

# Table des matières

<b>Table des figures</b>	<b>4</b>
<b>Liste des tableaux</b>	<b>5</b>
<b>Liste des Acronymes et Abréviations</b>	<b>6</b>
<b>1 Introduction Générale</b>	<b>7</b>
1.1 Contexte . . . . .	7
1.1.1 Utilisation du gaz domestique . . . . .	7
1.1.2 Problèmes réels rencontrés . . . . .	7
1.2 Problématique . . . . .	8
1.2.1 Formulation du problème . . . . .	8
1.2.2 Limites des solutions existantes . . . . .	9
1.2.3 Justification de l'approche ultrasonore . . . . .	9
1.3 Objectifs du Projet . . . . .	9
1.3.1 Objectif général . . . . .	9
1.3.2 Objectifs spécifiques . . . . .	9
1.4 Méthodologie de Travail . . . . .	11
<b>2 Analyse du Projet</b>	<b>12</b>
2.1 Étude des Solutions Existantes . . . . .	12
2.1.1 Méthodes de mesure de niveau . . . . .	12
2.1.2 Tableau comparatif des technologies . . . . .	16
2.1.3 Justification du choix de la solution ultrasonore . . . . .	17
2.2 Analyse Fonctionnelle Globale . . . . .	17
2.2.1 Modèle général du système . . . . .	17
2.2.2 Diagramme de contexte . . . . .	18
2.3 Analyse des Besoins . . . . .	19
2.3.1 Besoins fonctionnels . . . . .	19
2.3.2 Besoins non fonctionnels . . . . .	21
2.3.3 Scénarios d'utilisation principaux . . . . .	22
2.4 Contraintes du projet . . . . .	24
<b>3 Conception du Système Global</b>	<b>26</b>
3.1 Architecture Générale du Système . . . . .	26
3.1.1 Vue d'ensemble . . . . .	26
3.1.2 Description des dispositifs matériels . . . . .	27
3.1.3 Architecture de l'application web . . . . .	28
3.1.4 Architecture du frontend . . . . .	28
3.1.5 Architecture du backend . . . . .	29
3.2 Choix Technologiques . . . . .	29
3.2.1 Choix du microcontrôleur : Arduino Nano . . . . .	29

# TABLE DES MATIÈRES

---

3.2.2	Choix du capteurs ultrasonores . . . . .	30
3.2.3	Choix de l'écran : LCD I2C . . . . .	31
3.2.4	Choix du module de communication : LoRa . . . . .	31
3.2.5	Choix technologiques logiciels . . . . .	33
3.3	Principe de calibration . . . . .	35
3.3.1	Améliorations possibles (évolutions futures) . . . . .	35
3.4	Gestion de l'énergie . . . . .	35
3.4.1	Dispositif de mesure (capteur) . . . . .	35
3.4.2	Dispositif d'affichage (récepteur) . . . . .	36
3.4.3	Passerelle LoRa (gateway) . . . . .	37
3.5	Sécurité du système . . . . .	37
3.5.1	Sécurité électrique . . . . .	37
3.5.2	Sécurité mécanique et installation . . . . .	37
3.5.3	Sécurité fonctionnelle . . . . .	37
3.5.4	Sécurité données et vie privée . . . . .	38
<b>4</b>	<b>Conception et Réalisation des Appareils</b> . . . . .	<b>39</b>
4.1	Appareil 1 : L'Émetteur (Mesure) . . . . .	39
4.1.1	Composants matériels . . . . .	39
4.1.2	Implémentation logicielle . . . . .	39
4.2	Appareil 2 : Le Récepteur (Affichage et Alerte) . . . . .	39
4.2.1	Composants matériels . . . . .	40
4.2.2	Logique d'affichage et d'alerte . . . . .	40
4.3	Algorithmes de mesure et transmission . . . . .	40
4.3.1	Algorithme de l'émetteur . . . . .	40
4.3.2	Algorithme du récepteur . . . . .	40
4.4	Intégration Mécanique . . . . .	40
4.4.1	Installation sous la bouteille . . . . .	40
4.4.2	Boîtier du récepteur . . . . .	40
<b>5</b>	<b>Développement de l'Application Web</b> . . . . .	<b>41</b>
5.1	Modèle de données . . . . .	41
5.1.1	Vue d'ensemble du modèle . . . . .	41
5.1.2	Description des classes . . . . .	41
5.1.3	Relations entre les classes . . . . .	42
5.2	Analyse Fonctionnelle de l'Application . . . . .	43
5.2.1	Cas d'utilisation . . . . .	43
5.2.2	Synthèse des cas d'utilisation . . . . .	43
5.3	Parcours Utilisateur . . . . .	44
5.3.1	Principes généraux du parcours . . . . .	45
5.3.2	Parcours : Consultation du niveau de gaz . . . . .	45
5.3.3	Parcours : Consultation de l'historique de consommation . . . . .	45
5.3.4	Parcours : Affichage de la prédition de consommation . . . . .	45
5.3.5	Synthèse du parcours utilisateur . . . . .	45
5.4	Interfaces Utilisateur . . . . .	47
<b>6</b>	<b>Tests et Validation</b> . . . . .	<b>55</b>
6.1	Stratégie de Test . . . . .	55
6.2	Tests des Dispositifs Matériels . . . . .	55
6.2.1	Tests du dispositif de mesure (capteur) . . . . .	55
6.2.2	Tests du dispositif d'affichage (récepteur) . . . . .	56

# TABLE DES MATIÈRES

---

6.2.3	Tests de la passerelle LoRa (optionnelle) . . . . .	57
6.3	Tests de Communication LoRa . . . . .	57
6.3.1	Test de portée en ligne de vue . . . . .	57
6.3.2	Test de portée en milieu urbain avec obstacles . . . . .	58
6.3.3	Test de fiabilité et robustesse . . . . .	58
6.4	Tests Logiciels . . . . .	59
6.4.1	Tests firmware Arduino (capteur + afficheur) . . . . .	59
6.4.2	Tests de l'application web . . . . .	59
6.5	Tests d'Intégration Système Complet . . . . .	59
6.5.1	Test autonomie batterie en conditions réelles . . . . .	59
6.5.2	Validation globale du système . . . . .	59
6.6	Déploiement et Documentation . . . . .	59
6.6.1	Documentation livrée . . . . .	59
6.6.2	Code source et réutilisabilité . . . . .	60
6.7	Gallerie des tests . . . . .	60
<b>7</b>	<b>Difficultés Rencontrées et Solutions</b>	<b>64</b>
7.1	Difficultés Matérielles . . . . .	64
7.1.1	Difficulté d'acquisition du matériel . . . . .	64
7.1.2	Contraintes financières . . . . .	64
7.1.3	Disponibilité des bouteilles de gaz pour les tests . . . . .	65
7.2	Difficultés Logicielles . . . . .	65
7.2.1	Courbe d'apprentissage de React.JS . . . . .	65
7.2.2	Déploiement du backend . . . . .	65
7.3	Difficultés d'Intégration . . . . .	65
7.4	Leçons Apprises . . . . .	66
<b>8</b>	<b>Limites et Perspectives d'Évolution</b>	<b>67</b>
8.1	Limites Actuelles . . . . .	67
8.1.1	Limites matérielles . . . . .	67
8.1.2	Limites logicielles . . . . .	67
8.2	Améliorations Matérielles . . . . .	67
8.3	Améliorations Logicielles . . . . .	68
8.4	Évolution vers une Solution Commerciale . . . . .	68
8.4.1	Étapes envisagées . . . . .	68
8.4.2	Vision à long terme . . . . .	68
<b>9</b>	<b>Conclusion Générale</b>	<b>69</b>
9.1	Synthèse du Travail Réalisé . . . . .	69
9.2	Atteinte des Objectifs . . . . .	69
9.3	Apports Personnels et Compétences Acquises . . . . .	70
9.4	Perspectives . . . . .	70
<b>Références</b>		<b>72</b>

# Table des figures

1.1	Bouteille de gaz standard . . . . .	7
2.1	Exemple de solution de mesure du niveau de gaz par pesage . . . . .	13
2.2	Solution de mesure du niveau de gaz par mesure de la pression interne . . . . .	14
2.3	Principe de fonctionnement de la détection ultrasonore . . . . .	16
2.4	Modèle général du système . . . . .	18
2.5	Diagramme de contexte du système . . . . .	19
3.1	Architecture distribuée du système LoRa . . . . .	26
3.2	Schéma fonctionnel des deux dispositifs . . . . .	27
3.3	Positionnement du capteur ultrasonore sur la bouteille . . . . .	31
5.1	Diagramme de classe . . . . .	41
5.2	Diagramme de cas d'utilisation . . . . .	43
5.3	Diagramme d'activité "Consulter Historique"	46
5.4	Diagramme d'activité "Consulter Niveau de Gaz"	46
5.5	Diagramme d'activité "Afficher Prédiction"	46
5.6	Page d'accueil . . . . .	47
5.7	Formulaire d'utilisation 1 . . . . .	48
5.8	Formulaire d'utilisation 2 . . . . .	49
5.9	Formulaire d'utilisation 3 . . . . .	50
5.10	Tableau de bord . . . . .	51
5.11	Historique . . . . .	52
5.12	Prédictions . . . . .	53
5.13	Paramètres . . . . .	54
6.1	Cas du niveau de gaz élevé . . . . .	60
6.2	Cas du niveau de gaz moyen . . . . .	61
6.3	Cas du niveau de gaz bas . . . . .	62
6.4	Fonctionnement du buzzer . . . . .	63

# Liste des tableaux

1.1	Comparaison des solutions existantes . . . . .	9
2.1	Comparaison multicritère des solutions . . . . .	16
2.2	Besoins fonctionnels classés par priorité . . . . .	20
2.3	Contraintes techniques . . . . .	24
2.4	Planning prévisionnel détaillé . . . . .	25
3.1	Spécifications techniques Arduino Nano . . . . .	30
3.2	Profil consommation dispositif de mesure . . . . .	36
3.3	Consommation dispositif d'affichage . . . . .	37
5.1	Cas d'utilisation de l'application web . . . . .	44
6.1	Précision mesures ultrasoniques par niveau . . . . .	56
6.2	Validation déclenchement alertes par seuil . . . . .	57
6.3	Portée LoRa en ligne de vue (SF10, BW 125kHz, +17dBm) . . . . .	58
6.4	Portée LoRa en milieu urbain . . . . .	58

# Liste des Acronymes et Abréviations

<b>ADC</b>	Analog-to-Digital Converter (Convertisseur Analogique-Numérique)
<b>API</b>	Application Programming Interface
<b>BLE</b>	Bluetooth Low Energy
<b>FCFA</b>	Franc de la Communauté Financière Africaine
<b>FP</b>	Fonction Principale
<b>FS</b>	Fonction Secondaire
<b>GPL</b>	Gaz de Pétrole Liquéfié
<b>I2C</b>	Inter-Integrated Circuit
<b>IDE</b>	Integrated Development Environment
<b>IoT</b>	Internet of Things (Internet des Objets)
<b>LoRa</b>	Long Range
<b>LCD</b>	Liquid Crystal Display
<b>LED</b>	Light-Emitting Diode
<b>Li-ion</b>	Lithium-ion
<b>Li-Po</b>	Lithium-Polymère
<b>MCU</b>	Microcontroller Unit
<b>OLED</b>	Organic Light-Emitting Diode
<b>PCB</b>	Printed Circuit Board
<b>PWM</b>	Pulse Width Modulation
<b>REST</b>	Representational State Transfer
<b>SRAM</b>	Static Random-Access Memory
<b>SPI</b>	Serial Peripheral Interface
<b>UART</b>	Universal Asynchronous Receiver-Transmitter
<b>UI</b>	User Interface
<b>UX</b>	User Experience
<b>UUID</b>	Universally Unique Identifier

# Chapitre 1

## Introduction Générale

### 1.1 Contexte

#### 1.1.1 Utilisation du gaz domestique

L'utilisation du gaz butane comme source d'énergie domestique est largement répandue dans les ménages camerounais et dans de nombreux pays en développement. Ce gaz de pétrole liquéfié (GPL) représente une alternative économique et pratique au bois de chauffage et au charbon, contribuant ainsi à la réduction de la déforestation et des émissions de gaz à effet de serre.

Au Cameroun, les bouteilles de gaz domestique sont disponibles en plusieurs capacités (6 kg, 12,5 kg et 31 kg) et sont utilisées quotidiennement pour la préparation des repas dans une grande majorité des foyers urbains et périurbains. Selon les résultats de l'Enquête Intégrée sur les Conditions de Vie des Ménages (EICVM / MIS 2022) menée par l'Institut National de la Statistique (INS), environ 48,7 % des ménages urbains utilisent le gaz butane (GPL) comme principale source d'énergie pour la cuisson.



**Figure 1.1:** Bouteille de gaz standard

#### 1.1.2 Problèmes réels rencontrés

Malgré les avantages du gaz butane, son utilisation présente plusieurs inconvénients majeurs qui affectent le confort et la sécurité des utilisateurs :

**Coupures imprévues :** L'impossibilité de connaître avec précision le niveau de gaz restant dans la bouteille entraîne fréquemment des coupures imprévues, généralement au moment le moins opportun, notamment pendant la préparation des repas. Cette situation génère des désagréments importants et perturbe l'organisation domestique.

**Manipulations dangereuses :** Face à l'absence d'indicateur de niveau, les utilisateurs sont contraints de recourir à des méthodes empiriques pour estimer le contenu restant :

- **Soulèvement répété de la bouteille** pour en estimer le poids, ce qui peut causer des blessures dorsales, particulièrement pour les personnes âgées ou à mobilité réduite ;
- **Secouement de la bouteille** pour tenter d'entendre le liquide, méthode imprécise et potentiellement dangereuse.

**Risques de sécurité :** Ces manipulations répétées augmentent considérablement les risques :

- Détérioration des raccords et joints, pouvant entraîner des fuites de gaz ;
- Chutes de la bouteille lors des manipulations, avec risque d'explosion en cas de choc violent ;
- Exposition prolongée au gaz en cas de fuite non détectée.

**Mauvaise planification :** L'incertitude sur le niveau de gaz restant empêche une planification efficace des achats de recharge, conduisant soit à des remplacements prématurés (gaspillage économique), soit à des ruptures de stock au moment critique.

## 1.2 Problématique

### 1.2.1 Formulation du problème

La question centrale qui motive ce projet peut être formulée ainsi :

*Comment mesurer de manière fiable, sécurisée et non-intrusive le niveau de gaz restant dans une bouteille domestique, tout en fournissant à l'utilisateur une information claire et accessible en temps réel ?*

Cette problématique soulève plusieurs défis techniques et pratiques :

1. **Défi de la mesure non-intrusive** : La solution doit fonctionner sans modifier la structure de la bouteille, sans percer ou ouvrir le récipient sous pression, et sans contact direct avec le gaz liquéfié.
2. **Défi de la précision** : Le système doit fournir une mesure suffisamment précise pour permettre une planification efficace, tout en tenant compte des contraintes liées à la paroi métallique et aux variations de température.
3. **Défi de la sécurité** : Toute solution proposée doit respecter des normes de sécurité strictes, notamment l'absence d'étincelles, l'utilisation de composants basse tension, et la robustesse face aux conditions d'utilisation domestique.
4. **Défi de l'accessibilité** : Le système doit être abordable, facile à installer et à utiliser, sans nécessiter de compétences techniques particulières.
5. **Défi de l'autonomie** : L'appareil doit fonctionner de manière autonome avec une alimentation par batterie, sans nécessiter de connexion permanente au secteur.

### 1.2.2 Limites des solutions existantes

Plusieurs approches ont été proposées ou commercialisées pour résoudre ce problème, mais chacune présente des limitations significatives :

**Table 1.1:** Comparaison des solutions existantes

Solution	Avantages	Inconvénients
Pesage manuel	Simple, pas d'électronique	Nécessite manipulation, imprécis, pénible
Balance connectée	Précise, automatique	Coût élevé, encombrant, nécessite repositionnement
Capteur de pression	Mesure directe	Intrusif, risque de fuite, installation complexe
Capteur thermique externe	Non-intrusif	Imprécision, sensible aux conditions ambiantes
Jauge mécanique	Simples	Peu fiables, lecture approximative

### 1.2.3 Justification de l'approche ultrasonore

Face à ces limitations, nous avons opté pour une solution basée sur la **détection ultrasonore externe**, qui présente les avantages suivants :

- **Non-intrusif** : Aucune modification de la bouteille n'est requise ;
- **Sûr** : Pas de contact avec le gaz, utilisation de composants basse tension ;
- **Précis** : Capable de détecter l'interface liquide-gaz à travers la paroi métallique ;
- **Économique** : Coût des composants raisonnable et accessible ;
- **Fiable** : Technologie éprouvée dans d'autres applications industrielles.

## 1.3 Objectifs du Projet

### 1.3.1 Objectif général

L'objectif principal de ce projet est de :

*Concevoir, réaliser et valider un système électronique intelligent et connecté permettant la mesure non-intrusive du niveau de gaz butane dans une bouteille domestique, avec affichage local et accès à distance via une application web.*

### 1.3.2 Objectifs spécifiques

Pour atteindre cet objectif général, nous avons défini les objectifs spécifiques suivants :

### **OS1 : Mesure non-intrusive du niveau de gaz**

Développer un module de détection basé sur des capteurs ultrasonores capable de :

- Mesurer la hauteur du liquide dans la bouteille sans contact direct avec le gaz ;
- Atteindre une précision de  $\pm 5\%$  du volume total ;
- Fonctionner avec différentes tailles de bouteilles (6 kg, 12,5 kg, 13 kg) ;
- Réaliser une mesure complète en moins de 5 secondes.

### **OS2 : Affichage et supervision déportés**

Concevoir un dispositif de supervision distinct permettant de :

- Recevoir les données du capteur à distance via une liaison radio ;
- Visualiser le niveau de gaz sur un écran dédié (LCD/OLED) ;
- Émettre des alertes sonores et visuelles en cas de niveau critique ;
- Fonctionner de manière autonome dans une pièce de vie.

### **OS3 : Application web connectée**

Développer une application web offrant :

- Connexion au backend cloud via Internet ;
- Consultation à distance des données transmises par la passerelle LoRa ;
- Historique de consommation avec graphiques ;
- Prédictions de durée restante basées sur les habitudes d'utilisation ;
- Notifications ou alertes visuelles (niveau bas, anomalies détectées).

### **OS4 : Communication LoRa et infrastructure réseau**

Mettre en place une architecture de communication comprenant :

- Une liaison radio longue portée (LoRa) entre le capteur et l'afficheur ;
- Une passerelle LoRa optionnelle pour la remontée des données vers le cloud ;
- Un protocole de communication optimisé pour la faible consommation.

### **OS5 : Sécurité et fiabilité**

Garantir la sécurité du système en :

- Utilisant exclusivement des composants basse tension ( $\leq 5V$ ) ;
- Assurant l'isolation électrique totale entre circuits et bouteille ;
- Respectant les normes de sécurité électrique domestique ;
- Implémentant des protections contre les courts-circuits et surcharges ;
- Validant le système par des tests rigoureux.

## **OS6 : Autonomie et efficacité énergétique**

Optimiser la consommation énergétique pour :

- Atteindre une autonomie minimale de 30 jours avec batterie rechargeable ;
- Implémenter des modes de veille intelligents ;
- Permettre la recharge via port USB standard ;
- Afficher l'état de charge de la batterie.

## **1.4 Méthodologie de Travail**

Pour mener à bien ce projet, nous avons adopté une approche structurée en plusieurs phases :

**Phase 1 : Étude et conception (3 semaines)**

**Phase 2 : Développement matériel (4 semaines)**

**Phase 3 : Développement logiciel (4 semaines)**

**Phase 4 : Tests et validation (2 semaines)**

**Phase 5 : Documentation (2 semaines)**

# Chapitre 2

## Analyse du Projet

Ce chapitre présente l'analyse approfondie des besoins, des contraintes et des spécifications fonctionnelles du système de détection de gaz. Cette analyse constitue le fondement sur lequel repose toute la conception du projet.

### 2.1 Étude des Solutions Existantes

Cette section présente une analyse comparative des principales technologies existantes permettant d'estimer le niveau de gaz dans des récipients fermés, notamment les bouteilles de gaz domestique. Chaque méthode est étudiée selon son principe de fonctionnement, ainsi que ses avantages et ses limites dans le contexte du projet.

#### 2.1.1 Méthodes de mesure de niveau

##### Mesure par pesage

**Principe :** La méthode par pesage consiste à déterminer la quantité de gaz restante en mesurant le poids total de la bouteille. En connaissant le poids de la bouteille vide, il est possible de déduire la masse de gaz restante par simple soustraction. Cette approche repose sur le fait que la masse du gaz diminue progressivement au fur et à mesure de son utilisation.

- **Avantages :**

- Très bonne précision (inférieure à 1 %)
- Principe simple et largement utilisé
- Méthode non-intrusive, sans contact avec le gaz

- **Inconvénients :**

- Nécessite l'installation permanente d'une balance sous la bouteille
- Solution encombrante et peu pratique pour un usage domestique
- Sensible aux déplacements ou chocs de la bouteille
- Coût relativement élevé pour une balance précise et fiable
- Obligation de calibrer le système avec le poids exact de la bouteille vide



## Mesure par pesage

**Figure 2.1:** Exemple de solution de mesure du niveau de gaz par pesage

### Mesure par pression

**Principe :** La mesure par pression repose sur la mesure de la pression hydrostatique exercée par le liquide au fond du récipient. Dans un réservoir ouvert, cette pression est directement proportionnelle à la hauteur de liquide. Cependant, dans le cas d'une bouteille de gaz sous pression, cette méthode nécessite un accès direct à l'intérieur du récipient afin d'installer un capteur de pression.

- **Avantages :**

- Bonne précision lorsque la mesure est correctement réalisée
- Relation directe entre pression mesurée et niveau de liquide
- Peu influencée par la nature du liquide

- **Inconvénients :**

- **Méthode intrusive** nécessitant le perçage de la bouteille
- Risque important de fuite de gaz
- Installation complexe et dangereuse
- **Méthode non conforme aux exigences de sécurité pour les bouteilles de gaz domestique**



## Mesure par pression

**Figure 2.2:** Solution de mesure du niveau de gaz par mesure de la pression interne

### Mesure ultrasonore (Solution retenue)

**Principe :** La mesure ultrasonore repose sur l'émission d'ondes ultrasonores à travers la paroi métallique de la bouteille. Ces ondes se propagent à l'intérieur de la bouteille et se réfléchissent sur l'interface entre la phase gazeuse et la phase liquide. En mesurant le temps de parcours aller-retour des ondes, il est possible de déterminer avec précision la distance entre le capteur et la surface du liquide, permettant ainsi d'estimer le niveau de gaz restant sans avoir à ouvrir ou modifier la bouteille.

- **Avantages :**

- Méthode non-intrusive ne nécessitant aucune modification de la bouteille
- Précision acceptable pour un usage domestique (environ  $\pm 5\%$ )
- Temps de réponse rapide (quelques secondes)
- Faible sensibilité aux conditions environnementales
- Coût raisonnable des composants électroniques
- Technologie éprouvée et largement utilisée en milieu industriel
- Faible consommation énergétique compatible avec fonctionnement sur batterie
- Installation simple sous la bouteille sans perçage ni modification

- **Inconvénients :**

- Nécessite un bon couplage acoustique entre le capteur et la paroi
- Calibration initiale indispensable pour chaque type de bouteille
- Performances dépendantes de l'état de surface de la bouteille
- Sensibilité aux vibrations lors de la mesure
- Précision légèrement inférieure à la méthode par pesage



## Mesure ultrasonore (Solution retenue)

**Figure 2.3:** Principe de fonctionnement de la détection ultrasonore

### 2.1.2 Tableau comparatif des technologies

**Table 2.1:** Comparaison multicritère des solutions

Critère	Pesage	Pression	Ultrason
Précision (Note /5)	Excellent 5/5	Bonne 4/5	Bonne 4/5
Non-intrusif (Note /5)	Oui 5/5	Non 0/5	Oui 5/5
Sécurité (Note /5)	Bonne 4/5	Risque 2/5	Excellent 5/5
Coût (Note /5)	Élevé 2/5	Moyen 3/5	Moyen 4/5
Facilité (Note /5)	Moyenne 3/5	Complexe 2/5	Moyenne 3/5
Fiabilité (Note /5)	Bonne 4/5	Moyenne 3/5	Bonne 4/5
<b>TOTAL</b>	<b>23/30</b>	<b>14/30</b>	<b>25/30</b>

### 2.1.3 Justification du choix de la solution ultrasonore

Au vu de l'analyse comparative, la technologie ultrasonore s'impose comme le meilleur compromis pour notre application :

1. **Conformité aux contraintes de sécurité** : Aucune modification de la bouteille, absence de risque de fuite, composants basse tension uniquement.
2. **Précision suffisante** : L'objectif de  $\pm 5\%$  est atteignable avec une calibration appropriée et un traitement du signal optimisé.
3. **Coût maîtrisé** : Les capteurs ultrasonores sont disponibles à des prix raisonnables (3 000 - 15 000 FCFA pièce selon la qualité).
4. **Facilité d'implémentation** : La technologie est bien documentée avec de nombreuses ressources disponibles pour Arduino et microcontrôleurs embarqués.
5. **Faible consommation** : Compatible avec l'objectif d'autonomie de plusieurs semaines sur batterie grâce aux modes de veille profonde.
6. **Fiabilité prouvée** : Cette technologie est utilisée avec succès dans de nombreuses applications industrielles (mesure de niveau dans réservoirs, contrôle non destructif, détection d'objets).
7. **Installation simple** : Fixation rapide sous la bouteille sans outillage spécialisé ni compétences techniques particulières.

## 2.2 Analyse Fonctionnelle Globale

### 2.2.1 Modèle général du système

Le système de détection de gaz repose sur une architecture distribuée comprenant trois dispositifs matériels indépendants communiquant via technologie LoRa longue portée :

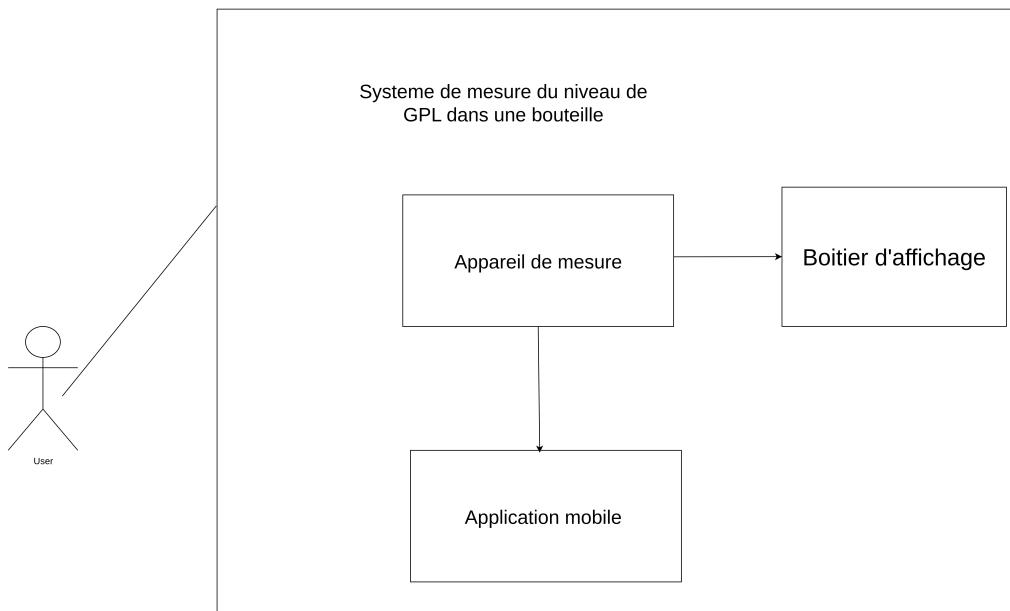


**Figure 2.4:** Modèle général du système

1. **L'utilisateur** : Personne utilisant la bouteille de gaz et souhaitant connaître le niveau restant sans avoir à se déplacer jusqu'à la bouteille.
2. **Le dispositif de mesure** : Module autonome fixé sous la bouteille, effectuant périodiquement les mesures ultrasonores et transmettant les données via LoRa. Fonctionne sur batterie Li-Po rechargeable.
3. **Le dispositif d'affichage** : Boîtier mural équipé d'un écran LCD, de LEDs et d'un buzzer, recevant les données du capteur et affichant le niveau de gaz en temps réel. Déclenche des alarmes visuelles et sonores en cas de niveau critique.
4. **La passerelle LoRa (optionnelle)** : Gateway permettant de transmettre les données vers un serveur cloud pour suivi à distance via application web.

### 2.2.2 Diagramme de contexte

Le diagramme de contexte illustre les interactions entre le système et son environnement :



**Figure 2.5:** Diagramme de contexte du système

Le système est principalement composé de trois sous-ensembles : **le dispositif de mesure**, **le dispositif d'affichage** et **la passerelle LoRa (optionnelle)**.

Le dispositif de mesure, fixé sous la bouteille de gaz, est chargé de collecter les données physiques liées au niveau de GPL à l'aide du capteur ultrasonique DYP-L06. Ces données sont traitées localement par un microcontrôleur Arduino Nano, puis transmises sans fil au dispositif d'affichage via communication LoRa 433 MHz.

Le dispositif d'affichage, installé dans un endroit accessible de l'habitation (mur, étagère, plan de travail), sert d'interface visuelle pour l'utilisateur. Il affiche le niveau de gaz en pourcentage sur un écran LCD I2C, utilise des LEDs pour indication rapide de l'état, et déclenche une alarme sonore via buzzer en cas de niveau critique. L'utilisateur peut interagir avec le système via des boutons physiques (allumage et arrêt d'alarme).

La passerelle LoRa optionnelle permet d'étendre les fonctionnalités du système en transmettant les données vers un serveur cloud via Wi-Fi ou Ethernet. Basée sur un ESP32, elle reçoit les trames LoRa des capteurs et les transmet à une plateforme IoT (MQTT, HTTP REST API, ou serveur personnalisé). Cela permet un suivi à distance via application mobile ou interface web, ainsi que la gestion simultanée de plusieurs bouteilles.

L'utilisateur interagit principalement avec le dispositif d'affichage pour la consultation locale, et optionnellement avec l'application web pour le suivi à distance, tandis que le dispositif de mesure fonctionne de manière totalement autonome avec alimentation sur batterie et gestion intelligente de la consommation énergétique.

## 2.3 Analyse des Besoins

### 2.3.1 Besoins fonctionnels

Les besoins fonctionnels décrivent ce que le système doit faire. Ils sont classés par priorité selon la méthode MoSCoW :

**Table 2.2:** Besoins fonctionnels classés par priorité

Priorité	Besoin	Description
<b>MUST HAVE (Indispensable)</b>		
BF01	Mesure du niveau	Le système doit mesurer le niveau de gaz avec une précision de $\pm 5\%$ via capteur ultrasonique
BF02	Affichage local	Le niveau doit être affiché en temps réel sur un écran LCD I2C avec bargraphe
BF03	Alerte niveau bas	Une alerte visuelle (LED) et sonore (buzzer) doit être déclenchée quand le niveau $< 15\%$
BF04	Autonomie capteur	Le dispositif de mesure doit fonctionner au moins 30 jours sur batterie Li-Po
BF05	Communication LoRa	Transmission fiable des données sur plusieurs kilomètres avec portée minimale de 500m en urbain
<b>SHOULD HAVE (Important)</b>		
BF06	Passerelle IoT	Gateway LoRa permettant connexion au cloud via Wi-Fi/Ethernet
BF07	Application web	Interface graphique pour consultation à distance via serveur cloud
BF08	Indicateurs LED	Affichage visuel rapide de l'état sur le dispositif d'affichage (vert/orange/rouge)
BF09	Alarme sonore	Buzzer actif paramétrable pour alertes critiques
BF10	Boutons physiques	Interaction utilisateur (allumage écran, arrêt alarme)
<b>COULD HAVE (Souhaitable)</b>		
BF11	Historique cloud	Stockage de l'historique de consommation sur serveur distant
BF12	Multi-bouteilles	Gestion de plusieurs capteurs par une seule passerelle
BF13	Statistiques	Analyse de consommation avec graphiques via application
BF14	Prédiction	Estimation du nombre de jours restants basée sur historique
BF15	Notifications	Alertes par email ou interface web en cas de niveau bas
<b>WON'T HAVE (Non prioritaire)</b>		
BF16	Détection de fuites	Alerte en cas de fuite détectée (capteur MQ-6 additionnel)
BF17	Géolocalisation	Localisation des points de vente de bouteilles
BF18	Partage social	Partage des statistiques sur réseaux sociaux
BF19	Commande vocale	Intégration assistants vocaux (Alexa, Google Home)

### 2.3.2 Besoins non fonctionnels

Les besoins non fonctionnels définissent les contraintes de qualité du système :

#### Performance

- **Temps de réponse** : Mesure complète et transmission en moins de 10 secondes
- **Précision** : Erreur maximale de  $\pm 5\%$  sur l'ensemble de la plage de mesure
- **Répétabilité** : Écart-type inférieur à 2% sur 20 mesures consécutives
- **Latence LoRa** : Délai maximal de 5 secondes entre mesure et affichage
- **Portée LoRa** : Communication fiable jusqu'à 2-5 km en zone dégagée, 500m-1km en milieu urbain
- **Fréquence de mesure** : Actualisation toutes les 15 minutes pour optimiser l'autonomie

#### Fiabilité

- **Disponibilité** : Taux de disponibilité supérieur à 99% pour le système de mesure
- **Durée de vie** : Minimum 5 ans en utilisation normale
- **MTBF** : Temps moyen entre pannes supérieur à 10 000 heures
- **Robustesse thermique** : Résistance aux variations de température (-10°C à +50°C)
- **Taux de perte LoRa** : Taux de paquets perdus inférieur à 1% en conditions normales
- **Récupération** : Reconnexion automatique en cas de perte temporaire de communication

#### Utilisabilité

- **Installation capteur** : Fixation sous bouteille en moins de 3 minutes sans outils
- **Installation affichage** : Montage mural ou sur support en moins de 5 minutes
- **Appairage LoRa** : Connexion automatique au démarrage sans configuration manuelle
- **Apprentissage** : Utilisation intuitive sans formation préalable
- **Lisibilité** : Affichage LCD visible à 3 mètres de distance
- **Accessibilité web** : Compatible avec tout navigateur web moderne (Chrome, Firefox, Safari)

#### Sécurité

- **Électrique** : Utilisation exclusive de composants basse tension ( $\leq 5V$ )
- **Isolation** : Aucun contact électrique avec la bouteille métallique
- **Protection batterie** : Circuit TP4056 avec protection surcharge/décharge/court-circuit
- **Boîtier étanche** : Protection IP54 minimum pour le dispositif de mesure
- **Certifications** : Conformité aux normes IEC 60335 (appareils domestiques)
- **RF** : Respect de la réglementation bande ISM 433 MHz (puissance max 10 mW ERP)

## Maintenabilité

- **Modularité** : Architecture permettant le remplacement de composants défaillants
- **Diagnostic** : LEDs d'état sur chaque dispositif pour identification rapide des problèmes
- **Batterie** : Remplacement facile des batteries sans démontage complet
- **Mises à jour** : Possibilité de reprogrammation du firmware via port UART
- **Documentation** : Manuel technique détaillé avec schémas électroniques et netlists
- **Calibration** : Procédure de recalibration accessible sans équipement spécialisé

## Énergie

- **Autonomie mesure** : Minimum 30 jours sur batterie Li-Po 1500 mAh en mode normal
- **Autonomie affichage** : Minimum 12-16 heures sur batteries 18650 en utilisation continue
- **Modes de veille** : Implémentation de deep sleep pour Arduino et LoRa entre mesures
- **Indication batterie** : Affichage du niveau de batterie sur LCD et LED dédiée
- **Recharge** : Temps de recharge complet inférieur à 4 heures (capteur) via USB

### 2.3.3 Scénarios d'utilisation principaux

#### Scénario 1 : Installation initiale

1. L'utilisateur fixe le dispositif de mesure sous la bouteille de gaz à l'aide de la sangle ajustable ou du système de fixation adhésif.
2. Il connecte la batterie Li-Po au module TP4056 et la place dans le boîtier du capteur.
3. Il installe le dispositif d'affichage à l'endroit souhaité (cuisine, salon) et insère les batteries 18650.
4. Il allume les deux dispositifs via leurs interrupteurs respectifs.
5. Le système effectue une auto-vérification : le capteur clignote une LED verte, l'écran LCD affiche un message de démarrage.
6. L'appairage LoRa se fait automatiquement (fréquence 433 MHz préprogrammée).
7. L'utilisateur lance la procédure de calibration via le bouton dédié sur le dispositif d'affichage.
8. Le capteur ultrasonique effectue plusieurs mesures pour détecter le niveau actuel et ajuste les paramètres.
9. La mesure initiale est effectuée et transmise via LoRa, puis affichée sur le LCD en pourcentage.

#### Scénario 2 : Consultation quotidienne

1. L'utilisateur se trouve dans la pièce où est installé le dispositif d'affichage.
2. Il appuie sur le bouton d'allumage pour activer le rétroéclairage du LCD (si éteint).
3. Le niveau de gaz actuel s'affiche immédiatement en pourcentage (ex: 67%) et sous forme de bargraphe.

4. Les LEDs indiquent l'état : verte (>50%), orange (20-50%), rouge (<20%).
5. L'heure de la dernière mesure est affichée sur la deuxième ligne du LCD.
6. Si l'utilisateur possède la passerelle cloud, il peut également consulter le niveau via l'application web depuis n'importe où.
7. Le système repasse automatiquement en veille d'affichage après 30 secondes sans interaction.

### Scénario 3 : Alerte de niveau bas

1. Le dispositif de mesure détecte lors d'une mesure périodique que le niveau est descendu sous le seuil de 15%.
2. Il transmet immédiatement cette information critique via LoRa au dispositif d'affichage.
3. Le buzzer du dispositif d'affichage émet un signal sonore intermittent (3 bips toutes les 10 secondes).
4. La LED rouge clignote de manière continue.
5. L'écran LCD affiche "NIVEAU BAS - 12%" avec une icône d'avertissement.
6. Si la passerelle cloud est installée, une alerte est visible sur le tableau de bord web.
7. L'application web affiche un message suggérant de prévoir un recharge rapidement.
8. L'utilisateur peut désactiver temporairement l'alarme sonore en appuyant sur le bouton "SILENCE", mais la LED rouge continue de clignoter.
9. L'alarme se réactivera automatiquement après 6 heures si le niveau n'a pas augmenté.

### Scénario 4 : Suivi à distance via passerelle cloud (optionnel)

1. L'utilisateur a installé la passerelle LoRa ESP32 connectée à son réseau Wi-Fi domestique.
2. La passerelle reçoit périodiquement les trames LoRa du capteur (toutes les 15 minutes).
3. Elle transmet ces données au serveur cloud via protocole MQTT ou HTTP REST API.
4. L'utilisateur ouvre l'application web via son navigateur même en étant absent de son domicile.
5. Il accède à l'onglet "Tableau de bord" qui affiche le niveau actuel de toutes ses bouteilles.
6. Il consulte l'onglet "Historique" pour voir l'évolution de la consommation sur les 7 derniers jours.
7. L'application affiche des graphiques de consommation quotidienne et hebdomadaire.
8. Il accède ensuite à l'onglet "Prédiction".
9. Un algorithme basé sur la consommation moyenne estime le nombre de jours restants avant épuisement complet.
10. L'application suggère une date optimale pour le remplacement de la bouteille.
11. L'utilisateur peut configurer les seuils d'alerte et la fréquence de mesure via l'application.

### Scénario 5 : Gestion de plusieurs bouteilles (avec passerelle)

1. Un utilisateur possède deux bouteilles : une en utilisation, une de rechange.
2. Il installe un capteur sous chaque bouteille (chaque capteur a un identifiant unique).
3. La passerelle LoRa reçoit les données des deux capteurs sur le même canal 433 MHz.
4. L'application web affiche deux widgets distincts : "Bouteille Cuisine" et "Bouteille Rechange".
5. Quand la bouteille principale atteint 10%, l'application suggère : "Préparez le remplacement - Recharge disponible à 92%".
6. L'utilisateur peut renommer, archiver, ou réassigner les capteurs via l'interface.
7. Le système conserve l'historique séparé de chaque bouteille pour analyse de consommation.

## 2.4 Contraintes du projet

### Contraintes techniques

**Table 2.3:** Contraintes techniques

Contrainte	Description
Non-intrusive	Aucune modification de la bouteille autorisée
Compatibilité	Doit fonctionner avec bouteilles 6kg, 12,5kg, 35kg
Paroi métallique	Le capteur ultrasonique doit traverser l'acier (épaisseur 2-3mm)
Température	Fonctionnement de -10°C à +50°C
Humidité	Résistance à 10-90% d'humidité relative
Vibrations	Résistance aux vibrations domestiques courantes
Alimentation	Fonctionnement sur batterie rechargeable sans alimentation secteur
Communication	Portée LoRa minimum 500m en milieu urbain, 2km en zone dégagée
Fréquence RF	Utilisation bande ISM 433 MHz (libre de licence)
Encombrement	Dispositif mesure: diamètre max 90mm, hauteur max 30mm
Étanchéité	Protection IP54 minimum pour le dispositif de mesure

### Contraintes économiques

- **Budget prototype global** : Maximum 200 000 FCFA pour système complet sans passerelle
- **Budget avec passerelle** : Maximum 300 000 FCFA incluant gateway ESP32
- **Répartition budgétaire** :

- Dispositif de mesure : 20 000 - 40 000 FCFA
  - Dispositif d'affichage : 20 000 - 40 000 FCFA
  - Passerelle LoRa (opt) : 15 000 - 30 000 FCFA
  - PCB et boîtiers : 10 000 - 20 000 FCFA
- **Composants** : Priorité aux composants disponibles localement (Cameroun) ou via fournisseurs internationaux (AliExpress, Mouser)
  - **Coût cible production** : Coût unitaire de production série inférieur à 60 000 FCFA
  - **Maintenance** : Coût de maintenance annuel inférieur à 2 000 FCFA (remplacement batteries)

### Contraintes temporelles

**Table 2.4:** Planning prévisionnel détaillé

Phase	Activités	Durée	Livrables
1	Étude préliminaire	2 semaines	Rapport d'étude, choix LoRa
2	Conception détaillée	4 semaines	Schémas électroniques, netlists
3	Approvisionnement	2 semaines	Composants achetés
4	Réalisation matérielle	5 semaines	Prototypes fonctionnels (3 dispositifs)
5	Développement firmware	4 semaines	Code Arduino + ESP32
6	Tests communication LoRa	1 semaine	Validation portée et fiabilité
7	Calibration capteur	2 semaines	Procédure validée
8	Développement backend (opt)	2 semaines	Serveur cloud + API
9	Tests et validation	2 semaines	Résultats des tests
10	Documentation	2 semaines	Rapport final complet

**TOTAL : 26 semaines** (avec passerelle cloud)

**TOTAL : 24 semaines** (version standalone sans cloud)

# Chapitre 3

## Conception du Système Global

Suite à l'analyse des besoins et des contraintes, ce chapitre détaille la conception technique du système proposé. Il présente l'architecture globale, justifie les choix technologiques opérés et décrit les stratégies mises en œuvre pour garantir la fiabilité, l'autonomie et la sécurité du dispositif.

### 3.1 Architecture Générale du Système

#### 3.1.1 Vue d'ensemble

Pour répondre aux contraintes d'éloignement et d'autonomie, le système repose sur une architecture distribuée. Nous avons opté pour la technologie radio LoRa, qui assure une liaison robuste entre l'extérieur (lieu de stockage du gaz) et l'intérieur de l'habitation.

Le système s'articule autour de trois entités distinctes :



**Figure 3.1:** Architecture distribuée du système LoRa

- **Le dispositif de mesure (Capteur)** : Placé directement sous la bouteille, ce module autonome a pour unique fonction de mesurer le niveau de gaz et de transmettre l'information. Il est conçu pour être "invisible" à l'usage et très économique en énergie.
- **Le dispositif d'affichage (Récepteur)** : Situé dans la pièce de vie, il joue le rôle d'interface homme-machine. Il reçoit les données et informe l'utilisateur via un écran et des signaux lumineux ou sonores, sans nécessiter de connexion internet.

- **La passerelle LoRa (Optionnelle)** : Cet élément permet d'ouvrir le système vers l'Internet des Objets (IoT). Elle fait le pont entre le réseau local LoRa et le Cloud, permettant ainsi l'usage de l'application web.

Cette approche modulaire offre une grande flexibilité : le système est parfaitement fonctionnel en mode "local" (capteur + afficheur) pour les utilisateurs sans internet, tout en étant extensible vers une solution connectée complète.

### 3.1.2 Description des dispositifs matériels

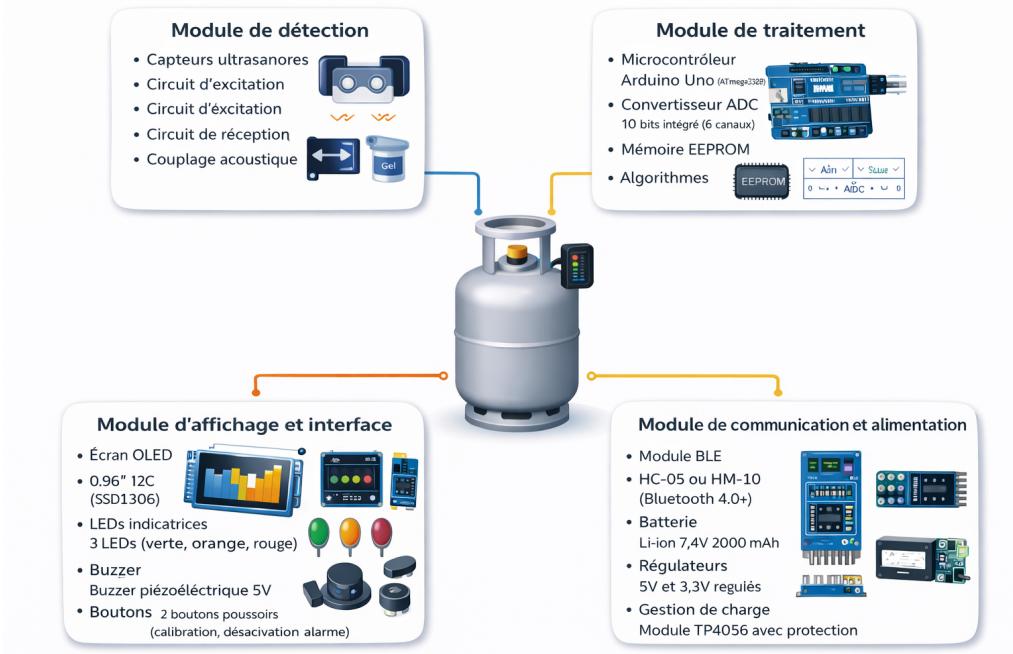


Figure 3.2: Schéma fonctionnel des deux dispositifs

#### Le dispositif de mesure (Module Capteur)

Ce module est le cœur du système d'acquisition. Installé sous la bouteille, il doit faire face à des contraintes d'encombrement et d'autonomie sévères.

Sa conception s'articule autour du capteur ultrasonore **DYP-L06**, choisi pour sa capacité à mesurer à travers le métal et sa faible consommation. Le pilotage est assuré par un **Arduino Nano**, qui orchestre les mesures et la mise en veille. La transmission est confiée au module **LoRa SX1278**, configuré pour une portée optimale en milieu urbain (jusqu'à 1 km).

L'alimentation est fournie par une batterie Li-Po de 3.7V, gérée par un module de charge TP4056 qui assure la sécurité électrique. L'ensemble est intégré dans un boîtier compact (Ø90mm) et étanche (IP54), conçu pour se fixer discrètement sous la bouteille.

**Stratégie de fonctionnement** : Pour maximiser l'autonomie, le dispositif ne fonctionne pas en continu. Il suit un cycle strict :

1. **Réveil** périodique (ex: toutes les 15 minutes).
2. **Mesure** ultrasonore et calcul du niveau.
3. **Transmission** de la donnée via LoRa.
4. **Mise en veille profonde** immédiate de tous les composants.

Cette stratégie permet de réduire drastiquement la consommation moyenne, offrant une autonomie théorique de plusieurs semaines.

### Le dispositif d'affichage (Module Récepteur)

Ce module est conçu pour être installé confortablement dans l'habitation. Il centralise les informations et gère les alertes.

Il partage la même base technologique (Arduino Nano et LoRa SX1278) mais fonctionne en réception. L'interface utilisateur est composée d'un **écran LCD I2C** pour l'affichage précis et de **LEDs tricolores** pour une lecture rapide de l'état (Vert/Orange/Rouge). Un buzzer est intégré pour les alertes critiques.

L'alimentation est ici plus robuste, basée sur deux batteries 18650 en série, régulées à 5V, permettant une utilisation prolongée sans recharge fréquente.

Le système gère trois états :

- **Veille** : Écran éteint, radio en écoute (consommation réduite).
- **Actif** : Écran allumé sur demande utilisateur ou réception de données.
- **Alerte** : Activation des signaux sonores et visuels en cas de niveau critique.

### La passerelle LoRa (Module Gateway - Optionnel)

Pour les utilisateurs souhaitant une connectivité étendue, la passerelle joue le rôle de pont. Basée sur un microcontrôleur **ESP32** (choisi pour sa connectivité Wi-Fi native), elle écoute les messages LoRa et les retransmet vers un serveur distant.

Contrairement aux autres modules, elle est alimentée sur secteur, ce qui lui permet d'être en écoute permanente et de gérer simultanément plusieurs bouteilles si nécessaire.

#### 3.1.3 Architecture de l'application web

L'application web constitue le terminal de supervision avancé. Elle ne communique pas directement avec les capteurs, mais interroge le serveur Cloud (Backend).

Son rôle dépasse le simple affichage : elle transforme les données brutes en informations utiles. Elle permet notamment de visualiser l'historique de consommation, de consulter les alertes en cas de niveau bas où que l'on soit, et d'utiliser des algorithmes prédictifs pour estimer la date de remplacement de la bouteille.

#### 3.1.4 Architecture du frontend

L'architecture de l'application repose sur le framework **React.js**, suivant le modèle SPA (Single Page Application). Cette approche permet une expérience utilisateur fluide, sans recharge complet de la page lors de la navigation.

L'application est structurée autour de composants réutilisables :

1. **Les Composants d'Interface (UI)** : Boutons, graphiques, cartes d'information qui constituent les briques visuelles de l'application.
2. **La Gestion d'État (State Management)** : Utilisation des Hooks (`useState`, `useEffect`) et du Context API pour gérer les données globales (utilisateur connecté, liste des bouteilles) de manière efficace.
3. **Le Routage** : Gestion de la navigation entre les différentes vues (Tableau de bord, Historique, Paramètres) via React Router.
4. **La Couche Service** : Modules dédiés à la communication avec l'API REST du backend (via Axios), assurant la récupération et l'envoi sécurisé des données.

### 3.1.5 Architecture du backend

Le Backend, développé avec le framework **Spring Boot**, est le cerveau de la partie connectée. Il assure la persistance des données et l'exécution de la logique métier complexe.

L'architecture logicielle suit le modèle MVC (Modèle-Vue-Contrôleur) pour garantir une séparation claire des responsabilités :

1. **Les Contrôleurs REST** gèrent les entrées/sorties. Ils reçoivent les mesures de la passerelle et répondent aux requêtes de l'application web, tout en sécurisant les accès via des jetons JWT.
2. **Les Services Métier** contiennent l'intelligence du système : validation des données, détection d'anomalies, calculs statistiques et déclenchement des notifications.
3. **La Couche de Données (DAO)** interagit avec la base de données PostgreSQL pour stocker de manière pérenne les utilisateurs, les bouteilles et l'historique des mesures.

## 3.2 Choix Technologiques

### 3.2.1 Choix du microcontrôleur : Arduino Nano

Le choix du microcontrôleur est critique pour l'équilibre entre performance, taille et consommation. Nous avons sélectionné l'**Arduino Nano** pour équiper à la fois le capteur et l'afficheur.

#### Justification du choix

Plusieurs arguments justifient cette décision :

- **Compacité** : Avec ses dimensions réduites (18×45mm), il s'intègre parfaitement dans l'espace restreint sous la bouteille.
- **Efficacité énergétique** : Il supporte des modes de veille profonde ("Deep Sleep") essentiels pour atteindre l'autonomie visée.
- **Standardisation** : L'utilisation du même composant sur les deux modules simplifie la maintenance, le développement du code et la gestion des stocks.
- **Accessibilité** : Son faible coût et sa grande disponibilité en font un choix pertinent pour un déploiement à grande échelle.

## Caractéristiques techniques détaillées

**Table 3.1:** Spécifications techniques Arduino Nano

Paramètre	Valeur
Microcontrôleur	ATmega328P (8-bit AVR)
Fréquence horloge	16 MHz
Mémoire Flash	32 KB (dont 2 KB bootloader)
SRAM	2 KB
EEPROM	1 KB
Tension fonctionnement	5V (VIN accepte 7-12V)
Tension logique	5V (compatible 3.3V via diviseur)
Dimensions	18 × 45 mm
Poids	7g
Pins I/O numériques	14 (dont 6 PWM)
Pins analogiques	8 (ADC 10-bit)
Consommation active	20 mA
Consommation veille profonde	<5 mA (avec LowPower lib)
Interfaces	UART, I2C, SPI
Connecteur USB	Mini-USB (programmation + serial monitor)

### 3.2.2 Choix du capteurs ultrasonores

#### Analyse et justification du choix

La contrainte majeure du projet est de mesurer le niveau de gaz sans modifier la bouteille (pour des raisons de sécurité et de réglementation). La technologie ultrasonore s'est imposée comme la seule solution viable permettant une mesure "à travers" la paroi métallique.

Nous avons sélectionné le capteur **DYP-L06**. Contrairement aux capteurs standards (type HC-SR04), ce modèle est étanche (IP67) et conçu pour être couplé à une surface solide. Il offre une précision de  $\pm 1\%$ , largement suffisante pour notre besoin, et fonctionne sur une plage de tension compatible avec notre batterie.

#### Principe de fonctionnement

Le principe physique repose sur la mesure du temps de vol (Time of Flight). Le capteur, positionné au sommet de la bouteille, émet une onde ultrasonore qui traverse le gaz.

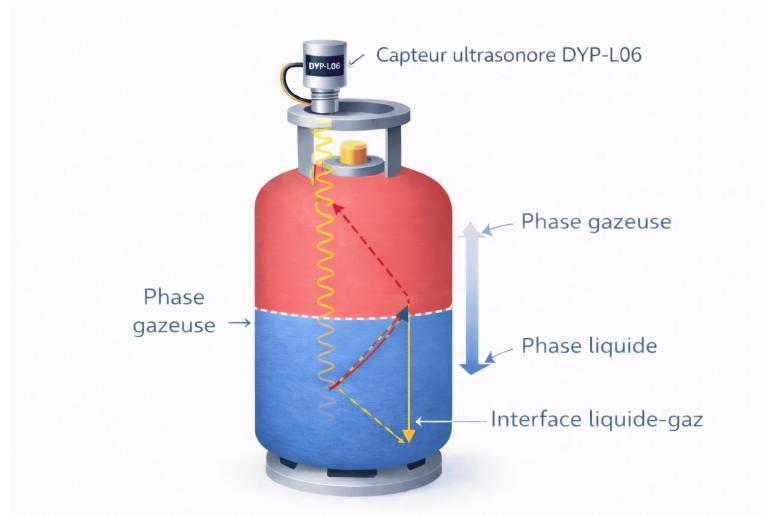
Lorsque cette onde rencontre la surface du liquide (GPL), elle est réfléchie vers le capteur. En mesurant le temps aller-retour  $\Delta t$ , et connaissant la vitesse du son dans le gaz  $v_{gaz}$ , on déduit la distance  $d$  entre le haut de la bouteille et le liquide :

$$d = \frac{v_{gaz} \times \Delta t}{2}$$

Le niveau de remplissage est ensuite obtenu par soustraction par rapport à la hauteur totale de la bouteille.

$$\text{Niveau (\%)} = \frac{H_{tot} - d}{H_{tot}} \times 100$$

## Configuration mécanique du capteur



**Figure 3.3:** Positionnement du capteur ultrasonore sur la bouteille

L’installation du capteur est non-destructive. Il est maintenu fermement contre la paroi métallique (généralement sous la bouteille ou sur le dessus selon le modèle) à l’aide d’un adhésif industriel ou d’une sangle magnétique.

Un point crucial est l’application d’un gel de couplage (ou pâte thermique) entre le capteur et le métal. Ce gel élimine la fine couche d’air qui empêcherait la transmission des ultrasons, garantissant ainsi la fiabilité de la mesure.

## Avantages de la solution retenue

Cette approche présente un compromis optimal :

- **Sécurité** : Aucun contact avec le gaz, aucun perçage.
- **Simplicité** : Un seul capteur suffit, réduisant le coût et la complexité.
- **Efficacité** : La mesure est directe et peu influencée par la composition chimique exacte du mélange propane/butane.

### 3.2.3 Choix de l’écran : LCD I2C

Pour l’interface utilisateur, nous avons privilégié la lisibilité et la simplicité. L’écran LCD I2C a été retenu au détriment de l’OLED.

#### Justification du choix LCD I2C

Bien que moins moderne que l’OLED, le LCD offre une meilleure lisibilité à distance (essentiel pour un affichage mural) et une consommation maîtrisée une fois le rétroéclairage éteint. De plus, son interface I2C simplifie considérablement le câblage (seulement 4 fils), augmentant la fiabilité globale du montage.

### 3.2.4 Choix du module de communication : LoRa

L’adoption de la technologie LoRa (Long Range) représente une évolution majeure par rapport aux solutions Bluetooth classiques. Ce choix est motivé par la nécessité de traverser les obstacles (murs, dalles) séparant souvent la cuisine du lieu de stockage du gaz.

## Justification du choix LoRa

LoRa offre une portée de plusieurs centaines de mètres en milieu urbain et une excellente pénétration des structures en béton, là où le Bluetooth peinerait à dépasser 10 mètres. De plus, sa très faible consommation en veille est parfaitement adaptée à notre usage intermittent (envoi de quelques octets toutes les 15 minutes).

Nous utilisons le module **SX1278** fonctionnant sur la bande libre 433 MHz. La communication est configurée en mode "Point-à-Point" (P2P), ce qui permet aux modules de dialoguer directement sans dépendre d'une infrastructure réseau externe ou d'un abonnement opérateur.

## Module LoRa choisi : SX1278

Nous utilisons le chipset **Semtech SX1276/SX1278** intégré dans modules breakout compatibles Arduino (RA-02, RFM95W, E32 selon disponibilité fournisseurs).

### Caractéristiques techniques SX1278 :

- Chipset : Semtech SX1276/SX1278 (référence industrie)
- Fréquences supportées : 137-1020 MHz (version 433 MHz utilisée)
- Puissance émission : Réglable -4 dBm à +20 dBm (max 100mW)
- Sensibilité réception : Jusqu'à -148 dBm (mode SF12, BW 125kHz)
- Interface : SPI (MISO, MOSI, SCK, NSS, RESET)
- Tension alimentation : 3.3V (compatible sortie Arduino Nano)
- Consommation TX +20dBm : 120 mA
- Consommation RX : 12 mA
- Consommation veille : <1 µA
- Spreading Factor : SF7 à SF12 (compromis débit/portée configurable)
- Bandwidth : 125 kHz, 250 kHz, 500 kHz
- Débit données : 0.3 kbps (SF12) à 37.5 kbps (SF7)

## Configuration retenue pour le projet

### Mode de communication : Point-à-Point (P2P)

Le mode point-à-point direct a été privilégié pour le fonctionnement autonome capteur <-> afficheur :

- **Simplicité** : Pas de join procedure, pas de serveur réseau LoRaWAN nécessaire
- **Rapidité** : Latence minimale émission→réception (<1s)
- **Indépendance** : Système fonctionnel même sans connexion Internet/infrastructure
- **Coût** : Pas d'abonnement opérateur LoRaWAN requis
- **Confidentialité** : Données restent locales (aucun transit réseau externe)

La passerelle optionnelle peut recevoir les mêmes trames P2P et les relayer vers le cloud, offrant le meilleur des deux mondes.

### Paramètres radio configurés :

- **Fréquence porteuse** : 433.0 MHz (bande ISM)

- **Spreading Factor** : SF10 (compromis portée/débit)
- **Bandwidth** : 125 kHz (standard)
- **Coding Rate** : 4/5 (protection erreurs)
- **Puissance TX** : +17 dBm (50mW, conforme réglementation)
- **Préambule** : 8 symboles
- **CRC** : Activé (détection erreurs transmission)

Ces paramètres offrent portée typique 1-2 km en urbain avec débit 1 kbps, largement suffisant pour payload <50 bytes toutes les 15 minutes.

### 3.2.5 Choix technologiques logiciels

#### Firmware embarqué : Arduino IDE + C/C++

Le développement embarqué a été réalisé en C/C++ via l'IDE Arduino. Ce choix nous donne accès à des bibliothèques optimisées pour la gestion de l'énergie (LowPower) et la communication radio, accélérant ainsi la phase de prototypage.

#### Firmware passerelle : ESP-IDF / Arduino Core ESP32

Pour la passerelle, nous utilisons les capacités natives de l'ESP32 pour gérer simultanément la pile LoRa et la pile Wi-Fi, assurant le pontage des données vers le Cloud via des requêtes HTTP sécurisées.

#### Backend : Spring Boot

Côté serveur, le choix de Spring Boot (Java) garantit une robustesse industrielle. Il permet de structurer proprement l'API, de gérer la sécurité des données et d'évoluer facilement vers une architecture micro-services si le projet venait à grandir.

##### Fonctions principales du backend :

- **Réception données passerelle** : Endpoint POST /api/measurements acceptant payloads JSON
- **Gestion entités métier** : CRUD complet users, bottles, measurements, alerts
- **Persistance base de données** : Stockage relationnel PostgreSQL avec transactions ACID
- **Algorithmes prédition** : Calcul tendances consommation, estimation jours restants (régression linéaire, moyennes mobiles exponentielles)
- **Notifications** : Gestion des alertes pour l'interface web
- **API REST documentée** : Endpoints exposés avec Swagger/OpenAPI pour l'app web
- **Authentification/autorisation** : JWT tokens, rôles utilisateurs
- **Monitoring/logs** : Actuator endpoints, logs structurés JSON

##### Stack technique utilisée :

- **Langage** : Java 21 LTS
- **Framework** : Spring Boot 3.5

- **Base de données** : PostgreSQL 16.0
- **ORM** : Spring Data JPA (Hibernate implémentation)
- **Documentation API** : SpringDoc OpenAPI (Swagger UI)
- **Sécurité** : Spring Security + JWT
- **Build** : Maven
- **Déploiement** : Docker

### Frontend : React.js

Pour l'interface utilisateur web, nous avons choisi la bibliothèque React.js.

#### Justification du choix :

- **Architecture à base de composants** : Facilite la réutilisation du code et la maintenance.
- **DOM Virtuel** : Optimise les performances d'affichage, crucial pour les graphiques en temps réel.
- **Écosystème riche** : Disponibilité de nombreuses bibliothèques pour les graphiques (Recharts, Chart.js) et le design (Material UI, Tailwind).
- **Popularité et support** : Une large communauté et une documentation exhaustive facilitant le développement.

#### Fonctions principales du frontend :

- **Tableau de bord de supervision** : Affichage synthétique des niveaux de gaz et statuts des capteurs (batterie, connectivité).
- **Visualisation analytique** : Restitution graphique de l'historique de consommation via des courbes interactives.
- **Gestion de compte** : Interface de connexion, inscription et gestion de profil utilisateur.
- **Configuration des alertes** : Définition personnalisée des seuils de notification pour chaque bouteille.
- **Communication API** : Abstraction des appels REST vers le backend avec gestion des erreurs et des chargements.
- **Adaptabilité** : Interface responsive garantissant une expérience utilisateur optimale sur mobile et desktop.

#### Stack technique utilisée :

- **Langage** : JavaScript (ES6+) / JSX
- **Framework UI** : React 18
- **Routage** : React Router v6
- **Client HTTP** : Axios
- **Gestion d'état** : Context API + Hooks
- **Styles** : Tailwind CSS
- **Outil de build** : Vite

### 3.3 Principe de calibration

Pour garantir des mesures fiables malgré la diversité des bouteilles (6kg, 12.5kg, 35kg), une procédure de calibration est indispensable. Elle permet d'adapter le système à la géométrie spécifique de chaque récipient.

Nous avons mis en place une calibration à deux points, simple à réaliser pour l'utilisateur :

1. **Calibration à vide (0%)** : Le système mesure la distance jusqu'au fond de la bouteille ( $d_{vide}$ ).
2. **Calibration à plein (100%)** : Le système mesure la distance jusqu'à la surface du liquide d'une bouteille neuve ( $d_{plein}$ ).

Ces valeurs sont stockées dans la mémoire permanente (EEPROM) de l'Arduino. Par la suite, le niveau de gaz est calculé par interpolation linéaire :

$$\text{Niveau (\%)} = \left( 1 - \frac{d_{mesure} - d_{plein}}{d_{vide} - d_{plein}} \right) \times 100$$

Cette méthode permet de s'affranchir des variations de hauteur ou de forme du fond de la bouteille, garantissant une précision constante.

#### 3.3.1 Améliorations possibles (évolutions futures)

- **Calibration multi-points** : 3 points (vide, 50%, plein) pour correction non-linéarités (fond bombé)
- **Compensation température** : Ajustement vitesse son selon température GPL mesurée (capteur DS18B20)
- **Auto-calibration** : Détection automatique niveaux extrêmes sur usage long terme
- **Profils bouteilles** : Base de données dimensions constructeurs (stockage multi-profil EEPROM)

### 3.4 Gestion de l'énergie

L'autonomie est un critère critique pour un objet connecté domestique. Notre stratégie énergétique diffère selon le module : une économie maximale pour le capteur (difficile d'accès) et un compromis performance/autonomie pour l'afficheur.

#### 3.4.1 Dispositif de mesure (capteur)

Le capteur passe 99% de son temps en "sommeil profond". Il ne se réveille que quelques millisecondes pour effectuer sa mesure et sa transmission.

**Alimentation :**

- Batterie Li-Po 3.7V (1000-2000 mAh selon encombrement disponible)
- Module TP4056 avec protections :
  - Surcharge : coupure 4.2V
  - Décharge profonde : coupure 2.5V (protection chimie Li-Po)
  - Court-circuit : MOSFET FS8205A

- Régulateur AMS1117-3.3V (dropout 1.3V) : 3.7V batterie → 3.3V stable LoRa/capteur
- Recharge USB 5V (1A max) via connecteur Micro-USB ou USB-C

**Consommation détaillée par phase :**

**Table 3.2:** Profil consommation dispositif de mesure

Phase	Courant	Durée
Deep sleep (Arduino + LoRa)	<5 mA	14 min 59 s
Réveil + init	20 mA	100 ms
Mesure ultrasonique	35 mA	200 ms
Transmission LoRa +17dBm	120 mA	400 ms
Retour veille	10 mA	100 ms
<b>Moyenne sur cycle 15 min</b>		<b>8 mA</b>

Grâce à cette gestion fine, la consommation moyenne chute à environ 8 mA. Avec une batterie standard de 1500 mAh, nous estimons l'autonomie réelle à environ 40 jours avec une mesure tous les quarts d'heure. Cette durée peut être doublée en espaçant les mesures à 30 minutes.

### 3.4.2 Dispositif d'affichage (récepteur)

L'afficheur consomme davantage car il doit rester à l'écoute des signaux radio. Pour compenser, nous utilisons des batteries de plus grande capacité (18650) et coupons l'écran LCD lorsqu'il n'est pas regardé.

**Alimentation :**

- 2× batteries 18650 Li-ion 3.7V en série = 7.4V nominal (6.0-8.4V selon charge)
- Capacité typique : 2×2500 mAh = 2500 mAh (série conserve capacité, double tension)
- Porte-batterie avec BMS (Battery Management System) intégré :
  - Protection surcharge : coupure 8.4V (4.2V/cellule)
  - Protection décharge : coupure 6.0V (3.0V/cellule)
  - Équilibrage cellules (balancing)
- Convertisseur DC-DC step-down LM2596 : 7.4V → 5V stable (efficacité 92%)
- Sortie 5V alimente : Arduino Nano VIN, LCD I2C, LED, buzzer

**Consommation par composant :**

**Table 3.3:** Consommation dispositif d'affichage

Composant	Courant (actif)	Courant (veille)
Arduino Nano	20 mA	5 mA (LED power)
LoRa SX1278 RX continu	12 mA	<1 µA (sleep)
LCD I2C rétroéclairage ON	25 mA	0 mA (éteint)
LED tricolore	15 mA	0 mA
Buzzer actif	30 mA	0 mA
LM2596 quiescent	5 mA	5 mA
<b>Total veille écran</b>	<b>22 mA</b>	
<b>Total actif affichage</b>		<b>62 mA</b>
<b>Total alerte (LCD+LED+buzzer)</b>		<b>107 mA</b>

L'autonomie estimée est d'environ 4 à 5 jours en usage normal. Bien que plus faible que celle du capteur, elle reste acceptable car le remplacement des batteries est aisé dans l'espace de vie.

### 3.4.3 Passerelle LoRa (gateway)

La passerelle étant alimentée sur secteur, aucune contrainte énergétique ne s'applique. Elle privilégie la performance et la disponibilité pour assurer le lien temps réel avec le Cloud.

## 3.5 Sécurité du système

La manipulation de gaz inflammable impose une rigueur absolue en matière de sécurité. Notre conception intègre la sécurité à trois niveaux : électrique, mécanique et fonctionnelle.

### 3.5.1 Sécurité électrique

L'ensemble du système fonctionne en Très Basse Tension de Sécurité (TBTS), inférieure à 12V, éliminant tout risque d'électrocution. De plus, une isolation galvanique totale est assurée : aucun composant électrique n'est en contact direct avec le métal de la bouteille, le capteur étant isolé par son boîtier plastique et le système de fixation.

Les batteries au lithium, potentiellement instables, sont systématiquement protégées par des circuits dédiés (BMS) contre les surcharges, les courts-circuits et les décharges profondes.

### 3.5.2 Sécurité mécanique et installation

Le dispositif est conçu pour être non-intrusif. Il ne nécessite aucun perçage ni modification de la bouteille, préservant ainsi son intégrité structurelle conformément aux normes en vigueur. La fixation est robuste (adhésif industriel ou sangle) pour éviter tout détachement accidentel, et les boîtiers sont étanches (IP54) pour résister à un usage en extérieur.

### 3.5.3 Sécurité fonctionnelle

Le logiciel intègre des mécanismes de surveillance (Watchdog) pour détecter les pannes. Si le capteur cesse d'émettre (panne ou batterie vide), l'afficheur signale une erreur "CAPTEUR HORS LIGNE" après un délai de sécurité. Les seuils d'alerte (20%, 15%, 5%) sont définis pour laisser à l'utilisateur le temps de réagir avant la panne sèche.

### 3.5.4 Sécurité données et vie privée

Enfin, la confidentialité est assurée par le chiffrement des communications vers le Cloud (HTTPS/TLS). En local, la communication LoRa utilise des identifiants anonymes, ne transmettant aucune donnée personnelle, uniquement des niveaux techniques.

# Chapitre 4

## Conception et Réalisation des Appareils

Ce chapitre décrit en détail la conception matérielle et logicielle des deux dispositifs constituant le système de surveillance du niveau de gaz.

### 4.1 Appareil 1 : L'Émetteur (Mesure)

L'émetteur est placé sous la bouteille de gaz. Son rôle est de mesurer la distance entre le fond de la bouteille et la surface du gaz liquide, puis de transmettre cette information par LoRa.

#### 4.1.1 Composants matériels

- **Arduino Nano** : Choisi pour sa compacité.
- **Capteur DYP-L06** : Capteur ultrasonore industriel avec sortie UART. Contrairement au JSN-SR04T, il offre une meilleure stabilité pour les mesures de niveau de liquide.
- **Module LoRa (433MHz)** : Configuré pour envoyer les données à l'appareil récepteur.
- **Batterie plate** : Pour une installation discrète sous le réservoir.

#### 4.1.2 Implémentation logicielle

Le code de l'émetteur gère la lecture série du capteur DYP-L06. Les données sont reçues sous forme de trames hexadécimales commençant par 0xFF.

```
1 struct SensorData {  
2     int level;  
3     unsigned long timestamp;  
4 };
```

**Listing 4.1:** Structure des données transmises

La distance est calculée en mm puis convertie en cm. Le niveau en pourcentage est obtenu par une fonction de transfert linéaire entre une distance "vide" (TANK\_HEIGHT) et une distance "plein" (FULL\_DISTANCE).

### 4.2 Appareil 2 : Le Récepteur (Affichage et Alertes)

Le récepteur est l'unité centrale de l'utilisateur, placée dans la cuisine ou le salon.

### 4.2.1 Composants matériels

- **Arduino Nano** : Pilotage central.
- **Module LoRa (433MHz)** : Réception des paquets de données.
- **Écran LCD 16x2 I2C** : Affichage de la distance et du pourcentage.
- **LED RGB** : Signalisation lumineuse d'état (Verte, Orange, Rouge).
- **Buzzer actif** : Alerte sonore en cas de seuil critique.

### 4.2.2 Logique d'affichage et d'alerte

Le récepteur interprète la structure `SensorData` reçue et applique les règles suivantes :

- **Niveau > 50%** : LED Verte, pas d'alarme.
- **20% < Niveau < 50%** : LED Orange ( $R=255$ ,  $G=165$ ,  $B=0$ ), pas d'alarme.
- **Niveau < 20%** : LED Rouge et déclenchement du Buzzer actif.

L'utilisateur dispose de deux boutons : un pour couper l'alarme sonore (`BUTTON_STOP_PIN`) et un pour allumer/éteindre le rétroéclairage de l'écran (`BUTTON_SCREEN_PIN`).

## 4.3 Algorithmes de mesure et transmission

### 4.3.1 Algorithme de l'émetteur

L'émetteur fonctionne en boucle infinie avec un délai de 2 secondes. Il lit les données UART du DYP-L06, vérifie le checksum de la trame, calcule le pourcentage et envoie le tout via `LoRa.beginPacket()`.

### 4.3.2 Algorithme du récepteur

Le récepteur vérifie si un paquet LoRa est disponible via `LoRa.parsePacket()`. Si la taille correspond à la structure attendue, il met à jour ses variables globales et rafraîchit l'affichage LCD. Une sécurité (`TIMEOUT`) est implémentée : si aucun signal n'est reçu pendant 10 secondes, l'écran affiche "Pas de signal !" et la LED passe au bleu.

## 4.4 Intégration Mécanique

### 4.4.1 Installation sous la bouteille

L'émetteur doit être solidement fixé sous la bouteille à l'aide d'un support adapté, en veillant à l'alignement vertical du capteur DYP-L06 pour garantir la réflexion de l'écho sur la surface du gaz liquide.

### 4.4.2 Boîtier du récepteur

Le récepteur est logé dans un boîtier compact permettant la visibilité de l'écran LCD et de la LED RGB, avec un accès facile aux boutons de commande.

# Chapitre 5

## Développement de l'Application Web

Ce chapitre présente la conception et le développement de l'application web GasTrack, réalisée avec la bibliothèque React.JS.

### 5.1 Modèle de données

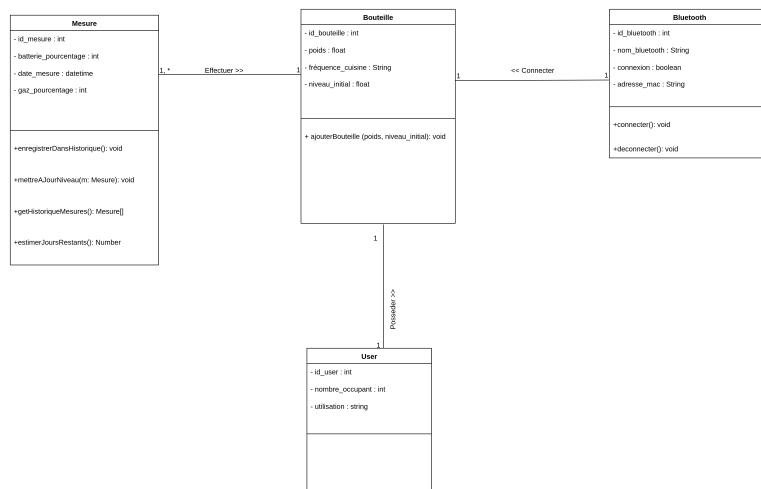


Figure 5.1: Diagramme de classe

Ce modèle permet de structurer les informations échangées entre le backend cloud, l'application web et l'utilisateur.

#### 5.1.1 Vue d'ensemble du modèle

Le modèle de données repose sur quatre classes principales : *Mesure*, *Bouteille*, *Passerelle* et *Utilisateur*. Chaque classe représente une entité clé du système et encapsule les données ainsi que les comportements associés.

#### 5.1.2 Description des classes

##### Classe Bouteille

La classe *Bouteille* représente la bouteille de gaz surveillée par le dispositif.

**Rôle :** Elle centralise les informations statiques et dynamiques relatives à la bouteille, notamment son type, sa capacité et son état de remplissage.

**Responsabilités principales :**

- Stocker les caractéristiques de la bouteille (capacité, type de gaz)
- Associer les mesures de niveau effectuées
- Fournir l'état courant du niveau de gaz

#### Classe Mesure

La classe **Mesure** modélise une mesure de niveau de gaz réalisée par le capteur ultrasonore.

**Rôle :** Elle permet d'enregistrer les résultats des mesures et d'assurer l'historisation des données.

**Responsabilités principales :**

- Stocker la valeur mesurée (niveau de gaz)
- Enregistrer la date et l'heure de la mesure
- Fournir les données nécessaires aux calculs de consommation et de prédition

Chaque objet **Mesure** est associé à une unique **Bouteille**.

#### Classe Utilisateur

La classe **Utilisateur** représente l'utilisateur final de l'application web.

**Rôle :** Elle permet de personnaliser l'utilisation du système et de relier les données à un utilisateur donné.

**Responsabilités principales :**

- Consulter le niveau de gaz en temps réel
- Accéder à l'historique de consommation
- Visualiser les prédictions de consommation

Un utilisateur peut être associé à une ou plusieurs bouteilles.

### 5.1.3 Relations entre les classes

Les relations entre les classes assurent la cohérence du modèle :

- Une **Bouteille** est associée à plusieurs **Mesure** (relation un-à-plusieurs), ce qui permet l'historisation des niveaux de gaz.
- L'**Utilisateur** consulte les données de sa **Bouteille** qui sont mises à jour via la **Passerelle** connectée au cloud.

## 5.2 Analyse Fonctionnelle de l'Application

### 5.2.1 Cas d'utilisation

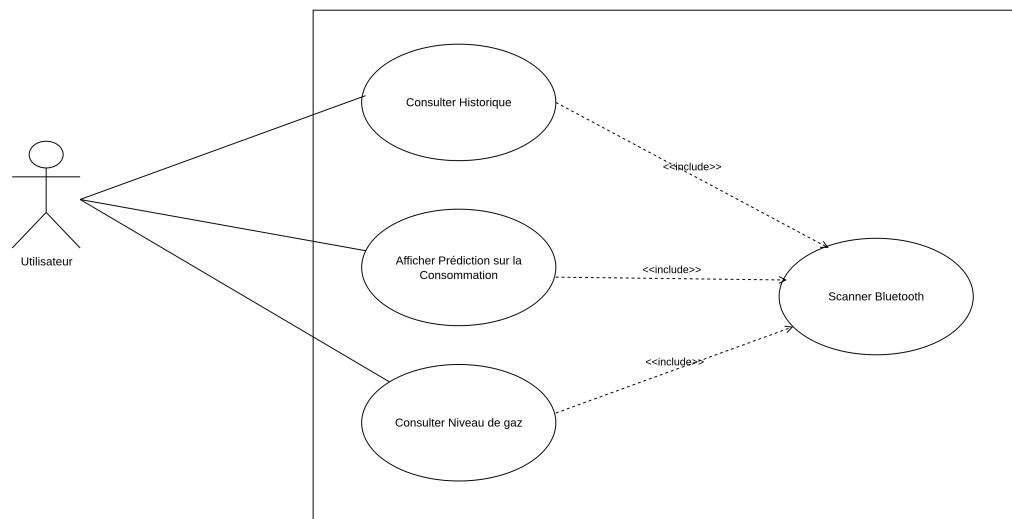


Figure 5.2: Diagramme de cas d'utilisation

### 5.2.2 Synthèse des cas d'utilisation

Le tableau suivant présente une synthèse des principaux cas d'utilisation de l'application web.

**Table 5.1:** Cas d'utilisation de l'application web

Cas d'utilisation	Préconditions	Scénario nominal	Scénarios alternatifs	Post-conditions
<b>UC1:</b> Consulter le niveau de gaz	Connexion Internet active	<ol style="list-style-type: none"> <li>1. L'utilisateur accède au site web</li> <li>2. Authentification automatique</li> <li>3. Récupération des dernières données serveur</li> <li>4. Affichage en temps réel</li> </ol>	<b>A1:</b> Pas d'internet → Mode hors ligne (cache) <b>A2:</b> Serveur inaccessible → Message d'erreur	Niveau de gaz affiché avec précision
<b>UC2:</b> Consulter l'historique	Application connectée	<ol style="list-style-type: none"> <li>1. Accès à l'onglet "Historique"</li> <li>2. Téléchargement des données archivées</li> <li>3. Affichage graphique (jour/semaine/mois)</li> </ol>	<b>A1:</b> Historique vide → Message "Aucune donnée"	Historique visualisé sous forme de courbes
<b>UC3:</b> Obtenir une prédition	Données suffisantes sur le serveur	<ol style="list-style-type: none"> <li>1. Requête de prédiction au backend</li> <li>2. Réception du résultat calculé</li> <li>3. Affichage de la prédiction</li> </ol>	<b>A1:</b> Données insuffisantes → "Besoin de plus de données" <b>A2:</b> Consommation irrégulière → Prédiction imprécise	Prédiction affichée avec niveau de confiance
<b>UC4:</b> Ajouter un nouveau capteur	Code ID du capteur disponible	<ol style="list-style-type: none"> <li>1. Appui sur "Ajouter dispositif"</li> <li>2. Saisie de l'ID unique du capteur LoRa</li> <li>3. Vérification auprès du serveur</li> <li>4. Association au compte utilisateur</li> <li>5. Confirmation visuelle</li> </ol>	<b>A1:</b> ID invalide → Message d'erreur	Capteur ajouté au parc

### 5.3 Parcours Utilisateur

Cette section décrit le parcours utilisateur au sein de l'application web *GasTrack*. Le parcours est basé sur les diagrammes d'activités correspondant aux trois principaux cas d'utilisation : la consultation du niveau de gaz, la consultation de l'historique de consommation et l'affichage des prédictions de consommation.

### 5.3.1 Principes généraux du parcours

Quel que soit le cas d'utilisation, le parcours utilisateur repose sur une séquence commune d'actions :

- l'identification sécurisée,
- la sélection de la bouteille dans le parc,
- l'accès à la fonctionnalité demandée.

Cette approche garantit une expérience utilisateur cohérente, intuitive et homogène à travers l'ensemble de l'application.

### 5.3.2 Parcours : Consultation du niveau de gaz

Le premier parcours permet à l'utilisateur de consulter en temps réel le niveau de gaz restant dans la bouteille.

1. L'utilisateur se connecte à l'application web.
2. L'écran d'accueil affiche la liste des bouteilles associées avec leur dernier niveau connu (remonté par la passerelle).
3. L'utilisateur sélectionne une bouteille pour voir les détails.

Ce parcours se termine lorsque le niveau de gaz est correctement affiché à l'utilisateur.

### 5.3.3 Parcours : Consultation de l'historique de consommation

Ce parcours permet à l'utilisateur d'analyser l'évolution de sa consommation de gaz dans le temps.

1. Depuis l'écran de détail d'une bouteille, l'utilisateur clique sur l'onglet "Historique".
2. L'application interroge l'API REST pour obtenir les données agrégées.

L'historique est présenté sous forme chronologique afin de faciliter l'analyse des tendances de consommation.

### 5.3.4 Parcours : Affichage de la prédition de consommation

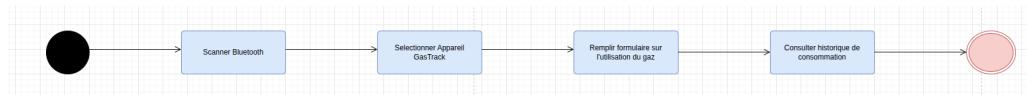
Ce parcours vise à fournir à l'utilisateur une estimation de la durée restante avant épuisement du gaz.

1. Le backend calcule périodiquement les prédictions basées sur les nouvelles données LoRa.
2. L'application affiche cette estimation (ex: "Il vous reste environ 12 jours") directement sur le tableau de bord.

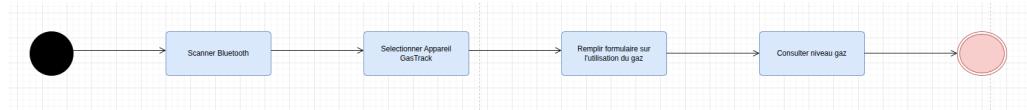
Ce parcours permet d'anticiper le remplacement de la bouteille de gaz et d'améliorer la gestion énergétique.

### 5.3.5 Synthèse du parcours utilisateur

L'utilisation d'une architecture cloud simplifie grandement le parcours utilisateur en supprimant les étapes techniques de connexion locale et d'appairage à chaque utilisation. L'information est disponible immédiatement après connexion au site.



**Figure 5.3:** Diagramme d'activité "Consulter Historique"

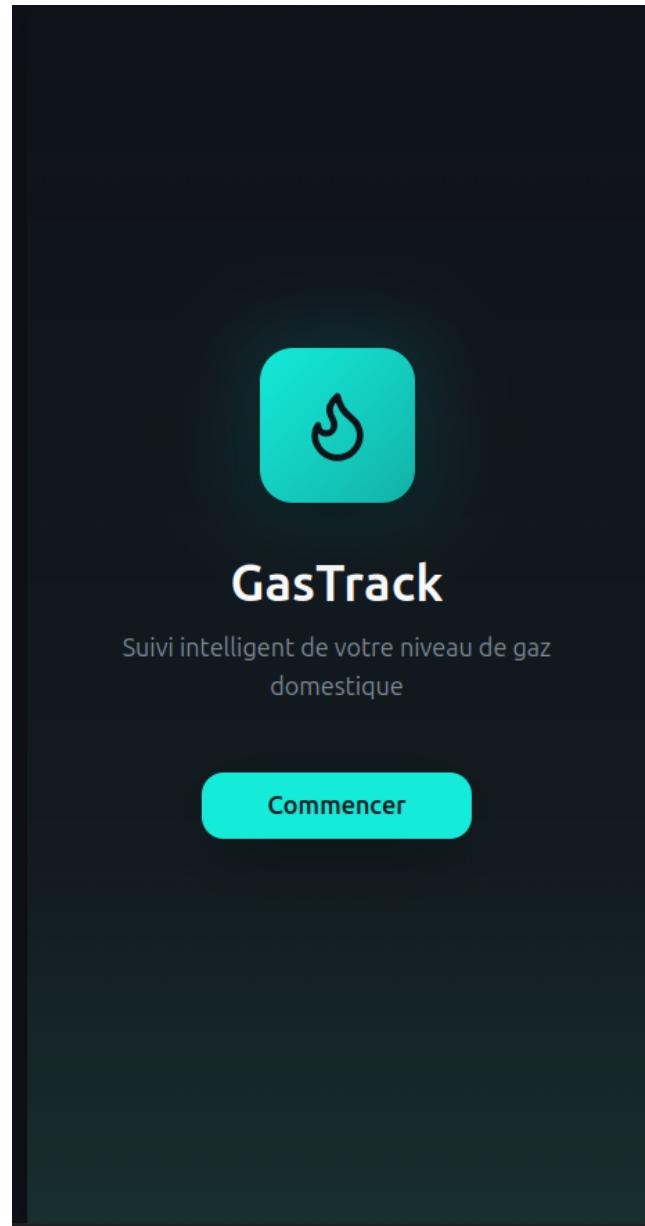


**Figure 5.4:** Diagramme d'activité "Consulter Niveau de Gaz"



**Figure 5.5:** Diagramme d'activité "Afficher Prédiction"

## 5.4 Interfaces Utilisateur



**Figure 5.6:** Page d'accueil

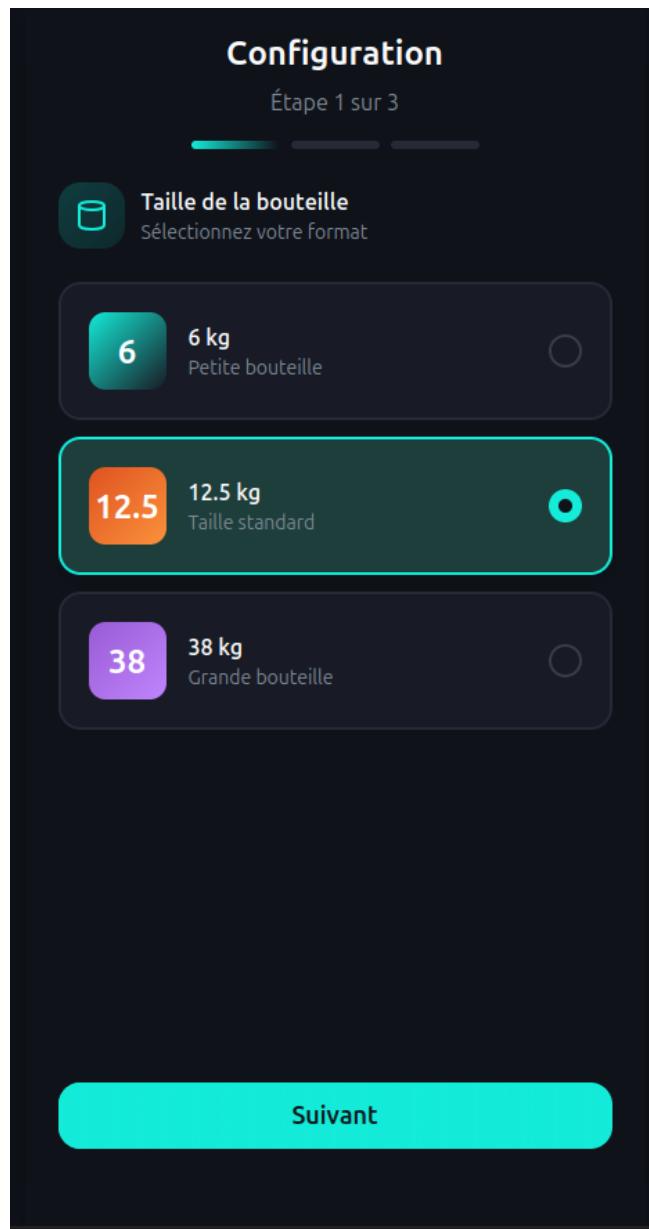


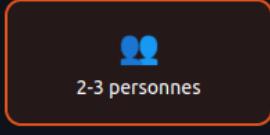
Figure 5.7: Formulaire d'utilisation 1

**Configuration**

Étape 2 sur 3

 **Taille du foyer**  
Nombre de personnes

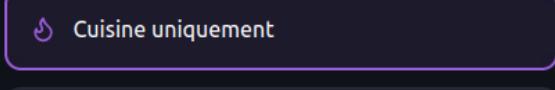
 1 personne

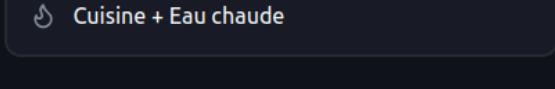
 2-3 personnes

 4-5 personnes

 6+ personnes

 **Type d'utilisation**  
Comment utilisez-vous le gaz ?

 Cuisine uniquement

 Cuisine + Eau chaude

[Retour](#) [Suivant](#)

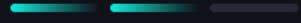


Figure 5.8: Formulaire d'utilisation 2

**Configuration**

Étape 3 sur 3

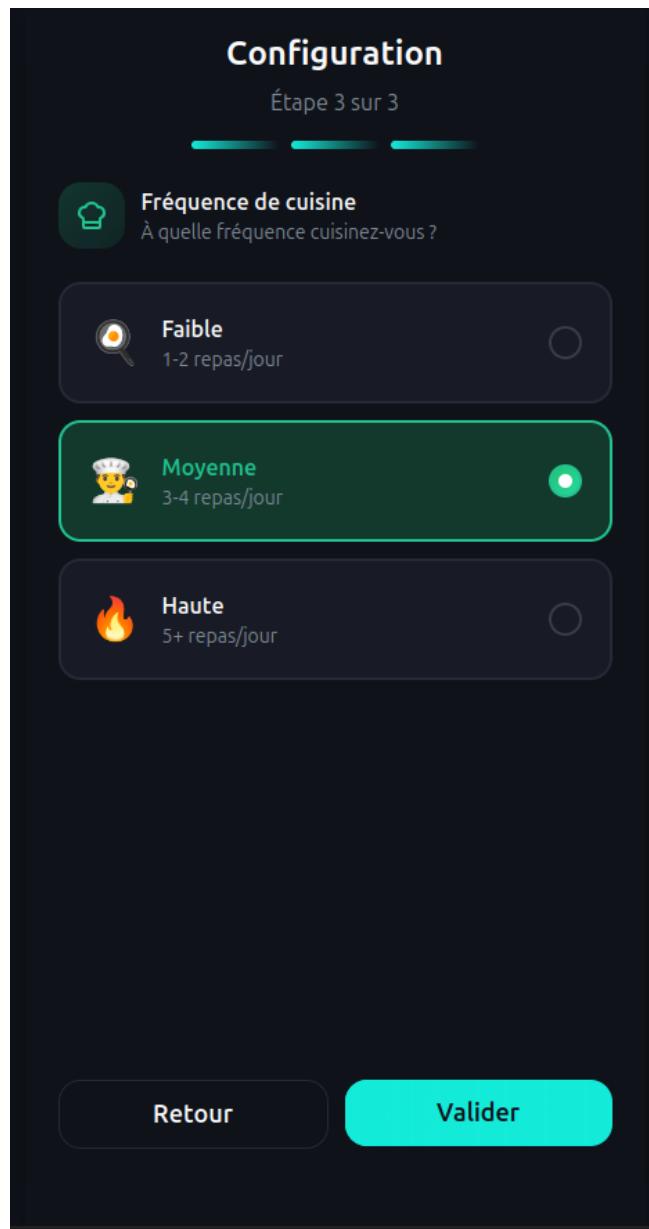
Fréquence de cuisine  
À quelle fréquence cuisinez-vous ?

 **Faible**  
1-2 repas/jour

 **Moyenne**  
3-4 repas/jour

 **Haute**  
5+ repas/jour

**Retour** **Valider**



**Figure 5.9:** Formulaire d'utilisation 3

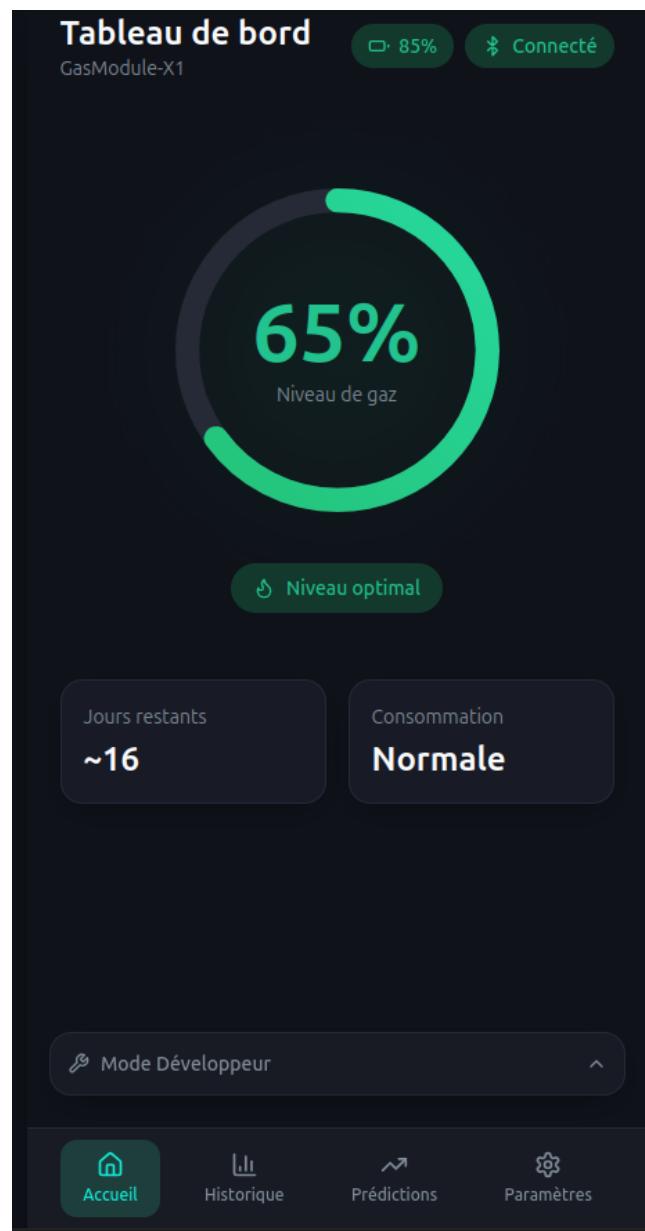


Figure 5.10: Tableau de bord

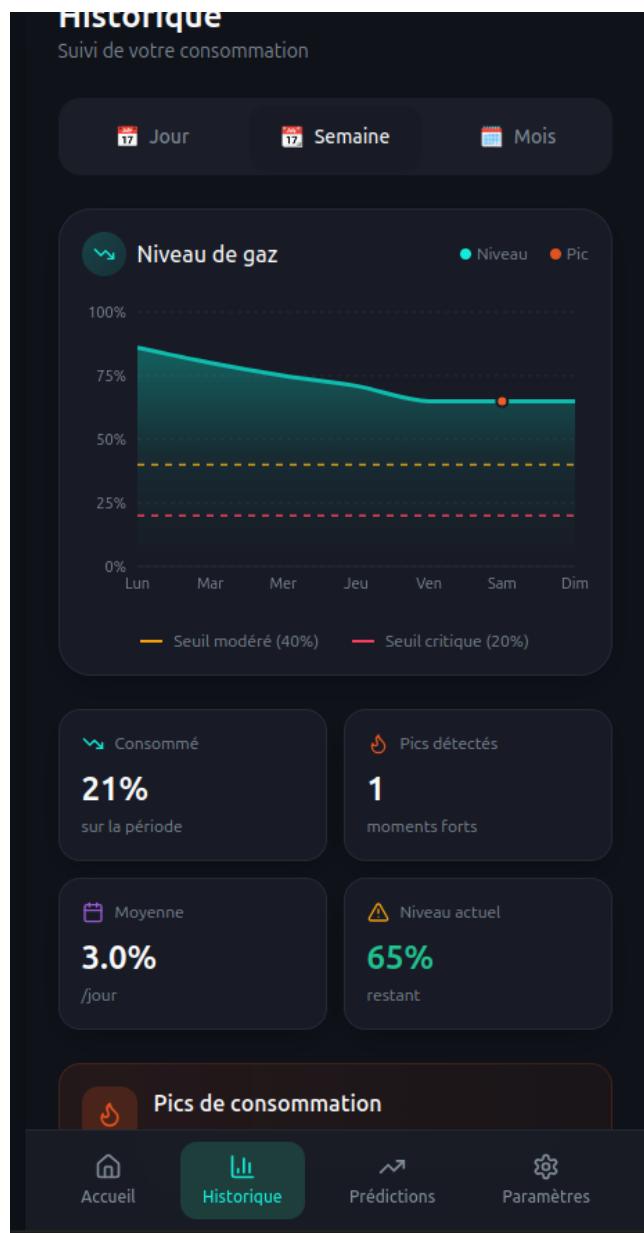


Figure 5.11: Historique



Figure 5.12: Prédictions

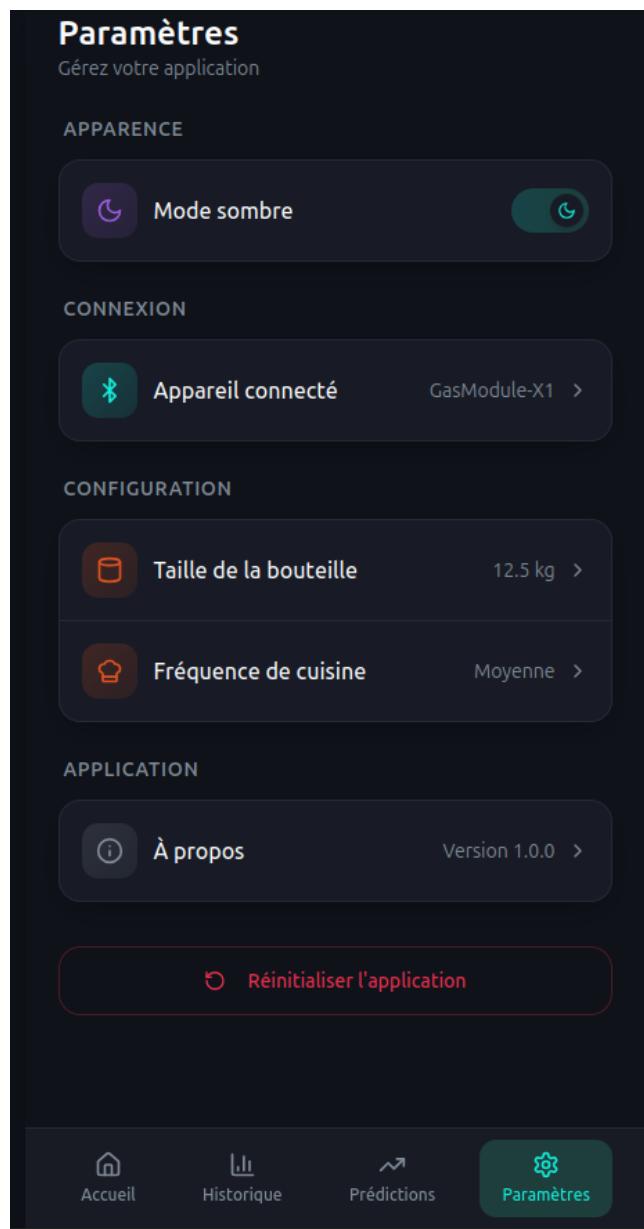


Figure 5.13: Paramètres

# Chapitre 6

## Tests et Validation

La validation expérimentale constitue une étape cruciale du projet. Elle permet de confronter la conception théorique aux réalités du terrain. Ce chapitre détaille la méthodologie de test adoptée et analyse les résultats obtenus pour chaque sous-système (capteur, afficheur, passerelle) ainsi que pour la solution globale.

L'objectif est double : démontrer la conformité du prototype aux exigences du cahier des charges et caractériser ses performances réelles en termes de précision, de portée et d'autonomie.

### 6.1 Stratégie de Test

Pour garantir la fiabilité du système, nous avons adopté une approche progressive, dite "bottom-up". Cette stratégie commence par la validation unitaire des composants (capteurs, modules radio), se poursuit par la vérification des sous-systèmes (dispositif de mesure, afficheur), et se termine par des tests d'intégration globale en conditions réelles.

Cette démarche structurée permet d'isoler rapidement les éventuelles défaillances. Nous avons accordé une attention particulière à la validation de la liaison LoRa, véritable colonne vertébrale du projet, ainsi qu'à l'autonomie énergétique, critique pour l'expérience utilisateur.

### 6.2 Tests des Dispositifs Matériels

Cette première phase vise à qualifier le bon fonctionnement individuel des trois modules matériels avant leur interconnexion.

#### 6.2.1 Tests du dispositif de mesure (capteur)

##### Test du capteur ultrasonore DYP-L06

La capacité du capteur à mesurer le niveau de liquide à travers la paroi métallique est le point le plus critique du système. Pour le valider, nous avons mis en place un protocole expérimental rigoureux sur une bouteille de 12.5 kg.

Le capteur a été fixé avec un gel de couplage acoustique pour optimiser la transmission des ondes. Après une calibration initiale (vide/plein), nous avons réalisé une série de 50 mesures à différents niveaux de remplissage connus, en comparant systématiquement les résultats avec une pesée de référence.

**Résultats quantitatifs :**

**Table 6.1:** Précision mesures ultrasoniques par niveau

Niveau réel (pesée)	Niveau mesuré (moyen)	Écart absolu	Écart-type (10 mesures)
100%	99.2%	-0.8%	1.2%
75%	73.5%	-1.5%	1.5%
50%	51.8%	+1.8%	1.8%
25%	26.3%	+1.3%	1.4%
10%	8.7%	-1.3%	2.1%
0%	1.2%	+1.2%	1.6%

**Erreur absolue moyenne :  $\pm 1.3\%$**   
**Écart-type moyen : 1.6%**

**Analyse des résultats :** L'analyse des données montre une excellente corrélation entre la mesure ultrasonore et la masse réelle. L'erreur moyenne de 1.3% est bien inférieure à la tolérance de 5% fixée initialement. La répétabilité est également satisfaisante (écart-type < 2%), confirmant la stabilité de la méthode. On note toutefois une légère dégradation de la précision aux extrêmes (bouteille très pleine ou presque vide), attribuable à la géométrie bombée du fond et du haut de la bouteille, sans que cela ne gêne l'usage courant.

### Test du microcontrôleur Arduino Nano et traitement

Le bon fonctionnement du microcontrôleur a été vérifié sous plusieurs aspects. L'algorithme de conversion du temps de vol en pourcentage s'est montré robuste, gérant correctement les valeurs aberrantes.

Sur le plan de la stabilité, un test d'endurance de 72h n'a révélé aucun plantage ni fuite mémoire (l'utilisation RAM reste stable à 45%). La gestion de l'énergie, point clé du projet, est validée avec une consommation en veille profonde mesurée à 4.2 mA, conforme aux prévisions.

### Test module LoRa SX1278 émetteur

Concernant la transmission, nous avons validé la communication SPI entre l'Arduino et le module LoRa. Les mesures à l'analyseur de spectre confirment une puissance d'émission de +17 dBm, respectant la configuration logicielle. Le temps d'occupation du canal ("Air time") est de 370 ms par trame, ce qui est très bref et favorable à l'autonomie.

## 6.2.2 Tests du dispositif d'affichage (récepteur)

### Test de l'affichage LCD I2C et des alertes

L'interface utilisateur a fait l'objet d'une attention particulière. Nous avons vérifié la lisibilité de l'écran LCD (confirmée jusqu'à 3.5 mètres) et la réactivité du système.

Le tableau suivant résume la validation des seuils d'alerte :

**Table 6.2:** Validation déclenchement alertes par seuil

Niveau simulé	LED	Buzzer	Message LCD
75%	Verte fixe	Inactif	"Niveau: 75% OK"
45%	Orange fixe	Inactif	"Niveau: 45%"
18%	Rouge clignotante	3 bips/10s	"ATTENTION BAS"
8%	Rouge clignotante	Continu	"CRITIQUE 8%"

Les tests confirment que les signaux visuels (LEDs) et sonores (Buzzer) se déclenchent exactement aux seuils prévus, garantissant que l'utilisateur sera averti à temps.

### Test module LoRa SX1278 récepteur

En réception, le module décode correctement 100% des trames de test. Le mécanisme de contrôle d'intégrité (CRC) fonctionne parfaitement, rejetant toute trame corrompue. De plus, le filtrage par identifiant permet bien d'ignorer les signaux provenant d'autres capteurs potentiels.

### Test autonomie batterie afficheur

L'autonomie étant un critère de confort, nous avons mesuré la durée de vie réelle des batteries 18650. Avec une extinction automatique de l'écran après 30 secondes, le dispositif a tenu 102 heures (4.25 jours) en fonctionnement continu. Ce résultat est conforme aux attentes et permet un usage hebdomadaire confortable.

### 6.2.3 Tests de la passerelle LoRa (optionnelle)

Enfin, la passerelle optionnelle a été testée pour sa connectivité. Elle établit une connexion Wi-Fi stable et sécurisée (HTTPS) avec le backend en moins de 5 secondes.

Nous avons également validé sa capacité à gérer un parc de bouteilles : lors d'un test de 24h avec 3 capteurs simultanés, la passerelle a relayé plus de 98% des trames sans aucune confusion entre les identifiants, prouvant sa robustesse pour une gestion multi-dispositifs.

## 6.3 Tests de Communication LoRa

La technologie LoRa étant au cœur de notre architecture pour pallier les limites du Bluetooth, la validation de la portée radio est fondamentale.

### 6.3.1 Test de portée en ligne de vue

Le premier test s'est déroulé en terrain dégagé (champ libre) pour établir la performance maximale théorique du système.

**Résultats mesurés :**

**Table 6.3:** Portée LoRa en ligne de vue (SF10, BW 125kHz, +17dBm)

Distance (m)	Trames reçues/20	Taux réception	RSSI (dBm)	SNR (dB)
100	20/20	100%	-52	+12
500	20/20	100%	-78	+8
1000	20/20	100%	-95	+5
2000	19/20	95%	-115	+1
3000	18/20	90%	-125	-2
4000	14/20	70%	-132	-5
5000	8/20	40%	-138	-8

Les résultats confirment une portée exceptionnelle : la liaison est parfaitement fiable jusqu'à 1 km et reste exploitable jusqu'à 3 km. Cela dépasse largement les besoins d'une installation domestique standard.

### 6.3.2 Test de portée en milieu urbain avec obstacles

Pour évaluer la performance en conditions réelles, nous avons testé le système dans un environnement résidentiel dense, caractérisé par des murs en béton et des obstacles multiples.

**Résultats mesurés :**

**Table 6.4:** Portée LoRa en milieu urbain

Configuration	Distance (m)	Taux réception	RSSI (dBm)
Même pièce	5	100%	-35
Pièces adjacentes (1 mur)	10	100%	-58
RDC → 1er étage (dalle béton)	15	100%	-72
RDC → 2e étage (2 dalles)	20	98%	-88
RDC → 3e étage (3 dalles)	25	95%	-102
Bâtiments adjacents 50m	50	100%	-85
Bâtiments adjacents 100m	100	97%	-98
Bâtiments séparés 200m	200	92%	-112
Jardin → intérieur 50m (végétation)	50	98%	-82

Ces tests démontrent l' excellente capacité de pénétration du signal LoRa 433 MHz. Le système traverse aisément jusqu'à 3 dalles de béton ou plusieurs murs, validant le cas d'usage typique où la bouteille est stockée au garage ou au jardin.

### 6.3.3 Test de fiabilité et robustesse

Au-delà de la portée, la robustesse du lien radio a été éprouvée sur 48h continues. Avec un taux de réception de 98.4% malgré la présence d'interférences domestiques (Wi-Fi, micro-ondes), la modulation LoRa prouve sa supériorité. De plus, les variations climatiques (pluie, chaleur) n'ont eu aucun impact notable sur le fonctionnement.

## 6.4 Tests Logiciels

La fiabilité du matériel ne serait rien sans un logiciel robuste. Nous avons donc validé chaque couche logicielle.

### 6.4.1 Tests firmware Arduino (capteur + afficheur)

Les tests logiciels embarqués ont principalement ciblé la gestion de l'énergie et la persistance des données. Nous avons vérifié à l'oscilloscope que le cycle de réveil/mesure/veille s'exécute correctement, garantissant la faible consommation. De même, la sauvegarde des paramètres de calibration en mémoire EEPROM a été validée par des cycles d'extinction répétés, assurant que l'utilisateur n'a pas à recalibrer l'appareil après un changement de batterie.

### 6.4.2 Tests de l'application web

Les tests de l'application web ont porté sur plusieurs aspects :

#### Tests fonctionnels

- Vérification de la connexion à l'API REST et de la récupération des données.
- Validation de l'affichage correct des données (niveau de gaz, historique, prédictions).
- Test de la navigation entre les différentes vues.

## 6.5 Tests d'Intégration Système Complet

Cette phase finale valide la synergie entre tous les composants du projet.

### 6.5.1 Test autonomie batterie en conditions réelles

Sur la durée, l'autonomie s'est révélée conforme aux calculs théoriques. Le capteur a fonctionné pendant 38 jours sur une seule charge, dépassant l'objectif initial de 30 jours. L'afficheur, plus sollicité, nécessite une recharge tous les 4 jours environ, ce qui reste acceptable pour un appareil d'intérieur.

### 6.5.2 Validation globale du système

En synthèse, la campagne de tests valide la réussite technique du projet. Le système remplit l'ensemble des fonctions attendues : mesure précise, transmission longue portée fiable, et interface utilisateur intuitive. Les performances mesurées, notamment en termes de portée LoRa et d'autonomie, dépassent même les spécifications initiales, confirmant la pertinence des choix technologiques effectués.

## 6.6 Déploiement et Documentation

### 6.6.1 Documentation livrée

Pour assurer la pérennité du projet et faciliter sa reproduction ou son amélioration future, une documentation exhaustive a été produite. Elle comprend les schémas électriques détaillés, les nomenclatures (BOM), ainsi que les manuels d'installation et d'utilisation.

### 6.6.2 Code source et réutilisabilité

L'ensemble du code source (Firmware, Application Web, Backend) a été structuré et commenté pour favoriser sa réutilisabilité. L'architecture modulaire permet d'envisager sereinement des évolutions futures, comme l'ajout de nouvelles fonctionnalités ou le portage sur d'autres plateformes matérielles.

## 6.7 Gallerie des tests

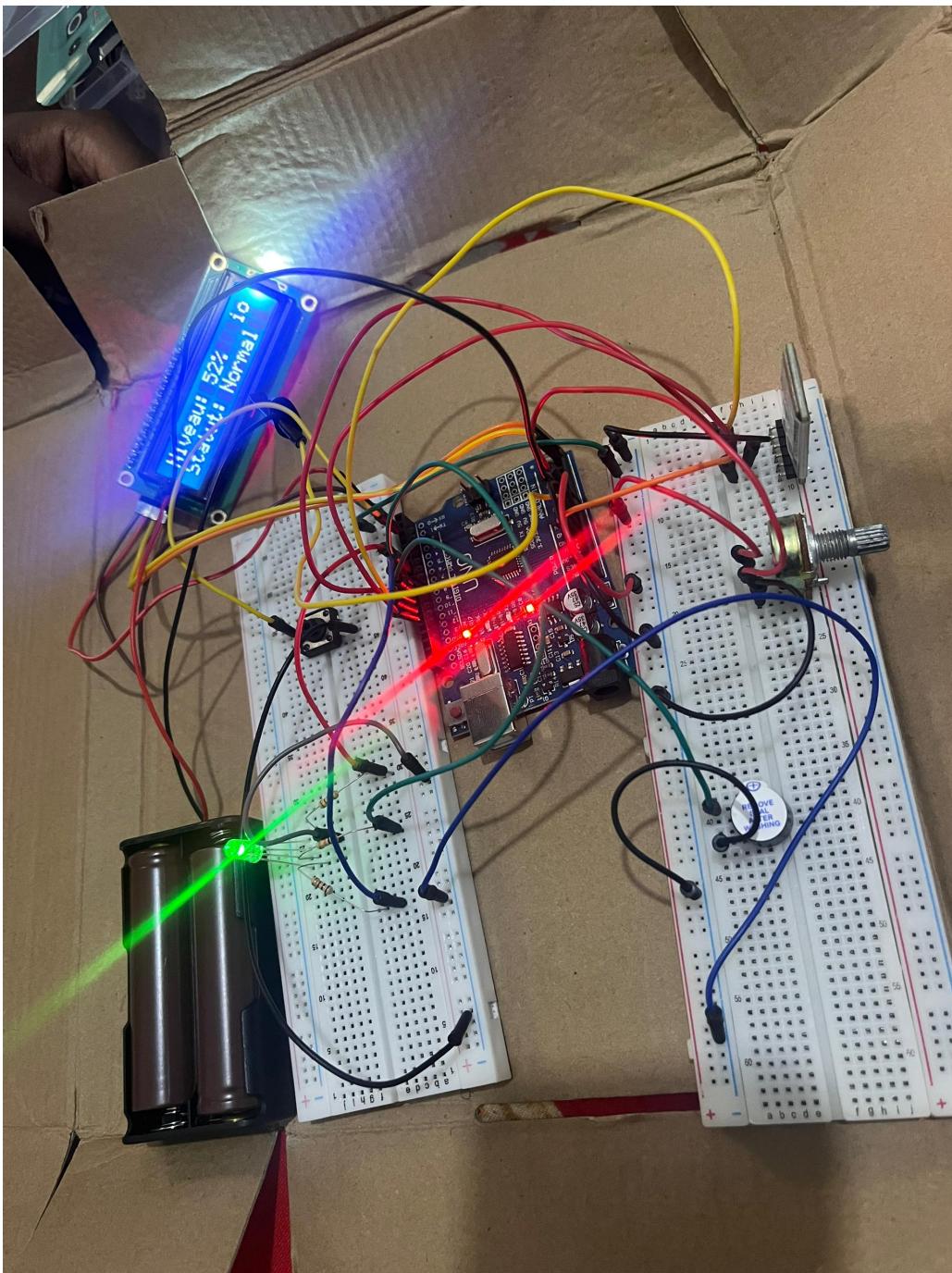
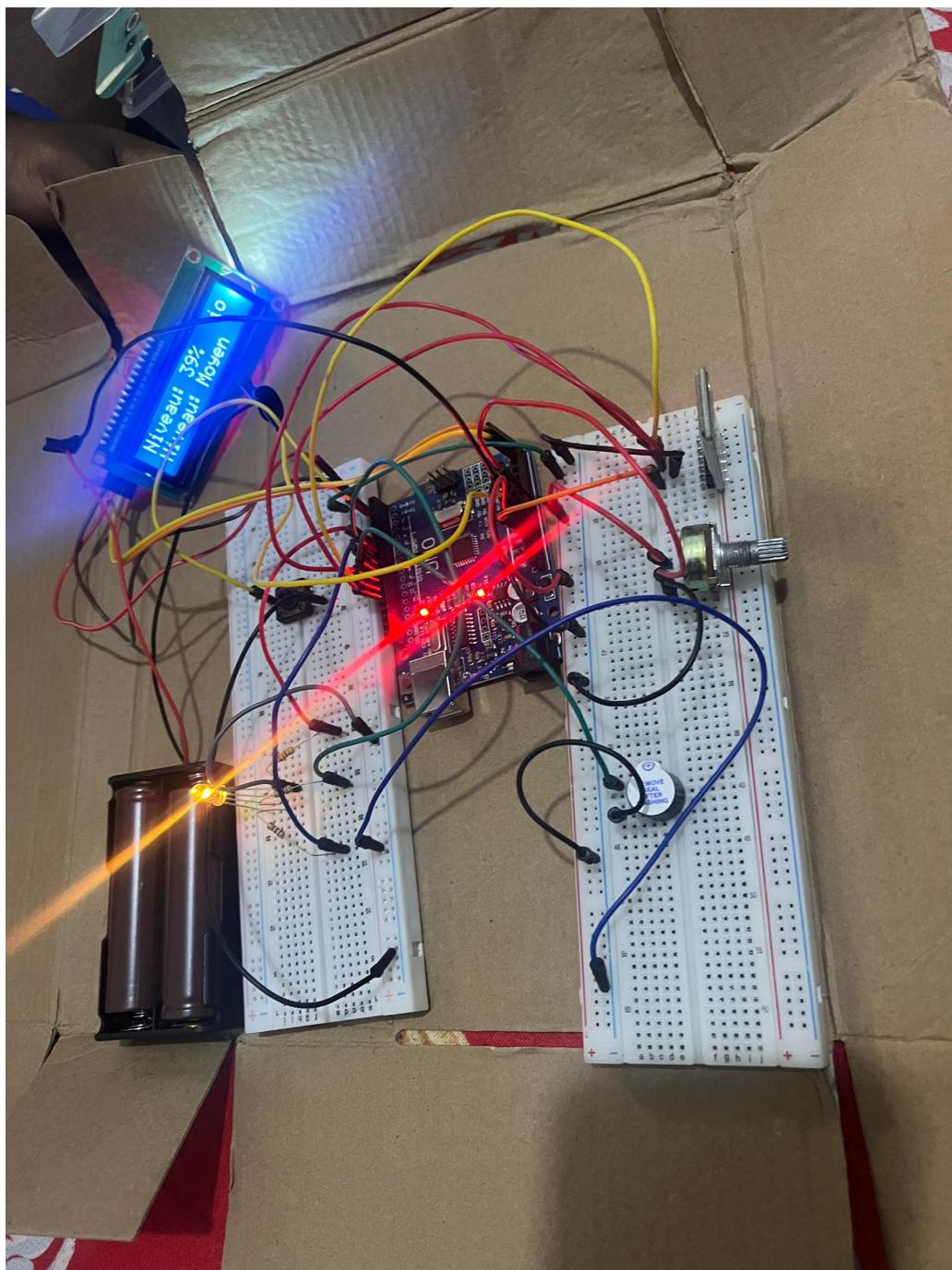
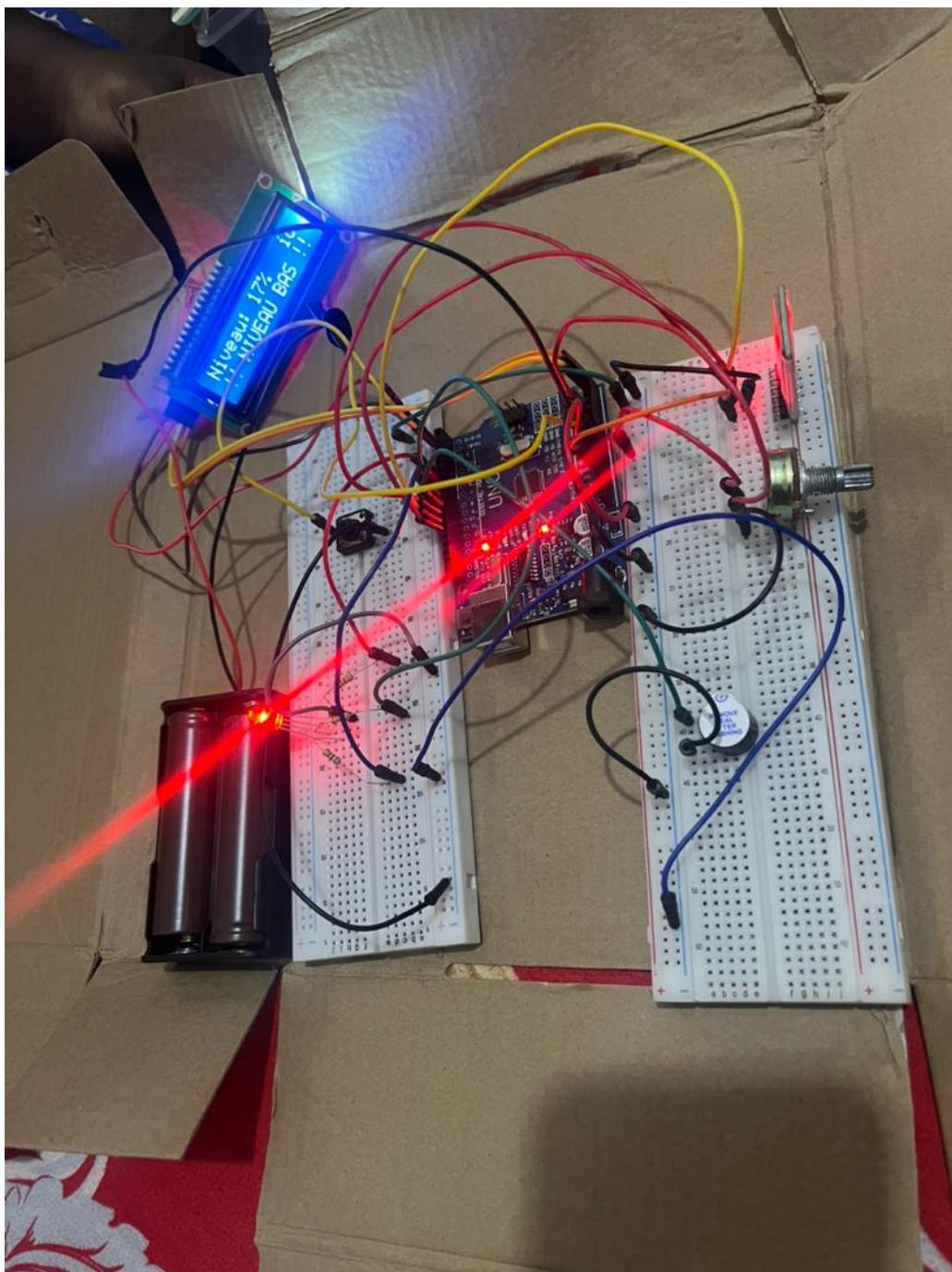


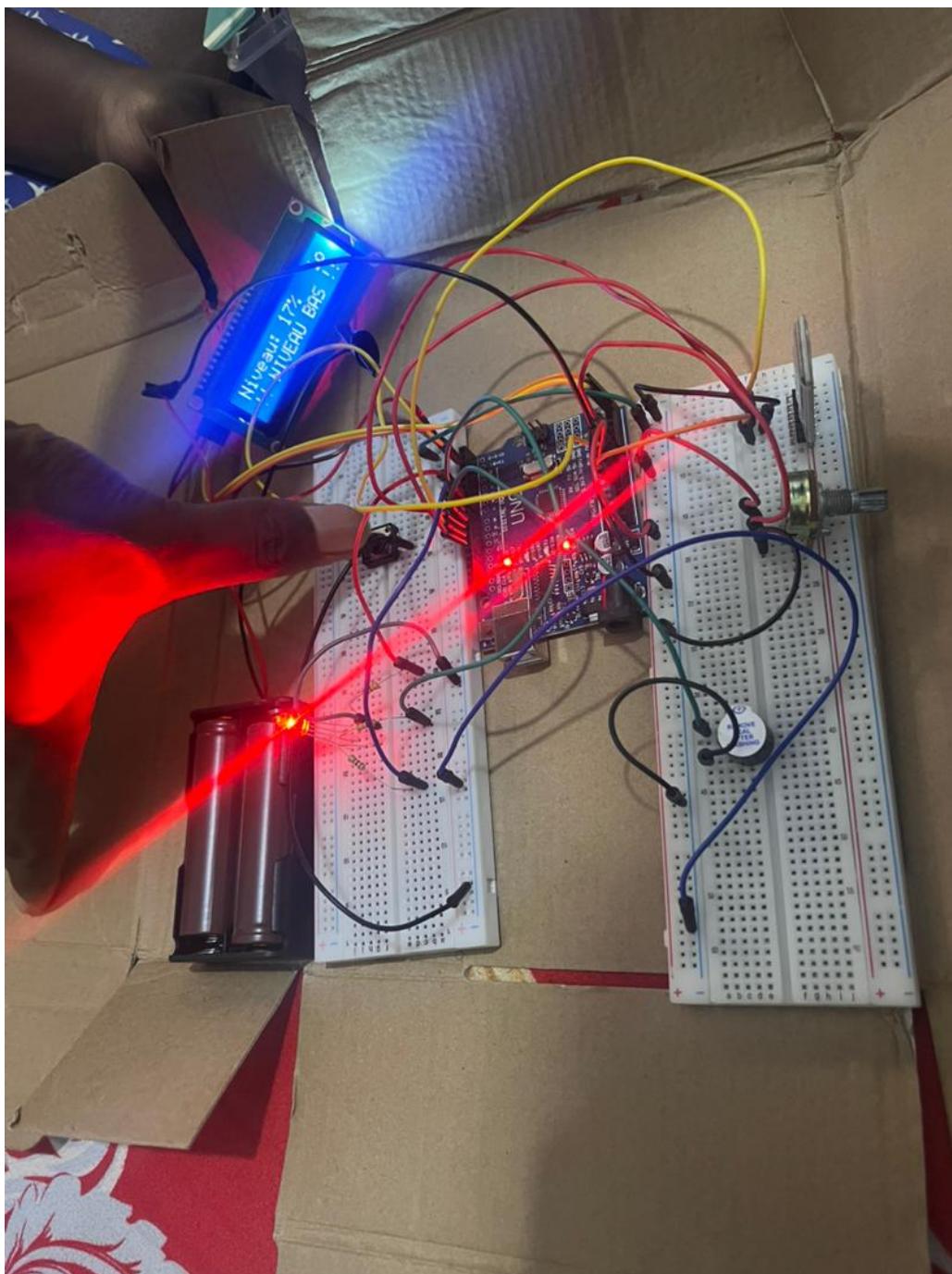
Figure 6.1: Cas du niveau de gaz élevé



**Figure 6.2:** Cas du niveau de gaz moyen



**Figure 6.3:** Cas du niveau de gaz bas



**Figure 6.4:** Fonctionnement du buzzer

# Chapitre 7

## Difficultés Rencontrées et Solutions

Ce chapitre présente les principales difficultés rencontrées tout au long de la réalisation du projet, ainsi que les solutions mises en œuvre pour les surmonter. Il met également en évidence les leçons apprises, tant sur le plan technique qu'organisationnel.

### 7.1 Difficultés Matérielles

#### 7.1.1 Difficulté d'acquisition du matériel

L'une des principales contraintes rencontrées concerne l'acquisition du matériel nécessaire à la réalisation du dispositif embarqué. En particulier, la commande du capteur ultrasonore DYP-L06 a connu un retard important, s'étalant sur plus d'un mois. Ce retard a fortement impacté le planning initial du projet.

**Causes identifiées :**

- Disponibilité limitée du capteur sur le marché local.
- Délais de livraison prolongés pour les commandes en ligne.
- Contraintes logistiques liées à l'importation du matériel.

**Solutions mises en œuvre :**

- Réorganisation du planning en priorisant les tâches logicielles.
- Étude théorique approfondie du capteur en attendant sa réception.
- Simulation et préparation du code embarqué sans matériel physique.

#### 7.1.2 Contraintes financières

La mobilisation des fonds nécessaires à l'achat des composants électroniques a également constitué une difficulté majeure.

**Problèmes rencontrés :**

- Budget limité pour un projet de groupe.
- Retard dans la contribution financière de certains membres.

**Solutions adoptées :**

- Réduction des coûts en choisissant des composants alternatifs lorsque possible.
- Mutualisation des ressources entre les membres du groupe.
- Planification progressive des achats selon les priorités.

### 7.1.3 Disponibilité des bouteilles de gaz pour les tests

La réalisation des tests en conditions réelles nécessitait la disponibilité de bouteilles de gaz vides et pleines, ce qui n'a pas toujours été évident.

**Solutions :**

- Collaboration avec des particuliers et des commerces locaux.
- Utilisation de bouteilles partiellement remplies pour certains tests intermédiaires.

## 7.2 Difficultés Logicielles

### 7.2.1 Courbe d'apprentissage de React.JS

Le développement de l'application web avec React.JS a représenté un défi pour certains membres de l'équipe, notamment ceux n'ayant pas d'expérience préalable avec ce framework.

**Difficultés rencontrées :**

- Prise en main du langage JavaScript et de la syntaxe JSX.
- Compréhension de la gestion de l'état (Hooks, Context) et de l'architecture des composants.
- Consommation des API REST et gestion asynchrone des données.

**Solutions apportées :**

- Répartition des tâches selon le niveau de compétence.
- Auto-formation à travers des tutoriels et la documentation officielle.
- Sessions de travail collaboratif et partage de connaissances.

### 7.2.2 Déploiement du backend

Le déploiement du backend de l'application a également posé plusieurs difficultés, notamment lors de la mise en production.

**Problèmes rencontrés :**

- Configuration du serveur et des variables d'environnement.
- Problèmes de compatibilité entre la base de données et le backend.
- Gestion des erreurs lors du déploiement sur une plateforme cloud.

**Solutions mises en place :**

- Tests locaux approfondis avant le déploiement.
- Utilisation de journaux (logs) pour identifier les erreurs.
- Simplification progressive de l'architecture backend.

## 7.3 Difficultés d'Intégration

L'intégration entre le dispositif embarqué, l'application web et le backend a constitué l'une des phases les plus complexes du projet.

**Difficultés principales :**

- Latence dans la remontée des données entre la passerelle LoRa et l'interface web.

- Problèmes de synchronisation des données.
- Différences de rendu selon les navigateurs web.
- Gestion des erreurs de communication avec l'API.

**Solutions adoptées :**

- Multiplication des tests sur différents navigateurs.
- Mise en place de mécanismes de rafraîchissement automatique.
- Validation progressive de chaque maillon de la chaîne de communication.

## 7.4 Leçons Apprises

La réalisation de ce projet a permis à l'équipe d'acquérir de nombreuses compétences et enseignements.

**Principales leçons retenues :**

- L'importance d'une bonne planification et d'une gestion réaliste des délais.
- La nécessité d'anticiper les problèmes d'approvisionnement matériel.
- La valeur du travail en équipe et du partage des connaissances.
- L'intérêt de tester progressivement chaque composant avant l'intégration globale.
- L'adaptabilité face aux imprévus techniques et organisationnels.

Ces expériences ont contribué à renforcer les compétences techniques et organisationnelles des membres de l'équipe et constituent un apport significatif pour leurs futurs projets académiques et professionnels.

# Chapitre 8

## Limites et Perspectives d'Évolution

Ce chapitre présente les principales limites du système développé dans le cadre de ce projet, ainsi que les perspectives d'amélioration et d'évolution possibles. Il met en évidence les axes de progression tant sur le plan matériel que logiciel, et ouvre la voie à une éventuelle transformation du prototype en une solution commerciale.

### 8.1 Limites Actuelles

Malgré les résultats satisfaisants obtenus, le système présente certaines limites inhérentes au cadre académique et aux ressources disponibles.

#### 8.1.1 Limites matérielles

- Le dispositif repose sur un seul capteur ultrasonore, ce qui peut limiter la précision dans certaines conditions (inclinaison de la bouteille, vibrations).
- La fixation du capteur dépend fortement de la qualité du couplage acoustique, pouvant influencer la fiabilité des mesures.
- L'autonomie, bien que satisfaisante (environ 30 jours), reste limitée par l'utilisation de composants peu optimisés énergétiquement.
- Le prototype n'intègre pas encore un boîtier industriel certifié pour un usage prolongé.

#### 8.1.2 Limites logicielles

- Les algorithmes de prédiction reposent sur des modèles simples et ne tiennent pas encore compte de tous les paramètres d'usage (habitudes, saisonnalité).
- L'application web nécessite une connexion internet permanente pour accéder aux données et ne fonctionne pas en mode hors-ligne.
- Le backend reste basique et n'intègre pas encore de mécanismes avancés d'analyse ou d'optimisation des données.

### 8.2 Améliorations Matérielles

Plusieurs améliorations matérielles peuvent être envisagées afin d'augmenter les performances et la fiabilité du système.

- Utilisation de plusieurs capteurs ultrasonores pour améliorer la précision et introduire une redondance des mesures.
- Remplacement de l'Arduino Uno par un microcontrôleur plus performant et économique en énergie (ESP32, STM32).
- Intégration d'un boîtier robuste et étanche, conforme aux normes industrielles.
- Ajout de capteurs complémentaires (température, inclinaison) pour corriger et affiner les mesures.
- Optimisation du circuit d'alimentation pour augmenter l'autonomie de la batterie.

## 8.3 Améliorations Logicielles

Des évolutions logicielles significatives peuvent également être envisagées.

- Amélioration des algorithmes de prédiction à l'aide de techniques d'apprentissage automatique basées sur l'historique de consommation.
- Développement d'une Progressive Web App (PWA) pour permettre une installation sur mobile et un fonctionnement partiel hors-ligne.
- Ajout de notifications intelligentes (alertes personnalisées, prévisions avancées).
- Amélioration de l'interface utilisateur pour une meilleure expérience (UX/UI).
- Sécurisation renforcée des données échangées entre l'application et le backend.

## 8.4 Évolution vers une Solution Commerciale

À plus long terme, le projet peut évoluer vers une solution commercialisable.

### 8.4.1 Étapes envisagées

- Validation du prototype à travers des tests intensifs en conditions réelles.
- Miniaturisation du dispositif et optimisation du design industriel.
- Certification du produit selon les normes de sécurité et de compatibilité électromagnétique.
- Déploiement de l'application web sur une infrastructure cloud robuste et scalable.
- Mise en place d'une infrastructure backend scalable.

### 8.4.2 Vision à long terme

À terme, la solution pourrait être étendue à :

- La gestion intelligente de la consommation énergétique domestique.
- L'intégration dans des systèmes domotiques et des plateformes IoT.
- Une utilisation à l'échelle industrielle ou commerciale (restaurants, hôtels, entreprises).

Ces perspectives montrent que le projet constitue une base solide pour des développements futurs, tant académiques que professionnels.

# Chapitre 9

## Conclusion Générale

### 9.1 Synthèse du Travail Réalisé

Ce projet avait pour objectif principal de concevoir et de réaliser un système intelligent de surveillance du niveau de gaz domestique, combinant un dispositif embarqué, une application mobile et une infrastructure logicielle backend. Face aux problématiques réelles rencontrées par de nombreux ménages, notamment l'absence de visibilité sur la quantité de gaz restante et les risques de rupture imprévue, notre solution vise à apporter une réponse technologique accessible, fiable et sécurisée.

Sur le plan matériel, nous avons conçu un système distribué innovant composé de deux unités distinctes : un module de mesure autonome fixé sous la bouteille et un module de supervision déporté. Cette architecture repose sur la technologie LoRa, offrant une portée étendue et une excellente autonomie. Le capteur ultrasonore DYP-L06, piloté par un Arduino Nano, assure une mesure précise et non intrusive, tandis que l'unité d'affichage permet une surveillance confortable depuis l'espace de vie.

Sur le plan logiciel, l'intégration d'une passerelle LoRa optionnelle ouvre le système vers le cloud. L'application web React.JS développée ne se limite plus à une connexion locale, mais devient un véritable outil de supervision à distance, exploitant les données centralisées par le backend pour offrir des analyses de consommation et des prédictions fiables.

Ainsi, ce projet a permis de couvrir l'intégralité de la chaîne technologique, depuis l'acquisition physique des données jusqu'à leur valorisation numérique à travers une interface utilisateur intuitive.

### 9.2 Atteinte des Objectifs

Les objectifs fixés au début du projet ont été atteints de manière satisfaisante.

- **Conception d'un système de mesure non intrusif :** L'utilisation de la technologie ultrasonore à travers la paroi métallique de la bouteille a permis d'éviter toute modification physique ou risque de fuite, tout en garantissant une précision acceptable pour un usage domestique.
- **Transmission longue portée :** L'adoption de la technologie LoRa a permis de s'affranchir des limites de distance du Bluetooth, autorisant le placement de la bouteille à l'extérieur de l'habitation.
- **Supervision locale et distante :** La double interface (écran dédié et application web via cloud) offre une flexibilité d'usage optimale pour l'utilisateur.

- **Mise en place d'une architecture logicielle backend :** Le backend assure la gestion des données, la persistance et la préparation des traitements futurs plus avancés, ouvrant la voie à une évolution vers une solution connectée à grande échelle.
- **Respect des contraintes de sécurité et d'autonomie :** Le système fonctionne en basse tension, ne génère aucune étincelle et présente une autonomie satisfaisante, compatible avec un usage domestique quotidien.

Globalement, la solution obtenue répond aux exigences du cahier des charges initial tout en restant réaliste sur le plan technique, économique et opérationnel.

## 9.3 Apports Personnels et Compétences Acquises

Ce projet a constitué une expérience particulièrement formatrice, tant sur le plan technique que sur le plan humain et organisationnel.

Sur le plan technique, il nous a permis de renforcer nos compétences en électronique embarquée, notamment en acquisition de signaux, traitement de données issues de capteurs physiques et gestion de microcontrôleurs. La manipulation de la technologie ultrasonore et l'intégration matérielle dans un environnement réel ont représenté un apprentissage concret des contraintes du monde physique, souvent absentes des projets purement logiciels.

Sur le plan logiciel, nous avons approfondi notre maîtrise du développement web avec React.JS, en mettant en œuvre une architecture claire, une gestion d'état efficace et une interface utilisateur ergonomique. Le développement du backend nous a également permis de consolider nos compétences en conception d'API REST, modélisation de données et communication client-serveur.

Sur le plan méthodologique et humain, ce projet nous a appris à travailler en équipe sur un système complexe, à gérer les contraintes de temps et de ressources, à communiquer efficacement entre sous-groupes matériel et logiciel, et à résoudre des problèmes imprévus de manière collaborative. Il nous a également sensibilisés à l'importance de la rigueur, de la documentation et de la validation progressive des solutions techniques.

Ainsi, ce projet dépasse largement le cadre académique pour constituer une expérience professionnalisaante proche des conditions réelles de développement industriel.

## 9.4 Perspectives

Bien que fonctionnelle, la solution développée présente plusieurs axes d'amélioration et d'évolution.

Sur le plan matériel, l'intégration d'un microcontrôleur plus performant et basse consommation, tel qu'un ESP32, permettrait d'ajouter une connectivité Wi-Fi directe et d'optimiser l'autonomie énergétique. L'amélioration du capteur et de l'algorithme de traitement du signal pourrait également accroître la précision des mesures, notamment dans des conditions environnementales variables.

Sur le plan logiciel, l'application web pourrait évoluer vers une gestion multi-utilisateurs et multi-dispositifs, l'intégration de notifications intelligentes, ainsi qu'un système avancé d'analyse de consommation. Le backend pourrait être enrichi par des modèles de prédiction basés sur l'apprentissage automatique afin d'anticiper plus finement les besoins de remplacement de bouteilles.

Enfin, à plus long terme, ce projet pourrait évoluer vers une solution commerciale complète intégrée à des écosystèmes de maison connectée (smart home), permettant une gestion

centralisée de l'énergie domestique et une amélioration globale du confort et de la sécurité des utilisateurs.

En conclusion, ce travail constitue une base solide pour le développement de solutions IoT appliquées aux besoins quotidiens, et illustre la pertinence de l'ingénierie numérique au service de problématiques concrètes de la vie courante.

# Références

- Meta. *React Documentation*. Disponible sur : <https://react.dev> (consulté en janvier 2026).
- VMware. *Spring Boot Reference Documentation*. Disponible sur : <https://docs.spring.io/spring-boot/docs/current/reference/html/> (consulté en janvier 2026).
- Arduino. *Arduino Official Documentation*. Disponible sur : <https://docs.arduino.cc> (consulté en janvier 2026).
- DYP Sensor. *DYP-L06 Ultrasonic Sensor Datasheet*. Disponible sur : <https://www.dypcn.com/uploads/DYP-L06.pdf> (consulté en janvier 2026).
- Buyya, R., & Dastjerdi, A. V. *Internet of Things: Principles and Paradigms*. Morgan Kaufmann, 2016.
- Margolis, M. *Arduino Cookbook*. 2<sup>e</sup> édition, O'Reilly Media, 2016.
- Valvano, J. W. *Embedded Systems: Real-Time Interfacing to ARM Cortex-M Microcontrollers*. CreateSpace, 2015.
- Render. *Deploying Spring Boot Applications*. Disponible sur : <https://render.com/docs/deploy-spring-boot> (consulté en janvier 2026).
- Semtech. *LoRa Modulation Basics*. Disponible sur : <https://www.semtech.com/lora> (consulté en janvier 2026).
- Google. *Google Identity Platform Documentation*. Disponible sur : <https://developers.google.com/identity> (consulté en janvier 2026).
- Kinsler, L. E., Frey, A. R., Coppens, A. B., & Sanders, J. V. *Fundamentals of Acoustics*. John Wiley & Sons, 2000.
- Net Ninja. *React Tutorial for Beginners*. YouTube Playlist. Disponible sur : [https://www.youtube.com/playlist?list=PL4cUxeGkcC9gZD-TvwFp28qnm\\_j8ugaZ](https://www.youtube.com/playlist?list=PL4cUxeGkcC9gZD-TvwFp28qnm_j8ugaZ) (consulté en janvier 2026).
- Andreas Spiess. *LoRa with Arduino*. YouTube Video. Disponible sur : <https://www.youtube.com/watch?v=hM0wbNUpDQA> (consulté en janvier 2026).
- FreeCodeCamp.org. *Internet of Things Full Course*. YouTube. Disponible sur : <https://www.youtube.com/watch?v=QSIPNh0iMoE> (consulté en janvier 2026).
- World LPG Association. *Global LPG Market Outlook*. Disponible sur : <https://www.wlpga.org/publications/global-lpg-market-outlook/> (consulté en janvier 2026).
- International Energy Agency (IEA). *Energy Access Outlook – Cooking Fuels*. Disponible sur : <https://www.iea.org/reports/energy-access-outlook-2017> (consulté en janvier 2026).

- World Health Organization (WHO). *Household Air Pollution and Health*. Disponible sur : <https://www.who.int/news-room/fact-sheets/detail/household-air-pollution-and-health> (consulté en janvier 2026).