

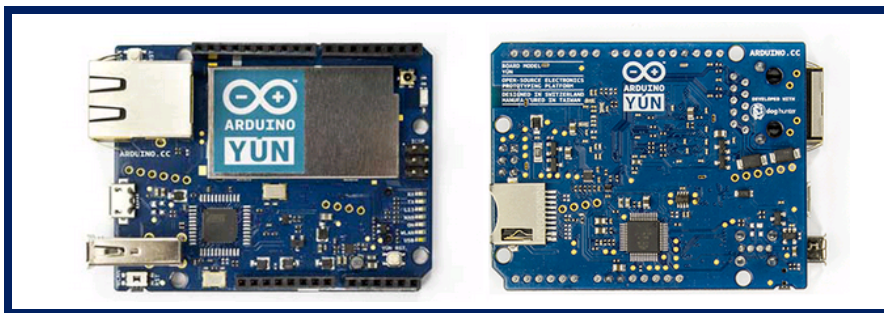


STI2D – Enseignement de spécialité SIN

CARTE ARDUINO YUN

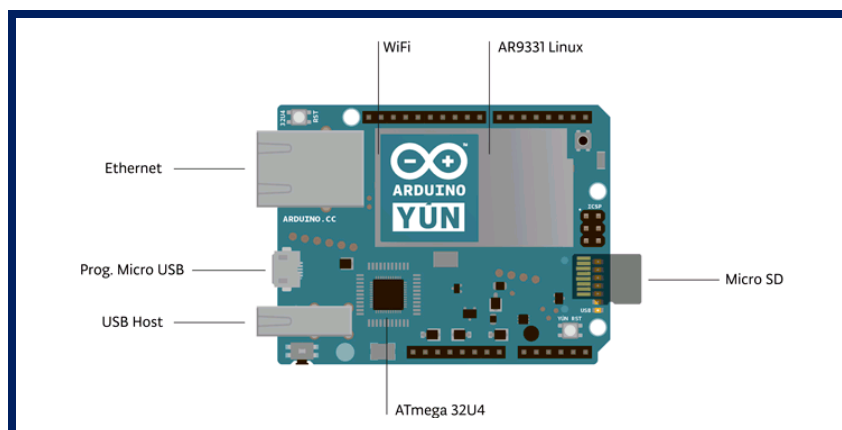
1 – PRESENTATION

La carte Arduino Yùn est une carte similaire à la carte Arduino Léonardo mais qui possède des fonctionnalités avancées permettant de réaliser des prototypes d'objets connectés. Elle dispose du **WiFi intégré**, d'un contrôleur **Ethernet** et un port RJ45 ainsi qu'un port USB host permettant la connexion de périphériques. Un slot pour carte SD permet le stockage externe de données.

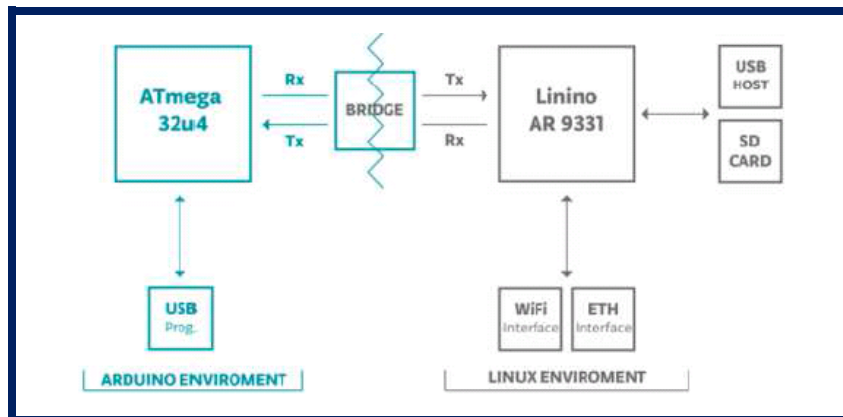


2 – CONSTITUTION

L'Arduino Yùn utilise le même microcontrôleur que la carte Arduino Léonardo, l'Atmel ATmega32U4. Elle comporte également un **microprocesseur additionnel**, Atheros AR9331, sur lequel tourne la distribution Linux pour systèmes embarqués appelée Linio basée sur OpenWrt wireless stack.



Le microcontrôleur ATmega32U4 et le microprocesseur Atheros AR9331 communiquent entre eux grâce à la **liaison série asynchrone (UART)**. Cette communication est facilitée par la librairie « bridge ».



3 – CARACTERISTIQUES TECHNIQUES

3.1 – Caractéristiques principales

Microcontrôleur ATmega32U4		Microprocesseur Atheros AR9331	
Tension de fonctionnement	5 V	Tension de fonctionnement	3,3 V
E/S numériques	14	Architecture	MIPS
Sorties PWM	7	Ethernet	802.3 10/100Mbit/s
Entrées analogiques	6	WiFi	802.11b/g/n 2.4 GHz
Intensité par broche E/S	40 mA max	USB Type	2.0 Host
Intensité broche 5 V	500 mA max	Lecteur de carte	Micro-SD
Intensité broche 3,3 V	50 mA max	RAM	64 MB DDR2
Mémoire Flash	32 Ko	Mémoire Flash	16 MB
SRAM	2,5 Ko	SRAM	2,5 Ko
EEPROM	1 Ko	EEPROM	1 Ko
Vitesse d'horloge	16 MHz	Vitesse d'horloge	400 MHz

3.2 – Alimentation

Il est recommandé d'alimenter la carte par l'intermédiaire de la connexion **micro-USB** avec une **tension de 5V continue**.

Il est possible d'alimenter la carte par l'intermédiaire de la broche Vin. Cependant la carte ne possédant pas de régulateur de tension, il est nécessaire de fournir une tension régulée à 5 V continue. La carte sera donc endommagée en cas de surtension.

La carte Yún est également compatible avec les **modules PoE** (Power Over Ethernet) il est nécessaire d'ajouter un module PoE pour bénéficier de cette fonctionnalité.



3.3 – Mémoires

L'ATmega32U4 dispose de **32 Ko** de mémoire de programme flash dont **4 Ko** utilisé par le bootloader. Il dispose également de **2,5 Ko de SRAM** et de **1 Ko d'EEPROM**.

La mémoire du AR9331 n'est pas incluse dans le processeur. La RAM et la mémoire de stockage sont disposées à l'extérieur du processeur. La RAM est de type DDR2 et sa capacité de **64 Mo**. La mémoire de stockage flash a une capacité de 16 Mo.

La mémoire flash est préchargée en usine avec une **distribution Linux** basée sur OpenWRT. Cette distribution s'appelle Linino.

Il est possible de modifier le contenu de la mémoire flash en installant un programme ou en modifiant le fichier de configuration. Pour revenir à la configuration d'usine il faut presser le bouton «WLAN RST» pendant 30 secondes.

3.4 – Les entrées/sorties

La carte YUN possède **20 broches d'entrée/sortie numériques**. Il s'agit des broches numériques **0 à 13** et des broches **A0 à A5** qui peuvent également être utilisées comme entrée/sortie numériques. Chaque broche peut fournir et recevoir un maximum de 40 mA et dispose d'une résistance de pull-up (déconnectée par défaut).

Ces broches d'entrée/sortie numériques possèdent des fonctions spécialisées :

Broche	E/S numérique	Liaison	Interruption	PWM	Analogique
0	E/S numérique	Rx (UART)	Interruption 2		
1	E/S numérique	Tx (UART)	Interruption 3		
2	E/S numérique	SDA (I2C)	Interruption 1		
3	E/S numérique	SCL (I2C)	Interruption 0	PWM	
4	E/S numérique				A6
5	E/S numérique			PWM	
6	E/S numérique			PWM	A7
7	E/S numérique		Interruption 4		
8	E/S numérique				A8
9	E/S numérique			PWM	A9
10	E/S numérique			PWM	A10
11	E/S numérique			PWM	
12	E/S numérique				A11
13	E/S numérique			PWM	
A0	E/S numérique				A0
A1	E/S numérique				A1
A2	E/S numérique				A2
A3	E/S numérique				A3
A4	E/S numérique				A4
A5	E/S numérique				A5

La **liaison série asynchrone** (UART) est disponible sur les broches numériques **0 (Rx)** et **1 (Tx)**.

Le **Bus I2C** (ou TWI : Two Wire) est disponible sur les broches numériques **2 (SDA)** et **3 (SCL)**.

Les **sorties PWM** sont disponibles sur les broches numériques **3, 5, 6, 9, 10, 11** et **13**.

Les **interruptions externes** peuvent être déclenchées par des événements sur les broches **3** (interruption 0), **2** (interruption 1), **0** (interruption 2), **1** (interruption 3) et **7** (interruption 4). Il n'est pas recommandé d'utiliser les broches 0 et 1 comme interruption parce qu'elles sont aussi utilisées par le port série. La broche 7 est connectée sur le processeur AR9331, il est donc recommandé d'être prudent vis-à-vis des possibles conflits si elle est utilisée comme interruption.

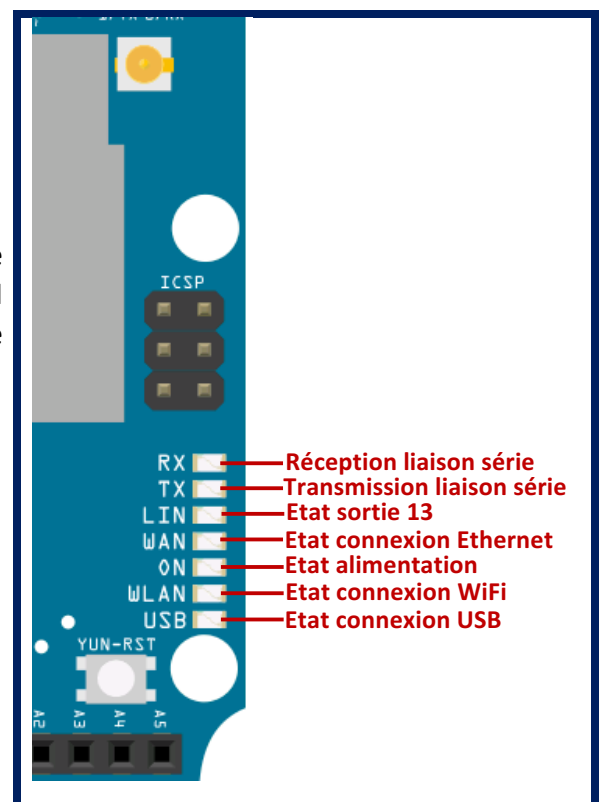
Le **bus SPI** n'est disponible que sur le **connecteur ICSP**.

Les **entrées analogiques** sont disponibles sur les broches **A0 à A5** et sur les broches numériques **4 (A6)**, **6 (A7)**, **8 (A8)**, **9 (A9)**, **10 (A10)** et **12 (A11)**. Le CAN d'une résolution numérique de 10 bits. Par défaut la tension de référence est de 5 V. Il est cependant possible de changer la valeur maximale utilisée en utilisant la broche AREF.

3.5 – Les DELs

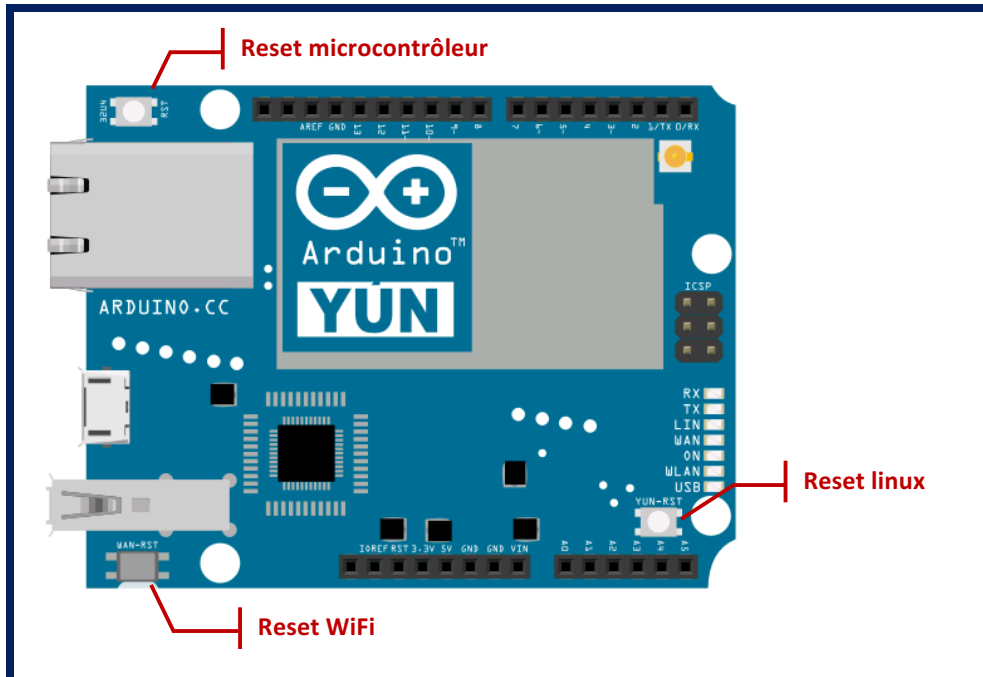
La carte YUN dispose d'un certain nombre de DEL

- La **DEL L13** ou **LIN** est connectée sur la broche numérique 13. Lorsque la broche est à l'état HIGH (état haut), la DEL est allumée. La DEL est éteinte lorsque la broche est à LOW (état bas).
- La **DEL ON** indique l'état de l'alimentation.
- La **DEL WAN** indique l'état de la connexion Ethernet.
- La **DEL WLAN** indique l'état de la connexion WiFi.
- La **DEL USB** indique l'état de la communication USB.
- Les **DEL Rx et Tx** indiquent l'état de la liaison série.



3.6 – Boutons de reset

Il y a 3 boutons de reset sur la carte Yùn :



Le bouton **Yùn RST** permet le **reboot (redémarrage) du système linux**. L'appui sur ce bouton permet de faire un reset du microprocesseur AR9331. Toutes les données stockées en RAM seront perdues et tous les programmes en cours d'exécution seront arrêtés.

Le bouton **32U4 RST** permet de faire un reset du microcontrôleur ATmega32U4.

Le bouton **WLAN RST** a une double fonctionnalité :

- Il permet de **restaurer la configuration d'usine Wifi**. Cette restauration provoque le reboot de l'environnement linux. Pour restaurer la configuration Wifi il faut presser et maintenir le bouton WLAN RST pendant 5 secondes.
- Il permet également de **restaurer l'environnement linux** par défaut. Pour cela il faut presser le bouton pendant 30 secondes. Cette restauration efface tous les données sauvegardées et tous les logiciels installés dans la mémoire Flash du AR9331.

3.7 – Configuration WiFi usine

La carte Yùn est configurée par défaut en mode "**access point**" (AP) et avec l'adresse IP **192.168.240.1**. Il est possible de connecter avec un ordinateur sur le réseau WiFi géré par la carte Yùn et de le nom SSID et "**Arduino Yun-XXXXXXXXXXXX**" (où les 12 'X' sont l'adresse MAC de la carte Yùn).

Une fois connecté, il est possible d'accéder au panneau de configuration web de la carte Yùn, via un navigateur, à partir l'adresse **192.168.240.1** ou de l'URL "**http://arduino.local**".



4 – BRIDGE, UN PONT ENTRE LINUX ET ARDUINO

4.1 – Communication bidirectionnelle

L'environnement Arduino et l'environnement Linux peuvent communiquer à l'aide de la bibliothèque « **Bridge** », présente à la fois du côté Arduino et du côté Linux (via une bibliothèque Python).

Les deux environnements communiquent entre eux selon les couches suivantes :

- 1 – **Liaison série** entre le microcontrôleur Arduino et le microprocesseur Linux.
- 2 – **Bibliothèque « Bridge »**, qui est la partie bas niveau du protocole de communication.
- 3 – **Bibliothèques spécifiques** s'appuyant sur « Bridge » :
 - Console : émulation du terminal série
 - Process : commandes sur l'environnement Linux à partir l'environnement Arduino, avec récupération de la sortie de ces commandes.
 - FileIO : lecture et écriture de fichiers sur la carte micro SD à partir d'Arduino.
 - YunClient : permet de réaliser un client HTTP simple.
 - YunServer : permet de réaliser un serveur HTTP simple.
 - Temboo : 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 environnements permet (entre autre) de déléguer une partie de la couche logicielle réseau à l'environnement Linux, et donc 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 Linux, et c'est ce dernier qui effectuera le formatage final avant le 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.) afin d'avoir des pages WEB dynamiques.

4.2 – Appels REST

Tout accès à l'adresse « **http://<ip>/arduino/chaine** » transfère automatiquement la chaîne de caractères « chaine » à l'Arduino. Le programme tournant sur le microcontrôleur utilise la bibliothèque « YunServer » pour traiter la chaîne de caractères, et « YunClient » pour d'éventuels retours vers le navigateur.

Il est possible ainsi de créer un **API REST (Application Programming Interface Representational State Transfer)** permettant de communiquer avec l'Arduino à travers le réseau.

Sur la carte Yún, il y a deux point d'accès REST 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 programme fonctionnant sur le microcontrôleur.

Le répertoire « /data » est utilisé pour fournir un accès à un stockage interne de type « Clé/Valeur ». Les appels possible sont :

- /put/KEY/VALUE : stocke une valeur « VALUE » dans la clé « KEY ».
- /get/KEY : récupère la valeur de la clé « KEY ». La valeur est retournée au format JSON.
- /get : récupère la liste des éléments stockés au format JSON.
- /delete : efface le contenu du stockage interne.

4.3 – Exemple API REST

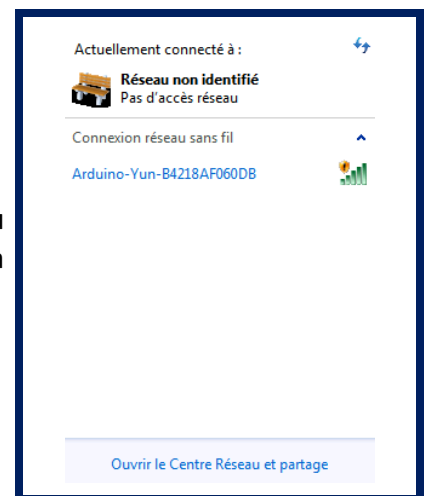
Le programme d'exemple « Bridge » de l'IDE Arduino permet d'interroger les entrées/sorties de l'Arduino ou bien de modifier leur état à l'aide d'une API REST du type : « <commande>/<broche>/<valeur> ». Par exemple :

- /arduino/digital/13 : lit l'état de l'entrée numérique 13.
- /arduino/digital/13/1 : place la sortie numérique à l'état 1.
- /arduino/analog/2 : lit la valeur de l'entrée analogique 2.
- /arduino/mode/13/input : configure la broche numérique 13 comme une entrée.
- /arduino/mode/13/output : configure la broche numérique 13 comme une sortie.

Alimenter la carte Yùn.

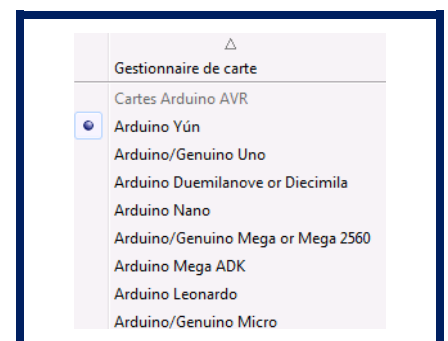
Lors de son démarrage, la carte va créer automatiquement un réseau WiFi intitulé « **Arduino Yun-XXXXXXXXXXXX** » (les 12 X correspondants à l'adresse MAC de la carte Yùn)

Connecter l'ordinateur à ce réseau WiFi.



Lancer le programme Arduino. Ouvrir le programme « **Bridge** » qui se trouve dans les exemples de l'IDE.

Sélectionner dans la liste de cartes la carte Yùn.



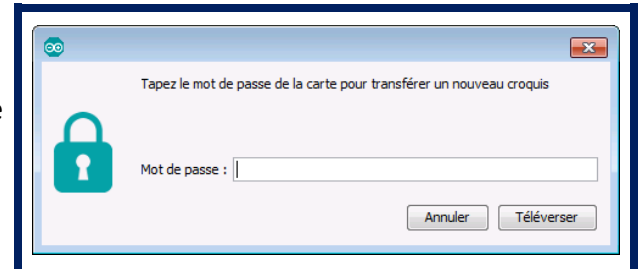
Choisir dans la liste des ports disponibles le port réseau « **serveur at 192.168.240.1 (Arduino Yùn)** ».



Téléverser le programme dans la carte Yùn.

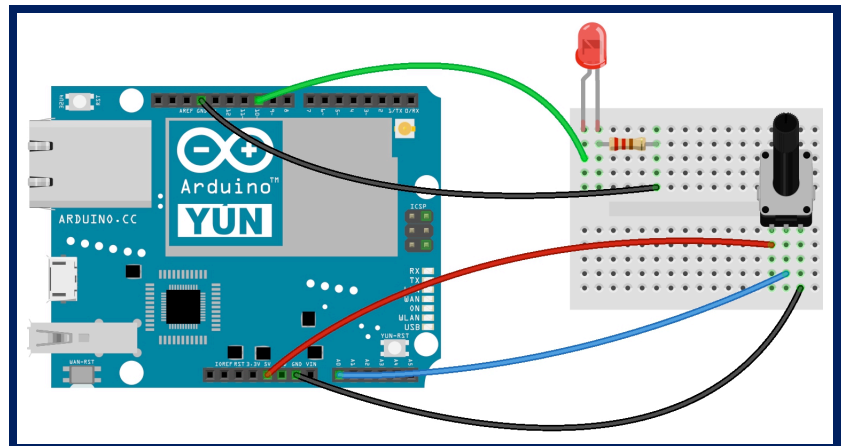
Lors du téléversement, une fenêtre demande un mot de passe.

Entrer le mot de passe « **arduino** ».



Le montage suivant permet la commande d'une DEL à partir de la broche 10 et de récupérer une tension analogique réglable sur l'entrée analogique A0.

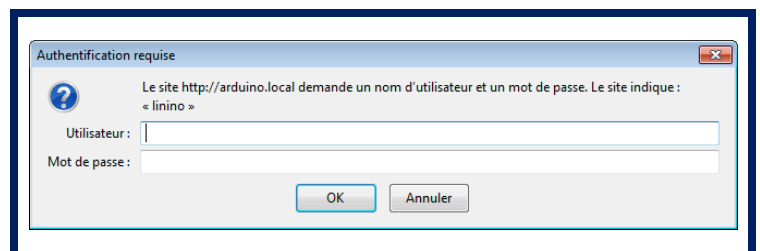
Câbler le montage ci-contre.



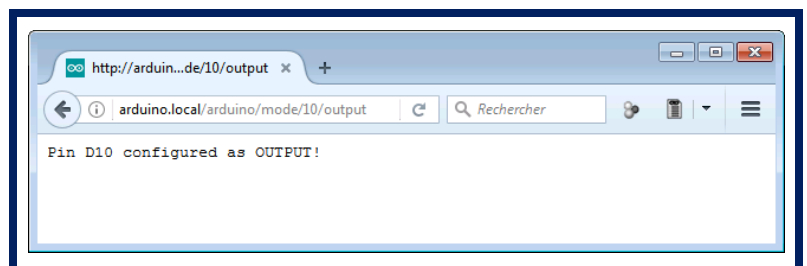
Lancer un navigateur WEB. Entrer, dans la barre d'adresse, l'URL suivante : « **arduino.local/arduino/mode/10/output** » ou « **192.168.240.1/arduino/mode/10/output** ».

Lors du premier accès une demande d'authentification est nécessaire.

Entrer « **root** » comme Utilisateur et « **arduino** » comme Mot de passe.



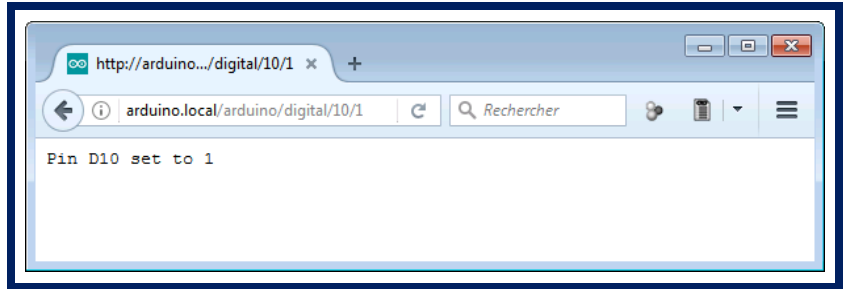
La broche 10 est configurée en sortie.





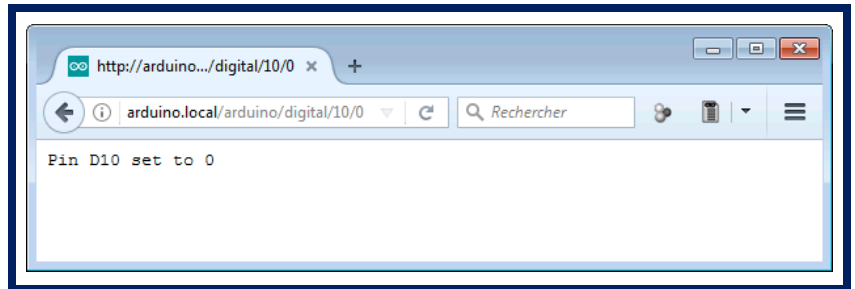
Entrer, dans la barre d'adresse, l'URL suivante : « **arduino.local/arduino/digital/10/1** » ou « **192.168.240.1/arduino/digital/10/1** ».

La broche 10 est placée à l'état 1 et la DEL s'allume.



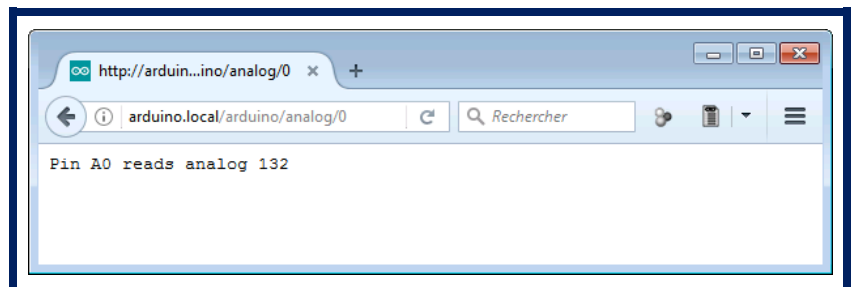
Entrer, dans la barre d'adresse, l'URL suivante : « **arduino.local/arduino/digital/10/0** » ou « **192.168.240.1/arduino/digital/10/0** ».

La broche 10 est placée à l'état 0 et la DEL s'allume.



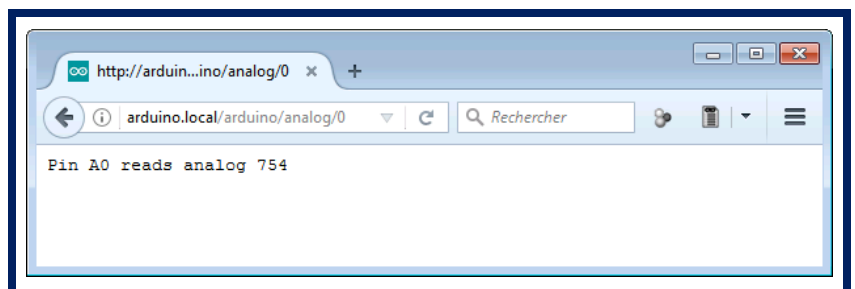
Entrer, dans la barre d'adresse, l'URL suivante : « **arduino.local/arduino/analog/0** » ou « **192.168.240.1/arduino/analog/0** ».

La valeur lue sur l'entrée A0 est de 132.



Modifier la position du potentiomètre et recharger l'URL précédente.

La nouvelle valeur lue sur l'entrée est de 754.



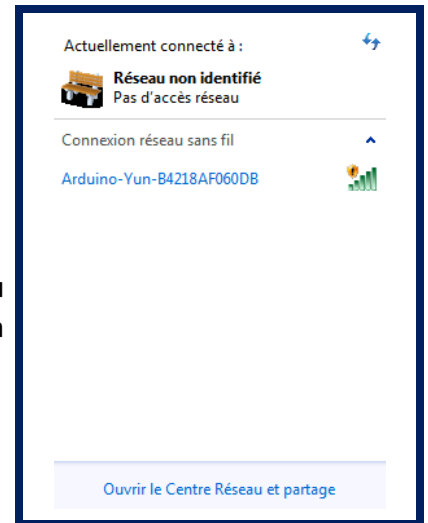


5 – CONFIGURATION WIFI

La carte Yún peut être utilisée en point d'accès WiFi, mais elle peut également se connecter à un réseau WiFi existant. Elle peut se connecter à des réseaux non protégés (non cryptés) ainsi que sur des réseaux supportant le chiffrement WEP, WPA et WPA2.

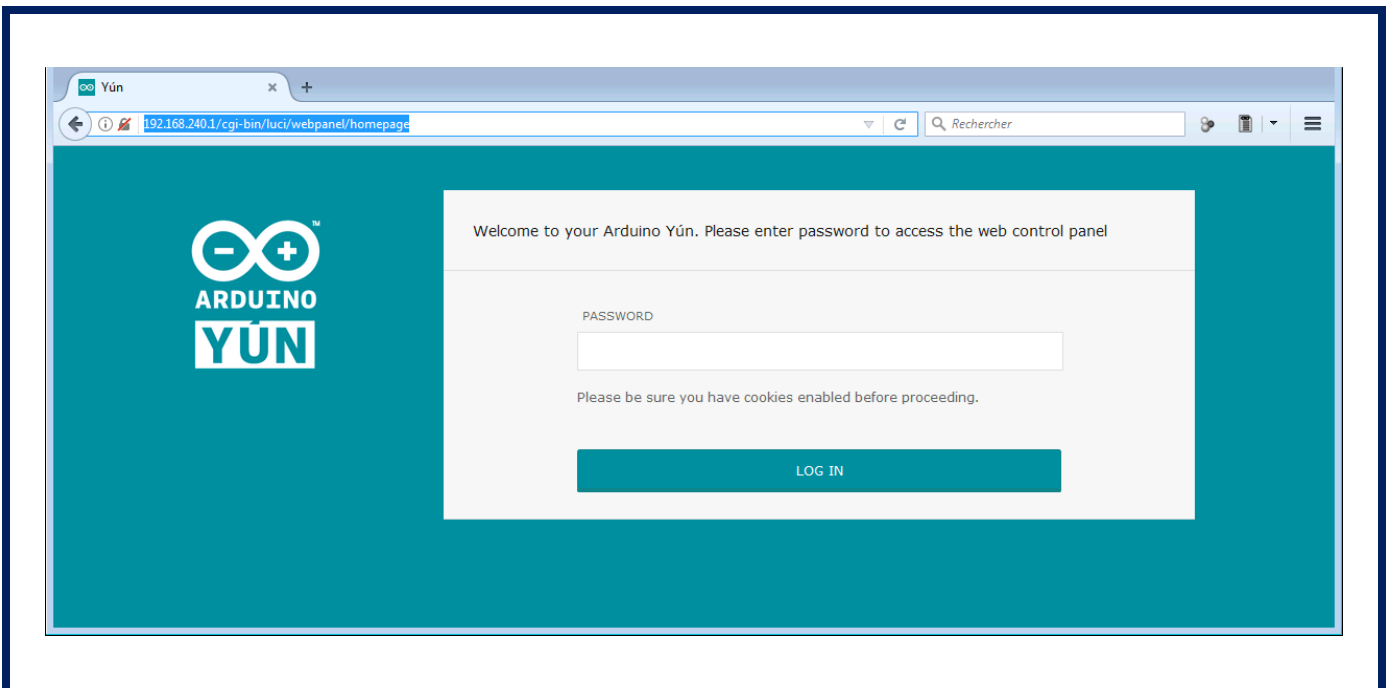
Lors de son démarrage, la carte va créer automatiquement un réseau WiFi intitulé « **Arduino Yun-XXXXXXXXXXXX** » (les 12 X correspondant à l'adresse MAC de la carte Yún)

Connecter l'ordinateur à ce réseau WiFi.



Lancer un navigateur Internet et entrer l'URL « **http://arduino.local** » ou l'adresse IP **192.168.240.1** dans la barre d'adresse.

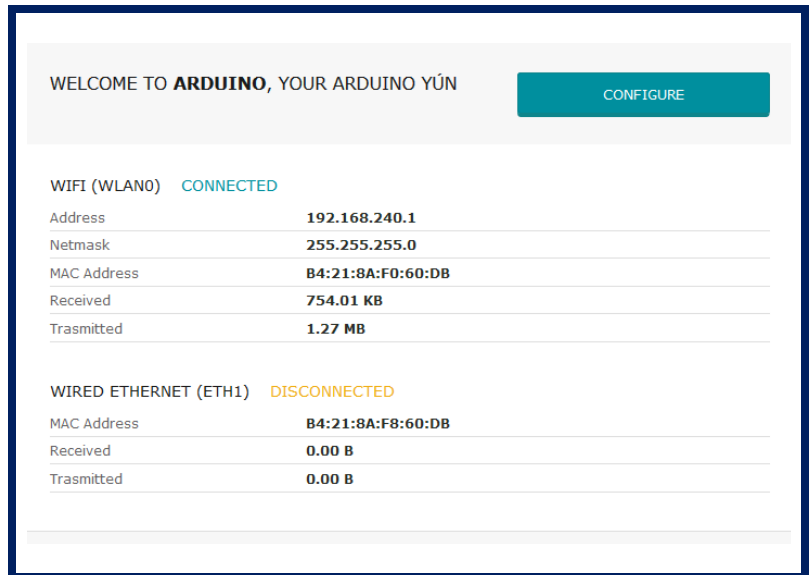
La page WEB obtenue demande un mot de passe d'accès (password). Entrer "**arduino**" et cliquer le bouton « **Login** ».





L'interface WEB configuration est alors accessible. La page obetnue contient les informations relative aux connexions réseaux actuelles.

Cliquer sur « **CONFIGURE** » pour poursuivre la configuration de la carte Yùn.

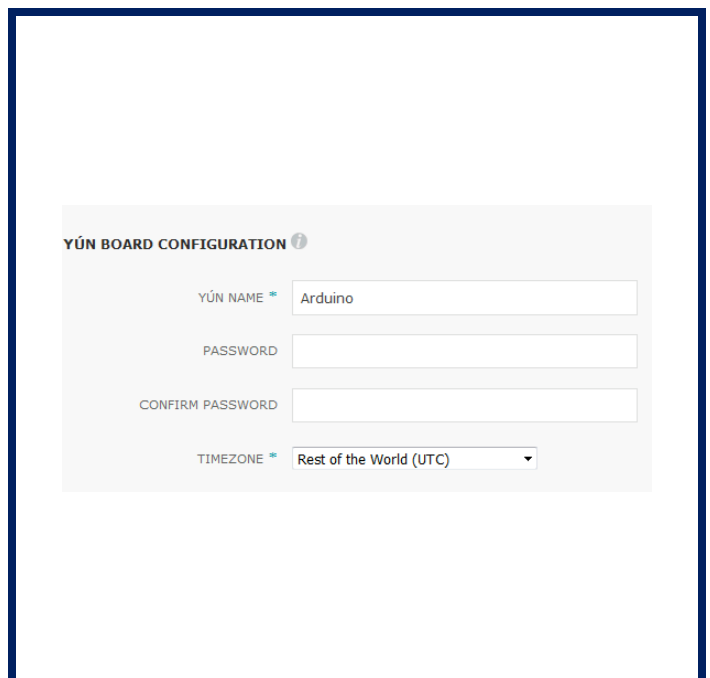


La première zone de configuration permet de paramètrer la carte Yùn.

Placer le **nom de la carte Yùn** (par défaut Arduino) dans le champ « **Board name** ». Ce nom sera utilisé par la suite pour se connecter à la carte.

Choisir et confirmer un **mot de passe** pour la carte Yùn. Il doit posséder au moins 8 caractères. Si les champs « **Password** » restent vides, le système maintiendra le mot de passe par défaut c'est-à-dire « arduino »

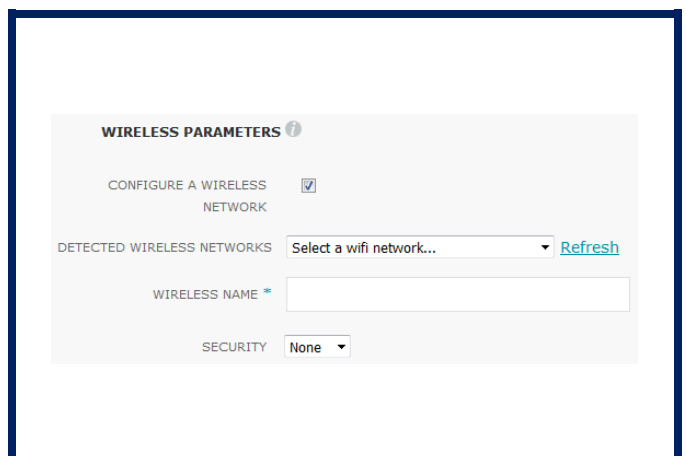
Le champ « **Timezone** » (fuseau horaire) permet de **configurer le fuseau horaire et le pays**. Il est recommandé de configurer ces options puisqu'elles peuvent faciliter la connexion sur le réseau Wifi local.



La zone de configuration suivante permet de configurer les paramètres WiFi.

Entrer dans le champ « **WIRELESS NAME** », le **nom du réseau WiFi** sur lequel doit se connecter la carte Yùn ou bien sélectionner le réseau WiFi à partir de la liste proposée par le champ « **DETECTED WIRELESS NAME** »,

Sélectionner à partir d champ « Security » le **type de cryptage** utilisé sur le réseau WiFi puis la **clé d'identification** selon option choisie.





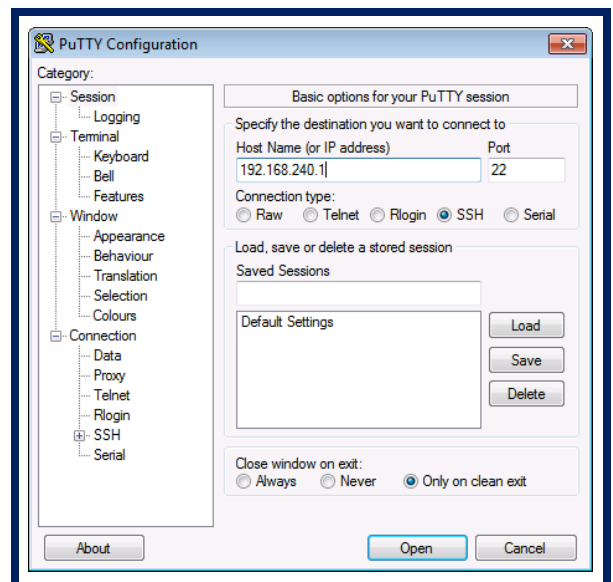
6 – MISE EN PLACE D'UN SERVEUR WEB A PARTIR DE LA CARTE SD

La mémoire Flash interne de la carte Yùn possède un nombre limité d'écritures possibles. Il est préférable d'utiliser la carte micro-SD comme mémoire externe pour sauvegarder les données, les pages WEB, etc...

Placer la carte SD dans le slot prévu à cet effet.

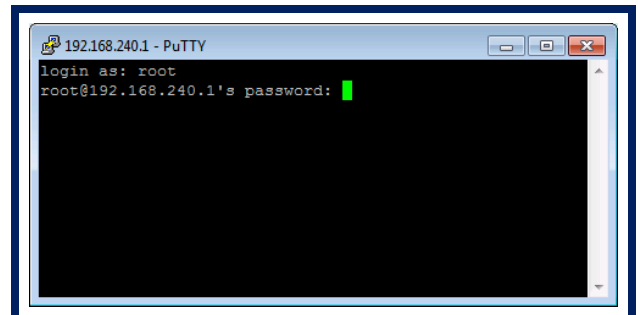
Lancer le programme **PuTTY** (ou WinSCP) afin de se connecter à l'environnement Linux de la carte Yùn à partir de l'ordinateur.

Dans le champ « **Host Name (or IP adress)** », entrer l'adresse IP de la carte Yùn.



Une fenêtre « terminal » s'ouvre alors permettant la connexion avec la carte Yùn.

Entrer le login « **root** » et le mot de passe « **arduino** » (s'il n'a pas été changé).



Une invite de commande s'affiche alors. Il est alors possible de manipuler la carte Yùn en lignes de commandes Linux.





Taper la commande suivante pour déterminer le point de montage de la carte SD.

```
mount
```

Dans notre exemple la carte SD est accessible à partir de « **/mnt/sda1** ».

```
192.168.240.1 - PuTTY
root@Arduino:~# mount
rootfs on / type rootfs (rw)
/dev/root on /rom type squashfs (ro,relatime)
proc on /proc type proc (rw,noatime)
sysfs on /sys type sysfs (rw,noatime)
tmpfs on /tmp type tmpfs (rw,nosuid,nodev,noatime,size=30560k)
tmpfs on /dev type tmpfs (rw,noatime,size=512k,mode=755)
devpts on /dev/pts type devpts (rw,noatime,mode=600)
/dev/mtdblock3 on /overlay type jffs2 (rw,noatime)
overlayfs:/overlay on / type overlayfs (rw,relatime,lowerdir=/,upperdir=/over
rlay)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
/dev/sda1 on /mnt/sda1 type vfat (rw,sync,relatime,fmask=0022,dmask=0022,cod
epage=cp437,iocharset=iso8859-1,shortname=mixed,errors=remount-ro)
none on /proc/bus/usb type usbfs (rw,relatime)
root@Arduino:~#
```

Créer un répertoire « **arduino** » sur la carte sd qui pourra être monter automatiquement sous « **/mnt/sd** » et utilisable dans les programmes arduino .ino.

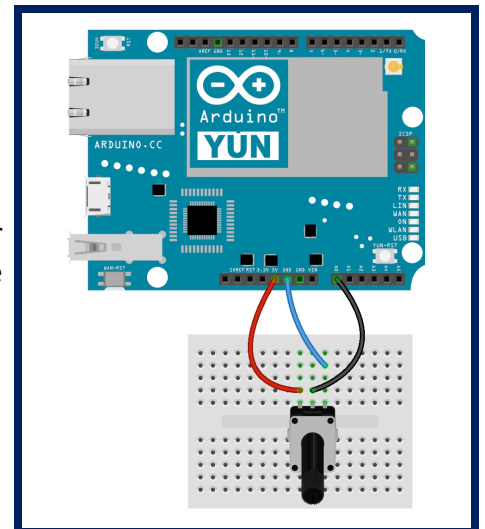
```
cd /
mkdir /mnt/sda1/arduino
chmod 777 /mnt/sda1/arduino
```

Il faut créer, dans le répertoire « **arduino** », le dossier « **www** » qui contiendra les pages WEB.

```
mkdir /mnt/sda1/arduino/www
```

6 – EXEMPLE

L'exemple suivant permet d'afficher sur une page WEB, la valeur d'une tension analogique variable appliquée sur l'entrée analogique A0 de la carte Yùn.



Le fonctionnement de cet exemple nécessite 3 fichiers :

- le programme arduino : « **mes_val.ion** » ;
- le fichier html de la page WEB : « **index.html** » ;
- une bibliothèque javascript permettant le transfert des valeurs de la carte Yùn vers le serveur WEB : « **zepto.min.js** ».



Le programme arduino est le suivant :

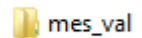
```
#include <Bridge.h>
#include <YunServer.h>
#include <YunClient.h>

YunServer server;

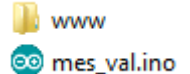
void setup() {
  Bridge.begin();
  server.listenOnLocalhost();
  server.begin();
}

void loop() {
  YunClient client = server.accept();
  if (client) {
    String command = client.readString();
    command.trim();
    if (command == "valeur") {
      int val = analogRead(A0);
      client.print(val);
    }
    client.stop();
  }
  delay(50);
}
```

Editer ce programme et enregistrer sous le nom « mes_val.ino ». Ce programme a été enregistré dans un dossier portant le même nom



Créer dans ce dossier un sous-dossier nommé « www ». Tout le contenu de ce sous-dossier sera chargé sur la carte SD lors du téléversement du programme arduino via le port WiFi.



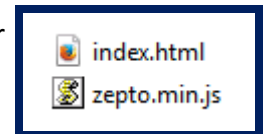
Le fichier html est le suivant :

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
    <script type="text/javascript" src="zepto.min.js"></script>
    <script type="text/javascript">
      function refresh() {
        $('#content').load('/arduino/valeur'); } </script>
  </head>
  <body onload="setInterval(refresh, 1000);">
    <h1>Valeur lue sur l'entrée A0 :</h1>
    <span id="content">Attente de la carte Arduino</span>
  </body>
</html>
```

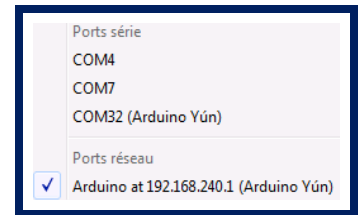


Editer ce fichier et enregistrer-le sous le nom « **index.html** » dans le sous-dossier « **www** » du dossier « **mes-val** ».

Sauvegarder également dans ce répertoire le fichier « **zepto.min.js** ».



Choisir, sur l'IDE Arduino, le port WiFi et téléverser l'ensemble des fichiers. Taper le passe de la carte Yún pour finaliser le téléversement.



Le programme arduino a été chargé dans le microcontrôleur ATmega32U4 et les fichiers « **index.html** » et « **zepto.min.js** » contenus dans le dossier « **mes_val/www** » ont été chargés dans le dossier « **/arduino/www** » de la carte SD.

Pour avoir accès à la page WEB, taper dans la barre d'adresse du navigateur WEB, l'URL « **http://<ip>/sd/mes_val/** » ou « **http://arduino.local/sd/mes_val/** ».

