CPNV - Centre Professionnel du Nord Vaudois

MCT - Modules complémentaires techniqeus

Rapport de projet (WIP)

P2213

Rédacteur : Quentin Surdez

Relecture : Rafael Dousse

École : CPNV

Date : Yverdon-Les-Bains, le 13 juin 2022

Table des matières

Table des figures

Liste des tableaux

1 Introduction

Notre projet P2213, Robot Autonome, est la construction d'un robot qui puisse remplir plusieurs tâches sans avoir d'intéractions avec l'homme. La tâche principale est le suivi de ligne. Nous avons d'autres tâches comme la reconnaissance de code QR pour effectuer certains programmes ou le contrôle à distance.

Ce document détaillera les choix effectués dans les catégories suivantes :

— Administratif
— Mécanique
— Électronique
— Informatique

2 Administratif

Nous avons d'abord commencé à utiliser les outils de la suite Office365 pour faire notre administration. En cours de route, nous avons fait le choix de changer et de passer à LaTeXpour faire nos documents. Vous trouverez ci-dessous un tableau comprenant nos critères pour ce choix.

	Office365	**LaTeX**
Installation	8	5
Facilité de prise en main	8	3
Documentation	8	8
Produit créé	5	8
Création de nouvelles compétences	3	9
Total	32	33

TABLE 2.1 – Comparaison des différents outils administratifs, échelle 1 à 10 (1 le moins bon, 10 le meilleur)

Nous pouvons observer que les deux outils se valent pour nous. La vraie différence réside dans la création de nouvelles compétences en apprenant LaTeX. Nous continuons d'utiliser la suite Office365 pour faire des graphiques avec Visio ou Excel pour des tableaux avec un nombre conséquent de données.

Cependant, Word a été mis de côté au profit de LaTeX pour la documentation. Cela nous permet d'avoir un plus grand contrôle de notre document et une justification du texte. Qui plus est, Office365 est payant alors que LaTeX est un projet Open-Source.

3 Mécanique

Notre robot a une forte partie mécanique. Nous avons du créer toutes les pièces pour construire un ensemble cohérent qui puisse rouler. Nous avons souhaité avoir une directive de base, créer des pièces nous permettant d'avoir de la place. Une strucure large, pour y intégrer les différents composants que nous souhaiterions intégré.

3.1 Extension arbre moteur

La première pièce à avoir eu plusieurs itérations est l'extension de l'arbre moteur. Nous avions besoin d'un arbre moteur plus grand que ceux déjà présent sur les moteurs. L'arbre du moteur fait 15mm et nous souhaitons que les roues ne rentrent pas en collision avec les autres éléments constituants le système de rotation.

Ce système est composé du moteur, d'une roue compteuse qu'on peut apercevoir dans l'image qui suit, de l'extension de l'arbre moteur et enfin d'une roue. Pour être certain que les roues ne rentrent en collision nous avons choisi de faire un arbre moteur de 45mm de long. Cette longueur nous offre la possibilité d'avoir de larges roues tout en ayant un port-à-faux réduit au minimum.

FIGURE 3.1 – Extension d'arbre moteur ronde

La première tentative a été celle qu'on peut voir ci-dessus. L'extension était ronde avec un embout plus fin. Pour intégrer les roues cela n'était pas stable. Nous nous sommes alors dirigé vers une extension hexagonale. En effet, les roues possèdent d'un côté une chambre hexagonale permettant l'insertion de l'extension.

FIGURE 3.2 – Extension d'arbre moteur hexagonale

L'extension hexagonale s'intègre correctement dans nos roues. Elle est facilement usinable avec la bonne pince.

	Extension Ronde	**Extension Hexagonale**
Temps d'usinage	5	8
Compatibilité	5	10
Maintien	8	8
Total	18	26

TABLE 3.1 – Comparaison des différentes extensions, échelle 1 à 10 (1 le moins bon, 10 le meilleur)

Ce tableau nous permet de voir les avantages offerts par l'extension hexagonale. Cela est pourquoi nous avons choisi de continuer avec des extension hexagonales.

3.2 Roues compteuses

Nous avons faits plusieurs itérations pour les roues compteuses. Ces itérations sont intriquement liées au changement dans le code du PID. Nos roues compteuses ont toujours fait le même diamètre, 45mm, mais le nombre de trous les composant a changé avec le temps.

FIGURE 3.3 – Roue compteuse avec 16 trous

Nous voyons notre roue compteuse avec 16 trous. Nous avons une autre version avec 24 trous. La voici :

FIGURE 3.4 – Roue compteuse avec 24 trous

Nous avons choisi la variante avec 24 trous, car c'est celle qui fonctionne le mieux avec notre code du PID. Elle permet d'avoir, même lorsque le calcul du PID se fait à haute fréquence, d'avoir au moins une incrémentation entre deux calculs.

3.3 Plaque de Fondation

Notre plaque de support, appelée "Foundation", a subi plusieurs itérations. La première les moteurs étaient en dessous et le pont H et les batteries en dessus. Maintenant, les moteurs, les batteries et le pont H sont au-dessus. Cela permet de protéger les composants d'un choc avec un obstacle.

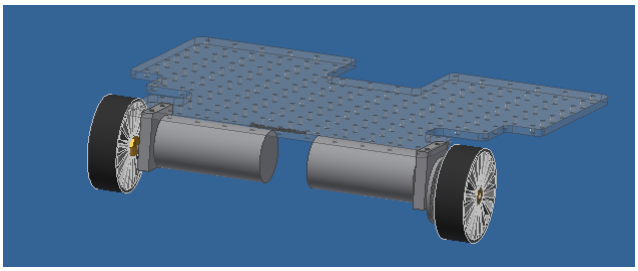

FIGURE 3.5 – Première itération "Foundation"

Ici, nous pouvons voir la première version de notre plaque fondation. Les moteurs étaient en dessous tout comme les capteurs pour les roues compteuses. Cela ne garantit par leur intégrité lors du fonctionnement.

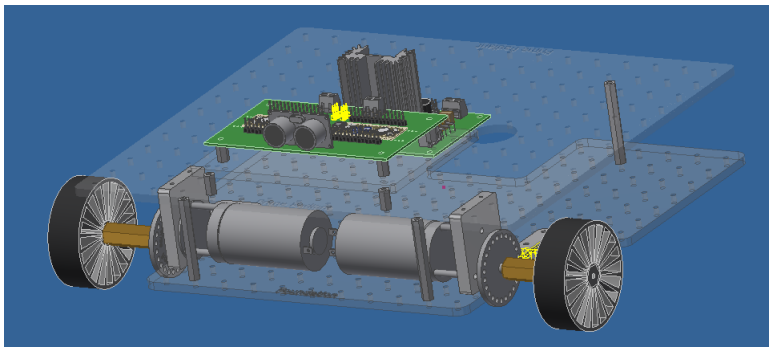

FIGURE 3.6 – Deuxième itération "Foundation"

Ci-dessus, nous avons la deuxième version de la plaque "Foundation". Cette-fois ci les composants sont en-dessus et seules les roues ont une partie qui touche le sol en dessous de la plaque.

	Plaque avec moteurs dessous	**Plaque avec moteurs en dessus**
Protection	5	8
Maintien des composants	5	10
Répartition du poids	3	8
Total	13	26

TABLE 3.2 – Comparaison des différentes plaques de fondation, échelle 1 à 10 (1 le moins bon, 10 le meilleur)

Nous pouvons voir que la deuxième version remplit mieux nos critères. Principalement par le fait que les éléments sont protégés des obstacles externes. La répartition du poids est aussi meilleure, tous les éléments lourds sont le plus bas possible.

3.4 Support moteur

Les moteurs DC que nous avons choisi, soit les MODELCRAFT, étaient assemblés avec une pièce en aluminium permettant de les soutenir sur une plaque. Cette pièce était déjà prête et ne nécessitait pas d'être refaite.

Nous avons, au cours de notre projet, fait une pièce y ressemblant mais en PMMA. Cela nous a permis de revoir quelques points qui ne nous satifaisaient pas sur la pièce en aluminium déjà faite.

	Pièce refaite en PMMA	**Pièce en aluminium**
Compatibilité	10	5
Solidité	7	10
Usinage	5	3
Total	22	18

TABLE 3.3 – Comparaison des différents support de moteurs, échelle 1 à 10 (1 le moins bon, 10 le meilleur)

Nous pouvons voir que le principal atout de la pièce en PMMA est sa compatibilité avec notre système. En effet, nous l'avons créé pour que sa structure soit celle dont nous avons besoin pour notre projet. Ainsi, ses trous pour la visser sur les plaques sont alignés avec les trous des plaques et sa hauteur correspond à la hauteur que nous souhaitons pour l'espacement entre les plaques. Son usinage prend du temps, comme les trous doivent être alignés.

La pièce en aluminium est surtout plus robuste que la pièce en plastique. Son usinage prend plus de temps et de compétence comme une CNC doit être utilisée. Notre système n'a pas pour but d'être robuste aux chocs et nous ne voyons donc pas l'intérêt d'utiliser cette pièce.

3.5 Support Caméra

Le support pour la caméra possède plusieurs contraintes importantes. La caméra doit vibrer le moins possible pour avoir une qualité d'image suffisante pour pouvoir la traiter. Nous avons besoin d'une pièce robuste qui pourra répondre aux différentes contraintes que le support possède.

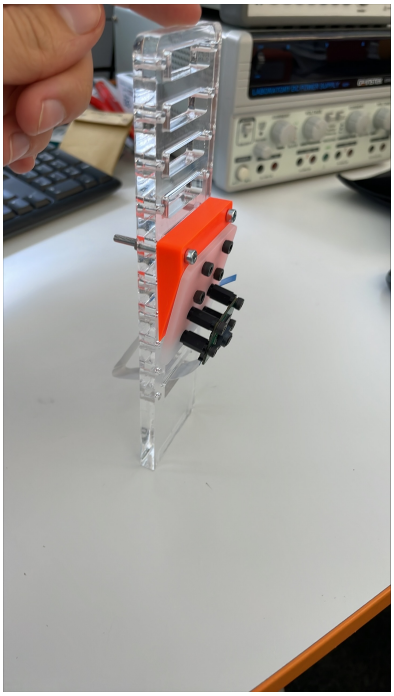

FIGURE 3.7 – Support CameraPI

Nous pouvons voir ici une première pièce. Cependant, elle est longue. Donc plus propice à bouger par les vibrations. Le module pour fixer la caméra est ajouté sur l'échelle et cela peut créer des vibrations supplémentaires.

Nous avons donc décidé de la faire plus courte et d'une seule pièce pour éliminer ces sources de transfert de vibrations. Pour ce faire nous avons envoyé à l'imprimante 3D.

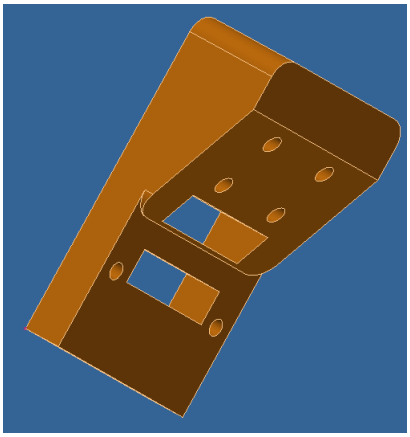

FIGURE 3.8 – Pilier CameraPI 3D

Les contraintes ont changé le développement de la pièce pour qu'elle s'y adapte. Nous avons donc décidé de la fusionner en une pièce et de l'imprimer en 3D.

4 Électronique

Nous avons réalisé plusieurs PCBs différents pour notre projet. Ils ont tous été faits par le CPNV sauf un qui est le PCB du pont en H. Une explication succinte accompagnera chaque présentation de PCB et une explication de nos choix concernant les différents composants.

4.1 Pont en H

4.2 PCBs Alimentation

4.3 Support Capteurs

5 Informatique

Nous avons plusieurs programmes ayant des rôles majeurs. Nous avons tout d'abord le programme du microcontrôleur qui calcule le PID à une fréquence de 250Hz. Ensuite, nous avons le programme qui gère la WebApp depuis le Raspberry. Ce programme en appele plusieurs autres qui gèrent les fonctions en lien avec la CameraPI.

La CameraPI communique par bus CSI avec le Raspberyy PI et le Raspberry PI communique avec le microcontrôleur par bus I2C. Nous allons commencer par expliquer les choix faits dans le programme du PID, ensuite nous discuterons des programmes qui gèrent la CameraPI pour finir sur le code de la WebApp.

5.1 PID

Nous avons besoin d'asservir nos moteurs avec un asservissement par PID. Nous souhaitons que le robot avance tout droit lorsqu'on le lui en donne l'ordre. L'équation du PID en fonction du temps et de l'erreur calculée est la suivante :

$$u = k_p e + k_i \int e \cdot dt + k_d \frac{de}{dt}$$

FIGURE 5.1 – Équation du PID

Nous avons utilisé cette équation pour mettre en place un calcul du PID dans notre système. Le dt a été set up via des interruptions attachées. Une clock externe permet d'envoyer un signal à une fréquence de 250Hz pour calculer le PID.

Nous avons deux possibilités pour déterminer les différents facteurs. Nous pouvons soit utiliser la méthode de Ziegler-Nichols ou alors la méthode empirique. Voici un tableau pour les départager :

	Ziegler-Nichols	**Empirique**
Justesse des facteurs	8	5
Facilité d'application	3	8
Total	11	13

TABLE 5.1 – Comparaison des différentes méthodes d'acquisition des facteurs PID, échelle 1 à 10 (1 le moins bon, 10 le meilleur)

Les méthodes se valent entre elles. Cependant, la méthode via Ziegler-Nichols n'a pas été tout à fait comprite pour l'appliquer dans notre cas avec des moteurs DC. C'est pour cela que nous avons choisi de continuer avec la méthode empirique.

Vous trouverez de plus amples informations concernant le code et notre utilisation des interruptions à ce fichier : Explication code Arduino

La méthode que nous avons choisie est celle empirique. Pour un détail de toutes les données utilisées et les graphiques correspondant veuillez vous référer au document suivant : Rapport PID

5.2 CameraPI

Pour utiliser la CameraPI nous avons dû commencer à apprendre à utiliser OpenCV. Open Source Computer Vision Library, est un logiciel open source de computer vision et de machine learning. Pour pouvoir l'installer sur un Raspberry PI, se référer à ce document : Guide d'installation à OpenCV

La découverte de ce logiciel s'est faite en plusieurs étapes :

— Son téléchargement, qui est plutôt difficile, si nous le téléchargeons depuis la source
— La première prise en main grâce à des tutoriels
— Création de projet correspondant à notre cahier des charges avec l'aide de tutoriels
— La prise en main des fonctions utilisées dans les projets pour les intégrer à notre code

La plupart des tutoriels ont été trouvé sur le site suivant : pyimagesearch.com

Le choix d'utiliser OpenCV s'est fait pour plusieurs raisons, dont plusieurs sont détaillées ici : Choix de la caméra. Cependant, voici une liste exhaustive de ce qui nous a fait choisir ce logiciel :

— Il permet de remplir notre cahier des charges, avec même quelques ajouts de fonctionnalités
— Il est difficile d'approche, mais permet de développer une compétence intéressante
— Il nous permettra de faire du machine learning dans notre avenir

On peut voir que les raisons sont surtout personnelles. Un projet futur personnel serait d'utiliser Keras et TensorFlow pour pouvoir apprendre le machine learning.

5.3 WebApp

La création de la WebApp s'est faite grâce au micro-framwork Flask. Nous avons construit le WebApp grâce aux langages HTML/CSS. Comme pour les autres inconnues de ce projet nous avons d'abord commencé à regarder des vidéos pour ensuite s'approprier le langage et le comprendre.

Les raisons qui nous ont amenées à choisir Flask sont les suivantes :

— Son installation est simple et rapide
— Il permet un transfert d'informations facilité entre la WebApp et le script python
— Il est compatible avec nos autres librairies, notamment OpenCV
— Il est un micro-framework et donc très léger

Il y a une hiérarchie importante lors de la communication pour établir la WebApp. Le script python appelle Flask et certaines de ses fonctions pour créer les pages Web. Il est donné un template HTML et ce dernier va chercher dans un dossier static notre code CSS. Ensuite, les infos de la WebApp sont renvoyés au script qui recommence une boucle. Voici un schéma pour représenter cette communication :

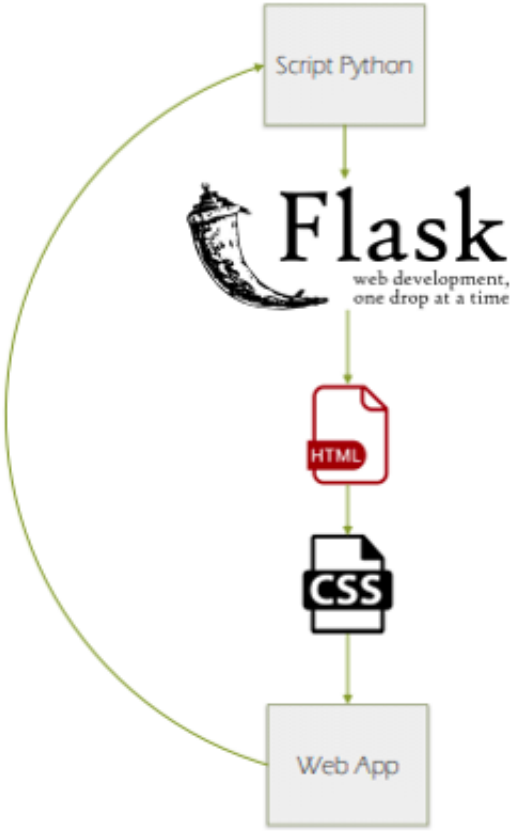

FIGURE 5.2 – Schéma de la WebApp

La WebApp permet ainsi une communication à distance via un autre appareil connecté au même réseau que le serveur, soit le Raspberry PI. Pour plus d'informations à propos de la communication externe : Communication externe

5.4 Communication

La communication est un point important de notre projet. En effet, c'est grâce à la communication I2C que notre Raspberry PI peut échanger des informations avec l'Arduino Nano Every. C'est aussi une communication via WIFI qui nous permet de contrôler le robot à distance.

Chaque communication possède un document qui lui est lié voici les liens : Communication externe & Communication interne

Nous allons donné les points clefs pour chaque communication et les raisons du choix de ces modes de communication.

Le bus de communication I2C permet une communication rapide entre un maître, Raspberry PI, et un ou plusieurs esclaves, les Arduinos. Ce bus de communication est rapide, soit 100kbits/s, cela permet d'avoir une transmission quasi instantanée de l'information. Les librairies d'un côté comme de l'autre sont simple à utiliser et se mette rapidement en place.

- Flexibilité pour ajouter des esclaves potentiels
- Rapide
- Facile à installer

Le protocole de communication WIFI est un protocole commun à presque tous les appareils connectés. Il est facile de se connecter à un WIFI avec nos téléphones ou nos ordinateurs. Ce protocole est sélectif. Il est construit de tel sorte que par défaut un password est requis pour avoir une connexion à un réseau commun. Cela permet d'éviter les interférences d'autres appareils.

- Protocole sécurisé par défaut
- Facile à mettre en place
- Facile à utiliser

6 Améliorations

Les améliorations de notre projet sont réparties avec la même structure que le rapport lui-même. Ce sont des observations que nous avons faites au cours du développement. Ce sont certaines choses que nous aurions aimé faire, mais soit le temps nous a contraint, ou nos décisions ne nous permettaient pas de les faire. Certaines améliorations sont aussi au-delà de nos compétences actuelles et nous demanderaient beaucoup trop de temps.

6.1 Mécanique

La mécanique est le point dans lequel nous avons investi le moins de temps dans le projet. En effet, nous souhaitions nous diriger principalement sur l'aspect électronique et programmation de notre projet. Ainsi, il existe des améliorations au niveau de la mécanique de notre projet.

Premièrement, notre robot n'est pas fermé. Le fait que sa connectique et ses moteurs ne soit pas protégé par une boîte ou des parois fait en sorte que les risques de dégâts restent relativement grand.

Nous avons aussi des pièces soutentant les moteurs dont la hauteur ne correspond pas à nos besoins. En effet, après l'ajout de nouvelles roues, nous avons dû réhausser le deuxième étage. Cela fragilise la structure de notre robot.

Le maintien des batteries n'est pas bon. Une cage d'entretoises les maintiennent dans un certain espace et les empêche de trop bouger. Cependant, elles sont libres dans cette cage d'entretoises, donc bougent. Ensuite, l'accès aux batteries est simplement mauvais. Il faut soit avoir une grande dextérité pour extraire les batteries, ou dévisser les plaques du haut pour y accéder.

Un vérin électrique pour faire bouger l'avant ou l'arrière du robot aurait aussi été grandement apprécié

6.2 Électronique

Notre électronique souffre d'un manque d'organisation dans les connectiques inter-PCBs. Notre connectique reste mal organisé et brouillon dans la structure du robot.

Nous aurions souhaité avoir un pilone de câble déservant les alimentations aux différents étages du robot. Une autre amélioration électronique serait d'organiser les connections inter-PCBs avec des connectiques dédiés et des câbles plats. Cela permettrait de s'assurer que les différents chocs que le robot subit ne défasse pas la connexion par les câbles. Cela permettrait aussi à l'ensemble d'être organisé et propre.

Notre PCB gérant la communication en I2C possède une particularité dont nous ne connaissons pas la cause. Au niveau de l'adaptateur de niveau logique, le GND du Raspberry PI ne

doit pas être branché pour que le tout fonctionne correctement. Le problème n'est pas connu, il faudra donc s'y attarder si le PCB est réutilisé.

6.3 Informatique

Notre installation d'OpenCV est lourde et lente, soit 4Go et 2-4 heures d'installation avec une optimisation de la valeur des SWAPFILES. Une amélioration potentielle serait de trouver une installation moins gourmande en temps et aussi plus légère. Pour ce faire, il faudrait savoir exactement les modules dont nous avons besoin pour nos différents programmes.

Une deuxième améiorations concernant notre code serait une optimisation de la régulation. En effet, nos différents facteurs PID peuvent être mieux déterminés par une méthode autre que celle empirique. Il serait nécessaire de faire des tests plus poussé pour avoir des graphiques représentant l'évolution de la vitesse et de la valeur de contrôle. Les facteurs pourraient être alternés en fonction. La fréquence à laquelle le calcul est fait devrait aussi faire l'objet d'une étude poussée pour déterminer la meilleure avec nos restrictions.

7 Conclusion

Les différentes décisions prises ont été dans le but de remplir notre cahier des charges. Nous avons cherché à trouver des solutions pour pouvoir accomplir les différents objectifs que nous nous étions fixés en début de projet.

Il y a certaines décisions qui auraient pu être mises en place dès le début pour que nous ayons une direction à suivre qui soit plus organisé, comme un tube central autour duquel les alimentations peuvent donner du courant.