

Arduino Yún

Table des matières

1. Introduction.....	2
1.1. Présentation.....	2
1.2. Caractéristiques.....	2
1.3. Wifi.....	2
2. Bridge, un pont entre GNU/Linux et Arduino.....	3
2.1. Appels REST.....	4
2.2. La Console.....	6
2.3. Les commandes Process.....	7
3. Installer les pilotes pour Yún.....	8
3.1. Windows.....	8
3.2. GNU/Linux.....	10
4. Configuration du WiFi de la carte.....	10
5. Mise en place d'un serveur web.....	12

Copyright ©, 01/2016 Lorenzo Leijnen

La platine Arduino YUN combine la puissance d'un transceiver radio WLAN avec système d'exploitation Linino (MIPS GNU/Linux basé sur OPENWRT) avec la facilité d'utilisation d'un Arduino (base similaire au Leonardo avec processeur ATmega32u4).



1. Introduction

1.1. Présentation

La carte Arduino Yún est bâtie sur la base d'un Arduino Leonardo auquel a été rajouté en parallèle un processeur Atheros AR9331 qui fait tourner la distribution GNU/Linux Linino, basée sur [OpenWrt](#).

1.2. Caractéristiques

Micro-contrôleur : ATMEGA32U4

Tension de fonctionnement nominale : 5V

Entrées/sorties digitales : 14 (dont 7 pouvant être utilisées comme sorties PWM)

Voies d'entrée analogique : 6 (plus 6 multiplexé sur 6 broches numériques)

Courant par I / O Pin : 40 mA

Mémoire Flash : 32 Ko (ATMEGA32U4) dont 4 Ko utilisé par bootloader

SRAM : 2,5 KB (ATMEGA32U4)

EEPROM : 1 Ko (ATMEGA32U4)

Vitesse d'horloge : 16 MHz

Processeur : MIPS 24K fonctionnant jusqu'à 400 MHz

Mémoire : DDR2 64Mo de RAM et 16 Mo Flash SPI

AP ou routeur : Complete IEEE 802.11bgn 1x1

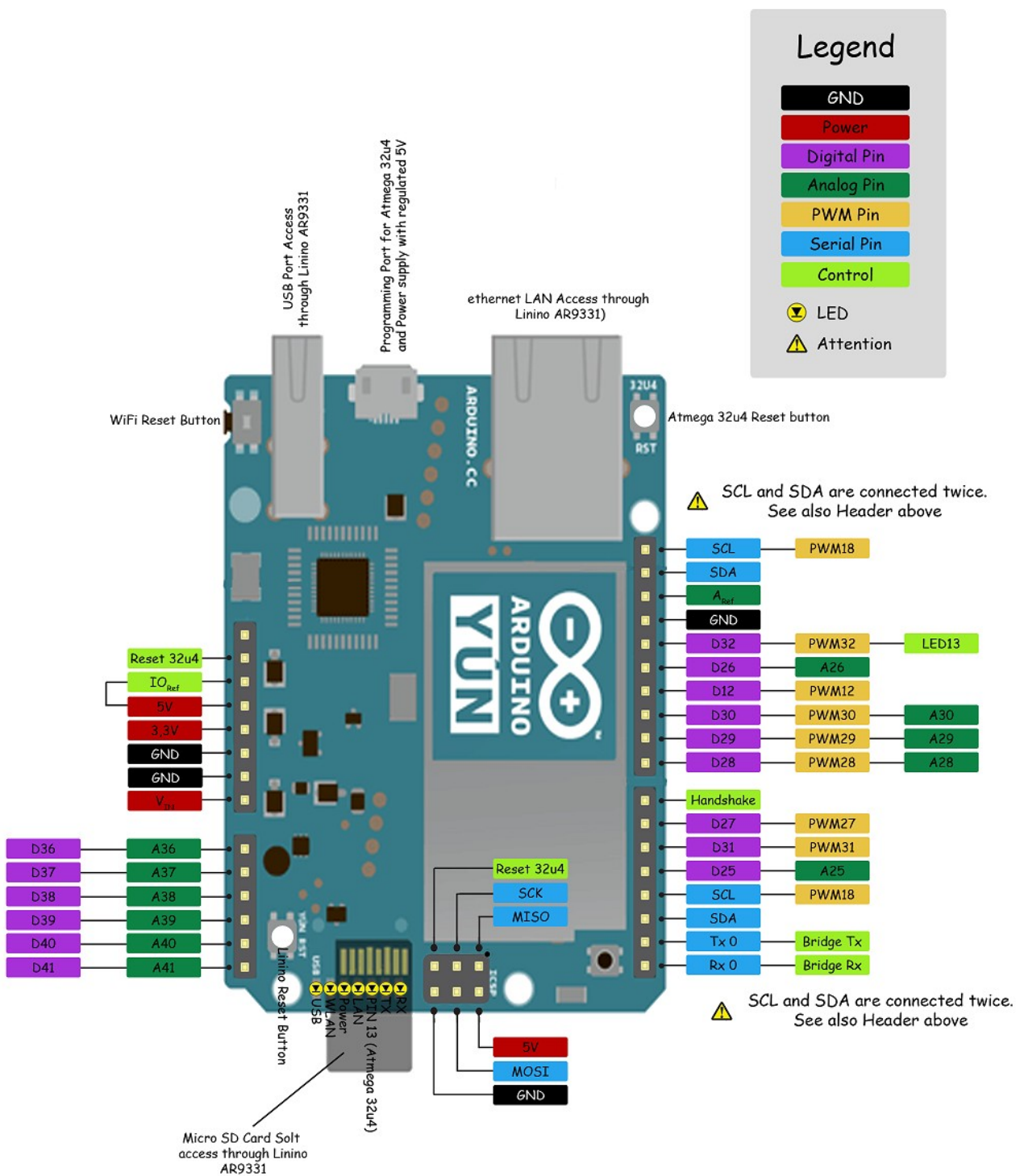
Hôte / Périphérique : USB 2.0

MicroSD : PoE 802.3af support de carte compatible

1.3. Wifi

Le réseau WIFI est configuré par défaut en mode point d'accès : une fois l'environnement GNU/Linux démarré, un nouveau réseau WIFI est disponible, sous le nom « Arduino Yun-XXXXXXXXXXXX », où « XXXXXXXXXXXXXXX » représente l'adresse MAC du contrôleur WIFI. L'interface web de configuration d'OpenWrt est ensuite accessible à l'adresse « <http://arduino.local/> ». Il est également possible de configurer OpenWrt pour se connecter à un réseau WIFI existant : dans ce cas, il se verra affecter une adresse IP par DHCP. L'interface web de configuration d'OpenWrt sera du coup accessible à l'adresse « [http://<ip>/](http://<ip>) ». Ce deuxième cas est aussi valable pour une connexion au réseau local via l'interface Ethernet.

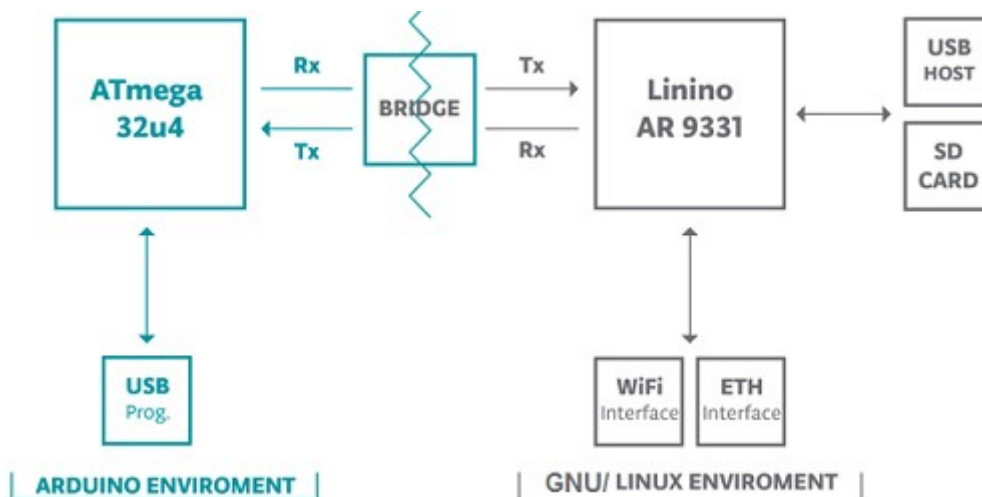
Côté GNU/Linux, c'est une distribution dérivée d'OpenWrt qui était à l'origine destinée à remplacer les firmwares des routeurs WIFI Linksys de la gamme WRTxxx (d'où le nom). Elle a ensuite été adaptée à d'autres plate-formes. L'interface de configuration de la distribution GNU/Linux est accessible via la page web mentionnée précédemment ou bien via un accès SSH. Elle est extensible via un système de paquets logiciels.



2. Bridge, un pont entre GNU/Linux et Arduino

L'environnement Arduino et l'environnement GNU/Linux ne sont pas deux mondes complètement isolés les uns des autres. Les deux peuvent communiquer à l'aide de la bibliothèque « Bridge », présente à la fois du côté Arduino, et du côté GNU/Linux (via une bibliothèque Python). Concrètement, les deux environnements communiquent entre eux selon les couches présentées ci-dessous (du plus bas niveau au plus haut) :

1. Un lien série entre les deux processeurs.
2. La bibliothèque « Bridge », qui est la partie bas niveau du protocole de communication.
3. D'autres bibliothèques spécifiques, qui s'appuient sur « Bridge » :
 - Console : une émulation du terminal série, mais qui fonctionne par dessus le réseau.
 - Process : permet de lancer des commandes sur l'environnement GNU/Linux, depuis l'environnement Arduino, avec récupération de la sortie de ces commandes.
 - FileIO : permet à l'environnement Arduino de lire et écrire des fichiers sur la carte micro SD.
 - YunClient : un client HTTP simple.
 - YunServer : un serveur HTTP simple.
 - Temboo : un SDK permettant d'accéder à une centaine de services web.
 - Spacebrew : un système de communication inter-objets basé sur WebSockets.



Cette communication bidirectionnelle entre les deux processeurs permet (entre autre) de déléguer une partie de la couche logicielle réseau à l'environnement Linux, et par là de soulager les ressources de l'environnement Arduino. Dans le cas d'un serveur web, l'environnement Arduino n'a pas à générer les entêtes HTTP : il n'envoie que la partie « utile » à l'environnement GNU/Linux, et c'est ce dernier qui effectuera le formatage final avant renvoi au navigateur client.

De plus, tout fichier présent dans un répertoire nommé « /arduino/www/ » sur la carte micro SD est automatiquement mis à disposition par le serveur web à l'adresse « <http://<ip>/sd/> ». Il est également possible d'exécuter des scripts CGI¹ (shell, PHP, etc.) pour avoir des pages dynamiques.

2.1. Appels REST

De même, tout accès à l'adresse « <http://<ip>/arduino/ce/que/vous/voulez> » transfère automatiquement la chaîne de caractère « ce/que/vous/voulez » à l'Arduino. Il faut ensuite que le code tournant sur le micro-contrôleur utilise la bibliothèque « YunServer » pour traiter la chaîne de caractère, et « YunClient » pour d'éventuels retours vers le navigateur. Avec ce système, vous pouvez créer votre propre API REST² pour communiquer avec l'Arduino à travers le réseau.

¹ Common Gateway Interface

² Application Programming Interface Representational State Transfer

Sur le Yún, il y a deux point d'accès REST (REST end points) reconnu :

- /arduino
- /data

Le répertoire "/arduino" n'est pas pré-configuré. Tout ce qui est ajouté dans l'URL derrière le point d'accès est transféré par le Serveur Web au sketch/croquis fonctionnant sur le 32U4.

Le répertoire "/data" est utilisé pour fournir un accès à un stockage interne de type Clé/Valeur (key/value). Les appels possible sont :

- /put/KEY/VALUE : stocke une valeur value pour le clé key.
- /get/KEY : obtenir la valeur de la clé KEY. Retour au format JSON
- /get : obtenir une liste de éléments stockés au format JSON.
- /delete : efface le contenu du stockage interne

Un programme d'exemple fourni avec l'IDE³ propose d'interroger les entrées/sorties de l'Arduino ou bien de modifier leur état à l'aide d'une API REST : « <commande>/<broche>/<valeur> ».

Exemples :

- http://<ip>/arduino/digital/13 : lit la valeur de l'entrée/sortie numérique 13 (0 ou 1).
- http://<ip>/arduino/digital/13/1 : met la valeur de la sortie numérique à 1.
- http://<ip>/arduino/analog/0 : lit la valeur de l'entrée analogique 0 (de 0 à 1023).
- http://<ip>/arduino/mode/13/input : règle l'entrée / sortie 13 en mode entrée.
- http://<ip>/arduino/mode/13/output : règle l'entrée / sortie 13 en mode sortie.

Du côté Arduino, on a le code suivant :

```
#include <Bridge.h>
#include <YunServer.h>
#include <YunClient.h>

// Démarre une instance de YunServer, qui réceptionnera
// toutes les requêtes transférées par le serveur web côté GNU/Linux
YunServer server;

void setup()
{
    // Démarre une instance du Bridge
    Bridge.begin();

    // Lance l'écoute du YunServer sur le réseau
    server.listenOnLocalhost();
    server.begin();
}

void loop()
{
    // Réceptionne les éventuels clients
    YunClient client = server.accept();

    // Un client s'est-il connecté ?
    if (client) {
        // On traite la requête
        process(client);
    }
}
```

3 Integrated Development Environment

```

        // On ferme la connexion et libère les ressources
        client.stop();
    }

    delay(50); // Pause de 50ms, puis retour au début.
}

void process(YunClient client)
{
    // Lit la commande
    String command = client.readStringUntil('/');

    // Est-ce la commande "digital" ?
    if ( command == "digital" ) {
        // Écriture de la réponse vers le navigateur
        client.print("Commande traitée");
    }

    // Est-ce la commande "analog" ?
    if ( command == "analog" ) {
        //...
    }

    // Est-ce la commande "mode" ?
    if ( command == "mode" ) {
        //...
    }
}

```

2.2. La Console

La Console, basée sur Bridge, permet d'envoyer des informations depuis Yún vers un ordinateur exactement comme si vous utilisez le moniteur série sauf que c'est sans fil. La Console crée une connexion SSH sécurisée entre Yún et votre ordinateur.

Téléversez le Croquis suivant sur votre Yún : Les interfaces WiFi et Ethernet, le port USB et support de carte SD sont tous connectés sur le AR9331. La librairie Bridge vous permet de travailler avec ces périphériques, d'exécuter des scripts et de communiquer avec des WebServices.

```

#include <Console.h>

const int ledPin = 13; // La broche sur laquelle la LED est branchée

void setup()
{
    // Initialisation de la communication série:
    Bridge.begin();
    Console.begin();

    while ( !Console )
    {
        ; //Attendre la connexion du port Console.
    }
    Console.println("Vous êtes connecté sur la Console !");

    // Initialiser la broche de la LED comme sortie:
    pinMode(ledPin, OUTPUT);
}

void loop()
{
    // Vérifier s'il y a une donnée entrante:

```

```

if ( Console.available() > 0 ) {
  // Lire le byte (l'octet) le plus vieux stocké dans
  // la mémoire tampon série (serial buffer):
  switch ( Console.read() ) {
    case 'H': // Si c'est un H majuscule (ASCII 72), allumer la LED:
      digitalWrite(ledPin, HIGH);
      break;
    case 'L': // Si c'est un L (ASCII 76), éteindre la LED:
      digitalWrite(ledPin, LOW);
      break;
    default:
      break;
  }
}
}

```

Pour voir la Console, sélectionner le nom de votre Yun (+ son adresse IP) dans le menu "Ports".

Le Yun ne s'affiche dans le menu "Ports" que si votre ordinateur est sur le même réseau (LAN) que votre Yun. Vous ne verrez pas la carte dans le menu "Ports série" si votre carte Yun se trouve sur un réseau différent.

Ouvrez le Moniteur (rebaptisé "Port Monitor" plutôt que "Serial Monitor"). Vous serez invité à entrer le mot de passe de votre Yun.

Vous pouvez également accéder à la Console en ouvrant une fenêtre terminal et en tapant la commande:

```
# ssh root@yourYunsName.local 'telnet localhost 6571'
```

Note: Si vous utilisez Windows alors vous devrez installer un émulateur de terminal. PuTTY est un choix raisonnable mais vous devrez entrer les deux commandes ci-dessus séparément.

Tapez 'H' pour allumer la LED sur la broche 13 et 'L' pour l'éteindre.

2.3. Les commandes Process

Les commandes Process permettent d'exécuter des processus GNU/Linux sur Linino par l'intermédiaire d'Arduino.

Dans l'exemple suivant, Linino va se connecter sur un serveur en utilisant [curl](#) et de télécharger du texte ASCII. Il affiche ensuite le texte sur la connexion série.

```

#include <Process.h>

void setup()
{
  // Initialisation du Bridge
  Bridge.begin();

  // Initialisation du port série
  Serial.begin(9600);

  // Attendre qu'un moniteur série soit connecté.
  while ( !Serial );

  // Exécuter les différents processus d'exemple
  runCurl();
}

void loop()
{
  // Rien à faire.
}

```

```

}

void runCurl ()
{
  // Execute la commande "curl" et obtenir un logo Arduino en "ascii art" depuis le Net
  // curl est un programme en ligne de commande pour transférer des données en utilisant
  // différents protocoles Internet
  Process p; // Créer un processus et appelons le "p"
  p.begin("curl"); // Processus qui démarre la commande "curl"
  p.addParameter("http://arduino.cc/asciilogo.txt"); // Ajouter le paramètre URL à "curl"
  p.run(); // Exécuter le processus et attendre sa fin d'exécution

  // Afficher le logo Arduino sur le port série
  // La sortie d'un processus ('process output') peut être lue à l'aide
  // des méthodes de streaming (gestion de flux)
  while ( p.available() > 0 ) {
    char c = p.read();
    Serial.print(c);
  }

  // S'assurer que les derniers bits de données soit bien envoyés
  Serial.flush();
}

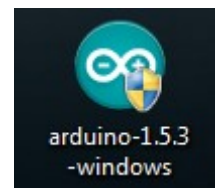
```

3. Installer les pilotes pour Yún

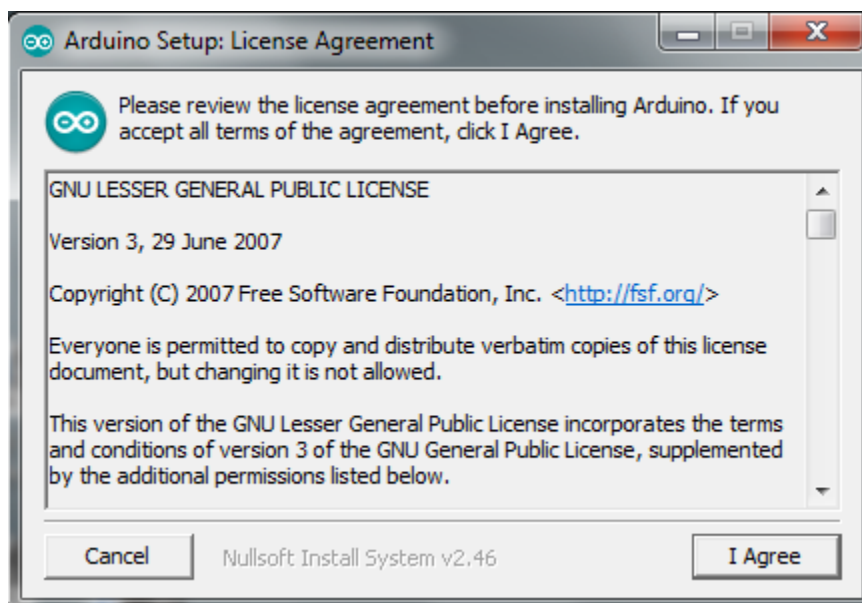
3.1. Windows

Windows dispose maintenant d'un programme d'installation pour l'IDE et les pilotes (drivers).

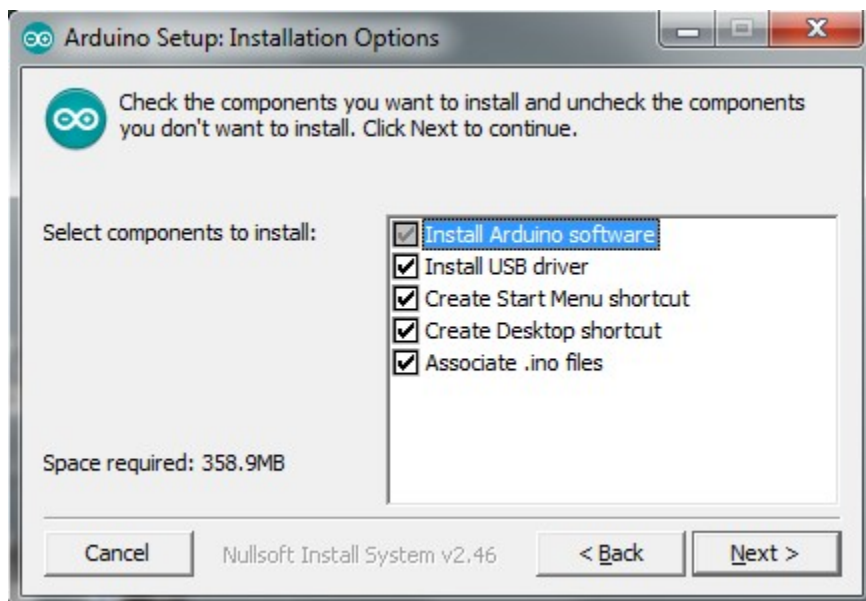
Téléchargez le programme d'installation "Arduino IDE 1.5.4" ou plus récent et double cliquez sur l'icone d'installation.



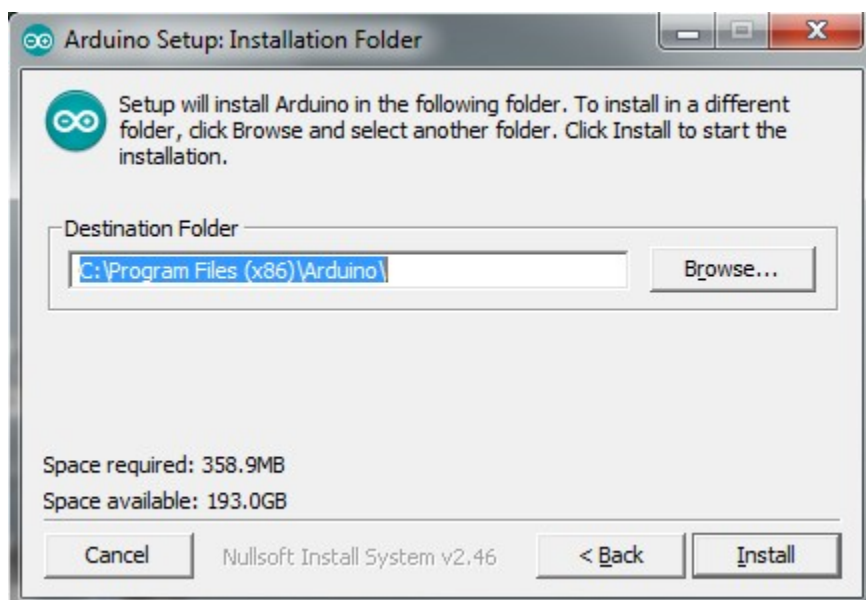
Une fois que vous avez pris connaissance de la licence, et que vous êtes en accord avec elle, cliquez sur le bouton "I agree" ("Je suis d'accord").



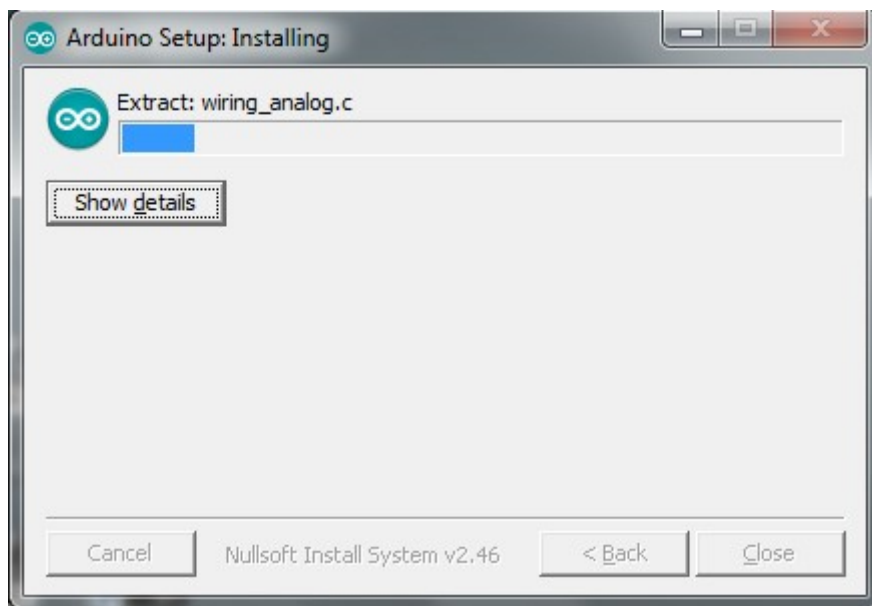
Tous les éléments sont sélectionnés par défaut. Cela inclus l'IDE, les pilotes et les raccourcis.



Sélectionnez l'endroit où vous désirez installer l'IDE (environnement de développement).



Le programme d'installation affiche une barre de progression pendant qu'il extrait et installe les fichiers à l'emplacement correct.



Une fenêtre de confirmation s'affichera si vous décidez d'installer les pilotes. Vous devrez confirmer votre sélection :



Lorsque l'installation est terminée, vous pouvez presser le bouton "Close" (Fermer ou Terminer).

NB: Yún utilise le service Bonjour pour permettre sa détection automatique sur les réseaux WiFi. Ce service n'est pas inclus par défaut dans les distributions de Windows. Si Bonjour n'est pas encore installé sur votre système, vous pouvez le télécharger [ici](#). Si vous disposez d'un logiciel anti-virus ou logiciel firewall, configurez le pour ne pas bloquer les communications sur le port 5353.

3.2. GNU/Linux

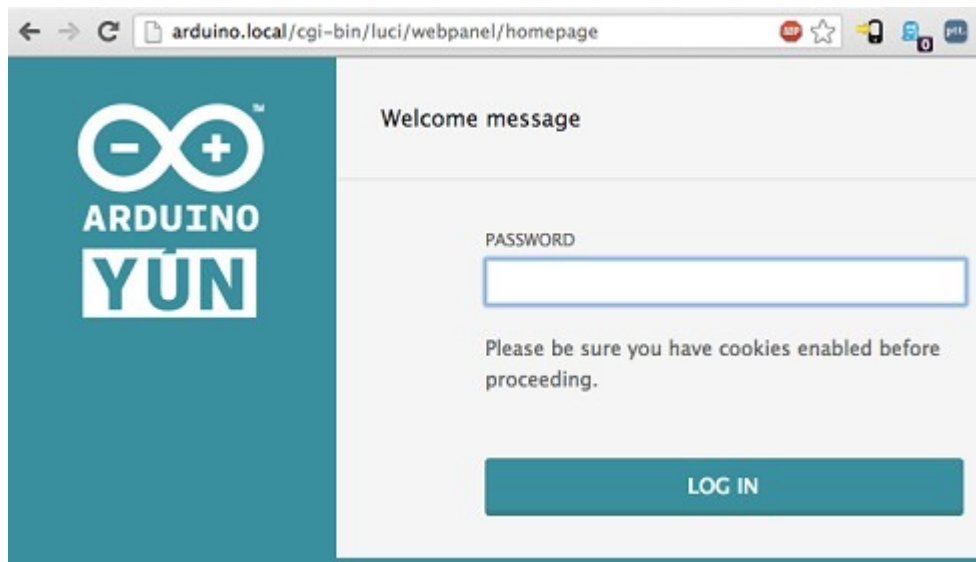
Il n'est pas nécessaire d'installer de pilotes pour Ubuntu 10.0.4 ou suivant.

4. Configuration du WiFi de la carte

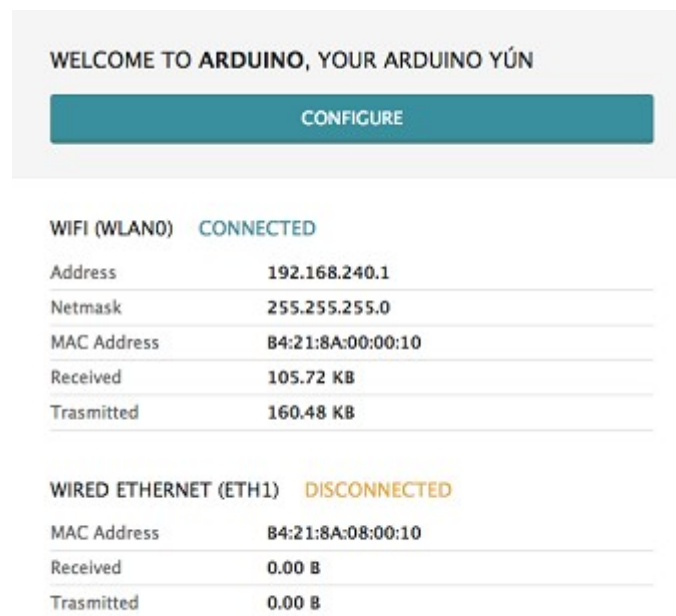
Le Yún a la possibilité d'agir comme un point d'accès (Access Point) mais il peut également se connecter sur un réseau existant. Ces instructions vous guident entre les différentes étapes vous permettant de connecter votre Yún sur un réseau sans fil. Le Yún peut se connecter sur des réseaux non protégés (non cryptés) ainsi que sur des réseaux supportant le chiffrement WEP, WPA et WPA2.

Lorsque vous mettez votre Yùn sous tension la première fois, il va créer un réseau WiFi nommé ArduinoYun-XXXXXXXXXXXX. Connectez votre ordinateur sur ce réseau.

Une fois que vous avez obtenu un adresse IP, ouvrez un navigateur Internet et entrez l'adresse <http://arduino.local> ou 192.168.240.1 dans la barre d'adresse. Au bout d'un moment, une page Web doit s'afficher, elle vous demande un mot de passe d'accès (password). Entrer "arduino" et cliquer sur le bouton "Log In" (se connecter).



Vous trouverez alors une page contenant des informations de diagnostics relative aux connexions réseaux actuelles. En premier l'interface Wifi et en second votre connexion Ethernet. Pressez le bouton "Configuration" pour continuer.



Vous configurerez votre Yùn sur cette nouvelle page, lui donnant un nom unique et en identifiant le réseaux sur lequel vous voulez vous connecter.

Placez le nom de votre Yùn dans le champ NAME, donnant ainsi un nom unique (qui n'existe pas encore sur le réseau) à votre Arduino. Vous l'utiliserez par la suite pour vous connecter sur votre Yùn.

Choisissez un mot de passe pour votre Arduino, il doit avoir au moins 8 caractères. Si vous laissez cette zone vide, le système maintiendra le mot de passe par défaut de votre Arduino.

Si vous le désirez, vous pouvez également configurer la fuseau horaire (Timezone en anglais) et le pays (country). Il est recommandé de configurer ces options puisqu'elles peuvent faciliter la connexion sur votre réseau Wifi local. Sélectionner un fuseau horaire (timezone) permet de présélectionner le domaine de réglementation qui y est applicable (country's regulatory domain).

Entrez le nom du réseau WiFi sur lequel vous désirez vous connecter dans la zone "Wireless Name".

Sélectionnez le type de sécurité (security type) et entrez le mot de passe (password) applicable à cette connexion Wifi.

Lorsque vous pressez le bouton "Configure & Restart", votre Arduino va se réinitialiser tout seul et se connecter sur le réseau spécifié. Le réseau Arduino (le point d'accès) va s'interrompre au bout d'un moment.

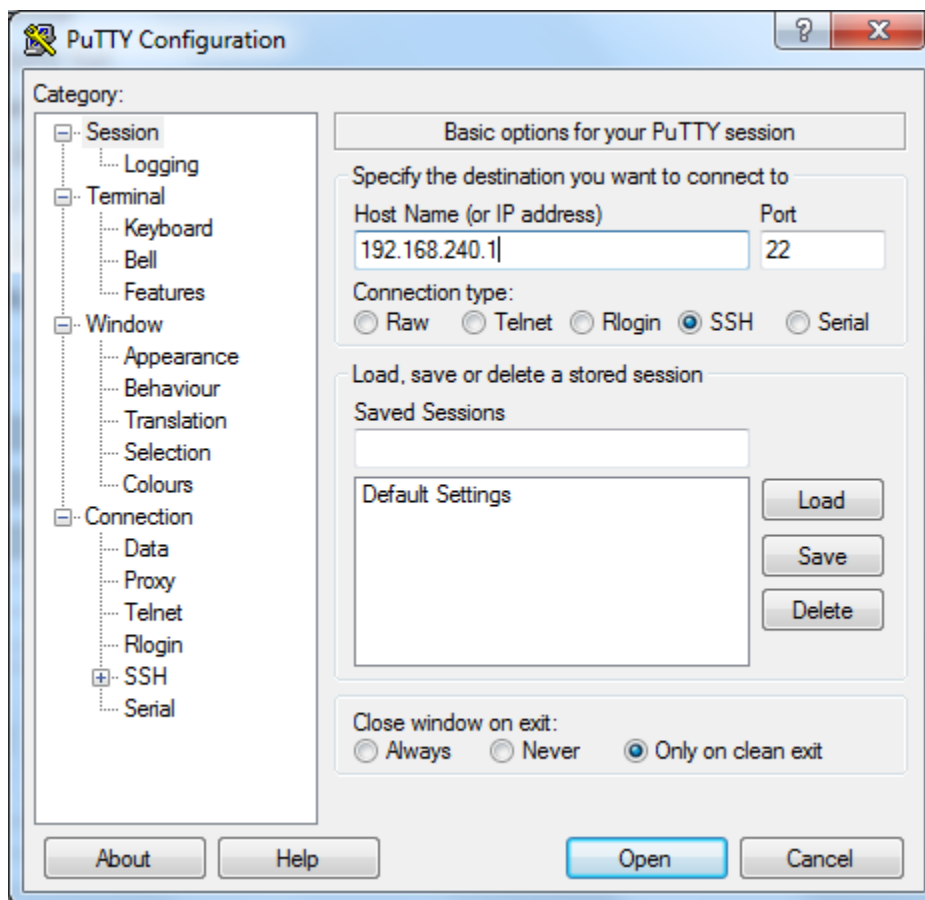
Note : Pressez le bouton "Discard" si vous ne désirez pas appliquer ces nouveaux paramètres sur votre Yùn.

Le message "Configuration Saved!" indique que votre Yùn a enregistré les nouveaux paramètres. Ce message vous indique qu'il fait également le nécessaire pour se connecter sur le réseau que vous avez mentionné.

Maintenant, vous pouvez vous connecter sur le réseau que vous avez assigné à votre Yùn.

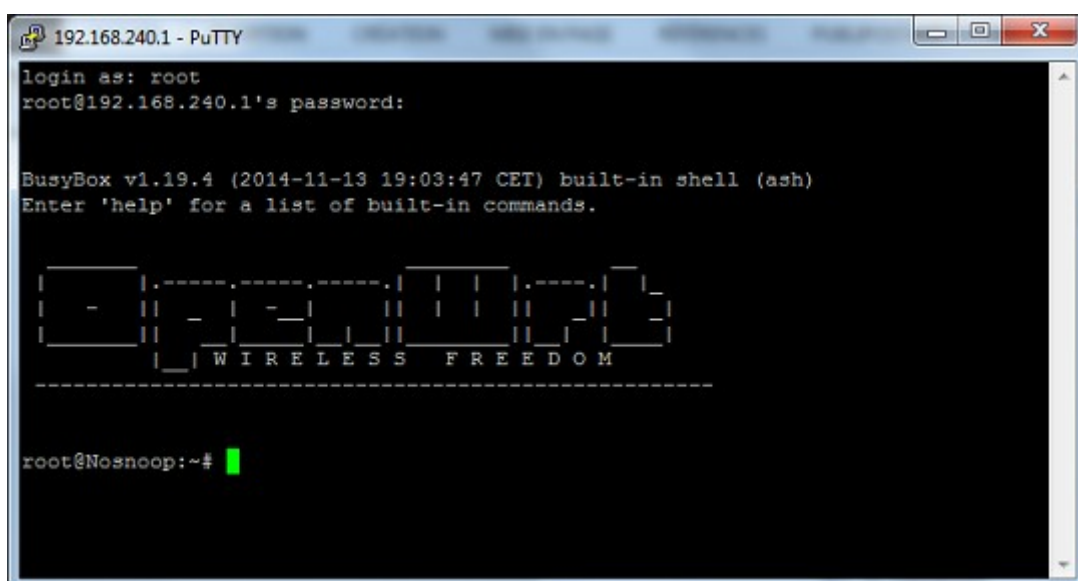
5. Mise en place d'un serveur web

La carte Yùn doit être au préalable connectée au réseau local (en wifi ou par connexion filaire). Sous Windows, pour vous connecter sur le système GNU/Linux de la carte Yùn, lancez le programme [PuTTY](#) à partir de votre PC. Vous obtenez ceci :



Entrez l'adresse IP de la carte, sélectionnez la connexion de type SSH sur le port 22 (port par défaut de la Yun si vous n'avez pas modifié). Cliquez sur « Open ».

Une fenêtre noire (console) apparaît. Vous devez vous connecter en tant que « root » (super utilisateur) avec le mot de passe de la carte (« arduino » si vous n'avez pas changé). Voici le résultat :



A la racine, créez un dossier www, dans lequel vous mettrez votre site :

```
# mkdir /www
```

Un raccourci vers ce dossier sera créé à la base de l'arborescence de fichiers et un serveur web est automatiquement démarré. Néanmoins, ce dernier ne prend pas en charge le PHP, ni MySQL... Nous allons donc installer des packages afin d'ajouter ces composants.

Vous allez à présent télécharger ces packages requis pour mettre en place un serveur FTP, PHP et MySQL Lite (le serveur FTP étant très pratique dans la mesure où on pourra directement accéder à la carte SD sans devoir la déconnecter). Pour ce faire, allez sur cette [page de téléchargement](#).

Voici la liste des packages dont on aura besoin :

- openssh-sftp-server_6.1p1-1_ar71xx.ipk
- php5_5.4.5-3_ar71xx.ipk
- php5-cgi_5.4.5-3_ar71xx.ipk
- php5-mod-sqlite3_5.4.5-3_ar71xx.ipk

Ces packages sont indispensables. Par la suite vous pourrez installer des modes en fonction de vos besoins tels que php5-mod-fileinfo_5.4.5-3_ar71xx.ipk par exemple.

Une fois téléchargés sur votre PC, copiez les sur la carte SD à l'aide [du logiciel WinSCP](#). Un redémarrage du 32U4 peut être nécessaire pour mettre à jour le contenu de la carte (poussez le bouton à côté du port Ethernet deux fois).

Exécutez les commandes suivantes :

```
# cd /  
# ls
```

Si vous avez bien créé le dossier www, celui-ci devrait apparaître dans la liste. Pour le moment, on ne s'en servira pas. Si le dossier n'apparaît pas, vérifiez que vous avez bien créé le dossier ou redémarrez la carte.

```
# cd mnt  
# cd sd1  
# opkg install php5_5.4.5-3_ar71xx.ipk  
# opkg install le_nom_des_autres_packages
```

Remarque : Ne paniquez pas si votre fenêtre n'est pas identique à la mienne. Dans cet exemple, la carte en question s'appelle « NOSNOOP », et la carte SD comporte d'autres packages que ceux cités précédemment.

Si les packages ne sont pas à la racine de la carte SD, tapez la commande ls pour afficher tous les dossiers, puis cd pour s'y déplacer. Voici ce que vous devrez avoir à peu près :

```

root@Nosnoop:~# cd ..
root@Nosnoop:/# ls
bin      etc      mnt      proc     root     sys      usr      www
dev      lib      overlay  rom      sbin     tmp      var
root@Nosnoop:/# cd mnt
root@Nosnoop:/mnt# cd sda1
root@Nosnoop:/mnt/sda1# cd packages
root@Nosnoop:/mnt/sda1/packages# ls
libevent2-extra_2.0.19-1_ar71xx.ipk      php5-mod-curl_5.4.5-3_ar71xx.ipk
libevent2_2.0.19-1_ar71xx.ipk           php5-mod-fileinfo_5.4.5-3_ar71xx.ipk
libmagic_5.11-1_ar71xx.ipk              php5-mod-gmp_5.4.5-3_ar71xx.ipk
libsqlite3_3071201-1_ar71xx.ipk         php5-mod-hash_5.4.5-3_ar71xx.ipk
libxml2_2.7.8-2_ar71xx.ipk              php5-mod-http_5.4.5-3_ar71xx.ipk
libyaml_0.1.3-1_ar71xx.ipk              php5-mod-openssl_5.4.5-3_ar71xx.ipk
libzip_0.10.1-1_ar71xx.ipk              php5-mod-session_5.4.5-3_ar71xx.ipk
openssh-sftp-server_6.1p1-1_ar71xx.ipk  php5-mod-sqlite3_5.4.5-3_ar71xx.ipk
php5-cgi_5.4.5-3_ar71xx.ipk             php5-mod-xml_5.4.5-3_ar71xx.ipk
php5-fastcgi_5.4.5-3_ar71xx.ipk         php5-mod-zip_5.4.5-3_ar71xx.ipk
php5-mod-calendar_5.4.5-3_ar71xx.ipk    php5_5.4.5-3_ar71xx.ipk
php5-mod-ctype_5.4.5-3_ar71xx.ipk
root@Nosnoop:/mnt/sda1/packages# opkg install php5-mod-zip_5.4.5-3_ar71xx.ipk
Installing php5-mod-zip (5.4.5-3) to root...
Configuring php5-mod-zip.
root@Nosnoop:/mnt/sda1/packages#

```

Attention : Certains packages peuvent être dépendant d'autres packages. Si cela se produit, repérez dans le code erreur le package concerné, téléchargez-le, et installez-le afin de pouvoir installer le package de départ.

Maintenant, une fois que tout a été installé, on va configurer le serveur ftp. Voici les commandes :

```

# uci set uhttpd.main.interpreter=".php=/usr/bin/php-cgi"
# uci set uhttpd.main.index_page="index.html index.htm default.html default.htm index.php"
# uci commit uhttpd
# sed -i 's,doc_root.*,doc_root = "",g' /etc/php.ini
# sed -i 's,;short_open_tag = Off,short_open_tag = On,g' /etc/php.ini

```

Et on redémarre le service uhttpd :

```

# /etc/init.d/uhttpd restart

```

Enfin, pour vérifier si votre serveur opérationnel, téléversez le code suivant sur la carte avec l'IDE d'arduino :

```

#include <Bridge.h>
#include <YunClient.h>
#include <YunServer.h>

const int led = 13;

YunServer server;

void setup()
{
    //Start bridge
    pinMode(led, OUTPUT);
    digitalWrite(led, LOW);

    //Start server
    server.listenOnLocalhost();
}

```

```
server.begin();

Bridge.begin();
blinkTwice(led);
}

void loop()
{
    //Welcome client
    YunClient client = server.accept();

    if ( client ) {
        blinkTwice(led);
        client.stop();
    }

    //Repeat every 50ms
    delay(50);
}

void blinkTwice(const int pin)
{
    for (int i(0); i < 2; i++) {
        digitalWrite(pin, HIGH);
        delay(300);
        digitalWrite(pin, LOW);
        delay(700);
    }
}
```

Dans la barre d'adresse de votre navigateur, indiquez l'adresse suivante : <http://arduino.local/sd>
La LED de la carte doit s'allumer et s'éteindre avec une période de 1s.