

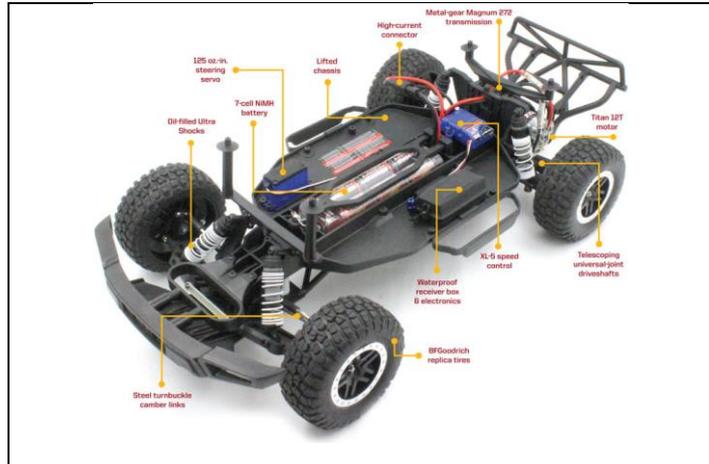
Noms :

Prénoms :

Classe :

Date :

Note : /20



1. Compétences abordées :

- Traduire le comportement attendu ou observé d'un objet : Comportement séquentiel Structures algorithmiques (variables, fonctions, structures séquentielles, itératives, répétitives, conditionnelles)
- Analyser le traitement de l'information : algorithme, programmes, langage informatique
- 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

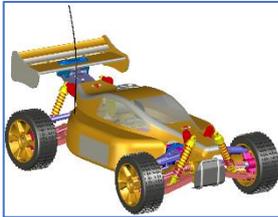
2. Problématique

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

3. Critères d'évaluation et barème

Présentation / Soin / quantité de travail	/3
Analyse des pièces impliquées dans la direction	/2
Mettre l'axe du servomoteur à des angles choisis	/3
Recopier la rotation du potentiomètre sur le servomoteur	/4
Lire la position réelle du servomoteur	/2.5
Fondamentaux de la programmation orientée objet (OOP)	/6.5

4. Description de l'objet étudié

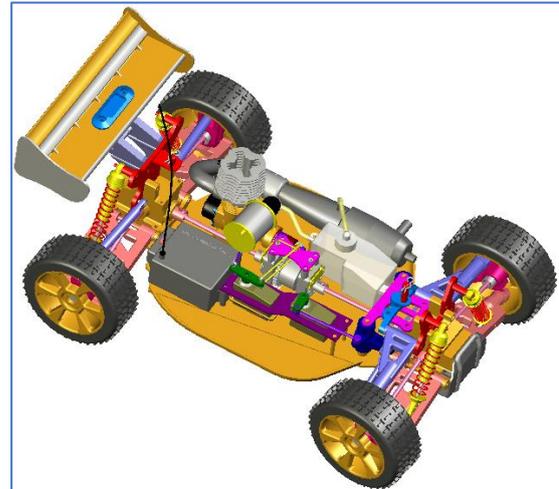
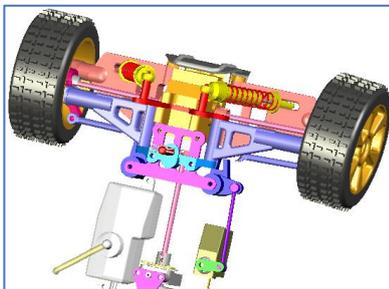


La voiture radiocommandée étudiée possède un servomoteur pour mettre en mouvement le contrôle de la direction de la voiture.

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

Q2) Réalisez un croquis simplifié pour expliquer comment est transmis le mouvement du servomoteur jusqu'aux

roues (vous pouvez mettre dans votre compte-rendu des captures d'écran annotées pour appuyer votre analyse).



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

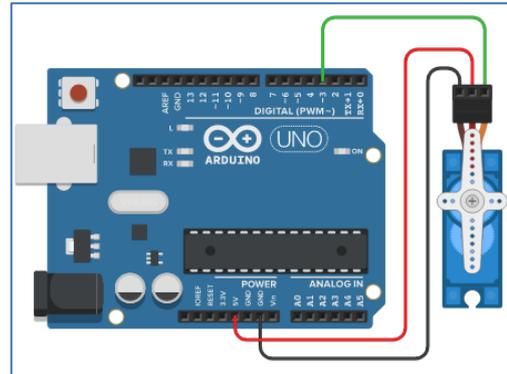
Objectif n°1 : programmer une séquence, qui s'exécutera en boucle, au cours de laquelle l'axe du servomoteur se placera à 10°, puis à 20°, puis à 30° avec une attente de 2 secondes entre chaque position.

Q3) Compléter l'algorithme et le code Arduino ci-dessous :

<pre> graph TD A[Début] --> B[inclure la bibliothèque "Servo"] B --> C["Créer un objet (une instance de la classe Servo) et l'appeler \"_\""] C --> D["Déclarer la broche _ comme étant celle sur laquelle on branche le fil du signal de pilotage du servo"] D --> E["Positionner l'axe à ___°"] E --> F["Attendre ___ seconde"] F --> G["Positionner l'axe à ___°"] G --> H["Attendre ___ seconde"] H --> I["Positionner l'axe à ___°"] I --> J["Attendre 1 seconde"] J --> E </pre>	<pre> #include <Servo.h> Servo _____; void setup() { monServo.attach(_____); } void loop() { monServo.write(_____); delay(_____); monServo.write(_____); delay(_____); monServo.write(_____); delay(_____); } </pre>
--	--

Q4) 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.



Q5) Identifier les numéros de lignes correspondant 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

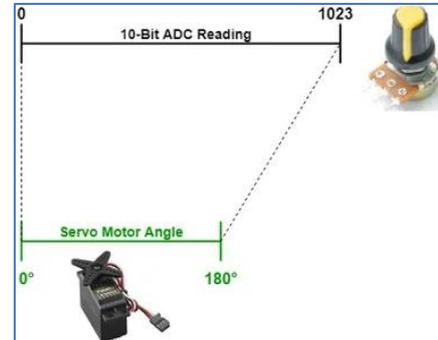
Q6) Transférez le programme sur la carte et vérifiez son bon fonctionnement.

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**.

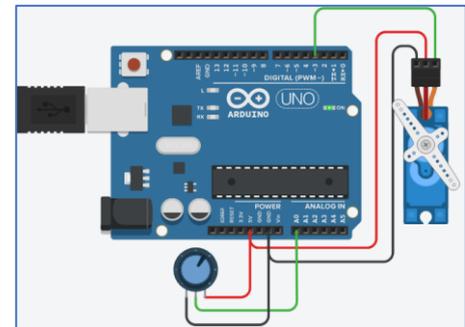
Syntaxe de la fonction :

`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

- Q7) Calculer que sera l'angle du servo lorsque la valeur lue sera 900
- Q8) Calculer quelle est la valeur lue lorsque l'angle du servo est 45°
- Q9) Compléter le programme ci-dessous, puis faire les branchements et testez 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 }

```



TP servomoteur

Voiture RC

Spé SI
2020-21

TP

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

Q10) 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 associées à cette classe
- Pour 3 de ces méthodes, indiquer la syntaxe et les paramètres