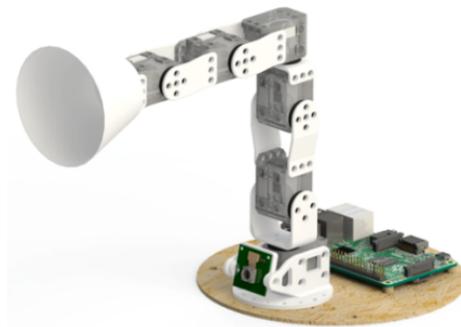




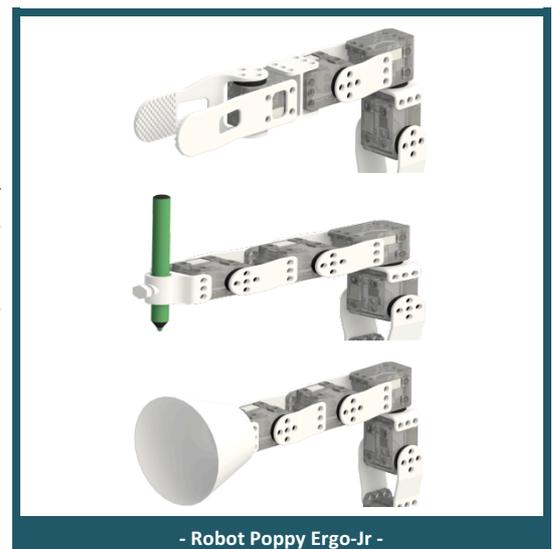
## ICN – Informatique et Création Numérique

# MODULE « PROGRAMMER UN ROBOT » : TP1 – DECOUVERTE DU ROBOT POPPY ERGO-JR



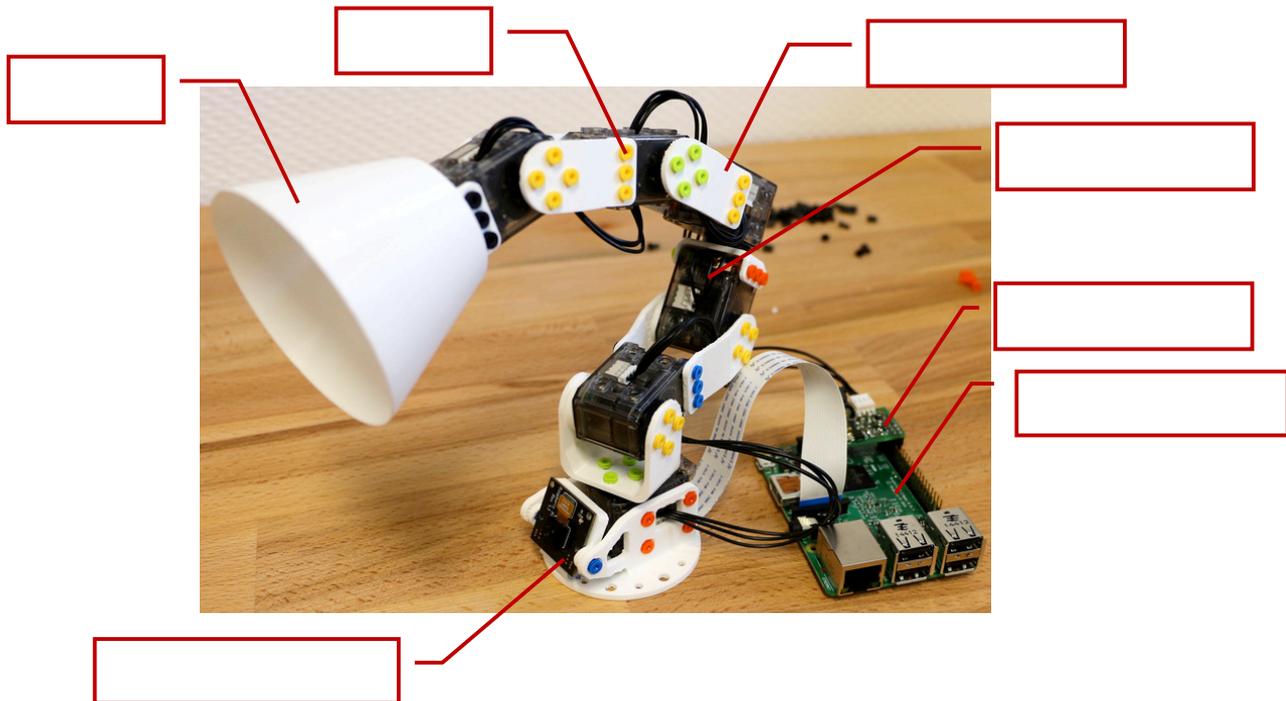
## 1 – LE ROBOT POPPY ERGO-JR

Le robot **Poppy Ergo-Jr** est un petit **bras robotisé** open-source disposant de 6 degrés de liberté. Il est constitué de pièces très simples pouvant facilement être réalisées au moyen d'une imprimante 3D. Le robot est livré avec trois embouts lui permettant d'interagir avec son environnement : un abat-jour, une pince, ainsi qu'un porte stylo.



### Activité n°1

1. **Indiquer** à quel type de robots appartient le robot Poppy Ergo-Jr.
2. **Indiquer** dans quel cadre ce robot a-t-il été créé.
3. **Citer** les autres robots de la famille Poppy.
4. **Compléter** l'image ci-dessous en indiquant les éléments constitutifs du robot Poppy Ergo-Jr.



## 2 – LE LOGICIEL SNAP!

Le langage de programmation Snap! Il s'agit d'une ré-implémentation du logiciel Scratch développée par l'université de Berkeley en Californie.



Comme Scratch, Snap! est basé sur l'imbrication de blocs de commandes. Cependant Snap! permet de créer ses propres blocs de programmation et intègre des fonctionnalités plus pointues en programmation et peut donc être utilisé par un public plus avancé.

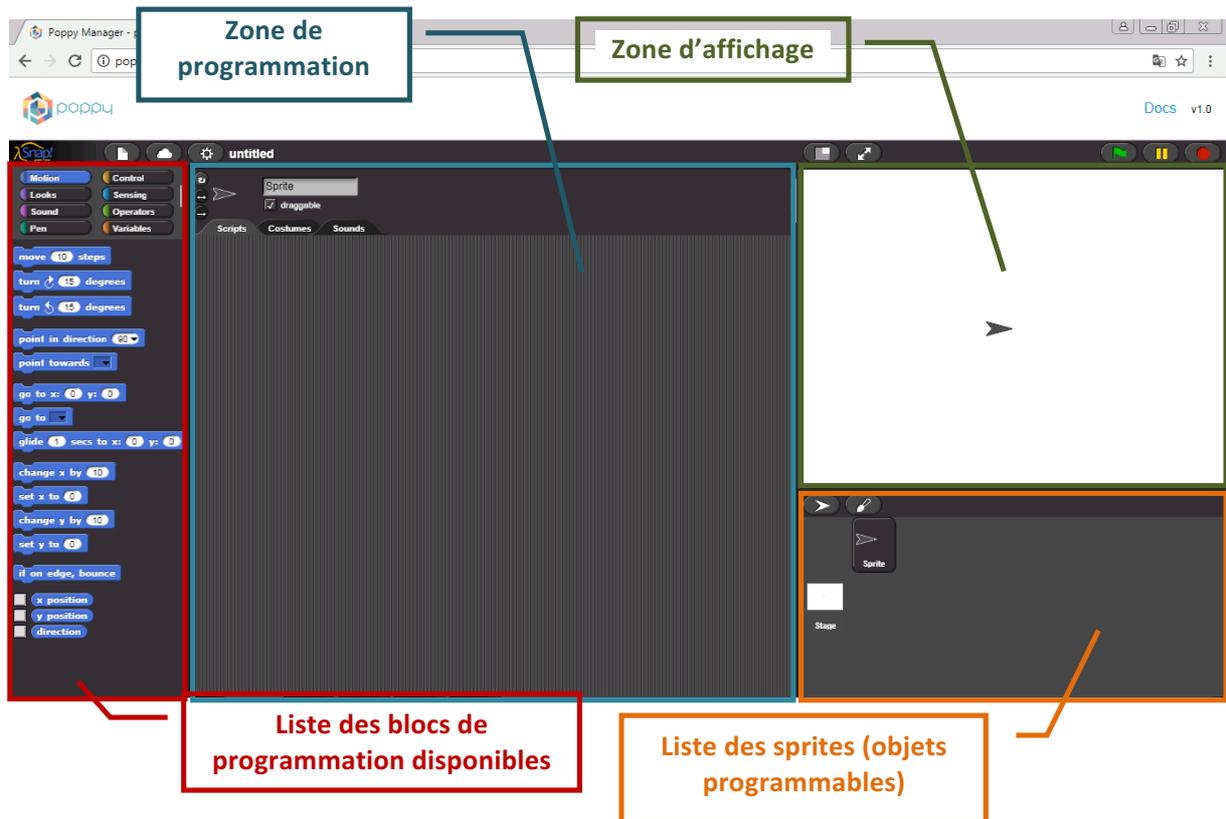


### Activité n°2

1. **Connecter** l'ordinateur au robot en entrant l'URL « <http://adresse IP robot/> » dans un navigateur.
2. Sur la page d'accueil, **cliquer** sur la case :



L'interface de développement Snap! est découpée en plusieurs zones :



Snap! possède un certains nombres de blocs dont la forme et la couleur varient en fonction de leur utilité.

Pour programmer le robot Poppy, il faut charger des blocs spécifiques à sa programmation. Ces blocs se trouvent dans le fichier « **pypot-snap-blocks.xml** » déjà présent sur la carte Raspberry Pi. Pour charger ces blocs :

### Activité n°3

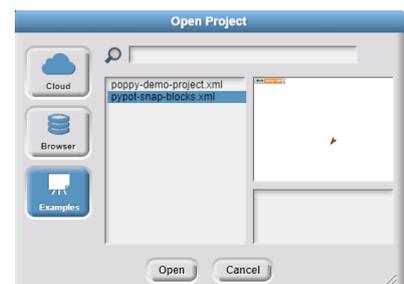
1. Cliquer sur l'icône.



2. Cliquer sur « Open... ».



3. Cliquer sur « Exemples ». Sélectionner « pypot-snap-blocks.xml » puis cliquer sur « Open »





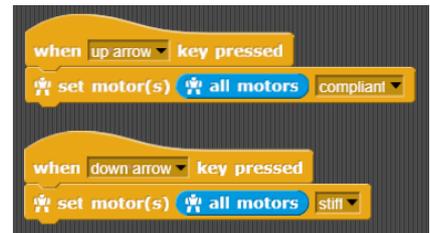
Deux blocs se trouve dans la zone de script. Le bloc « set » permet de régler l'adresse IP du robot et le bloc « test connection » permet de vérifier la connexion entre le robot et l'ordinateur.



## 3 – PREMIER PAS AVEC LE ROBOT POPPY ERGO-JR

### 3.1 – Activation et désactivation des servomoteurs

Le script ci-contre permet d'**activer** tous les servomoteurs (mode « **stiff** ») lorsque la flèche bas du clavier est appuyée et de **désactiver** tous les servomoteurs (mode « **compliant** ») lorsque la flèche haut du clavier est appuyée.

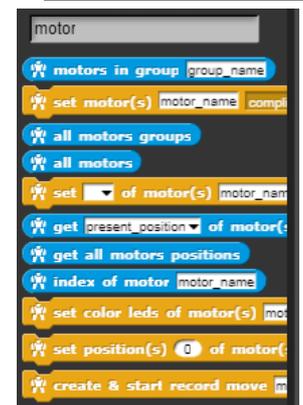


**FRAGILE**

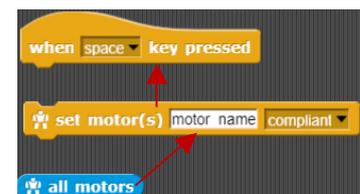
En mode « compliant », les servomoteurs sont désactivés, c'est-à-dire au repos. Ils peuvent être manipulés à la main. **En mode « stiff », les servomoteurs sont actifs, ils se bloquent et ne peuvent plus être manipuler manuellement.**

Pour trouver les blocs dans la zone des blocs il est possible d'effectuer une recherche par :

- **catégorie** (couleur). Le bloc « When... » et le bloc « set motor(s) » se trouvent dans la palette « Control ». Le bloc « all motors » dans la palette « Sensing ».
- **mot clé** : cliquer sur « Ctrl + f » ou alors sélectionner « finds blocks » avec un clic droit de la souris dans la zone des blocs de programmation. Entrer le mot clé dans l'onglet qui est apparu. Par exemples « when » ou « motor » permettront d'obtenir les blocs nécessaires au script ci-dessus.



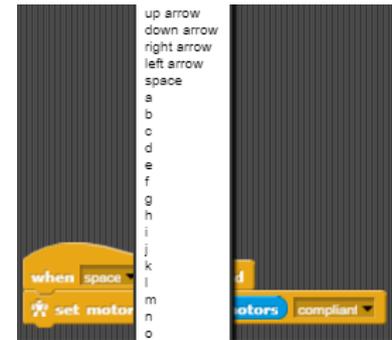
Pour placer un bloc dans la zone de script, il suffit de le sélectionner avec le bouton gauche de la souris et de le glisser dans la zone de script. Il suffit alors d'emboîter les blocs pour obtenir le script voulu.





Pour choisir la touche sur laquelle il faut appuyer, il faut cliquer sur la flèche noire dans le bloc « When ».

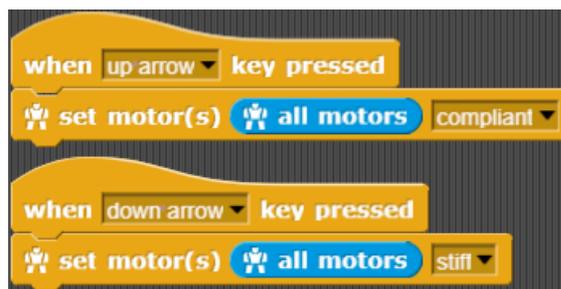
De même pour passer de « compliant » à « stiff », il faut cliquer sur la flèche noire dans le bloc « set motor(s) ».



Pour activer un bloc, il faut **l'exécuter**. Pour cela, il faut cliquer sur le bloc ou sur le groupe de blocs, celui-ci envoie alors une **commande** au robot.

#### Activité n°4

1. **Editer** le script ci-contre.



2. **Cliquer** sur la touche « up arrow » du clavier.



3. **Préciser** ce qui se passe au niveau des blocs dans le fenêtre des scripts.

4. **Manipuler** délicatement le robot. **Préciser** s'il est possible de manipuler manuellement le robot.

5. **Cliquer** sur la touche « down arrow » du clavier.



6. **Préciser** ce qui se passe au niveau des blocs dans le fenêtre des scripts.

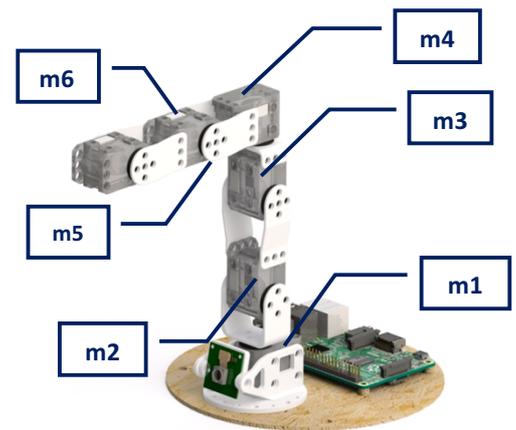
7. **Manipuler** délicatement le robot. **Préciser** s'il est possible de manipuler manuellement le robot.



Ces deux groupes de blocs seront toujours placés dans les différents scripts réalisés dans les TPs.

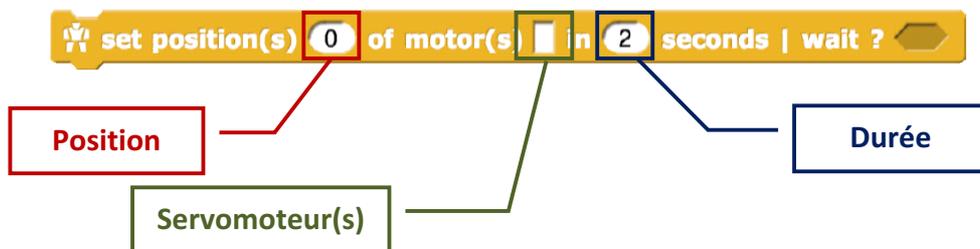
### 3.2 – Animer le robot

Chaque moteur a un nom, ils sont numérotés de **m1** (à la base) à **m6** (l'extrémité).



**FRAGILE**

Les servomoteurs ont certaines limites, toutes les positions ne peuvent pas être atteintes. Il est conseillé de **respecter des positions comprises entre -90° et 90°**.



Le bloc « set position(s) ... » permet de **modifier la position** d'un ou de plusieurs servomoteurs. Ce bloc prend comme paramètres la position à atteindre, le(s) servomoteur(s) concerné(s) et la durée nécessaire pour atteindre la position.



Pour exécuter ce bloc, il est nécessaire que les **servomoteurs soient en mode « stiff »**.

#### Activité n°5

1. **Ajouter** au script précédent le bloc ci-contre.



2. **Préciser** quel servomoteur est concerné par le bloc.

3. **Placer** les servomoteurs en mode « stiff ».

4. **Excuter** le bloc « set position ... » en cliquant dessus délicatement le robot. **Vérifier** que le fonctionnement du robot correspond bien au comportement attendu.

5. **Remplacer** « m1 » par le bloc ci-contre.



6. **Préciser** quels servomoteurs sont concernés par le bloc.

### Activité n°5 (Suite)

7. **Exécuter** le bloc « set position ... ». **Vérifier** que le fonctionnement du robot correspond bien au comportement attendu.
8. **Réaliser** le même travail avec les blocs ci-contre.  
10. **Modifier** le script afin que les servomoteurs m1 et m3 soient placés dans la position  $-30^\circ$  en 2 s. **Exécuter** le script et **vérifier** que le fonctionnement correspond bien au comportement attendu.

### Activité n°6

1. **Créer** un script afin que :
  - tous les servomoteurs soient placés dans la position  $0^\circ$  en 3 s lorsque la touche « right arrow » du clavier est appuyée. 
  - les servomoteurs m1 et m4 soient placés dans la position  $60^\circ$  en 2 s lorsque la touche « left arrow » du clavier est appuyée. 
2. **Exécuter** le script et **vérifier** que le fonctionnement du robot correspond bien au comportement attendu.

## 3.3 – Créer des mouvements

### Activité n°7

1. **Editer** le script ci-contre.



2. **Indiquer** l'action du 1<sup>er</sup> bloc sur le robot.
3. **Exécuter** le groupe de blocs et **observer** ce qui se passe.
4. **Indiquer dans ordre sont exécuter** les blocs.
5. **Modifier** le script afin remplacer le second  par 



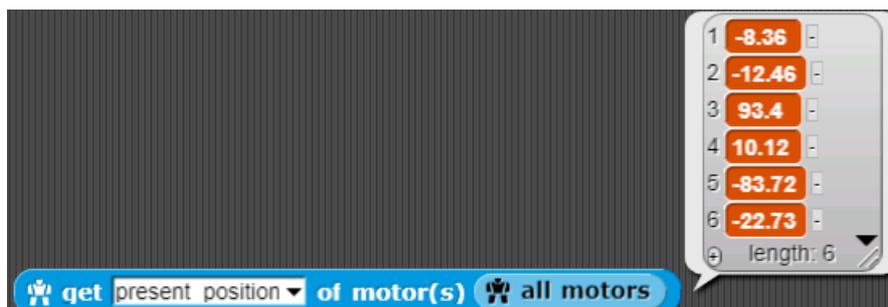
6. **Exécuter** à nouveau le groupe de blocs et **observer** ce qui se passe.
7. **Préciser ce qui se passe** quand « wait ? » est égal à **true** (  ). **Préciser ce qui se passe** quand « wait ? » est égal à **false** (  ).



## Activité n°8

**Cahier des charges** : Créer un script afin que le robot Poppy effectue un mouvement signifiant « Bonjour » lorsque la touche « b » du clavier soit appuyée.

1. **Choisir** les servomoteurs qui seront utilisés pour réaliser le mouvement.
2. **Placer** tous les servomoteurs en mode « compliant ».
3. **Placer** et **exécuter** le bloc suivant qui permet de récupérer la position des servomoteurs.



4. **Réaliser** manuellement le mouvement désiré (**attention** : en mode compliant) et relever les différentes positions des servomoteurs à utiliser.
5. **Editer** le script permettant de réaliser le mouvement moteur par moteur.
6. **Exécuter** le script et **vérifier** que le fonctionnement du robot correspond bien au comportement attendu.

## 4 – DEFI ROBOTIQUE : POPPY JOUE AU CHAMBOULE-TOUT

## Activité n°9

**Cahier des charges** : Créer un script permettant au robot de lancer une balle afin de faire tomber les gobelets. **Vidéo** : [https://www.youtube.com/watch?time\\_continue=30&v=Z8fklhn5r-l4](https://www.youtube.com/watch?time_continue=30&v=Z8fklhn5r-l4)

