

# **PRONTO :**

# **Système embarqué de pilotage autonome**

23/05/2025

Groupe 88 :  
Titouan Daniélou  
Nicolas Dode  
Julien Fontan  
Loïc Fournier  
Antoine Jourdain

Professeur encadrant :  
Christophe Lohr



# **Objectif de la présentation**

Présenter les **enjeux** et les **étapes** de réalisation du projet Pronto, en mettant en avant la **méthodologie** utilisée, les **résultats** obtenus, et les **compétences** mobilisées.

# Plan

01

**Présentation du projet**

02

**Informatique**

03

**Mécanique et  
Électronique**

04

**Tests et difficultés  
rencontrées**

05

**Résultats finaux obtenus**

06

**Axes d'amélioration**

01

# Présentation du projet

# Notre projet



Réalisation d'un **pilote de voiture autonome embarqué** sur une maquette (voiture en modèle réduit).

# Notre projet

## Pourquoi ?

Self-driving systèmes uniquement disponibles sur les véhicules récents directement implantés dans les voitures concernées.

## Solution

Notre projet est donc de créer un pilote intelligent qui pourrait s'**adapter sur des voitures ne disposant pas d'origine de système de ce type** (sans modification de la voiture).

# **Fonctions de plus haut niveau**

- Réaliser un prototype d'un conducteur autonome qui détecte deux lignes blanches de part et d'autres de la voiture
- Motorisation de la direction de la voiture commandée par un algorithme afin de suivre la route

## **Contraintes**

- La voiture doit se déplacer sur un circuit plat, en intérieur, avec un sol foncé et des lignes blanches
- La distance de la voiture aux lignes doit être telle que la voiture reste dans sa voie

# **Fonctionnement du groupe**

- **Chef de projet** : Loïc Fournier
- **Responsables Code** : Titouan Daniélou, Nicolas Dode et Julien Fontan
- **Responsables Mécanique & Électronique** : Loïc Fournier et Antoine Jourdain

02

# Informatique

# Architecture globale

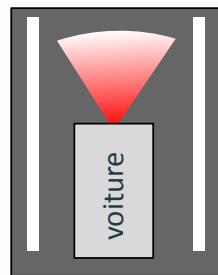
## Projet

```
|- camera
  |- camera_stream.py
|- detection
  |- lane_detection.py
  |- parameter_adjuster.py
|- decision
  |- lane_follower.py
|- motor
  |- motor_controller.py
|- main.py
|- config.json
```

# Détection de lignes

## Prétraitement

raspberry  
1 caméra



Acquisition  
images en  
temps réel

Réduction de  
résolution

Conversion  
niveaux de  
gris

Flou  
gaussien

3280x2464 px  
 $\rightarrow$  640x480 px

Réduction  
du bruit

Image traitée



Sélection  
région d'intérêt

Détection de  
contours

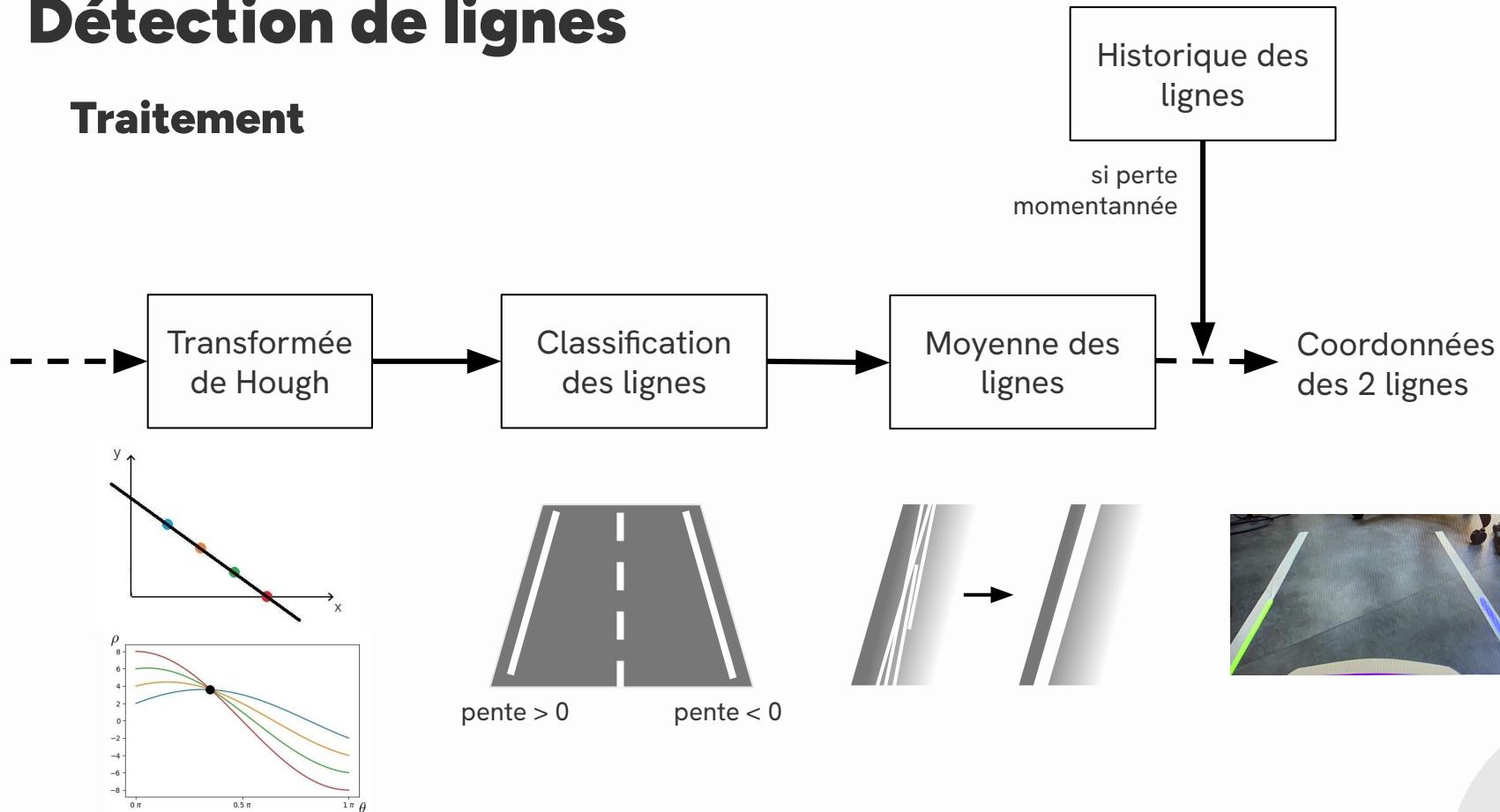
Seuillage  
binaire

Filtre de Canny

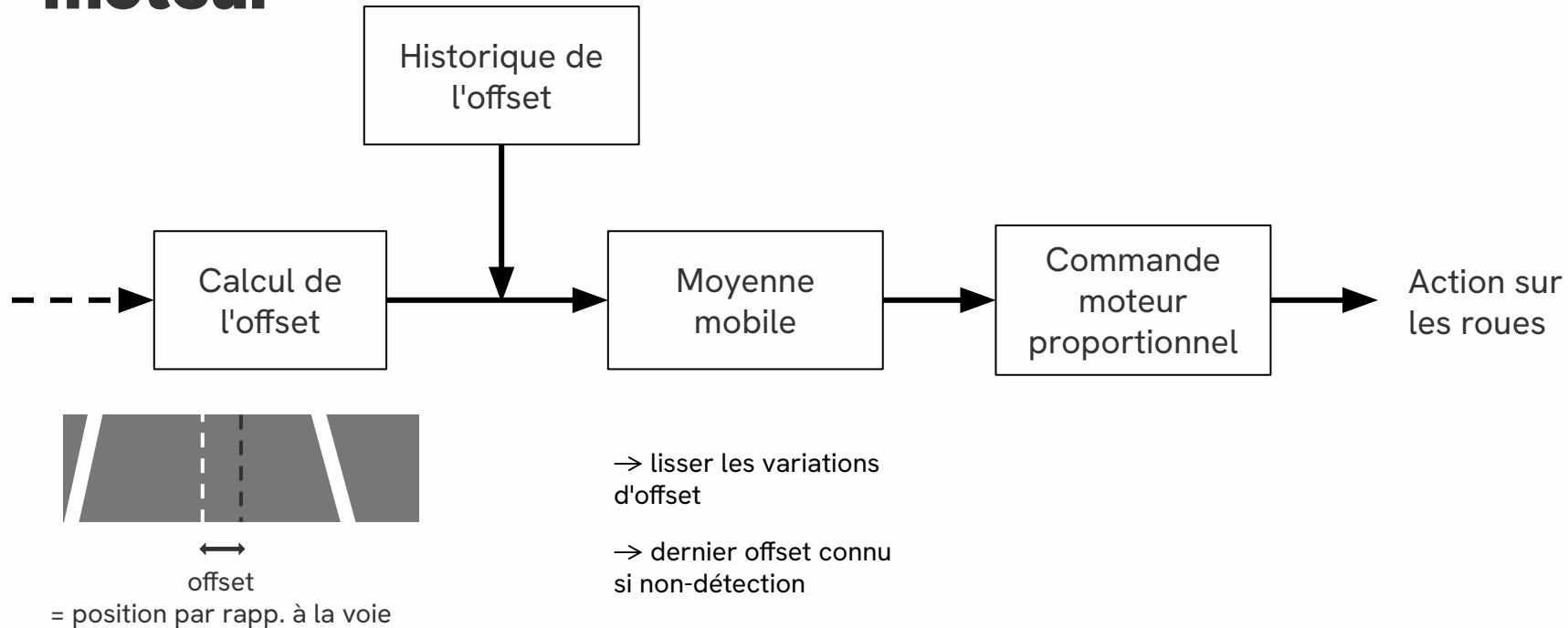
nécessite l'ajustement  
de paramètres

# Détection de lignes

## Traitement



# Prise de décision & Commande moteur



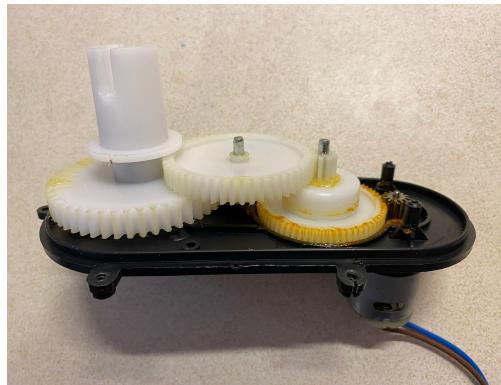
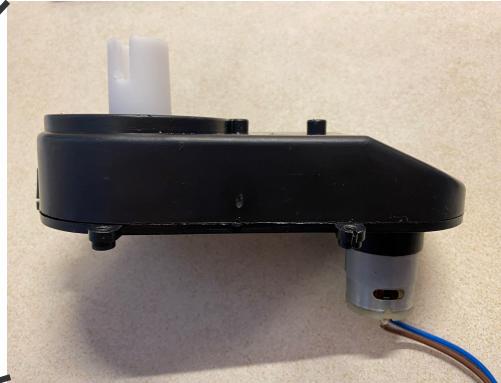
03

# Mécanique et électronique

Motorisation du système de direction et  
câblage des éléments électroniques

# Motorisation de la direction

Action sur l'arbre de direction grâce au dispositif déjà présent sur la voiture



# Motorisation de la direction

Transmission de la rotation du système à l'arbre de direction



Collier de serrage utilisé sur le prototype



Collier de serrage plus optimal avec 2 points de contacts mais moins de force de serrage

Matériels utilisés pour l'adaptation : lime, joints en téflon

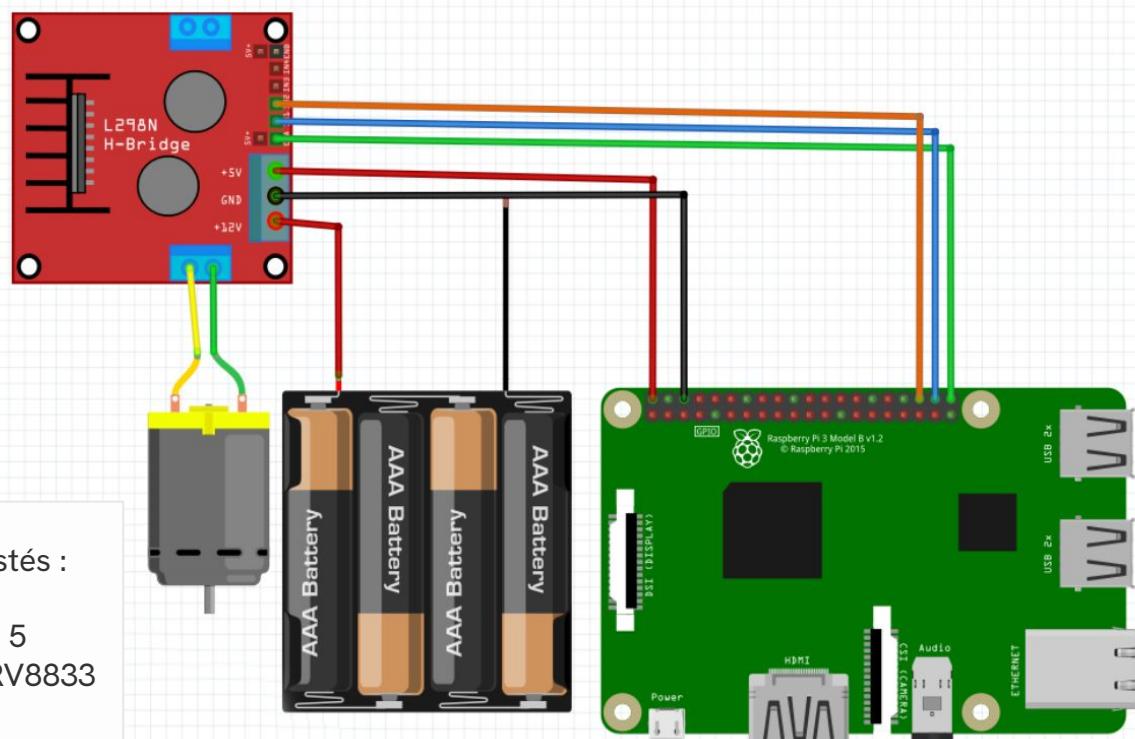
# Mise en place de la motorisation de la direction



Fixation à l'intérieur de l'habitacle : 2 vis,  
support avec rilsan (protection du moteur)

# Câblage des éléments électroniques

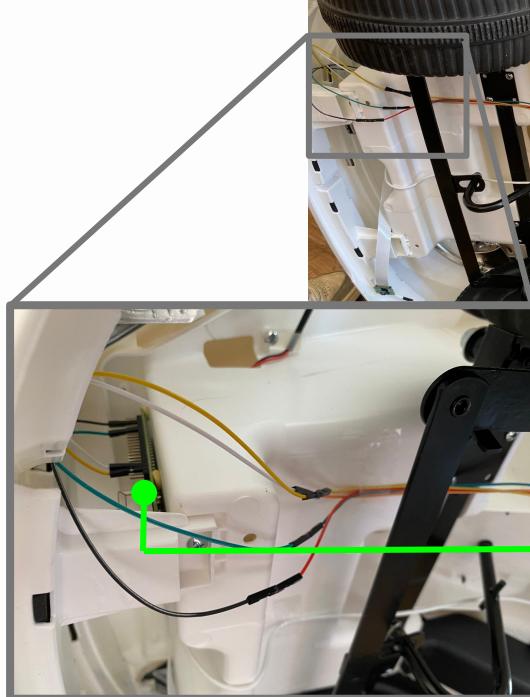
Matériels utilisés : Raspberry Pi 4, Pont en H L298N, Moteur DC, Batterie 6V



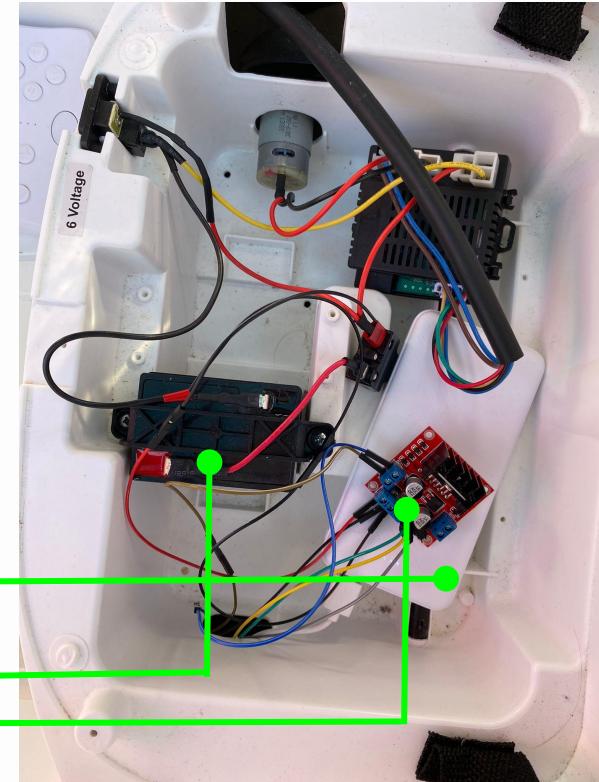
#### Autres matériels testés :

- Raspberry Pi 5
  - Pont en H DRV8833

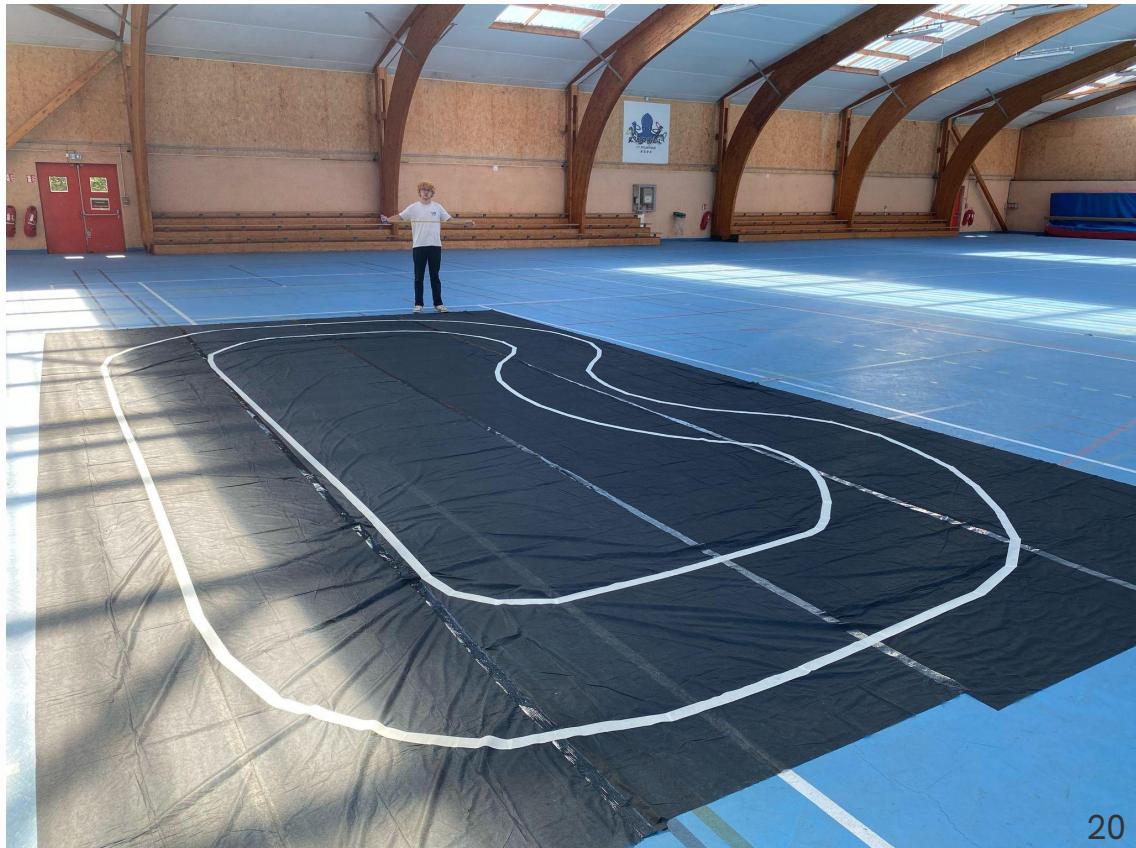
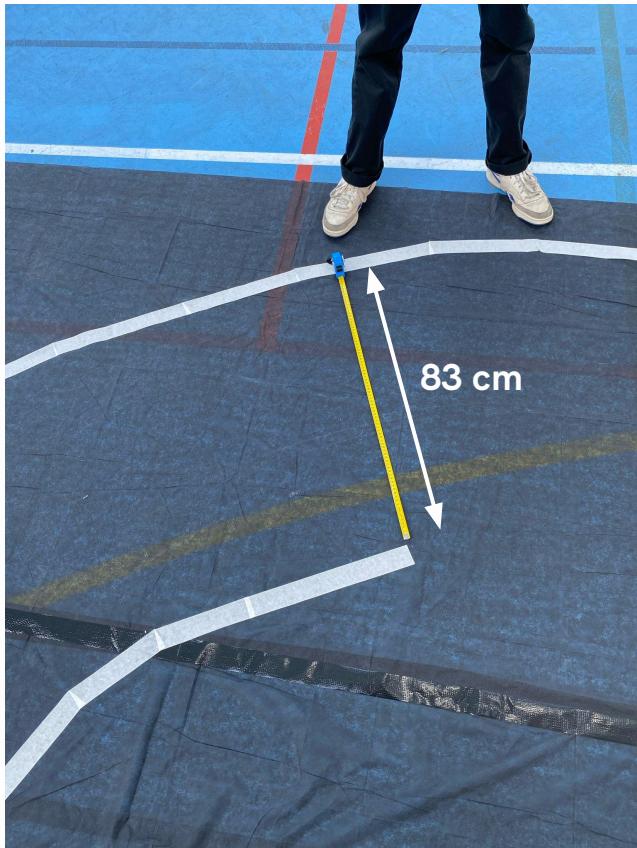
# Câblage du pont en H et de la carte de commande



Batterie 5V 3A  
Raspberry Pi 4  
Batterie 6V  
L298N



# Fabrication du circuit



**04**

# **Difficultés**

# Difficultés rencontrées



## Raspberry Pi 5 -> Caméras indétectables

Les caméras fournies pour notre projet n'étaient pas compatibles avec notre Raspberry Pi 5. En effet, les branchements fonctionnaient mais la raspberry ne détectait pas les caméras. → changement de carte de commande pour une Raspberry Pi 4



## Retard dû aux commandes

- La gestion des commandes de matériel → Système de commandes groupées du Fablab + erreurs dans les demandes initiales. Cela a entraîné des retards pour le codage.
- Non réception du pont en H à deux semaines de la fin du projet



## Pas de retour vidéo pour les tests

Pas de retour en direct de ce que voit la caméra lorsque l'on teste le système complet

# Conséquences



## Retard

Arrivée très tardive à l' étape "Test système complet"



## Position caméra

Difficulté à trouver une position où la caméra détecte les lignes comme voulu



## Dernier recours

Mise en place rapide d'une solution alternative

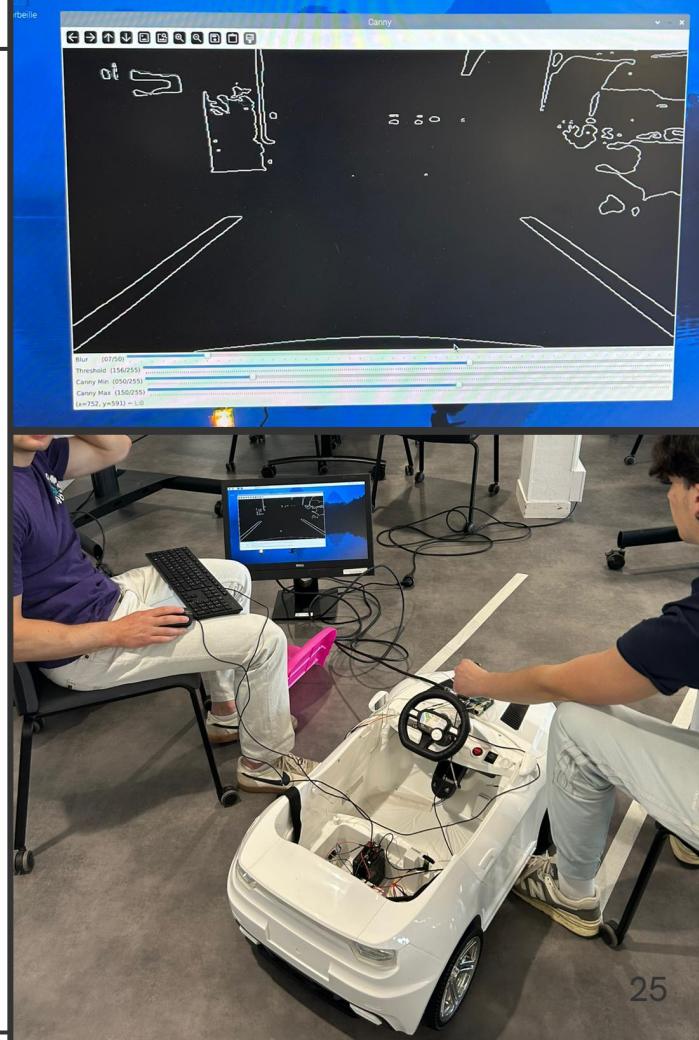
# Test 1

- Caméra fixée sur le pare-choc avant
- Le volant tourne → le code fonctionne
- La voiture ne suit pas du tout les lignes  
→ probable problème de prise de vue



# Test caméra

- Nécessité de brancher écran, clavier et souris à la raspberry pour visualiser ce qu'observe la caméra
- Emplacement précédent de la caméra → angle pas assez large : la caméra perd les lignes de vue dans les tournant serrés
- Besoin de trouver une nouvelle position



# Repositionnement de la caméra

- Siège de la voiture accrochée sur le capot, permettant de surélever la caméra
- Caméra accrochée en haut du siège, orientée vers le bas pour capter au mieux les lignes devant la voiture



## Test 2

- Sur le circuit : la voiture suit parfaitement la route lorsque c'est en ligne droite
- Difficultés sur les virages du circuit  
→ virages trop serrés
- Essai sur la piste d'athlétisme  
(virages plus larges) → la voiture reste toujours entre les deux lignes



**05**

# Résultats finaux obtenus

# Vidéo finale PRONTO 88



# Nos résultats



**Reconnaissance de lignes droites et légèrement courbées**



**Estimation de la distance du véhicule à la ligne**



**Réaction en temps réel sur l'axe du moteur**

# Notre produit final

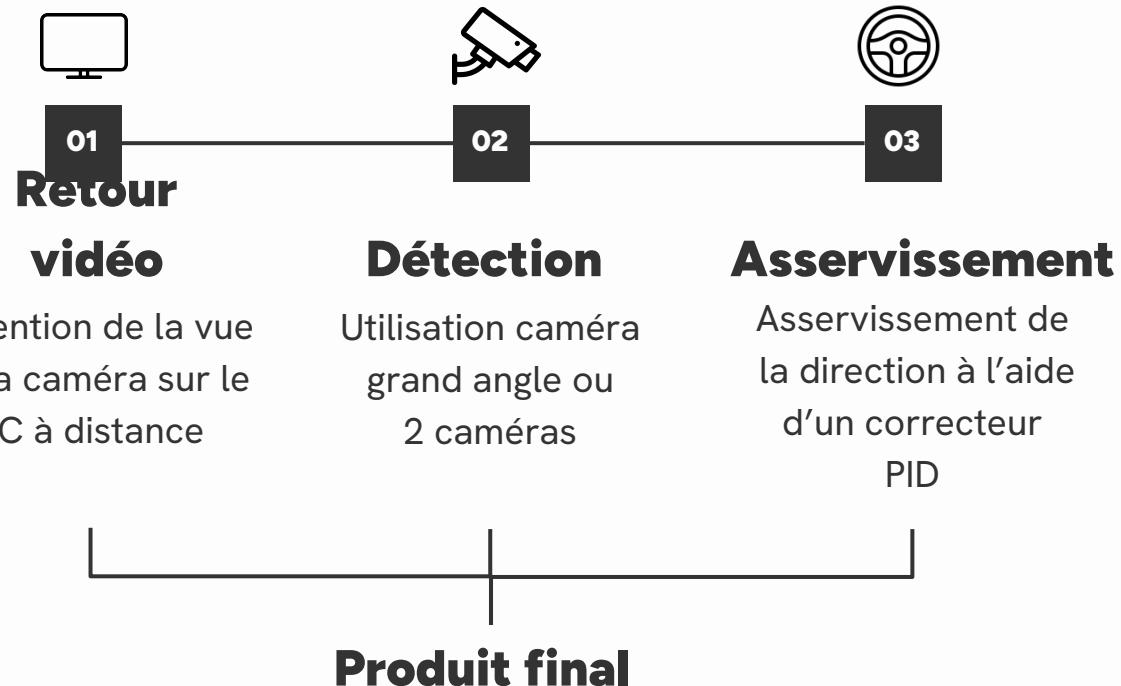


06

# Axes d'amélioration

# Axes d'amélioration

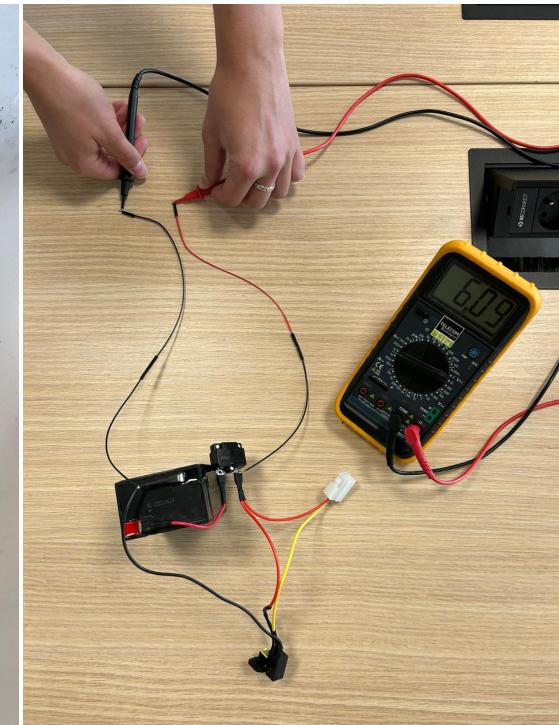
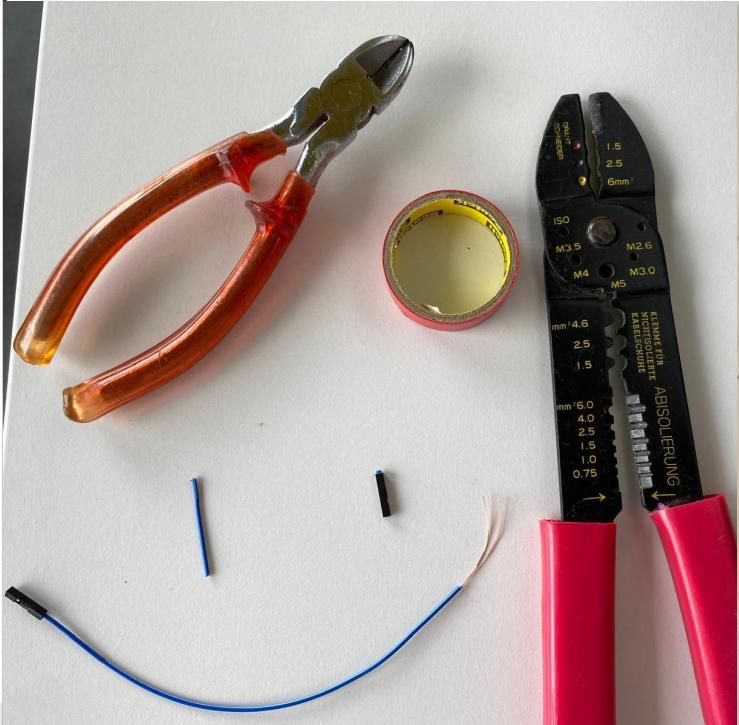
Le produit proposé est un prototype dont certains éléments restent perfectibles. Dans l'optique de l'améliorer, les pistes suivantes seraient à développer.



# Conclusion



# Annexe



Matériel utilisé : câbles, pince coupante, pince d'électricien, scotch d'électricien