

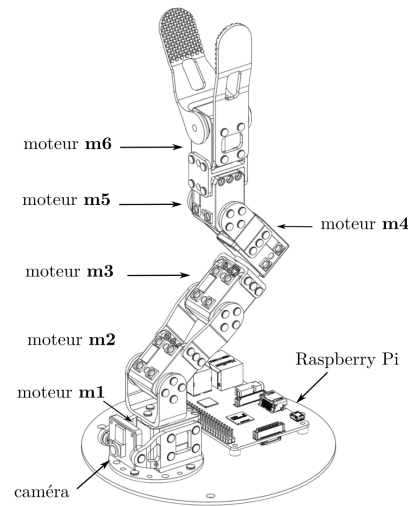
À la découverte de PoppyErgoJr



Le robot PoppyErgoJr a été conçu au laboratoire Flowers de l'INRIA de Talence (33). Il est imprimable en 3D, entièrement *opensource* et *openhardware*.

Il est utilisé dans de nombreux lycées et collèges de Nouvelle-Aquitaine (projets PoppyÉducation et Persévérans)

[plus d'informations sur poppy-education.org](http://poppy-education.org)



I- INFORMATIONS GÉNÉRALES

PoppyErgoJr est composé de 6 servomoteurs, contrôlés par une carte Raspberry Pi.

Sur cette carte Raspberry Pi tourne un système d'exploitation (*Raspbian*). Au démarrage la RPI, celle-ci récupère sur le réseau une adresse IP. Un serveur Python se lance automatiquement, et il devient donc possible d'envoyer (et de recevoir) des informations au robot depuis n'importe quel ordinateur connecté sur le même réseau local.

Le robot peut être alors commandé avec le langage de programmation graphique Snap! (ou Scratch), ou avec le langage Python.

II- PREMIERS MOUVEMENTS AVEC SNAP!

1. Moteurs « stiff » et « compliant »



- moteurs **stiff** : ils sont rigides, ne doivent pas être bougés à la main. Ils « attendent » de recevoir une instruction de mouvement.
- moteurs **compliant** : ils peuvent être bougés à la main. Leur position instantanée peut être renvoyée par le bloc `get all motors positions`.

2. Instruction de mouvement des moteurs

Attention : les angles des moteurs doivent rester dans l'intervalle $[-90; 90]$, **excepté pour le moteur m6** (la pince) où l'angle ne doit pas être inférieur à -20 degrés (environ).

Pour pouvoir être mis en mouvement, le moteur doit impérativement être en mode **stiff**.

Exemple de mouvement du moteur m5 à la position 40 degrés, effectué en 1 seconde.



Exemple d'application :

- Faites saluer le robot par des mouvements du moteur m4, alternés de gauche à droite .

3. Apprentissage par démonstration

Il est possible d'enregistrer des mouvements faits à la main et de les faire reproduire ensuite par l'ErgoJr.

- Mettez tous les moteurs en **compliant**.
- Créez le script ci-dessous. Lorsque vous le déclencherez, vous aurez 5 secondes pour manipuler le robot à la main, avant qu'il ne redevienne **stiff**. Attention! Ne pas forcer sur le robot lorsqu'il devient **stiff**.



- Vous pouvez maintenant rejouer le mouvement effectué :



Comme vous pouvez le constater, les mouvements sont reproduits, mais la précision n'est pas parfaite ... Il est préférable de travailler avec des positions enregistrées dans des variables.

III- ENCHAÎNEMENT DE MOUVEMENTS

1. Enregistrement d'une position dans une variable

1. Créez une variable nommée `maPosition`.
2. Créez le bloc `maPosition prend la valeur get all motors positions`
3. À la main, placez le robot dans la position désirée (il faut qu'il soit **compliant**!) puis cliquez sur le bloc de la question précédente. Vous pouvez observer dans la scène de Snap! que la variable `maPosition` est maintenant une liste contenant les angles des 6 moteurs.
4. Bougez le robot pour le mettre dans une autre position, repassez les moteurs en **stiff**, puis créez et exécutez le bloc : `set position(s) maPosition of motor(s) all motors in 2 seconds | wait ?`
5. Si le mouvement doit être corrigé, il est possible de modifier directement les valeurs des angles de la variable `maPosition` en double-cliquant dessus dans la scène Snap!.

2. Défis...

- Attrapez un cube posé sur un gobelet retourné et reposez-le dans un autre gobelet.
- Reprenez le défi précédent, mais enlevez le cube au moment où le robot essaie de l'attraper : le robot doit ré-essayer jusqu'à ce qu'il y arrive.

IV- PROGRAMMATION EN PYTHON

La programmation peut se faire grâce à l'interface Jupyter accessible depuis la page d'accueil du robot.

```
In [1]: from poppy.creatures import PoppyErgoJr
```

```
In [2]: poppy = PoppyErgoJr()
```

```
In [3]: poppy.m4.compliant = False
```

```
In [4]: import time
for i in range(3):
    poppy.m4.goto_position(-30, 0.5)
    time.sleep(1)
    poppy.m4.goto_position(30, 0.5)
    time.sleep(1)
```