

# Démonstrateur retournement d'une pièce par deux robots

Notice d'utilisation




## Table des matières

Téléverser le programme sur la carte Arduino.....	3
Lancer le programme sur Niryo Studio.....	3
Principe général de fonctionnement des programmes.....	3
Arduino.....	3
setup().....	3
initialiseRobot1().....	3
initialiseRobot2().....	3
loop().....	3
move1().....	4
Robots.....	4
Robot 1.....	4
Robot 2.....	5

## Téléverser le programme sur la carte Arduino

Une fois le programme écrit, vous pouvez le téléverser sur la carte Arduino. Pour ce faire, vous devez d'abord indiquer à l'environnement Arduino, sur quel type de carte vous allez mettre le programme. Cliquez sur Outils, puis sur Type de carte, et choisissez la bonne carte « Arduino/Genuino Mega ou Mega 2560 ». Ensuite, vous devez spécifier sur quel port la carte est connectée. Cliquez sur Outils, puis Port, et choisissez le port avec le nom de la carte – par exemple, COM3 (Arduino/Genuino Mega ou Mega 2560). Vous pouvez enfin cliquer sur le bouton de téléversement (celui avec une flèche).

## Lancer le programme sur Niryo Studio

Lorsque les programmes sont sur leur carte, vous devez lancer le programme sur le robot. Pour cela, lancez l'application Niryo Studio, connectez-vous au robot, cliquez sur cet icône , et cliquez sur les deux flèches pour ajouter votre programme, cherchez le bon et ajoutez-le sur Niryo Studio. Vous pouvez ensuite cliquer sur la flèche verte pour démarrer le programme sur le robot.

## Principe général de fonctionnement des programmes

### Arduino

Définition des pins du robot, des leds et du bouton. Puis on rentre dans la fonction `setup()`.

#### `setup()`

Définition des pins en tant qu'Input ou Output, puis on effectue les fonctions `initialiseRobot1()` et `initialiseRobot2()`.

#### `initialiseRobot1()`

Cette fonction met les pins du robot 1 à l'état bas.

#### `initialiseRobot2()`

Cette fonction met les pins du robot 2 à l'état bas.

Une fois les configurations terminées, on entre dans la boucle.

#### `loop()`

On fait clignoter une fois les deux leds. Puis on attend que le bouton poussoir soit appuyé. Lorsque c'est le cas, on entre dans la fonction `move1()`.

#### `move1()`

On commence par inspecter l'état des pins 1C des deux robots, et on exécute la suite du programme uniquement si les deux pins sont à l'état bas. Ces 2 pins représentent l'état des robots, s'ils sont à

l'état bas, cela signifie que les robots n'effectuent aucune action et sont donc prêt à recevoir des commandes.

On commence par allumer la led correspondante au robot 1, et on s'assure que la led correspondante au robot 2 est éteinte. On met ensuite le pin 1A du robot 1 à l'état haut, pour que celui-ci effectue son premier mouvement. On attend ensuite qu'il ait fini son déplacement. Pour ce faire, on inspecte le pin 1C et on attend qu'il passe à l'état Haut. Lorsque c'est le cas, le robot a fini son déplacement. On repasse alors le pin 1A à l'état Bas pour arrêter d'envoyer la commande.

On passe ensuite au déplacement du robot 2, pour cela, on allume la led correspondante au robot 2 et on éteint celle correspondante au robot 1. Puis on passe à l'état Haut le pin 1A du robot 2. Le robot se déplace et on attend qu'il ait fini son mouvement, comme pour le robot 1, on attend que le pins 1C passe à l'état Haut. Puis on repasse le pin 1A à l'état Bas pour arrêter d'envoyer la commande.

On effectue la suite du mouvement du robot 1 en allumant la led correspondante et en mettant le pin 1B du robot à l'état Haut, on attend que le robot 1 ait fini son mouvement toujours de la même façon, puis on passe le pin 1B à l'état Bas.

Et enfin, on effectue la seconde et dernière partie du mouvement du robot 2 en allumant la bonne led et en passant le pin 1B du robot à l'état Haut. On attend la fin du mouvement et on repasse le pin 1B à l'état Bas.

Lorsque ceci est fait, la boucle retourne à son point de départ, et attend qu'on appuie sur le bouton poussoir.

## Robots

### Robot 1

On commence par définir les pins du robot en tant qu'Input (pin 1A et 1B) ou Output (pin 1C), puis on se déplace jusqu'à la position de départ, et on rentre dans la boucle.

On met le pin 1C à l'état Bas, ce pin définit si le robot effectue une action ou non. Ensuite, on évalue les pins 1A et 1B.

Si le pin 1A est à l'état Haut et que le pin 1B est à l'état Bas, on effectue la première partie du déplacement, on se déplace jusqu'à l'objet, on ouvre le gripper, on saisi l'objet, puis on le déplace à mi-chemin vers le second robot, et on passe le pin 1C à l'état Haut pour indiquer que le robot à effectué la première partie de son déplacement.

Si le pin 1A est à l'état Bas et le pin 1B à l'état Haut, on effectue la seconde partie du déplacement du robot 1. On ouvre le gripper, on attend un peu, on se déplace jusqu'à la position de départ et on passe le pin 1C à l'état Haut pour indiquer que le déplacement est terminé.

### Robot 2

La structure du programme du second robot est sensiblement la même que celle du programme pour le premier robot.

On commence par définir les pins du robot en tant qu'Input (pin 1A et 1B) ou Output (pin 1C), puis on se déplace jusqu'à la position de départ, et on rentre dans la boucle.

On met le pin 1C à l'état Bas, ce pin définit si le robot effectue une action ou non. Ensuite, on évalue les pins 1A et 1B.

Si le pin 1A est à l'état Haut et que le pin 1B est à l'état Bas, on effectue la première partie du déplacement, on se déplace jusqu'à l'objet, on ouvre le gripper, on saisi l'objet, et on passe le pin 1C à l'état Haut pour indiquer que le robot à effectué la première partie de son déplacement.

Si le pin 1A est à l'état Bas et le pin 1B à l'état Haut, on effectue la seconde partie du déplacement du robot 1. On retourne l'objet, on le dépose devant le robot 2, on retourne à la position de départ et on passe le pin 1C à l'état Haut pour indiquer que le déplacement est terminé.