

# Suivi de ligne ROS2 Humble

Le suivi de ligne repose surtout sur du traitement de l'image avec OpenCV, et l'envoi de commandes de vitesse au robot. Le code sous ROS1 devrait donc être portable assez directement sous ROS2.

## Environnement de simulation Gazebo

## Modélisation

## Créer un paquet

- Créer un paquet ROS2 python « autonomy » qui implémente follower\_p.py
- Adapter le CMakeLists.txt et package.xml en vous inspirant de :
  - turtlebot3\_behavior\_demos (/tb3\_autonomy/scripts/test\_vision.py)
  - turtlebot3/turtlebot3\_example/turtlebot3\_example/turtlebot3\_position\_control/turtlebot3\_position\_control.py
- Lancer ros2 launch turtlebot3\_gazebo turtlebot3\_circuit\_competition.launch.py
- Démarrer la simulation en plaçant le robot au début de la piste
  - Caméra en vue de la ligne
- Lancer votre nœud python avec ros2 run autonomy follower\_p.py

## Ressources

Pour le suivi d'une ligne blanche :

- [https://github.com/gabrielnhn/ros2-line-follower/blob/main/follower/follower/follower\\_node.py](https://github.com/gabrielnhn/ros2-line-follower/blob/main/follower/follower/follower_node.py)
    - `sudo apt install python3-cv-bridge python3-opencv`
    - ajouter au `~/.bashrc` : `export GAZEBO_MODEL_PATH=~/.turtlebot3_ws/src/ros2-line-follower/follower/models`
  - Le TP2 de robotique de Loïc Cuvillon donné à Telecom Physique Strasbourg
  - ROS1 / OpenCV :  
[https://github.com/osrf/rosbook/blob/master/followbot/follower\\_line\\_finder.py](https://github.com/osrf/rosbook/blob/master/followbot/follower_line_finder.py)
-

Revision #4

Created 4 October 2024 14:52:06 by admin\_idf

Updated 18 October 2024 06:32:06 by admin\_idf