
Programmation de robots*

1 Le robot

Le but de ces séances d'initiation est de vous apprendre les bases de la programmation du robot en quelques heures. Pour arriver au plus vite au cœur du sujet, nous avons construit les modèles au préalable. Vous disposez ainsi d'un robot 'standard' (voir figure 1) composé des éléments suivants :

- Un *cerveau*, la brique NXT. Elle contient un processeur (comme les ordinateurs) capable d'exécuter des suites d'instructions de base : le *programme*.
- Deux moteurs commandant les roues, connectés aux sorties B et C de la brique NXT.
- Un ou plusieurs capteurs de différents types, selon les exercices.

1.1 La brique NXT

Cette brique comporte 4 entrées (on parle de *ports d'entrée*), numérotées de 1 à 4. Chaque entrée peut recevoir un capteur qui permet au robot de récupérer des informations sur son environnement :

*Ce document peut être téléchargé à l'adresse suivante : <http://www.lifl.fr/~decomite/download/downloadF.html>

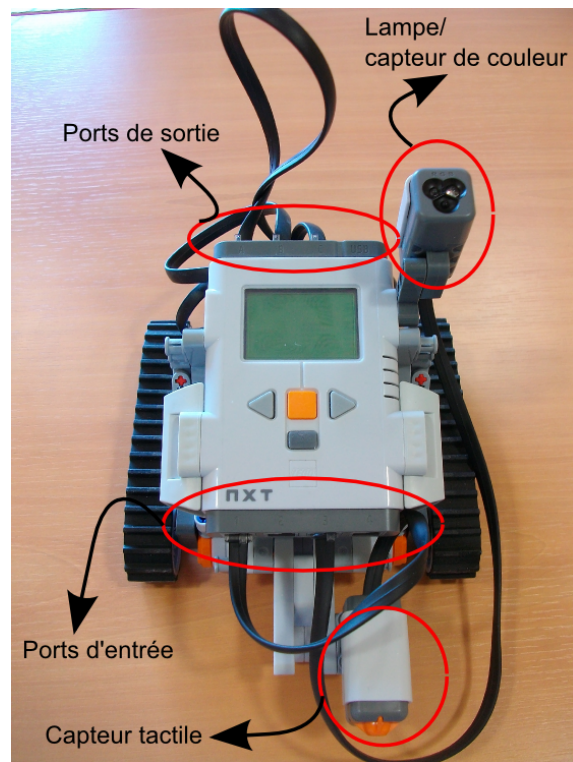


FIGURE 1 – Une vue générale du robot

- Présence d'un obstacle (capteur tactile ou détecteur à ultrasons).
- Couleur d'un objet ou luminosité ambiante (capteur de couleurs).

Il y a aussi, de l'autre côté de la brique, trois sorties, notées A, B et C qui permettent au robot d'agir. Sur le modèle que nous vous avons préparé, ces trois sorties sont connectées aux trois moteurs : les moteurs B et C commandent les roues (ou les chenilles). Le moteur connecté sur A est là pour faire beau.

A côté de la sortie C se trouve l'entrée USB qui permet de connecter le robot au PC. C'est par cet intermédiaire que les programmes, développés sur le PC, seront transmis au robot. On pourra alors déconnecter le robot du PC, et le faire évoluer de façon autonome.

Sur le devant de la brique se trouve un écran qui permet d'une part de naviguer dans différents menus et de contrôler le robot, d'autre part d'afficher différents messages pendant l'exécution d'un programme.

2 L'interface de développement

Lancez le programme, en cliquant sur son icône. Vous obtenez l'écran de la figure 2. Pour avoir une zone de travail plus grande, cliquez sur la croix entourée, en haut à droite de la figure 2.

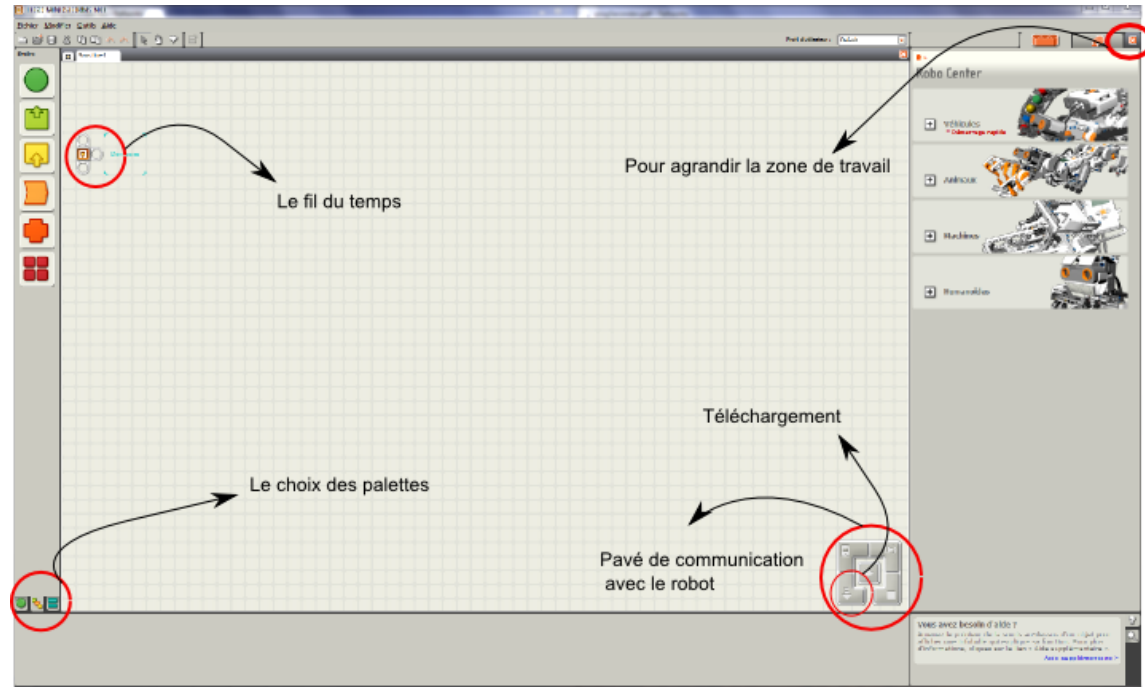


FIGURE 2 – L'environnement de programmation

La programmation du robot se fait en plaçant des blocs de commandes sur une sorte de *fil du temps*. Quand le robot voudra exécuter le programme qu'on y aura enregistré, il effectuera chacune des commandes l'une après l'autre en les lisant de la gauche vers la droite. La figure 3 vous montre le *fil du temps*, obtenu en cliquant-tirant sur le plot originel.

Les blocs de commandes se trouvent dans le menu de gauche. Il y a trois palettes disponibles :

- La palette des commandes courantes (l'onglet représenté par un rond vert en bas à gauche de l'écran).
- La palette complète (l'onglet avec les trois petits carrés de couleurs différentes en bas à gauche).
- Les blocs construits par l'utilisateur ou bien téléchargés (l'onglet représenté par les deux barres bleues en bas à droite).

Nous nous servons de la palette complète. Celle-ci comprend six boutons regroupant les commandes de même type (voir figure 4).

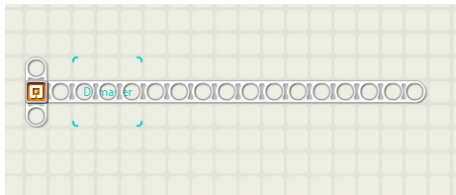


FIGURE 3 – Le fil du temps

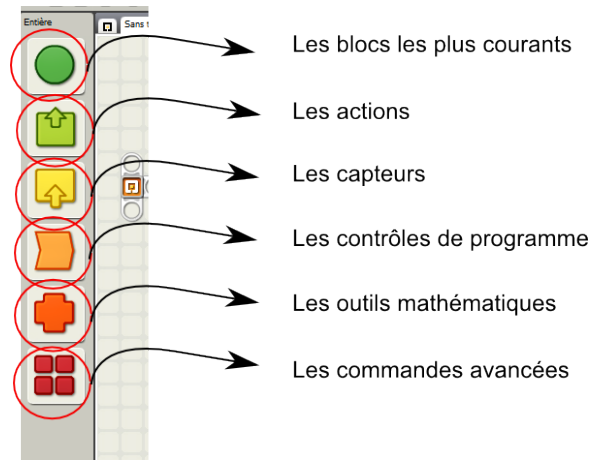


FIGURE 4 – Les boîtes à outils regroupant les commandes.

3 Un premier programme

3.1 Programmer

Le premier programme sera très simple : nous allons demander au robot d'avancer tout droit pendant 2 secondes, puis de tourner pendant 2 secondes. Pour cela, on dispose de deux blocs **déplacer** (disponibles dans la première boîte à outils : voir figure 5) sur le *fil du temps*.



FIGURE 5 – Les blocs du sous-menu Commune

Le premier bloc commandera d'aller tout droit pendant 2 secondes, et le bloc suivant lui ordonnera de tourner pendant 5 secondes.

La première chose à faire est de placer les deux blocs sur le fil du temps (facile !). Il faut ensuite fixer les paramètres de chacun des blocs.

Cliquez sur le premier bloc, ce qui le rend actif et affiche une série de six fenêtres en bas à gauche de l'écran (figure 6) où vous pouvez rentrer des informations :

- **Port** : Indique quels sont les moteurs concernés (ici, c'est B et C : s'ils ne sont pas cochés, faites-le).
- **Direction** : Avant, Arrière ou Arrêt : le sens de la marche (on choisit Avant).
- **Diriger** : Permet de faire tourner le robot.
- **Alimentation** : Permet de régler la puissance des moteurs.
- **Durée** : fixez-la à cinq secondes (l'unité utilisée est à choisir dans le menu ouvert à l'aide de la flèche à droite de cette fenêtre).
- **Action suivante** : indiquer si l'on veut que le robot continue en roue libre, ou bien s'arrête brusquement. Ce n'est pas très important pour nos exemples, vous n'avez qu'à essayer les deux pour voir la différence.

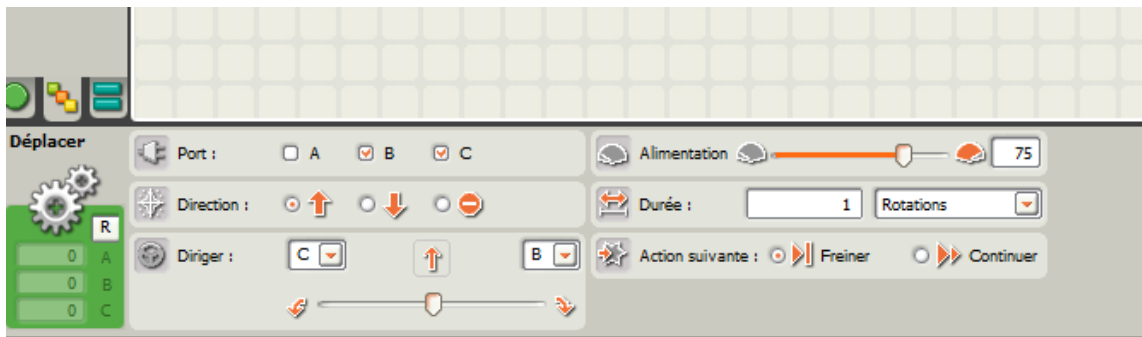


FIGURE 6 – Les paramètres d'un moteur

A vous maintenant : réglez les paramètres pour que le robot ait le comportement décrit au début du paragraphe.

Sauvegardez votre travail (Fichier -> enregistrez sous), connectez le robot au port USB, allumez-le (bouton orange sur la brique) et chargez votre programme dans le robot : utilisez la touche en bas à gauche du pavé de communication (voir figure 2). Une fenêtre s'ouvre au milieu de l'écran pour vous informer de l'état d'avancement du chargement. Si tout s'est bien passé, le programme est maintenant chargé dans le robot, vous pouvez débrancher le câble USB. Sinon, appelez au secours.

3.2 Tester

Sur la brique, à l'aide du bouton orange, sélectionnez **My Files** puis **Software Files** (les programmes que vous avez chargés), puis le nom du programme que vous avez sauvegardé. Appuyez sur le bouton orange encore une fois (Run) : ça y est, c'est parti. Pas très impressionnant, seulement 4 secondes de fonctionnement...

4 Tourner en rond

...mais si on lui demande de répéter ce comportement à l'infini, on obtient un robot qui tourne en rond sans s'arrêter, ce qui est déjà un peu mieux. Comment faire ?

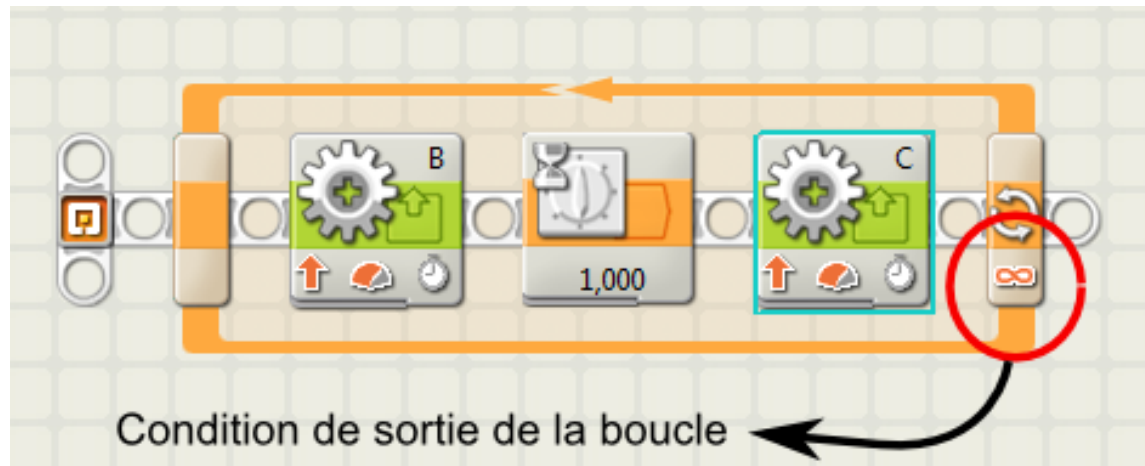


FIGURE 7 – Un exemple de boucle

Commencez un nouveau projet, puis allez chercher la boucle dans le sous-menu **Commun** (l'avant-dernière icône), et positionnez-la sur le fil du temps. Toutes les commandes figurant à l'intérieur de cette boucle seront exécutées de gauche à droite. Lorsque le robot aura exécuté la commande la plus à droite, il recommencera au début de la boucle. La figure 7 est un exemple simple d'utilisation de la boucle : en y entrant par la gauche, on active le moteur B pendant une seconde, puis on attend une seconde, et on active le moteur C pendant une seconde. Puis la boucle recommence.

A vous de jouer : programmez le robot pour qu'il tourne *en carré* : avancer 3 secondes, tourner à angle droit, avancer 3 secondes, tourner à angle droit, etc ...

Quelques petits réglages seront nécessaires.

5 Tourner en spirale

On voudrait maintenant faire tourner le robot en spirale, comme sur le schéma de la figure 8 : avancer une seconde, puis tourner de 90 degrés, avancer encore une seconde, puis tourner de 90 degrés, avancer deux secondes, tourner, etc ... Comment indiquer au robot de suivre ce comportement ? Il faudrait lui dire :

1. Fixe une longueur initiale n à 1.
2. Avance d'une distance n .
3. Tourne à 90 degrés.

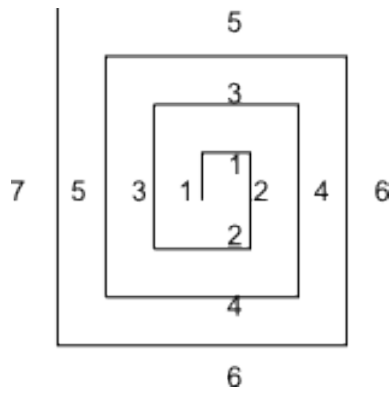


FIGURE 8 – Avancer en spirale

4. Avance de la même distance n .
5. Augmente n de une unité.
6. Recommence au point 2.

Le déroulement de ces instructions, du point 2 au point 6 correspond au fonctionnement de la boucle dont on vient de parler. Il faudrait juste avoir un bloc mémorisant la valeur actuelle de n , et un ou plusieurs autre blocs qui permettraient d'ajouter 1 à cette valeur.

Les boîtes nécessaires existent, elles sont dans le menu des données (figure 9)



FIGURE 9 – Les blocs du menu des données

Le petit programme de la figure 10 peut vous aider à comprendre comment les utiliser : On a créé une variable (c'est-à dire un endroit pour stocker une valeur numérique) qui s'appelle durée : pour cela on va dans le menu Modifier->Définir des variables->Créer, puis on choisit le type numérique, et le nom de la variable). De la même façon, on crée une constante (valeur numérique qui ne changera jamais). Ensuite commence la boucle :

- Les trois premiers blocs servent à afficher la valeur courante de la variable durée sur l'écran de la brique.
 - Lire la valeur de la variable.
 - Transformer le nombre en mot : l'écran ne *comprend* que les chaînes de caractères.
 - Afficher le texte sur l'écran.
- Les quatre blocs suivants changent la valeur de la variable durée :
 - On va chercher la valeur actuelle de la variable.

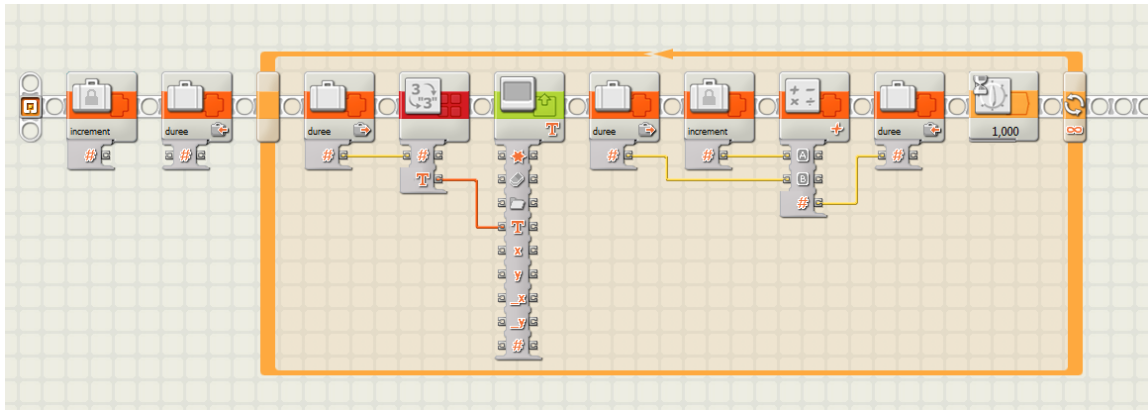


FIGURE 10 – Boucle et utilisation des variables

- On va chercher la valeur de la constante.
- On utilise un bloc addition pour calculer la somme de ces deux valeurs.
- On range le résultat dans la variable.
- Enfin, on place un bloc de temporisation, sinon l’affichage avance trop vite.

Une fois que vous avez compris (éventuellement testé) ce programme, vous devriez savoir le modifier pour qu’il accomplisse notre souhait initial, à savoir programmer le robot pour qu’il avance en spirale.

Pour demander au moteur d’avancer pendant une durée variable, il faut entrer la valeur voulue dans le bloc **déplacer**. Pour cela, cliquez sur le côté bas du bloc : une série de plots se déploie (figure 11). Branchez alors votre variable sur le plot **durée**.

6 Réagir à l’environnement

Les robots décrits jusqu’à maintenant fonctionnent comme des automates : ils font ce qu’on leur demande, mais ne prennent jamais de décisions, aussi simples soient-elles. En ajoutant des capteurs à un robot, on lui permet de percevoir son environnement. Si on peut le programmer pour que son comportement dépende de cette perception, on obtient une machine un peu plus intelligente. Le degré d’intelligence dépend de l’acuité de ses sens, et de la puissance de calcul qu’il utilisera pour prendre ces décisions.

Pour permettre au robot de percevoir son environnement, nous avons à notre disposition plusieurs types de capteurs :

- Des capteurs tactiles : le robot peut voir si ces interrupteurs sont enfoncés ; on s’en sert pour savoir si on a heurté un obstacle, par exemple.
- Des capteurs sonores : les oreilles du robot (tous les robots utilisés durant le stage n’ont pas de tels capteurs).
- Capteurs à ultrasons : ils permettent de calculer des distances.
- Capteurs photosensibles et capteurs de couleur : les yeux du robot.

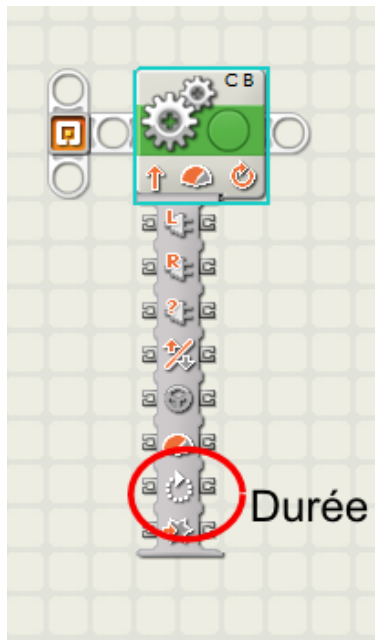


FIGURE 11 – Les plots de contrôle d’un bloc déplacer

- D’autres capteurs peuvent connaître l’état des moteurs, on peut les comparer aux nerfs qui nous renseignent sur le fonctionnement de nos muscles.

6.1 Application simple

Essayer de concevoir un programme ordonnant au robot d’aller tout droit jusqu’au moment où il heurte un obstacle, et lui demander alors :

- de s’arrêter.
- de reculer.
- de se remettre en marche.

6.2 Un peu plus dur (quoi que ...)

Le robot avance tout droit jusqu’à ce qu’il perçoive la présence d’un obstacle à moins de 5 cm. Il tourne alors de 90 degrés, et se remet en route.

Quelques indices :

- Les exemples de boucles que nous avons vus jusqu’à présent étaient des boucles infinies. On peut aussi indiquer que la boucle doit s’arrêter lorsqu’une condition est remplie : cliquez sur le symbole de l’infini à droite d’un bloc de boucle (voir figure 7). Vous pouvez alors fixer une condition d’arrêt dépendant des informations fournies par un capteur.

- Rien n’interdit d’avoir une boucle à l’intérieur d’une autre.

6.3 Nettement plus dur... mais pas infaisable

Le robot se promène dans un environnement parsemé d’obstacles. Lorsqu’il perçoit un obstacle dans la direction de son déplacement à moins de 5 cm, il change de direction jusqu’à temps qu’il constate que la voie est libre devant lui. Il se remet alors à avancer.

Et si on l’autorisait à changer parfois de direction comme ça, sans raison, au hasard ?