

# Boite Intelligente de Traitement d'Environnement - Cahier des charges

## Contexte :

Dans le cadre des cours d'informatique du 2e trimestre nous avons eu pour projet de réaliser un objet connecté permettant d'évaluer la « qualité » d'une salle. Pour cela nous avons conçu et réalisé une boîte de mesures connectée. La boîte intelligente de traitement d'environnement est un objet connecté qui nous permet de mesurer différents paramètres comme : la qualité de l'air, la température, l'hygrométrie, le niveau sonore ou encore la luminosité. En récupérant ces données nous pouvons ainsi déterminer s'il fait bon vivre dans une salle (ou non). L'objectif principal est de centraliser ces informations pour un bâtiment et d'accéder à un panneau de contrôle via un navigateur web. Pour cela, notre boîte est connectée à un serveur qui s'occupe de centraliser les informations des différentes boîtes installées dans chaque pièce.

## Éléments mesurés :

- ✓ La Température (°C)
- ✓ L'humidité (%)
- ✓ La luminosité (lux)
- ✓ Le niveau sonore (dB)
- ✓ La pression barométrique (hP)
- ✓ La qualité de l'air (capteur de CO<sub>2</sub>, en ppm)

## Conception de la boîte :

- L'objet connecté est réalisé à partir de différents capteurs et d'un ESP32 et le serveur à partir d'un Raspberry Pi, les deux communiquent en MQTT via un serveur hébergé sur la Raspberry et disponible sur le réseau wifi.
- L'alimentation de l'esp et des capteurs est assurée par une batterie rechargeable.
- Tous les composants sont dans la boîte qui est déplaçable et fonctionne sans fil.

## Contenu de l'objet :

- ESP 32
- Batterie 3P1S de 3,7V et 9000mAh + Module de chargement de batterie
- Chargeur secteur micro USB
- Switch ON/OFF
- Notice
- Outils de mesure :

Utilité	Module
Température	DHT11
Humidité	DHT11
Luminosité	TSL2561
Bruit	MAX4466
Pression	BMP180
CO2	MQ135

## Boîte

La coque de l'objet est carrée, en deux parties et de dimension 10cmX10cmX19cm. La partie haute est en bois et la partie basse en PLA (impression 3D). Entre les deux parties, il y a un espace permettant un accès pour les capteurs à l'air ambiant afin d'assurer les relevés dans les meilleures conditions.



## Fonctionnement :

Lors de la première connexion, l'ESP32 charge les données enregistrées dans sa mémoire et essaie de se connecter au WiFi. Les données implémentées par défaut étant génériques, la connexion est fortement susceptible d'échouer.

Si la connexion au réseau WiFi échoue, l'ESP32 se présente en point d'accès et active alors un « captive portal ». Le captive portal correspond à un serveur web hébergé sur la carte, et qui est présenté à tout client se connectant au réseau. Comme la carte n'a pas été capable de se connecter au réseau, elle demande donc à l'utilisateur, par le biais du captive portal, de remplir les informations de connexion.

Les informations attendues sont le nom du réseau voulu (choisi dans le menu déroulant) et le mod de passe d'accès. Une fois le formulaire de connexion de la page du captive portal rempli, l'utilisateur peut valider la saisie. La validation de saisie entraîne une requête à l'ESP32 et les données sont enregistrées sur la carte.

Une fois les informations enregistrées, l'ESP32 redémarre et retente une connexion au réseau WiFi. Si la connexion échoue de nouveau, le processus se répète. Sinon, il essaie de se connecter au server Raspberry.

Les informations par défaut étant génériques, cette connexion échouera probablement, ce qui relancera le processus d'ouverture de captive portal, mais cette fois pour récupérer l'IP et le port du Raspberry (ou du serveur MQTT plus généralement).

Une fois l'initialisation terminée (connexion au WiFi et serveur MQTT réussies), l'ESP32 publie un message contenant son adresse mac au serveur, afin de permettre la création d'un nouvel objet dans la base de données, et d'être sélectionnable sur le tableau de bord.

S'ensuit alors la fonction de relevé, qui s'active à intervalle défini et envoie régulièrement les informations et états des différents capteurs au serveur. La Raspberry qui héberge le serveur va alors actualiser la base de données et le Dashboard.

Il est possible de connecter plusieurs Boîtes Intelligentes de Traitement d'Environnement au même serveur, il y aura alors plusieurs boîtes sur le Dashboard. Si un capteur ne fonctionne plus, un signal d'erreur peut être rapporté sur le Dashboard.

Le Dashboard est dynamique est intuitivement modulable. Vous pouvez créer, modifier et supprimer vos salles à volonté. Vos salles seront chacune reliées aux relevés d'une ou plusieurs boîtes. Chaque salle est reliée à un bâtiment. Sélectionner une salle sur le tableau de bord permet d'afficher des informations générales du système, l'intervalle de relevé des différentes boîtes et les relevés numériques, ainsi que leurs historiques, sous forme de graphiques.

## Evolution du cahier des charges :

Nous avons initialement prévu un écran sur l'objet connecté pour afficher l'état de marche des composants. Nous avons finalement jugé que cette fonctionnalité superflue et trop gourmande en énergie, c'est pourquoi nous avons abandonné cette idée.

Nous nous posons la question d'intégrer un signal audio pour prévenir d'une hausse anormale de certains relevés comme le CO2. Nous avons décidé d'abandonner l'idée car cela va plus ennuyer l'utilisateur que lui rendre service.

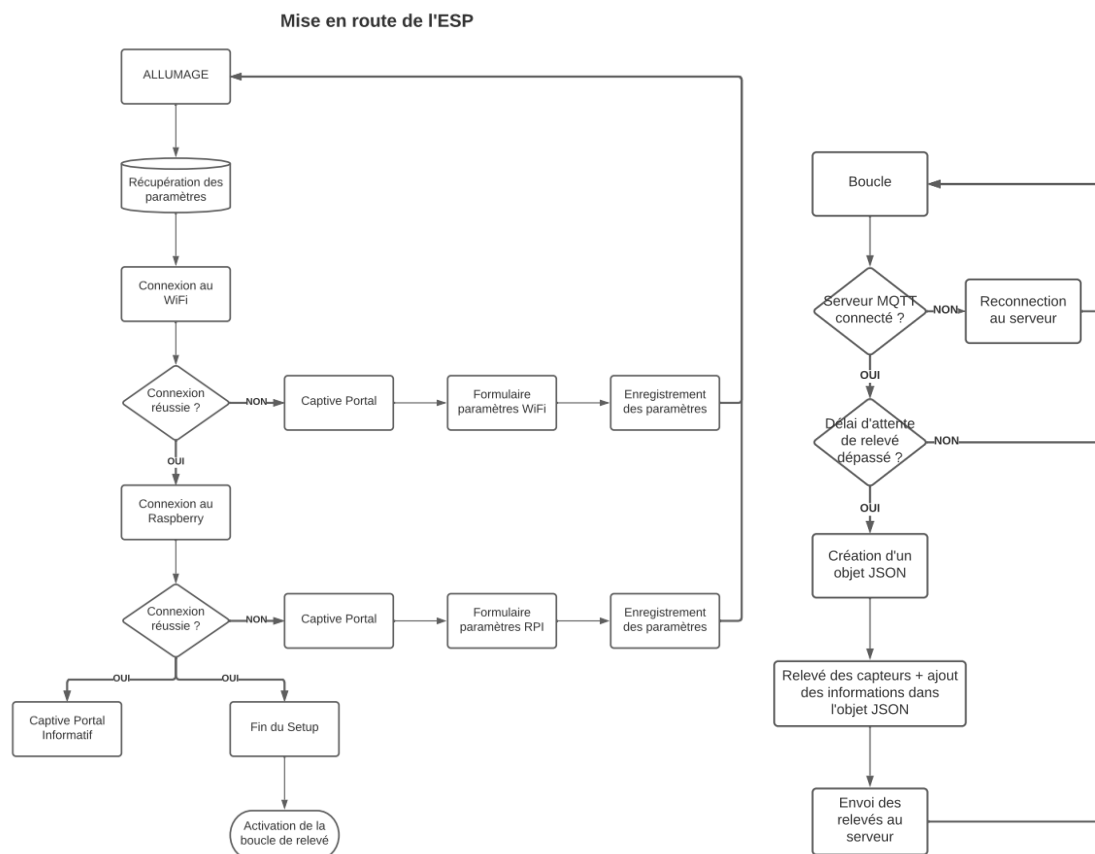
Nous voulions également étudier le fait d'intégrer un panneau photovoltaïque pour garantir l'autonomie de la boîte. Or, il aurait fallu un panneau bien plus grand que la face supérieure de la boîte, rendant celle-ci plus difficilement déplaçable et ne concorderait pas avec le design épuré que l'on a voulu lui donner.

Dans une démarche écologique, le châssis de la boîte a été réalisé en grande partie à base de bois massif de récupération, avec une impression 3D à son centre (cf section « Boîte »). Le design de notre boîte est simple, épuré et s'intègre facilement dans la décoration d'une maison, et est mis en avant par l'impression 3D qui contraste et fait ressortir le bois clair.

## Partie Technique :

### Le Code ESP32 :

Logigramme -



### Outils utilisés :

Librairies Arduino :

- PAHO
- Pub sub client

- BMP085
- MQ135
- DHT11
- TSL2561
- MAX4466

Base de données :

- Django rest framework
- Mariadb

Dashboard :

- React
- SASS