

Université Claude Bernard Lyon 1

Rapport de Projet

Application web de visualisation et d'interprétation de consommation électrique personnelle



Axel Danguin – Florian Garcia
11/04/2019

Table des matières

Introduction.....	2
I – Le projet.....	3
A – La proposition	3
B – Prototypage	4
1 – interface	4
2 – circuit de collecte de données	5
II – Les technologies utilisées	6
Python	6
Django	6
SQL Lite.....	6
Materialize.....	6
Le Raspberry Pi	6
III – Travail réalisé et avancement	7
A – L’état actuel du projet	7
B – Les Difficultés et limites actuelles	7
C – Ce qui est prévu pour le futur	9
IV – Bilan.....	10
V – Liens, sources.....	11

Introduction

Depuis environ 2010, Enedis installe une nouvelle génération de compteurs électriques, le compteur Linky. Intelligent, et connecté, il a la capacité de communiquer et d'effectuer certaines opérations à distance, Enedis espère ainsi pouvoir intervenir plus rapidement en cas de problème, et prévoir par exemple les pics de consommation.

Malgré les controverses, débats et poursuites légales qui entourent ce compteur, Enedis poursuit son installation avec pour objectif d'équiper le territoire français d'ici le début des années 2020. Malgré tout cela on ne peut ignorer les intérêts qu'il présente.

Il peut notamment effectuer des relevés de consommation réguliers et les communiquer avec EDF. Enedis propose aux usagers de visualiser ou de télécharger un récapitulatif de leur consommation sous la forme d'une table de données. Notre projet a pour but de réaliser un outil de visualisation des données qui tire parti de ces informations pour permettre aux personnes disposant de ce compteur de suivre précisément leur consommation électrique.

C'est dans ce climat de doute de la part des utilisateurs qu'il est important de démystifier et de rendre un peu de contrôle, ou du moins, de compréhension, à l'individu. Ce projet s'inscrit aussi dans cette lignée.

I – Le projet

A – La proposition

Notre proposition, contrairement aux solutions précédemment réalisées pour ce projet, ne se repose pas sur les données fournies par Enedis, sur leur site internet. Nous avons choisi d'utiliser la prise télé-info du Linky, accessible sur sa façade. Ceci apporte non seulement une granularité plus fine (relevés plus fréquents), mais aussi une plus grande variété d'informations.

L'objectif final du projet, est de pouvoir relever, traiter, stocker, et afficher les informations fournies par le Linky sur une interface web, et d'effectuer l'ensemble de ces opérations via une machine unique, comme un Raspberry pi, pour rendre l'installation minimaliste, et non onéreuse. L'application devra fournir une vue résumée de la consommation sur une période courte, et offrir la possibilité d'un affichage réglable pour comparer, et visualiser la consommation sur des données et durées variables. Il est aussi prévu de pouvoir importer, et exporter les données relevées.

Notre proposition présente l'avantage de la désintermédiation, on soustrait Enedis de l'équation, en lisant les données à la source, ce qui rend notre application indépendante d'une connexion internet, elle fonctionne entièrement en local.

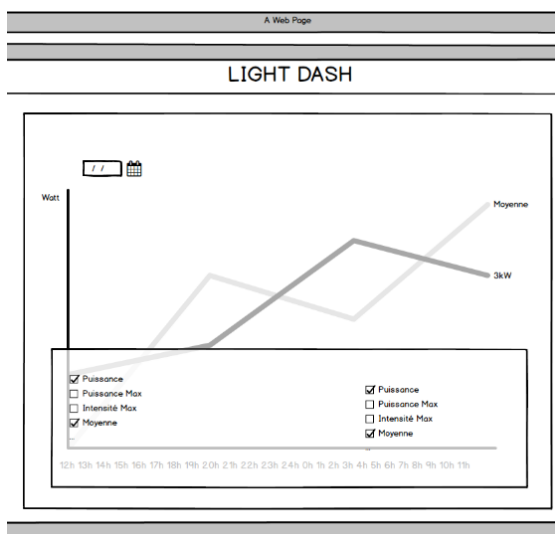
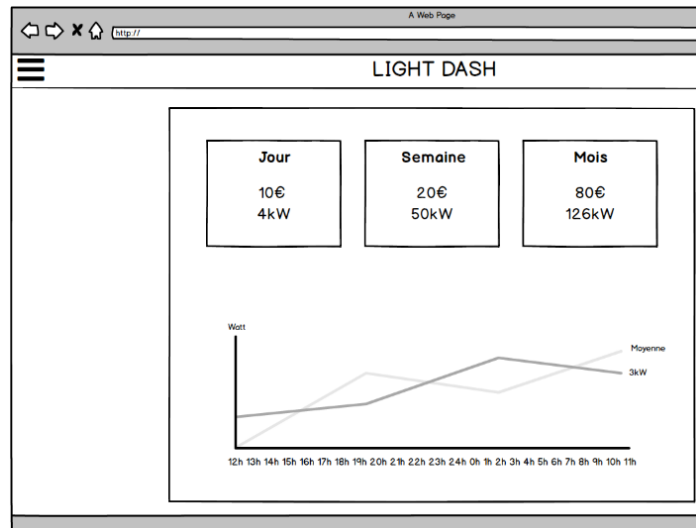
Il est aussi important de noter qu'accéder au port télé-info du Linky ne nécessite **pas de modification ou de violation du matériel d'Enedis**, il est **accessible, et sans sécurité**.

B – Prototypage

Avant les tests et le développement, nous avons réalisé un prototype d'interface, et un schéma électrique, il s'agit d'un genre de preuve de concept.

1 – interface

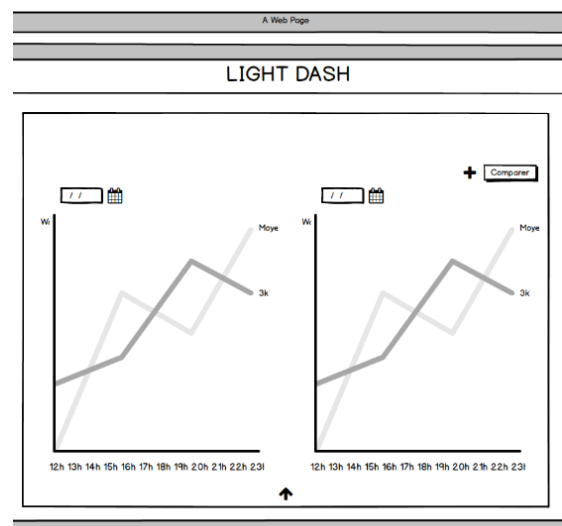
La première page présente un graphe de la consommation récente et quelques chiffres clefs, l'idée étant de proposer à l'utilisateur les informations les plus utiles, sans que celui-ci ait à accéder aux fonctions avancées du logiciel.



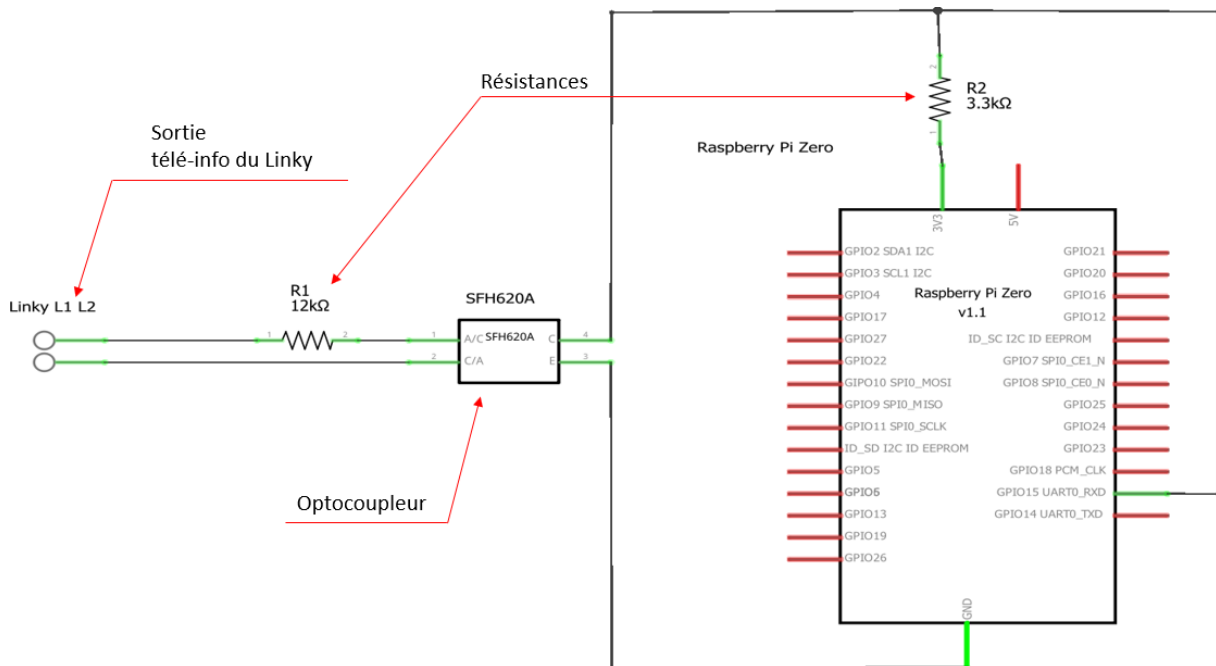
Le second écran, la vue avancée, permet de visualiser n'importe laquelle des mesures disponibles et de choisir une échelle de temps, par un menu dépliant.

Le second écran permet aussi d'afficher plus d'un graphique. On peut choisir de dupliquer un graphe pour effectuer des comparaisons, ou simplement d'en avoir plusieurs simultanément.

Le but de cette vue personnalisable est de laisser un contrôle total à l'utilisateur, il peut ainsi effectuer toutes les mesures souhaitées sans avoir à changer de page.



2 – circuit de collecte de données



Le circuit est simple la liaison entre le Raspberry et le Linky est exploitée comme une liaison série. Il est bon de noter que la fréquence sur cette liaison est très basse, certains équipements ne sont plus capables de descendre assez bas pour exploiter ce port, le Raspberry en est cependant capable.

Les propriétés du signal sont les suivantes : base 7 bits avec un bit de parité, à 1200 bauds.

Ce circuit fait usage d'un **optocoupleur**, qui peut transmettre un signal d'un circuit à un autre sans contact galvanique (liaison conductrice). Il s'agit d'une sécurité qui isole le compteur du Raspberry.

L'ordre des ports sur le Linky n'importe pas. Puisqu'il s'agit d'une connexion série, n'importe quel logiciel, bibliothèque ou Framework capable d'observer un port série, est utilisable.

II – Les technologies utilisées

Python

Python est un langage de haut niveau interprété, et multiplateforme, il s'agit du langage principal pour ce projet. Il s'agit d'un choix pratique puisque Django nécessite l'installation de python, la compatibilité inter plateforme assure que l'on puisse déplacer le projet d'une machine et d'un système d'exploitation à un autre, et l'absence de compilation rend l'application facilement modifiable et modulaire. On peut facilement ajouter des librairies qui facilitent le développement, comme APScheduler qui permet de planifier des tâches pour les exécuter à intervalle régulier.

Django

Django est un Framework Web en python qui a pour avantage d'être modulaire, intègre un ORM (Mapping objet-relationnel) pour une base de données SQLite. Django dispose d'une architecture respectant le pattern MVC (Modèle-Vue-Contrôleur). L'utilisation de Django permet de développer des applications web sans avoir à recourir à du développement lourd en PHP. Utiliser Django nous permet aussi d'utiliser python pour la quasi-totalité du projet.

SQLite

SQLite est un système de base de données très léger (pas de système client-serveur, directement intégré dans l'application) qui est inclus dans Django, et une bibliothèque standard de Python. Son interfaçage avec Django et Python en fait le choix optimal pour ce projet, la base de données dont nous avons besoin étant simple (pas plus de 5 tables).

Cette base de données nous permet de stocker les données extraites du Linky par le Raspberry pi.

Materialize

Il s'agit d'un Framework CSS (frontend) qui s'appuie jQuery. Il respecte la charte graphique *Material Design* de Google. Il a comme caractéristique de faciliter le responsive design et permet d'avoir un affichage parfait sur smartphone. Se basant sur la charte graphique des smartphones Android, il permet une bonne continuité de l'interface.

Chart.js

Un Framework JavaScript open source permettant l'affichage de graphiques et de diagrammes.

Le Raspberry Pi

Il a le double avantage de pouvoir servir de système d'acquisition de données et d'héberger l'ensemble des composants du projet, c'est une machine légère et économe, qui est peu onéreuse. Le Raspberry Pi est simple à obtenir et à utiliser, et s'inscrit donc dans la logique de ce projet, qui se veut accessible pour n'importe quel possesseur d'un Linky.

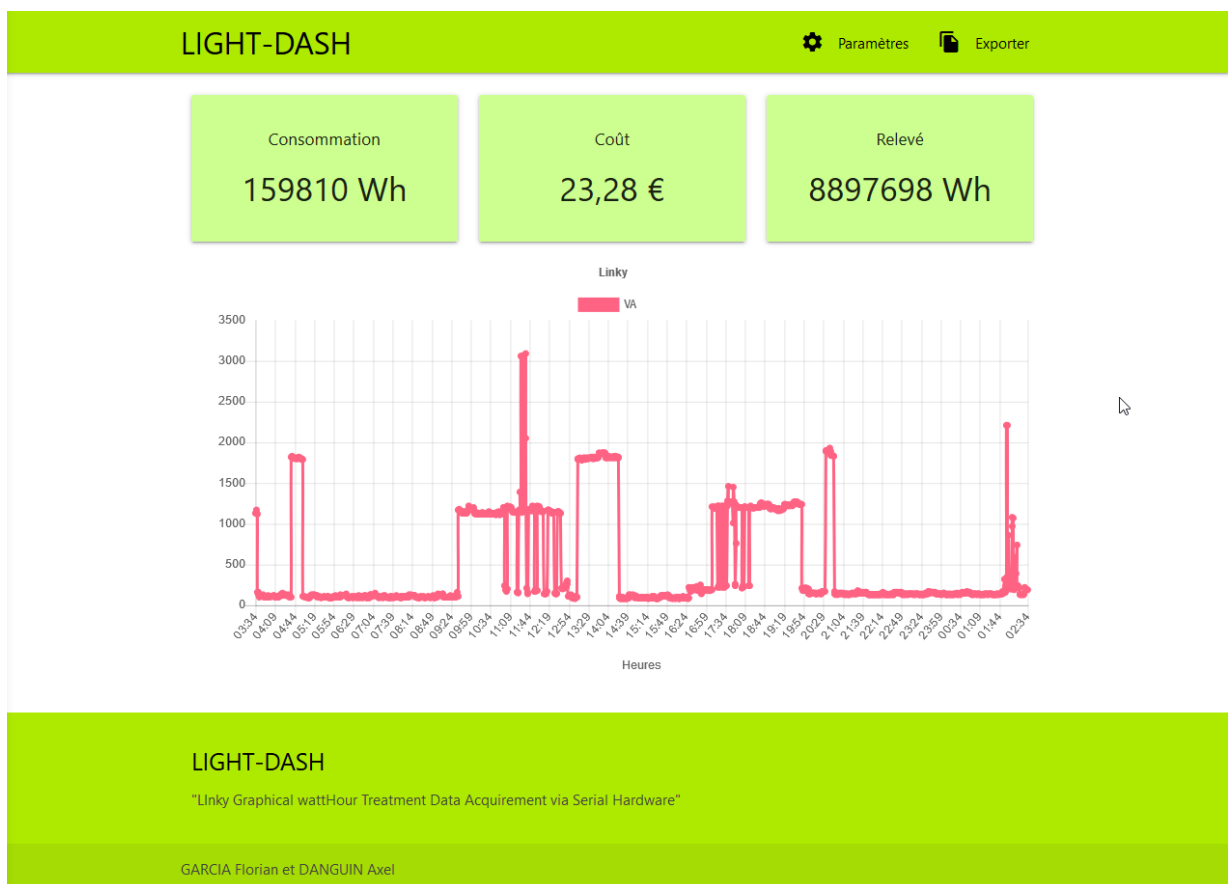
III – Travail réalisé et avancement

A – L'état actuel du projet

Au moment d'écrire ce rapport, le projet n'est pas terminé, il ne remplit pas l'ensemble des fonctions proposées à son début.

Il est capable de récupérer et de sérialiser les données fournies par un Linky en temps réel, et de les stocker dans une base de données.

Il contient un site web fonctionnel, capable d'afficher un graphique et les chiffres clefs, comme vu dans le prototype d'interface.



LIGHT-DASH

Paramètres

Abonnement HPHC

Tarif Heures Pleines

€ 0,1579

Tarif Heures Creuses

€ 0,1228

Début Heures Creuses

22:00:00

Fin Heures Creuses

06:00:00

Fréquences de paiement

Bimestrielle

Jour de la facturation

14

Début de facturation

Février

APPLIQUER

LIGHT-DASH

"Linky Graphical wattHour Treatment Data Acquisition via Serial Hardware"

GARCIA Florian et DANGUIN Axel

On peut aussi y régler des options concernant le type d'abonnement de l'utilisateur, les tarifs, la fréquence de paiement...

L'application contient aussi de quoi exporter les données récupérées dans un fichier JSON.

La base de données a été étendue à postériori, et un ensemble de tâches planifiées a été développé, dans le but de créer des tables avec une granularité moins élevée (relevés par heure, jour, mois ...) pour rendre l'accès aux données sur une grande période moins lourd. Cet ajout s'est prouvé être nécessaire, puisqu'un grand nombre d'accès à la base de données est nécessaire pour construire les graphiques.

B – Les Difficultés et limites actuelles

Nous avons choisi d'utiliser de Chart.js dans ce projet, celui-ci nous a limité sur la personnalisation du graphique de la consommation. De plus il ne permet pas une bonne lisibilité avec beaucoup de points et n'est pas Responsive Design, ce qui est problématique pour une utilisation sur smartphone.

Le projet n'est pas terminé, mais cela est principalement dû à la quantité de fonctions que nous avons prévues, dès le début du développement. Lorsque nous avons commencé le projet, nous avons volontairement établi un cahier des charges très lourd, puisque nous avons l'intention de maintenir le projet après la fin de la licence. La finalité de ce projet étant de proposer une application utile à tout utilisateur de Linky.

C – Ce qui est prévu pour le futur

Après la fin de la licence, il est prévu de compléter les fonctions clefs du projet, ainsi que de proposer un système d'installation automatisé, et en assurer la maintenance.

Notre projet est publié sous la licence GNU GPL V3, et pourra donc être réutilisé par d'autres développeurs, nous allons donc continuer de fournir une documentation complète pour celui-ci.

Il est aussi prévu d'envisager une compatibilité avec des compteurs électriques plus anciens, certains d'entre eux ayant des fonctionnalités similaires au Linky.

Axel :

Une des idées que j'ai pour le futur, est de développer une solution alternative, qui ne repose pas sur une application web, avec le recul, il me semble qu'en considérant la forme actuelle de notre projet, la composante web, ne porte finalement plus énormément d'intérêt, si ce n'est, de pouvoir visualiser les données, sur n'importe quel écran connecté au réseau, en vérité, il me semble que cela serait probablement plus sécurisé.

Florian :

Je pense continuer le projet afin d'intégrer les dernières fonctionnalités clefs. Peut-être je changerai le serveur Web Django par web.py afin de tester cette alternative. Mon objectif de proposer cet outil pour le plus grand nombre et de faciliter la mise en place par un utilisateur lambda. La meilleure évolution pour ce projet serait de l'intégrer dans un système domotique grand public pour rendre ses données encore plus utiles. Par exemple une détection de surconsommation permettrait d'éteindre certains appareils automatiquement.

IV – Bilan

En conclusion, bien qu'il eût été souhaitable que le projet soit à un stade de développement plus avancé (on déplore surtout l'absence de la page « avancée ») nous avons néanmoins produit une application capable d'effectuer son rôle principal, c'est-à-dire de visualiser la consommation électrique de son utilisateur. Elle est légère, et indépendante de tout service externe.

Prendre une voie différente de celle proposée par Enedis s'est prouvé être intéressant, puisque cela nous a ouvert à bien plus de données, et une précision plus élevée, que ce qui est proposé sur le site internet d'Enedis.

V – Liens, sources

Dépôt GitHub : <https://github.com/Jikhai/LIGHT-DASH>

Documentation du projet (FR/EN) : <https://jikhai.github.io/LIGHT-DASH/>

Site D'Enedis : <https://www.enedis.fr/>

Prototype d'interface réalisé avec [balsamiq mockup](#).