Centre professionnel du Nord vaudois
Filière d'automatique et d'électronique
Modules complémentaires techniques

CPNV - Centre Professionnel du Nord Vaudois

MCT - Modules complémentaires techniqeus

Communication externe

P2213

Rédacteur : Quentin Surdez

Relecture : Rafael Dousse

École : CPNV

Date : Yverdon-Les-Bains, le 6 juin 2022

Table des matières

Table des figures

Liste des tableaux

1 Introduction

Ce document a pour but d'expliquer notre construction de la communication externe du robot. La communication externe se compose du Raspberry PI, de l'Arduino Nano ainsi que d'un objet connecté via un réseau WIFI à notre Raspberry PI. Nous discuterons de nos approches pour utiliser la communication et de comment nous avons résolu les différents problèmes que nous avons rencontrés.

2 Serveur

Nous avons recherché à faire de notre Raspberry PI un serveur permettant d'héberger une web application. Cela permettrait à notre application d'être facilement accessible tant que le Raspberry PI est connecté au même réseau WIFI que l'appareil qui servira de contrôle pour le robot.

2.1 Apache

FIGURE 2.1 – Logo du serveur Web Apache

Apache est la première alternative que nous avons essayé. C'est un serveur Web HTTP qui aurait permis d'héberger notre web application. Nous l'avons peu à peu pris en main et nous avons réussi à utiliser le Raspberry en hotspot. Cependant, Apache restait relativement obscure pour nous. Les différentes possibilités qu'Apache offrait ne correspondait pas à nos besoins. Nous ne souhaitions pas héberger sur le web une application, nous ne souhaitions pas faire de notre Raspberry PI un serveur.

2.2 ROS

Nous avons eu une autre alternative pour faire de notre Raspberry PI hotspot. Le système d'exploitation ROS, pour Robotic Operating System, nous offrait cette possibilité, tout en étant pensé et developpé pour des robots. Nous avons pris connaissance des possibilités que cet OS offrait, comme la création de noeud de communications. Cependant, nous avons vite compris que ROS était principalement fait pour des robots industriels ou de qualité supérieure au notre. Après plusieurs tentatives, nous n'avons pas réussi à installer l'OS sur notre Raspberry PI. Nous nous sommes éloignés de l'idée de paramétré notre Raspberry PI en hotspot.

2.3 Flask

FIGURE 2.2 – Logo du micro-framework Flask

Nous avons découvert Flask via un tutoriel pyimagesearch.com la possibilité d'utiliser le micro-framework Flask pour facilement intéragir avec une web application via notre script Python. Un framework est un logiciel facilitant l'intéraction avec un utilisateur, donc facilitant la création d'applications. Flask est un micro-framework car il ne nécessite aucune librairies ou fichier additionnel pour fonctionner. Des extensions existent pour, par exemple, lui ajouter une fonctionnalité de database avec SQLAlchemy.

Ainsi, Flask nous a permis de communiquer avec une page web et de communiquer l'information donnée sur cette page à notre script Pyhton. L'information récupérée peut être utilisée dans notre programme. Ce protocole est celui que nous avons choisi pour notre communication externe. Nous pouvons nous intéresser de plus près sur la strucutre de notre web app.

2.4 Tableau récapitulatif

TABLE 2.1 – Comparaison des différentes solutions échelle 1 à 10 (1 le moins bon, 10 le meilleur)

	Apache	**ROS**	**Flask**
Installation	10	1	10
Facilité de prise en main	5	1	8
Compatibilité	?	?	10
Total	15	2	28

Ci-dessus, les critères utilisés pour choisir ce avec quoi construire notre communication externe. Le choix s'est fait après avoir réussi un à utiliser OpenCV et Flask facilement l'un avec l'autre. La compatibilité avec OpenCV d'Apache et ROS n'a pas été testé.

3 Web Application

Nous avons pu nous lancer dans la conception de notre Web Application. Sa structure sera détaillée dans la section suivante. Nous avons utilisé HTML/CSS pour pouvoir construire l'architecture du site hébergé sur le Raspberry PI.

3.1 Structure

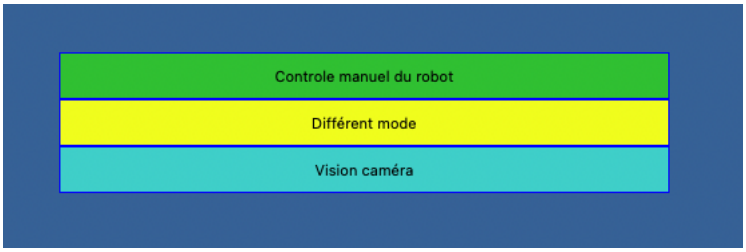

FIGURE 3.1 – Page d'accueil

La page d'accueil possède un menu nous permettant de naviguer dans notre site. Nous avons troix choix à notre diposition.

- Le premier "Controle manuel du robot" nous permet d'accéder à la page de contrôle à distance du robot.
- Le second "Différent mode" renvoit sur la page de choix pour les modes de notre robot.
- Le dernier "Vision Caméra" permet d'accéder à une page ayant un stream direct de la caméra du robot.

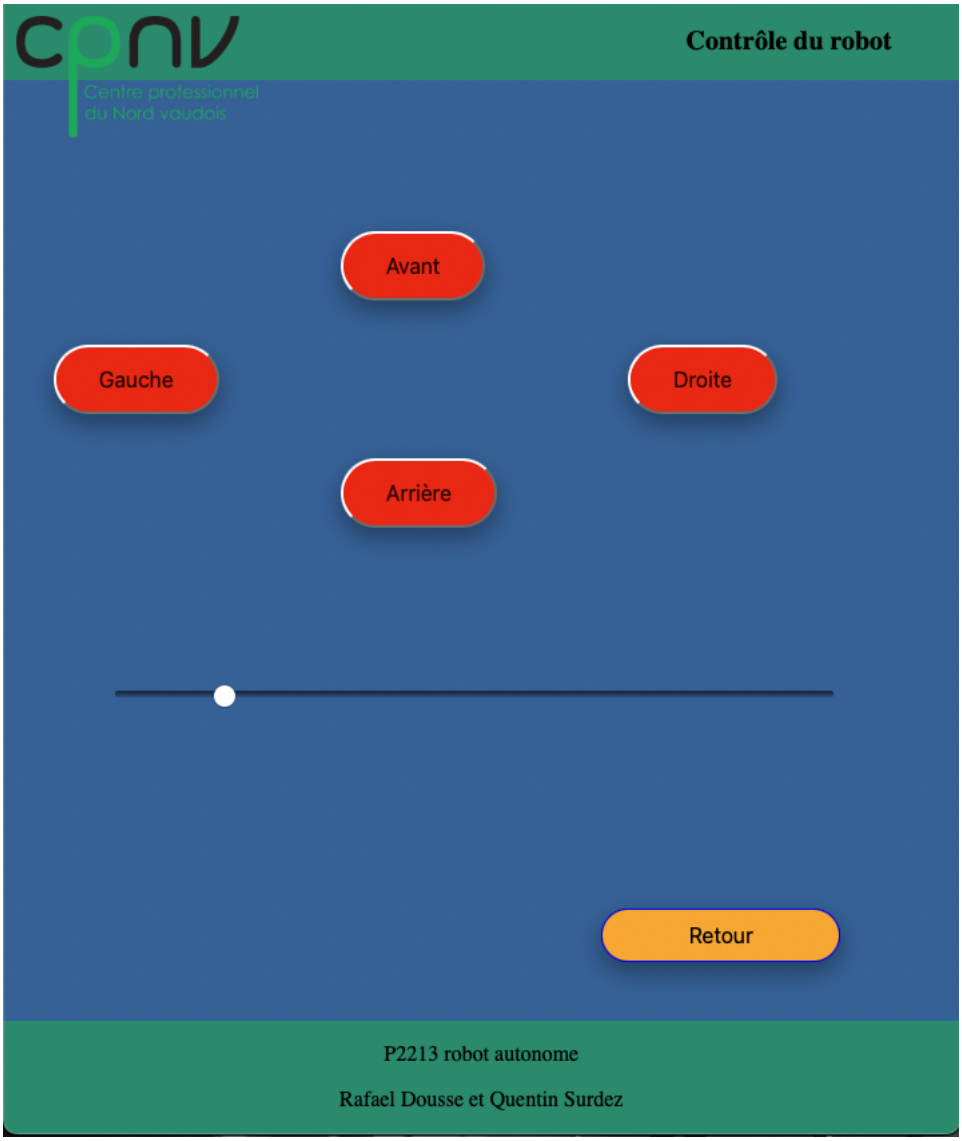

FIGURE 3.2 – Page de contrôle du robot

La page de Contrôle du robot possède 4 boutons disposés en étoiles. Ces boutons permettent d'envoyer des ordres au robot pour se diriger dans les quatre directions d'un plan. Une barre latérale ou verticale selon si vous êtes sur un smartphone ou un ordinateur, permet de contrôler la vitesse du robot. Nous ne savons si elle restera dans l'implémentation finale. Un bouton "Retour" permet de revenir à la page d'accueil.

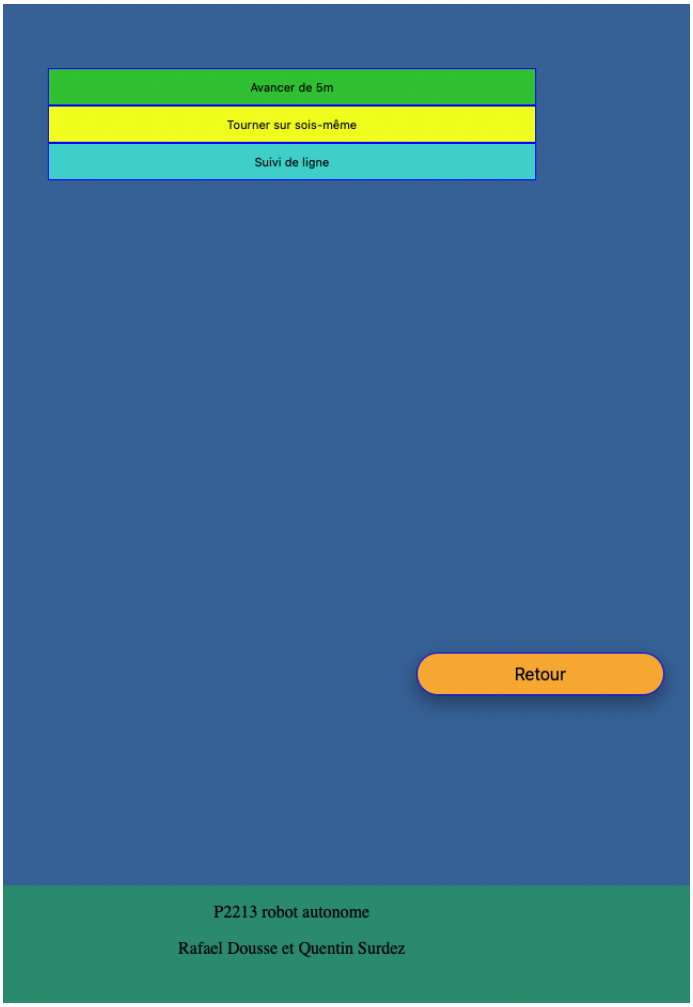

FIGURE 3.3 – Page de choix des modes

La page de choix de mode du robot possède 3 boutons. Ces boutons permettent d'envoyer des ordres pour excéuter les différents modes du robot.

— Le premier "Avancer de 5m" fait avancer le robot de 5m tout droit.
— Le second "Tourner sur soi-même" fait faire un tour au robot sur lui-même.
— Le dernier "Suivi de ligne" fait suivre une ligne au robot.

4 Conclusion

Nous avons pu tester plusieurs méthodes pour accéder à l'hébergement d'une web app sur un Raspberry PI. L'outil nous permettant le plus de possibilité est Flask.

Grâce à cet outil, nous avons pu facilement communiquer entre la page Web et notre Raspberry PI.