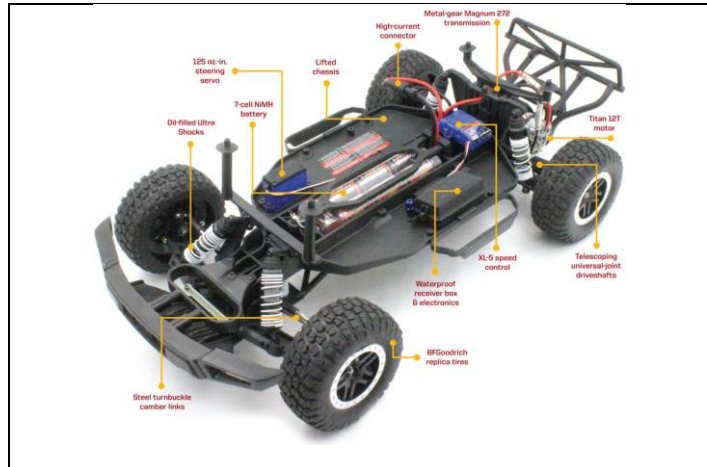


Noms :

Prénoms :

Classe :

Date :



Note : /20

1. Compétences abordées :

- Traduire le comportement attendu ou observé d'un objet : comportement séquentiel, structures algorithmiques
- Analyser le traitement de l'information : algorithmes, programmes, langages informatiques
- Traduire un algorithme en un programme exécutable
- Associer un modèle à un système asservi : capteurs
- Évaluer une solution : mesures et tests des performances de tout ou partie de la solution

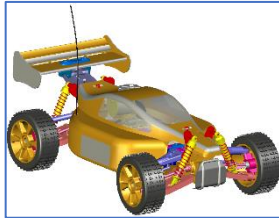
2. Problématique

Comment peut-on piloter la direction sur une voiture radio-commandée ?

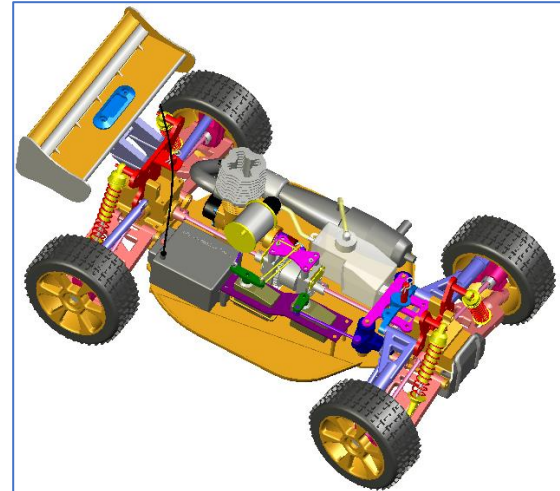
3. Critères d'évaluation et barème

Présentation / Soin / quantité de travail	/2
Analyse des pièces impliquées dans la direction (Q1, Q5)	/5
Mettre l'axe du servomoteur à des angles choisis (Q6 à Q9)	/6
Recopier la rotation du potentiomètre sur le servomoteur (Q10, 11, 12)	/4
Fondamentaux de la programmation orientée objet (OOP) (Q13)	/3

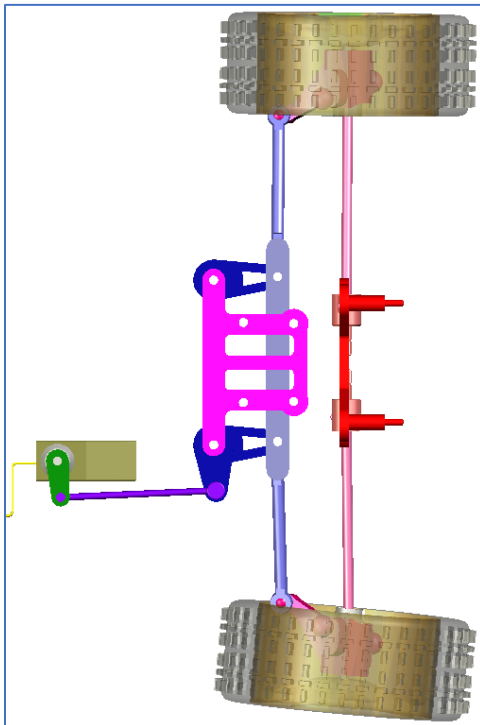
4. Description de l'objet étudié



La voiture radiocommandée étudiée possède un servomoteur pour mettre en mouvement le contrôle de la direction de la voiture.



- Q1) Ouvrez la maquette edrawing. Analysez-la, vous pouvez virtuellement la démonter et/ou cacher certaines pièces, puis entourez sur l'image ci-contre le servomoteur **de direction**.



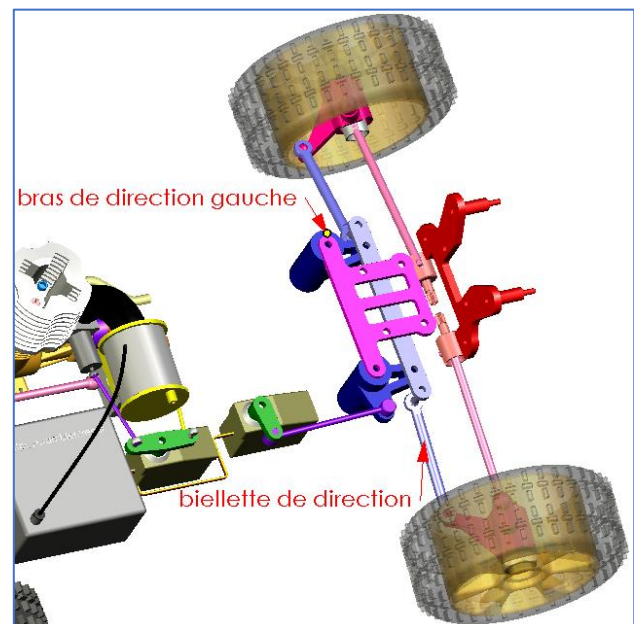
Q2) Sur le schéma ci-contre, ajouter des flèches pour indiquer le nom des composants impliqués dans le fait de faire pivoter les roues avant.

Q3) Expliquer en quelques lignes comment le mouvement est transmis du servomoteur de direction jusqu'aux roues.

Le repère de référence à utiliser pour les questions Q4 et Q5 est celui figurant sur l'espace graphique de la maquette numérique eDrawing.

- Q4) Identifier la nature des liaisons mécaniques (exemples : glissière d'axe y, ponctuelle de normale z, hélicoïdale d'axe x, ...) entre le bras de direction gauche et les autres pièces du mécanisme.

- Q5) Identifier la nature des liaisons mécaniques (exemples : glissière d'axe y, ponctuelle de normale z, hélicoïdale d'axe x, ...) entre la biellette de direction et les autres pièces du mécanisme.



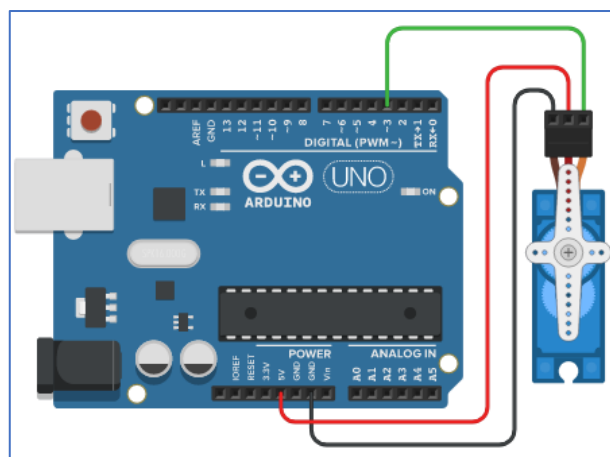
5. Mettre l'axe du servo à des angles choisis

Objectif n°1 : programmer une séquence, qui s'exécutera en boucle, au cours de laquelle l'axe du servomoteur se placera à 10°, puis à 20°, puis à 30° avec une attente de 2 secondes entre chaque position.

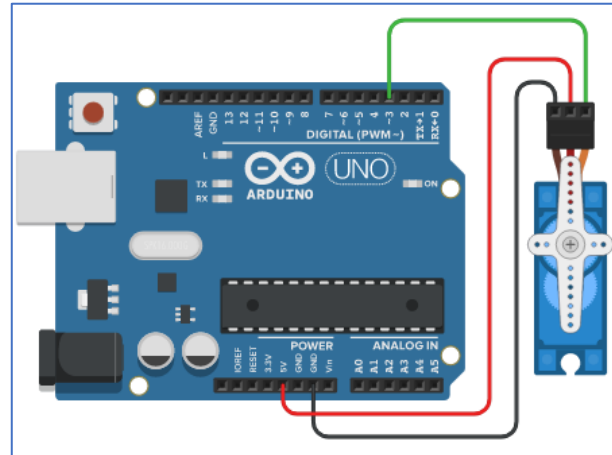
Q6) Compléter l'algorithme et le code Arduino ci-dessous :

<pre> graph TD Start([Début]) --> Step1[inclure la bibliothèque "Servo"] Step1 --> Step2[Créer un objet (une instance de la classe Servo) et l'appeler "_____"] Step2 --> Step3[Déclarer la broche _____ comme étant celle sur laquelle on branche le fil du signal de pilotage du servo] Step3 --> LoopStart(()) LoopStart --> Step4[Positionner l'axe à ____°] Step4 --> Step5[Attendre ____ secondes] Step5 --> Step6[Positionner l'axe à ____°] Step6 --> Step7[Attendre ____ secondes] Step7 --> Step8[Positionner l'axe à ____°] Step8 --> Step9[Attendre ____ secondes] Step9 --> LoopStart </pre>	<pre> #include <Servo.h> Servo _____; void setup() { monServo.attach(_____); } void loop() { monServo.write(_____); delay(_____); monServo.write(_____); delay(_____); monServo.write(_____); delay(_____); } </pre>
---	--

Q7) Saisissez le code sur l'IDE Arduino, réalisez le branchement sur la carte, et faites valider par votre professeur que l'objectif n°1 est atteint.



Objectif n°2 : programmer une séquence, qui s'exécutera en boucle, au cours de laquelle l'axe du servomoteur se placera de la position 10° à la position 170° par pas de 10°, avec une pause de 0,5 seconde entre deux positions successives.



Q8) Identifier les numéros de lignes associés aux éléments cités dans le tableau

```

1  #include <Servo.h>
2  Servo monServo;
3   int angle = 0;
4   int angle_initial = 10;
5   int angle_final = 170;
6   int increment = 10;
7
8  void setup() {
9   monServo.attach(3);
10 }
11
12 void loop() {
13   for(angle=angle_initial ;angle<=angle_final; angle+=increment){
14   monServo.write(angle);
15   delay(500);
16   }
17 }

```

N° de la ligne	Description
	Attendre 0,5 seconde
	Indiquer la broche sur laquelle on connecte le servo
	Créer un objet à partir d'une classe
	Inclure une bibliothèque
	Déplacer l'axe du servo à une certaine position
	Définir la condition de sortie de la boucle

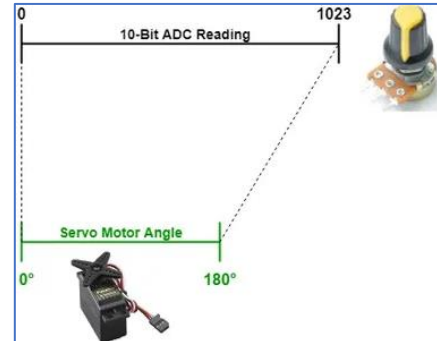
Q9) Transférez le programme sur la carte et vérifiez son bon fonctionnement.

6. Recopier la rotation du potentiomètre sur le servomoteur

Objectif n°3 : faire tourner l'axe du servomoteur proportionnellement à la rotation du potentiomètre.

Le potentiomètre est branché sur une entrée analogique qui convertit la tension lue en une valeur comprise entre 0 et 1023 (convertisseur 10 bits).

L'axe du servomoteur peut tourner d'un angle compris entre 0° et 180°



Pour réaliser ce changement d'échelle, on va utiliser la fonction **map**.

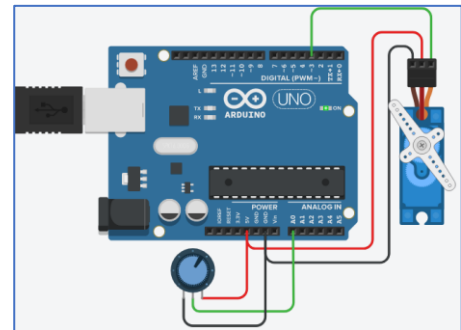
Syntaxe de la fonction :

`map` (valeur, limite_basse_source, limite_haute_source, limite_basse_destination, limite_haute_destination)

Paramètres de la fonction

- **valeur** : le nombre à ré-étalonner
- **limite_basse_source**: la valeur de la limite inférieure de la fourchette de départ
- **limite_haute_source**: la valeur de la limite supérieure de la fourchette de départ
- **limite_basse_destination**: la valeur de la limite inférieure de la fourchette de destination
- **limite_haute_destination**: la valeur de la limite supérieure de la fourchette de destination

- Q10) Calculer la position angulaire de l'axe du servomoteur lorsque la valeur lue sur la broche associée au potentiomètre est 900
- Q11) Calculer la valeur lue sur la broche associée au potentiomètre lorsque l'angle de l'axe du servo est 45°
- Q12) Compléter le programme ci-dessous, puis faire les branchements et tester le résultat



```

1  #include <Servo.h>
2  Servo _____;
3  int angle=10;
4
5  void setup()
6  {
7    monServo.attach(_____);
8    pinMode(A0, INPUT);
9  }
10
11 void loop()
12 {
13   angle=map(analogRead(A0), 0, _____, 0, _____);
14   monServo.write(_____);
15   delay(500);
16 }

```

7. Les fondamentaux de la programmation orientée objet (OOP)

Q13) En vous aidant de la fiche « structuration du code » et de recherches sur le web, expliquer par rapport au TP réalisé sur le servomoteur :

- Le nom de la classe
- Quel est le nom de l'instance de la classe
- Quel est le nom de l'objet
- Quelles sont toutes les méthodes (vous devez en trouver 6) associées à cette classe ?
- Pour chaque méthodes, indiquer sa fonction et les paramètres à renseigner.

