

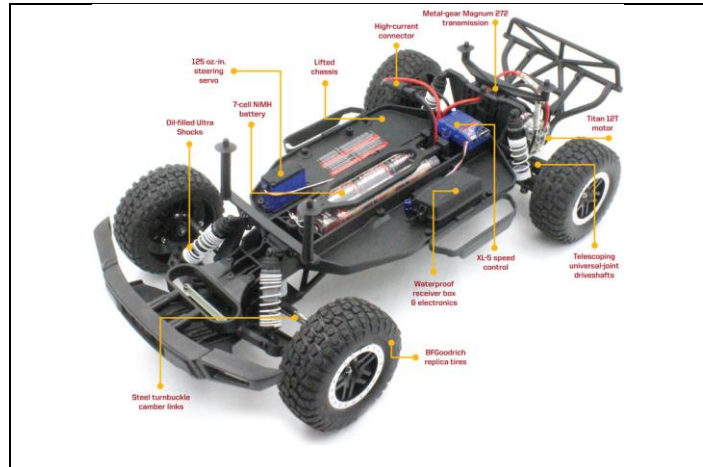
Noms :

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

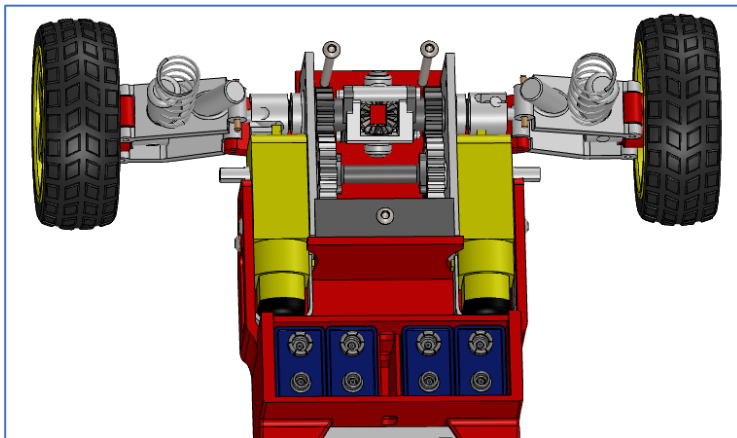
Comment est propulsée une voiture radio-commandée électrique ?

3. Critères d'évaluation et barème

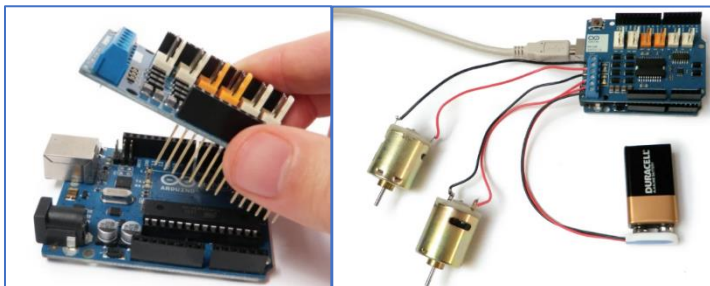
Présentation / Soin / quantité de travail	/1
Analyse des pièces impliquées dans la propulsion (Q1, Q2)	/2
Faire tourner un moteur CC à différentes vitesses (Q3, 4, 5, 6)	/4
Etude du réducteur (Q7, Q8)	/2
Faire tourner un moteur CC dans un sens puis dans l'autre (Q9 à Q14)	/6
Structurer le code grâce à l'utilisation de fonctions (Q15)	/2
Faire varier la vitesse et le sens grâce à un potentiomètre (Q16, 17, 18)	/3

4. Analyse de la propulsion de la voiture RC

- Q1) A partir de l'analyse de la maquette eDrawing et de recherches sur le web, identifier combien de moteurs servent à la propulsion, ce qu'est un différentiel et à quoi cela sert.
- Q2) Entourer et identifier ces éléments sur l'image ci-dessous.



5. Utilisation d'un shield moteur



La carte Arduino ne peut pas fournir un courant suffisant pour alimenter les moteurs de propulsion donc il faut utiliser une alimentation externe. Pour faciliter les branchements et la programmation, nous utiliserons un shield moteur.

Pour faire tourner un moteur, il faut paramétrer 3 choses :

- Il faut choisir le **sens** de rotation en mettant la broche associée sur « HIGH » ou « LOW »
- Il faut désengager le **frein** en mettant la broche associée sur « LOW »
- Il faut sélectionner la **vitesse** en attribuant à la broche PWM une valeur comprise entre 0 et 255 (car la valeur du PWM est codée sur 8 bits).

Broches d'interface

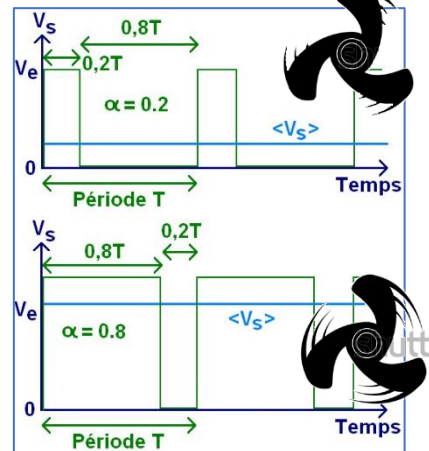
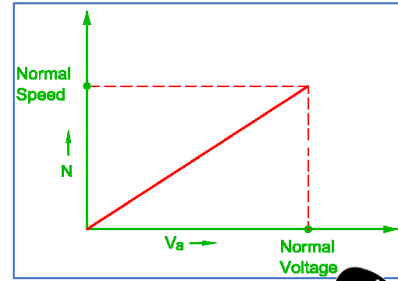
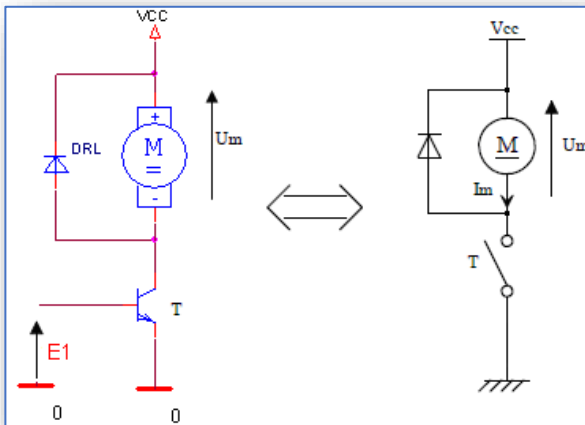
	Moteur A	Moteur B
<i>Choix du sens de rotation</i>	D 12	D 13
<i>Paramétrage de la vitesse (PWM)</i>	D 3	D 11
<i>Activation du frein</i>	D 9	D 8

6. Moduler la vitesse d'un moteur à courant continu

La vitesse de rotation du moteur est proportionnelle à la tension d'alimentation du moteur

Pour faire varier la tension moyenne d'alimentation du moteur, sans dissiper trop d'énergie, on alimente celui-ci avec une tension en créneaux

Pour générer cette tension en créneau, on envoie un signal rectangulaire sur la base d'un transistor ce qui a pour effet de le rendre bloqué ou passant.

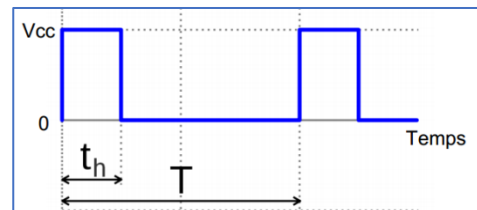


On appelle **rapport cyclique** le quotient, exprimé en pourcentage, de la durée du signal au niveau haut par rapport à la période du signal

Si $t_h=0$, alors $\alpha=0\%$ et la tension moyenne de sortie est nulle.

Si $t_h=T$, alors $\alpha=100\%$ et la tension moyenne de la sortie est égale à V_{cc} .

$$\alpha = 100 \times \frac{t_h}{T}$$



- Q3) Pour que le moteur tourne à la moitié de sa vitesse maximale, quel rapport cyclique (en %) et quelle valeur PWM (comprise entre 0 et 255) doit-on appliquer ?
- Q4) Pour que le moteur tourne à un quart de sa vitesse maximale, quel rapport cyclique (en %) et quelle valeur PWM (comprise entre 0 et 255) doit-on appliquer ?

Objectif n°1 : faire tourner le moteur aux 3/4 de sa vitesse maximale pendant 5 secondes puis à 50% de sa vitesse maximale pendant 3 secondes et le freiner pendant 2 seconde. Répéter indéfiniment ce cycle.

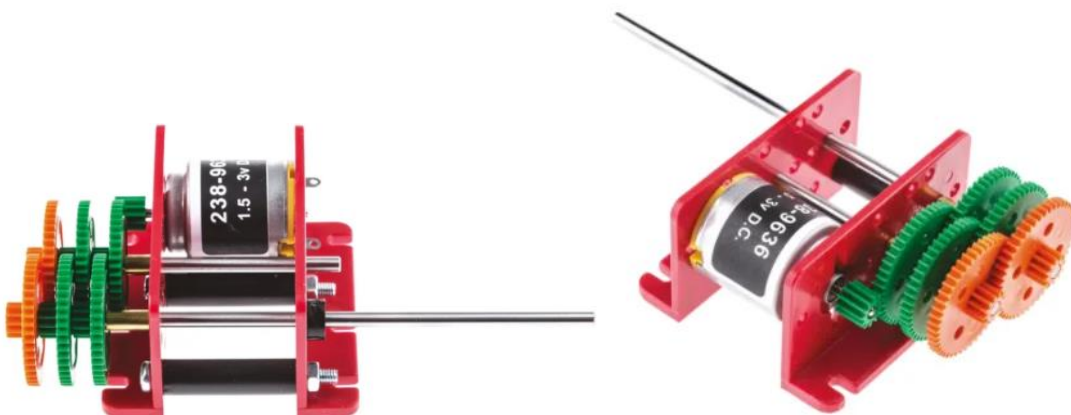
- Q5) Compléter l'algorithme et le programme ci-dessous :

- 1 broche pour choisir le sens (D12)
- 1 broche pour activer le frein (D9)
- 1 broche pour paramétrer la vitesse (D3)

Structure du programme :

Algorithme	Code Arduino
<pre> graph TD A[Début] --> B[Déclaration et paramétrage des broches utilisées] B --> C[] C --> D[] D --> C </pre>	<pre> void setup() { pinMode(12, OUTPUT); pinMode(9, OUTPUT); pinMode(3, OUTPUT); } void loop() { digitalWrite(____); digitalWrite(____); analogWrite(____); delay(____); digitalWrite(____); digitalWrite(____); analogWrite(____); delay(____); digitalWrite(____); delay(____); } </pre>

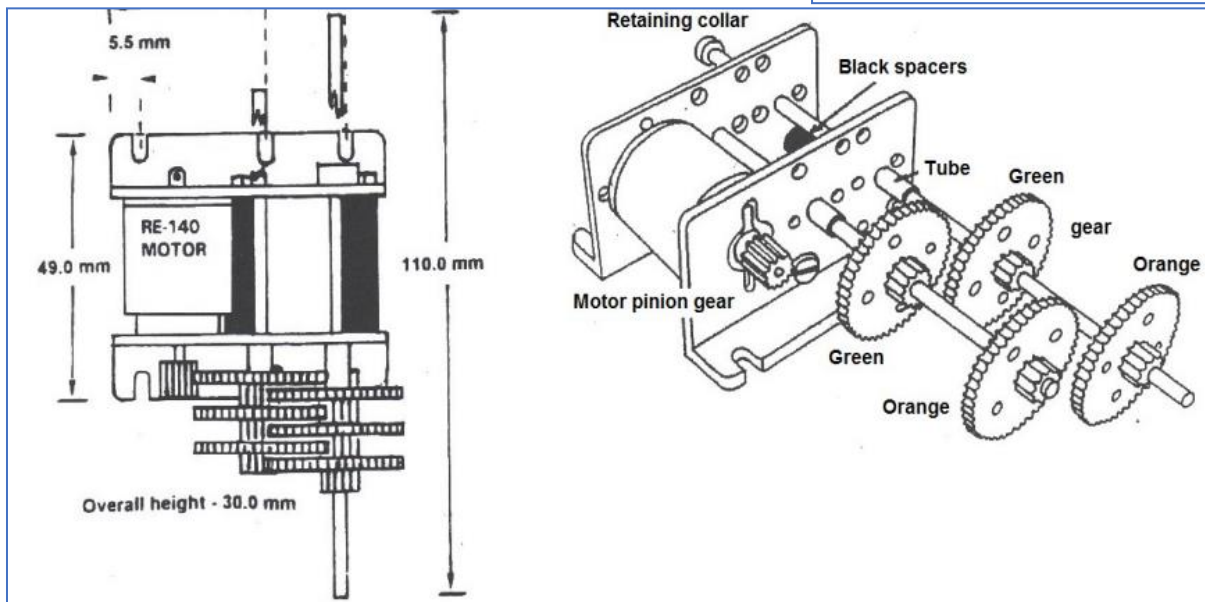
Q6) Réaliser le montage, **attention, régler l'alimentation externe sur 3V**, transférer le programme et faites constater le fonctionnement conforme à l'objectif auprès de votre professeur.



Q7) En vous appuyant sur la documentation ci-contre, calculer la vitesse du moteur à vide (lorsqu'il n'entraîne aucune roue), lorsqu'il est alimenté sous 3V ?

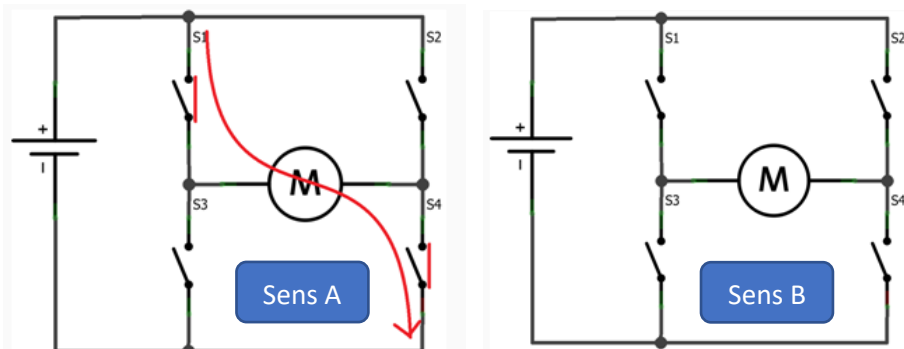
Q8) Pour avoir un rapport de réduction de 1/64, quelles sont les roues à garder (les colorier sur les deux dessins) ?

Reduction Table RPM	1.5v	3v
4:1	1850	3700
16:1	462	925
64:1	115	231
256:1	29	57
1024:1	7	14
4096:1	2	4



7. Piloter le changement de sens de rotation du moteur

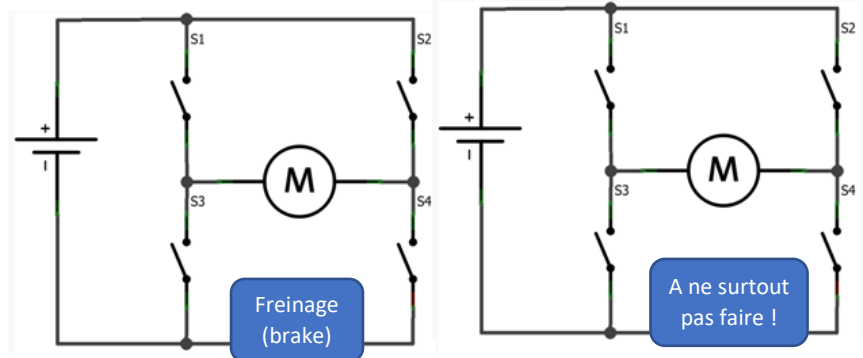
Le schéma, noté « sens A » vous indique quels sont les interrupteurs (qui modélisent les transistors boqués/passants du pont en H du shield moteur) qui doivent être fermés pour que le moteur tourne dans le sens A



Q9) Tracer sur le schéma les interrupteurs qui doivent être fermés pour sens B

Q10) Quels sont les interrupteurs qui doivent être fermés pour bloquer l'axe du moteur, en position ?

Q11) Quels sont les interrupteurs qu'il ne faut surtout pas fermer simultanément ? Pourquoi ?



Analyse du code :

Q12) Associer les numéros de lignes du code avec leur description

N° de ligne	Description
	Attendre 1 seconde
	Déclaration et paramétrage des broches utilisées
	Freiner le moteur
	Marche avant à pleine vitesse
	Marche arrière
	Attendre 5 secondes

Q13) Quelles lignes devez-vous modifier, et comment, pour que les moteurs tournent dans le sens 1, puis dans le sens 2, pendant 6 secondes ?

Q14) Quelle ligne devez-vous modifier, et comment, pour que lorsqu'il tourne, le moteur tourne toujours à sa vitesse maximale ?

```

1 void setup() {
2   pinMode(12, OUTPUT);
3   pinMode(9, OUTPUT);
4   pinMode(3, OUTPUT);
5 }
6
7 void loop(){
8   digitalWrite(12, HIGH);
9   digitalWrite(9, LOW);
10  analogWrite(3, 255);
11  delay(5000);
12
13  digitalWrite(9, HIGH);
14  delay(1000);
15
16  digitalWrite(12, LOW);
17  digitalWrite(9, LOW);
18  analogWrite(3, 123);
19  delay(3000);
20
21  digitalWrite(9, HIGH);
22  delay(1000);
23 }

```

8. Piloter le changement de sens de rotation du moteur

Après avoir lu le document « Structuration du code », modifier le code précédent en créant les fonctions suivantes :

- Forward_full_speed()
- Backward_half_speed()
- Brake()

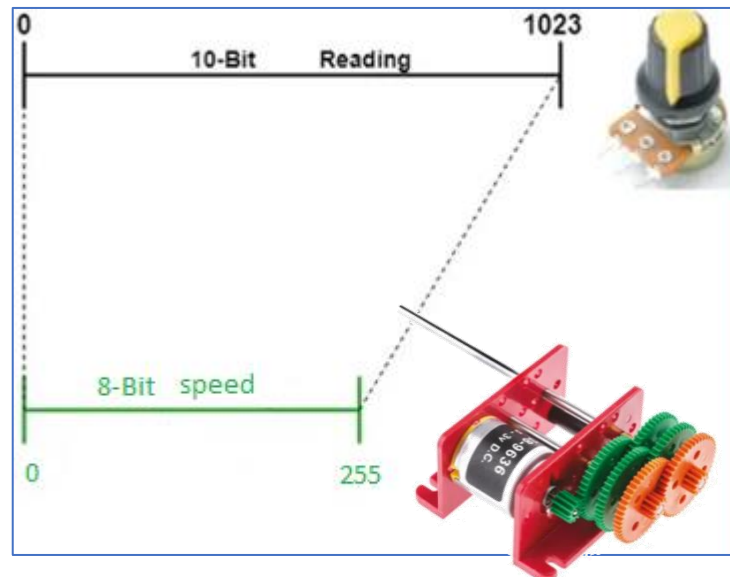
Q15) Insérer la copie d'écran de votre code dans votre compte-rendu et testez votre programme

9. Utiliser un potentiomètre pour faire varier la vitesse du moteur

Objectif n°3 : faire varier la vitesse du moteur entre 0% et 100% 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).

La vitesse du moteur est paramétrée par un PWM qui code le rapport cyclique sur 8 bits.



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

Q16) Calculer quel sera le rapport cyclique envoyé lorsque la valeur lue sera 900

Q17) Calculer quelle est la valeur lue lorsque le rapport cyclique envoyé est 200

Q18) Compléter le programme ci-dessous, puis faire les branchements et tester le résultat

```

1  int vitesse;
2
3  void setup(){
4    pinMode(12, OUTPUT);
5    pinMode(9, OUTPUT);
6    pinMode(3, OUTPUT);
7    pinMode(A0, INPUT);
8  }
9
10 void loop(){
11  vitesse=map(analogRead(_____),0,_____,0,_____);
12  digitalWrite(12,_____);
13  digitalWrite(9,_____);
14  analogWrite(3,_____);
15  delay(50);
16 }

```