

Rapport final de projet SAE 3.02

Par Elouan Pierrot et Elowan Gouez

I. Présentation du projet

Nous avons décidé dans le cadre du projet SAE 3.02 de choisir le projet de programmation d'application communiquant avec un robot en Python. Plus précisément, nous avons choisi d'utiliser le Robot Chenille 10. Celui-ci est accompagné d'une interface d'entrée sortie Raspberry que nous utiliserons pour la communication tout au long de ce projet afin de lui envoyer des ordres d'exécution depuis nos ordinateurs personnels.

II. équipe

Notre équipe est formée de deux membres : Elouan Pierrot et Elowan Gouez, dont les rôles sont égaux, nous travaillons tous les deux au développement et l'amélioration de l'application et mise en forme des rapports de projet.

III. Organisation

Nous avons commencé le projet le 20/11/23 et profitons d'un temps de travail total de 48 heures (12 heures encadrées et 36 heures en autonomie).

IV. Objectifs

- **Contrôle déplacements à distance**

Contrôle manuel du robot avec UDP.

- **Programmation objets pour l'application**

Utilisation de classes pour faire fonctionner l'application.

- **Client unique**

Seule une personne à la fois contrôlera le robot.

- **Interface graphique**

L'application aura une interface graphique permettant notamment de configurer les paramètres réseau.

- **Sérialisation - Dé-sérialisation Json**

Utilisation du format json pour échanger avec le robot.

VI. Évolutions possibles

- **Réponse aux obstacles**
possibilité de détecter lorsque le robot est stoppé par un obstacle puis de corriger sa trajectoire.
- **Surveillance et adaptation des roues**
Possibilité de surveiller de façon permanente la vitesse des roues et d'adapter leurs vitesses afin d'éviter la dérive.
- **Utilisation de Pyaudio**
Possibilité de lancer des sons grâce au module Pyaudio
- **Utilisation d'une manette**
Possibilité de contrôler le Robot grâce à une manette

V. Rapports

Les objectifs que nous avons atteint lors de ce projet sont les suivants :

- Le contrôle à distance par UDP avec la mise en place du robot en tant que serveur UDP et utilisation d'un client unique.
- Mise en place d'une interface graphique IHM pour la partie connexion au robot nous permettant d'entrer l'adresse IP et le port du robot auquel on souhaite se connecter.
- Contrôle du robot par l'utilisation d'une manette Xbox, plus précisément nous avons fait en sorte que la direction robot soit contrôlable par un Joystick en lisant les positions X et Y de celui-ci. Nous avons aussi fait en sorte que le robot soit éteignable grâce à un des boutons de la manette.
- Gestion vitesse, grâce à la gâchette droite de la manette (Right Trigger) de la manette grâce à une valeur de vitesse comprise entre 0 et 100.
- Utilisation de classes pour l'IHM, le contrôle de la manette, le serveur et client UDP.

VI. Gantt et déroulement du projet

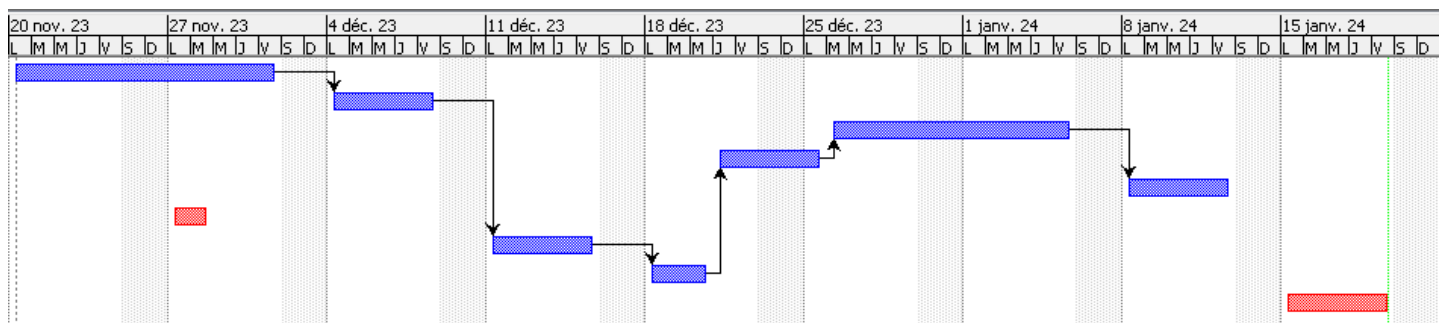
Nous avons débuté le projet le 20 Novembre, nous avons mis une dizaine de jours à mettre en place les paramètres d'initialisation du robot, la connexion sans point de connexion externe et à le prendre de manière générale en main. Au même moment, nous avons rédigé notre cahier des charges afin de présenter nos objectifs pour ce projet.

Puis nous avons commencé à mettre en place les première fonctionnalité du robot, qui ont été un serveur et un client TCP afin d'envoyer des paquets au robot et un contrôle du robot par clavier. À la fin de leurs mises en place et de quelques essais nous avons d'abord décidé de remplacer le contrôle par clavier par un contrôle à la manette, mais un contrôle à la manette, mais nous nous sommes rendu compte que cela ne fonctionnait pas très bien avec une communication en TCP ce qui nous a poussé à convertir nos programmes vers une communication en UDP nous faisant donc repartir de zéro peu avant le début des vacances de Noël. Nous avons adapté nos programmes pendant les vacances de Noël et à la rentrée, quelques jours après nous avons ajouté la gestion de la vitesse par la manette en utilisant une de ses gâchettes.

Pour finir nous avons rédigé le rapport final du projet jusqu'à peu après la soutenance de présentation du projet. Ce qui nous fait une fin technique du projet le 12 Janvier et du rapport de projet le 19 Janvier.

Tableau du Gantt

| | 🕒 | Nom | Durée | Début | Fin | Prédécesseurs |
|---|---|---|-----------|----------------|----------------|---------------|
| 1 | | Mise en place du Robot | 10 jours? | 20/11/23 08:00 | 01/12/23 17:00 | |
| 2 | | Création du client | 5 jours? | 04/12/23 08:00 | 08/12/23 17:00 | 1 |
| 3 | | Conversion vers Contrôle à distance UDP | 9 jours? | 26/12/23 08:00 | 05/01/24 17:00 | 4 |
| 4 | | Contrôle par manette | 3 jours? | 21/12/23 08:00 | 25/12/23 17:00 | 8 |
| 5 | | Gestion de la vitesse | 5 jours? | 08/01/24 08:00 | 12/01/24 17:00 | 3 |
| 6 | 📅 | création du cahier des charges | 2 jours? | 27/11/23 08:00 | 28/11/23 17:00 | |
| 7 | | Contrôle par TCP | 5 jours? | 11/12/23 08:00 | 15/12/23 17:00 | 2 |
| 8 | | Contrôle par clavier | 3 jours? | 18/12/23 08:00 | 20/12/23 17:00 | 7 |
| 9 | 📅 | Création du rapport Final | 5 jours? | 15/01/24 08:00 | 19/01/24 17:00 | |



VI. Présentation des fichier de programmation

Nous avons organisé le fonctionnement de notre projet en trois fichiers/programmes. Le premier se trouvant dans le robot, contenant une classe serveur UDP, en écoute sur le port 5000 avec des fonctions servant à recevoir les instructions du client UDP et à traiter ces informations et les convertir en allumage des différentes LEDs du robot servant aux mouvements et la vitesse. Les informations reçues par le serveur est une liste contenant la positions des différents boutons de la manette sous formes de valeur entre 0 et 1 pour les boutons, -100 et 100 pour le joystick et entre 0 et 100 pour la gâchette.

Le deuxième et troisième se trouvent au sein de l'ordinateur du client, l'un est le code servant à lire les commandes de la manette à partir d'une bibliothèque inputs. Le deuxième est le programme contenant les classes client UDP (permettant l'envoi des informations de commandes de la manette transformées au serveur UDP) et IHM (initialisant les paramètres de connexions au serveur sous forme d'une interface graphique pré rempli et modifiable) et d'une fonction contrôle servant à convertir les données des commandes de la manette en des entier afin de réduire la taille des paquets envoyés par UDP. L'envoi des données est séparé par des Time sleep afin de ne pas surcharger d'informations le serveur UDP.