|  |  |
| --- | --- |
| **TP 1 - Cobotique**  **Palettisation** |  |

# Introduction

Les robots collaboratifs (ou cobots), comme leur nom l’indique, sont beaucoup utilisés dans des applications de collaboration entre l’utilisateur et le robot, car ils sont plus limités en vitesse, effort etc, et donc moins dangereux que les robots industriels classiques. Cependant, ils sont largement utilisés pour réaliser des tâches industrielles, comme le soudage et l’usinage ou même pour des applications non-industrielles comme l’enseignement et la cuisine.

Dans ce TP, on utilisera un cobot UR3, le modèle le plus petit mais quand-même très puissant de la gamme de robots Universal Robots. On va mettre en place un programme de palettisation dont l’objectif est de remplir une boîte de chocolat avec des chocolats placés en pile. Pour cela, on utilisera l’apprentissage de points et une fonctionnalité très utile du robot, le contrôle en effort.

D’abord, avant de passer à l’étape de programmation, on va apprendre à configurer le robot et son environnement, en définissant ses plans de travail et l’outil utilisé.

Les étapes principales du TP seront :

* Création des plans de travail
* Configuration de l’outil
* Programmation
* Optimisation programme

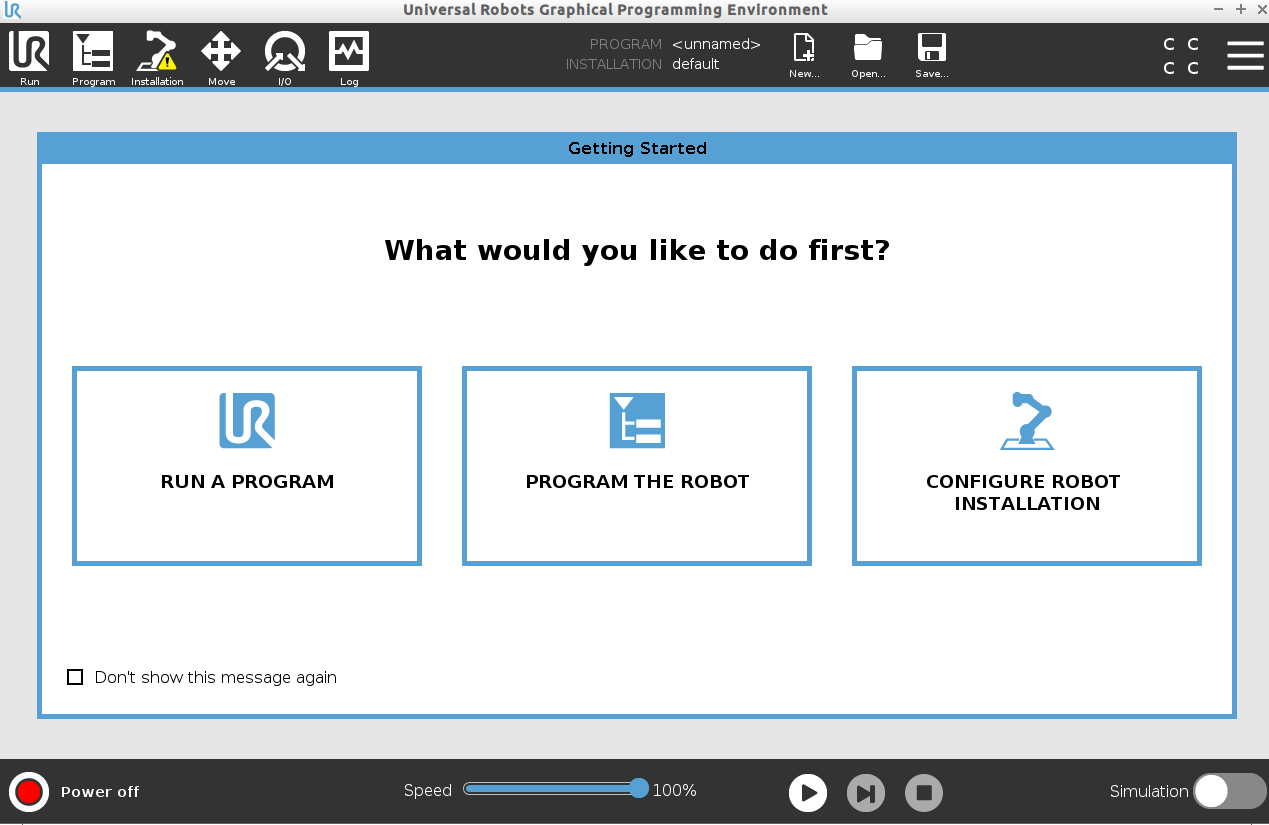
# Démarrage du Système

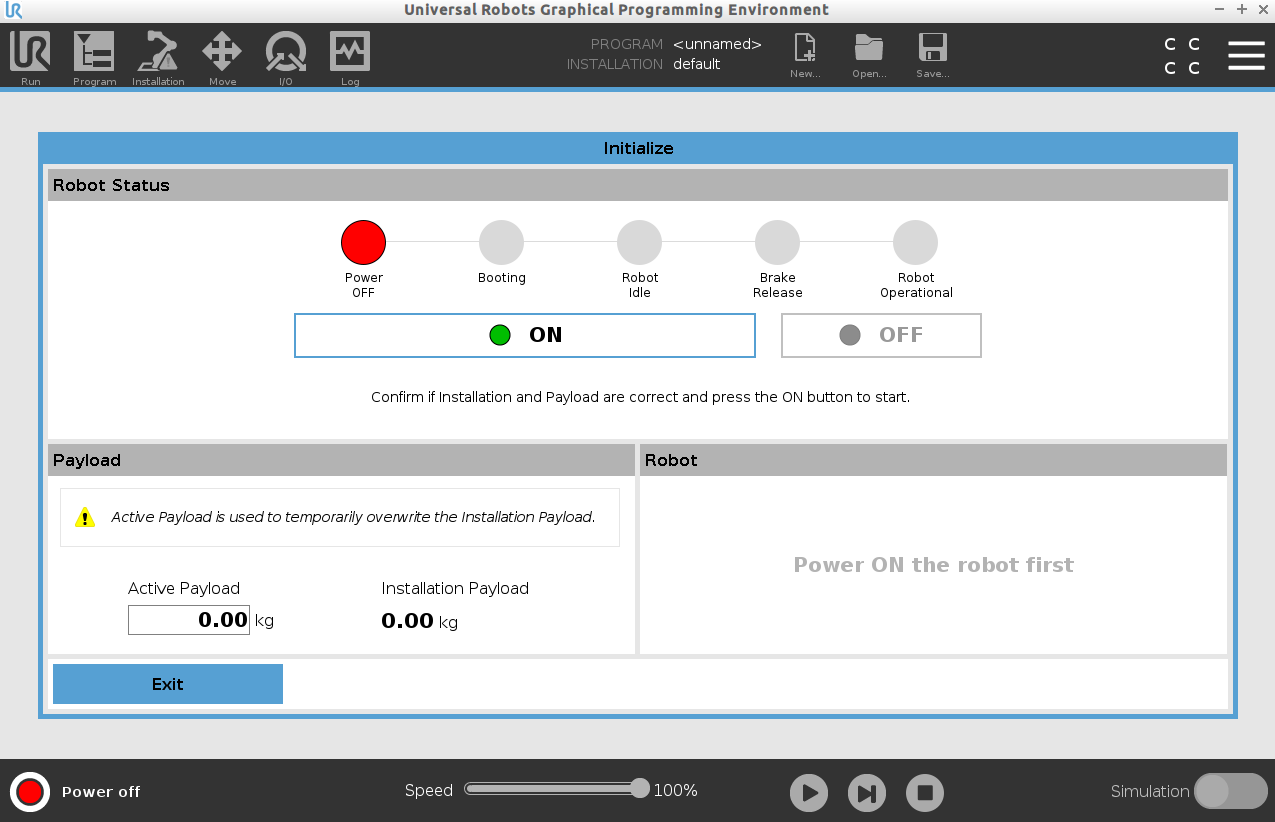
## Démarrage robot

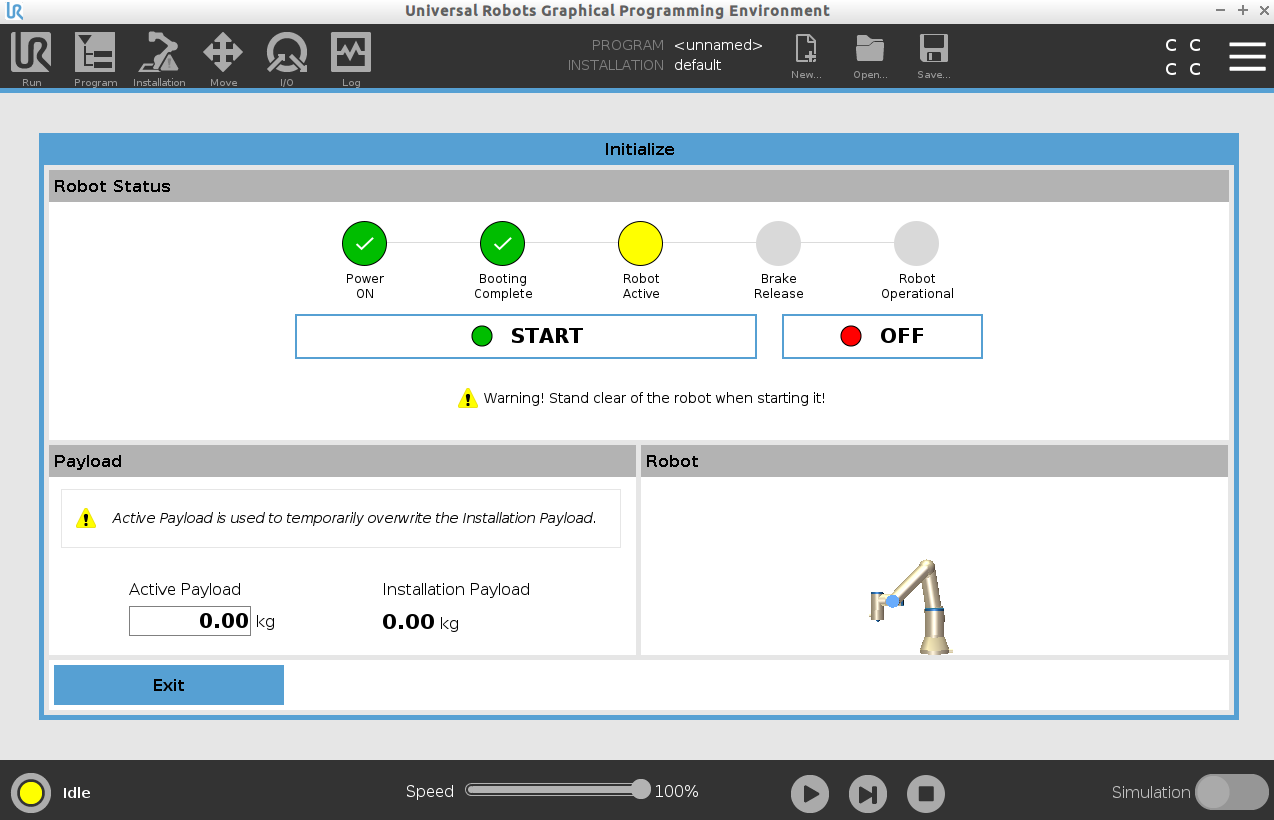
* Allumez le système en appuyant sur le bouton de démarrage de la tablette, cela peut prendre quelques minutes.

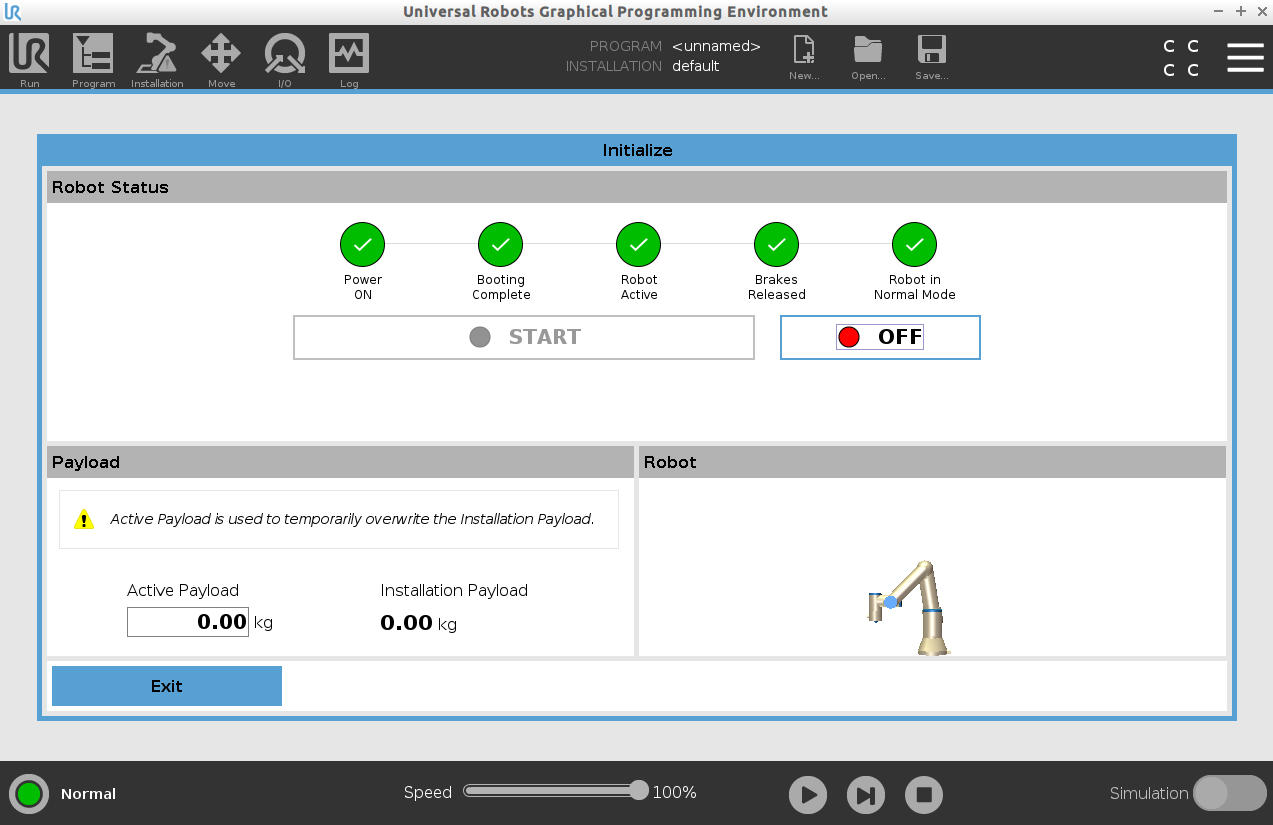
image

* Sur polyscope, démarrez le robot en suivant les étapes ci-dessous :





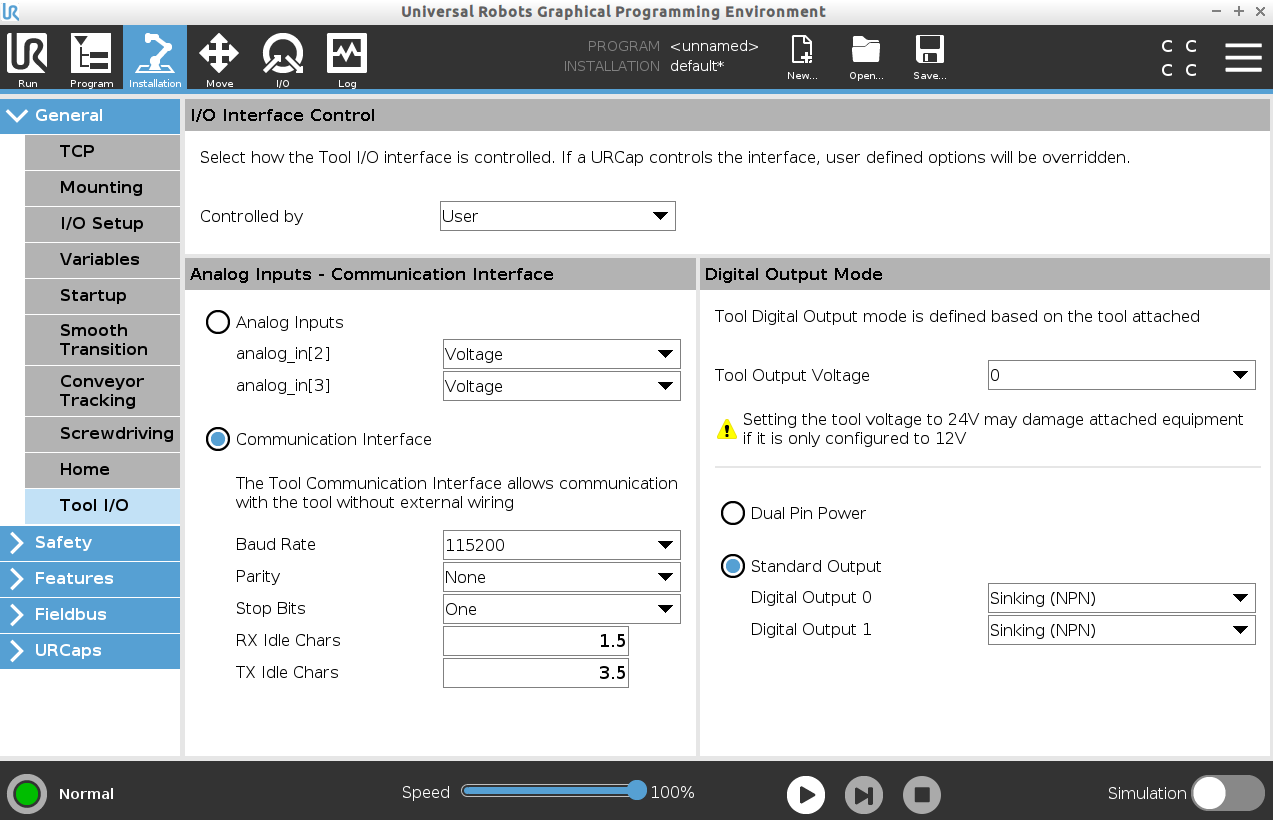




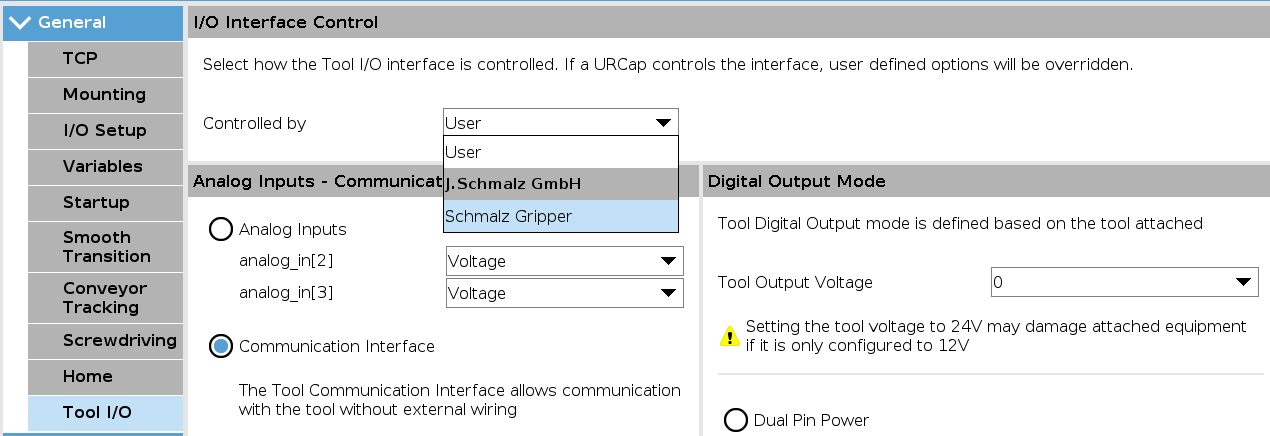
## Connexion de l’outil

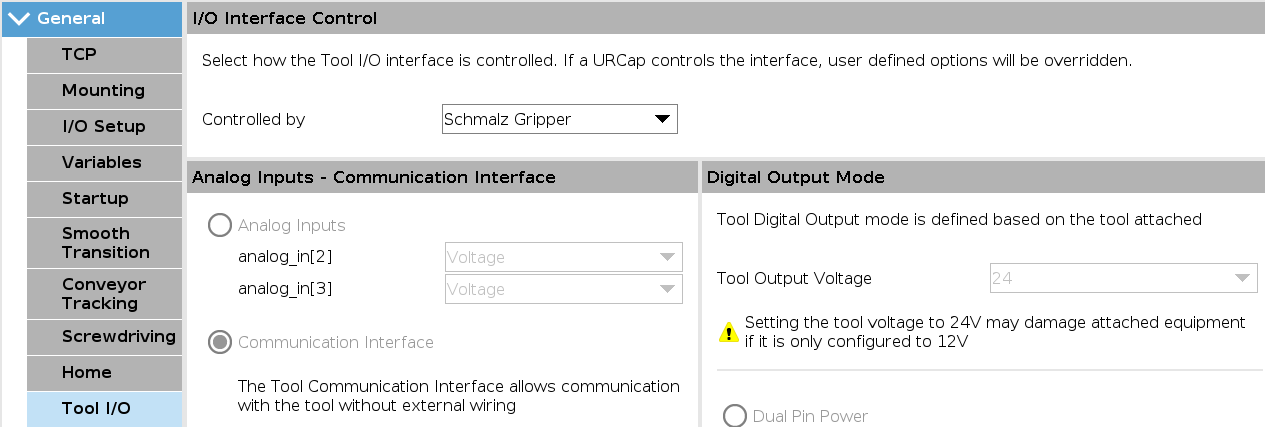
Pour pouvoir utiliser l’outil et ses fonctions, on a besoin de le connecter au robot, pour faire cela, suivez les étapes suivantes :

* Allez sur l’onglet « Installation » en haut de l’interface, puis, allez sur « General > Tool I/O ».

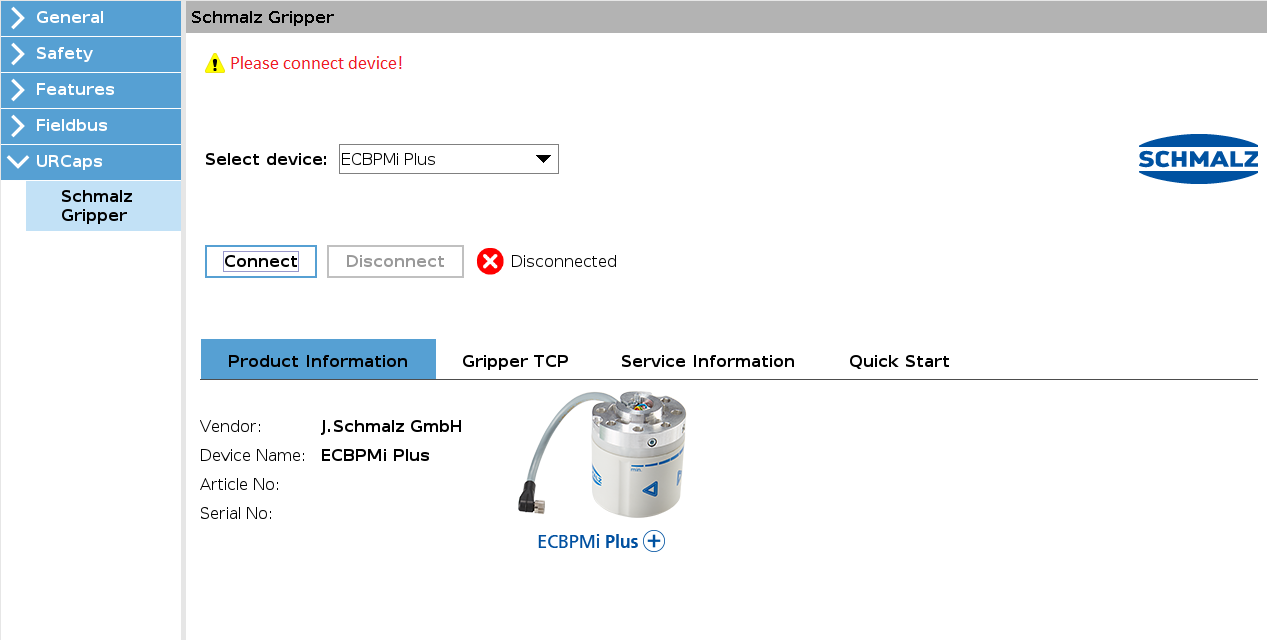


* Dans « I/O Interface Control », ouvrez la liste « Controlled by » et sélectionnez l’option « Schmalz Gripper »





* Après, cliquez sur « URCaps > Schmalz Gripper », une fenêtre de connexion sera affichée, sélectionnez « ECBPMi Plus » dans le menu déroulant.



* Appuyez sur le bouton « Connect » et attendez quelques secondes, l’état « Connected » s’affichera quand l’outil sera connecté.

|  |  |
| --- | --- |
|  |  |

Maintenant qu’on a démarré le robot et l’outil, on peut commencer. Pour ce TP on va suivre quelques étapes pour configurer le robot et le programmer.

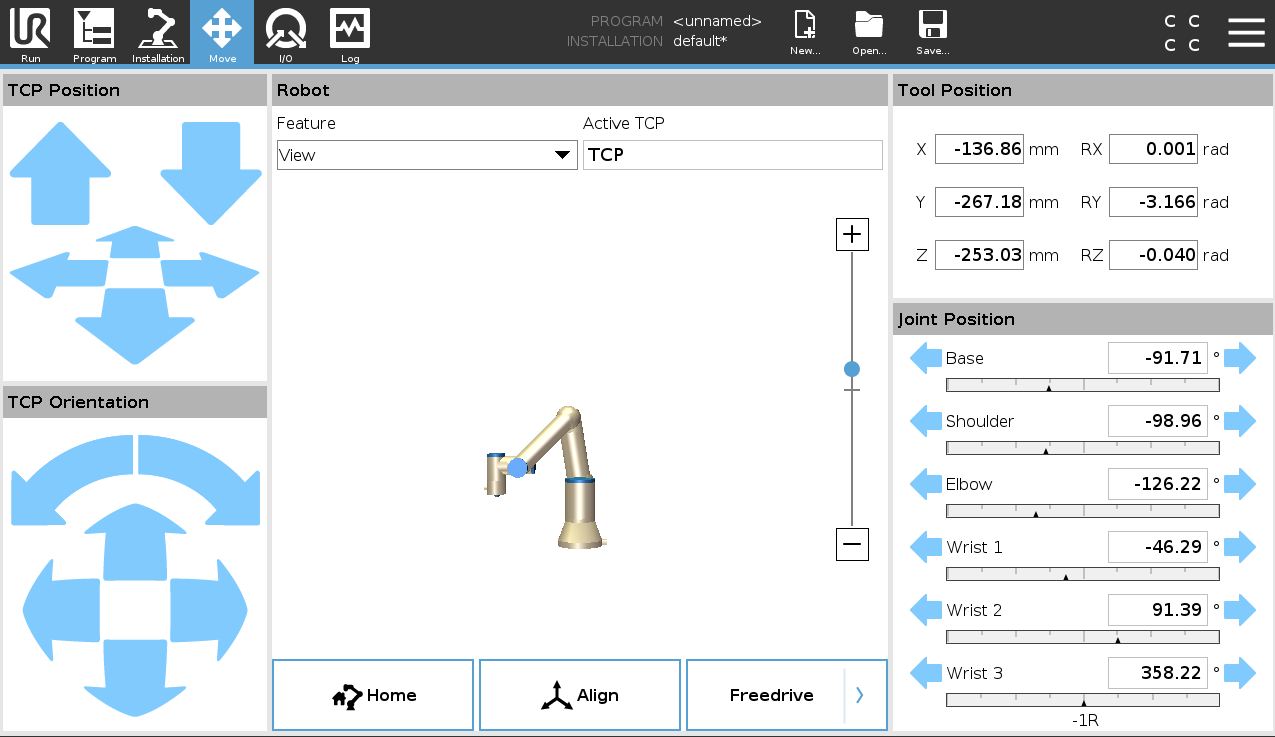
Pour pouvoir faire la configuration et la programmation, on doit apprendre comment manipuler le robot.

# Manipuler le robot

On a 4 formes différentes de déplacement :

1. Un mouvement cartésien, à l’aide des flèches de translation et rotation
2. Un mouvement articulaire pour tourner chaque articulation de -360° - 360° de façon indépendante
3. Entrer les valeurs qu’on veut pour les coordonnées (X, Y, Z, RX, RY, RZ) et le robot se déplacera tout seul vers le point choisi.
4. Utiliser le mode « Freedrive », qui nous permet de déplacer librement le robot avec nos mains.

Tous ces déplacements sont disponibles dans l’onglet « Move » en haut de l’interface. Ouvrez l’onglet.



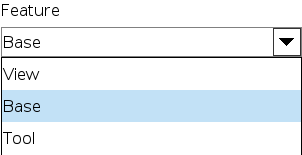
**(C)**

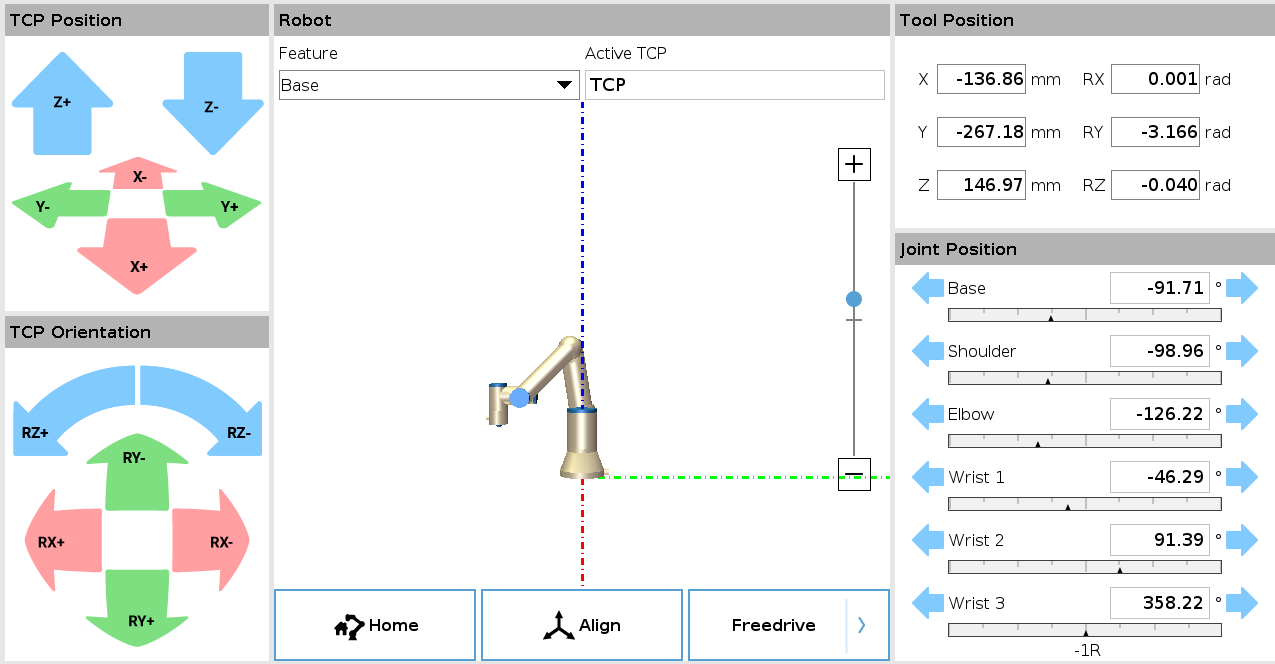
**(D)**

**(B)**

**(A)**

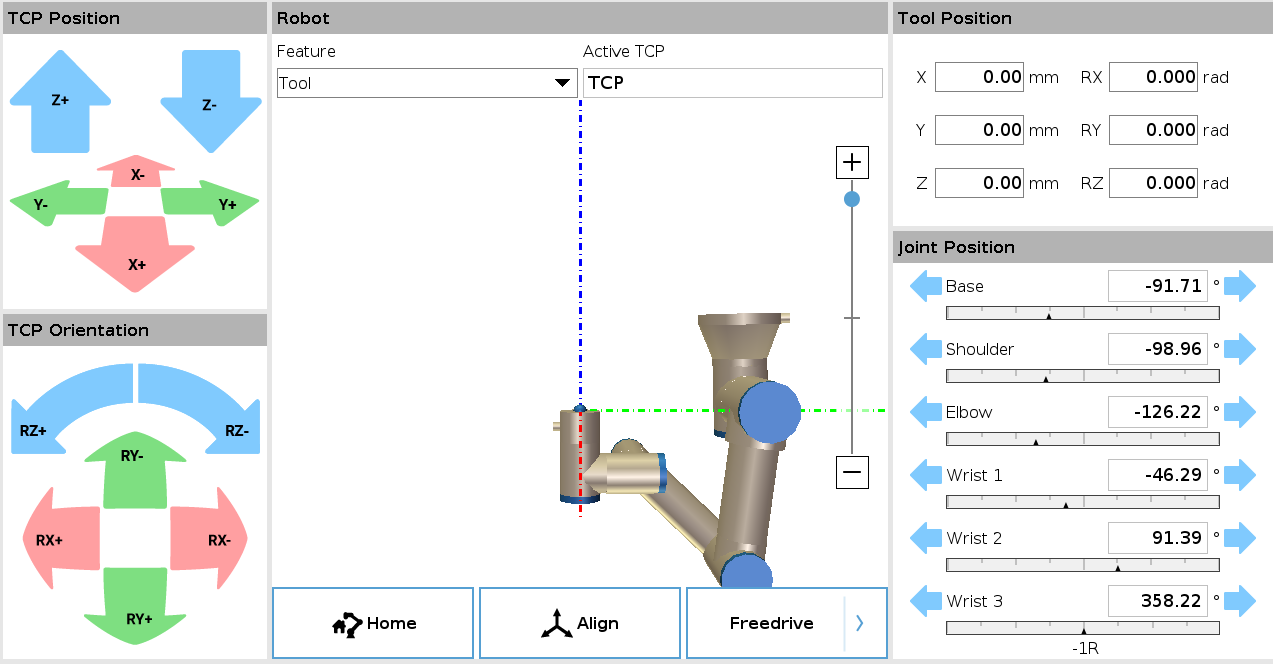
Dans cet onglet, sur « Robot », on a aussi deux champs importants, le « Feature » et le « Active TCP ». Le menu déroulant « Feature » sert à choisir le plan par rapport auquel on veut bouger le robot, par exemple, si on sélectionne « Base » dans la liste, cela va faire que le mouvement (A) et (C) soient relatives au plan « Base » du robot.





On voit que la couleur des flèches change et une visualisation des axes du plan apparaissent.

Si on choisit l’option « Tool » dans la liste, les axes du TCP actif apparaîtront.



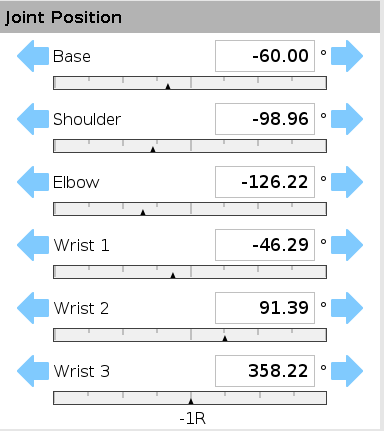
On verra comment changer le TCP actif dans la section 4.

## Mouvement cartésien

Ce type de mouvement est localisé à gauche de l’interface, dans le bloc en haut se trouvent les flèches responsables par la translation du robot dans tous les axes et dans le bloc en bas on a les flèches responsables par la rotation par rapport aux axes du plan choisi.

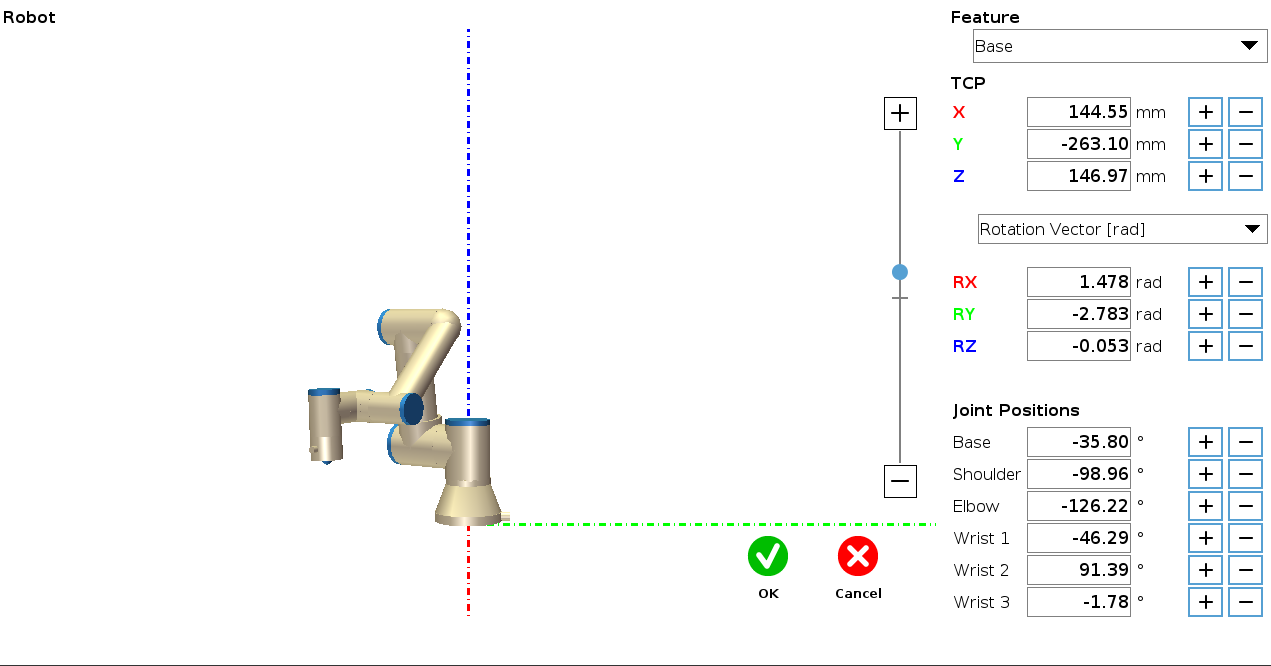
## Mouvement articulaire

Avec ce mouvement on peut tourner chaque articulation du robot (on a 6 au total) de façon indépendante soit à l’aide soit des flèches, soit en entrant la valeur d’angle qu’on veut dans le champ correspondant.

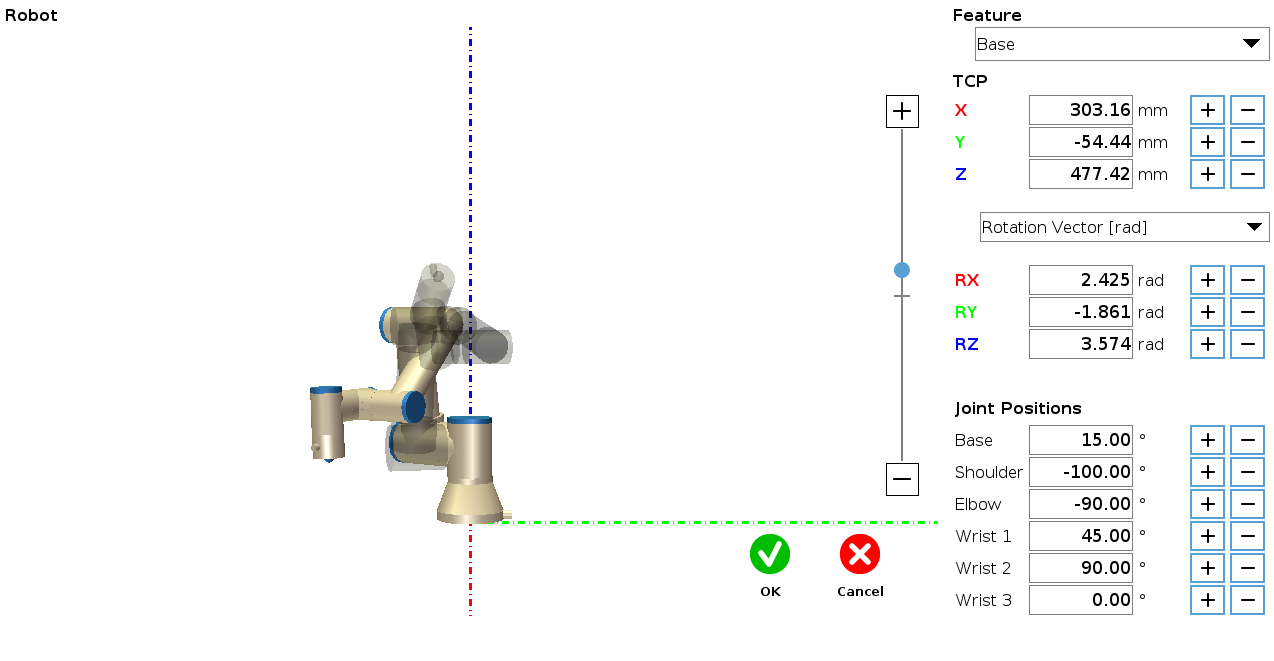


Tenez appuyé une des flèches pour tourner l’articulation, attention, vous ne pouvez appuyer qu’une seule flèche à la fois.

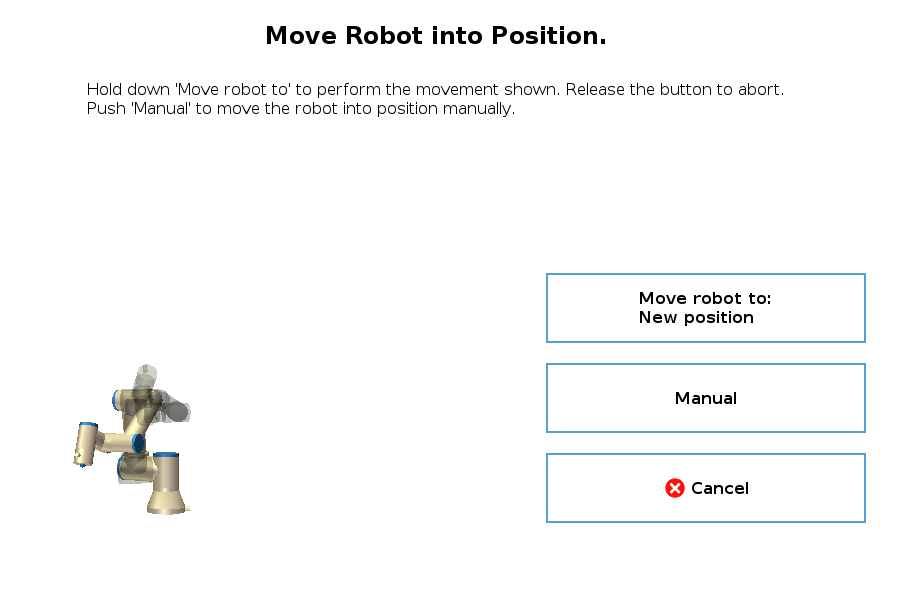
Pour entre une valeur d’angle spécifique, cliquez sur le champ de l’articulation choisie, Polyscope va vous afficher une nouvelle fenêtre :



Dans cette fenêtre, sur « Joint Positions » comme marqué dans l’image ci-dessus, cliquez sur le/les champ/s correspondant à l’articulation choisie et entrez une valeur entre -360° et 360°. Vous pouvez modifier l’angle pour toutes les articulations dont vous voulez tourner, il suffit de modifier la valeur dans leurs champs.



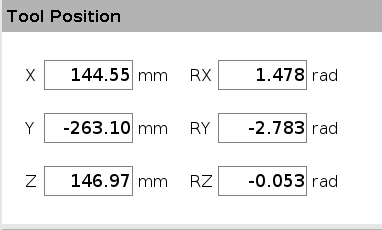
Après finir de faire la modification, un fantôme représentant la nouvelle position va apparaître. Appuyez sur « OK », encore une nouvelle fenêtre sera affichée.



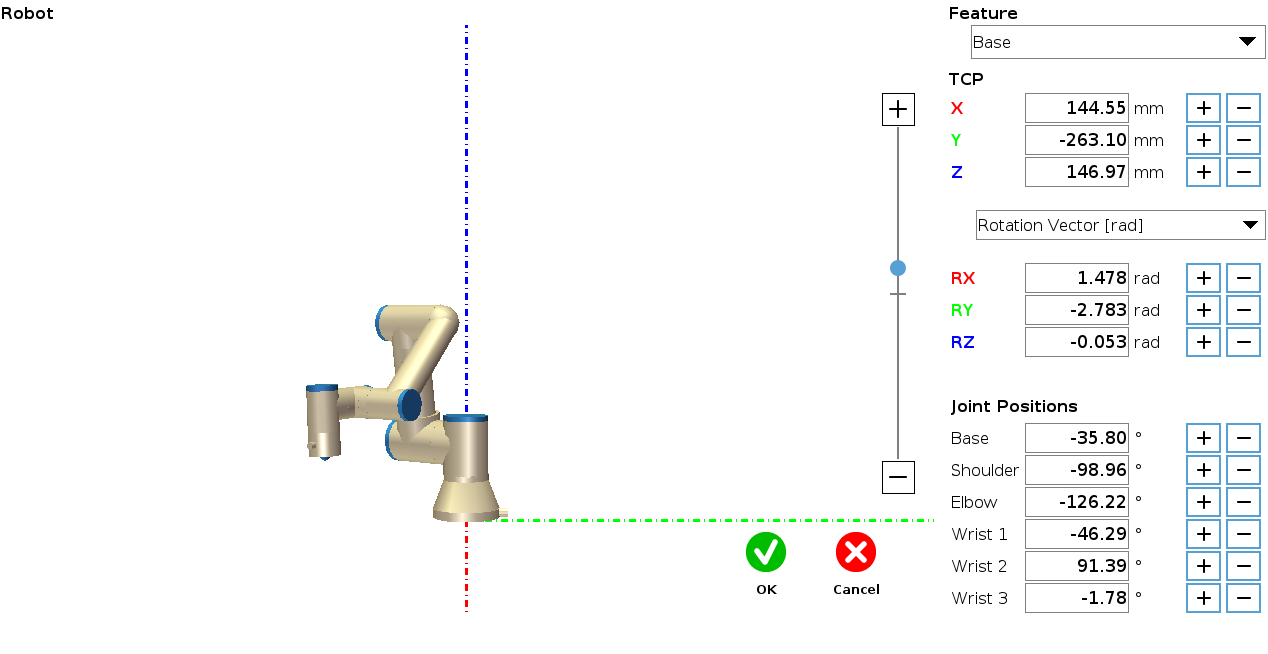
Maintenant, tenez appuyé le bouton « Move robot to : New position », le robot se déplacera automatiquement vers la nouvelle position.

## Mouvement par coordonnée

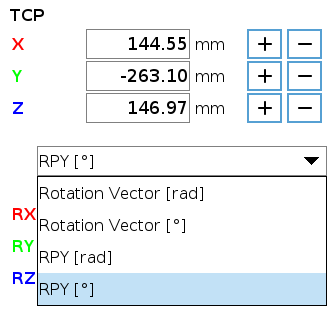
Ce troisième type fonctionne pareil au mouvement précédent, sauf que maintenant on doit entrer les coordonnées en translation et rotation (X, Y, Z, RX, RY, RZ) du point où on veut placer le robot.



Cliquez sur un des champs, la fenêtre ci-dessous sera affichée.



Vous pouvez noter aussi qu’on a un menu déroulant pour la rotation, avec ce menu on peut choisir le type de rotation et si on veut des valeurs en radian ou degrés. Sélectionnez « RPY [rad] » ou « RPY [°] » pour tourner le TCP par rapport aux axes du plan actif.



De la même manière que pour le mouvement articulaire, on doit entrer les valeurs qu’on veut, appuyer sur « OK » et puis sur « Move robot to : New position » pour bouger le robot, sauf que maintenant on se servira des champs disponibles dans la propriété « TCP ».

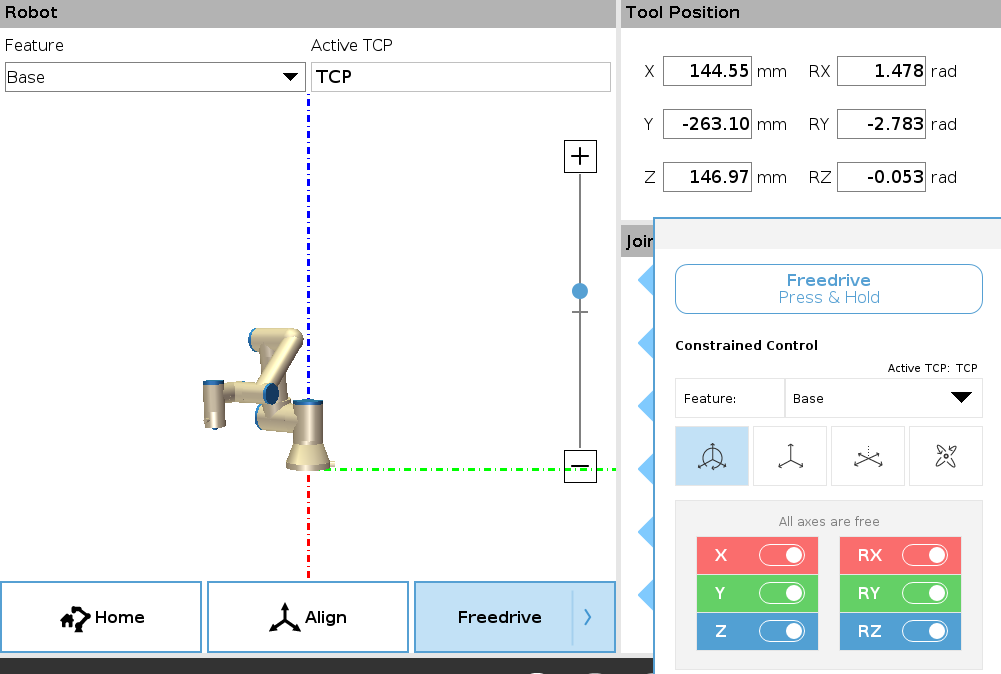
## Freedrive

Finalement, le quatrième mode de déplacement du robot s’appelle « Freedrive » (conduite libre), qui est une fonctionnalité très utile pour faire de l’apprentissage de trajectoires. Le Freedrive, comme son nom l’indique, nous permet de bouger librement toutes les articulations du robot avec nos mains, car il relâche ses freins.

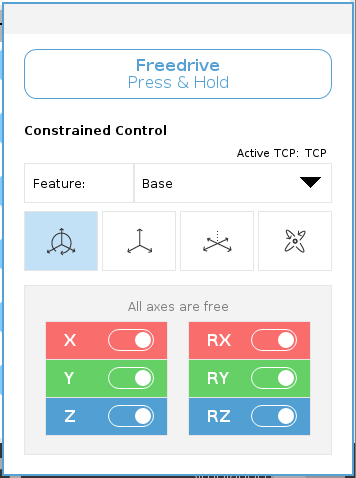
Image exemple Freedrive

On a deux manières d’activer le Freedrive, par le bouton dans l’interface graphique « Freedrive : Press & Hold » ou en appuyant le bouton au-dessus de la tablette, comme montré ci-dessous :

Image bouton Freedrive



En plus, dès que vous cliquez un de ces boutons, Polyscope vous affichera quelques options, montrées dans l’image ci-dessous.



**(B)**

**(A)**

Dans ce menu on a 3 blocs importants, le bloc « Feature » où on peut choisir le plan par rapport auquel on veut déplacer le robot, le bloc (A) où on peut choisir entre 4 options de mouvement différent (Libre, PlanXYZ, PlanXY, Rotation) et le bloc (B) avec lequel on peut personnaliser notre mouvement en bloquant ou débloquant une ou plusieurs translations et/ou rotations.

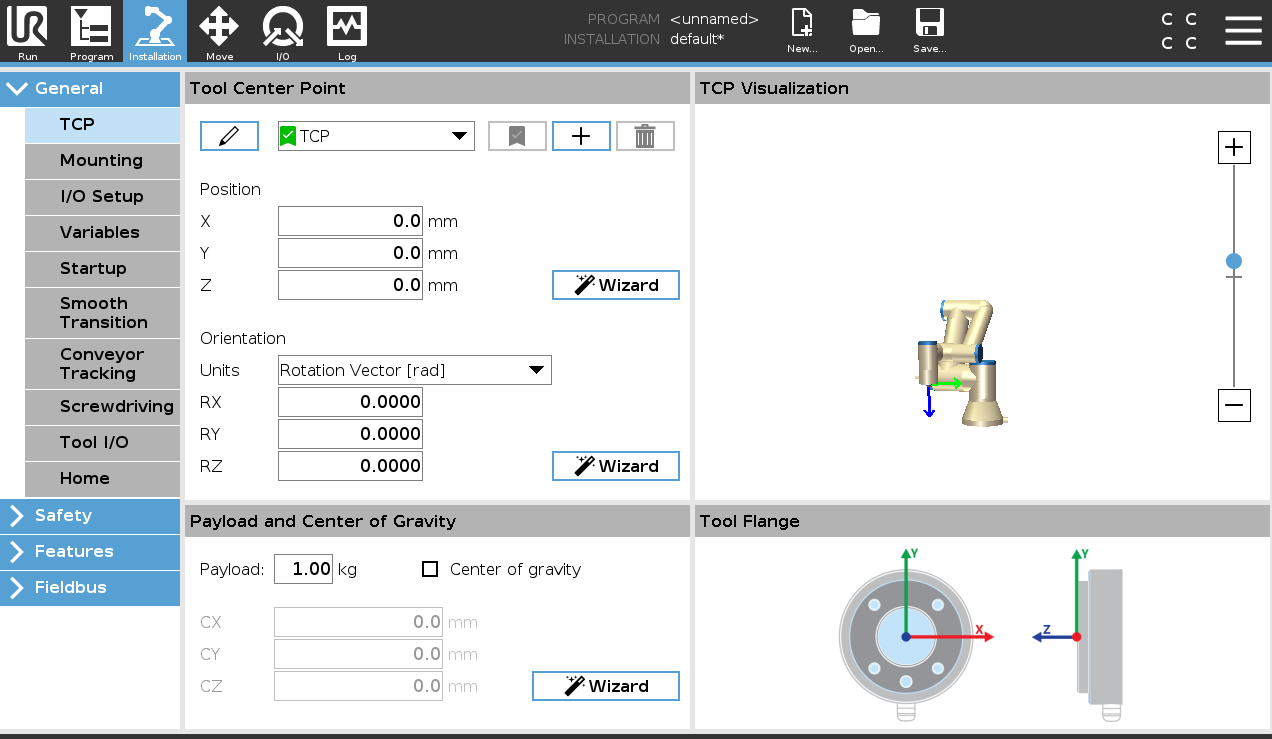
|  |  |
| --- | --- |
|  |  |

**Attention 1** : Ne pas forcer le bras robotique avant de déclencher le Freedrive ! Si vous sentez que le robot ne bouge pas, vérifiez si le Freedrive est bien actif. Le robot doit bouger au moindre effort.

**Attention 2** : Déplacez le robot doucement pour éviter les arrêts d’urgence.

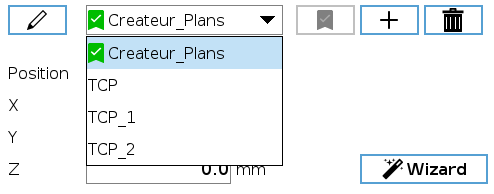
# Configuration Outil

Par défaut, le TCP du robot et sa charge sont configurés comme (0, 0, 0) et 0kg, alors on doit modifier les deux par rapport à l’outil utilisé dans ce TP. Pour cela, dans « Installation », allez sur « General/TCP » pour le TCP et l’orientation ou sur « General/Payload » pour la charge et le centre de gravité.



## TCP

D’abord, renommez le TCP comme « Createur\_Plans » en appuyant sur le bouton stylo à côté du nom « TCP », vous pouvez aussi créer d’autres TCPs si vous avez d’autres outils à ajouter, la liste de TCPs existants montrera tous vos TCPs et pour en choisir un vous devez le sélectionner dans la liste et après cliquer sur l’icône du drapeau à droite pour l’activer, seul le TCP actif aura l’icône du drapeau vert à côté de son nom.

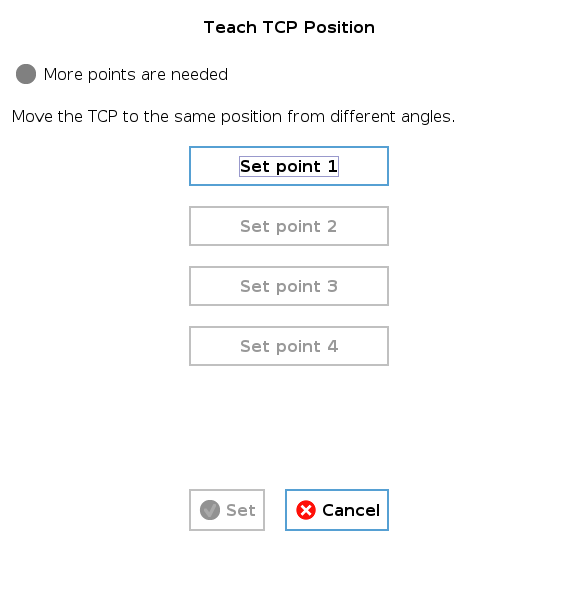


Après avoir renommé le TCP, on doit le configurer. Le TCP (Tool Center Point) est l’endroit de l’outil qui sera en contact avec les pièces, dans notre cas le bout de la ventouse. Pour calculer ses coordonnées, on a deux options, soit mesurer la distance entre l’extrémité du bras du robot et la ventouse à l’aide d’une règle (utile dans les cas où X et Y sont constants), soit utiliser la fonction « Wizard », incluse dans Polyscope et qui calculera automatiquement les coordonnées du TCP.



Cliquez sur « Wizard », le robot vous demandera 4 points sur une même position, mais avec des orientations différentes, utilisez la pièce (C) pour créer les points comme montré dans les images suivantes et puis cliquer sur « Set ».

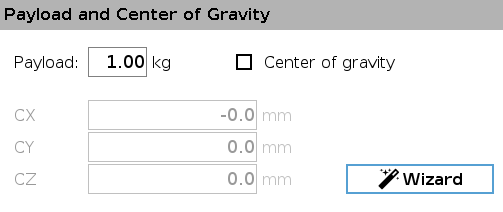
Images avec 2 positions pour comprendre le principe et indiquer le pièce (C) sur l’image



A la fin, vous devez avoir une coordonnée (0, 0, Z). C’est probable que vous ayez X et Y ≠ 0 avec des valeurs petites, cela se doit aux imprécisions de positionnement, simplement changez ses valeurs pour 0 mm.

## Charge

Pour configurer la charge on peut soit peser l’outil à l’aide d’une balance soit utiliser son « Wizard », ce qui nous calculera aussi le centre de gravité de l’outil.



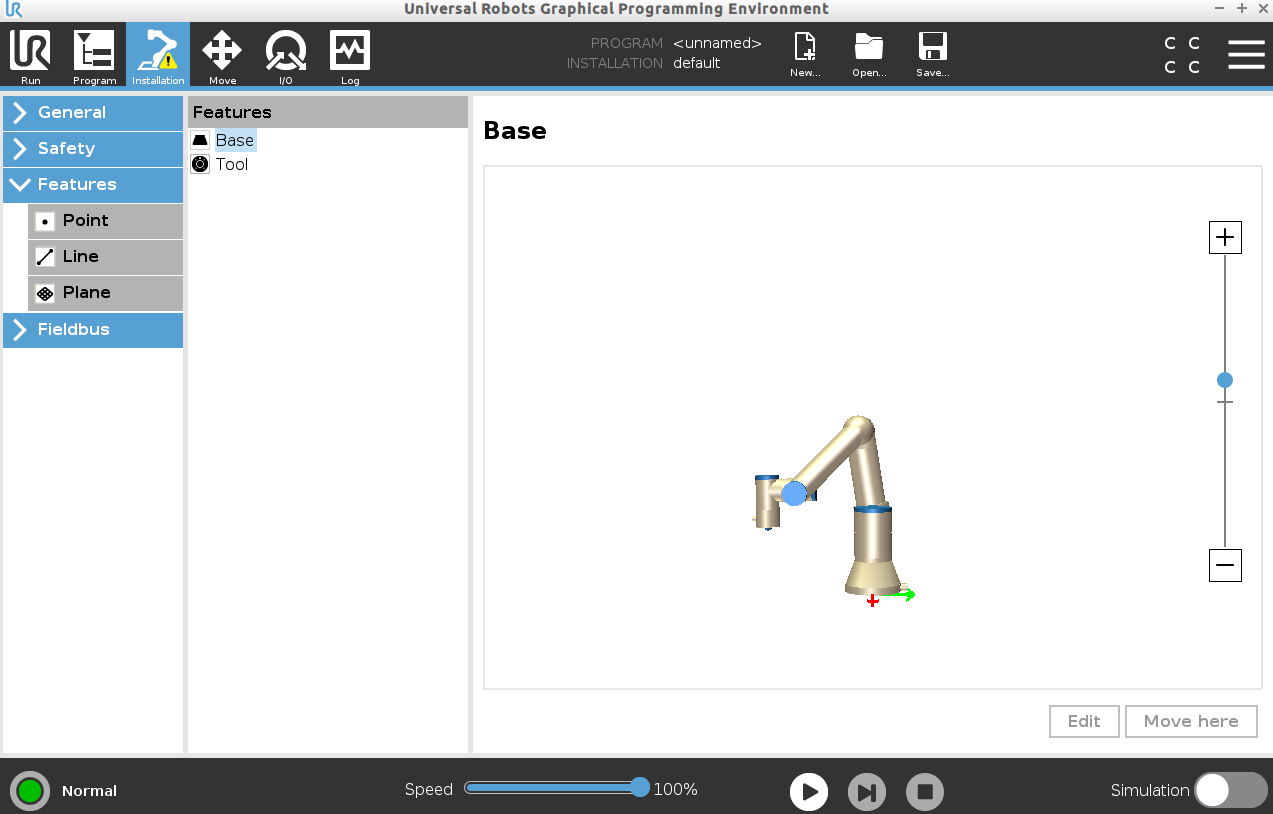
Cliquez sur « Wizard » et suivez les étapes demandées par l’interface, positionnez l’outil dans 4 positions aléatoires différentes, puis appuyez sur « Finish », les nouvelles valeurs de centre de gravité et de charge seront maintenant enregistrées.

# Création des plans

Pour créer les plans, on va se servir de l’embout (A) et de son TCP qu’on a défini précédemment, qui nous permettra de mieux positionner le centre de l’outil.

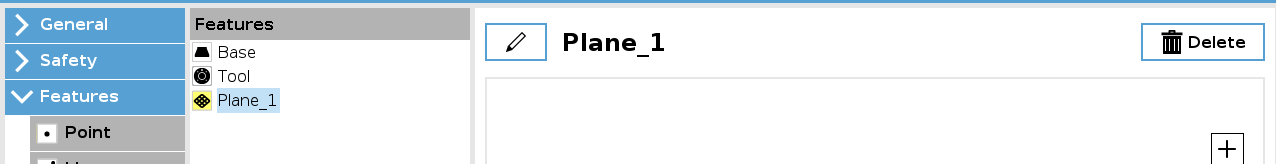
Image embout A

Dans l’interface graphique du robot UR (aussi appelée Polyscope), on a quelques onglets importants. Pour créer les plans de travail, ouvrez l’onglet « Installation > Features », vous avez trois options « Point/Line/Plane » :

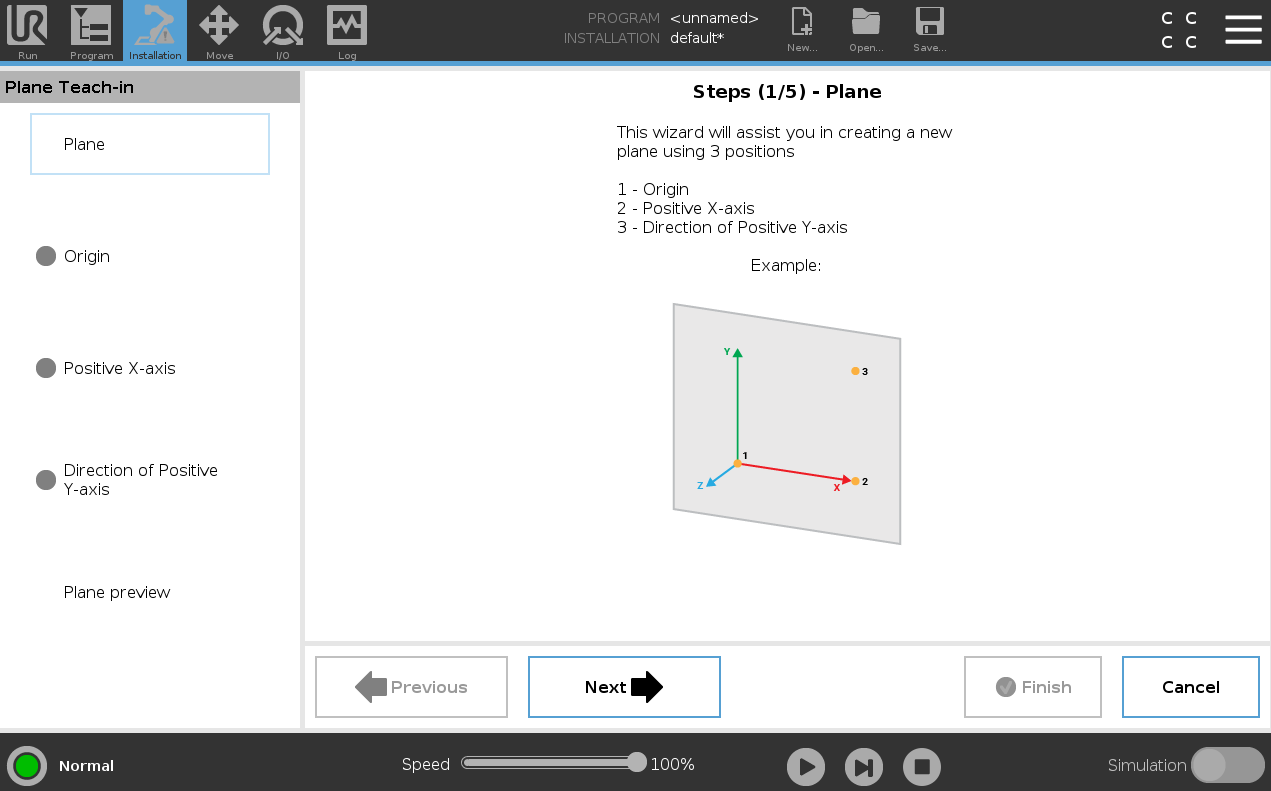


## Piles

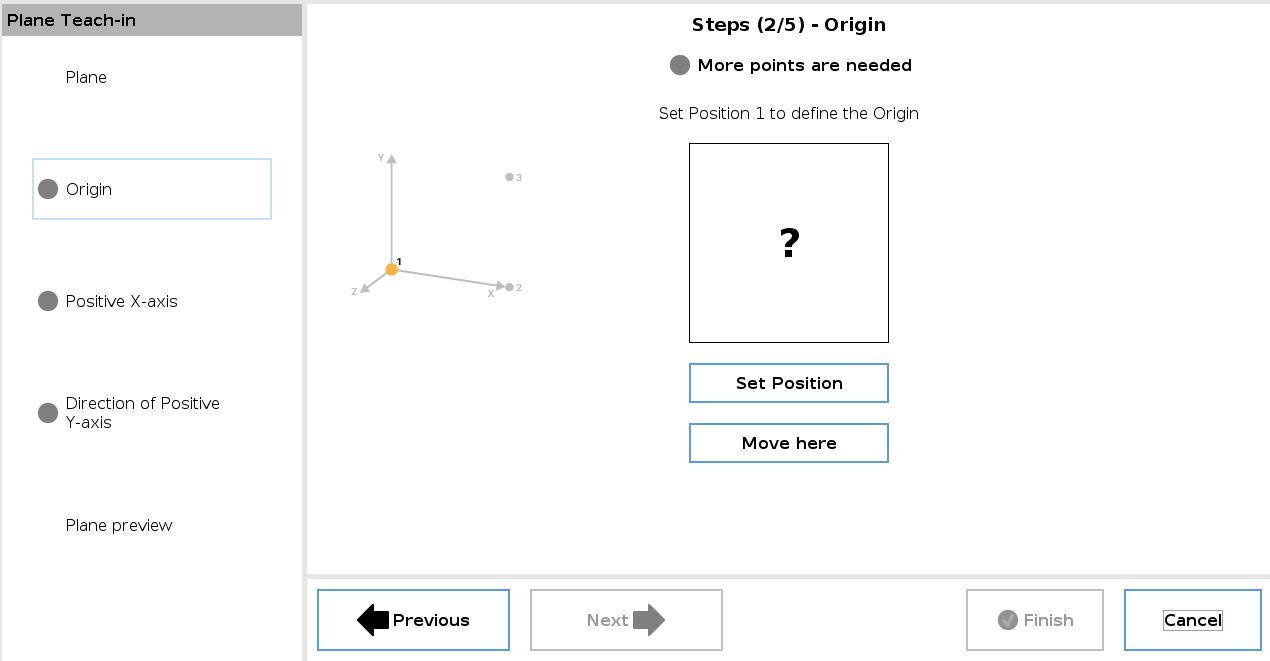
Cliquez sur « Plane », vous allez voir qu’un nouveau plan sera ajouté à la liste des « Features », renommez-le comme « Pile » en appuyant sur l’icône du stylo en haut à côté du nom « Plane\_X ».



Alors, pour apprendre le plan au robot, en bas à droite cliquez sur le bouton « Teach this Plane », le robot vous demandera 3 positions différentes pour créer le plan, son origine, un point sur l’axe X et un point sur l’axe Y.



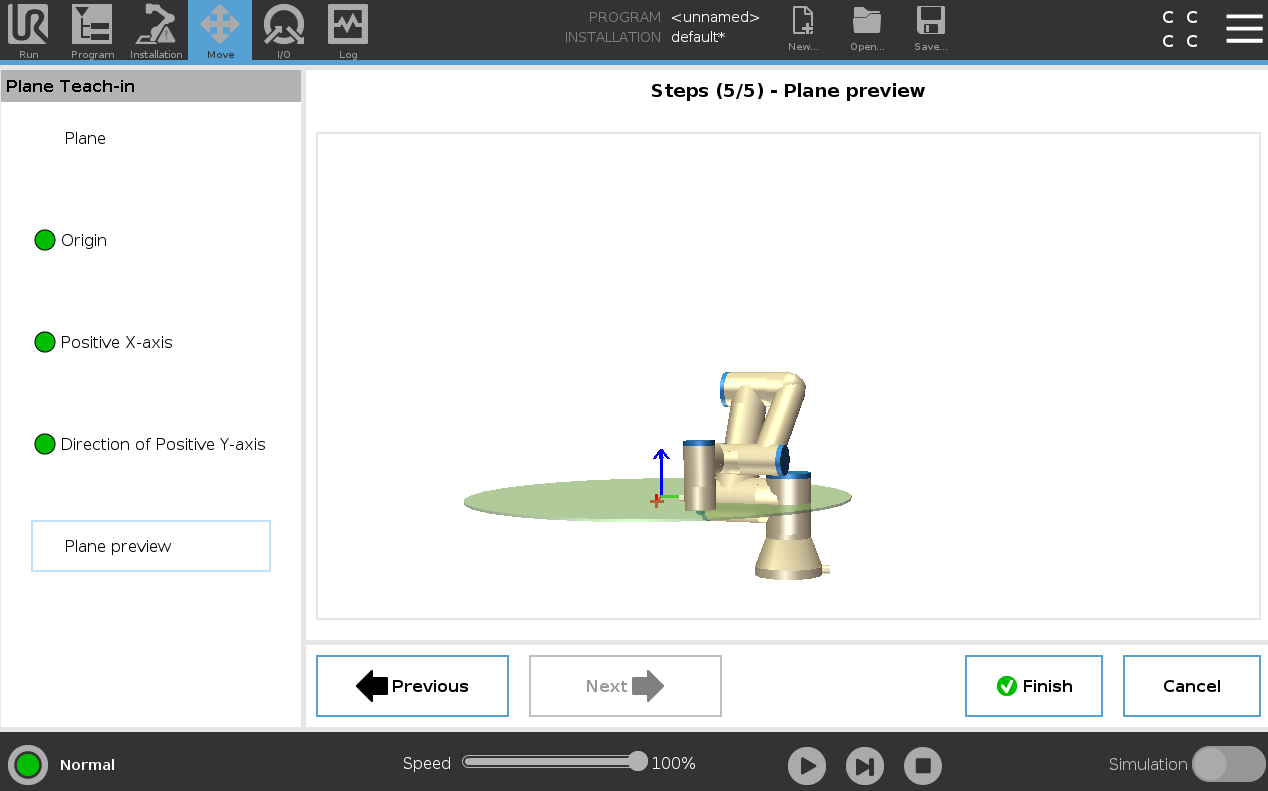
Pour choisir l’origine, appuyez sur « Next » puis « Set position », vous allez être rédirigé vers l’onglet « Move » pour pouvoir bouger le robot.



Pour définir l’origine, positionnez le TCP du robot sur le point 1 (l’extrémité de la pile), cliquez sur « Ok », puis sur « Next » pour passer aux autres étapes, où vous allez positionner le TCP sur le point 2 (axe X) et sur le point 3 (axe Y).

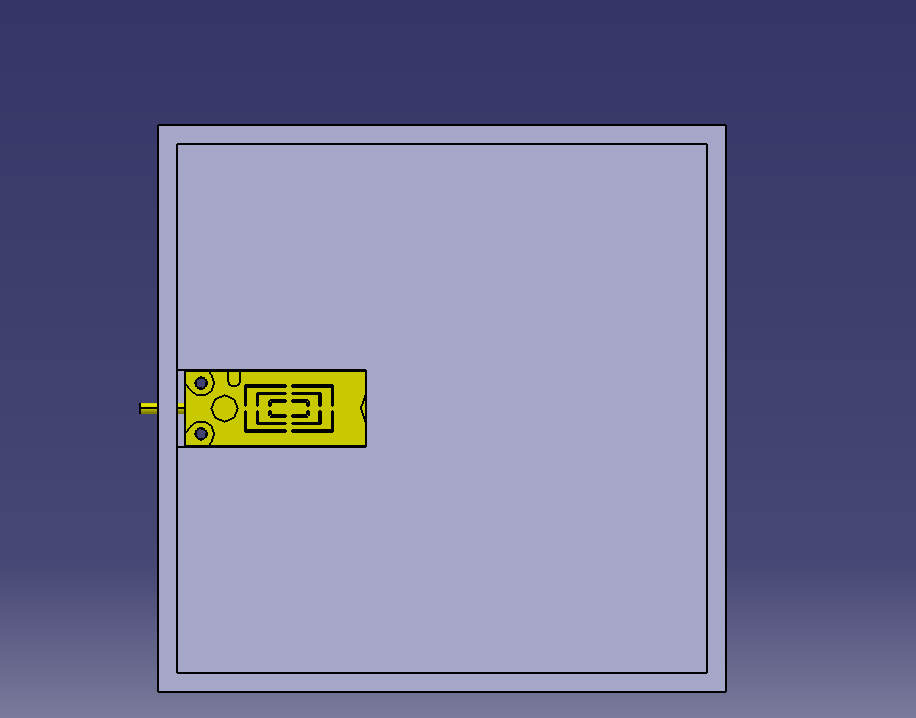
|  |  |
| --- | --- |
| (1)  (3)  (2)  132 mm  50 mm | |
|  |  |

Après suivre ces 3 étapes, le robot calculera automatiquement le sens de l’axe Z et vous affichera une prévisualisation du plan. Appuyer sur « Finish » pour finaliser la création.



## Boîte

De même que la session précédente, vous allez créer un plan pour la boîte de chocolat en le renommant comme « Boite ». Pour l’origine, l’axe X et l’axe Y, placez le TCP du robot aux points 4, 5 et 6, respectivement :



150 mm

150 mm

(5)

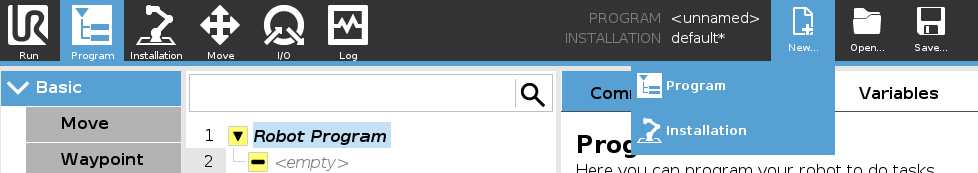
(6)

(4)

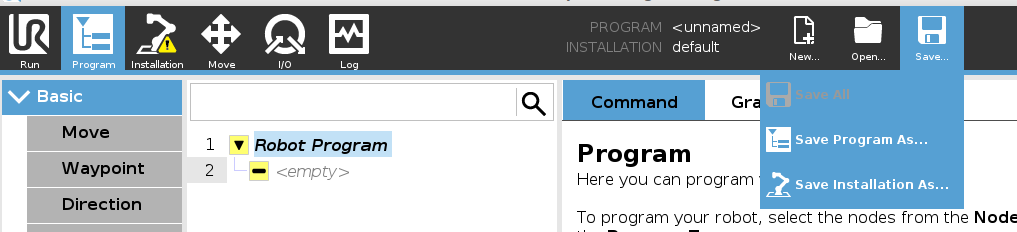
## Programme test

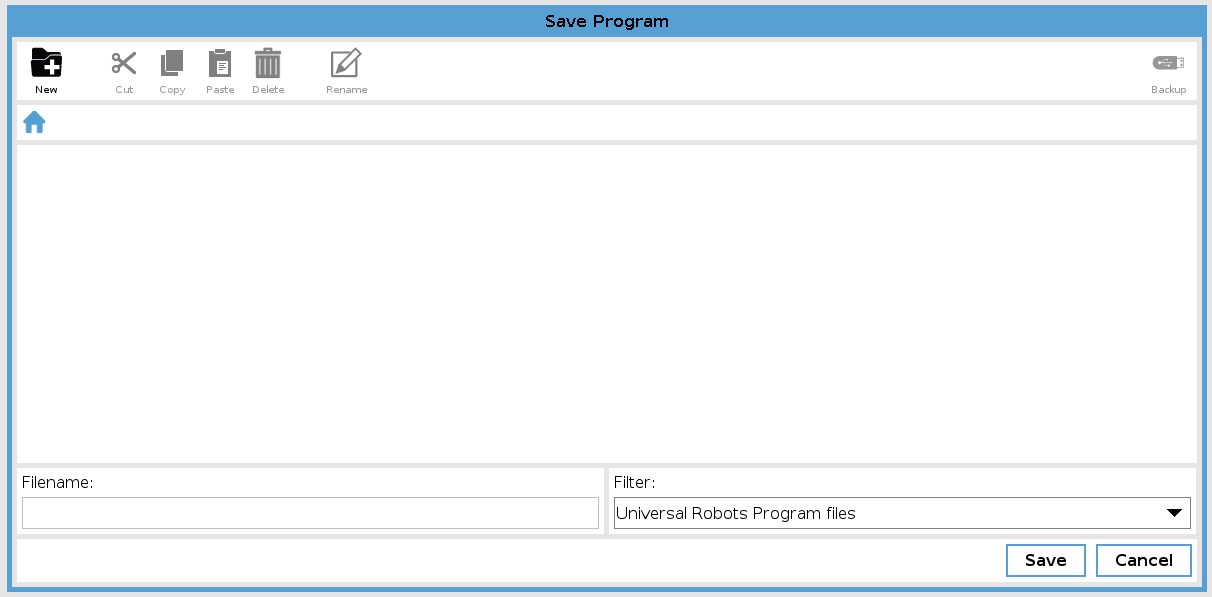
Maintenant qu’on a configuré le plan, on va le tester en créant un programme test pour réaliser une trajectoire rectangulaire sur la boîte.

Pour créer un nouveau code, allez sur l’onglet « Program », où vous allez avoir toutes les fonctions existantes pour programmer le robot, et cliquez sur « New > Program » en haut à droite.

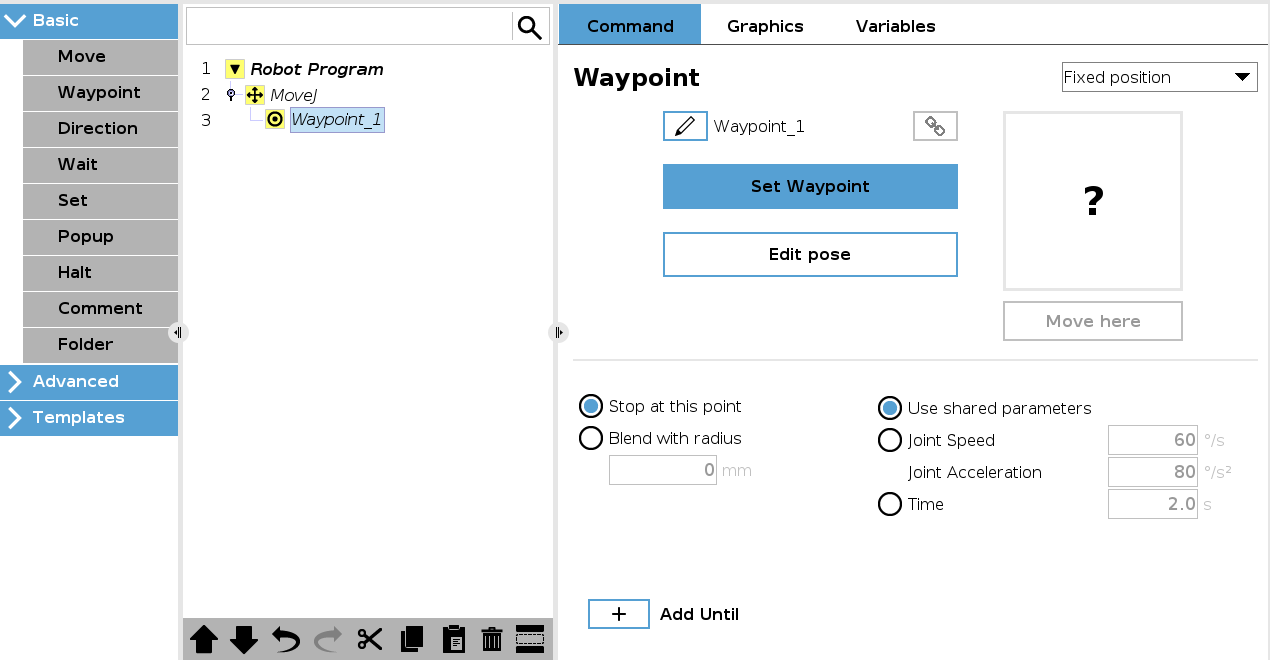


Pour enregistrer votre code, cliquez sur « Save », puis sur « Save Program As… », cliquez sur le champ « Filename » et le nommez comme « Test » et enfin cliquez sur « Save ».



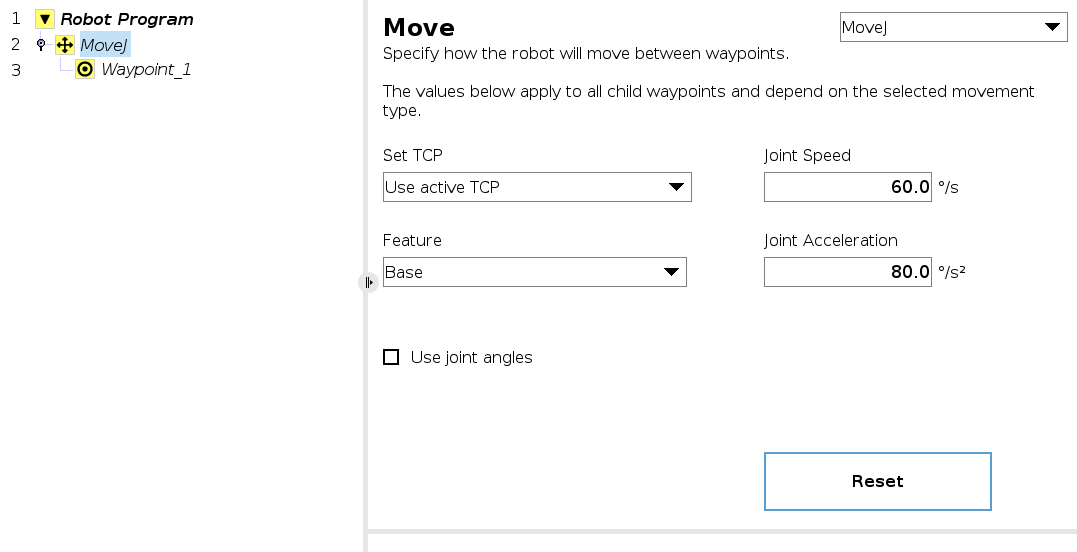


Pour créer une trajectoire rectangulaire, on a besoin de 4 points, pour les ajouter au code, cliquez sur la fonction « Waypoint » dans l’onglet « Basic ».

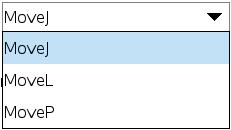


Comme on peut voir dans l’image ci-dessus, un nouveau point « Waypoint\_1 » a été créé, mais il n’a pas encore été appris, ce qui est indiqué par la couleur jaune à côté de son nom et le point d’interrogation.

En plus, une ligne « MoveJ » a été ajoutée au programme, cette ligne nous permet de définir le type de mouvement que le robot va réaliser et d’autres paramètres, comme la vitesse, l’accélération, le TCP et le plan de référence. Cliquez sur « MoveJ » :



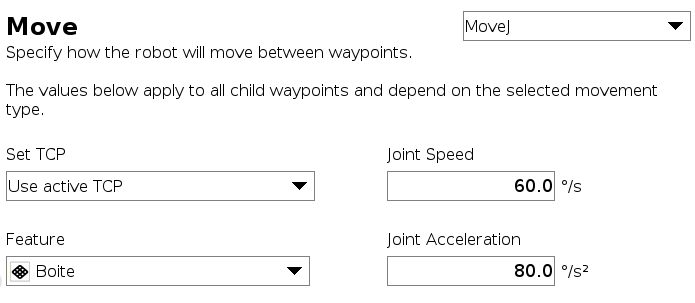
Les types disponibles de mouvement pour le robot UR sont :



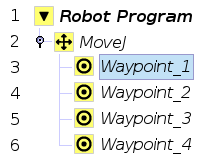
Où :

* MoveJ : crée des déplacements qui sont calculés dans l’espace articulation du bras du  
  robot.
* MoveL : déplace le Point central de l’outil (TCP) linéairement entre les points de  
  passage
* MoveP : déplace l'outil linéairement à vitesse constante avec lissages circulaires

Laissez MoveJ sélectionné. Après, dans le menu déroulant « Feature », sélectionnez « Boîte », cela fera que le mouvement soit relatif au plan Boîte. Laissez le reste des paramètres par défaut.

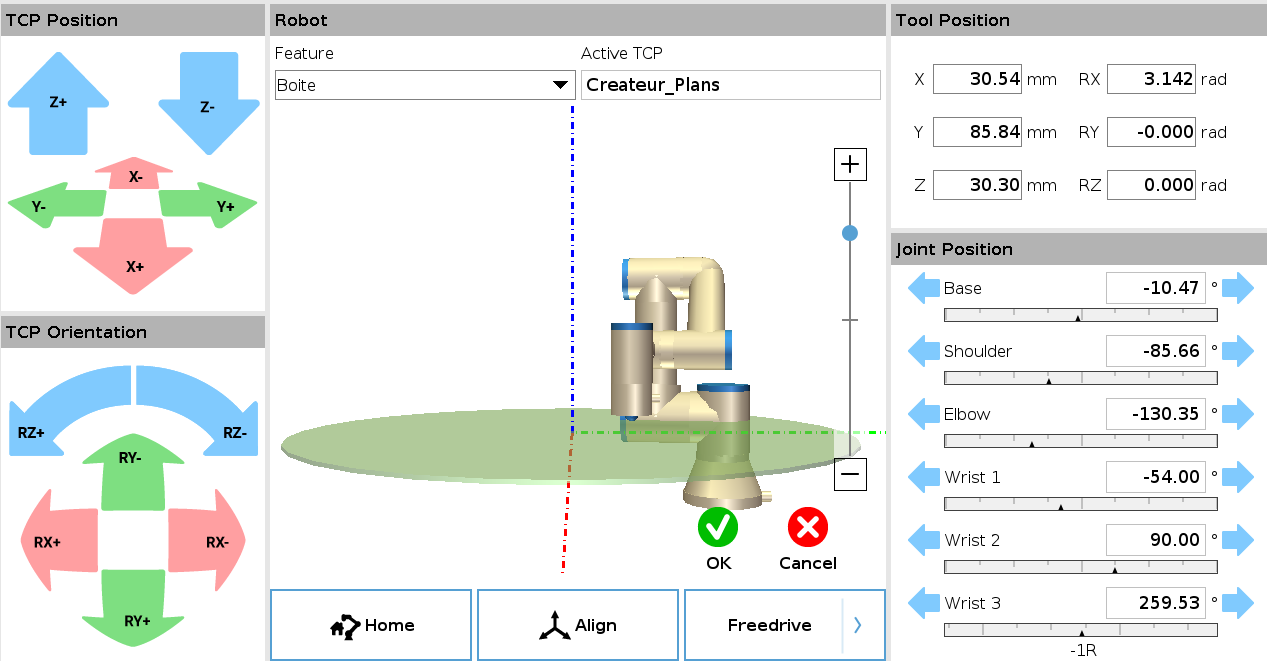


Revenez sur « Waypoint\_1 » et appuyez sur la fonction « Waypoint » pour créer les autres 3 points du rectangle, vous auriez donc cette structure de code :

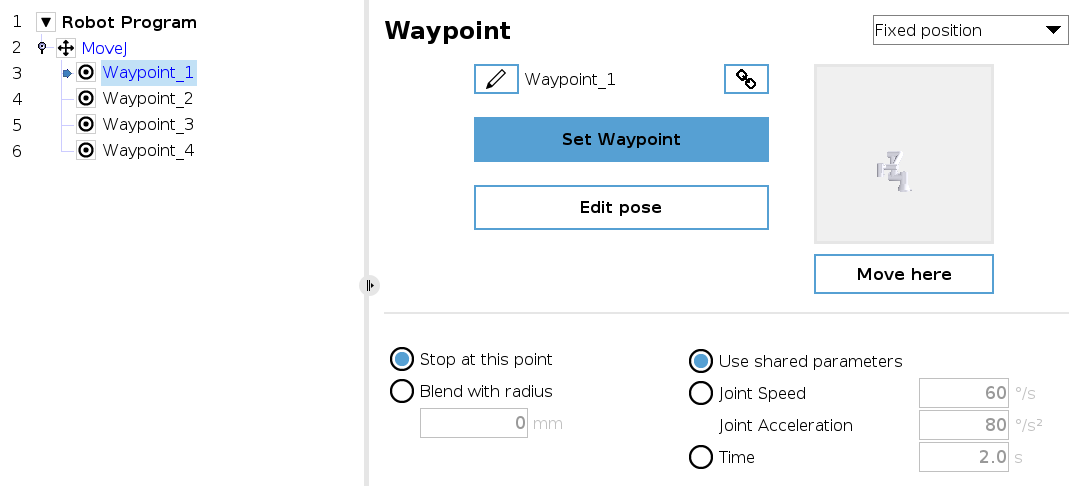


Pour apprendre les points, cliquez sur « Set Waypoint », ce qui vas vous amener vers l’onglet « Move », présentée dans la section 3, positionnez chacun des 4 points dans les coordonnées indiquées dans le tableau ci-dessous et faites « OK » pour enregistrer le point:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | X | Y | Z | RX | RY | RZ |
| 1 | 30 | 30 | 30 | 180° | 0° | 0° |
| 2 | 90 | 30 | 30 | 180° | 0° | 0° |
| 3 | 90 | 90 | 30 | 180° | 0° | 0° |
| 4 | 30 | 90 | 30 | 180° | 0° | 0° |



Les points seront alors appris et vous pouvez voir que la couleur de l’icône à côté du point est devenue blanche et à la place de l’interrogation on a une représentation 3D du robot dans la position choisie.



Lancez le code pour le tester, le robot doit réaliser une trajectoire rectangulaire relative à l’origine de la boîte.

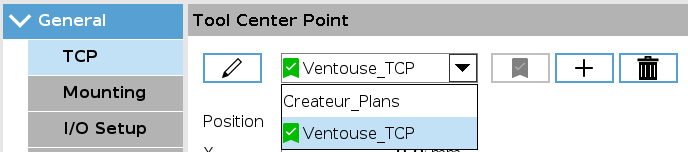
Essayez de changer le type de mouvement (« MoveL » et « MoveP ») et regarder les différences.

# Programmation du robot

Après avoir configuré les plans et l’outil, on peut passer à la programmation du code de palettisation. D’abord, enlevez l’embout (A) et installez la ventouse (B).

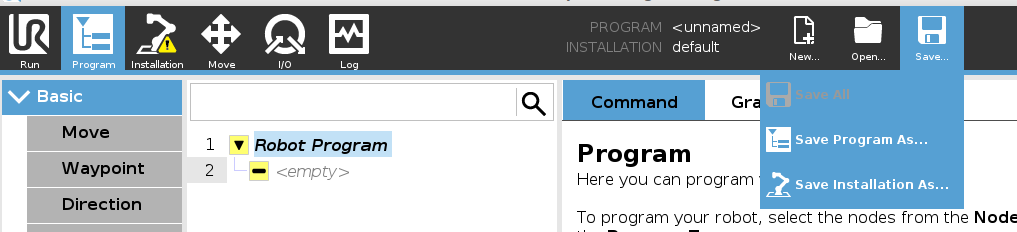
Image ventouse (B)

Puis, dans « Installation > General > TCP», activez le TCP « Ventouse\_TCP ».



Dans ce TP, on apprendra au robot les positions en utilisant l’apprentissage de points, on va aussi se servir du modèle de palettisation inclus dans Polyscope et le capteur d’effort pour détecter les pièces (chocolats).

Allez sur l’onglet « Program » et cliquez sur « New > Program » pour créer un nouveau programme. Puis, enregistrez votre code en cliquant sur « Save > Save Program As… » et choisissez « TP1\_Groupe\_X » comme nom pour votre programme.

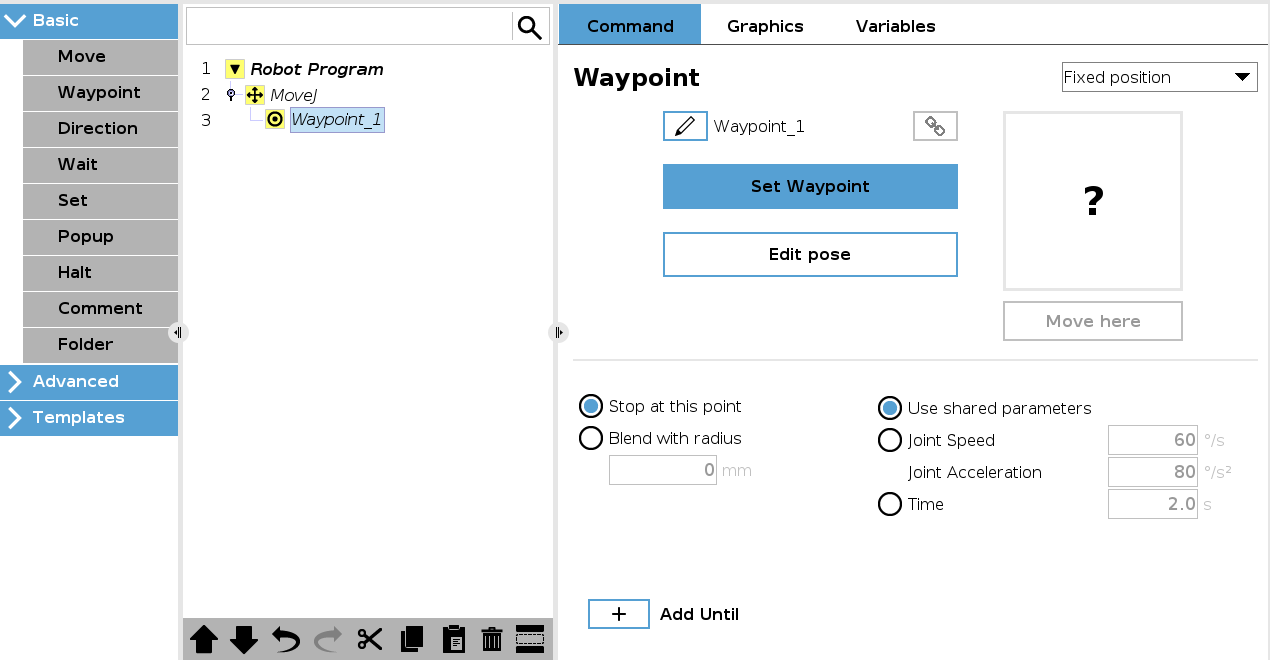


Dans notre cas, on va diviser le code en 3 parties :

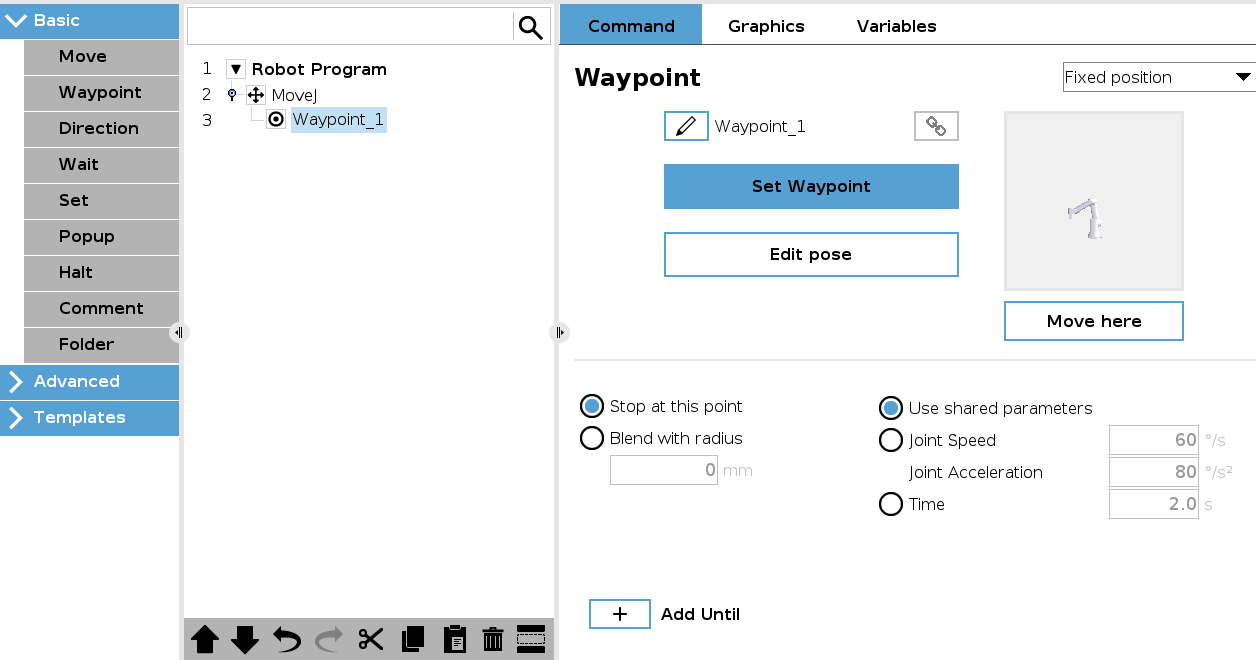
## Définition position initiale

D’abord, on a besoin de positionner le robot dans une position où il sera au début du programme. Vous êtes libre pour choisir la position que vous vas mieux, mais choisissez une position assez proche de la pile et de la boîte et perpendiculaire à ses plans.

Pour apprendre cette position au robot, cliquez sur la fonction « Waypoint » dans l’onglet « Basic ».



Comme on peut voir dans l’image ci-dessus, un nouveau point « Waypoint\_1 » a été créé, mais il n’a pas encore été appris, ce qui est indiqué par la couleur jaune à côté de son nom et le point d’interrogation. Pour apprendre le point, cliquez sur « Set Waypoint », ce qui vas vous amener vers l’onglet « Move ». Positionnez le robot à la position de votre choix puis appuyez sur « Ok », le point sera alors appris et vous pouvez voir que la couleur de l’icône à côté du point est devenue blanche et à la place de l’interrogation on a une représentation 3D du robot dans la position choisie.

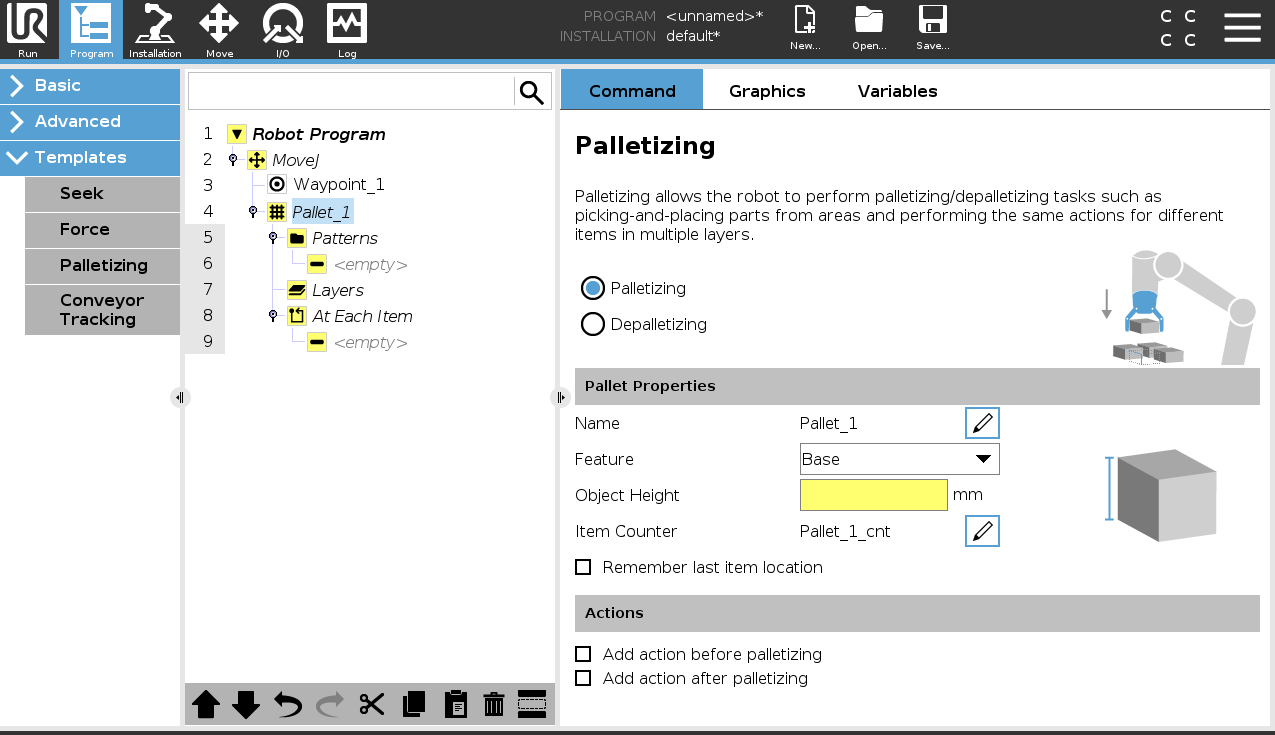


Renommez le point comme « PosInit » en cliquant sur l’icône du stylo à côté de « Waypoint\_1 ».

## Création palette « Pile »

Les chocolats qui seront utilisés pour remplir la boîte sont placés en 3 piles de 3 chocolats chaque et positionnées une à côté de l’autre en ligne. Pour que le robot sache où sont les piles et comment récupérer les chocolats on a besoin d’ajouter dans le code une fonctionnalité qui s’appelle « Palletizing », un modèle déjà inclus dans Polyscope qui nous aide à définir des grilles d’objets pour faire du stockage et/ou déstockage.

Pour créer une palette, allez sur l’onglet « Templates > Palletizing », un bloc appelé « Pallet\_1 » sera ajouté au code comme suite :

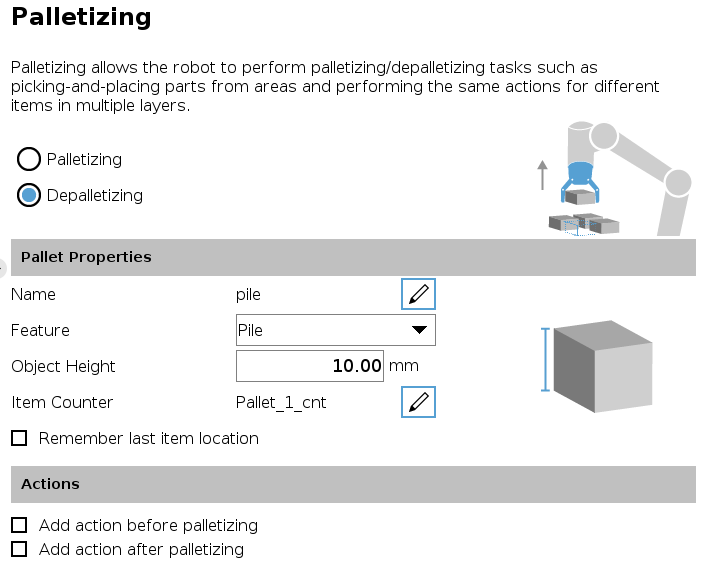


### Configuration palette « Pile »

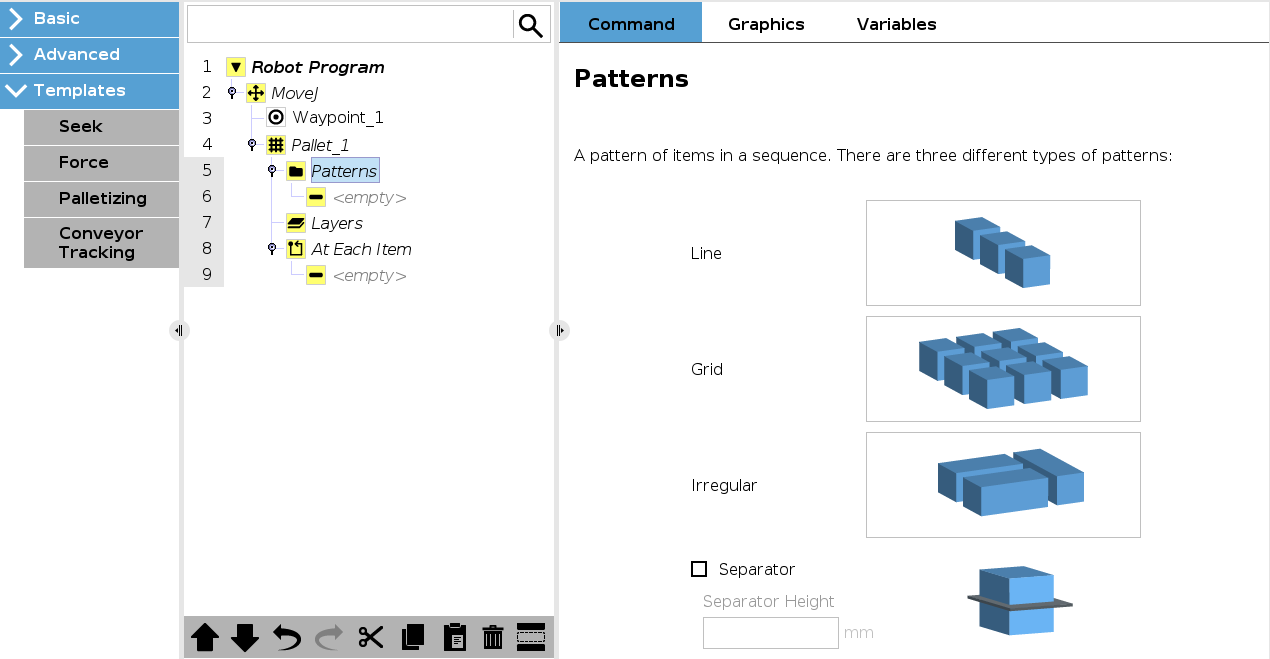
On a quelques configurations à faire pour définir notre palette. D’abord, choisissez l’option « Depalletizing », puisque on va enlever des pièces de la pile.

Dans « Pallet Properties », sur « Name », renommez la palette comme « Pile ». Dans « Feature », sélectionnez l’option « Pile » dans la liste, cela va faire que tous les points qu’on va créer pour cette palette seront définis en relation à l’origine du plan « Pile » qu’on a créé dans la section 3, alors si jamais on bouge la pile, il faudra juste déplacer l’origine du plan, et donc on ne va pas avoir besoin de redéfinir toute la palette.

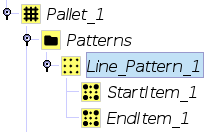
Dans « Object Height », entrez 10 mm, c’est la hauteur de nos chocolats, laissez le reste des options comme par défaut. On doit avoir cela :



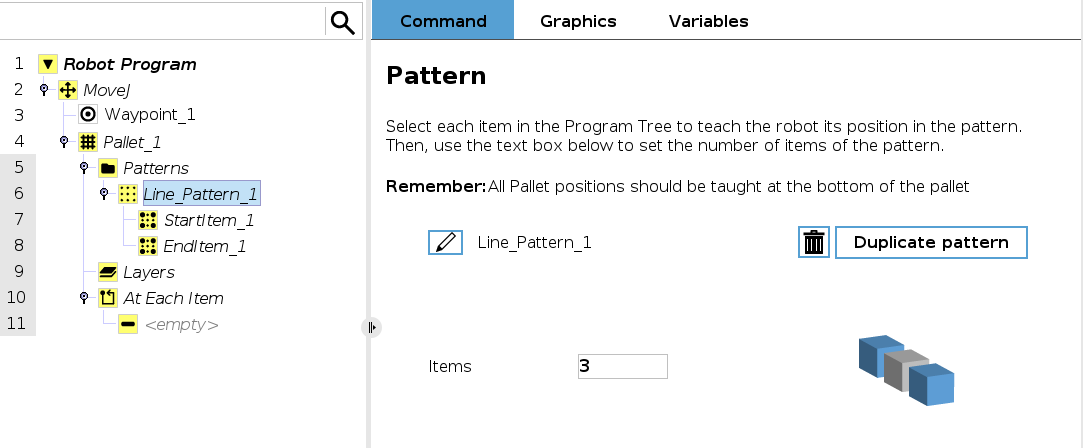
Après, dans le programme du robot, cliquez sur « Patterns », une nouvelle fenêtre de commande sera ouverte à droite avec les modèles disponibles de palette (voir image ci-dessous).



Choisissez l’option « Line », cela va ajouter dans le dossier « Patterns » un nouveau bloc de code « Line\_Pattern\_1 ».

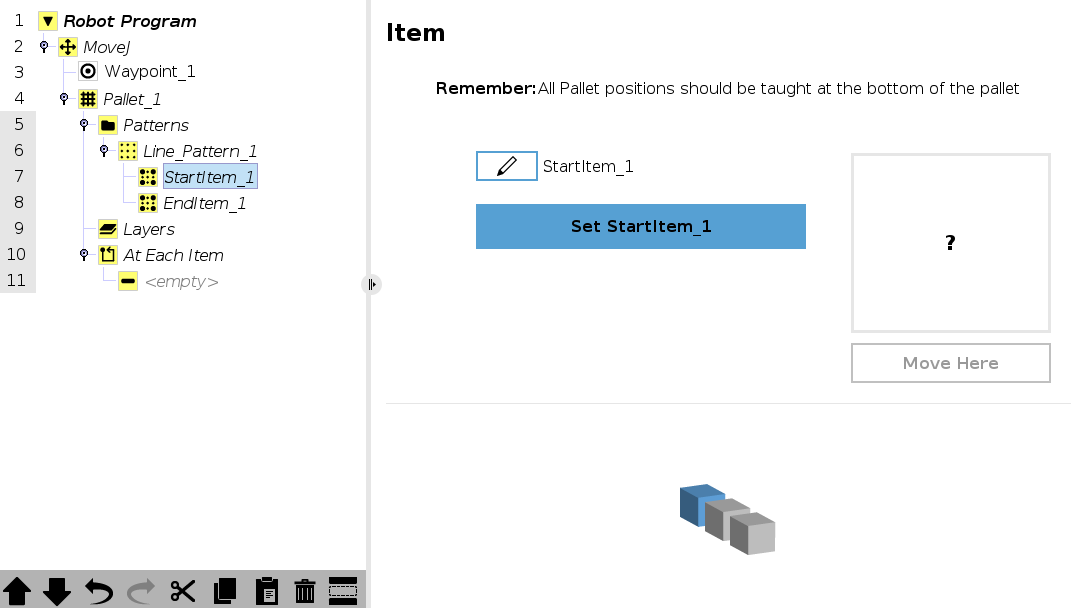


Cliquez sur « Line\_Pattern\_1 », Polyscope vous demandera de définir la quantité de items dans la ligne, entrez « 3 » comme valeur, car on a 3 piles côte à côte.



Maintenant on doit apprendre au robot la position de chaque pile dans la ligne, Polyscope vous demandera de définir deux points, appelés « StartItem\_1 » et « EndItem\_1 », qui sont les points du début et de la fin de la ligne, les points entre ces deux seront automatiquement calculés par Polyscope, avec la même distance entre eux, par exemple, si vous définissez la quantité de items comme 5 et positionnez l’item initial et l’item final écartés par 100 mm, Polyscope ira positionner les 3 items restants aux positions 25mm, 50mm et 75mm, respectivement.

Alors, pour enregistrer le point initial, cliquez sur « StartItem\_1 » dans le bloc de code, puis sur « Set StartItem\_1 ».



Vous allez être redirigé vers l’onglet « Move », vérifiez d’abord si le plan « Pile » est sélectionnez dans le menu déroulant « Features » et placez l’outil sur le centre de la première pile, avec Z la hauteur où la ventouse touche le premier chocolat.

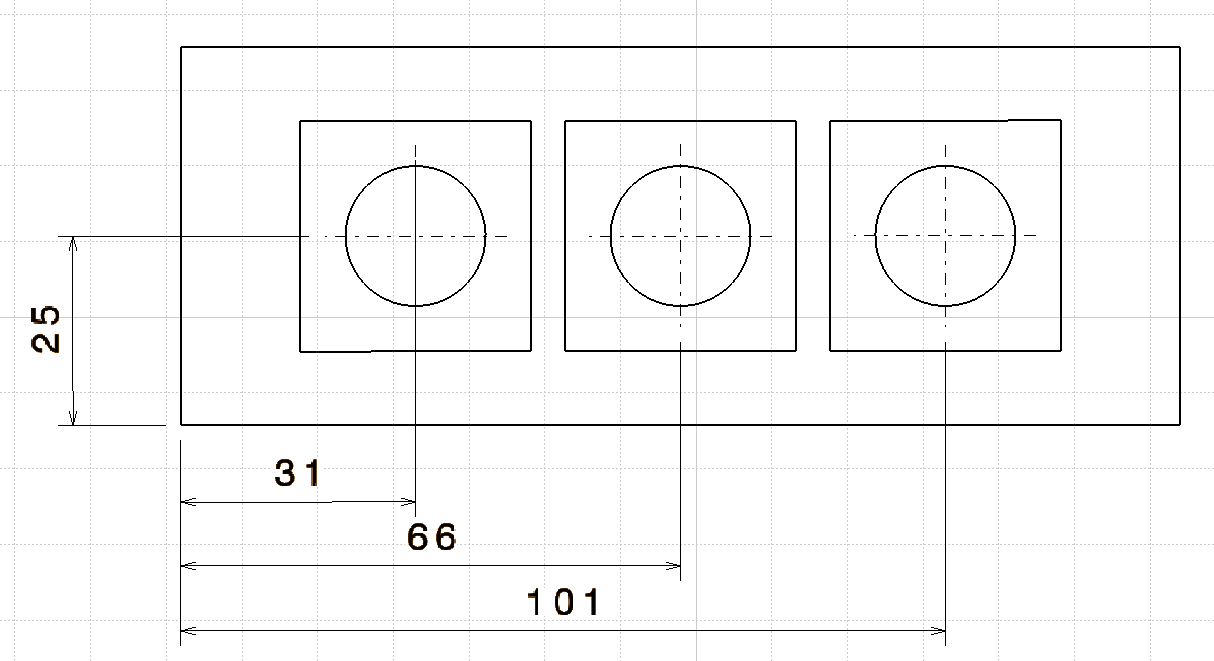


Image robot sur pile

Enfin, placez la ventouse de façon qu’elle touche le centre de la pièce de la première pile et faites « Ok ».

Utilisez plutôt les coordonnées par rapport au plan « Pile »

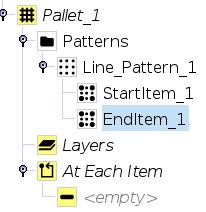
Image

Faites la même procédure pour « EndItem\_1 » sauf que maintenant vous devez positionner la ventouse de façon qu’elle touche le centre de la pièce de la troisième pile.

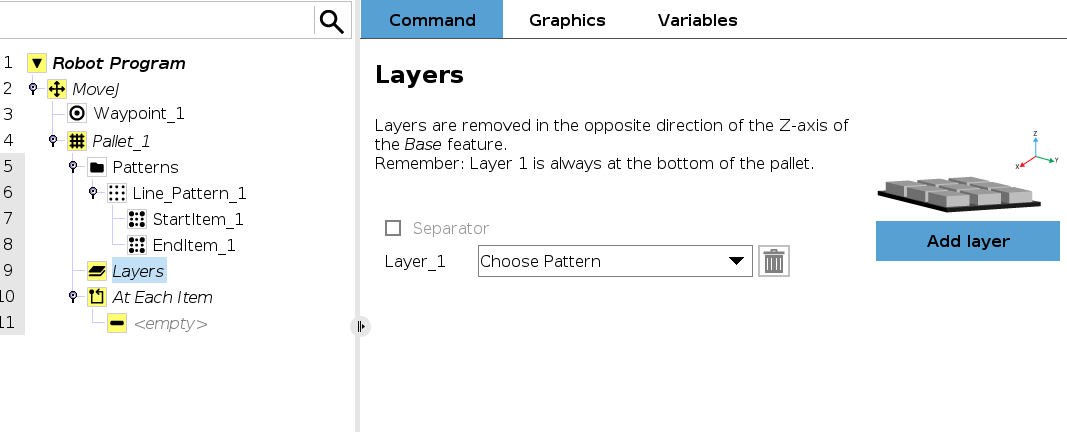
Répétez la procédure pour « EndItem\_1 », sauf que maintenant vous devez positionner la ventouse sur le centre de la troisième pile.

Image robot sur troisième pile

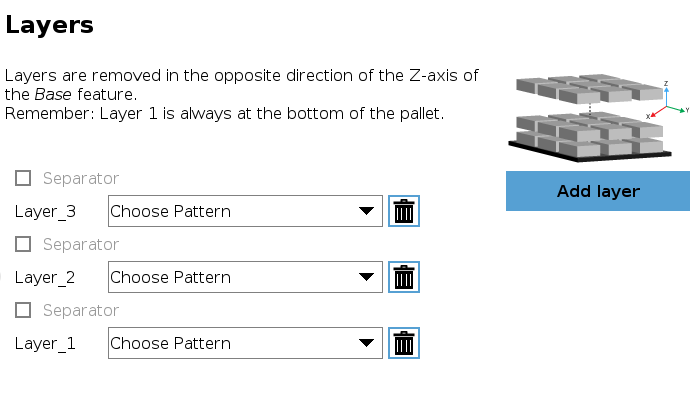
Après finir cette étape, les deux points et le bloc « Pattern » passeront à la couleur blanche pour dire qu’ils ont été appris.



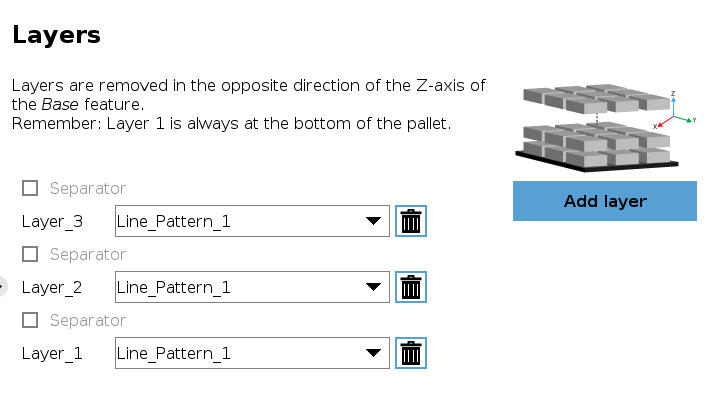
On doit maintenant programmer le nombre de couches existantes dans les piles. Comme on a 3 pièces par pile, ce nombre est 3, pour entrez cette valeur dans le code, cliquez sur « Layers », la fenêtre suivante s’ouvrira :



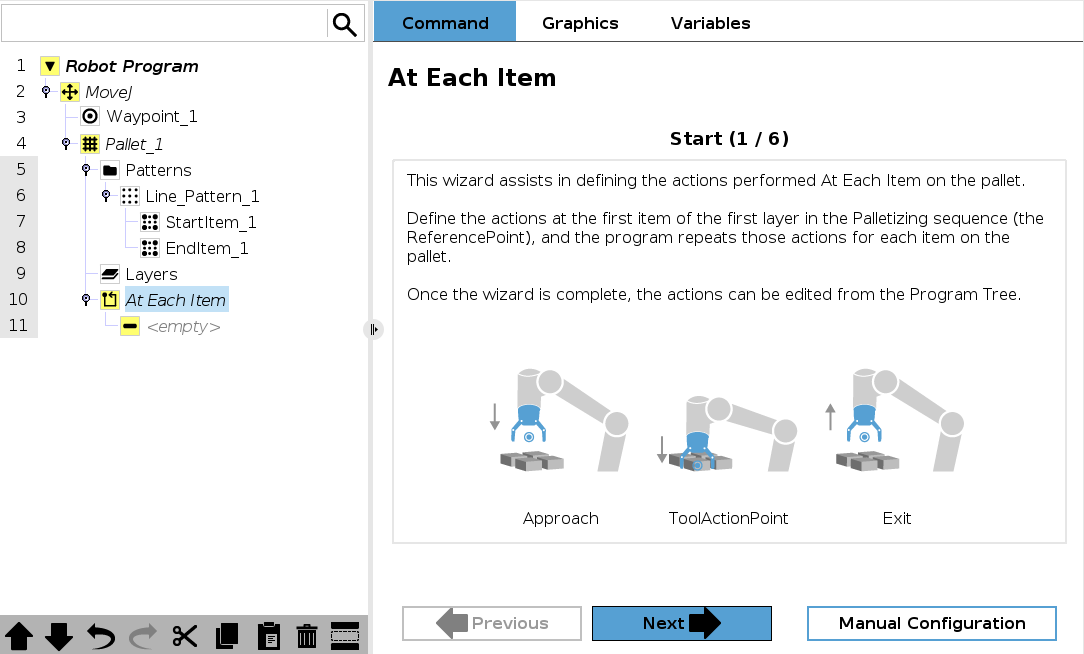
Par défaut, seulement une couche est créée, pour ajouter les deux autres, appuyer sur le bouton « Add Layer », les couches 2 et 3 seront ajoutées, dont l’ordre des couches de bas en haut est 1, 2 et 3.



Maintenant, sélectionnez « Line\_Pattern\_1 » dans les listes des couches, cela signifie qu’on a 3 couches avec une ligne de 3 pièces chaque. Voici ce que vous devez avoir :



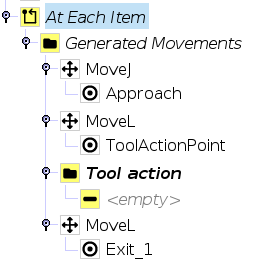
Enfin, une fois qu’on a configuré les couches, il faut aussi configurer les points d’approche et de sortie pour les pièces. Dans le bloc de code, cliquez sur « At Each Item », la fenêtre suivante s’affichera :



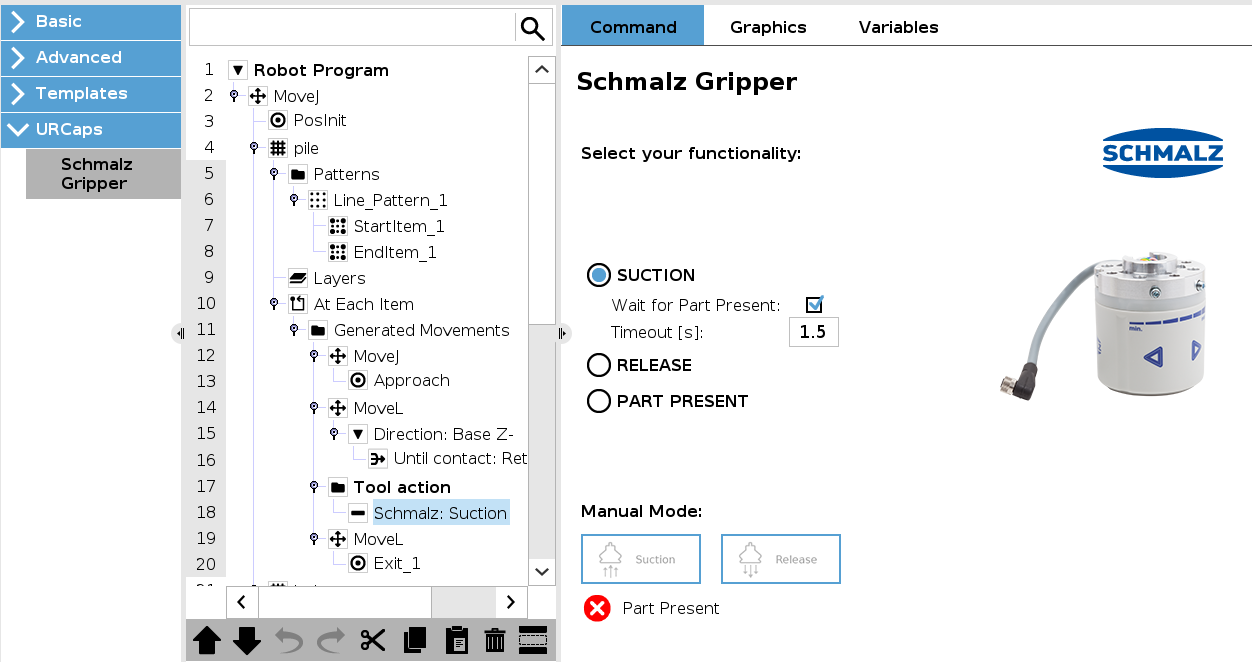
Cliquez sur « Next » et suivez la procédure demandée par Polyscope pour créer les 2 points nécessaires. Pour avoir bons points d’approche et de sortie, écartez la ventouse de la pièce en Z de quelques mm (30 mm suffit) comme montré dans l’image ci-dessous. Vous n’êtes pas obligé de mettre le même écart de l’image, mais choisissez quand-même une distance assez proche de la pièce et qui soit suffisamment distante de possibles obstacles pour éviter que le robot les heurte lors de son mouvement.

Image

Un nouveau bloc de code appelé « Generated Movements » sera ajouté dans « At Each Item » :



Il nous reste maintenant définir l’action de l’outil lorsqu’elle touche la pièce. Comme on utilise une ventouse, on doit activer la succion pour qu’elle attrape le chocolat. Pour ajouter la fonction de succion dans le code, allez sur l’onglet « URCaps », puis sélectionnez la fonction « Schmalz Gripper » et dans la fenêtre de commande de l’outil, cochez la case « Suction » et laissez-le reste par défaut.

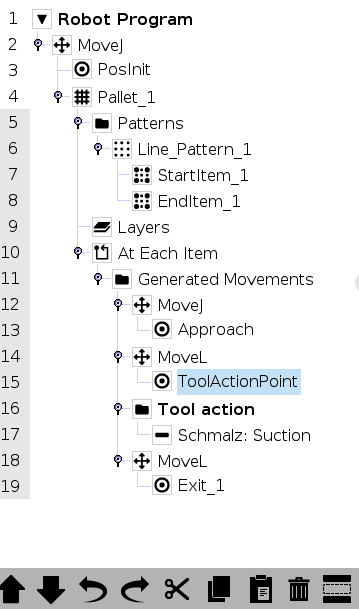
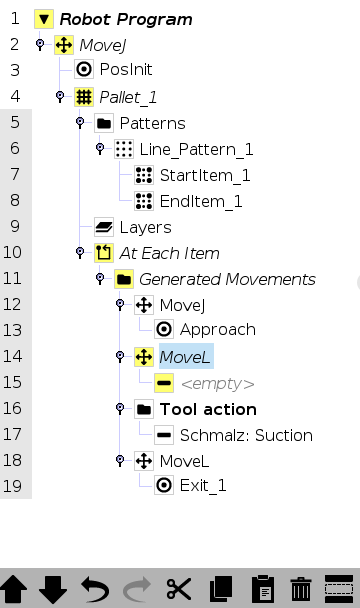


Vous pouvez tester la Succion ou le Relâche avec les boutons du Mode Manual si vous voulez.

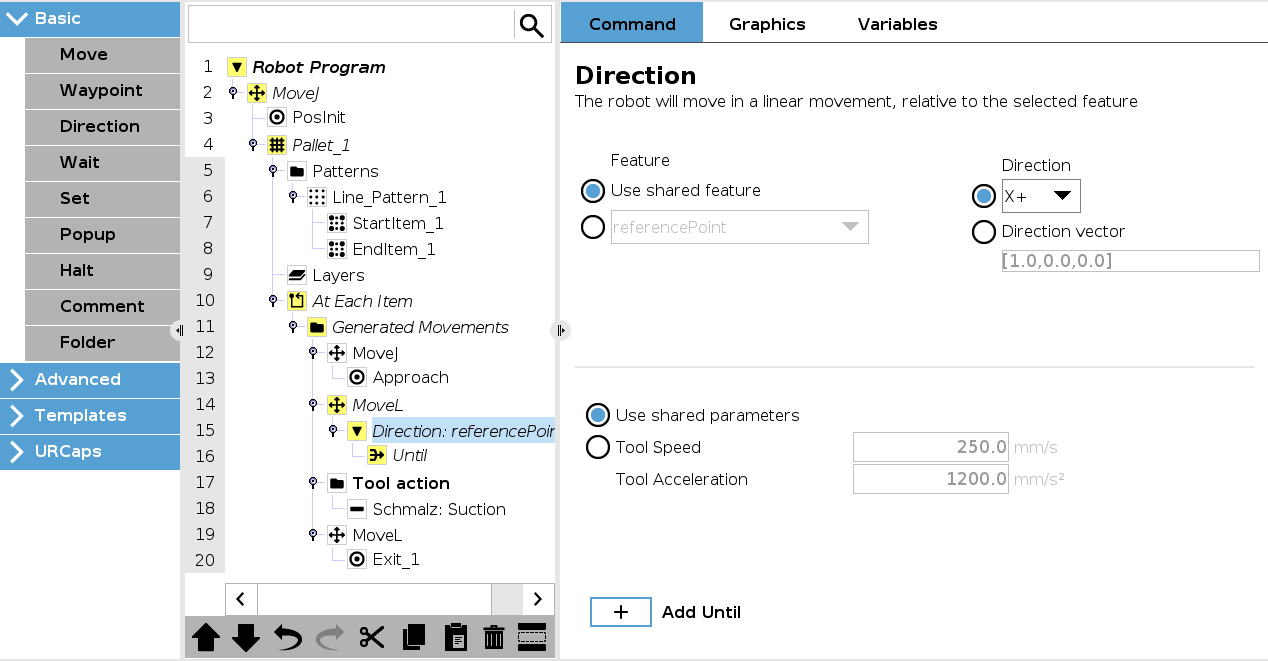
Enfin, tout le bloc de code passera à la couleur blanche pour dire que la palettisation a été enregistrée, par contre, on va faire une modification dans ce code pour ajouter la fonctionnalité du capteur d’effort du robot, qui est plus précise que le « ToolActionPoint » et très utile quand on veut savoir quand le robot a touché un objet.

### Capteur d’effort

Supprimez le point « ToolActionPoint » en cliquant dessus, puis sur l’icône de la poubelle dans le menu en bas.

Pour détecter quand la ventouse touche le chocolat, on doit utiliser la fonction « Direction » disponible dans l’onglet « Basic » de la liste de fonctions. Cliquez sur « empty » dans l’endroit où était le point supprimé, puis cliquez sur « Direction », le bloc de code « Direction » sera ajouté :



(3)

(2)

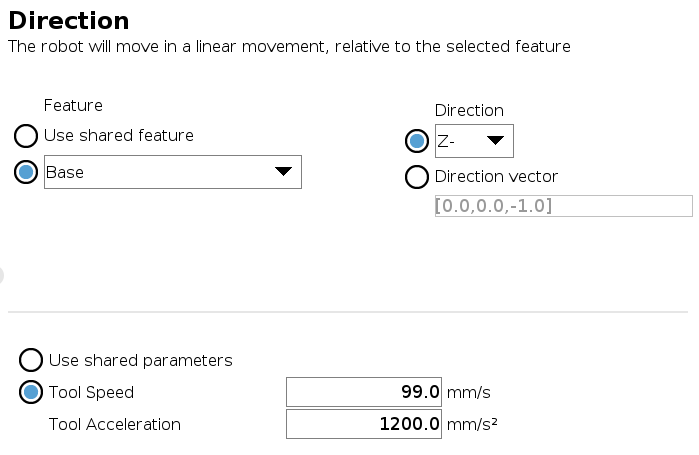
(1)

La fonction « Direction » nous permet de déplacer le robot le long d’un axe choisi. Dans notre cas on veut que le robot descend sur l’axe Z jusqu’à ce qu’il touche la pièce, alors, pour configurer cela, on a besoin de changer les valeurs dans les 3 champs (1), (2) et (3).

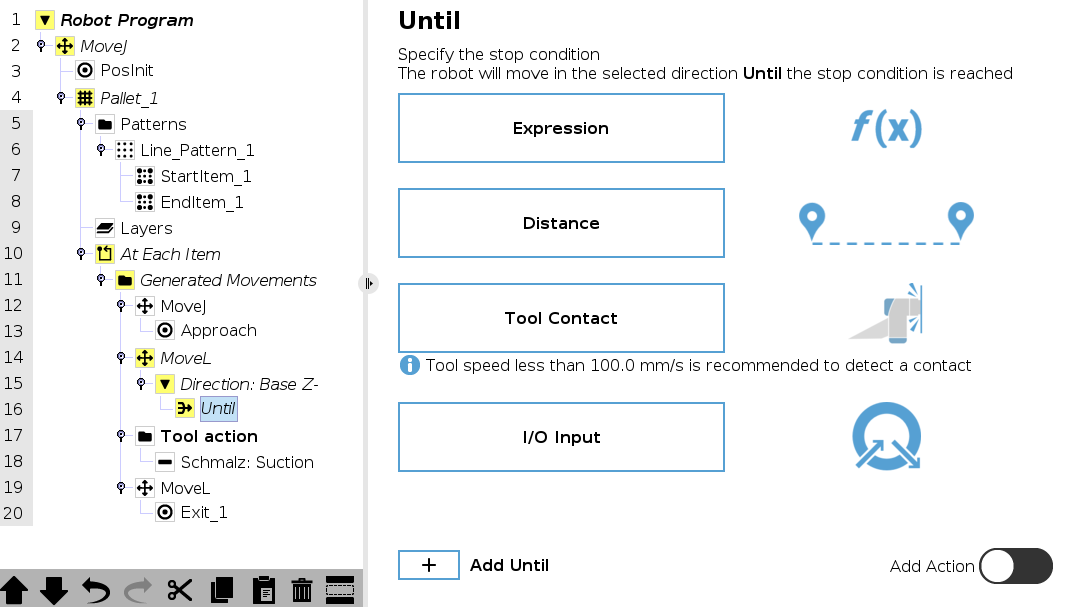
En (1), cochez la deuxième case et sélectionnez « Base » dans la liste, cela veut dire que l’outil se déplacera par rapport au plan « Base ».

En (2), cochez la première case et sélectionnez « Z- » dans la liste, pour que le robot se déplace dans la direction négative de l’axe Z.

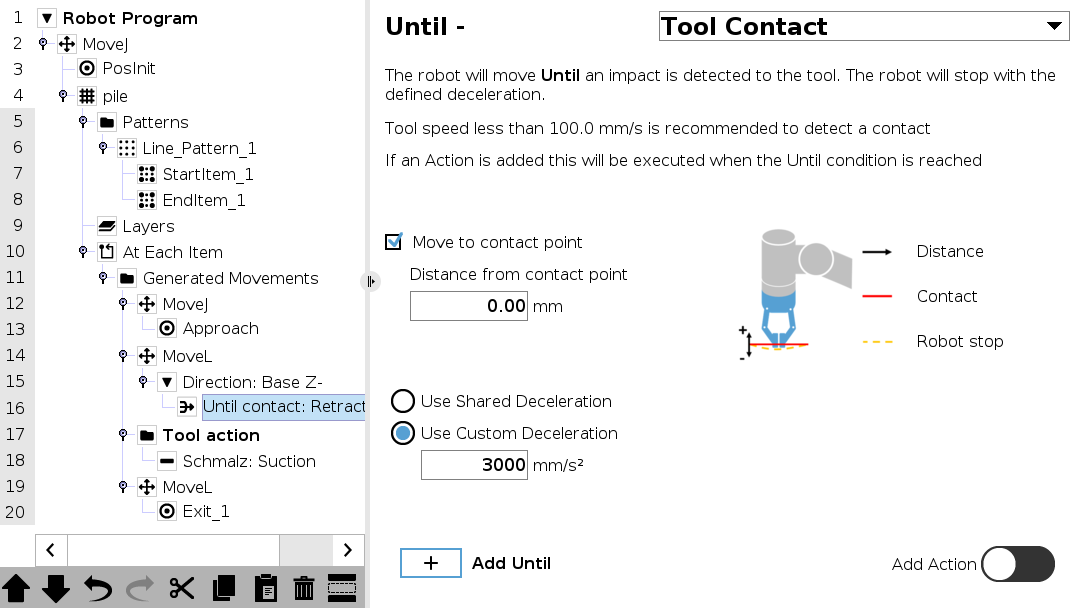
En (3), cochez le deuxième cercle et entrez 99 mm/s comme vitesse de l’outil, qui est la vitesse maximale autorisée pour que le capteur d’effort puisse détecter le contact avec un obstacle.



Maintenant, sélectionnez « Until » dans le bloc de code, Polyscope vous demandera de choisir une condition pour que le robot s’arrête. Appuyez le bouton « Tool Contact » pour que le robot s’arrête dès qu’il touchera le chocolat. Vous voyez qu’au-dessous du bouton il y a une observation par rapport à la vitesse maximale de l’outil (moins que 100 mm/s).



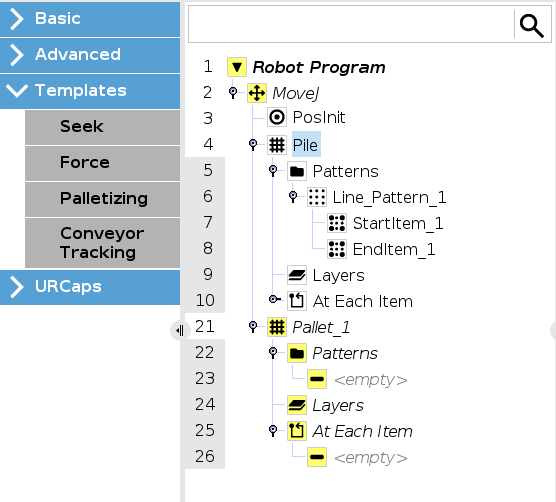
Une nouvelle fenêtre apparaîtra avec des informations sur l’option choisie, comme la distance par rapport au point de contact et la décélération de l’outil. Laissez ces paramètres par défaut.



Finalement, on a fini le codage pour la palette « Pile », il nous reste maintenant programmer le remplissage de la boîte.

## Création palette « Boîte »

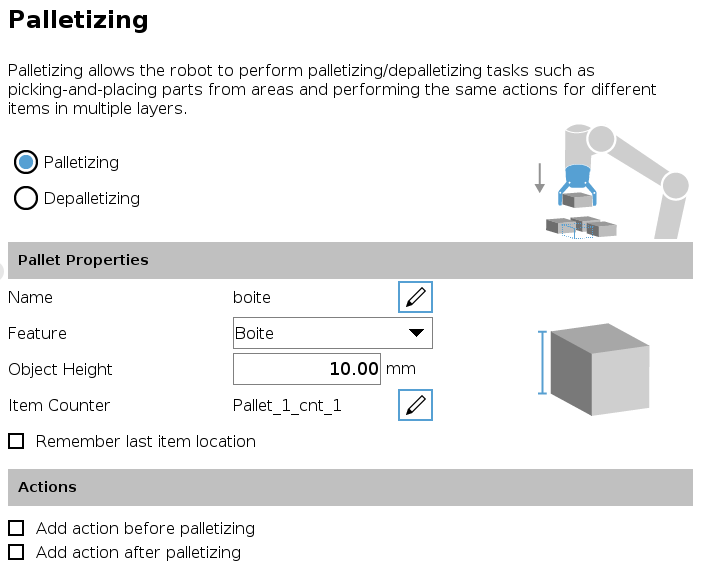
De la même manière qu’on a fait pour la section précédente, on va programmer la palette « Boîte » pour réaliser le remplissage de la boîte avec les chocolats pris de la pile. D’abord, pour créer un nouveau bloc de code de palettisation après celui de la pile, cliquez sur la line « Pile » et ajoutez le template « Palletizing », cela va créer un nouveau « Pallet\_1 ».



Renommez-le comme « boite ».

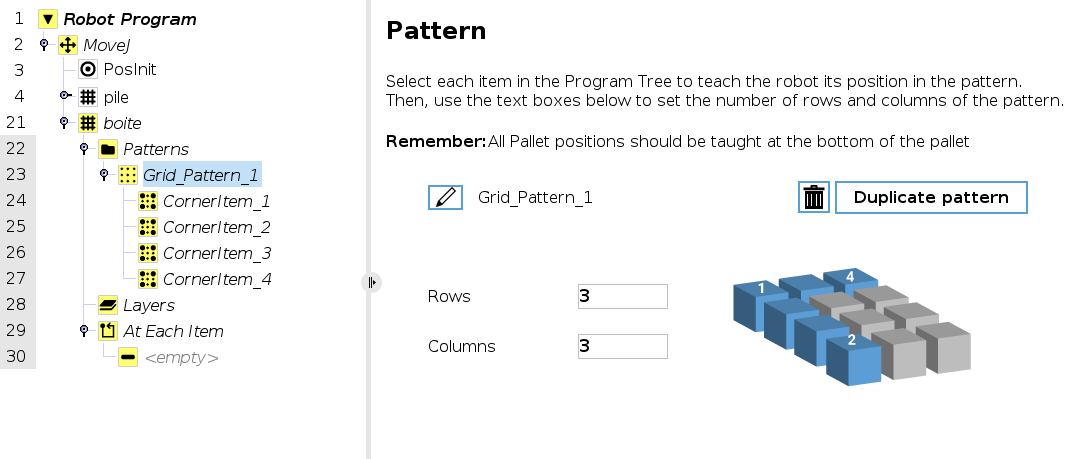
### Configuration palette « Boîte »

Suivez les mêmes procédures qu’on a fait pour la pile, sauf que maintenant on va choisir « Palletizing » comme option et « Boite » comme « Feature » dans « Pallet Properties ».



Dans « Patterns », choisissez « Grid », car la boîte est une matrice de 3x3. Un nouveau bloc de code sera ajouté, où on doit définir la taille de la matrice et apprendre les 4 coins de la matrice au robot.

Tout d’abord, entrez le nombre de lignes et de colonnes comme 3.



Puis, enregistrez les positions « CornerItem\_X » en positionnant l’outil sur le centre de chacune des 4 alvéoles des coins de la boîte (coordonnées montrées ci-dessous). Placez le TCP dans une hauteur assez proche du fond de l’alvéole, pour que le robot lâche le chocolat dans son espace démarqué. Vérifiez que le plan « Boîte » est bien sélectionné dans le menu déroulant « Feature » sur l’onglet « Move » quand vous irez enregistrer les points.

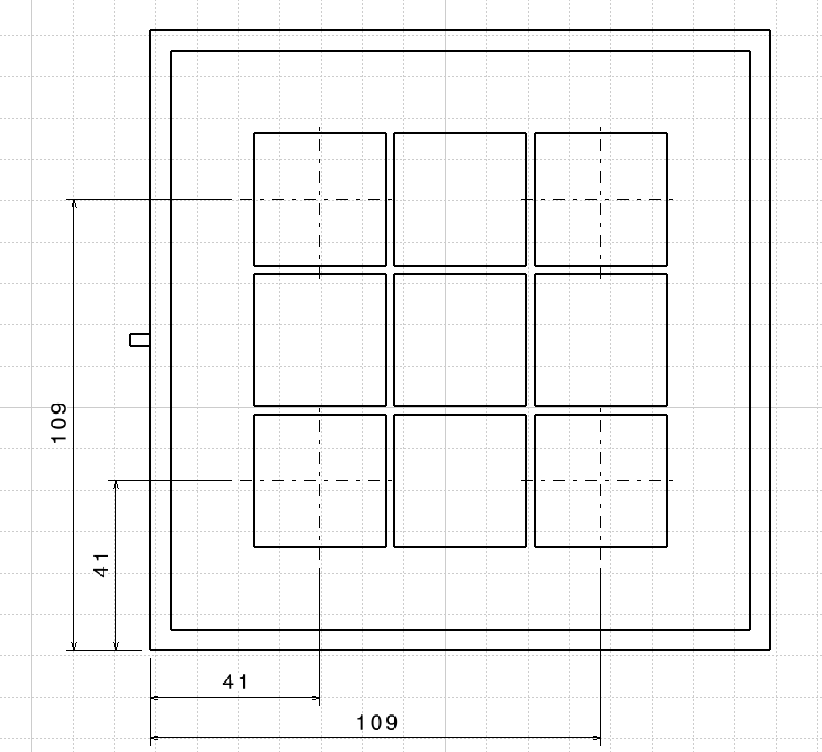
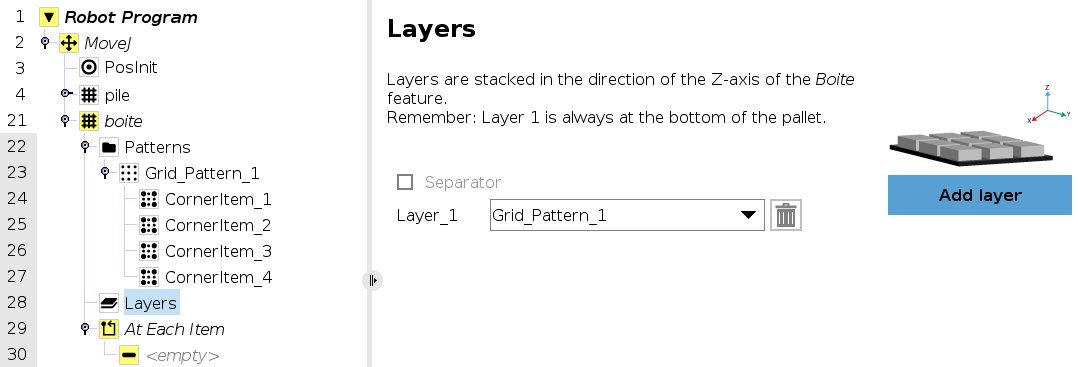
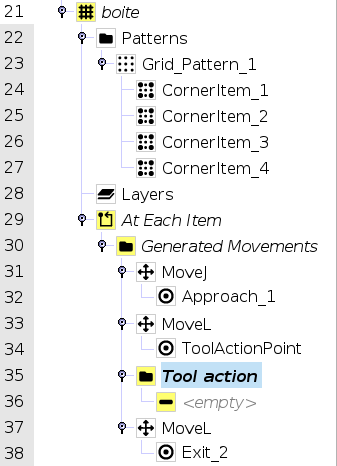


Image Outils sur Centre des Alveoles

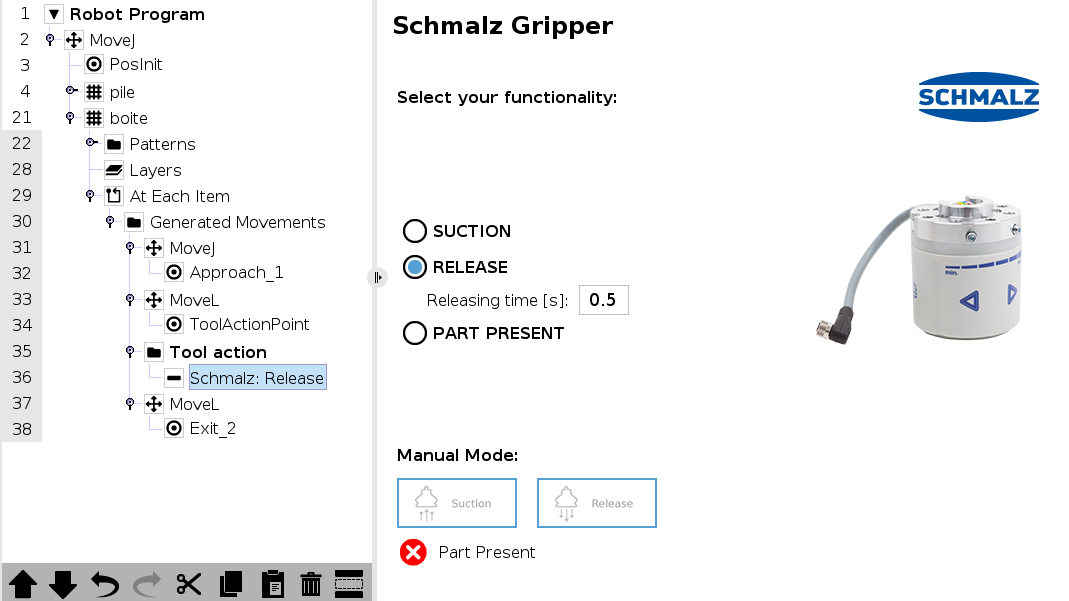
Pour les couches, comme on a qu’un seul chocolat par alvéole, laissez 1 couche et sélectionnez « Grid\_Pattern\_1 » dans la liste.



Pour configurer les actions « At Each Item », faites comme dans la section précédente et suivez les étapes demandées par Polyscope pour créer les points d’approche et de sortie, cela va ajouter le bloc « Generated Mouvements ».

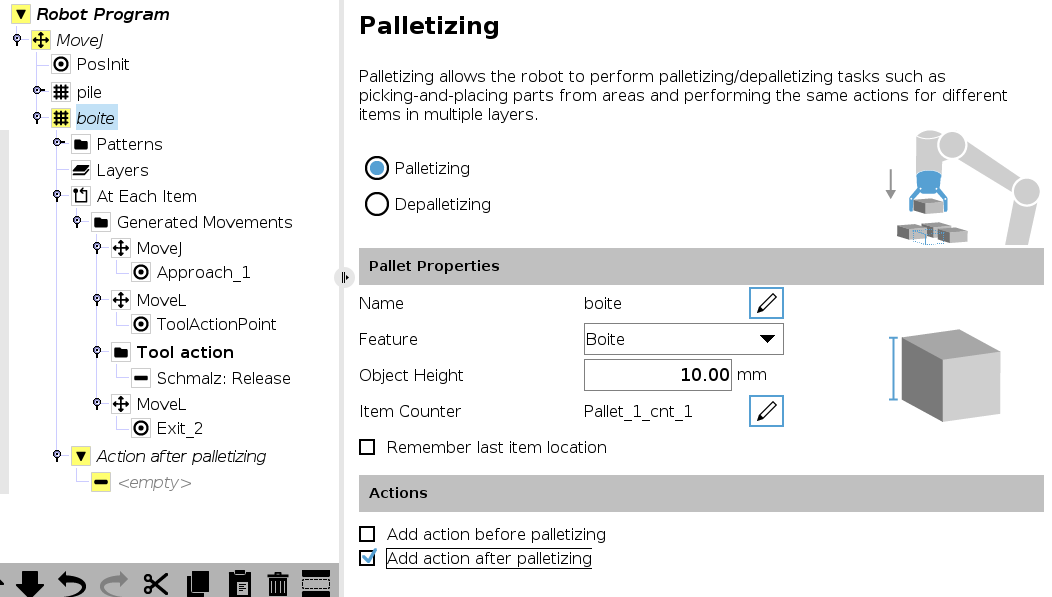


En plus, la différence par rapport à la pile est que maintenant on doit lâcher les pièces dans les alvéoles, pour réaliser cette commande, dans « Tool action », cliquez sur « empty » et ajoutez l’URCap « Schmalz Grippper » puis cochez la case « Release ».

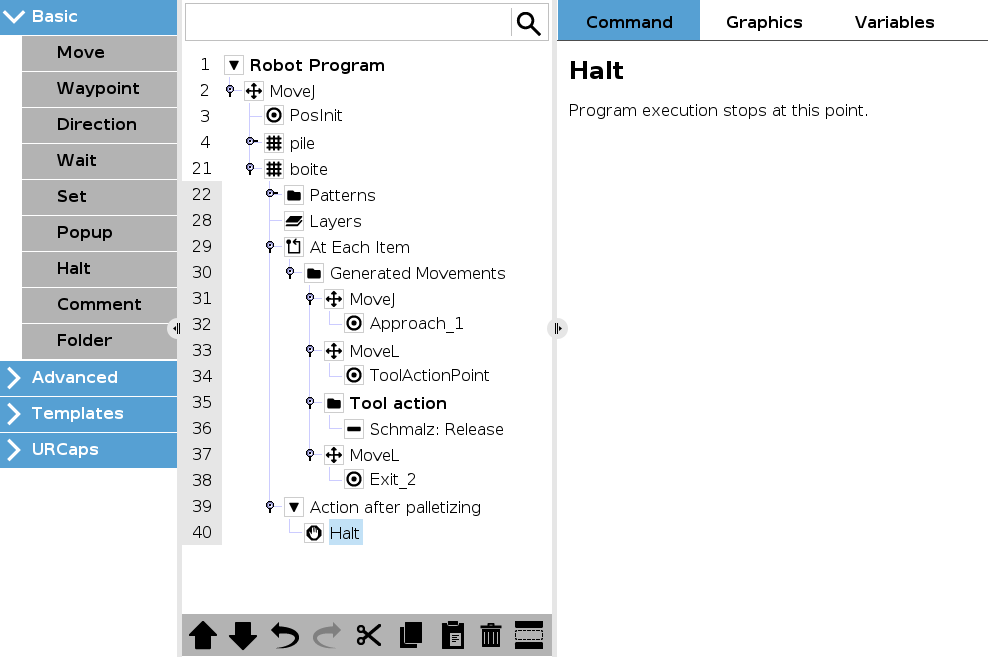


Notre programme est presque fini, mais il reste encore un détail important à analyser. Si on laisse le code comme ça, dès que le robot lâchera le neuvième chocolat dans son alvéole il recommencera le code du tout début et refera toutes les palettisations, même que les piles soient vides, parce que le robot ne sait pas qu’elles le sont, vu qu’on n’a pas un capteur de présence pour lui envoyer cette information et arrêter son programme. Alors, pour résoudre ce problème, le modèle de palettisation a une fonction appelée « Add action after palletizing », qui nous permet de créer des actions qui seront exécutées après que le robot ait fini de traiter le dernier élément de la palettisation.

Pour ajouter cette action, cliquez sur le bloc de code « boite » pour ouvrir la fenêtre de configuration de la palette, puis, en bas, dans « Actions », cochez la case « Add action after palletizing », cela va ajouter deux lignes de code à la fin du bloc.



Dans notre cas, on veut arrêter l’exécution du programme, pour faire cela, on utilisera la fonction « Halt ». Cliquez sur « empty » et dans l’onglet « Basic » de la liste des fonctions à gauche, cliquez sur « Halt ». Cette fonction sert à déclencher l’arrêt du programme.



Enfin, pour cette palette, on n’a pas besoin d’utiliser le capteur d’effort, vu qu’on a déjà saisi les pièces et on n’a pas besoin de détecter le fond de la boîte pour relâcher le chocolat.

Très bien, on a fini le code pour la palettisation, FAITES VERIFIER VOTRE CODE AVEC LE PROFESSEUR avant de le lancer.

# Conclusion

* *On voit que le robot retourne à « PosInit » toujours avant de prendre un chocolat de la pile, pourquoi ? Comment éviter cela ? Décrivez une solution en utilisant les fonctions disponibles sur Polyscope.*
* *Comment optimiser les mouvements du robot ?*
* *On utilise un outil avec une seule ventouse, proposez un nouvel outil pour optimiser le remplissage de la boîte de chocolat.*
* *Changez la boîte et/ou la pile d’endroit et puis repositionnez ces plans correspondants, lancez le programme et analysez s’il marche encore.*
* *Pourquoi la vitesse pour que le capteur d’effort fonctionne ne doit pas dépasser 100mm/s ?*