



IUT de St Etienne
Département GEII

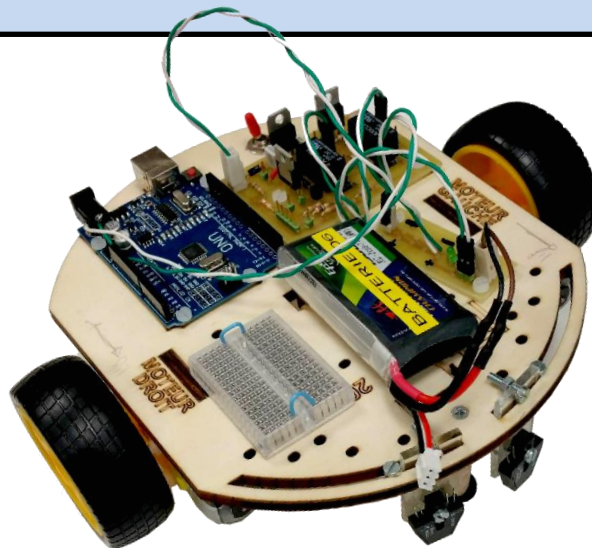


S.A.É. 2

(Situations d'Apprentissage et d'Évaluation)

Phase 8

Commande des moteurs du robot



Rappel de consignes:

Pour chaque nouveau programme, vous devez donner un algorithme textuel et/ou un algorithme. Vous devez créer un fichier (.ino) différent par programme.

N'hésitez pas à commenter abondamment vos programmes.

Dans cette phase, on va créer les fonctions logicielles nécessaires à la commande des moteurs du robot.

Matériel nécessaire:

Votre carte commande 2 moteurs

- Un châssis de robot avec 2 motoréducteurs.
- Une carte Arduino
- Un breadboard
- Un potentiomètre.

I. COMMANDER UN MOTEUR

Sur la carte de commande moteur (driver), on dispose, par moteur, de 2 entrées: PWM et SENS.

Q1. Rappeler ce que signifie PWM. Pourquoi utilise-t-on une commande PWM pour commander un moteur.

Q2. Donner le schéma de câblage de la carte Arduino, carte driver, motoréducteur et batterie pour commander le moteur 1.

Q3. Quelle est la fonction logicielle à utiliser pour commander la sortie PWM? Quels sont ses paramètres?

Q4. Quelle est la fonction logicielle à utiliser pour commander la sortie SENS? Quels sont ses paramètres?

Effectuer le câblage de la question Q2.

Q5. Donner l'algorithme du programme qui devra effectuer:

- Le moteur tourne en avant à 50% pendant 2s.
- Le moteur s'arrête pendant 1s.
- Le moteur tourne en arrière à 75% pendant 4s.
- Le moteur s'arrête pendant 1s.
- Etc .

Q6. Traduire en C++, tester.

Faire valider

Mise en fonction

Pour faciliter la programmation, on crée une fonction de commande d'un moteur.

On veut une fonction qui permette les deux fonctionnalités :

- marche avant/arrière
- vitesse variable avec PWM de 0 à 100% (0 à 255)

Son prototype sera:

```
void Com_Mot_Gauche(int vitesse);
```

vitesse sera un paramètre compris entre - 255 et + 255.

Si **vitesse** ≥ 0 , le moteur tournera en marche **avant**, avec le PWM réglé à **|vitesse|**

Si **vitesse** < 0 , le moteur tournera en marche **arrière**, avec le PWM réglé à **|vitesse|**

On définira des équivalences dans l'en-tête du fichier pour les GPIO qui commandent le PWM et le SENS.

Q7. Écrire un algorithme de cette fonction.

Q8. Modifier l'algorithme la question Q5 pour tester la fonction.

Q9. Les traduire en C++/Arduino. Tester.

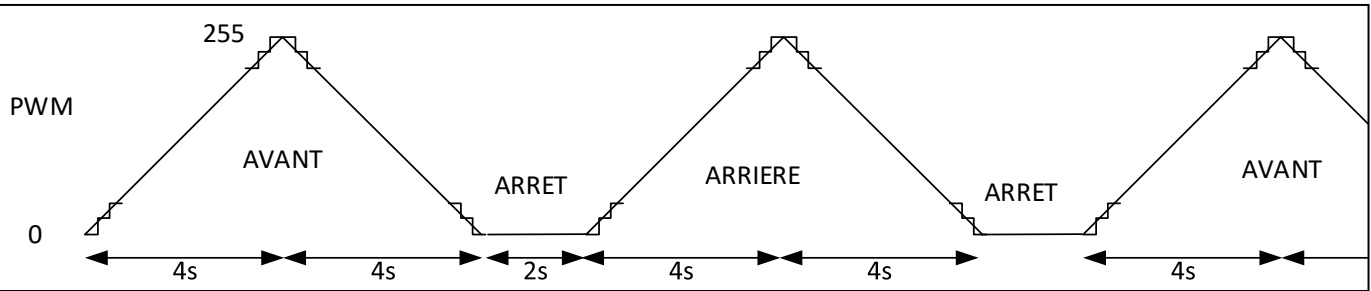
Faire valider

II. COMMANDER DEUX MOTEURS

Q10. De la même façon créer la fonction `void Com_Mot_Droit(int vitesse);`

Q11. Modifier l'algorithme la question Q8 pour tester les 2 fonctions `Com_Mot_Gauche` et `Com_Mot_Droit`.

Q12. Écrire l'algorithme d'un programme permettant sur chaque moteur d'effectuer les actions suivantes:



Q13. Traduire en C++. Tester.