

Roadmap
SENSOR SENSEI
(T-IOT-902)

Auteur	Cyril BANCQ
Auteur	Lorenzo Giralt
Auteur	Lyann Bourlon
Auteur	Aniss Belhaq
Lieu	Epitech Marseille

Table des matières

Table des matières.....	2
1. Introduction	3
1.1. Objectif du projet	3
1.2. Contexte du projet.....	3
2. Description du projet.....	4
2.1. Spécifications générales	4
2.2. Composants du système.....	4
3. Exigences Techniques	4
3.1. Compatibilité des capteurs.....	4
3.2. Fonctionnalités du Firmware.....	5
3.3. Implémentation de la technologie LoRa	5
3.4. Configuration et Surveillance	6
3.5. Compatibilité Matérielle	6
3.6. Tests et Validation.....	6
4. Gestion de L'Energie et Autonomie	7
4.1. Optimisation du Code.....	7
4.2. Utilisation du mode Veille	7
4.3. Gestion des communications	7
4.4. Choix des batteries	8
4.5. Intégration de sources d'Energie Alternatives	8
4.6. Surveillance de l'Etat de l'Energie	8
4.7. Tests d'Endurance.....	9
4.8. Optimisation basée sur les résultats	9
5. Livrables.....	9
5.1. Capteur Autonome avec LoRa	9
5.2. Passerelle LoRa/WiFi	10
5.3. Firmware Amélioré.....	10
5.4. Documentation Technique	11
5.5. Rapport de Tests.....	11
6. Méthodologie et Outils.....	11
6.1. Langages de Programmation.....	11
6.2. Outils de Test et de Débogage.....	12
6.3. Outils de Test et de Débogage.....	12

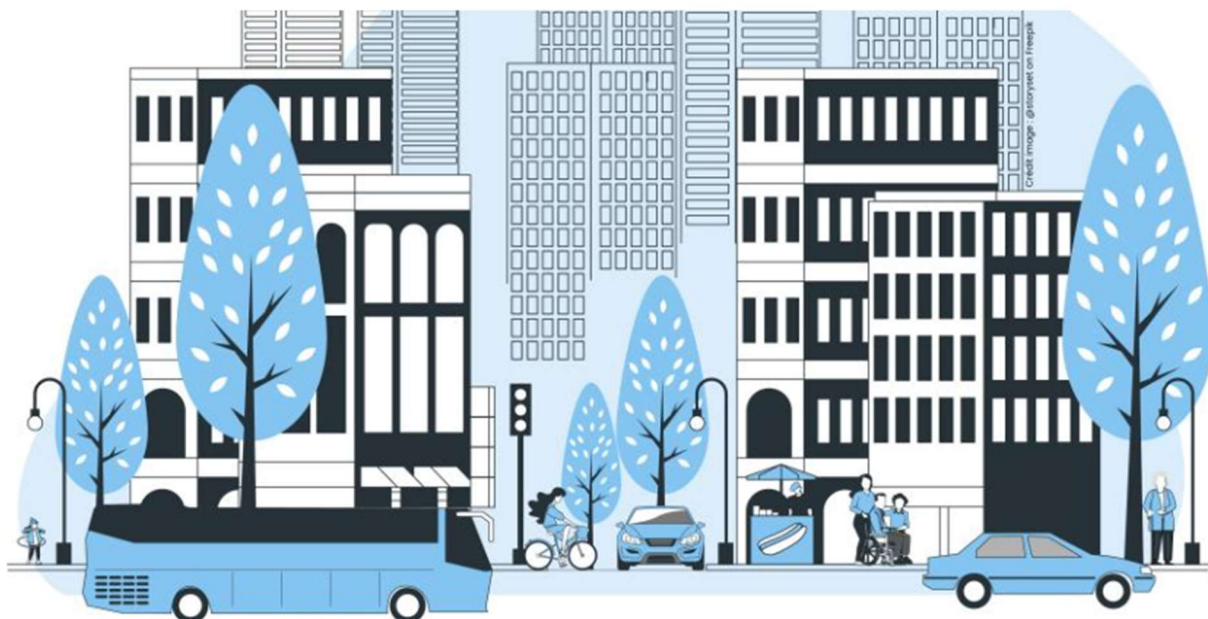
1. Introduction

1.1. Objectif du projet

Le projet vise à développer un capteur IoT pour le réseau sensor.community qui utilise la technologie de communication LoRa (Long Range) pour transmettre les données de mesure. L'objectif principal est de remplacer les connexions WiFi traditionnelles, réduire la consommation d'énergie et augmenter la portée des communications, ce qui est particulièrement bénéfique dans les zones rurales où l'accès à Internet est limité.

1.2. Contexte du projet

La technologie LoRa est privilégiée pour ce projet en raison de sa capacité à fonctionner sur de longues distances tout en conservant une très faible consommation d'énergie. Cela permet d'implanter des capteurs dans des zones moins accessibles, sans la nécessité d'une infrastructure réseau complexe et coûteuse. De plus, cette technologie est déjà largement adoptée pour diverses applications IoT dans le domaine agricole, la surveillance environnementale, et plus, ce qui en fait un choix stratégique pour l'intégration et la future expansion du projet sensor.community.



2. Description du projet

2.1. Spécifications générales

Le système développé sera composé de deux éléments principaux : une partie autonome (le capteur) et une passerelle (Gateway). Les capteurs, dispersés sur le terrain, recueilleront les données environnementales et les enverront à la passerelle via LoRa. La passerelle, ensuite, transmettra ces informations au serveur de sensor.community via une connexion WiFi sécurisée.

2.2. Composants du système

Partie autonome (Le capteur) :

Chaque capteur sera équipé de modules spécifiques pour mesurer des variables environnementales telles que la température, l'humidité, la qualité de l'air, etc. Ils seront également équipés d'une source d'alimentation autonome (batteries, avec option de recharge solaire).

Le Gateway (La passerelle) :

La passerelle servira de relais entre les capteurs et le serveur sensor.community. Elle sera conçue pour être facilement configurable via WiFi et capable de gérer efficacement les transmissions de plusieurs capteurs simultanément.

3. Exigences Techniques

Cette section détaille les spécifications techniques requises pour le développement et l'implémentation du capteur IoT utilisant la technologie LoRa dans le cadre du projet sensor.community.

3.1. Compatibilité des capteurs

Le nouveau système devra être compatible avec la liste des capteurs déjà supportés par le firmware actuel afin de garantir une intégration sans heurts avec les infrastructures existantes. Ces capteurs mesurent divers paramètres environnementaux tels que la température, l'humidité, la concentration de particules fines, etc.

Cf : Document Identification des Capteurs.

3.2. Fonctionnalités du Firmware

Le firmware développé devra inclure toutes les fonctionnalités du firmware existant, tout en y intégrant la technologie LoRa. Les fonctionnalités clés à maintenir sont :

Point d'accès WiFi pour la configuration et la surveillance :

Permet aux utilisateurs de configurer le capteur et de surveiller son activité en temps réel via une interface web accessible par WiFi.

Compatibilité avec les capteurs actuels :

Doit supporter les mêmes types de capteurs que ceux déjà utilisés par le projet sensor.community, assurant ainsi la continuité et la compatibilité des données recueillies.

Mise à jour Over-The-Air (OTA) :

Capacité de mettre à jour le firmware à distance sans nécessiter un accès physique au capteur, essentielle pour les déploiements à grande échelle.

3.3. Implémentation de la technologie LoRa

L'intégration de la technologie LoRa doit permettre une transmission efficace des données sur de longues distances sans nécessiter une connexion constante à Internet, avec les spécifications suivantes :

Protocole LoRaWAN :

Utilisation du protocole LoRaWAN pour la communication entre les capteurs et la passerelle, optimisant la portée et la consommation d'énergie.

Chiffrement des Données :

Toutes les données transmises via LoRa devront être sécurisées via chiffrement, garantissant la confidentialité et l'intégrité des données environnementales mesurées.

Gestion de l'Alimentation :

Le capteur devra être capable de fonctionner avec une source d'énergie limitée, incluant des mécanismes de gestion de l'énergie tels que le mode veille profond pour prolonger la durée de vie des batteries.

3.4. Configuration et Surveillance

Interface Utilisateur Conviviale :

La configuration du capteur et la surveillance des données devront être accessibles via une interface graphique utilisateur conviviale et sécurisée.

Diagnostics et Reporting :

Le firmware devra inclure des fonctionnalités de diagnostic pour aider à identifier et résoudre les problèmes de fonctionnement du capteur. Des rapports périodiques sur l'état de santé du capteur et les performances du réseau LoRa seront également nécessaires.

3.5. Compatibilité Matérielle

Microcontrôleur ESP32 :

Le choix de l'ESP32 comme microcontrôleur est dicté par sa compatibilité avec le firmware actuel (bien que toujours en phase expérimentale), ses capacités de gestion de l'énergie, et son support intégré pour la communication WiFi et Bluetooth.

Modules LoRa :

Sélection de modules LoRa qui peuvent être facilement intégrés avec l'ESP32, tout en supportant les exigences du protocole LoRaWAN.

3.6. Tests et Validation

Tests de Portée et Fiabilité :

Réaliser des tests exhaustifs pour vérifier la portée effective de la communication LoRa dans divers environnements, ainsi que la fiabilité et la régularité de la transmission des données.

Tests de Compatibilité :

Assurer que le nouveau firmware est totalement compatible avec les capteurs et infrastructures existants du projet sensor.community.

Tests d'Endurance et de Durabilité :

Évaluer la durabilité des capteurs et la passerelle dans des conditions environnementales variées pour s'assurer qu'ils fonctionnent correctement sur de longues périodes.

4. Gestion de L'Énergie et Autonomie

La gestion efficace de l'énergie est cruciale pour le succès des capteurs IoT, surtout dans des applications où l'accès à l'alimentation est limité. Cette section décrit les stratégies et technologies à utiliser pour minimiser la consommation d'énergie et maximiser l'autonomie des capteurs utilisant la communication LoRa.

Minimisation de la Consommation Energetique

4.1. Optimisation du Code

Écrire un code efficace qui utilise les fonctionnalités de basse consommation de l'ESP32.

Implémenter des algorithmes de gestion de l'énergie pour activer le capteur et les modules de communication uniquement lors de l'acquisition ou la transmission des données.

4.2. Utilisation du mode Veille

Exploiter le mode veille profond de l'ESP32 qui permet de réduire drastiquement la consommation d'énergie en désactivant les fonctions non essentielles lorsque le capteur n'est pas en activité.

Réveiller périodiquement le capteur pour les mesures et la transmission des données pour maintenir une balance optimale entre performance et consommation d'énergie.

4.3. Gestion des communications

Programmer la fréquence de transmission des données basée sur les besoins réels de l'application pour réduire la charge de travail du transmetteur LoRa.

Utiliser des protocoles de transmission de données qui minimisent le temps d'activation du transmetteur, tels que les protocoles basés sur les événements ou le changement d'état.

Solutions pour Maximiser l'Autonomie

4.4. Choix des batteries

Sélectionner des batteries à haute capacité et à décharge lente, appropriées pour l'utilisation en extérieur et dans des conditions climatiques variées.

Evaluer l'utilisation de batteries rechargeables avec des panneaux solaires pour prolonger la durée de vie du capteur sans intervention manuelle.

4.5. Intégration de sources d'Energie Alternatives

Concevoir les capteurs pour qu'ils puissent être alimentés par des sources d'énergie renouvelables, telles que le solaire, le thermique ou même le mouvement.

Inclure un circuit de gestion de l'énergie qui optimise la charge et la décharge des batteries, et qui gère efficacement l'alimentation provenant de sources multiples.

4.6. Surveillance de l'Etat de l'Energie

Intégrer des capteurs de niveau de batterie pour surveiller et rapporter l'état de charge des batteries, permettant une maintenance proactive.

Configurer des alertes automatiques pour informer les utilisateurs lorsque les niveaux de batterie sont bas et nécessitent une attention.



Tests et validation de l'autonomie

4.7. Tests d'Endurance

Réaliser des tests d'endurance pour valider la durée de vie des batteries sous différentes conditions de fonctionnement et configurations.

Simuler des scénarios d'usage intensif pour identifier les besoins en optimisation et ajuster les paramètres de gestion de l'énergie.

4.8. Optimisation basée sur les résultats

Analyser les données recueillies lors des tests pour optimiser davantage la consommation d'énergie.

Apporter des modifications au design matériel ou logiciel basées sur les insights obtenus pour améliorer la performance énergétique.

5. Livrables

Cette section détaille les composants à livrer, les fonctionnalités requises, et les critères d'acceptation qui doivent être satisfaits pour considérer le projet de capteur IoT utilisant la technologie LoRa comme complet et réussi.

Fonctionnalités à Livrer

5.1. Capteur Autonome avec LoRa

Mesures Environnementales :

Capable de mesurer des paramètres comme la température, l'humidité, la qualité de l'air, etc.

Communication LoRa :

Transmission des données via LoRa à la passerelle. Les données doivent être cryptées et conformes aux standards LoRaWAN.

Autonomie Énergétique :

Intégration de solutions pour une gestion d'énergie efficace, incluant l'utilisation de batteries à longue durée et potentiellement un support pour la recharge solaire.

5.2. Passerelle LoRa/WiFi

Réception et Transmission :

Recevoir les données des capteurs via LoRa et les transmettre au serveur de sensor.community via une connexion WiFi sécurisée.

Configuration et Surveillance :

Point d'accès WiFi pour la configuration de la passerelle et la surveillance en temps réel des données collectées.

Support Multi-Capteurs :

Capable de gérer les communications de plusieurs capteurs simultanément sans pertes de données.

5.3. Firmware Amélioré

Maintien des Fonctionnalités Existantes :

Toutes les fonctionnalités du firmware original doivent être conservées et pleinement fonctionnelles.

Intégration de LoRa :

Ajout efficace de la communication LoRa tout en maintenant la performance et la stabilité du système.

Mise à jour OTA :

Support de mise à jour du firmware à distance pour faciliter les upgrades et la maintenance.



Documentation à fournir

5.4. Documentation Technique

Guide d'Installation et de Configuration :

Instructions détaillées sur l'installation des capteurs et de la passerelle, la mise en place de la communication LoRa, et la configuration initiale du système.

Spécifications Techniques :

Détails complets sur les spécifications des capteurs, la passerelle, et le firmware.

Diagrammes de Séquence :

Diagrammes illustrant l'architecture du système, les flux de données, et les interactions entre les différents composants. Intégration de solutions pour une gestion d'énergie efficace, incluant l'utilisation de batteries à longue durée et potentiellement un support pour la recharge solaire.

5.5. Rapport de Tests

Tests de Performance :

Résultats des tests validant la portée de la communication LoRa, la stabilité de la connexion WiFi, et l'efficacité de la gestion de l'énergie.

Tests de Compatibilité :

Confirmation que le système est compatible avec divers capteurs et réseaux LoRaWAN existants.

6. Méthodologie et Outils

Dans cette section, nous détaillons la méthodologie et les outils qui seront utilisés pour le développement et la mise en œuvre du nouveau contrôleur du système de convoyeur.

6.1. Langages de Programmation

C/C++ : Utilisé pour le développement du firmware du contrôleur.

JavaScript : Utilisé pour le développement de l'API REST.

6.2. Outils de Test et de Débogage

Simulateurs Arduino : Utilisés pour tester le firmware avant le déploiement sur le matériel réel.

Postman/Swagger : Utilisé pour tester l'API REST et vérifier son intégration avec le WMS Dolibarr.

6.3. Outils de Test et de Débogage

Trello : Utilisé pour le suivi des tâches, des bugs et des jalons du projet.

Git : Utilisé pour le contrôle de version du code source et la chaîne CI/CD.