**Travail pratique 3 - Contrôle d'un robot**

**420-22P-SI**

**Hiver 2023**

**Informations préliminaires**

- Date de remise de la description du travail : Lundi le 24 avril 2023.

- Date de remise du travail complété : Jeudi le 11 mai 2023, avant 17h00.

- Représente 50% de la note finale.

- Travail personnel.

- Environnement de programmation : Processing, Arduino et Tinkercad.

- Vous devez remettre sur **Léa** le dossier de votre projet, sous l'énoncé **Travail pratique 3**.

**Description générale**

Effectuez, à l'aide d'une interface graphique en Processing, le contrôle des mouvements du robot mobile **Cherokey 4WD** de DFRobot.

Les mouvements du robot sont contrôlés à l'aide de ses 4 roues motorisées.

Sur le robot est embarqué une carte **Arduino Uno WiFi**, laquelle contrôle les mouvements du robot.

Le lien avec le robot est effectué via une connexion Wi-Fi.

Votre interface graphique doit permettre de contrôler les mouvements suivants. Les roues doivent toujours tourner à pleine vitesse lors d'un déplacement, à l'exception de celles (côté droit ou côté gauche) devant tourner à une vitesse inférieure ou nulle pour permettre un virage :

- Avancer en ligne droite.

- Reculer en ligne droite.

- Arrêter.

- Pivoter sur la droite.

- Pivoter sur la gauche.

- Avancer en tournant vers la droite.

- Avancer en tournant vers la gauche.

- Reculer en tournant vers la droite.

- Reculer en tournant vers la gauche.

L'intensité (ou rayon de courbure) d'un virage peut aller de 1 (faible virage) à 5 (virage serré), avec un incrément de 1.

Le protocole à respecter entre Processing et Arduino est composé de deux octets, le premier étant le mouvement à réaliser, le second l'intensité des virages.

Si un mouvement rectiligne ou un arrêt est demandé, deux octets doivent quand même être transmis au robot afin de respecter le protocole, mais le second octet ne sert à rien.

Le contrôle de votre robot doit être possible à partir de plusieurs clients connectés.

Afin de sauver du temps, d'économiser les piles du robot, et d'avoir la possibilité de travailler à l'extérieur des heures du cours, débutez votre système de contrôle en simulant certains traitement, tels que décrits ci-dessous.

Lorsque ces simulations sauront répondre aux exigences d'une partie de votre système de contrôle, des modifications mineures pourront être effectuées afin que les mouvements du robot physique puissent être contrôlés via le réseau.

**Étapes à réaliser**

**1. Tinkercad et le robot virtuel (10% du fonctionnement)**

Utilisez la modélisation de votre robot sur Tinkercad. Simulez l'envoi et la réception d'octets de contrôle (selon le protocole) en utilisant le Moniteur série (voir **Simulation d'une réception d'octets** sur Tinkercad).

**2. Processing et l'interface de contrôle (10% du fonctionnement)**

Créez votre interface de contrôle. En fonction des mouvements désirés, simulez l'envoi d'octets de contrôle (selon le protocole) en affichant ces derniers dans la console d'affichage.

Les étapes précédentes doivent être complétées avant d'avoir accès au kit Arduino contenant la carte Uno WiFi.

**3. Processing et Arduino (10% du fonctionnement)**

Ajoutez la programmation réseau. Simulez les mouvements de votre robot et l'intensité de ses virages à l'aide de LEDs, tel qu'illustré ci-dessous.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Avancer** | **Reculer** | **Arrêter** | **Pivoter sur la droite** | **Pivoter sur la gauche** |
|  |  |  |  |  |
| **Avancer en tournant vers la droite** | **Avancer en tournant vers la gauche** | **Reculer en tournant vers la droite** | **Reculer en tournant vers la gauche** |  |
|  |  |  |  |  |

L'étape précédente doit être complétée avant d'avoir accès au robot physique.

**4. Processing, Arduino et le robot physique (45% du fonctionnement)**

Demandez au professeur d'installer votre carte Arduino Uno WiFi sur votre robot. Modifiez votre code afin que les mouvements de votre robot physique puissent être contrôlés.

**5. Ajout d'une LED au robot physique (5% du fonctionnement)**

Ajoutez une LED sur votre robot. Afin que votre robot puisse indiquer qu'il est connecté au réseau Wi-Fi, une LED doit être allumée aussitôt la connexion réalisée. Si cette connexion est perdue, la LED doit être éteinte avant que le robot ne tente de la rétablir.

**6. Transmission de données vers Access (20% du fonctionnement)**

Demandez au robot de transmettre, pour chaque mouvement réalisé, des données vers un serveur Processing, afin qu'elles soient insérées dans une base de données Access. Les données à transmettre, pour chaque mouvement réalisé, sont : heure, minute, seconde, IP du poste demandeur, mouvement réalisé, intensité du virage (fait partie du protocole même si aucun virage). Un ensemble de données, pour un mouvement réalisé, constitue un enregistrement dans la base de données.

Les requêtes à la base de données seront les suivantes, avec affichage de tous les champs :

- Connaître tous les postes ayant demandé au robot d'avancer en tournant vers la droite.

- Connaître tous les mouvements demandés par un poste en particulier.

- Connaître les moments où un poste en particulier a demandé au robot de reculer en ligne droite.

Sauvegardez votre code à chacune des étapes précédentes. Si vous ne parvenez pas à réaliser toutes les étapes, la dernière complétée servira pour la correction du code.

Vous devez démontrer en classe le fonctionnement de votre système de contrôle, avant la date limite de remise du travail.

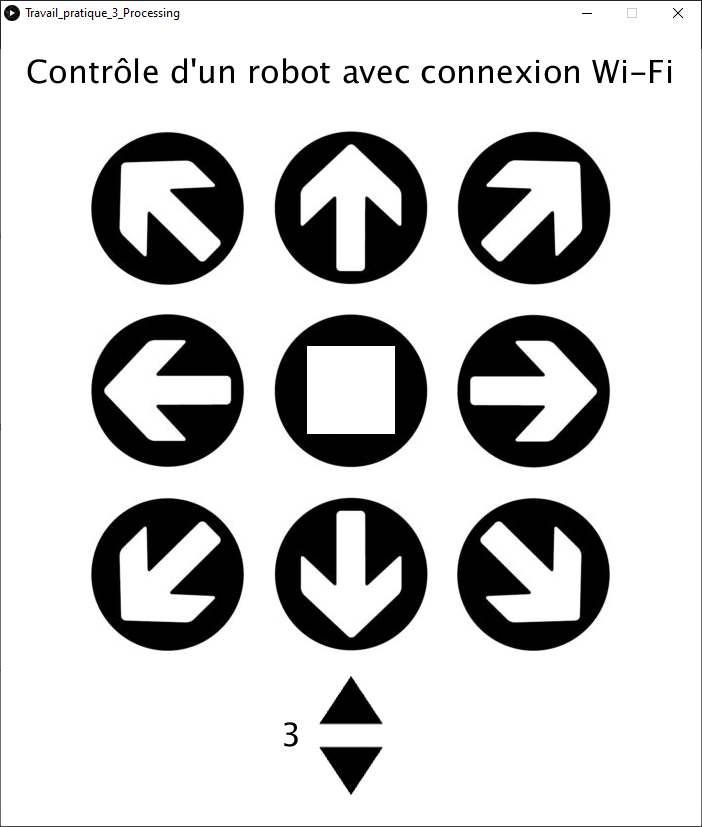
Vous n'avez pas à remettre d'algorithmes. Toutefois, il est **fortement** recommandé d'en produire avant de vous lancer dans la programmation, et ainsi gagner du temps.

Vos programmes en Processing et en Arduino doivent posséder un en-tête complet, et être bien commenté aux endroits appropriés.

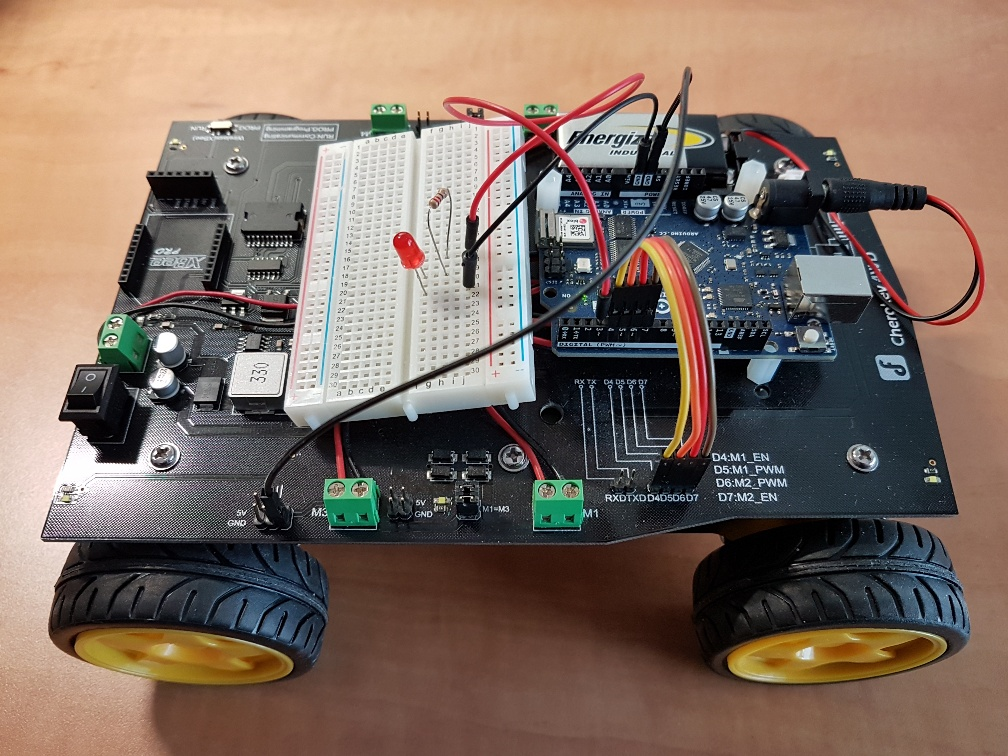
Vos variables doivent posséder des noms significatifs et être commentées.

La présentation de votre code doit être impeccable au niveau de sa modularité, de sa structure et de son indentation.

**Exemple d'interface Processing**



**Configuration physique du robot**



**Directives importantes**

- Reliez le GND du robot avec le GND de la carte Arduino.

- Débranchez l'alimentation 9V avant de brancher le câble USB.

- Débranchez le câble USB avant de rebranchez l'alimentation 9V.

- Mettre l'interrupteur du robot à ON seulement lorsque le robot est déposé au sol.

- Appuyez sur le bouton "reset" de la carte Arduino pour redémarrer le programme, une fois le robot déposé au sol.

**Autres informations**

N'hésitez pas à utiliser la console de Processing pour le dépannage, avec les fonctions **print()** et **println()**.

N'hésitez pas également à utiliser le Moniteur série d'Arduino pour le dépannage, avec les fonctions **Serial.print()** et **Serial.println()**.