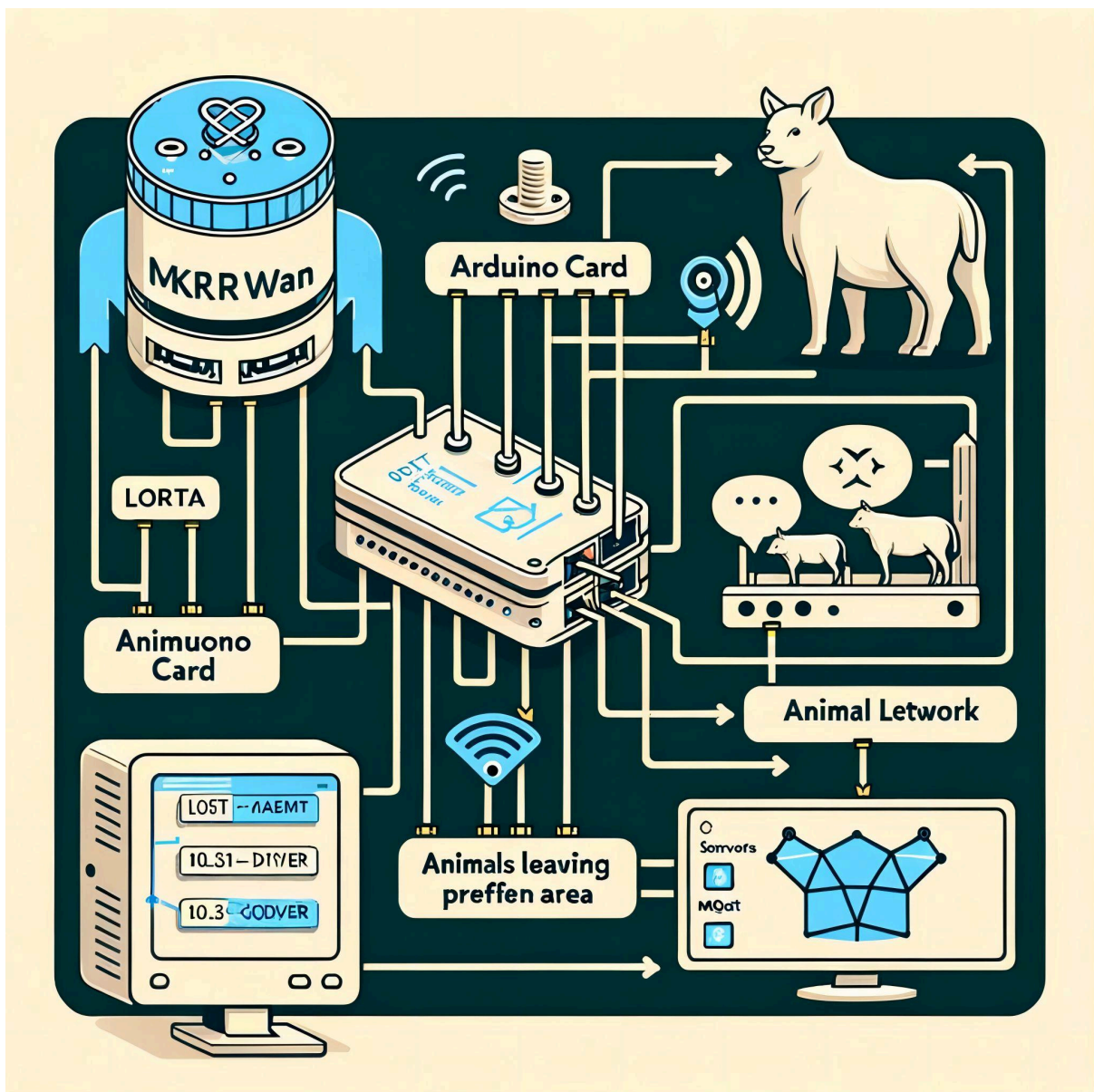


# Projet - (Année 2023-2024)

## Spécification\_info\_S4

Mouzheng LI  
Lianghong LI  
Guokuang DENG  
Zhengdong XU



# Description du Projet

Le but de ce projet est de développer un prototype pour l'identification, le positionnement et, éventuellement, le suivi de l'état de santé des animaux tels que les chats, les chiens, les chevaux, et les vaches. Ce prototype comprendra un collier connecté, utilisant une carte Arduino MKRWAN, et un tableau de bord pour afficher en temps réel les données reçues. Le système alertera également lorsque les animaux sortent d'une zone prédéfinie ou, potentiellement, lorsqu'ils subissent un niveau de stress élevé.

## Architecture Technique

- Carte Arduino MKRWAN : Sert de base au collier connecté pour chaque animal. Capable d'envoyer des données, y compris la position et éventuellement l'état de santé, via le réseau LoRa.
- Réseau LoRa: Utilisé pour la communication sans fil à basse consommation entre les colliers connectés et le serveur.
- Serveur MQTT (`srv-lora`) : Reçoit les données envoyées par les colliers via le réseau LoRa.
- Tableau de bord: Interface utilisateur affichant les positions en temps réel et les alertes de sortie de zone pour chaque animal. Il y aura deux versions du tableau de bord, une version minimale utilisant la fenêtre "turtle" pour un affichage graphique simple, et une version évoluée présentant les données sur une page web.

## Spécifications Fonctionnelles

- Collier connecté :
  - Développer un programme Arduino pour simuler le déplacement aléatoire de l'animal à partir d'une position initiale (0,0).
  - Envoie des données de position (et éventuellement de santé) toutes les 10 secondes.
  - Format des données : `Nom : Coordonnée X:Coordonnée Y[:Données Supplémentaires]` (ex: `Marguerite :34:-12:T=37` pour inclure la température).

## - Tableau de bord :

- Écoute les messages sur le broker MQTT `srv-lora`.
- Affiche les positions de tous les animaux avec des couleurs distinctes.
- Alertes pour les animaux hors de la zone prédéfinie (-200, -200) à (200, 200).
- La version minimale utilise "turtle" pour l'affichage et des "prints" pour les alertes par une application de Python.
- La version évoluée utilise une page web pour afficher les positions et les alertes.

## Planning

- Développement du projet : Réparti sur 6 séances encadrées et 3 séances en autonomie, avec l'utilisation de Git et GitHub pour la gestion de version et le partage du code.

	Lianghong LI	Mouzheng LI	Guokuang DENG	Ensemble
Séance 1 et 2	Développer le programme pour la carte Arduino MKRWAN.	Configurer le réseau LoRa et le serveur MQTT.	Commencer le développement du tableau de bord. C : Back-end et interface "turtle".	Conception de l'interface frontale.
Séance 3 et 4	Tests en situation réelle, ajustement et optimisation de la communication et du traitement par le serveur.	Tests en situation réelle, ajustement et optimisation de la communication et du traitement par le serveur.	Continuer le développement et l'amélioration du tableau de bord, débiter l'intégration et les tests.	Continuer le développement et l'amélioration du tableau de bord, débiter l'intégration et les tests.
Séance 5 et 6	Intégration du système, assurer la connexion fluide entre les parties. Tests parallèles et préparation de la maintenance.	Intégration du système, assurer la connexion fluide entre les parties. Tests parallèles et préparation de la maintenance.	Intégration du système, assurer la connexion fluide entre les parties. Tests parallèles et préparation de la maintenance.	Intégration du système, assurer la connexion fluide entre les parties. Tests parallèles et préparation de la maintenance.