



Date : 06/2018

Auteurs : -BENDOU Yassir
 -BENGHABIT Ilyas
 -HAMDI Mohamed

Destinataires : -Groupe de pilotage du projet de développement
 -Mr. BAZIN Jean-Noël
 -Mr. JEZEQUEL Michel

Rapport technique V2

*Projet N°10: Mise-en place d'un réseau de capteurs de température
pour évaluer l'efficacité du chauffage de la Maisel*

Résumé:

Afin d'évaluer l'efficacité du chauffage dans la Maisel, il est nécessaire de pouvoir suivre l'évolution de la température au sein de ses bâtiments.

A travers ce projet, notre groupe doit mettre en place un réseau de capteurs qui permettra à la direction de la Maisel un tel suivi.

Pour cela nous avons réparti le travail en trois grandes parties.

La première consistait en la détection de température par un capteur thermique. Pour ce faire, nous avons d'abord réalisé l'implémentation du module thermique sur la carte arduino, nous avons aussi construit des boîtiers pour contenir les capteurs, et réaliser un système d'alimentation pour la carte arduino permettant une alimentation sur secteur et par piles de celle-ci au même temps.

Dans la deuxième partie, qu'on peut nommer partie Web, nous nous sommes dirigés vers la conception d'une interface pour l'utilisateur. Ainsi, on a créé une plateforme Web sous django et conçu une interface ergonomique qui va être utilisée par l'administration de la Maisel pour accéder aux données de températures.

La dernière partie consistait en l'envoi des données depuis la carte Arduino vers le serveur Web, leur archivage et affichage. Pour cela nous avons d'abord implémenté le shield ethernet sur la carte arduino. Une base de donnée gérée par Django et l'utilisation de ressources Javascript ont été aussi nécessaires pour pouvoir archiver les données de températures reçues et les afficher à l'utilisateur sous forme de graphes de température temps réel.

Finalement, Le réseau réalisé permet maintenant de mesurer la température dans n'importe quel lieu où est posé un capteur, d'envoyer les données de température via Wifi au serveur Web et afficher ces données sur la plateforme.

Rédacteur : Ilyas Benghabrit

Relecteur : Yassir Bendou

Sommaire :

1.Introduction.....	4
2.Développement.....	5
2.1.Mesure de la température.....	5
2.2.Serveur Web.....	5
2.2.1.Framework utilisé: Django.....	5
2.2.2. Base de données.....	6
2.2.3. Les étapes de développement.....	6
2.2.4. Sécurisation de la plateforme.....	7
2.2.5. Déploiement du serveur.....	8
2.3.Interface de visualisation des données.....	9
2.4.Envoi et archivage des données.....	12
2.4.1.Implémentation du shield ethernet sur l'Arduino.....	12
2.4.2.Protocole d'envoi.....	13
2.4.3.Archivage et affichage des données de température ..	13
2.5.Alimentation et protection des capteurs.....	14
2.5.1.Conception des boîtiers.....	14
2.5.2.Alimentation de la carte Arduino.....	16
3.Conclusion.....	18
Références bibliographiques.....	19
Glossaire.....	20
Annexes.....	21

Introduction :

La Maisel loge une grande partie des étudiants de l'IMT Atlantique sur le campus de Brest. Le confort de ces étudiants est un point important. Il leur permet d'étudier dans de bonnes conditions et de pouvoir ainsi être à jour dans leurs cours et projets, et de se reposer après de longues journées à l'école dans de bonnes circonstances. Pour pouvoir offrir cela, et vu le temps qu'il fait à Brest le long de l'année scolaire, il est indispensable d'avoir un chauffage efficace au sein des bâtiments de la Maisel. Ainsi, la maison des élèves désire évaluer l'efficacité du chauffage dans ses bâtiments. Pour cela, son administration a besoin d'un système qui lui permet de suivre la température dans différentes chambres en temps réel.

Notre projet consiste en la mise en place d'un réseau de capteurs de température qui permet d'envoyer les données de températures via Wifi à un site Web accessible par l'administration de la Maisel.

Dans un premier temps, nous allons décrire la capture de température : les outils nécessaires et les démarches qui ont permis cela. Ensuite, nous nous intéresserons à la conception de l'interface du site utilisée pour l'accès aux données. Après nous allons détailler comment l'envoi des données de température a pu être fait vers le site Web. Et finalement, nous expliciterons la manière dont les boîtiers contenant les capteurs ont été réalisés et comment ces derniers seront alimentés.

Rédacteur : BENGHABRIT Ilyas

Relecteur : HAMDI Mohamed

Développement :

Mesure de la température :

La première partie de notre projet consiste à monter un capteur capable de mesurer la température afin que cette donnée soit traitée plus tard. Nous nous sommes basés sur des cartes Arduino et un capteur thermique DHT22 permettant de mesurer la température. Le montage est comme tel :

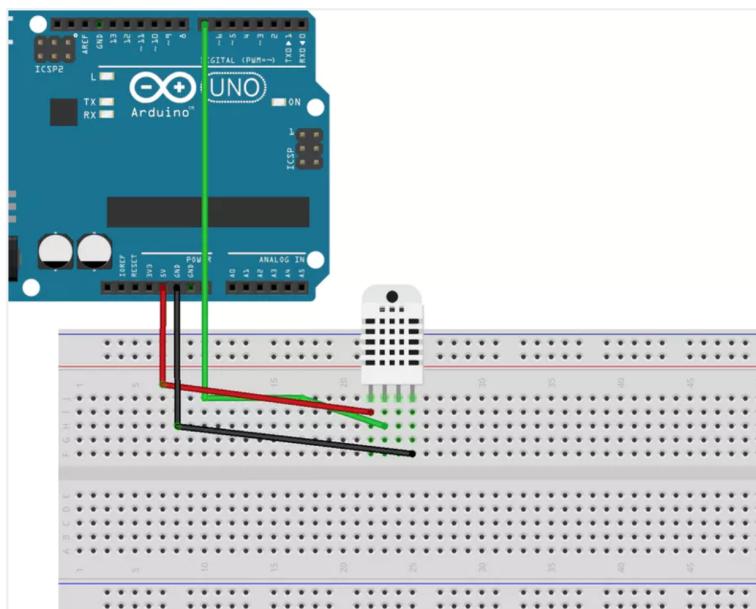


Figure 1 : Montage du module de température DHT22

Pour permettre un suivi en temps réel, la température est mesurée chaque 5 minutes et est envoyée après au site web.

Serveur Web :

Framework utilisé : Django

Une partie de notre projet consiste à mettre en place un serveur web capable d'interagir avec les utilisateurs, de recevoir les données de température et les afficher proprement : Notre plateforme doit être accessible à la fois avec autant d'utilisateurs qu'elle désire et doit pouvoir archiver autant de données que souhaitable. Nous avons donc opté pour Django [1], qui est un framework Web Python open source de haut niveau permettant un développement rapide et un design propre et pragmatique. Étant assez complet, ce framework prend en charge une grande partie des tracas du

développement Web, permettant ainsi de se concentrer sur l'écriture de l'application sans avoir besoin de tout recréer.

Django permet, à partir du contrôleur (Figure 2), de faire le lien entre le serveur web, la base de donnée et le client émettant des requêtes. Dans notre cas, nous disposons de deux clients : L'utilisateur qui est la Maisel et les capteurs qui envoient également des requêtes Http.

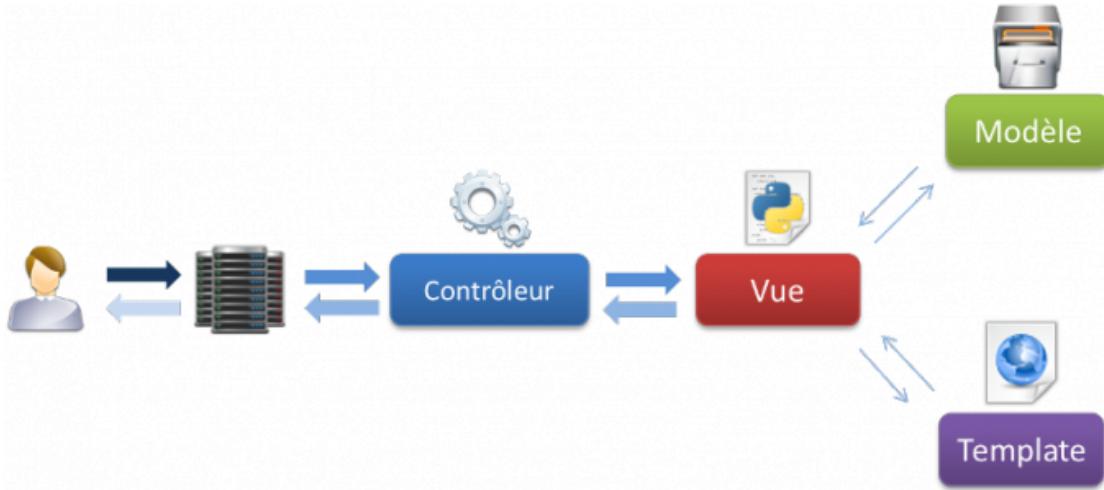


Figure 2 : Architecture du serveur sous Django

Sous la demande de notre tuteur, nous avons partagé une partie de notre code permettant la réalisation du serveur web sous Django.

Base de données

Comme tout site web, sa réalisation requiert une base de données permettant de stocker et gérer les données. Django étant assez complet, lors de la création d'une nouvelle application sous Django, une base de donnée SQL est créée et lui est associée. Nous avons rajouté un administrateur (superuser en django) qui a accès à cette base de donnée via une interface graphique créée par django ou via un shell Python.

Les étapes de développement

Les modèles utilisés:

Une fois que nous avons créé une nouvelle application avec django sous le nom de "collecte". La première étape de développement du serveur consistait à créer les différents objets qui seront manipulés par le serveur web et la base de données. Dès lors, nous avons créée 3 classes comme suivant:

-Room : classe des chambres de la maisel et qui contient deux attributs, le numéro de chambre et le bâtiment où se situe la chambre.

-Capteur : classe des capteurs qui ont été déployés à la maisel. Cette classe contient deux attributs, l'identifiant du capteur et une chambre qui lui est affectée (Cette dernière pourra être modifiée par l'utilisateur).

-Température : classe des températures qui seront envoyées par les capteurs. Cette classe contient trois attributs, la valeur de la température, la date de prélèvement de la température et la chambre où cette température a été prélevée.

Urls et views:

Une fois ces objets créés, nous avons définis les différents urls qui seront utilisés [2]. Chaque url renvoie la requête http du client à une fonction “view” qui lui est associée, ce sont les views qui permettent d’interagir avec la base de donnée et de répondre à la requête du client en lui affichant un contenu html qui correspond à sa demande.

Sécurisation de la plateforme :

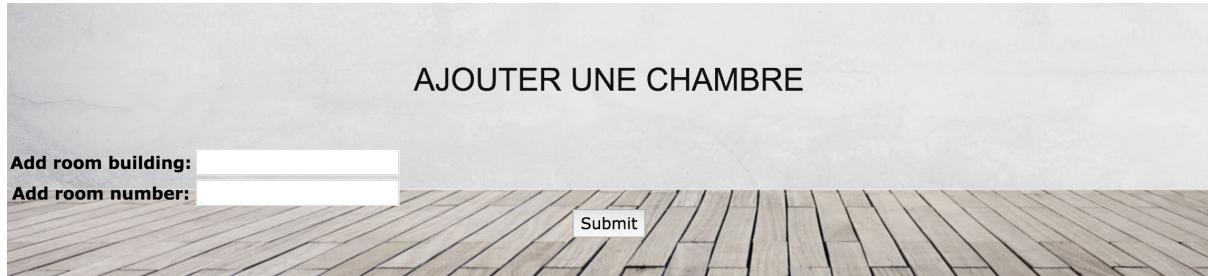
Un site web déployé est toujours susceptible à des attaques. Afin de garantir une longévité du site web, nous avons mis en place plusieurs paramètres de sécurité.

La première couche de sécurité consistait à la mise en place d'un système d'authentification de l'utilisateur (Figure 3). Toutes les pages qui permettent une modification des données nécessitent une authentification afin d'y avoir accès. Un utilisateur ayant accès est nommé “user”, ce dernier ne peut être créé que par un administrateur. Chaque user possède un login, un mot de passe et une adresse mail.



Figure 3 : Interface d'authentification de l'utilisateur

La deuxième couche consistait à passer toutes les requêtes sensibles de l'utilisateur à travers des forms (Figure 4): Afin que notre client (La maisel) souhaite agir sur les éléments de la base de données, nous avons créé des forms lui permettant une manipulation simple des données sans compromettre la sécurité du serveur. Les forms permettent d'envoyer au serveur des requêtes POST (Figure 5) au lieu de GET et qui sont plus sécurisées.



AJOUTER UNE CHAMBRE

Add room building: _____

Add room number: _____

Submit

Figure 4 : Exemple de forms permettant d'ajouter une chambre à la base de données

[17/May/2018 00:53:30] "POST /addRoom HTTP/1.1" 200 46

Figure 5 : Exemple de requête POST

Finalement, afin de consolider la sécurité du site web contre les attaques web, nous avons activé plusieurs paramètres de sécurité :

-https : protocole qui garantit théoriquement la confidentialité et l'intégrité des données envoyées par l'utilisateur tel que la saisie de mots de passe.

-csrf protection qui permet d'éviter les attaques de type cross-Site Request Forgery en transformant des caractères susceptibles de permettre une infiltration de l'attaquant en caractères non nuisants.

- X-XSS : Protection bloque les attaques de types X-XSS.

Déploiement du serveur :

Cette étape consiste à rendre le site web accessible à tous les utilisateurs. Nous avions prévu d'héberger le serveur web chez le resel. Toutefois, dû au manque de personnel dans le resel pendant la phase de notre projet, nous avons opté pour une solution extérieure. Ainsi nous avons hébergé le serveur web dans le domaine "aneutron.me", le site est disponible

à partir de n'importe quel machine reliée à internet sur le lien suivant :
[“https://maisel.aneutron.me/”](https://maisel.aneutron.me/)

Rédacteur : BENDOU Yassir
Relecteur : BENGHABRIT Ilyas

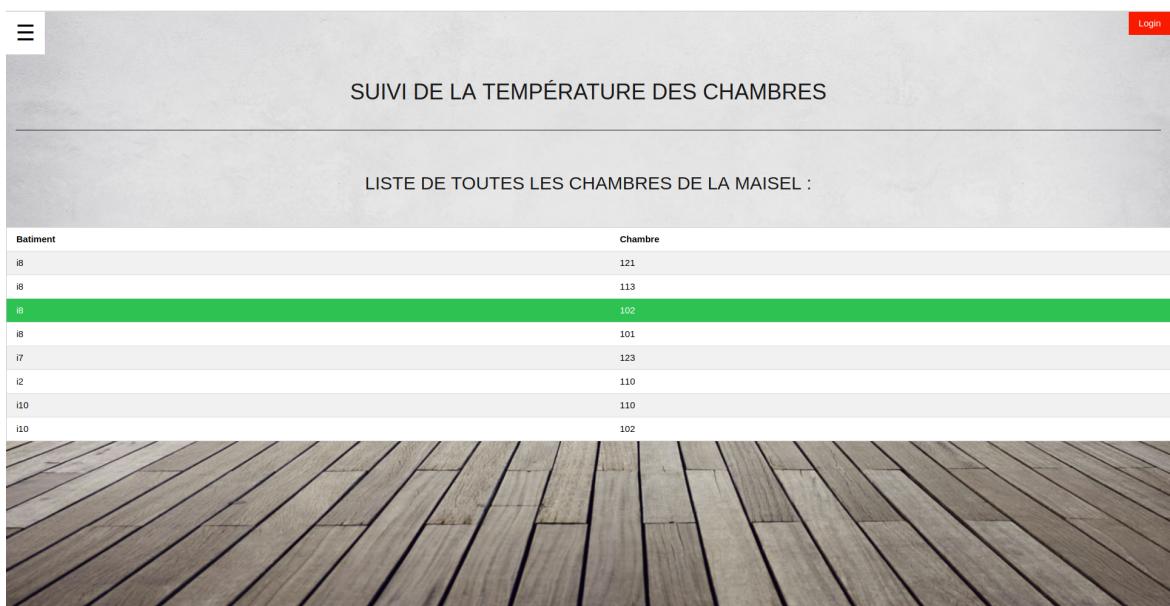
Interface de visualisation des données

Le site web créé doit être à la fois ergonomique et esthétique.
Pour assurer cela un code HTML simple s'est avéré insuffisant .
Ainsi pour subvenir aux différents besoins du site graphe , Bouton , tableau et une interface plus dynamique un choix technique plus adapté a été adopté : l'utilisation de JavaScript et de CSS (Cascading Style Sheets)[3][4].

L'utilisation de CSS présente plusieurs avantages pour la mise en forme du site puisqu'il permet de modifier tous les titres du site en une seule fois en modifiant une seule partie du fichier CSS. Sans cela , on serait obligé de modifier les pages du site séparément. Dans le cas de notre site le seul fichier qui gère le design est : base_generic.html et après cela chaque fichier html commence par le code suivant qui permet d'appliquer le design :

```
<!DOCTYPE html>
{%
    extends "collecte/base_generic.html"
    %}
{%
    block content
%}
```

- Les Tableaux:



Batiment	Chambre
i8	121
i8	113
i8	102
i8	101
i7	123
i2	110
i10	110
i10	102

Figure 6 : Exemple de tableau

Les tableaux sont créés comme le montre la figure 6 avec une coloration verte lorsqu'on met le curseur sur une ligne pour faciliter l'utilisation et le lecture des valeurs[3] .

- Les Graphes :

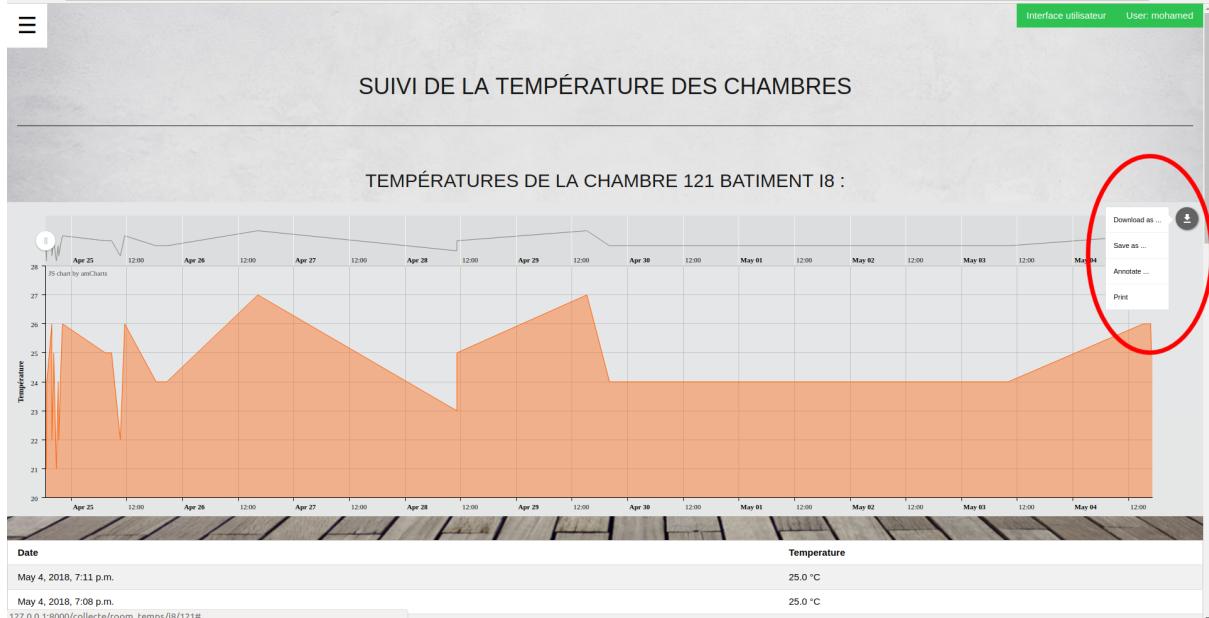


Figure 7 : Exemple de graphe

Cette page de chambre ci-dessus (figure 7) offre à l'utilisateur la possibilité d'accéder au donnée de température, d'enregistrer le graphe en PDF ou les données en CSV grâce à l'outil "Download as" (entouré en rouge) et assure aussi d'autres fonctionnalités comme l'impression .

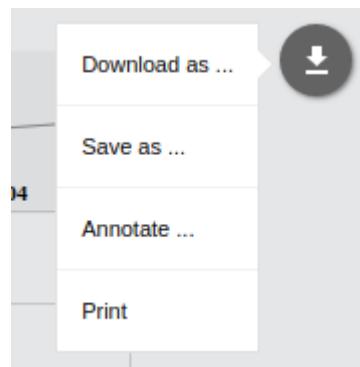


Figure 8 : Bouton de fonctionnalités du graphe

Il a été implémenté grâce à la librairie JavaScript Amcharts[4] .

- Le bouton Menu

Pour faciliter la navigation et rendre le site plus ergonomique , On a muni le site d'un bouton Menu qui contient des liens vers les pages utiles. Le code

suivant dans le fichier base_generic.html permet l'affichage et le fonctionnement du bouton dans toutes les pages du site.



Figure 9 : Bouton Menu

- L'interface utilisateur

Si l'utilisateur est authentifié , 2 boutons verts remplacent le bouton login rouge et il a accès à l'interface utilisateur pour pouvoir effectuer l'une des 4 opérations suivantes cf figure 10 et 11 :



Figure 10 : Interface utilisateur A



Figure 11 : Interface utilisateur B

Cette interface a été mise en place grâce au framework w3.css [3].

Au début on vérifie si l'utilisateur est bien connecté si oui on lui affiche les boutons contenant les liens vers les pages de gestion sinon on affiche le bouton login .

Rédacteur : HAMDI Mohamed

Selecteur : BENGHABRIT Ilyas

Envoie et archivage de données

Implémentation du shield ethernet

L'envoie des données a été fait par cable ethernet au lieu du Wifi à cause d'un retard d'achat des modules wifi (Livraison reçue en mi-mai). Nous avons alors eu recours à cette solution alternative en utilisant un shield ethernet présent au Fablab. Ce shield (Figure 12) permet à la carte Arduino d'envoyer des requêtes http au serveur web [5].

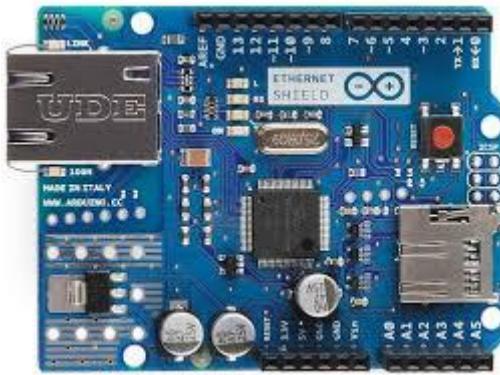


Figure 12 : Shield Ethernet

Protocole d'envoi:

Une fois la mesure de température faite par le capteur thermique implémenté sur la carte Arduino, cette dernière envoie une requête http au serveur web contenant la donnée de température et l'identifiant du capteur.

Afin de sécuriser l'envoie d'une telle donnée sensible, nous avons mis en place une mesure de sécurité en ayant recours à un chiffrement de données symétrique : Le cryptage de Vigenère. Ce chiffrement est assez robuste pour intimider des attaquants amateurs. La requête envoyée (Figure 13) est décryptée par le serveur web en utilisant la même clé de chiffrement et s'assure que la donnée envoyée contient une chaîne de caractère spéciale, autrement la requête est rejetée.

```
[17/May/2018 00:04:56] "GET /collecte/send/%x~oe*js(_j@EB97J HTTP/1.1" 200 9
```

Figure 13 : Envoi de données chiffrées

Affichage et archivage des données de température :

Une fois que les données de températures envoyées par le capteur sont reçues par le serveur web, un nouvel objet de type Température est créé dans la base de données du serveur. Cette température est affectée à la chambre associé au capteur et la date de prélèvement de la température est enregistrée au moment de la réception puis est affichée sur une page html (Figure 14) sous forme de graphe en temps réel et de tableau, permettant de remonter à toutes les températures mesurées dans une chambre.

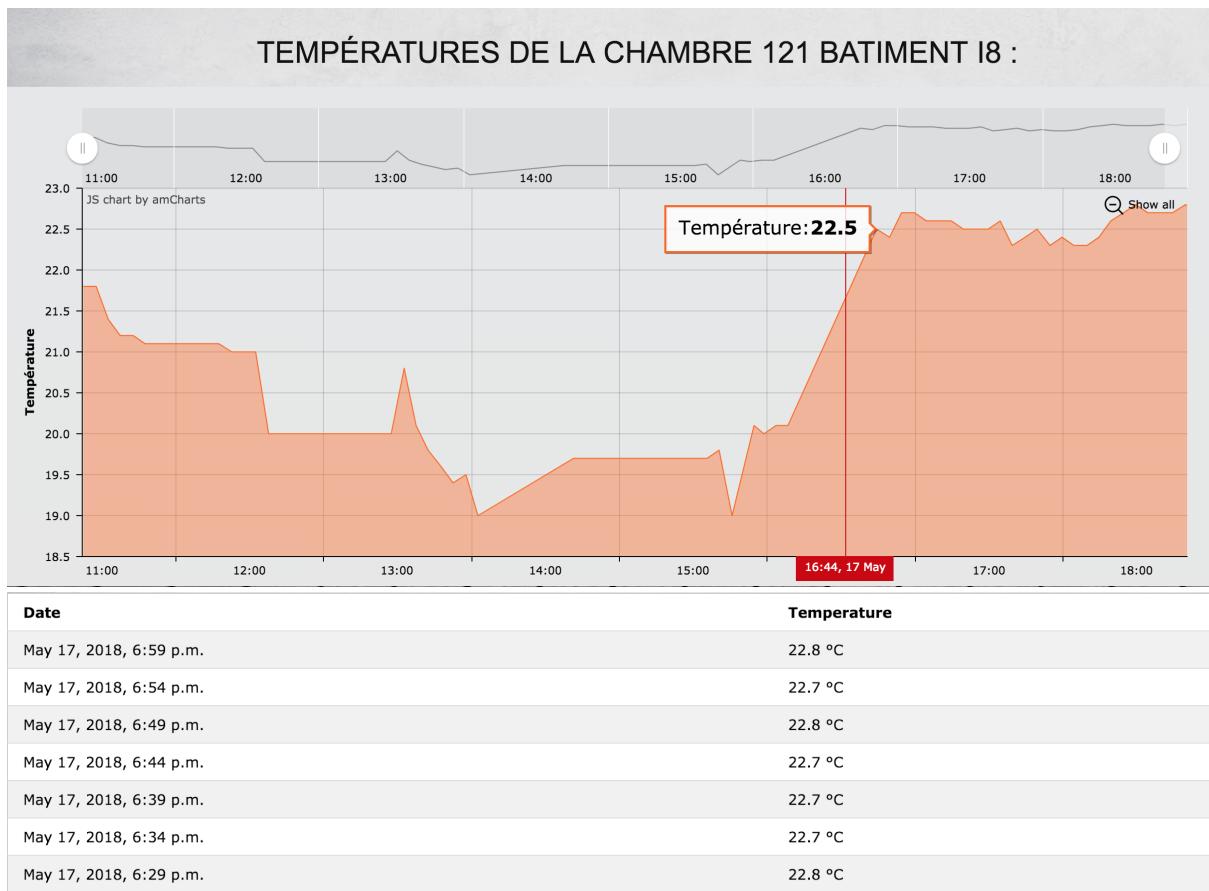


Figure 14 : Affichage des données de température

Rédacteur : BENDOU Yassir

Selecteur : BENGHABRIT Ilyas

Alimentation et protection des capteurs:

Conception des boîtiers :

Pour protéger les capteurs, la solution retenue a été, comme annoncé dans le cahier des charges fonctionnel un boîtier sous forme parallélépipède en bois. Sauf que après une première réalisation à l'aide de la scie manuelle disponible au Fablab avec les dimensions mentionnées dans le Cdcf (15 cm x 15 cm x 10 cm), il s'est avéré que le boîtier était un peu très grand. Ainsi nous nous sommes résiliés à des boîtiers sous forme de cube de 13cm de côté (Figure 16).

Nous avons essayé de réaliser des boîtiers pratiques à l'utilisation. Cela veut dire que notre objectif a été de permettre une facilité de chargement de la

carte arduino et téléversement du code sur celle-ci, et aussi une possibilité d'ouverture et fermeture du boîtier. Pour cela, nous avons choisi de réaliser des boîtiers grâce à l'imprimante laser du Fablab.

On a donc fait le choix de réaliser le plan du boîtier sur le logiciel CorelDraw [6].

Sur ce plan on a fait figurer les 6 faces du cube comme suit :

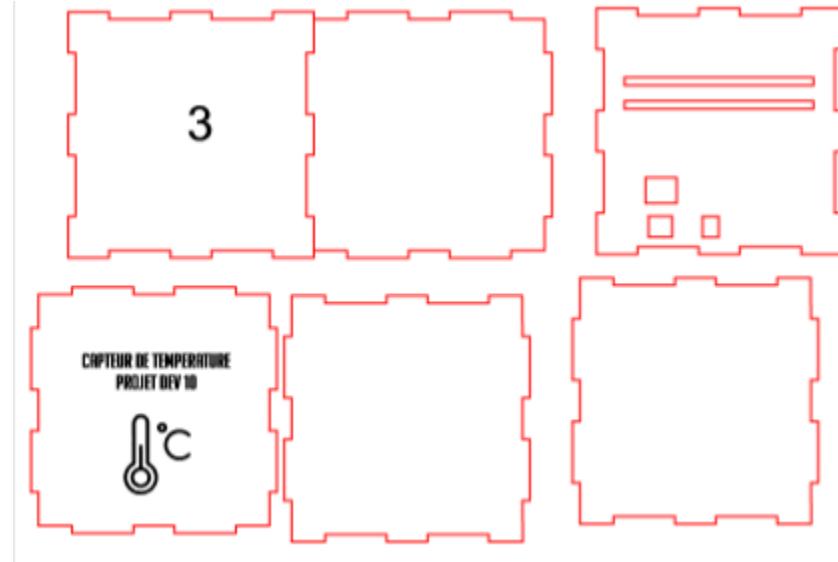


Figure 15 :Plan des faces du boitier sur le logiciel CorelDraw

Les côtés contiennent des encoches. Cela va permettre l'ouverture et fermeture du boîtier sur une face, par contre, les autres faces seront collés en passant de la colle spéciale sur la surface de ces encoches.

Produit final :

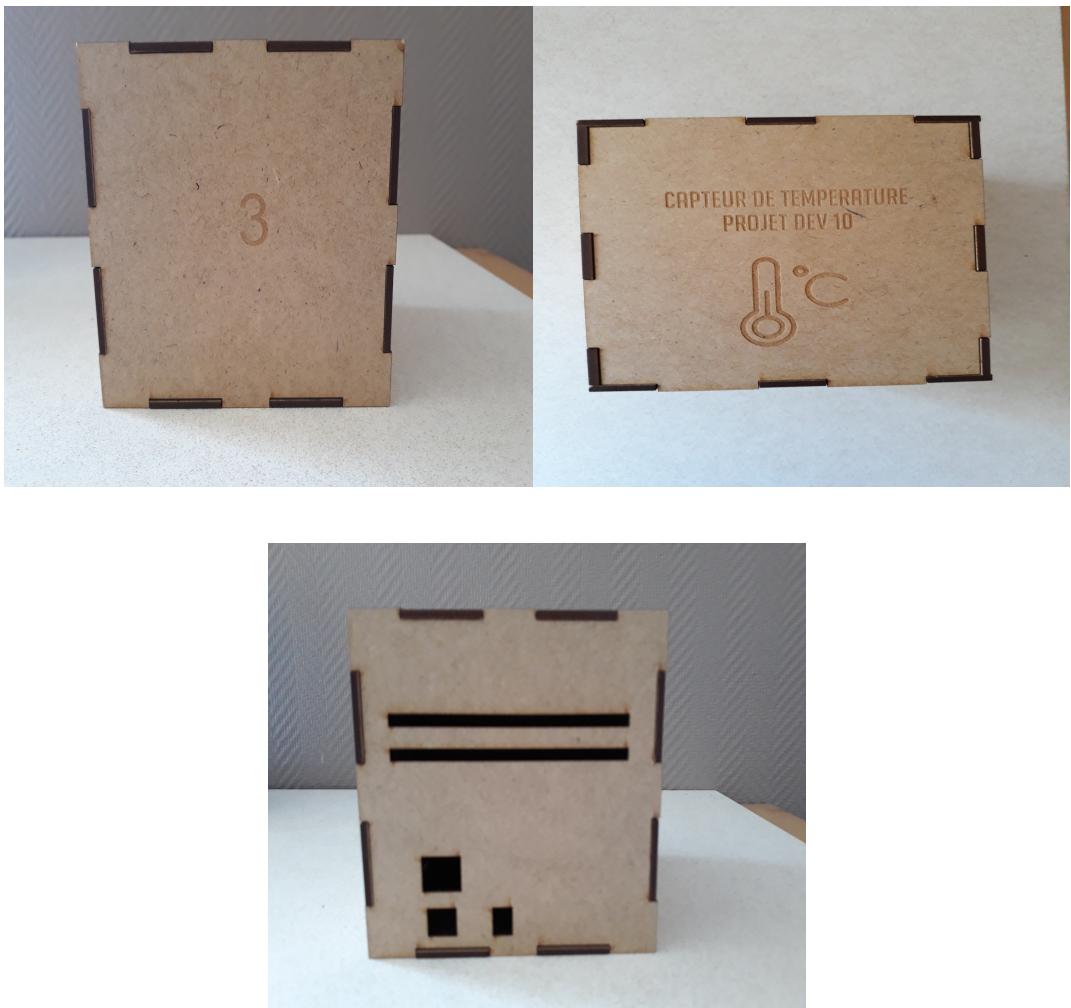


Figure 16 :Boîtier réalisé vu de 3 faces

Alimentation de la carte Arduino:

L'alimentation de la carte Arduino est, bien sûr, nécessaire pour le fonctionnement du système. Pour permettre une autonomie des boîtiers, comme déclaré dans le cahier de charges fonctionnel, un système d'alimentation par piles a été adopté. Cela permet aussi aux capteurs d'être déplaçables sans coupure de leur fonctionnement.

Cependant, une fois les piles déchargés, il va falloir maintenir le fonctionnement des capteurs sans attendre leur recharge. Nous avons alors adopté au même temps une alimentation sur secteur de la carte Arduino. En effet, les deux types d'alimentation seront disponibles et

utilisés : Le capteur fonctionnera avec les piles jusqu'à l'approche de leur déchargement, à ce moment, la carte Arduino doit être branché sur secteur en attendant le changement de piles (ou leur chargement).

Pour réaliser un tel système, la solution choisie est de concevoir le circuit suivant :

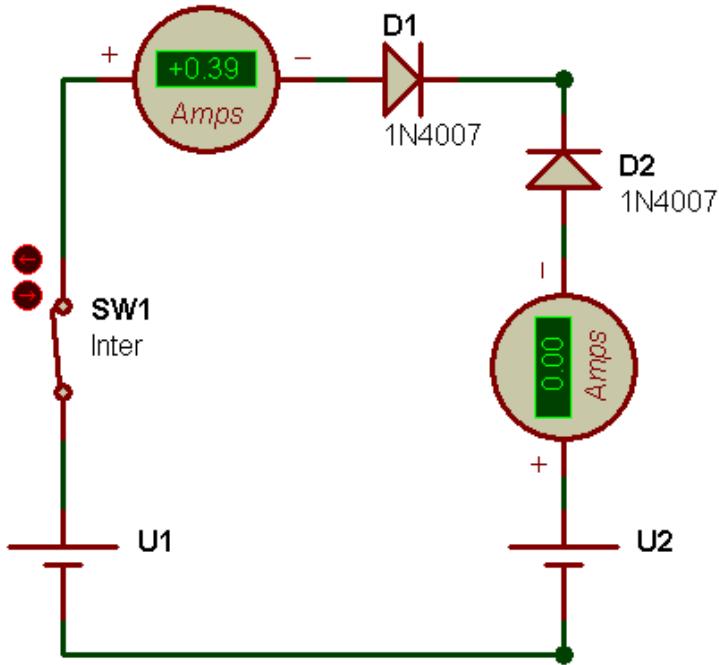


Figure 17 :Schéma du circuit d'alimentation

Son fonctionnement est comme suit : Chaque diode est passante lorsque son anode est portée à une tension plus élevée que son cathode. Sinon elle est bloquante.

En mettant alors une tension d'alimentation sur secteur plus élevée que celle des piles (Différence peut être estimée à 1V), D1 est passante et D2 est bloquante tant que l'alimentation sur secteur est activé, et on a un chargement de piles en parallèle, quand on l'enlève, c'est l'inverse [7].

Rédacteur : BENGHABRIT Ilyas

Relecteur : HAMDI Mohamed

Conclusion :

Pour conclure, la mise en place d'un réseau de capteurs de température permettant la visualisation des données de température par la Maisel a été un projet enthousiasmant qui a nécessité un travail assez varié sur différents domaines.

Nous avons donc pu réaliser un réseau de capteurs permettant d'envoyer les données de température en temps réel au site web, les afficher et archiver. La mesure de température a pu être faite après implémentation du module thermique sur la carte Arduino et téléversement du code approprié. La réalisation du site Web a été faite ensuite de manière à ce qu'il soit ergonomique à utiliser par l'administration de la Maisel. Par la suite, les données de température ont pu être envoyées au site ensuite par le biais d'un protocole d'envoi.

Pour protéger les capteurs, des boîtiers pratiques ont été réalisés par imprimante laser. Et finalement, un système d'alimentation secourue par piles a été adopté pour permettre une mobilité aux capteurs pendant leur utilisation.

Ainsi, la réalisation d'un tel réseau de capteurs va s'avérer certainement très utile. Que ce soit pour la Maisel ou pour n'importe quel autre établissement désirant évaluer l'efficacité de leurs systèmes de chauffage ou de refroidissement, une production massive de ces capteurs permettra d'avoir un suivi de température en temps réel et ainsi d'atteindre son objectif.

Rédacteur : BENGHABRIT Ilyas

Relecteur : HAMDI Mohamed

Références bibliographiques:

- [1] DJANGO, *Documentation* [en ligne], disponible sur le lien : <<https://docs.djangoproject.com/en/2.0/>> (consulté le 12.03.2018).
- [2] DJANGO, *Distribution des Urls* [en ligne], disponible sur le lien : <<https://docs.djangoproject.com/fr/2.0/topics/http/urls/>> (consulté le 15.04.2018).
- [3] W3SCHOOLS, *W3.CSS Tutorial* [en ligne], disponible sur le lien : <<https://www.w3schools.com/w3css/default.asp>> (consulté le 2 mai)
- [4] AMCHARTS. *JavaScript Charts and Maps* [en ligne], disponible sur le lien: <<https://www.amcharts.com/>> (consulté le 02.05.2018)
- [5] Arduino, *Ethernet / Ethernet 2 library* [en ligne], disponible sur le lien : <<https://www.arduino.cc/en/Reference/Ethernet>> (consulté le 30.04.2018).
- [6] CorelDRAW. *Welcome to CorelDRAW Help* [en ligne]. Disponible sur: <<http://product.corel.com/help/CorelDRAW/540229932/Main/EN/Documentation/wwhelp/wwhimpl/js/html/wwhelp.htm#href=CorelDRAW>Welcome-to-CorelDRAW-Help.html>> (consulté le 15.03.2018).
- [7] Sonelec-Musique. *Alimentation secourue par batterie 001* [en ligne]. Disponible sur : <http://www.sonelec-musique.com/electronique_realisations_alim_secours_bat_001.html> (consulté le 25.05.2018)

Glossaire:

-CSS : Cascading Style Sheets
-CSV : Comma-separated values
-CSRF PROTECTION : Cross-site request forgery, type de vulnérabilité des services d'authentification web.
DHT22 : Sonde qui permet de mesurer la température.
HTTP : Hypertext Transfer Protocol
HTTPS : Hypertext Transfer Protocol Secure
GET : Requête Http pour demander une ressource d'un serveur web
POST : Requête Http pour modifier une ressource d'un serveur web
X-XSS : Filtre désactivant les attaques de type Cross-site scripting

Annexe:



Analyse du planning:

Planning prévisionnel :

GANTT project			
	Nom	Date de début	Date de fin
▲	• Phase Prototype	12/02/18	14/05/18
♀	• Phase Capteur	12/02/18	27/03/18
♀	• Phase Arduino	27/02/18	07/03/18
•	• Implémentation module thermique	27/02/18	28/02/18
•	• Implémentation module Wifi	06/03/18	06/03/18
•	• Envoi des données via Wifi	07/03/18	07/03/18
•	• Données envoyées	07/03/18	07/03/18
♀	• Phase alimentation	20/03/18	20/03/18
•	• Alimentation par pile	20/03/18	20/03/18
•	• Alimentation sur secteur	20/03/18	20/03/18
•	• Adruino alimenté	20/03/18	20/03/18
♀	• Phase Boîtier	12/02/18	27/03/18
•	• Faire le schème d'un boîtier	05/03/18	19/03/18
•	• Réaliser un boîtier	20/03/18	27/03/18
•	• Boîtier prêt	12/02/18	12/02/18
♀	• Phase Serveur Web	27/02/18	23/04/18
•	• Crédit du site	27/02/18	27/02/18
•	• Hébergement du serveur par le Resel	06/03/18	19/03/18
•	• Crédit d'une base de donnée	20/03/18	02/04/18
•	• Conception d'une interface Web	03/04/18	09/04/18
•	• Réception des données	10/04/18	16/04/18
•	• Archivage des données	10/04/18	16/04/18
•	• Affichage des données	17/04/18	23/04/18
•	• Données affichées	23/04/18	23/04/18
♀	• Phase liaison	27/03/18	14/05/18
•	• Protocole d'envoi arduino - serveur	27/03/18	16/04/18
•	• Envoi des données de température via Wifi	24/04/18	14/05/18
•	• Réception des données de température	24/04/18	14/05/18
•	• Données échangés entre Arduino et Serveur	14/05/18	14/05/18
♀	• Phase Production massive	15/05/18	29/05/18
•	• Crédit de 5 produits	15/05/18	29/05/18

Planning effectif:

- Implémentation module thermique : 27 février
- Premier prototype de boîtier : 05 mars
- Documentation sur la conception de serveur web : 12 mars
- Réalisation du serveur web sous Django configuré sans Design : 24 avril
- Nouveau prototype pour les boîtier : 24 avril
- Implémentation module Ethernet : 30 avril
- Envoi de données de températures au serveur web : 4 mai
- Envoie de données Chiffrées : 5 mai

- Version Site Web finalisée : 17 mai
- Hébergement du site Web : 22 mai
- Test de l'envoie de données : du 28 mai au 05 juin
- Conception du circuit d'alimentation : 27 mai
- Réalisation du circuit d'alimentation : 4 juin
- Production massive des capteurs : 4 juin

Tout au long de notre projet nous nous sommes trouvés dans l'obligation d'ajuster notre planning à plusieurs reprises à cause de différentes contraintes. Le fait qu'un de nos collègues ait abandonné le projet nous a obligé à répartir les tâches à nouveau entre nous.

Nous avons réalisé le module de température et nous avons conçu le premier prototype du boîtier en accord avec le planning initial. La conception du serveur sous Django s'est avérée plus longue que prévue puisque nous avons dû tout d'abord acquérir les compétences nécessaires. De plus, l'indisponibilité des modules Wifi au fablab nous a empêché de progresser dans la phase d'envoi de données et nous avons dû nous orienter vers une autre démarche qui consiste à envoyer les données via ethernet.

Pour conclure, en dépit des obstacles rencontrés qui ont engendré plusieurs nuances entre le planning initial et le planning effectif, nous avons pu aboutir au résultat attendu de notre projet.