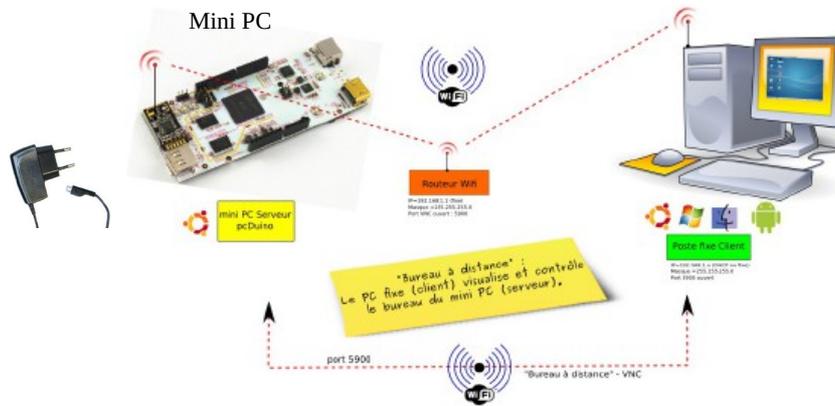


NOM :

## Electronique Embarquée : Prise en main du mini PC Pcdduino V3

Ce TP est inspiré des différents tps proposés sur le site de Xavier Hinault :

[http://www.mon-club-elec.fr/pmwiki\\_mon\\_club\\_elec/pmwiki.php?n=MAIN.ATELIERSPYDUINO](http://www.mon-club-elec.fr/pmwiki_mon_club_elec/pmwiki.php?n=MAIN.ATELIERSPYDUINO)



Adresse MAC du Pcdduino :  
08:D8:33:21:B4:FF

Adresse « réservée » au Pcdduino :  
192.168.0.125

### Activité 1 : Prise à distance du mini PC par WIFI

A partir du document « Contrôle à distance du PCduinoV3.pdf », activer le wifi du mini PC , installer tightvnc pour windows sur un poste fixe et prendre le contrôle à distance du mini PC ( en étant sur le même réseau)

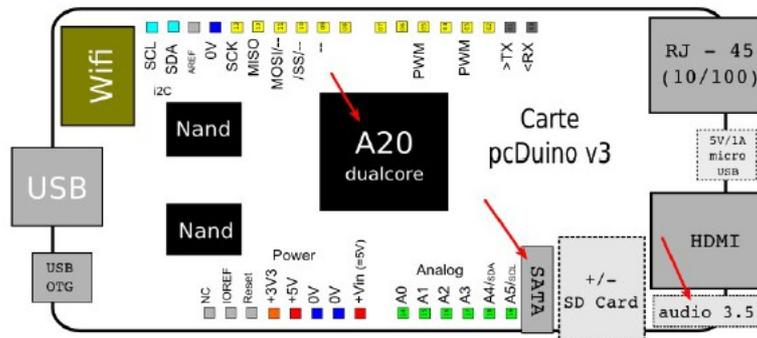
	Validation du professeur :
Activation du Wifi du mini PC + ifconfig	
Installation tightvnc + commande arp -a sur pc distant avec windows	
Contrôle à distance du mini PC	

### Activité 2 : Initiation à Pyduino

Pyduino est un portage en langage **Python** des instructions du langage **Arduino** pour les nouvelles plateformes de type mini-pc . Vous accédez aux broches E/S numériques et analogiques de votre mini-pc comme vous le feriez avec Arduino, sauf que vous codez en Python et donc vous pouvez au sein d'un même code programmer les broches E/S, faire des mesures, etc... et interagir avec le système, utiliser les périphériques système (vidéo, réseau, fichiers,..), etc... De plus, Pyduino est indépendant du matériel utilisé : un même code Pyduino pourra être exécuté sur des plateformes différentes (pcDuino, RaspberryPi,...)

NOM :

## Présentation de la carte Pcdduino V3



### Pcdduino V3 est un mini PC intégrant :

- un processeur dual core ARM ( Allwinner A20) 1GHz
- de la mémoire flash NAND intégrée (4Go) (  $\Rightarrow$  accès plus rapide ) ( extensible avec une micro carte SD jusqu'à 32Go)
- un module WIFI intégré
- système Lubuntu 12 .04 pré-installé
- connectique : USB x1, Ethernet RJ 45 10/100 Mbps , USB OTGx1, 1x audio jack 3,5mm, 1 x connecteur SATA pour disque dur externe , HDMI 1.4
- alimentation par un adaptateur +5V DC - 2A

une connectique « Arduino-like » sur borniers droits femelles associant :

- 14 broches E/S numériques
  - 6 broches analogiques ( dont 4 en résolution numérique sur 12 bits)
  - 2 sorties PWM
  - communication 1xUART, 1xI2C, 1xSPI
- Attention les broches d'entrées - sorties doivent être alimentées en +3 ,3V**

**Réaliser le câblage des entrées-sorties HORS TENSION !**

- A partir du document « Initiation au langage Python », quelles sont les **fonctions obligatoires** dans tout programme Pyduino ?

- Quelle est la touche qui permet l'**indentation** en Python ( « Tout bloc d'instructions doit être indenté ») ?

A partir du document « Contrôle à distance du Pcdduino V3 » (pages 19 à 25), réaliser le programme permettant de **faire clignoter une led sur la sortie numérique O2 de la carte Pcdduino** (à câbler sur une platine d'expérimentation « Labdec » ( "breadboard" )). Exécuter ensuite ce programme de façon **autonome** et le stopper après votre test.

	Validation professeur :
Câblage et programmation de la carte pcdduino	
Rendre autonome l'application	
Stopper l'application	

Modifier ce programme pour faire **clignoter alternativement 2 leds** ( utiliser les sorties numériques **O2 et O4** de la carte Pcdduino)

	Validation professeur :
Câblage et modification du programme	

NOM :

### Activité 3 : Broches numériques en entrée : utiliser un bouton poussoir et des dispositifs mécaniques ON/OFF monovoie.

- A partir du document « pyduino\_ES\_entree\_BP.pdf », câbler un bouton poussoir à l'entrée numérique O2 de la carte Pduino et réaliser le programme permettant de lire l'état logique du bouton poussoir et d'afficher sur le terminal « appui BP » lorsque celui-ci est appuyé ( utiliser « une pause anti-rebond » )

	<b>Validation professeur :</b>
Câblage et programme lecture BP sans timer	

- Quelle est l'instruction qui permet de paramétrer l'entrée numérique avec une résistance de rappel interne reliée au +Vcc ?

- Quelle est l'instruction qui permet de tester le bouton poussoir ?

Exécuter le programme.

Ouvrir un terminal et taper la commande **top**

- Quel est le taux d'occupation du processeur pour les applications utilisateurs (%us) ? pour les applications système (%sy) ?

- Quel est le taux d'occupation CPU et mémoire utilisé par votre programme (%CPU et %MEM) ?

- Quelle est la fonction du programme qui monopolise le processeur ?

Modifier le programme pour optimiser le temps CPU en utilisant l'instruction **timer** (délai, fonction).

	<b>Validation professeur :</b>
Câblage et programme lecture BP avec l'instruction timer	

Exécuter le programme.

Ouvrir un terminal et taper la commande **top**

- Quel est le taux d'occupation CPU et mémoire utilisé par votre programme (%CPU et %MEM) ?

- Conclure sur le rôle de l'instruction timer .

Réaliser le câblage d'un bouton poussoir (entrée numérique O2) et d'une Led (sortie numérique O4) pour concevoir une minuterie (temps d'allumage = 10 secondes)

	<b>Validation professeur :</b>
Câblage BP et Led + programme minuterie 10s	

### Activité 4 : Réalisation d'un serveur TCP / Http renvoyant la mesure d'une broche analogique (entrée analogique A2 de la carte Pduino sur laquelle vous câblerez une résistance variable (tension variant de 0V à 3,3V) )

Quelle est l'adresse IP du Pduino?

Commande utilisée :

NOM :

A partir du document « ReseauServeurTCPAnalogReadx1.pdf », câbler et réaliser le programme.

	<b>Validation professeur :</b>
Câblage et réalisation du programme	

Faire varier la résistance ajustable pour avoir les valeurs numériques min et max .

	Valeur numérique obtenue	Code binaire sur 12 bits	Tension obtenue en mV
Résistance variable au minimum			
Résistance variable au maximum			

- Quelle est l'instruction permettant de lire et de convertir en binaire la grandeur analogique ?

- Quelle est l'instruction permettant de lire et calculer en mV la grandeur analogique ?

- Quel est le rôle de la fonction str() ?

Nota : \n est un saut de ligne

**Modifier le programme pour lire 2 grandeurs analogiques** (utiliser l'entrée analogique A3)

	<b>Validation professeur :</b>
Modification du programme	

**Activité 5 : Contrôler la luminosité d'une LED (en PWM) à l'aide d'une interface graphique intégrant un potentiomètre rectiligne (slider) et un afficheur numérique.**

A partir du document « Sortie PWM et seekbar.pdf » , câblez et réalisez le programme.

	<b>Validation professeur :</b>
Câblage et test du programme	

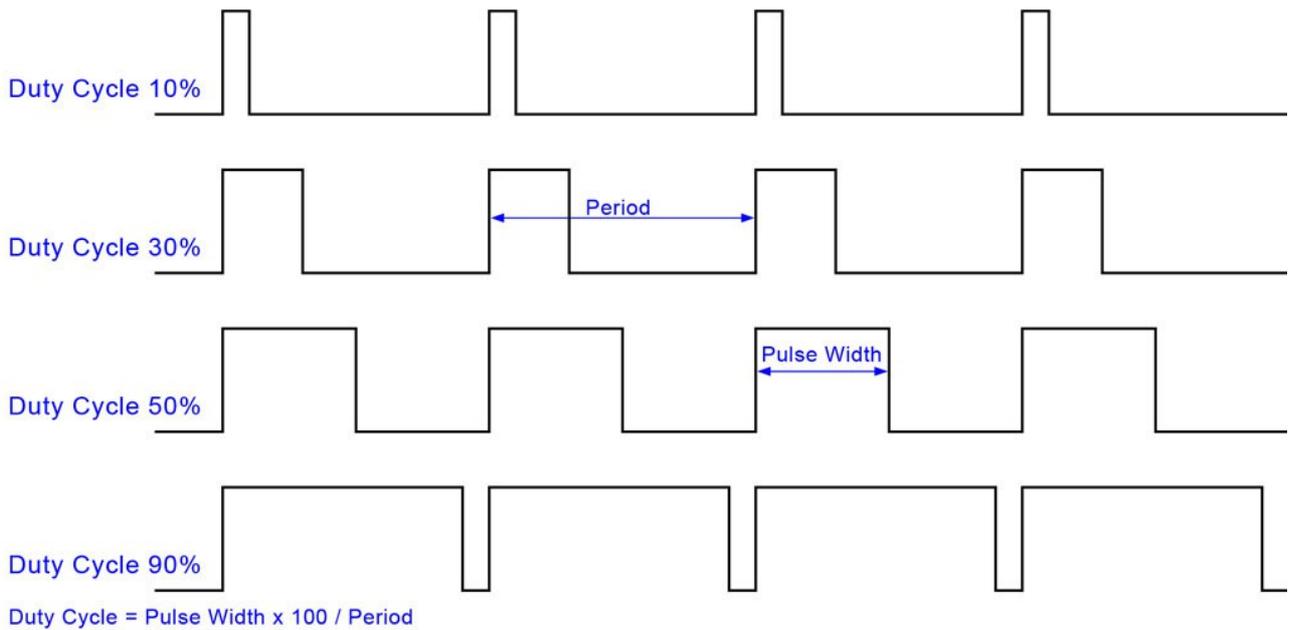
- Quelle est l'instruction qui permet de contrôler la luminosité par variation du rapport cyclique ( modulation PWM) ?

NOM :

**La modulation PWM consiste à faire varier le rapport cyclique d'un signal rectangulaire unidirectionnel sans modifier la période de celui-ci (donc la fréquence du signal n'est pas modifiée)**

**Chronogrammes expliquant la modulation PWM :**

Duty Cycle en % = rapport cyclique en % = (durée à l'état haut / période) \* 100



- Si l'amplitude des signaux est de 3,3V , calculer la **valeur moyenne de la tension** obtenue pour chaque signal.

$$\text{Valeur moyenne (en V)} = \frac{V_{\text{max}} * \text{durée à l'état haut}}{\text{période}}$$

Cas n°1 :

Cas n°2 :

Cas n°3 :

Cas n°4 :

- Expliquer pourquoi la luminosité de la Led varie ( penser à l'incidence de la valeur moyenne de la tension sur l'intensité de courant traversant la led ).