

2024 - 2025

# Cahier de recette - reprise

Canne Intelligente pour Aveugle

**Encadrant :** Gilles VENTURINI

**Client :** Gilles VENTURINI

**Etudiant :** Irfan AKTURK

## Table des matières

1. Introduction .....	3
2. Périmètre de la recette .....	4
3. Comparaison entre spécifications et réalisation .....	5
4. Points non implémentés ou problèmes rencontrés.....	7
4.1 Fonctionnalités non implémentées.....	7
4.2 Problèmes techniques rencontrés .....	7
5. Recommandations pour la continuité du projet .....	8
6. Conclusion.....	9

## 1. Introduction

Ce document regroupe à la fois la validation des fonctionnalités mises en place et les recommandations pour la continuité du projet canne intelligente pour personnes malvoyantes. Il permet d'évaluer ce qui a été réalisé par rapport aux spécifications établie en début de projet et de préparer la reprise du projet pour des développements futurs.

L'objectif est d'abord de vérifier que les fonctionnalités développées respectent les exigences définies en début de projet. Ce document met en évidence ce qui a été validé, ce qui a été partiellement réalisé et ce qui pourrait être amélioré.

Il sert également de support pour les futurs étudiants qui pourraient travailler sur ce projet. Certaines fonctionnalités mériteraient d'être améliorer ou complétées. Ce document détaille les problèmes rencontrés et propose des pistes pour la poursuite du développement.

Cette documentation est donc essentielle pour assurer la pérennité du projet et permettre une reprise de celui-ci.

## 2. Périmètre de la recette

Le périmètre de cette recette couvre l'ensemble des fonctionnalités développées dans le cadre du projet canne intelligente pour personnes malvoyantes.

L'objectif est de s'assurer que les principales fonctionnalités du système ont bien été mises en place et qu'elles correspondent aux attentes définies dans les spécifications initiales. Ces fonctionnalités sont essentielles pour garantir le bon fonctionnement du dispositif et son utilité pour l'utilisateur final.

Les éléments pris en compte dans cette recette sont :

- La détection des obstacles grâce au capteur ultrason et l'alerte envoyée à l'utilisateur sous forme sonore et vibratoire.
- La gestion des modes de fonctionnement : mode marche (vibration pour signaler un obstacle) et mode exploration (reconnaissance d'objets et annonce vocale).
- L'utilisation d'un bouton poussoir pour permettre à l'utilisateur de changer de mode.
- Le système de synthèse vocale pour donner des indications à l'utilisateur sur les objets détectés et les distances.
- La gestion de l'alimentation, avec un interrupteur pour allumer et éteindre la canne, ainsi qu'une batterie permettant une autonomie suffisante.
- Le lancement automatique du programme et la correction des erreurs au démarrage, bien que cette fonctionnalité n'ait pas encore été totalement mise en place.

Ce périmètre définit donc les critères de validation qui permettront d'évaluer la conformité du projet avec les attentes initiales.

### 3. Comparaison entre spécifications et réalisation

Cette section compare les fonctionnalités prévues dans les spécifications initiales avec celles réellement mises en place. Elle permet de voir ce qui a été réalisé avec succès et ce qui n'a pas été mis en place.

Fonctionnalité	Spécifiée dans les exigences ?	Implémentée ?	Commentaires
Détection des obstacles avec capteur ultrason	<input checked="" type="checkbox"/> Oui	<input checked="" type="checkbox"/> Oui	Le capteur ultrason mesure la distance et envoie un retour sonore et vibratoire à l'utilisateur.
Mode Marche (alerte vibratoire uniquement)	<input checked="" type="checkbox"/> Oui	<input checked="" type="checkbox"/> Oui	Le vibreur s'active proportionnellement à la distance de l'obstacle par paliers.
Mode Exploration (reconnaissance d'objets + annonce vocale)	<input checked="" type="checkbox"/> Oui	<input checked="" type="checkbox"/> Oui	L'IA détecte et annonce les objets ainsi que leur distance.
Changement de mode via un bouton poussoir	<input checked="" type="checkbox"/> Oui	<input checked="" type="checkbox"/> Oui	Le passage d'un mode à l'autre est fluide et fonctionne correctement.
Synthèse vocale pour guider l'utilisateur	<input checked="" type="checkbox"/> Oui	<input checked="" type="checkbox"/> Oui	La synthèse vocale annonce le mode actif, les objets détectés et la distance des obstacles.
Gestion de l'alimentation (batterie et interrupteur)	<input checked="" type="checkbox"/> Oui	<input checked="" type="checkbox"/> Oui	Un interrupteur permet d'allumer et d'éteindre la Jetson Nano sans devoir la brancher/débrancher.
Lancement automatique du programme au démarrage	<input checked="" type="checkbox"/> Oui	<input checked="" type="checkbox"/> Non réalisé	Le programme ne se lance pas correctement au démarrage car les permissions sur ttyTHS1 ne sont pas conservées après un redémarrage.

Fonctionnalité	Spécifiée dans les exigences ?	Implémentée ?	Commentaires
Correction des erreurs au démarrage	<input checked="" type="checkbox"/> Oui	<input type="checkbox"/> réalisé	Non Certaines erreurs sont détectées, mais le système ne corrige pas automatiquement les problèmes liés aux permissions du capteur ultrason.

Comme on peut le voir la majorité des fonctionnalités prévues dans les spécifications ont été implémentées avec succès.

Cependant, deux points n'ont pas été réalisés :

- Le lancement automatique du programme : Le programme ne peut pas fonctionner correctement après un redémarrage à cause des problèmes de permissions sur le port série ttyTHS1.
- La correction des erreurs au démarrage : Certaines erreurs sont détectées, mais l'utilisateur doit encore intervenir manuellement pour restaurer certaines fonctionnalités, comme le capteur ultrason.

Des améliorations sont donc nécessaires pour assurer une meilleure autonomie du système et éviter les interventions manuelles après chaque redémarrage.

## 4. Points non implémentés ou problèmes rencontrés

Cette section liste les fonctionnalités non réalisées et les problèmes techniques rencontrés durant la réalisation du projet.

### 4.1 Fonctionnalités non implémentées

#### Lancement automatique du programme au démarrage :

- Le programme devait s'exécuter automatiquement dès l'allumage de la Jetson Nano.
- Un script de démarrage a été mis en place, mais l'exécution du programme rencontre des erreurs en raison de problèmes de permissions sur le port série ttyTHS1.
- Une solution définitive pour conserver les permissions après redémarrage reste à trouver.

#### Correction automatique des erreurs au démarrage :

- Le système devait détecter et corriger certaines erreurs critiques au lancement.
- Actuellement, certaines erreurs sont bien détectées, mais l'utilisateur doit intervenir manuellement pour les corriger (notamment les erreurs liées aux permissions du port série et aux capteurs).

### 4.2 Problèmes techniques rencontrés

#### Problème de permissions sur le port série ttyTHS1 :

- À chaque redémarrage, les permissions sont réinitialisées, ce qui empêche le programme d'accéder au capteur ultrason sans intervention manuelle.
- Plusieurs solutions ont été testées (chmod, udev rules, systemd), mais aucune n'a permis de conserver définitivement les permissions après un reboot.

#### Erreur TypeError dans main.py :

- Une erreur TypeError: unsupported operand type(s) for /: 'NoneType' and 'int' est apparue à plusieurs reprises, entraînant l'arrêt du programme.
- Cette erreur survient lorsque la fonction ultrasonic\_sensor.get\_distance() retourne None, ce qui signifie que la lecture du capteur a échoué.
- Une meilleure gestion des erreurs devrait être implémentée pour éviter que le programme ne plante lorsqu'une valeur invalide est reçue.

## 5. Recommandations pour la continuité du projet

Afin d'assurer la pérennité et l'amélioration du projet, voici quelques recommandations pour les futurs étudiants qui reprendront ce projet.

### Résolution du problème de permissions sur ttyTHS1 :

- Trouver une solution définitive pour conserver les droits d'accès au port série après un redémarrage, sans nécessiter d'intervention manuelle.
- Tester d'autres approches, comme une configuration plus avancée avec udev ou un lancement avec les permissions adéquates via systemd.
- Explorer des solutions pour attribuer définitivement les permissions au port ttyTHS1 pour tous les utilisateurs ou exécuter le programme constamment en super utilisateur.

### Correction automatique des erreurs au démarrage :

- Ajouter une gestion plus robuste des erreurs dans main.py pour éviter que le programme ne plante en cas de mauvaise lecture des capteurs.
- Intégrer un mécanisme de reconnexion automatique au capteur ultrason si une erreur de lecture survient.

## 6. Conclusion

Le projet de canne intelligente a permis de concevoir un dispositif intégrant un capteur ultrason, une caméra, un vibreur et un système audio pour assister les personnes malvoyantes. Plusieurs fonctionnalités ont été développées conformément aux spécifications initiales, notamment la détection d'obstacles, les alertes sonores et vibratoires, ainsi que les modes de fonctionnement adaptés.

Cependant, certains points restent à finaliser, comme le lancement automatique du programme au démarrage et la gestion permanente des permissions du port ttyTHS1. Ces éléments sont essentiels pour garantir une utilisation fluide et sans intervention manuelle.

Les futurs étudiants travaillant sur ce projet devront donc se concentrer en priorité sur ces améliorations afin d'aboutir à un système entièrement autonome et optimisé pour une utilisation quotidienne.