**

Linux Embarqué

*Gestion de capteur de température*

Gestion et simulation d'un capteur de température en utilisant une plateforme embarquée de type Raspberry Pi.

BARBESANGE Benjamin – GARCON Benoît

15/11/2015

Linux Embarqué

Gestion de capteur de température

Table des matières

[Introduction 2](#_Toc435366072)

[Présentation de l'environnement 3](#_Toc435366073)

[Le PC hôte 3](#_Toc435366074)

[La carte Raspberry Pi 3](#_Toc435366075)

[Description 3](#_Toc435366076)

[Configuration 3](#_Toc435366077)

[Présentation de la solution 3](#_Toc435366078)

[Capteur de température 3](#_Toc435366079)

[Serveur Raspberry Pi 3](#_Toc435366080)

[Acquisitions 3](#_Toc435366081)

[Site web 3](#_Toc435366082)

[Interactions 3](#_Toc435366083)

[Conclusion 3](#_Toc435366084)

# Introduction

Ce projet s'inscrit dans le cursus de seconde année à l'ISIMA. Le but est d'implémenter la simulation d'un capteur de température sur une machine hôte. Ce capteur devra être géré par un serveur, en utilisant une carte embarquée de type Raspberry Pi.

L'affichage des résultats du capteur sera effectué par le biais d'un site web, également hébergé sur la carte Raspberry Pi. L'utilisateur sera également en mesure d'interagir avec le capteur, pour stopper l'acquisition de température ou modifier la fréquence d'acquisition.

# Présentation de l'environnement

## Le PC hôte

Nous disposons d'un PC hôte sur lequel nous allons simuler le capteur de température. Il est équipé d'Ubuntu 12.04 et permet de communiquer avec la carte Raspberry Pi que nous détaillerons plus tard.

Sur cette machine, nous avons un accès internet ainsi que les droits d'administrateur afin de pouvoir gérer au mieux notre projet.

## La carte Raspberry Pi

### Description

Cette carte fonctionne comme un PC dans un format réduit. Elle dispose d'un processeur, d'une mémoire vive ainsi que d'une mémoire flash (carte mémoire). Nous disposons également de connectiques comme de l'USB, Ethernet, HDMI.

Pour pouvoir faire fonctionner cette carte, nous avons besoin de matériel :

* Un clavier (USB),
* Une souris (USB),
* Un écran,
* Un câble Ethernet,
* Un câble d'alimentation (USB),
* Un adaptateur VGA → HDMI,
* Un adaptateur Carte Mémoire → USB,
* Un câble RS232 → USB.

Afin d'établir la communication avec notre PC hôte, nous allons utiliser la connectique RS232.

### Configuration

Afin de faire fonctionner notre capteur, il nous a été indispensable d'installer des paquets. Pour ce faire, nous avons dû configurer une connexion réseau dont voici les données :

* Adresse MAC : b8:27:eb:3d:c4:9d
* Adresse IP : 172.16.44.15
* Passerelle : 172.16.47.250
* DNS : 172.16.32.250 OU 172.16.32.252

Un serveur ftp est mis en place afin de faciliter les transferts de données entre le PC hôte et notre système embarqué. Son adresse est la suivante : <ftp://172.16.44.15>.

L'interaction avec ce système se faisant par le biais d'un navigateur internet, nous avons installé un serveur web ; dans notre cas, c'est APACHE2 couplé au paquet PHP5, afin de produire une page dynamique.

# Présentation de la solution

## Capteur de température

## Serveur Raspberry Pi

### Acquisitions

Le Raspberry Pi se compose de différents éléments contribuant aux fonctionnement général du capteur. Dans un premier temps, nous trouvons un programme qui ouvre le connexion au port RS232 et se compose de threads :

* Thread de lecture : attend les trames de températures et les acquittements de commandes,
* Thread d'écriture : envoie les commandes et les acquittements de trames de température,
* Thread de commandes : surveille les commandes envoyées à partir du serveur web.

Etant donné que certains threads vont se partager des informations comme les files d'attente des trames ou les booléens d'attente d'acquittements, nous avons décidé d'utiliser des mutex, qui vont ainsi éviter les soucis lors d'une écriture dans la même structure.

### Acquisitions

Les acquisitions de trames de température débutent dans le thread de lecture où l'on va sans cesse lire les caractères envoyés. Lorsque l'on trouve le caractère de début de trame 'W', nous lisons les caractères suivants et les stockons dans une chaîne.

Cette chaîne est ensuite vérifiée. Si la chaîne correspond à ce que l'on attendait, on la place dans la file de trames à afficher, puis on indique que l'on doit envoyer un acquittement avec un booléen.

Le thread d'écriture va ensuite à son tour voir qu'il faut envoyer un acquittement par le biais du booléen. On envoie cet acquittement puis on remodifie le booléen pour indiquer que l'on a envoyé l'acquittement. Juste après ceci, le thread va récupérer la trame dans la file correspondante et l'écrire dans un fichier qui sera lu en PHP pour être affichée sur le site.

### Site web

Notre serveur étant équipé d'APACHE2 ainsi que de PHP5, nous pouvons implémenter l'interface web.

Pour ce faire, nous avons choisi la bibliothèque bootstrap sb-admin. Nous avons choisi cette bibliothèque puisqu'elle permet de créer facilement des pages web dont le rendu est de qualité. De plus, son utilisation est très simple.

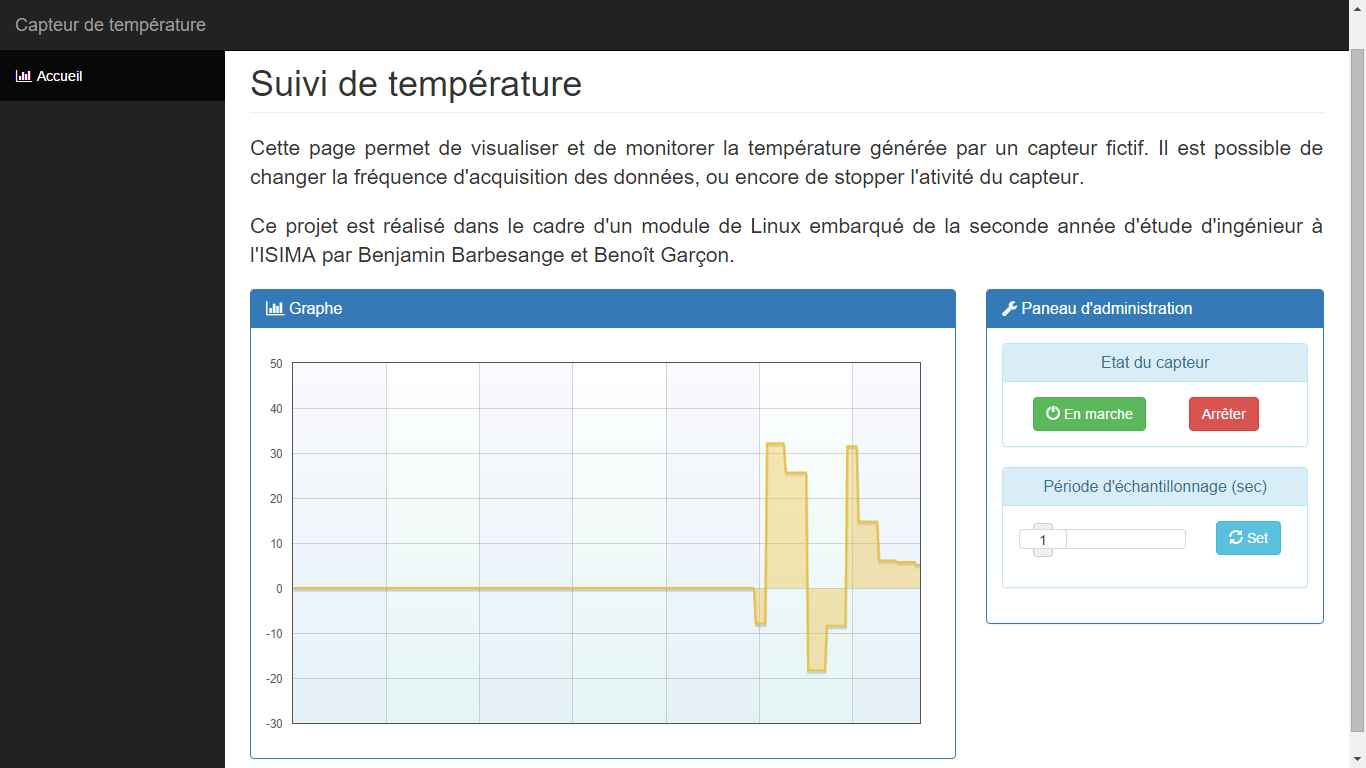


Figure – Capture d'écran du site web

Comme nous pouvons le vois, il est possible de visualiser l'évolution de la température sur un graphe. Le panneau se situant sur la droite permet quant à lui d'activer ou stopper le capteur, et de changer sa fréquence d'acquisition au moyen d'un slider.

Etant donné que la connexion internet est établie de manière locale à l'ISIMA, il n'est pas possible d'accéder à cette page en dehors de l'école. Cependant, une reconfiguration de la connexion permettrait d'y accéder n'importe où dans le monde.

### Interactions

Les interactions sont possibles par le biais du site web présenté précédemment.

Si l'on souhaite démarrer ou stopper le capteur, il suffit de cliquer sur le bouton correspondant. Ceci va appeler un script PHP correspondant, qui va aller écrire dans un fichier l'état dans lequel le capteur doit se trouver. Notre programme de gestion va se charger de lire cet état et d'envoyer la commande correspondant au PC hôte.

Le changement de température se fait de la même manière.

# Conclusion

La finalité de ce projet est qu'il est très complet. En effet, nous avons dû réfléchir à l'organisation de nos différents programmes afin qu'ils collaborent.

Après une phase de test et de découverte du matériel, nous avons configurés les différents éléments dont nous nous sommes servis, tels que la connexion internet, le serveur web.

Il en résulte une application permettant effectivement de visualiser les résultats du capteur de température dans un navigateur, pour toute personne connaissant l'adresse de celui-ci. De plus, l'utilisateur est capable d'interagir en temps réel avec le système, soit pour le mettre en marche, soit pour le stopper ; ou encore changer la fréquence d'acquisition.

Ce projet permet la gestion d'un capteur de température, cependant le fonctionnement serait similaire si l'on souhaitait implémenter la gestion d'arrosage de plantes, ou encore pour des applications de domotique.