

Robots LEGO EV3 : panneau de contrôle

1 Présentation

Dans ce deuxième TP, vous allez réaliser une application en Python pour commander à distance un robot LEGO EV3 via un panneau de contrôle. L'objectif est de se familiariser avec la programmation réseau en Python pour les robots LEGO.

1.1 Les sockets

Les sockets sont des objets qui permettent d'ouvrir une connexion avec une machine locale ou distante et échanger avec elle. On associe à chaque socket un protocole de transport TCP ou UDP. Dans ce travail, nous utilisons le protocole TCP (Transmission Control Protocol), soit « protocole de contrôle de transmission ». Concrètement, il permet de connecter deux applications et de leur faire échanger des informations. Des détails sont disponibles sous ce lien : <https://openclassrooms.com/fr/courses/235344-apprenez-a-programmer-en-python/234698-gerez-les-reseaux>.

1.2 Les différentes étapes

Notre application est type client/serveur. Un programme client s'exécute sur l'une de machine de la salle et un programme serveur s'exécute sur le robot. Pour que le client se connecte au serveur, il lui faut : le nom d'hôte ou son adresse IP (celle du robot dans notre cas), et un numéro de port entre 0 et 65535 que nous devons choisir. Le programme serveur réalise les étapes suivantes :

- attend une connexion de la part du client ;
- accepte la connexion quand le client se connecte ;
- échange des informations avec le client ;
- ferme la connexion.

Le programme client réalise les étapes suivantes :

- se connecte au serveur ;
- échange des informations avec le serveur ;
- ferme la connexion.

La figure suivante, illustre ces étapes :

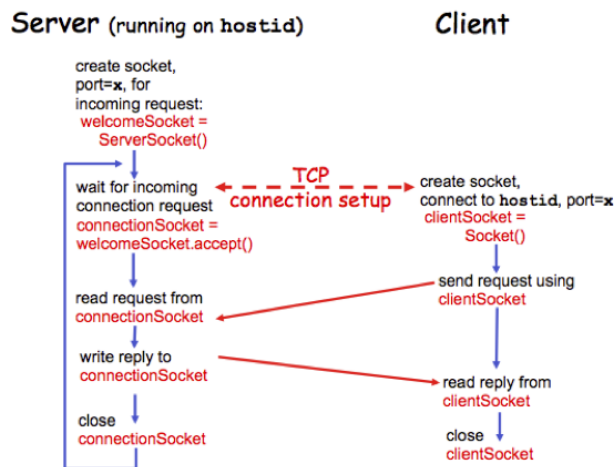
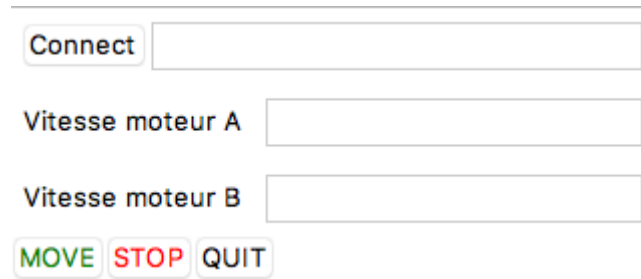


FIGURE 1: Les étapes d'une connexion TCP entre un client et un serveur.

2 Travail à réaliser

Nous souhaitons programmer un simple **panneau de contrôle** pour commander le robot. Il s'agit d'une interface homme machine (IHM) qui ressemble à la figure suivante :



The figure shows a simple graphical user interface for controlling a LEGO robot. It consists of the following elements:

- A **Connect** button followed by a text input field for entering an IP address.
- A label **Vitesse moteur A** followed by a text input field for entering the speed of motor A.
- A label **Vitesse moteur B** followed by a text input field for entering the speed of motor B.
- Three buttons at the bottom: **MOVE** (green), **STOP** (red), and **QUIT** (grey).

FIGURE 2: Panneau de contrôle du robot LEGO

Le champ devant le bouton Connect, permet de saisir l'adresse IP du robot. Ensuite, en appuyant sur Connect, une connexion est établie avec le robot. Les deux champs vitesse moteur A et vitesse moteur B permettent de saisir les vitesses respectives de moteurs A et B. Le bouton MOVE permet d'activer les moteurs en envoyant un message au serveur qui contient les vitesses de A et B, et le bouton STOP permet d'arrêter les moteurs en envoyant un message contenant le mot STOP au serveur. Le bouton QUIT, permet de quitter l'application et ferme la connexion avec le serveur.

Cette interface graphique est à réaliser avec la bibliothèque **tkinter**. Vous trouverez sous Arche, le fichier fourniture **tutorat2.zip** qui contient les squelettes de deux programmes à réaliser :

- le programme **panel.py**. Il contient le code de l'application tkinter et aussi une classe TCPClient pour réaliser la connexion et l'échange d'information avec le programme serveur.
- le programme **robot_server.py** qui contient le code pour lancer un serveur TCP sur le robot et interagir avec le programme client.

Vous allez maintenant, réaliser les tâches suivantes :

- Vous allez compléter le programme **robot_server.py** pour exécuter les commandes envoyées par le programme client : déplacer le robot et arrêter le robot. Il faut compléter la fonction **executer_commande** de la classe **Handler_TCPServer** qui prend en argument le message envoyé par le client.
- Dans une deuxième étape, vous allez ajouter dans l'interface graphique du programme client un label texte qui indique l'état du robot : marche ou arrêt. Cet état est à récupérer de la réponse du serveur suite à l'exécution de votre message de commande.