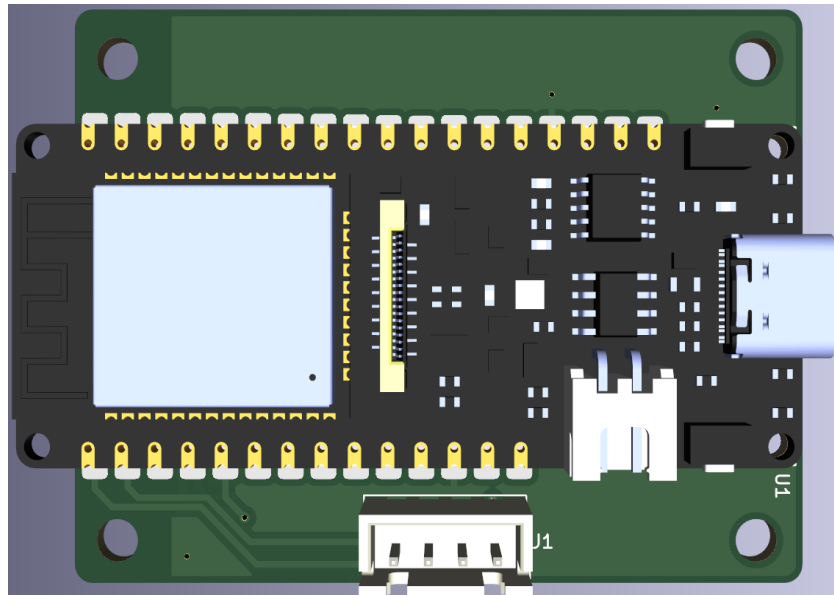


Cuve connectée



Projet réalisé par
Gaëtan Corabœuf, Pierre Alexandre Herve

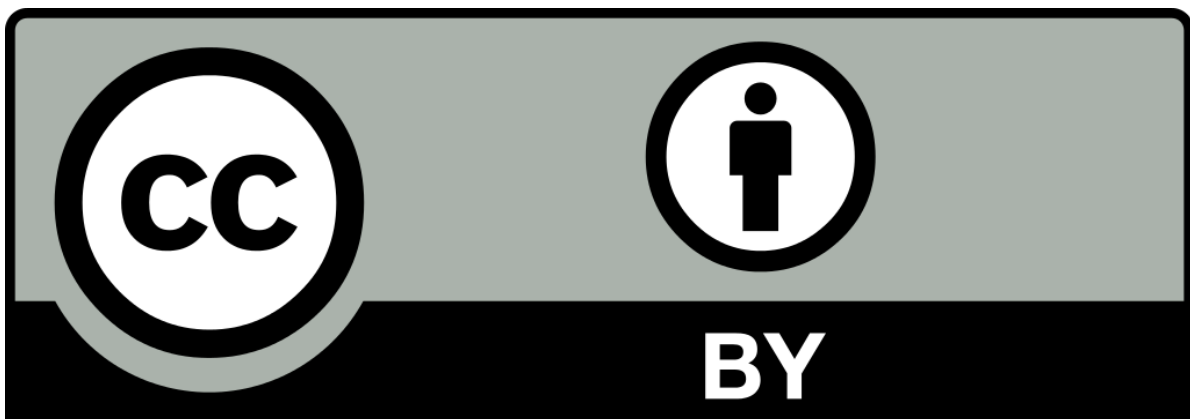
Projet encadré par
Philippe Lucidarme

IUT Angers-Cholet
4 boulevard Lavoisier - BP 42018
49016 - Angers Cedex

Année 2023-2024

Ce rapport présente le travail réalisé par un groupe d'étudiants dans le cadre d'un projet pédagogique. Les auteurs et l'Université d'Angers ne garantissent pas que l'information, les documents, la méthodologie et le matériel présentés dans ce document soient complets, conformes à l'état de l'art et exacts ni n'assurent en toutes circonstances la sécurité des biens, des personnes et des utilisateurs. Les auteurs et l'Université d'Angers ne seront pas tenus responsables des dommages éventuels qui pourraient résulter de l'utilisation du contenu du présent rapport.

Gaetan Coraboeuf, Pierre-Alexandre Herve auteurs du présent rapport, publions et divulguons celui-ci sous la Licence Creative Commons suivante « CC BY » pour le monde entier et pendant la durée légale de protection des droits d'auteur. Cette licence autorise la représentation, la reproduction, la modification, la création d'œuvres dérivées et l'utilisation y compris à des fins commerciales sous réserve de mentionner les noms et prénoms des auteurs.



Remerciements

Tout d'abord, nous souhaitons remercier l'ensemble des personnes citées ci-dessous pour leur contribution à ce projet :

- ★ M. Philippe Lucidarme, enseignant-chercheur à l'Université d'Angers, pour le temps qu'il nous a consacré, les connaissances qu'il nous a apporté et sa bienveillance
- ★ L'IUT d'Angers-Cholet et plus particulièrement le département GEII pour le matériel et les lieux mis à disposition ainsi que l'achat des composants nécessaires au projet
- ★ Mme. Denécheau pour la vérification et la commande des PCB
- ★ Nous remercions notre prestataire "Banana Helmet Production" pour la réalisation de la CAO 3D
- ★ Nous remercions aussi Ugo Maiurano pour l'impression 3D de notre mécanique.

Enfin, nous remercions de manière générale notre groupe d'alternant ESE 2023-2024. Pour l'entraide et la cohésion d'équipe pour à la réalisation de ce projet de SAE.

Sommaire

<u>I. Le cahier des charges.....</u>	<u>5</u>
A. Les spécifications techniques.....	5
B. Les livrables.....	6
<u>II. Les choix technologiques.....</u>	<u>7</u>
A. Les choix hardwares.....	7
B. Les choix logiciels.....	9
C. Firebase.....	10
D. PlatformIO.....	11
<u>III. La réalisation.....</u>	<u>12</u>
A. Le schéma électronique.....	12
B. Le Routage.....	16
C. Le logiciel embarqué.....	16
D. Réalisation de l'application web.....	19
E. La mécanique.....	20
<u>IV. La conclusion et les perspectives d'amélioration.....</u>	<u>22</u>
A. Ce que l'on retient de ce projet.....	22
B. Les perspectives d'amélioration.....	22

Introduction

Dans le cadre du 5^{ème} semestre de notre formation en BUT GEII, nous avons entrepris un projet ayant pour objectif d'offrir une solution pour la surveillance de cuve à récupération d'eau de pluie. Ces cuves sont régulièrement utilisées pour l'arrosage des jardins/potagers ainsi que pour l'utilisation d'eau sanitaire. L'inconvénient de telles installations est la surveillance de l'état des cuves car leurs installations sont principalement enterrées.

Notre projet se base sur la conception d'un système embarqué muni d'un capteur à ultrasons et d'un capteur température/humidité. Ainsi que sur la réalisation d'une page web afin d'exploiter les informations acquis par notre capteur. Dans le but de visualiser les chiffres clefs de notre cuve de récupération d'eau de pluie. Dans le but de faciliter la surveillance et l'utilisation du produit.

Ce rapport détaillé présentera les différentes phases de conception et de développement de notre capteur cuve connectée, ainsi que les choix technologiques et méthodologiques. Nous décrirons également les tests et les évaluations effectués pour garantir la qualité et les performances du produit final.

I. Le cahier des charges

A. Les spécifications techniques

Le cahier des charges pour le projet du capteur cuve connectée comprend plusieurs spécifications techniques clés :

Dimensions et Intégration : Le capteur doit être conçu de manière à pouvoir être inséré dans un tube en PVC de 100 mm de diamètre, garantissant ainsi une intégration discrète dans l'environnement des cuves de récupération d'eau de pluie.

Communication : Le capteur doit être capable de communiquer par wifi pour permettre le transfert d'informations vers un serveur distant, en l'occurrence Firebase. Cela permettra d'exploiter les données collectées tout au long de la journée pour surveiller l'état des cuves et du capteur lui-même.

Composants Électroniques : La partie électronique du capteur doit comporter un circuit imprimé sur mesure qui intègre un capteur TOF (Time-of-Flight, capteur ultrason) ainsi qu'un capteur de température/humidité. Ces composants sont cruciaux pour la collecte précise des données environnementales.

Carte Principale : Le choix a été fait d'utiliser un ESP32 comme carte principale. Cette carte jouera le rôle central dans le fonctionnement du capteur et dans la gestion des connexions wifi.

Autonomie Énergétique : Le capteur doit être capable de fonctionner pendant au moins 1 an sur batterie sans nécessiter de recharge. Pour atteindre cet objectif, l'utilisation du mode "deep sleep" est recommandée afin d'optimiser la consommation d'énergie.

Interface Web interne: Une interface web interne à l'ESP32 doit être développée. Cette interface servira de portail captif pour faciliter la connexion du capteur à un réseau wifi. Elle peut également être utilisée pour la configuration et la surveillance du capteur.

Interface Web externe : Une seconde page web externe permettra de visualiser les informations transférées sur le serveur firebase. L'utilisateur pourra consulter l'état de ses cuves via cette interface.

L'ensemble de ces spécifications vise à assurer le bon fonctionnement du capteur cuve connectée, en garantissant son intégration pratique, sa capacité à collecter des données pertinentes, sa communication efficace avec un serveur distant, et son autonomie énergétique sur une période prolongée.

B. Les livrables

Dans le cadre de ce projet, plusieurs livrables ont été produits pour concrétiser la conception du capteur cuve connectée. Ces éléments représentent notre travail au niveau du hardware, du logiciel et de l'interfaces dédiées à l'exploitation des données.

Circuit Imprimé sur Mesure : Le cœur technologique du capteur, ce circuit intègre les composants essentiels, tels que l'ESP32, le capteur ultrason SEN0311 et le capteur de température/humidité SHT45. Il assure la partie hardware des fonctionnalités du système.

Boîtier 3D Imprimé : Un boîtier sur mesure, créé par impression 3D. La conception 3D à été réalisée par l'un de nos prestataires externes. Ainsi que la production de celui-ci.

Logiciel Embarqué (ESP32) : Le logiciel intégré dans l'ESP32 est la partie la plus sensible du capteur. Il gère les capteurs, la connectivité, et assure le bon fonctionnement du dispositif.

Application Web pour Firebase : Une interface web dédiée à Firebase a été développée pour exploiter les données stockées. Elle offre aux utilisateurs la possibilité de visualiser, analyser et interpréter les mesures du capteur de manière conviviale.

II. Les choix technologiques

A. Les choix hardwares

1. Le choix du MCU

Nous allons vous présenter les différentes raisons qui nous ont poussé à choisir la carte de développement ESP32 pour piloter les différentes fonctions de notre cuve connectée.

Tout d'abord, nous avons besoin d'un microcontrôleur capable de gérer des technologies sans fil, comme le Wifi, pour répondre à notre cahier des charges. De plus, ce microcontrôleur devait pouvoir prendre en charge l'environnement de développement Arduino pour gagner du temps lors du développement logiciel. Enfin, il devait consommer peu afin que notre capteur cuve connectée puisse tenir 1 an sur batterie.

De plus, lorsque nous avons consulté la documentation technique du microcontrôleur, nous avons pu nous apercevoir que L'ESP32 possède deux cœurs et nous permet donc d'effectuer 2 tâches en parallèle. De plus, L'ESP32 possède de nombreuses entrées / sorties (GPIO), ce qui ouvre plus de possibilités d'évolutions pour l'avenir.

Enfin, nous avons choisi un ESP32 DevKit au lieu d'un ESP32 SoCs. En effet, cette option avait l'avantage d'être simple à mettre en œuvre et fiable. Elle nous assurait le bon fonctionnement des entrées / sorties et des périphériques de l'ESP32. De plus, le DevKit intègre la gestion d'une batterie, ce qui simplifie le routage et la schématique électronique.

2. Le capteur ultrason

Nous avons pris la décision d'intégrer le capteur ultrason SEN0311 dans notre cuve connectée en raison de ses caractéristiques adaptées à nos besoins spécifiques. Voici quelques éléments clés justifiant ce choix :

Étanchéité : Le SEN0311 est conçu pour être étanche, ce qui le rend idéal pour des applications liées à la gestion des cuves d'eau, ce qui est crucial pour assurer la durabilité du capteur.

Communication via UART : Le capteur communique via l'interface UART (Universal Asynchronous Receiver-Transmitter). Cette méthode de communication est largement utilisée et bien prise en charge dans l'environnement Arduino, simplifiant ainsi l'intégration du capteur dans notre système.

Portée de mesure : Le SEN0311 offre une portée de mesure allant jusqu'à 4.5 mètres. Cette portée est adéquate pour nos besoins, permettant une surveillance précise du niveau d'eau dans les cuves de récupération.

Compatibilité avec Arduino : Étant un capteur fréquemment utilisé dans l'environnement Arduino, le SEN0311 offre une intégration fluide avec notre microcontrôleur ESP32, basé sur Arduino. Cela facilite le processus de programmation et garantit la cohérence dans notre écosystème de développement.

3. Le capteur température/humidité

Nous avons opté pour l'intégration du capteur SHT45 pour la mesure de la température et de l'humidité dans notre cuve connectée. Les raisons de ce choix sont les suivantes :

Précision des Mesures : Le SHT45 est réputé pour sa haute précision dans la mesure de la température et de l'humidité. Cette caractéristique est cruciale pour garantir la fiabilité des données environnementales collectées par notre capteur.

Communication I2C : Le capteur SHT45 utilise le protocole de communication I2C, qui est un standard bien établi dans le domaine de l'électronique. Cette méthode de communication facilite l'intégration avec notre microcontrôleur ESP32, contribuant ainsi à la simplicité du système.

Large Plage de Mesure : Le SHT45 offre une plage étendue pour les mesures de température et d'humidité, ce qui le rend adaptable à divers environnements. Cela nous permet de recueillir des informations précises sur les conditions environnementales autour de la cuve.

Fiabilité et Durabilité : En tant que capteur de la famille Sensirion SHT, le SHT45 est reconnu pour sa fiabilité et sa durabilité. Ces caractéristiques sont essentielles, surtout dans des environnements susceptibles de présenter des variations de température et d'humidité.

Compatibilité avec l'ESP32 : Le SHT45 peut être facilement intégré à notre système basé sur ESP32. Grâce à de nombreuses bibliothèques disponibles sur internet, ce qui simplifie le processus de développement et assure une cohérence dans l'utilisation de composants électroniques.

B. Les choix softwares

Pour le développement du capteur cuve connectée, plusieurs librairies ont été judicieusement sélectionnées pour garantir des fonctionnalités avancées et une gestion optimale des ressources. Chacune de ces librairies contribue à des aspects spécifiques du projet, allant de l'intégration de la base de données à la communication asynchrone avec l'ESP32, en passant par la gestion de l'énergie et la manipulation des données. Nous allons examiner de plus près ces librairies clefs, qui sont au cœur du bon fonctionnement du capteur, permettant d'offrir des solutions robustes dans le respect de notre cahier des charges.

1. Firebase

Intégration de la base de données Firebase pour le stockage et l'exploitation des données collectées.

Fonctions Principales :

- Gestion de l'authentification.
- Configuration de la base de données.
- Manipulation des données.

2. AsyncWebServer

Mise en place d'un serveur web asynchrone pour la configuration et la communication avec l'ESP32.

Fonctions Principales :

- Gestion des requêtes HTTP de manière asynchrone.

3. DNSServer

Configuration d'un serveur DNS local pour faciliter la connexion à un réseau WiFi via un portail captif.

Fonctions Principales :

- Gestion des requêtes DNS pour le portail captif.

4. Wire

Bibliothèque pour la communication I2C avec le capteur de température/humidité.

Fonctions Principales :

- Facilite la communication série I2C.

5. EEPROM

Gestion de l'EEPROM (Flash dans notre cas, car L'ESP32 en n'est dépourvue) pour le stockage des informations de configuration WiFi.

Fonctions Principales :

-Lecture et écriture dans la Flash.

6. ESP Sleep

Gestion du mode deep sleep pour optimiser la consommation d'énergie.

Fonctions Principales :

-Configuration du réveil programmé pour sortir du mode deep sleep.

7. Spiff

Permet de créer une partition dans la flash.

Fonctions Principales :

-Rend possible le stockage de fichier (comme une clef USB).

C. Firebase

Dans le cadre de notre projet de capteur cuve connectée, la sélection de la solution Firebase est selon nous une décision stratégique. Les Différent critères techniques que nous avons pris en considérations sont les suivantes :

Facilité d'Intégration : Firebase offre une intégration aisée avec des dispositifs IoT, réduisant ainsi la complexité du développement. Cette facilité d'utilisation nous a permis de concentrer nos efforts sur les aspects spécifiques du projet plutôt que sur des détails techniques complexes.

Gestion des Données Simplifiée : La configuration simple de la base de données Firebase facilite la gestion des données collectées par le capteur. Cela nous a permis de rapidement appréhender et d'exploiter les fonctionnalités offertes par Firebase.

Sécurité et Authentification : FirebaseAuth intégré à Firebase offre une sécurité robuste. Notamment par l'ajout de règles d'authentification. La sécurité des données est une préoccupation majeure, et Firebase répond à cette exigence avec des protocoles avancés.

Flexibilité pour les Évolutions Futures : La flexibilité de Firebase nous donne la latitude nécessaire pour faire évoluer notre projet au fil du temps. Cette adaptabilité est essentielle pour anticiper les éventuelles améliorations du projet.

D. PlatformIO

Le choix de PlatformIO pour le développement de notre projet de capteur cuve connectée s'est imposé pour plusieurs raisons. Tout d'abord, l'intégration avec Visual Studio Code (VSCode) a été un critère important, il offre une interface de développement conviviale et une prise en charge multiplateforme, notamment pour le développement d'applications web.

La collaboration étant un aspect essentiel de notre projet, l'intégration native de PlatformIO avec Git et la facilité de travail en équipe sur des plateformes telles que GitHub ont renforcé notre choix. Cette approche a permis une gestion efficace des versions, des modifications et des branches du code.

PlatformIO s'est également avéré plus flexible lors du développement avec l'ESP32, offrant une compatibilité améliorée par rapport à d'autres environnements de développement tels que l'ancienne version de l'Arduino IDE. Cela a été crucial, notamment pour l'utilisation de la bibliothèque SPIFFS (Système de fichiers Flash SPI) et la possibilité de redimensionner des partitions dans l'ESP32. (voir annexes)

De plus, l'utilisation de Open Live Server dans le cadre de notre développement web a été grandement simplifiée grâce à la compatibilité de PlatformIO avec ces outils.

En résumé, notre choix de PlatformIO s'aligne sur notre volonté d'utiliser des outils modernes, flexibles et collaboratifs pour assurer le succès de notre projet de capteur cuve connectée

III. La réalisation

A. Le schématique électronique

1. Objectif

L'objectif premier de notre projet était de concevoir une carte électronique qui s'intégrerait parfaitement à une structure respectant les dimensions d'un tuyau en PVC de 100 mm de diamètre. Cette contrainte impliquait la création d'une carte compacte, permettant un accès facile au connecteur du capteur à ultrasons ainsi qu'à l'emplacement prévu pour la future batterie.

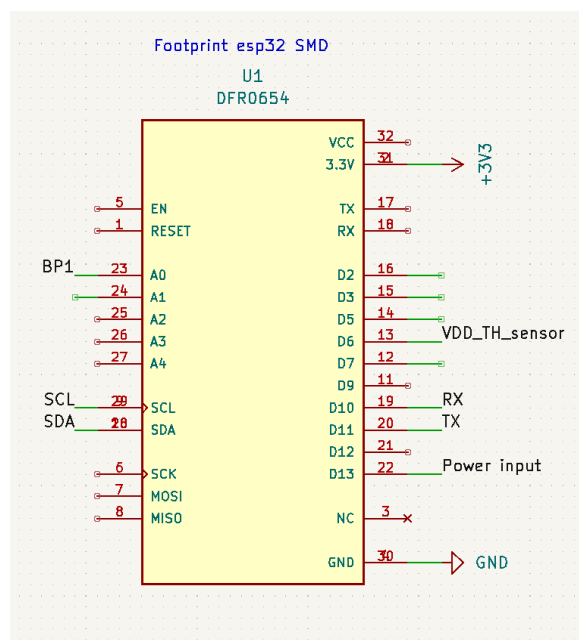
Pour optimiser l'espace et faciliter l'intégration mécanique, notre conception devait également tirer parti des fonctionnalités du mode "deep sleep". Ce mode permet une économie d'énergie significative en mettant certains composants du microcontrôleur dans un état de faible consommation entre les périodes d'activité.

Un autre aspect crucial était l'utilisation de l'antenne intégrée de base de notre ESP32. Cela nous obligeait à être attentifs pour ne pas compromettre sa sensibilité tout en assurant une connectivité fiable.

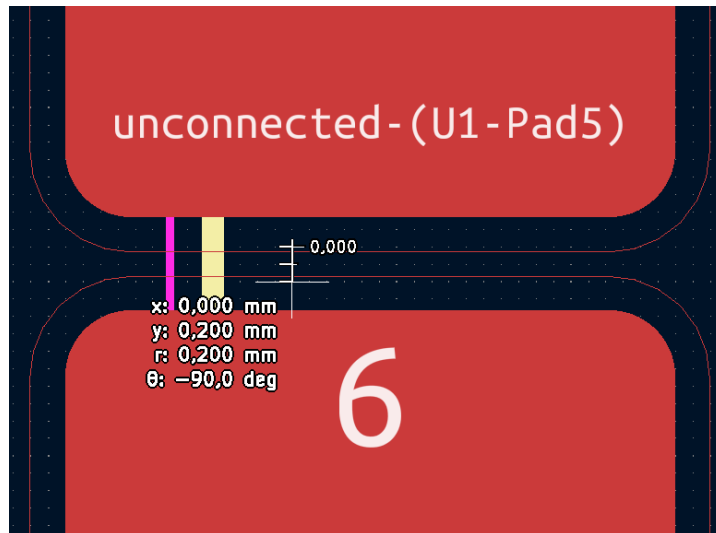
Dans la suite de cette section, nous détaillerons la schématique complète de notre conception, mettant en lumière les choix de routage effectués pour répondre à ces exigences spécifiques.

2. Création d'un footprint SMD pour Firebeetle

Notre objectif était de minimiser les dimensions de la carte tout en réduisant les coûts. En utilisant judicieusement les pads SMD de l'ESP32 de la marque DFRobot, nous avons optimisé l'espace, assurant compacité et économie. Cette approche, associée à des composants SMD, a permis une conception efficace répondant à nos contraintes budgétaires et dimensionnelles.



Lors de la conversion d'une carte initialement traversante en SMD, il est impératif de prendre en compte une isolation minimale de 0,15 de chaque côté des pads. Cette précaution est essentielle pour garantir la faisabilité du processus de fabrication. Un respect rigoureux de ces spécifications assure la réussite de la transition, évitant les complications potentielles et assurant la qualité du produit final.

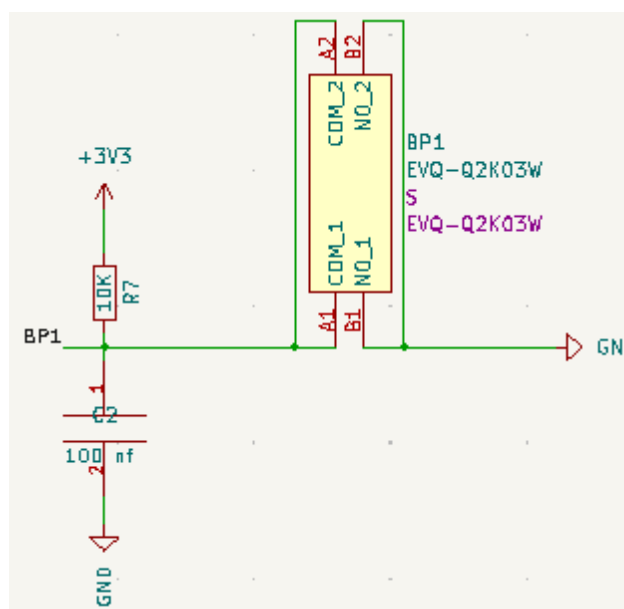


Astuce:

Pour les utilisateurs de KiCad : tirez pleinement parti des modèles 3D fournis par les fabricants ainsi que des datasheets mécaniques. En intégrant ces ressources, vous assurez la précision des empreintes sur votre carte électronique. Cette approche vous offre une vérification fiable pour garantir la conformité et la justesse de vos conceptions, contribuant ainsi à la réussite de votre projet.

3. Intégration du bouton poussoir

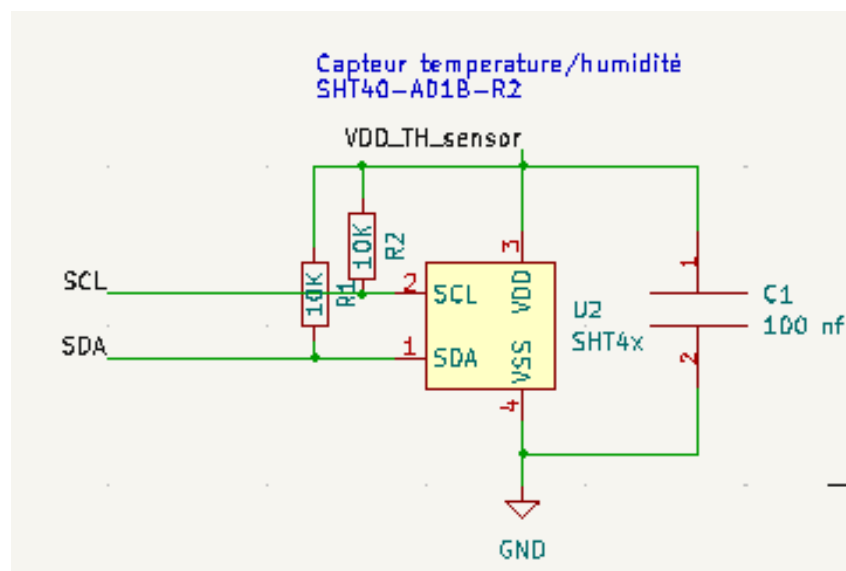
Nous avons inclus un bouton poussoir dans notre conception pour simplifier le développement de la carte. L'objectif était de permettre la sortie forcée du mode deep sleep afin de vérifier son bon fonctionnement pendant les phases de développement. Cette fonctionnalité offre une flexibilité accrue lors des tests et contribue à assurer la fiabilité du système dans des conditions réelles.



Nous avons pris la décision d'intégrer un filtre RC dans le circuit pour prévenir les risques de rebond associés au bouton poussoir. Ce filtre est constitué d'une résistance de 10 k Ω et d'un condensateur de 100 nF. De plus, nous avons connecté Com_1 à Com_2 ainsi que No_1 à No_2, car il s'agit d'un bouton poussoir à deux voies. Cette configuration garantit une connexion fiable du bouton, renforçant ainsi la stabilité de l'ensemble du système.

4. Intégration du capteur SHT45

Pour le capteur de température/humidité, notre choix s'est porté sur le SHT45. Ce capteur offre une précision de $\pm 0,1$ °C pour la température. Il utilise le protocole de communication I2C, ce qui facilite son intégration dans notre système. De plus, son format compact est idéal pour notre carte électronique. La disponibilité de nombreuses bibliothèques pour piloter ce capteur a également été un facteur déterminant, nous permettant de simplifier le développement en évitant la gestion des couches basses.



Pour garantir un contrôle optimal du capteur SHT45 pendant les phases de deep sleep, nous avons pris la décision stratégique d'ajouter une connexion VDD_TH_sensor. Cela nous offre la possibilité de forcer l'extinction du composant lorsque nécessaire.

En suivant rigoureusement les recommandations de la datasheet du fabricant, nous avons également intégré deux résistances et un condensateur essentiels au bon fonctionnement du capteur. Cette approche assure une intégration conforme aux spécifications du fabricant, contribuant ainsi à la stabilité des mesures de température et d'humidité.

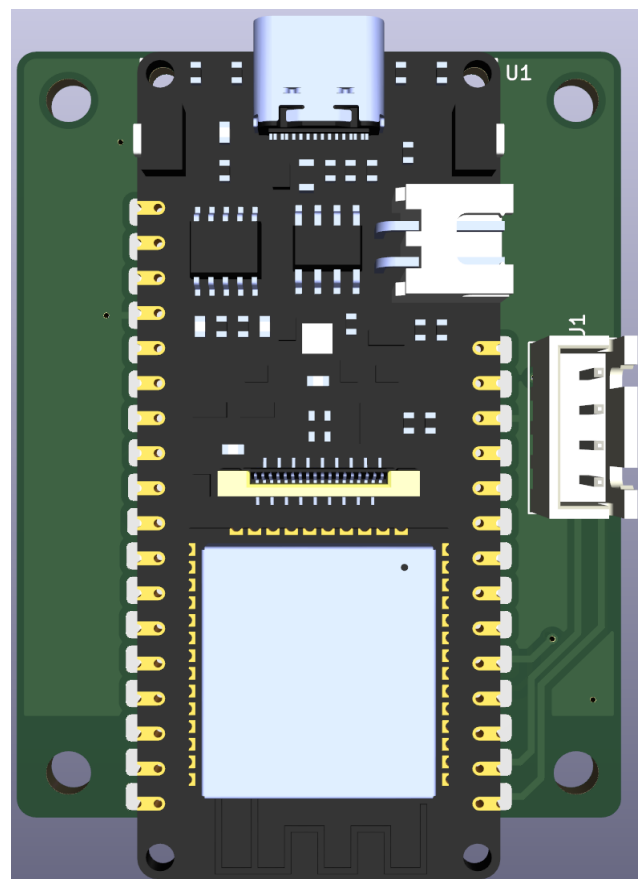
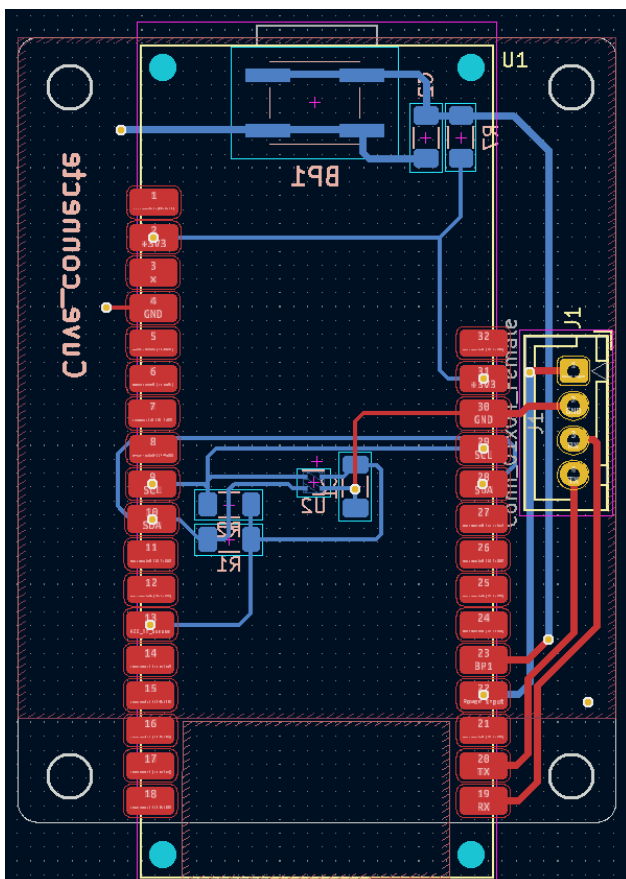
Notre schématique reste simple dans son ensemble. Il faut cependant faire attention à la connexion de chaque composant sur l'ESP32 pour profiter des différentes fonctions que proposent les ports GPIO.

B. Le Routage

Les contraintes de routage ont principalement été dictées par le placement stratégique des connecteurs, en veillant à ce que le port de recharge et le bouton poussoir soient facilement accessibles. Sur les conseils de notre tuteur de projet, nous avons décidé de placer les vias en dehors de l'emplacement de notre ESP32 en raison de préoccupations concernant l'isolation des vias.

Après vérification, il s'est avéré que les vias étaient correctement vernis, éliminant ainsi tout risque potentiel d'interférence. Cette décision de placement minutieuse garantit une disposition optimale tout en assurant l'intégrité structurelle de la carte électronique de la cuve connectée.

Une autre contrainte importante de routage consistait à éviter de placer un plan de masse sous l'antenne de l'ESP32. Pour résoudre ce problème, nous avons intégré une zone d'interdiction de placement sous l'antenne. Cette mesure vise à prévenir toute interférence nuisible à l'antenne et à garantir des performances de communication sans fil optimales pour la cuve connectée. En prenant en compte ces aspects, nous avons veillé à concevoir une disposition qui maximise l'efficacité du système tout en respectant les spécifications techniques.



1. Objectif

Le logiciel embarqué dans le microcontrôleur (MCU) est d'une importance cruciale dans le cadre de notre projet. En utilisant le langage de programmation C++, ce logiciel assure la liaison entre le monde physique et l'utilisateur. Son rôle principal est de collecter les données provenant des capteurs, de les transmettre à une base de données Firebase. Puis de passer en mode veille pour économiser l'énergie. Cette approche vise à optimiser l'efficacité énergétique du capteur cuve connectée tout en garantissant la capture précise des informations environnementales.

2. Mise en place du deep sleep

La gestion de la consommation énergétique constitue un élément crucial de notre conception. La prise de mesure a révélé une consommation de seulement 12 μA pour l'ESP32, démontrant ainsi une grande efficacité en termes d'économie d'énergie.

Nous avons opté pour un mode de fonctionnement permettant de d'allumer l'ESP32 après expiration du délai du watchdog, contribuant ainsi à minimiser la consommation énergétique pendant les périodes d'inactivité.

Pour réduire davantage la consommation, nous avons délibérément choisi de ne pas utiliser la mémoire RAM RTC, préférant plutôt stocker les informations WiFi dans la mémoire flash. Cette approche permet d'optimiser l'autonomie de la batterie et d'assurer un fonctionnement durable du capteur cuve connectée.

3. Intégration de firebase

La prise en main de la bibliothèque Firebase a été une étape essentielle de notre projet. Nous avons mis en place le nécessaire afin d'écrire des valeurs dans la base de données Firestore, permettant ainsi au capteur cuve connectée de stocker et d'exploiter les informations collectées.

Nous avons mis en place la lecture des champs de valeurs des paramètres depuis la base de données Firebase. L'utilisateur peut entrer les valeurs souhaitées sur l'application web. Cette fonctionnalité offre l'avantage de contrôler l'ESP32 à distance via une page web hébergée à l'extérieur.

Nous avons développé la possibilité de mettre à jour les valeurs du fichier de paramètres directement depuis l'ESP32. Cela permet d'ajuster dynamiquement des

paramètres tels que le nombre de mesures effectuées, ajoutant ainsi la possibilité d'envoyer à l'utilisateur le nombre total des mesures effectuées par le capteur.

4. Portail captif

Le portail captif fonctionne en redirigeant toutes les requêtes sortantes vers une page web interne à l'ESP32. Lorsqu'un utilisateur se connecte au réseau WiFi de l'ESP32, il est automatiquement redirigé vers cette page web grâce à un serveur DNS local, facilitant ainsi l'accès aux données et aux fonctionnalités du capteur cuve connectée.

La page web interne, permet d'entrer les informations d'identification d'un réseau wifi, afin de pouvoir s'y connecter. De plus, cette solution permet de visualiser en temps réel les données des capteurs directement depuis l'application web intégrée à l'ESP32.

ESP32 Cuve Serveur

<p>No WiFi Connected</p> <p>eduroam ▼ ↻</p> <p></p> <p>Valider</p>	<h3>Données</h3> <p>Température : 26.09 °C</p> <p>Humidité : 33.91 %</p> <p>Batterie : 4442 mV</p> <p>Distance : 900 mm</p>
--	---

"Page web interne"

5. Intégration d'un serveur de temps

Le serveur NTP (Network Time Protocol) joue un rôle essentiel en fournissant une référence temporelle précise à l'ESP32. Cette synchronisation temporelle permet d'associer une horodatage à chaque mesure effectuée par le capteur cuve connectée. Cette fonctionnalité offre la possibilité d'afficher des données temporelles précises, ce qui peut être particulièrement utile lors de la création d'interfaces utilisateur avec des fonctionnalités telles que des graphiques temporels pour une meilleure analyse des données.

Cela permet de mettre à jour l'horloge RTC (Real-Time Clock) intégré dans l'ESP32. Cette fonctionnalité permet de garder l'heure atomique exacte durant plusieurs mois.

6. Test de connexion

Pour vérifier la connectivité Internet de l'ESP32, nous avons mis en place un mécanisme de ping vers un serveur Google. Cette vérification utilise la bibliothèque "ESP32Ping.h", qui permet à l'ESP32 d'émettre des requêtes de ping vers un serveur distant, dans ce cas, les serveurs de chez Google. Cette fonctionnalité assure que l'ESP32 peut établir une connexion Internet fonctionnelle.

D. Réalisation de l'application web

4. Configuration de firebase

Pour la configuration de Firebase, nous avons intégré Node.js et ajouté les modules Firebase nécessaires pour l'authentification et la récupération des valeurs stockées sur Firestore. Cela nous a permis d'exploiter pleinement les fonctionnalités de Firebase dans notre projet.

5. Récupération des bases de données firestore

Pour la récupération des données depuis Firebase, nous avons établi deux collections distinctes pour organiser nos informations. La première collection, dédiée aux valeurs des capteurs transmises par l'ESP32, offre un aperçu précis des conditions de la cuve connectée. La deuxième collection est un fichier de paramètres, contenant notamment la fréquence de mesure, permettant ainsi un réglage flexible du capteur cuve connectée.

Une étape cruciale consiste à trier et à extraire soigneusement les informations stockées dans ces collections. Ce processus de triage nous a permis de structurer les données de manière à les rendre accessibles et exploitables pour une visualisation en temps réel.

6. Réalisation de l'application web

Pour le développement de l'application web, nous avons utilisé HTML, CSS et JavaScript, chacun jouant un rôle spécifique dans la conception de l'interface utilisateur. HTML a été utilisé pour déclarer la structure fondamentale du site, décrivant les éléments et leur organisation. CSS a pris en charge la mise en forme et le style, apportant une esthétique à la page. Quant à JavaScript, il a joué un rôle crucial dans les interactions dynamiques entre l'utilisateur et la base de données, facilitant des fonctionnalités telles que les menus déroulants et l'affichage de graphiques et d'histogrammes.

À noter que bien que HTML et CSS ne soient pas considérés comme de véritables langages de programmation, mais plutôt des langages descriptifs, ils jouent un rôle essentiel dans la création d'une interface utilisateur conviviale et visuellement attrayante.

(voir annexes)

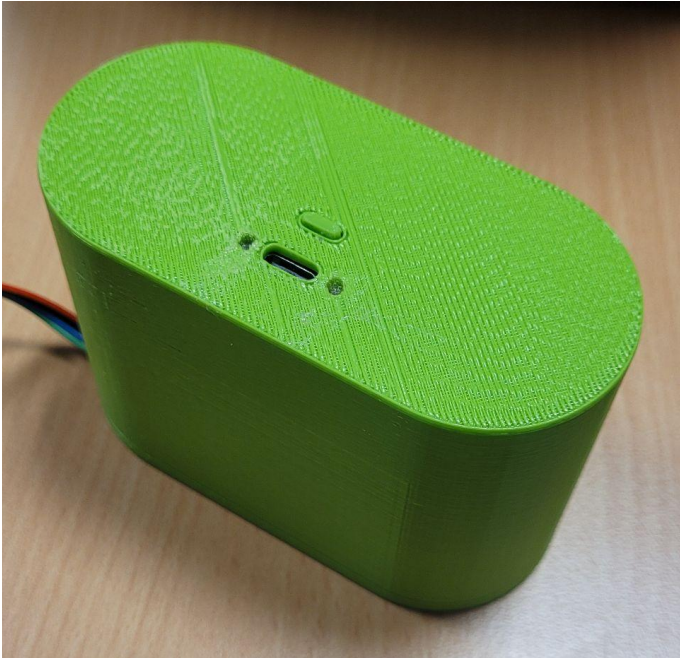
7. Les difficultés de Firebase

L'intégration de Firebase a constitué un défi majeur tout au long du projet, tant au niveau de son installation que de son utilisation complexe. La mise en place de cette plateforme a nécessité une compréhension approfondie de ses fonctionnalités et de son architecture.

E. La mécanique

Pour compléter notre projet, nous avons fait appel à un prestataire externe pour la fabrication d'une mécanique. Cette structure a été réalisée en impression 3D avec du PLA par un camarade de promo. La mécanique se compose d'une séparation dédiée à la batterie, qui fait également office de support pour notre ESP32.

L'ensemble est intégré dans un réceptacle. Nous avons également ajouté une cale spécifique pour le bouton poussoir, et nous avons laissé des ouvertures pour le passage du port USB C et pour la visibilité des LED témoins. Cette mécanique contribue à la robustesse et à l'ergonomie de notre solution.



IV. La conclusion et les perspectives d'amélioration

A. Ce que l'on retient de ce projet

Ce projet s'est révélé être une expérience technique enrichissante pour notre équipe. Il a constitué une opportunité unique d'explorer des technologies que nous n'avions pas encore rencontrées au cours de notre formation à l'IUT ainsi que lors de nos périodes en alternance.

Parmi les apprentissages notables, la création d'une application web a été une expérience novatrice. La familiarisation avec Firebase et l'utilisation de sa base de données Firestore ont été des aspects particulièrement formateurs. Cette plongée dans des technologies émergentes a élargi notre champ de compétences.

Travailler avec un binôme différent a également été un point bénéfique de ce projet. Cette collaboration a favorisé le développement de compétences en travail d'équipe, renforçant ainsi notre capacité à collaborer efficacement dans des contextes professionnels variés.

Dans l'ensemble, ce projet a été bien plus qu'une simple réalisation technique. Il a été une immersion dans de nouveaux domaines, un exercice de collaboration, et une occasion de relever des défis stimulants. Ces acquis constituent des fondations solides pour notre évolution professionnelle future.

B. Les perspectives d'amélioration

Dans cette partie, nous allons explorer les perspectives d'amélioration et d'évolutions futures envisagées pour la continuité du projet capteur cuve connectée. Ces ajustements et ajouts ont pour objectif de renforcer les performances, l'efficacité énergétique du capteur cuve mais aussi la convivialité de l'application web, tout en répondant aux besoins des futurs utilisateurs.

Notification d'Alerte Batterie : Intégrer une fonctionnalité de notification pour alerter l'utilisateur lorsque la batterie du capteur nécessite une recharge. Cela garantit une utilisation continue sans interruption.

Paramétrage de la Fréquence de Mesure : Permettre à l'utilisateur de définir la fréquence des mesures des capteurs, offrant ainsi une flexibilité quant au nombre de mesures quotidiennes et à l'autonomie du produit.

Utilisation de l'ESP32 Dernière Génération : Mettre à jour vers l'ESP32 de dernière génération pour réduire la consommation énergétique du capteur cuve, améliorant ainsi son efficacité. Ainsi que ses performances.

Amélioration de l'Exploitation des Mesures : Optimiser l'exploitation des mesures en intégrant des fonctionnalités telles que l'historique et les graphiques. Cela permettra aux utilisateurs d'effectuer des bilans annuels de l'état de leurs cuves, facilitant la gestion de la consommation d'eau, entre autres.

Amélioration de la Mécanique : Repenser le boîtier en vue de le rendre étanche et adapté aux mesures d'humidité, garantissant ainsi la durabilité et la précision du capteur.

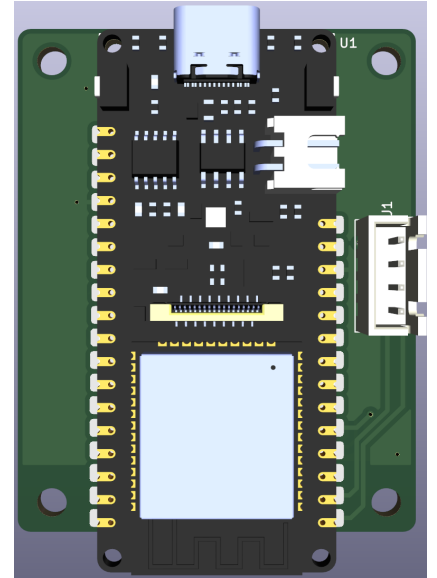
Notification de Connexion Internet : Intégrer une notification indiquant si l'ESP32 est correctement connecté à Internet, assurant ainsi la fiabilité des données collectées.

Mises à jour à Distance : Ajouter la possibilité de mettre à jour le capteur à distance, permettant l'amélioration continue du produit et l'ajout de nouvelles fonctionnalités.

Suppression de l'Historique de Mesure : Intégrer une fonctionnalité de suppression de l'historique des mesures pour éviter de dépasser les quotas définis par Firebase, assurant ainsi une gestion efficace des données. Ainsi que des coûts d'exploitation du produit.

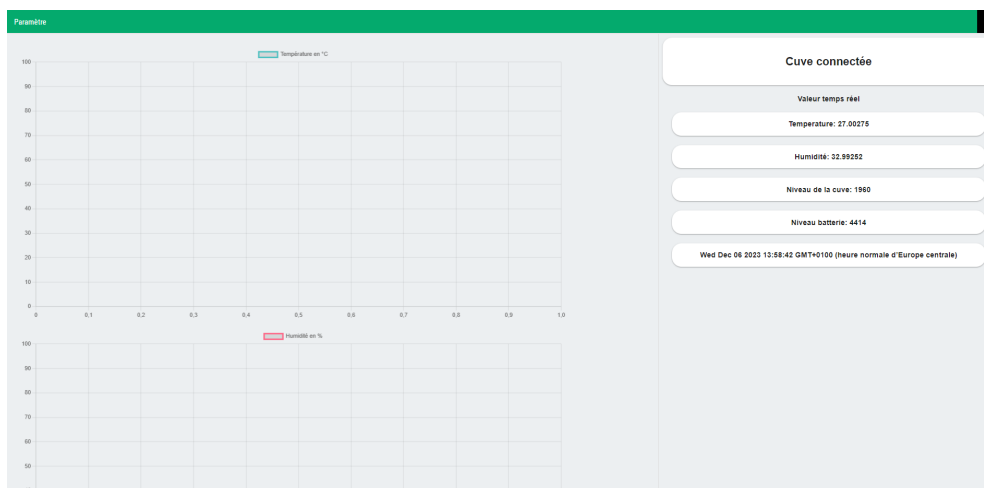
V. Le résumé

Le projet *capteur cuve connectée* a été réalisé par des étudiants de troisième année de BUT GEII en alternance. Il a pour objectif de rendre la surveillance des cuves de récupération d'eau de pluie plus facile. Ainsi avec notre capteur cuve, il est possible de superviser l'état de nos cuves à distance depuis une application web. Mais aussi de le connectée à internet depuis une page web locale. Notre capteur est sans fil et autonome pour 1 années de charge. Et est capable de données le niveau des cuves ainsi que la température/humidité.



VI. Abstract

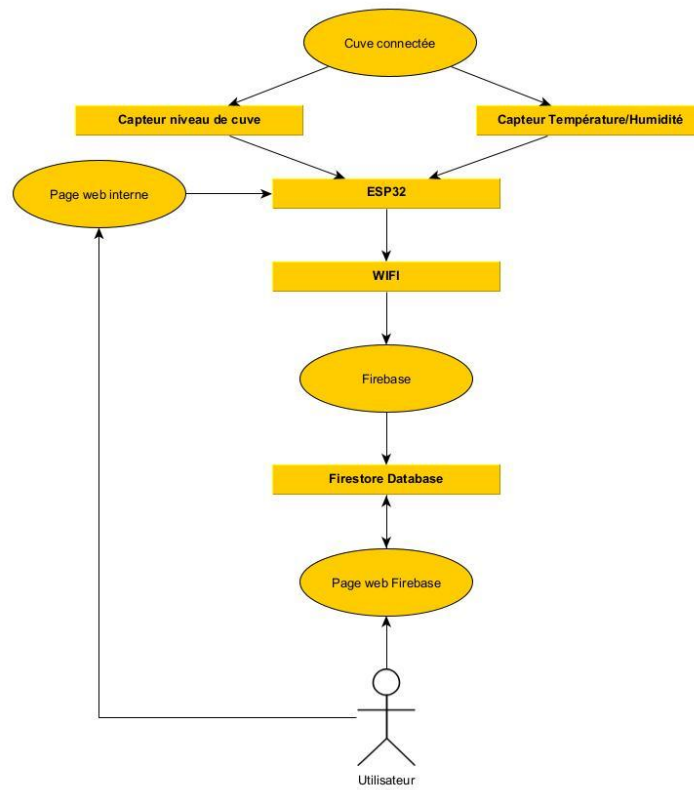
The connected tank sensor project was carried out by third year students from BUT GEII on a work-study basis. Its aim is to make monitoring of rainwater collection tanks easier. So with our tank sensor, it is possible to monitor the state of our tanks remotely from a web application. But also connected to the internet from a local web page. Our sensor is wireless and autonomous for 1 year of charge. And is capable of data on the level of the tanks as well as the temperature/humidity.



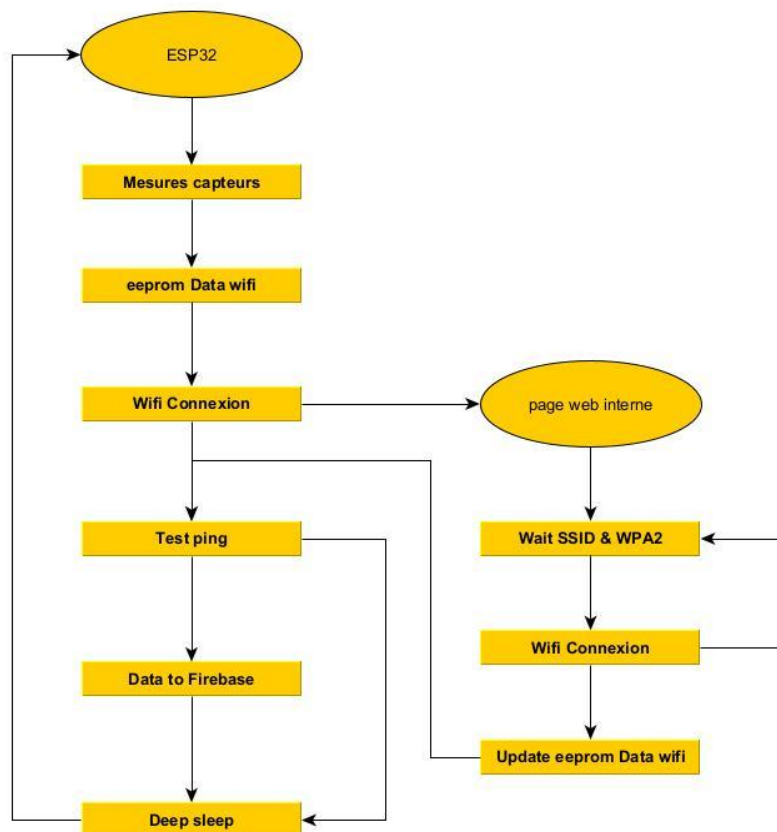
VII. Application web



VIII. Annexes



"Description global du capteur cuve connectée"



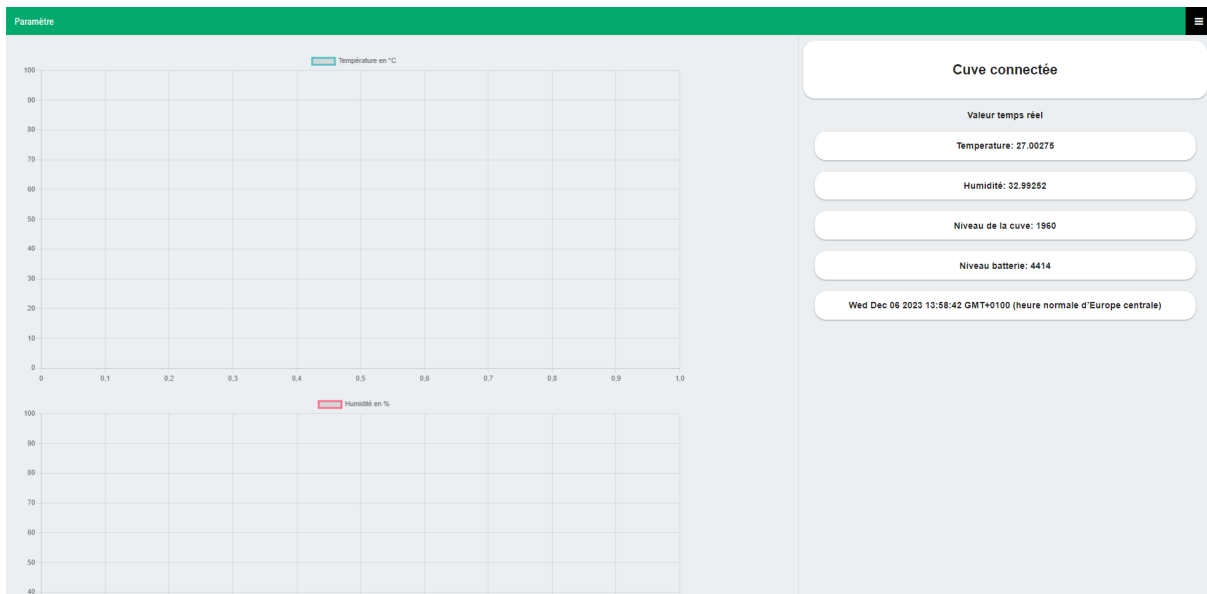
"Description du fonctionnement du logiciel embarqué"

```

# Name, Type, SubType, Offset, Size, Flags
# Note: if you have increased the bootloader size, make sure to update the offsets to avoid overlap
nvs, data, nvs, 0x9000, 0x5000,
otadata, data, ota, 0xe000, 0x2000,
app0, app, ota_0, 0x10000, 0x300000,
spiffs, data, spiffs, 0x310000, 0xE0000,
coredump, data, coredump, 0x3F0000, 0x10000,

```

“Description de la partition de l’ESP32”



“Application web”