminal  Aller  Aide

xavier@xavier-Aspire-M3641:~$ 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=7.66 ms
64 bytes from 192.168.1.1: icmp_req=2 ttl=64 time=8.00 ms
64 bytes from 192.168.1.1: icmp_req=3 ttl=64 time=8.18 ms
64 bytes from 192.168.1.1: icmp_req=4 ttl=64 time=6.93 ms
64 bytes from 192.168.1.1: icmp_req=5 ttl=64 time=6.85 ms

--- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4001ms
rtt min/avg/max/mdev = 6.859/7.527/8.182/0.547 ms
xavier@xavier-Aspire-M3641:~$ ping -c 5 192.168.1.100
PING 192.168.1.100 (192.168.1.100) 56(84) bytes of data:
64 bytes from 192.168.1.100: icmp_req=1 ttl=128 time=0.464 ms
64 bytes from 192.168.1.100: icmp_req=2 ttl=128 time=0.074 ms
64 bytes from 192.168.1.100: icmp_req=3 ttl=128 time=0.072 ms
64 bytes from 192.168.1.100: icmp_req=4 ttl=128 time=0.076 ms
64 bytes from 192.168.1.100: icmp_req=5 ttl=128 time=0.381 ms

--- 192.168.1.100 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3997ms
rtt min/avg/max/mdev = 0.072/0.213/0.464/0.173 ms
xavier@xavier-Aspire-M3641:~$
```

Vous avez réussi ? Bravo, votre réseau local et votre shield Ethernet fonctionnent !

Truec : Pour obtenir plus de détails sur la commande ping, sous Gnu/Linux, saisissez la commande **man ping** dans une console.

30. Pour info : Un réseau local utilisant des blocs CPL

C'est quoi un bloc « CPL » (réseau par « courants porteurs »)

- Un bloc CPL se présente sous la forme d'un « adaptateur » qui s'enfiche sur une prise 220V et dispose d'une sortie réseau Ethernet ou même wifi !
- Il existe plusieurs modèles, les critères de choix importants étant la bande passante (utiliser 200 Mbps = 25Mo/s – permet la vidéo) et la présence d'une prise 220V en surface, sinon la prise utilisée est condamnée... ce qui est idiot et oblige à utiliser des rallonges... ! Compter 40 à 50€ le bloc.

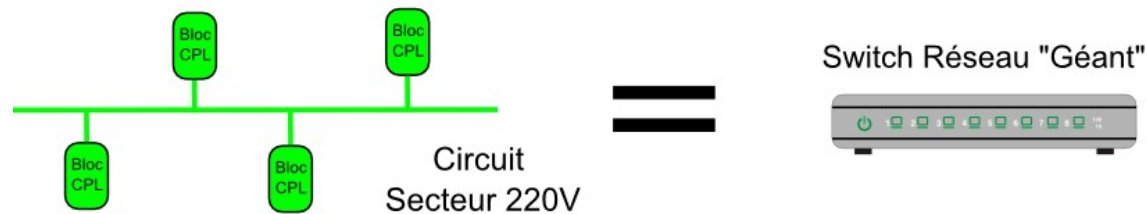
Je conseille pour l'utiliser depuis plusieurs années, la marque Devolo (<http://www.devolo.fr/>) : marque allemande garantie 3 ans, très grande stabilité du réseau, fluide, facile à paramétrer, utilitaire Gnu/Linux, permettant de faire plusieurs réseaux dans le réseau !!

Existe actuellement en 500Mbps.



Le principe

- Un bloc CPL est un outil assez génial : il permet de transformer le réseau électrique 220V de la maison en une sorte de « switch réseau » géant ! Chaque bloc CPL devient l'équivalent d'une prise ethernet d'un switch réseau (je rappelle que le switch réseau est un « multiprise réseau »)



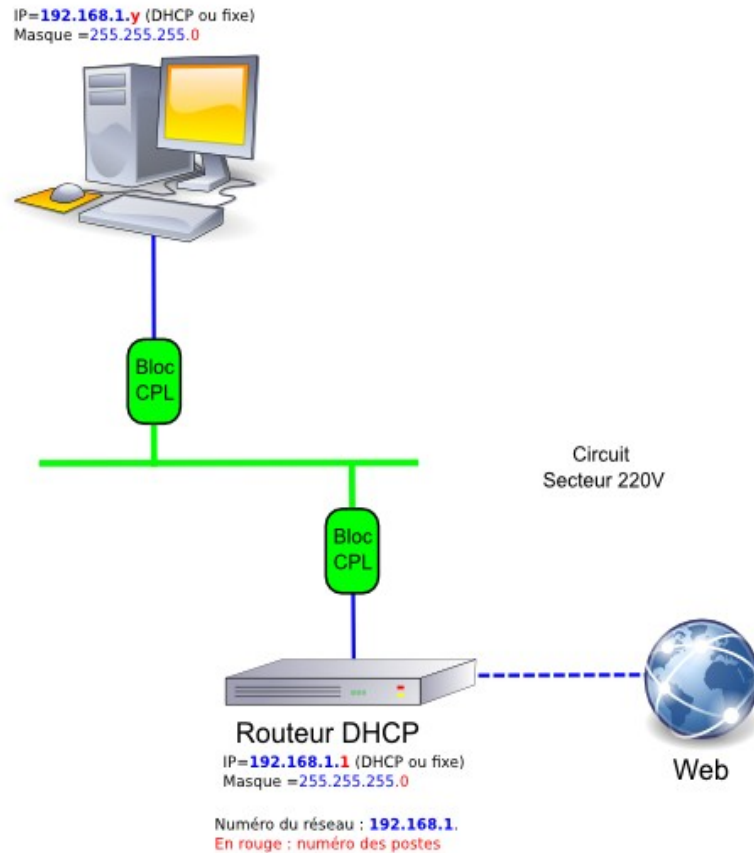
- Les avantages de ce système sont multiples :
 - pas besoin de câbles pour mettre internet ou la télévision numérique dans une pièce : un bloc CPL suffit.
 - N'importe quelle prise électrique 220V dispose ainsi potentiellement du réseau ethernet.
 - Permet également déviter l'utilisation inutile du wifi et améliore la stabilité par rapport au wifi.
 - Portée jusqu'à 200m dans une maison, répéteur possible. Protection par cryptage. Avec ça vous pourrez mettre une Arduino au grenier ou dans le garage sans problème !

Il suffit simplement de respecter quelques règles d'usage pour que tout se passe bien : notamment mettre le bloc en premier sur la prise électrique (=pas sur un multiprise) et ne pas utiliser de gros électro-ménager sur une prise voisine.

31. Réseaux local avec blocs CPL : exemples de réseaux

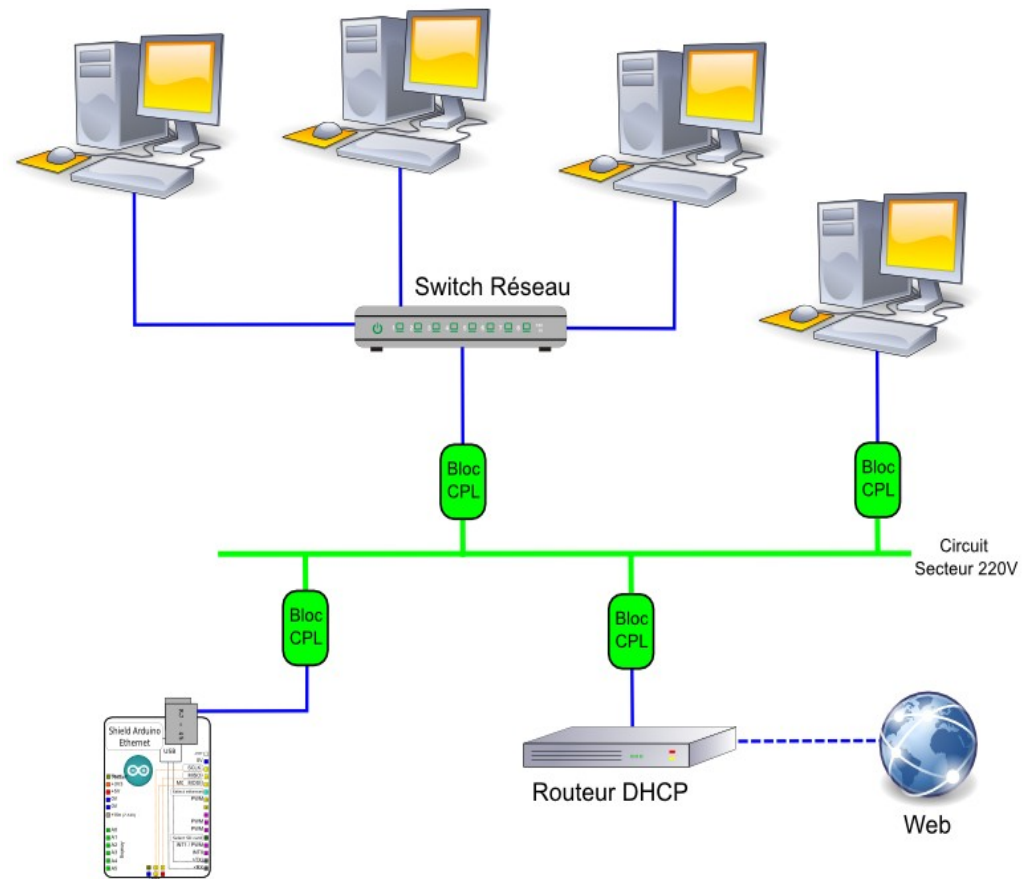
Le réseau minimum en CPL

- Il faut au moins 2 blocs CPL, chacun représentant une prise du « switch géant » avec au moins l'un d'eux connecté au routeur (obligatoire!) :



Réseau multi-postes en CPL

- On peut faire ce que l'on veut... :
 - connecter un couple Arduino + shield Ethernet derrière un bloc CPL
 - connecter plusieurs postes derrière un switch connecté à un bloc CPL
 - connecter le routeur (obligatoire !) lui-même connecté à internet
 - connecter un poste seul sur un bloc CPL
 - etc...



J'ai l'expérience de l'utilisation de 8 blocs simultanément à domicile sans problème particulier de fluidité et de stabilité de la connexion pour chacun des postes et l'utilisation de sous-réseaux connectés au réseau CPL.

A savoir : les blocs CPL permettent de créer plusieurs réseaux locaux sur le même réseau électrique !

32. Les éléments du langage Arduino étudiés dans cet atelier

Les fonctions de la librairie Ethernet

Chaque classe dispose de plusieurs fonctions associées :

Classe *Ethernet* (configuration matérielle du shield Ethernet)

- | begin() | localIP() | maintain()

Classe *EthernetServer* (serveur TCP)

- | begin() | available() | write() | print() | println()

Classe *EthernetClient* (client TCP)

- | connected() | connect() | write() | print() | println() | available() | read() | flush() | stop()

Classe *EthernetUDP* (communication UDP)

- | begin() | read() | write() | beginPacket() | endPacket() | parsePacket() | available() | remoteIP() | remotePort()

La documentation complète du langage Arduino en français est disponible ici :
http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.ReferenceMaxi

33. *A présent, vous devriez être capable :*

- Monter un réseau local par vous-même et de l'explorer, notamment connaître les IP des différents poste
- Programmer la carte Arduino couplée au Shield Ethernet pour que l'ensemble soit reconnu sur le réseau

Table des matières

Introduction au réseau local et apprendre à monter un réseau local avec Arduino.

Intro |

Matériel utile ou nécessaire pour cet atelier |

Matériel spécifique nécessaire pour cet atelier (suite) |

Un réseau local : ... c'est un peu comme une équipe de foot ! |

Au sein du réseau local : le n°1, un joueur un peu particulier ! |

Note technique : comment savoir quels numéros de l'adresse IP représentent la « couleur du maillot » ? |

Exercices d'entraînement |

Technique : Réseau : Comprendre les adresses IP : Synthèse |

Fonctionnement du réseau |

Techniquement : structure d'un vrai réseau et matériel nécessaire |

Synthèse : Un peu de vocabulaire pour avoir les idées claires |

Monter le réseau local minimum |

Tester et explorer le réseau : mise sous tension et récupération de l'adresse IP du poste fixe |

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

Tester le réseau : Se connecter à l'interface du routeur et connaître la plage des adresses DHCP |

Monter un réseau multi-postes : utiliser un « switch » ethernet si besoin... |

Truc avancé : connaître tous les postes présents sur un réseau local à partir d'un poste fixe connecté au réseau ! |

Les IP statiques : « allez, on fait tout à la main... ! » |

Un cas à part : « le gars qui se pointe avec son maillot déjà sur lui »... |

Et en wifi ??? |

Monter un réseau local avec Arduino : La structure du réseau que nous allons réaliser |

Les éléments du réseau local avec Arduino que nous allons utiliser |

Le shield Ethernet : description et principe d'utilisation |

Monter le réseau utilisant le shield Ethernet Arduino |

Monter le réseau utilisant le shield Ethernet Arduino sur un réseau avec « box » existant |

Rappel : Langage Arduino : Introduction aux bibliothèques |

La bibliothèque Ethernet |

Faire un simple « ping » vers le shield Ethernet à partir d'un poste fixe : le réseau |

Faire un simple « ping » vers le shield Ethernet à partir d'un poste fixe : le programme |

Pour info : Un réseau local utilisant des blocs CPL |

Réseaux local avec blocs CPL : exemples de réseaux |

Les éléments du langage Arduino étudiés dans cet atelier |

A présent, vous devriez être capable : |

Bravo !
vous avez terminé cet atelier Arduino !



Prêt pour la suite ? Retrouvez de nombreux autres thèmes d'ateliers Arduino ici :
http://www.mon-club-elec.fr/pmwiki_mon_club_elec/pmwiki.php?n=MAIN.ATELIERS