



Poppy-ergo-jr

Je programme !

On souhaite demander à une tierce personne les positions de chaque moteur, **rédigier** un programme qui demande la valeur angulaire de chaque moteur, positionner le bras à chaque saisie.

Rédiger un programme identique sauf que le bras se positionnera lorsque les valeurs angulaires de tous les moteurs auront été toutes saisies.

Rédiger maintenant un programme moins fastidieux pour celui qui l'utilise, pour cela on demandera uniquement le numéro du moteur dont on doit modifier la position puis la valeur angulaire à changer.

Il existe une autre ligne de commande pour piloter les moteurs :

```
ergo.m1.goto_position(90,1,wait=True)
```

Ou encore

```
ergo.goto_position({'m1': 30,'m2': 90,'m3': 0,'m4': -90,'m5': 90,'m6': 60}, 5, wait=True)
```

Expliquer l'avantage dont on pourrait tirer parti de cette dernière commande ?

Réécrire le dernier programme à l'aide de cette ligne de commande.

Proposer une autre façon de programmer une trotteuse des secondes avec la phrase
`ergo.m1.....goto_position(90,1,wait=True)`

Tester maintenant cette ligne de code :

```
ergo.m1.present_position
```

Expliquer le type d'information que renvoie le notebook après quelques essais.

Afficher la position réelle du bras dans le programme de trotteuse, **déduire** le pourcentage de précision de la simulation.

Le bras réel est constitué de connectique entre les moteurs, ce qui crée un bus de communication entre la carte raspberry et tous les moteurs. **Expliquer** en quelques lignes ce qu'est un bus informatique et l'intérêt pour notre bras.

Cette connectique, même réduite empêche le bras de faire plus d'un tour alors que la simulation le permet. **Corriger** votre programme de trotteuse pour que celui-ci fasse 59 secondes et fasse ensuite en une seconde un tour dans l'autre sens avant de reprendre.

J'ai réussi cette partie, je montre à mon professeur pour avoir le code de la partie suivante !

