

Introduction

But du stage

Le but du stage est d'effectuer un projet qui consiste à concevoir un dispositif afin de récupérer les données de température et d'humidité de la bibliothèque de la fac de médecine au campus Carreire car celle-ci n'est équipée que de capteurs mesurant en temps réel. Les données doivent être accessible par internet au responsable des lieux sous une forme compréhensible. Celui-ci doit également être prévenu en cas de données alarmantes.

Les contraintes sont :

- ne pas utiliser le réseau WiFi de la fac car celui-ci est déjà très sollicité
- le dispositif doit être autonome pendant au moins 1 mois (l'idéal étant qu'il soit autonome pendant toute la durée de fermeture de la bibliothèque l'été c'est-à-dire 2 mois) et facile d'entretien
- les données doivent être stockées sur un serveur distant (20 rue Tourat aux Chartrons), une copie de ces données sera conservée sur le serveur du FABLAB.

Si possible, il faut également mesurer les pourcentages de charge des batteries afin de savoir quand les recharger.

Un impératif est que le projet soit reproductible, notamment grâce à une bonne documentation et l'utilisation de logiciels et solutions open source voire libres. Le projet en lui-même est libre.

Enjeux

Si le projet réussit cela pourrait intéresser d'autres sites recherchant une solution accessible et libre. D'autre part le FABLAB gagnerait en réputation et crédibilité pour d'éventuels futurs prestations ou projets.

La bibliothèque

La bibliothèque en question est particulière puisqu'elle stocke des livres et documents anciens et très précieux. Ainsi le dispositif devra être sûr afin de ne pas endommager les documents. Le principal risque direct est lié à l'utilisation des batteries que nous utilisons du fait du faible nombre de prises. Un risque indirect est les erreurs de mesures qui empêcheraient de prendre conscience d'un éventuel problème. Nous prendrons en compte ces risques lors de la conception du dispositif. Un des responsables de la bibliothèque nous a indiqué les seuils à respecter :

	Température	Taux d'humidité
Seuil en valeur absolue	$18^{\circ}\text{C} < T < 20^{\circ}\text{C}$	$45\% < H < 55\%$
Seuil de variation	5°C en 1 semaine	5 % en 1 semaine

Il nous a précisé qu'en valeur absolu les seuils sont indicatifs, c'est-à-dire qu'on peut tolérer un léger dépassement à la condition que la variation soit lente. Autrement dit les seuils de variation sont cruciaux pour la conservation par rapport aux seuils absolus.

Le FABLAB

Le FABLAB de Talence est une association gérée au quotidien par Jean-Baptiste Bonnemaïson et Pierre Grangé-Pradéras qui sont aussi mes tuteurs de stage. De par la nature du lieu, la mentalité est orientée vers le partage des connaissances, le « faire soi-même » et la documentation ouverte afin de permettre la reproductibilité et l'ouverture des projets. Cette vision fait que l'évocation de logiciels ou solutions propriétaires provoque des sourires et une discussion instructive et agréable

sur les problèmes qu'ils induisent et les alternatives ouvertes et libres. Ainsi nous avons privilégié l'utilisation de logiciels et de documents open source voire libre. Ceux-ci offrent plusieurs avantages (liste non exhaustive) :

- la possibilité d'accéder aux sources des codes et documents
- la stabilité dans le temps puisque le développement est assuré de manière décentralisée et bénévole par des passionnés et non par une seule organisation privée et fermée
- la gratuité d'un grand nombre d'entre eux

Par ailleurs tous les PC présents au FABLAB tournent sous Debian, une fameuse distribution Linux. L'association dispose d'un grand nombre de PC, d'outils électroniques de base (oscilloscope, poste à souder, etc, ...), de matériels (divers microcontrôleurs, câbles, capteurs, etc, ...) et de machines (fraiseuse de PCB, etc, ...). Elle réunit des adhérents et d'autres stagiaires qui nous ont aidé en nous donnant des conseils pratiques. Le FABLAB possède un site web sur lequel sont recensés et organisés les projets et annonces. Celui-ci nous sert beaucoup puisque nous y avons un compte avec une page dédiée pour notre journal de stage et une page pour le projet. Nous avons été très tôt sensibilisés à la documentation puisque le projet est libre et il est impératif que le projet soit reproductible à partir de celle-ci. Ainsi nous postons régulièrement nos avancées sur le site ce qui nous force à faire de la documentation (ce qui est très formateur).

État du projet à notre arrivé

Avant le début du stage nos tuteurs ont déjà déterminé le matériel que l'on va utiliser ainsi qu'une grande partie du cahier des charges.

Matériel de départ

- **module ESP32 avec support de batterie Li-ion 18650**, ([lien du revendeur](#)), [datasheet technique ESP32](#) (source Espressif)

Environ 5 euros

Ces microcontrôleurs sont très polyvalents et abordables. Ils peuvent se connecter par WiFi et possèdent une fonctionnalité de « deep sleep » permettant de réduire considérablement la consommation électrique. De plus, du fait de sa popularité il existe de nombreuses documentations et aides sur internet.

- **module capteur de température et d'humidité DHT22**

[datasheet technique DHT22](#) (source www.sparkfun.com)

2-3 euros/unité

Capteur abordable dont les caractéristiques sont suffisantes pour notre utilisation (données constructeur) :

	Taux d'humidité	Température
Précision	<5%	<0,2°C
Résolution	0,1 %	0,1°C

De plus il possède des bibliothèques d'utilisation libres et peut être alimenté directement par l'ESP32. La précision du taux d'humidité sera améliorée en calculant la moyenne des mesures de plusieurs capteurs.

- **batterie Li-ion NCR18650b – Li-ion MH12210**

[datasheet](#) (source www.imrbatteries.com)

40 euros pour 20 batteries

Capacité	3350 mAh
Tension nominale	3,6V

Étant donné que nous serons dans une bibliothèque nous devons mettre le support de pile dans un boîtier en métal pour minimiser les risques en cas de combustion des batteries.

- **Présent au FABLAB**

câbles Dupont, multimètre et autres matériels et outils de base et machine fraiseuse de PCB.

Début du travail et élaboration d'un système

Les solutions de communication entre la bibliothèque et le serveur distant

Dans un premier temps, étant donné que nous ne pouvons pas utiliser le réseau WiFi de la fac, nous recherchons des solutions pour transmettre les données au serveur distant.

Sigfox : réseau disponible où nous sommes et abonnement abordable (3 à 9 euros/an), module de connexion 5 euros mais limité à 140 messages de 12 octets par jour, reproductibilité non assurée car ce réseau n'est pas accessible partout => **solution non retenue**

Lora : réseau disponible où nous sommes et abonnement abordable (36 euros pour 3 ans par ex) , module de connexion 12 euros, possibilité de créer notre propre réseau mais reproductibilité non assurée car ce réseau n'est pas accessible partout, incertitude quant à la couverture réseau proposé => **solution non retenue**

GSM : réseau fiable et largement disponible, module de connexion < 5 euros mais abonnement plus cher (autour de 5 euros/mois) => **solution retenue** pour sa fiabilité et sa reproductibilité

Transmission des données au serveur distant

Pour transmettre les informations récoltées par les ESP32 à notre serveur nous utilisons le protocole MQTT. Il est composé de 3 acteurs : le publisher, le broker et le subscriber. Le publisher envoie (publish) des données au broker, le broker les stocke dans ce qu'on appelle un topic et les distribue ensuite aux subscribers qui sont « abonnés » (subscribe) au topic. L'avantage de ce protocole est que la publication des données par le publisher et la distribution aux subscribers est asynchrone, c'est-à-dire que le publisher peut envoyer des données au broker puis se déconnecter du réseau. C'est le broker qui se chargera de l'envoi des données aux subscribers lorsqu'ils se connecteront au broker. En outre, ce protocole permet de faire en sorte que le publisher stocke les données si jamais il n'arrive pas à se connecter au broker. De même le broker peut stocker les données tant qu'il ne les a pas distribués aux subscribers. Cela rend ce protocole très robuste aux éventuels problèmes de réseau et est utilisé dans l'IOT (Internet Of Things).

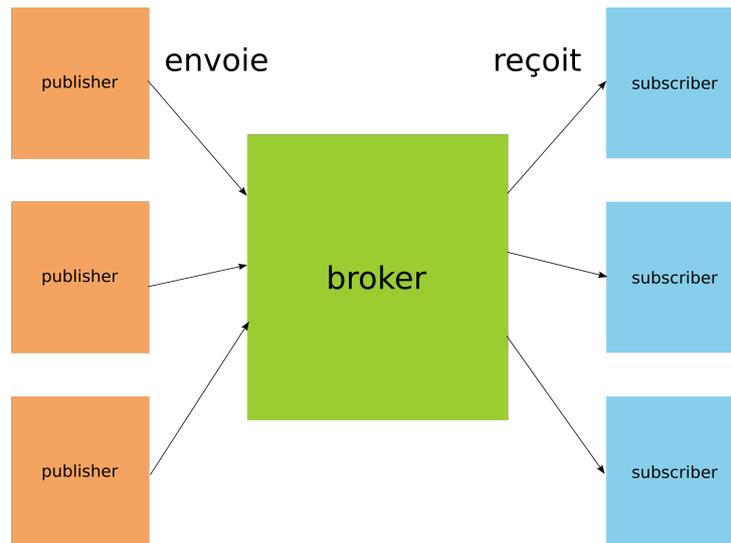


Figure 1: protocole MQTT

Dans notre cas :

- publisher ⇒ ESP32
- broker ⇒ serveur distant
- subscriber ⇒ serveur distant

Cela ne pose pas de problème d'avoir le broker et le subscriber sur la même machine puisque ce seront deux programmes distincts, le programme du broker enverra ses données localement au programme du subscriber.

Architecture du système envisagé

Après donc cette phase de découverte et recherche de solutions, nous sommes en mesure de proposer un système :

Les ESP32 récolteront les données de température et d'humidité via les DHT22. Ils se connecteront à internet en WiFi par l'intermédiaire d'un hotspot (microcontrôleur équipé d'un module GSM) pour accéder à internet. Les données seront envoyées sur un serveur distant en utilisant le protocole MQTT et rangées dans une base de données InfluxDB (gestionnaire de base de données spécialisée dans les données temporelles, licence MIT). Une copie de cette base sera conservée par le FABLAB. L'affichage de ces données sera fait sur un site web avec Grafana (logiciel open source permettant entre autres d'exploiter des bases de données temporelles pour l'affichage et de gérer des alertes). L'accès au site et à la base sera protégé par un mot de passe. La gestion des alertes en cas de données alarmantes est également gérée par Grafana par mail.

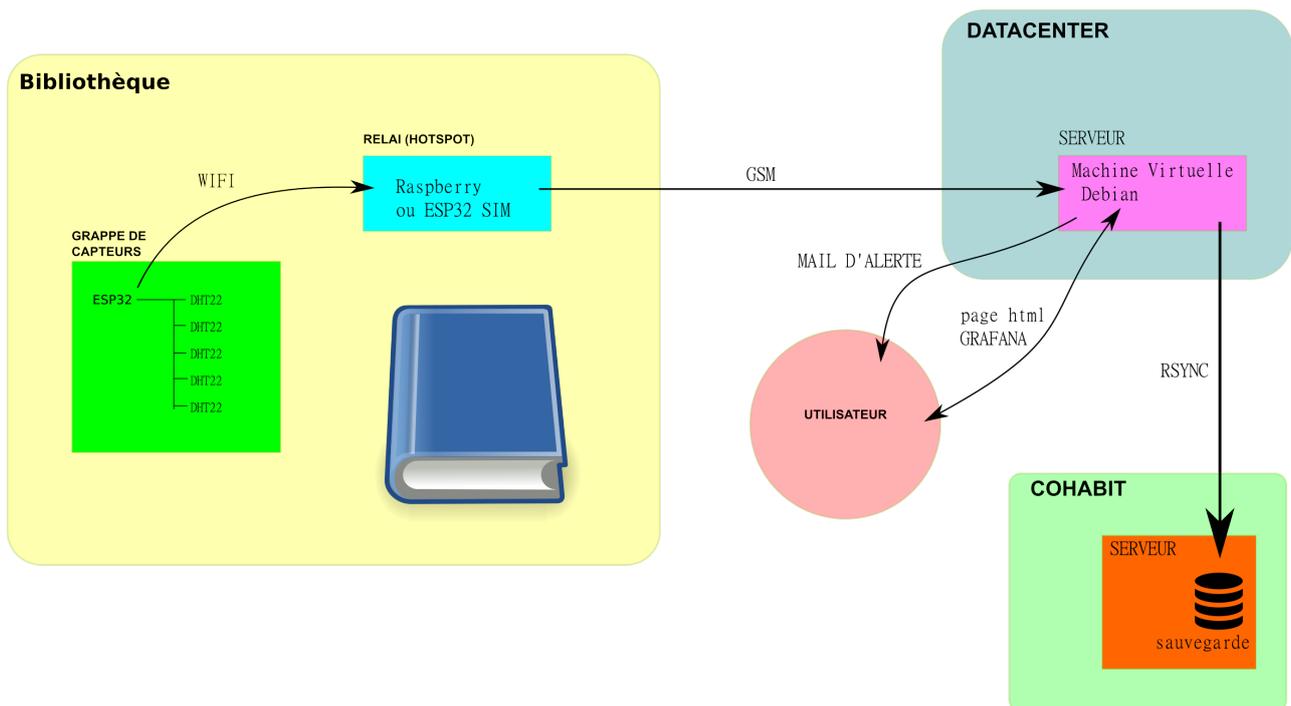


Figure 2: schéma général du système envisagé

Nous utiliserons un hotspot pour éviter de devoir équiper chaque ESP32 d'un module GSM. Nous gardons une copie de la base de données grâce à RSYNC, un logiciel libre de synchronisation de fichiers.

Mise au point du système

Découverte du matériel

Nous avons rapidement soudé les pins des ESP32 à notre disposition et ensuite installé l'environnement de développement. Nous avons mis un certain temps avant d'avoir notre setup opérationnel. Nous avons installé un premier environnement qui fonctionnait mais il ne nous convenait pas car il était moins pratique à cause de l'outil de compilation et de téléversement en CLI (ligne de commande) avec de nombreuses commandes non évidentes et un donc un long temps de développement. Sur les conseils de notre tuteur nous sommes passés à Visual Code Studio (open source) avec l'extension PlatformIO (libre sous licence MIT) qui permet de compiler et téléverser le code sur notre microcontrôleur par USB en un seul clic. De plus cette extension permet d'installer les bibliothèques sans erreurs et très rapidement. Nous avons réalisé une documentation décrivant comment installer et utiliser notre setup.

Le code de l'ESP32 est en C++ ce qui a nécessité des recherches sur les classes et leur fonctionnements. Étant donné que la plupart des tutoriels et bibliothèques intéressants sont faits pour l'arduino nous avons inclus dans nos codes une librairie arduino.h qui permet d'importer les commandes arduino (les mêmes fonctions, la partie void setup et void loop) mais en C++ toujours. Avec ce setup installé nous nous sommes familiarisé avec l'ESP32 et le DHT22 pour lire la température et l'humidité. Pour cela on installe les librairies Adafruit Unified Sensor (par Adafruit) et DHT sensor library (par Adafruit).

Conception d'une « grappe » de capteurs

Nous concevons un prototype de « grappe » de DHT22 qui mesurera la température et le taux d'humidité. Nous créons une « grappe » pour calculer la moyenne des taux d'humidité mesurés et ainsi réduire les erreurs de mesure. Celle-ci est composée de cinq DHT22 connectés à un ESP32. Nous collons les DHT22 de bas en haut d'un mur pour se rendre compte des variations de la température et du taux d'humidité selon la hauteur (hauteur = 1,80m).



Figure 3: prototype de grappe

Nous obtenons un premier jeu de données dont nous traçons les graphes :

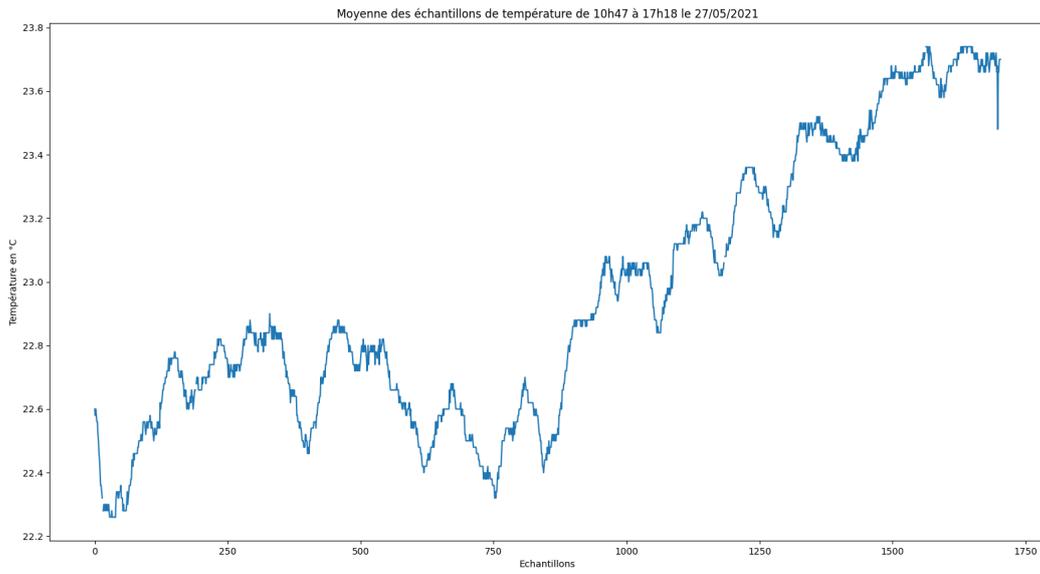


Figure 4: premier jeu de données, température

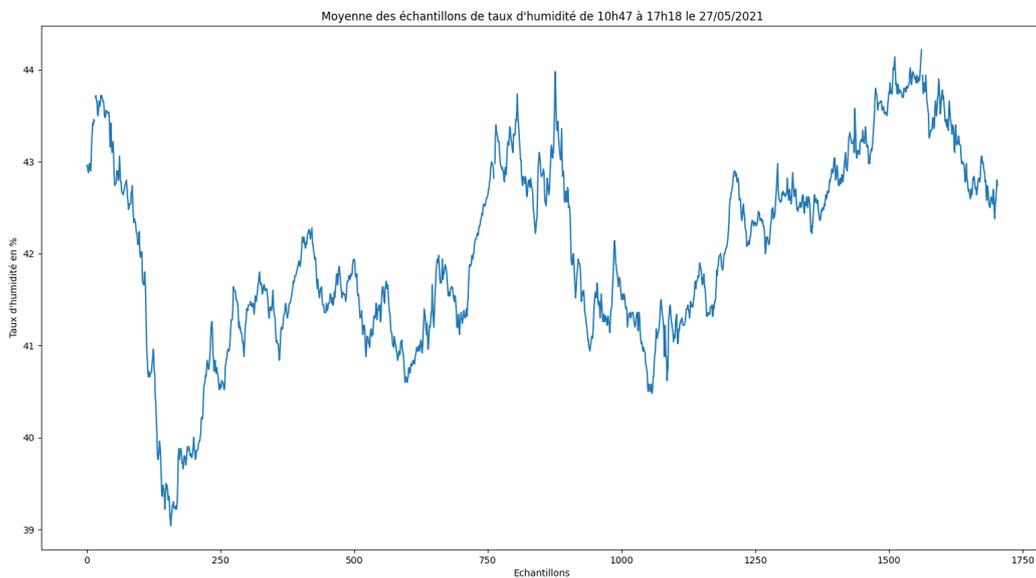


Figure 5: premier jeu de données, taux d'humidité

Remarque sur les conditions de mesures

- deux fenêtres sont ouvertes non loin de la grappe
- nous passons souvent à côté pour accéder à l'ordinateur qui l'alimente

Commentaires sur les essais

Je précise que nous n'avons pas été délicat avec les capteurs (chutes, fréquentes manipulations avec les mains) ce qui peut influencer négativement sur la précision.

Pour la température :

Les mesures de températures restent stables malgré les conditions de mesures évoquées précédemment. L'écart maximum entre les mesures des cinq capteurs est de 1,5°C. Nous avons par exemple (gauche à droite <==> bas en haut sur le mur) :

Temp: 21.60 23.10 22.30 23.10 22.80

La mesure du 2^e capteur est clairement anormale. Si l'on considère les autres capteurs, leurs mesures semblent valider ce qu'on sait théoriquement : l'air est plus chaud avec la hauteur.

Pour le taux d'humidité :

Les conditions de mesures peuvent expliquer les variations d'humidité soudaine, les capteurs étant très sensible pour la mesure du taux d'humidité. Par exemple nous avons une fois soufflé sur le capteur pour observer les variations du taux d'humidité : la mesure est montée à 70-80 % et est resté plusieurs minutes à cette valeur avant de redescendre au même niveau que les autres capteurs. Le même phénomène se produit de manière moins intense lorsqu'on approche la main.

Conclusion

Les résultats sont satisfaisants au vu des conditions dans lesquels le test a été réalisé. Sachant que la bibliothèque est isolé en température et en humidité, les capteurs devraient remplir leur fonction correctement. Cela sera confirmé lors d'un essai sur site. Nous notons qu'il ne faut pas approcher de trop près ni manipuler les capteurs pour ne pas fausser les mesures notamment du taux d'humidité. Par conséquent il faudra réfléchir à la localisation de la grappe dans la bibliothèque. Une question est de savoir le nombre de capteurs dont nous avons besoin pour garder des mesures suffisamment satisfaisantes. Cependant le mieux étant l'ennemi du bien, nous en restons ici pour l'instant.