Rapport écologique

GreenIT – Amélioration de la consommation énergétique

EPSI

PFARR - POTIRON

Table des matières

[1] Introduction 2](#_Toc434820261)

[2] Récupération de la puissance consommée en temps réel 3](#_Toc434820262)

# 1] Introduction

Dans le cadre de l’amélioration de la consommation énergétique de notre entreprise ainsi que de l’impact écologique dans le monde, il est nécessaire d’analyser les consommations électriques de nos différents outils.

Ce projet aura pour but de donner un aperçu des consommations énergétiques en temps réel de notre cluster de serveurs Web, hébergés sur Raspberry PI. Le second objectif est de réduire leurs consommations énergétiques par différents moyens présentés dans ce dossier.

Nous travaillons en groupe de deux personnes du service informatique.

* POTIRON Baptiste
* PFARR Vincent

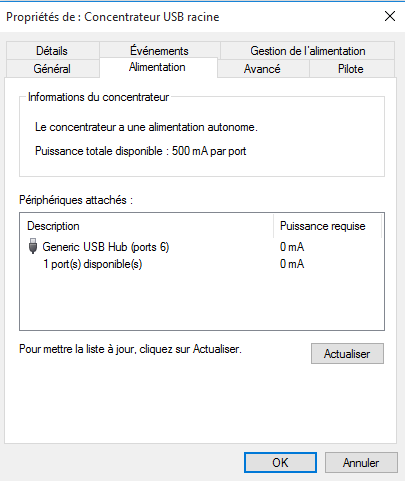
Ce rapport sera une preuve de l’amélioration de notre impact environnemental au sein du service informatique de notre entreprise.

# 2] Récupération de la puissance consommée en temps réel

A l’aide d’un Raspberry PI b+ hébergeant un serveur web, que nous avons détaché de notre infrastructure, nous allons effectuer des tests pour récupérer la puissance consommée par notre Raspberry en temps réel. Nous décrirons 3 méthodes distincts pour relever cette consommation électrique P.

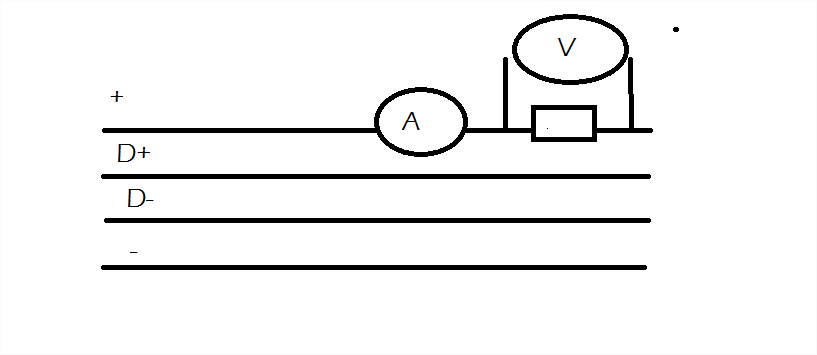
## Méthode 1 – Gestion de l’alimentation Windows

A l’aide de Windows, il est possible de récupérer le courant de sortie d’un port USB. Cette mesure se situe dans la gestion de périphérique. Cette mesure est très peu précise car arrondie. De plus, nous ne pourrons pas récupérer la tension de sortie. Nous serons donc rapidement limités.



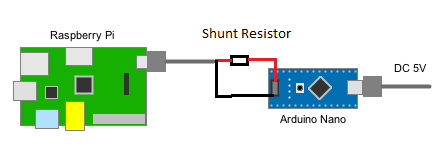
## Mesure 2 – Résistance de Shunt avec rallonge modifiée

Pour cette mesure, nous aurons besoin d’une rallonge USB. Celle-ci sera dénudée et sera couplée à un ampèremètre en série. La rallonge ira du Raspberry à une prise de courant pour l’alimenter. Via une résistance de shunt, nous relèverons la tension consommée aux bornes de cette résistance. Le calcul final sera donc U x I pour obtenir la puissance P.



Les relevés seront notés en temps réel dans un tableau Excel sur une durée de 20s pour lisser la puissance consommée. L’inconvénient de cette méthode est que les relevés ne sont pas automatisés informatiquement. En revanche, l’avantage est que les relevés seront très précis.

## Méthode 3 - Shunt resistor avec Arduino – méthode indirecte



A partir d’un câble d’alimentation mini USB modifié (une résistance de shunt est ajoutée au câble comme dans la méthode 1), nous allons alimenter le Raspberry Pi 2 depuis l’Arduino.

Grâce à la valeur de la tension de sortie (U) sur le port USB de l’Arduino et à notre résistance de shunt (R), nous serons capables de déterminer la puissance en temps réel avec les formules :

I = U/R

Pr = U \* I

U = tension de sortie du Arduino en Volt

I = Intensité de sortie du Arduino en Milli Ampère

R = Résistance de la résistance de shunt en Ohm

Pr = Puissance consommé par le Raspberry Pi 2 en Watt

A la suite de quoi nous utiliserons les commandes linux suivantes pour relever les valeurs :

lsusb -v|egrep "^Bus|MaxPower"

Nous stockerons ensuite ces données avec un script afin de pouvoir visualiser la consommation en fonction de l’utilisation du Raspberry PI 2. L’avantage de cette méthode est que les relevés sont entièrement automatisés et sauvegardés. Il est donc possible, à l’aide d’un Cron, d’effectuer la mesure toute les seconde sur une durée de quelques heures et de tracer la courbe de consommation par la suite.

## Méthode 4 – Outil USB

Après plusieurs recherches, nous avons trouvé un testeur de port USB. Celui-ci se branche sur le câble USB connecté sur le Raspberry. La puissance consommée P est donc affichée en temps réel. Nous devons donc rentrer les valeur à la main dans l’ordinateur pour pouvoir tracer une courbe. Cette méthode est rapide pour récupérer les valeurs, peu coûteuse (5€) mais peu facilement exploitable.



# Conclusion

Après avoir analysé les différentes méthodes, nous avons décidé d’utiliser la méthode 3 qui est la plus adaptée à nos besoins. Nous aurons donc besoin de peu de matériel, et les résultats seront rapidement exploitables.

# Mesure de la consommation

A l’aide de l’Arduino, nous allons relier les deux pattes de la résistance de shunt sur les pins de l’Arduino. Celui-ci recevra le courant utilisé par le raspberry, nous pourrons donc en déduire la puissance.

A l’aide de la commande LS que nous effectuerons de façon récursive à la racine du système, nous allons utiliser le processeur. Nous allons ainsi prouver que le raspberry consomme plus lorsque le processeur est sollicité.

Nous allons également utiliser la commande « top » pour nous permettre de retourner le pourcentage d’utilisation processeur en temps réel.

2,2ohm résistance

0,02

/ 2,2

9mA

// le potentiomètre, branché sur la broche analogique 0

const int potar = 0;

//variable pour stocker la valeur lue après conversion

int valeurLue;

//on convertit cette valeur en une tension

float tension;

float puissance;

float intensite;

void setup()

{

//on se contente de démarrer la liaison série

Serial.begin(9600);

}

void loop()

{

//on convertit en nombre binaire la tension lue en sortie du potentiomètre

valeurLue = analogRead(potar);

//on traduit la valeur brute en tension (produit en croix)

tension = valeurLue \* 3.0 / 1023;

intensite = (tension / 0.2) /10 ;

puissance = 5 \* (tension / 0.2);

//on affiche la valeur lue sur la liaison série

Serial.print("valeurLue = ");

Serial.println(valeurLue);

//on affiche la tension calculée

Serial.print("Tension = ");

Serial.print(tension,5);

Serial.println(" V");

Serial.print("Intensite = ");

Serial.print(intensite,5);

Serial.println(" mA");

Serial.print("Puissance = ");

Serial.print(puissance,5);

Serial.println(" W");

//on saute une ligne entre deux affichages

Serial.println();

//on attend une demi-seconde pour que l'affichage ne soit pas trop rapide

delay(500);

}