IBM program **Back To School**

Instructions rédigées par

Jan Van der Sypt

Software Group - Senior IT Specialist

Engineer a Robot

# Table des matières

[Table des matières 1](#_Toc338079003)

[Lego Robot NXT - Instructions 2](#_Toc338079004)

[Préambule 2](#_Toc338079005)

[Introduction 2](#_Toc338079006)

[Senseurs (ou capteurs) 3](#_Toc338079007)

[Servomoteur 3](#_Toc338079008)

[Environnement de programmation 4](#_Toc338079009)

[Mission 1 - Déplacement 5](#_Toc338079010)

[Aperçu – Objectif du programme 5](#_Toc338079011)

[Instructions, étape après étape 5](#_Toc338079012)

[Mission 2 – Capteur sensitif (ou de contact) 8](#_Toc338079013)

[Aperçu – Objectif du programme 8](#_Toc338079014)

[Instructions, étape après étape 8](#_Toc338079015)

[Mission 3 – Capteur auditif (bruit) et capteur visuel (ou de lumière) 12](#_Toc338079016)

[Aperçu – Objectif du programme 12](#_Toc338079017)

[Instructions, étape après étape 12](#_Toc338079018)

[Quatrième mission 19](#_Toc338079019)

[Aperçu – Objectif du programme 19](#_Toc338079020)

[Instructions 19](#_Toc338079021)

[Mission optionnelle 21](#_Toc338079022)

[Répétition 21](#_Toc338079023)

[Activités parallèles 22](#_Toc338079024)

[Referentiemateriaal 24](#_Toc338079025)

[De tastsensor 24](#_Toc338079026)

[De geluidssensor 24](#_Toc338079027)

[De ultrasone sensor 25](#_Toc338079028)

[De lichtsensor 25](#_Toc338079029)

[De herhaling 26](#_Toc338079030)

# Lego Robot NXT - Instructions

## Préambule

**Lego Mindstorms**

***-http://mindstorms.lego.com/en-us/default.aspx-***

**Tu as un intérêt pour les techniques, pour la robotique ?**

La robotique c’est l’avenir. Les robots humanoïdes et autres machines de chantier sont le fleuron de ce type de technologie. Alors un robot humanoïde qui marche ou un engin de chantier en action fait en [Lego Mindstorms](http://mindstorms.lego.com/en-us/default.aspx), c’est le summum de la classe !

**Lego Mindstorms, mais qu’est-ce que c’est ?**

Legos techniques associés à de petits ordinateurs de bord et autres composants électroniques, les jeux et robots développés par Lego Mindstorms sont sources de créativité. Ils éveillent en tout un chacun l’envie de développer sa créativité.

**Qu’est-ce qui t’attend dans cet atelier ? Que pourras-tu y découvrir ?**

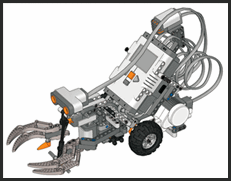
Dans cet atelier, tu auras à ta disposition un robot opérationnel doté d’une mâchoire préhensile.

Pour que ce robot puisse réaliser les tâches que tu lui auras assignées, c’est-à-dire celles d’attraper une balle positionnée dans le périmètre défini, tu devras le programmer. Le programmer ? Oui, lui donner des instructions claires afin qu’ils comprennent tes attentes.

Ton robot sera muni de différents capteurs et moteurs. A toi de paramétrer ces composants, afin de guider les actions de ton robot, grâce au programme que tu auras réalisé.

Intéressé, n’attends plus ! 8-) Joins-toi à nous et exerces-toi à programmer ces super petits robots afin qu’ils répondent aux actions que tu auras toi-même définies et que tu leur auras communiquées. 8-)

## Introduction

A l’achat, le robot doit être assemblé. Dans cet atelier, nous te fournissons cependant un robot déjà opérationnel qu’il ne te reste qu’à programmer selon tes désires afin qu’il accomplisse les tâches souhaitées.

Par programmation de l’ordinateur de bord associé (PLC/NXT), ton robot recevra au niveau de divers capteurs (4 max) et les moteurs (3 max) l’impulse nécessaire à ces actions.

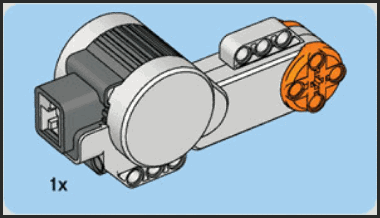
Le PLC interprète en effet les signaux entrants des capteurs selon le programme activé & chargé et commande de la sorte le servo-moteurs afin de lui faire exécuter les activités visée.

### Senseurs (ou capteurs)

Quatre capteurs peuvent être connectés via quatre ports disponibles au niveau du robot et son ordinateur de bord.

|  |  |
| --- | --- |
| Port 1  tastsensor | Le ‘capteur de contact’, c’est les “doigts” de ton robot`.  Avec ce capteur de contact sensible à la pression (sorte d’interrupteur), tes robots LEGO® MINDSTORMS® NXT vont découvrir la sensation du “toucher” !  Il peut détecter une ou plusieurs pressions de boutons et envoie toutes ces informations à la brique intelligente NXT |
| Lichtsensor  Port 3 | Le ‘capteur de lumière’ réagit selon l’intensité de la lumière (lumière présente ou absente). Le signal est transmis dès que l’intensité de la lumière atteint un niveau suffisant. |
| ultrasone sensor  Port 4 | Le ‘capteur ultrasonique’ réagit à la présence ou à l'absence d'un objet. La distance entre le robot et l'objet conditionne le moment où le signal est envoyé.. |
| geluidssensor  Port 2 | Le ‘capteur de bruit’ réagit au son, au bruit. Le niveau de bruit doit avoir atteint un seuil suffisant avant envoi du signal. |

### Servomoteur

Trois moteurs liés au PLC conditionnent les mouvements du robot. Chacun offre des fonctionnalités.

Chaque moteur permet d’aller vers l’avant ou l’arrière devant ou la marche arrière. La puissance, la distance, et la durée (temps) sont configurables.

Les trois servomoteurs sont raccordés via les sorties output du PLC. L'ordre de raccordement est important.

Le programme donne en effet les commandes selon l’ordre des outputs, et c’est cette commande qui permettra de générer le mouvement.

Par défaut, les sorties pour les deux roues sont la porte B et la porte C. Le troisième moteur est raccordé sur la porte A (au milieu). Ce moteur actionne la ‘pince de préemption’.

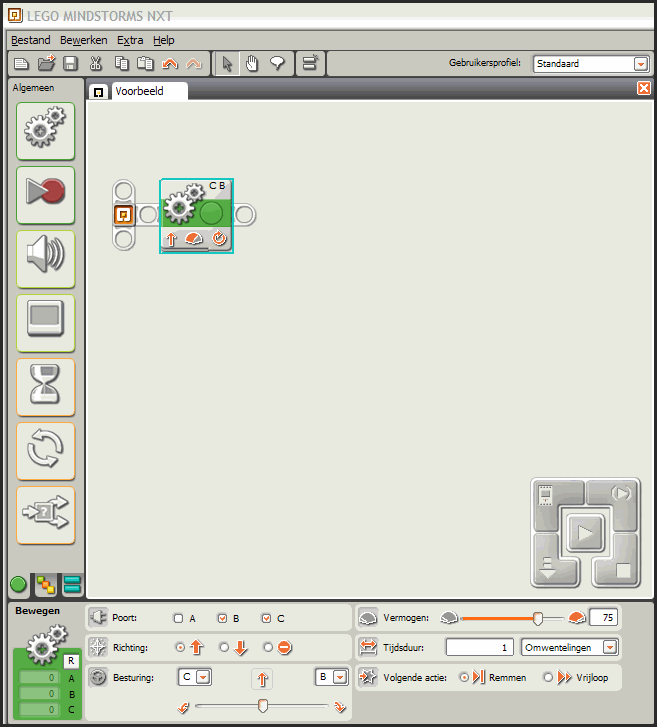
MCIN00695_0000[1]

Le moteur de commande de la ‘pince de préemption’ et permet à cette dernière de pouvoir bouger d’avant en arrière et de la sorte, permettant par la même occasion d’ouvrir la ‘pince de préemption’

## Environnement de programmation

L’environnement de programmation doit être installé sur un PC. Les instructions programmées sur le PLC sont transmises au robot via liaison USB. Les instructions sont construites graphiquement. Les composantes visuelles deviennent les commandes de programmation que le PLC pourra comprendre et interpréter afin d’effectuer les actions. Le transfert de commande est nommé compilation.

L’illustration suivante reflète l’environnement de programmation et ces principaux paramètres de commande.



Dans l'écran de configuration, le bloc de programmation sélectionné est adapté.

Les blocs de programmation doivent être glissés et ordonnancés dans l’éditeur visuel.

Le contrôleur permet de transmettre (charger) les instructions au PLC

La palette reprend les blocs de programmation disponibles

## Mission 1 - Déplacement

### Aperçu – Objectif du programme

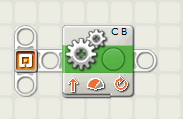
* Le robot avance de deux rotations.
* Le robot dit "Hello".
* Ensuite le robot fait marche arrière jusqu'à la position de départ.

### Instructions, étape après étape

1. Crée un nouveau programme et donne-lui un nom spécifique

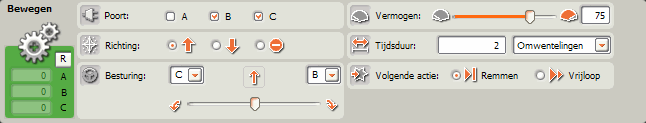


1. Déplace (glisse) un bloc "déplacer" vers l’espace vide de l’écran afin de démarrer l’écran d’édition

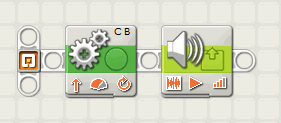


Le robot roule pendant deux rotations

1. Configure le bloc "déplacer" avec, comme durée, deux rotations. Les servomoteurs B et C ont été sélectionnés. Les deux roues du robot sont actionnées par ces deux servomoteurs.

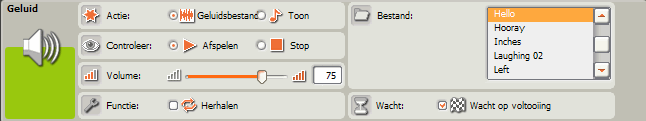


1. Glisse un bloc "bruit" derrière le bloc "déplacer" dans l’éditeur visuel



1. Configure le bloc "bruit" afin de produire un signal sonore. L’environnement de programmation offre une série de bruits et de signaux parmi lesquels choisir. Sélectionne par exemple "Hello". Adapte éventuellement le volume.

Volume plus fort?



1. Déplace (glisse) un bloc "déplacer" vers l’écran d’édition, après le bloc "bruit"

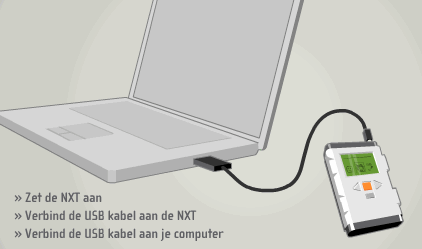


Le robot roule en arrière durant deux rotations

1. Configure le second bloc "déplacer"



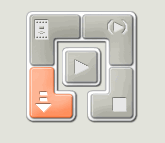
1. Connecte le robot au PC via le câble USB et allume le (position ‘on’).



LE BUT EST QU’A LA FIN DE LA PREMIERE LESSON, UN DEMO DE CE QUE DOIT PERMETTRE DE REALISER LE PROGRAMME PARAME SOIT DONNEE, APRES CHARGEMENT DU PROGRAMME DANS UN ROBOT DANS UN ROBOT

IL FAUT DONC PREVOIR EN AVANCE UNE SAUVEGARDE DE CHAQUE PROGRAMME SUR LE DISQUE.

1. Charge le programme dans le robot



1. Déconnecte le robot du PC
2. Place le robot dans le ‘playground’ de test
3. Exécute le programme



Si le comportement du robot n'est pas celui espéré, c’est que le programme chargé par le stagiaire est erroné. Une version correcte du programme (version de démo) sera dans ce cas chargée sur le robot du stagiaire, en remplacement de la version erronée. Le même nom de programme pourra être utilisé.

## Mission 2 – Capteur sensitif (ou de contact)

### Aperçu – Objectif du programme

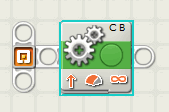
* Le robot roule vers avant jusqu’à atteindre la position de la balle
* Après une seconde, le robot saisit la balle.
* Le robot fait ensuite marche arrière jusqu'à la position de départ.

### Instructions, étape après étape

1. Crée un nouveau programme et donne-lui un nom spécifique

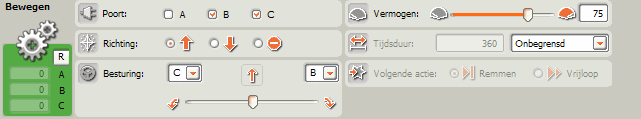


1. Déplace (glisse) un bloc "déplacer" vers l’espace vide de l’écran afin de démarrer l’écran d’édition

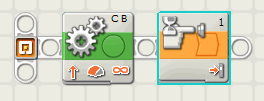


Le robot doit continuer à avancer jusqu’à la position de la balle. La distance à parcourir n’est pas connue par le robot.

1. Configure le bloc "déplacer" avec une durée illimitée.



1. Glisse un bloc "attendre" derrière le bloc "déplacer" dans l’éditeur visuel



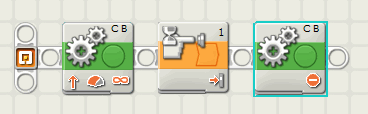
Un cycle d’attente est déclenché. Ce cycle d’attente se prolongera jusqu’à ce que l’interrupteur du capteur sensitif soit enfoncé, c’est à dire jusqu’au moment où le robot atteindra et touchera la balle.

1. Configure le bloc "attendre" du capteur sensitif.

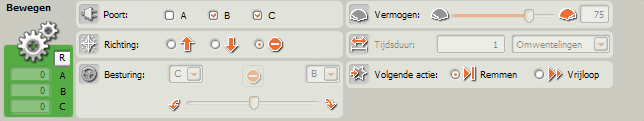


1. Glisse un bloc "déplacer" derrière le bloc "attendre" dans l’éditeur visuel

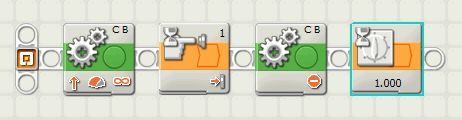
Dès que le robot atteint la balle, le capteur donne le signal. Les cycles d’attente sont interrompus le programme se poursuit avec comme instruction "arrêter".



1. Configure le second bloc "déplacer" avec un ‘freinage immédiat’.

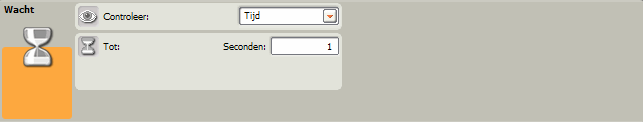


1. Glisse un bloc "attendre" derrière le bloc "déplacer" dans l’éditeur visuel

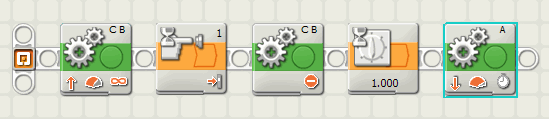


1. Configure le bloc "attendre" avec une durée de 1 seconde.

Le robot attend pendant 1 seconde.

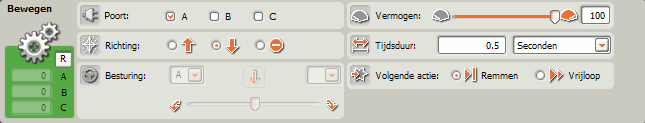


1. Glisse un bloc "déplacer" derrière le bloc "attendre" dans l’éditeur visuel



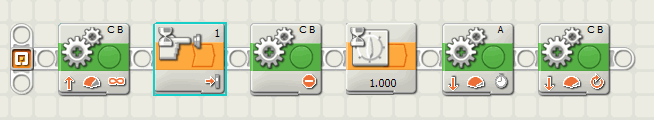
Le servomoteur connecté sur l’output du port A doit effectuer une rotation inverse pendant ½ seconde afin de provoquer la fermeture de la pince.

1. Configure le troisième bloc "déplacer" afin d’induire la fermeture de la ‘pince de préemption’. Le troisième servomoteur actionne la pince. La fermeture de la pince est provoquée par la rotation inversée du moteur pendant une demi-seconde. La puissance maximale est requise pour induire cette action dans les plus brefs délais.

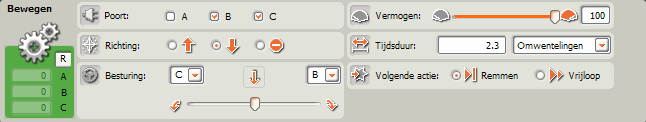


Le mouvement n’impacte que le servomoteur de la pince

1. Glisse un bloc "déplacer" derrière le troisième bloc "déplacer" dans l’éditeur visuel



1. Configure le quatrième bloc "déplacer" afin d’induire le recul du robot (roule en arrière). Un déplacement de 2,3 rotations permet de ramener le robot à sa position initiale.



Le robot roule en arrière sur une distance de 2,3 rotations

1. Connecte le robot au PC via le câble USB et allume le (position ‘on’).
2. Charge le programme dans le robot
3. Déconnecte le robot du PC
4. Place le robot dans le ‘playground’ de test
5. Exécute le programme

Pense aussi à sauvegarder le programme sur

le disque dur du PC!!

Si le comportement du robot n'est pas celui espéré, c’est que le programme chargé par le stagiaire est erroné. Une version correcte du programme (version de démo) sera dans ce cas chargée sur le robot du stagiaire, en remplacement de la version erronée. Le même nom de programme pourra être utilisé.

## Mission 3 – Capteur auditif (bruit) et capteur visuel (ou de lumière)

### Aperçu – Objectif du programme

* Le robot roule vers avant jusqu’à atteindre la position de la balle et attend
* Dès qu’un certain niveau sonore est détecté, le robot attrape (agrippe) la balle
* Le robot effectue ensuite une rotation de 180° sur lui-même, sans heurter la pince préemptive
* Le robot avance par la suite jusqu’à atteindre un bord noir dessiné sur le ‘playground’ et y libère la balle.

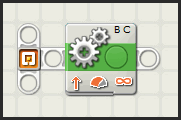
### Instructions, étape après étape

1. Crée un nouveau programme et donne-lui un nom spécifique

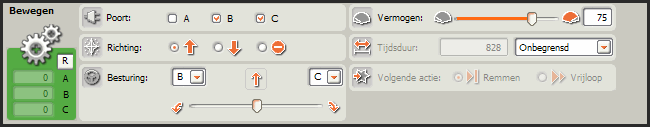


1. Déplace (glisse) un bloc "déplacer" vers l’espace vide de l’écran afin de démarrer l’écran d’édition

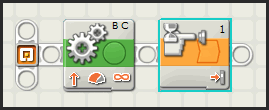
Le robot reçoit comme commande d’avancer (rouler vers l’avant), sans connaître en avance le point d’arrêt. Ni la durée, ni la distance à parcourir ne lui son communiquée.



1. Configure le bloc "déplacer" avec une durée illimitée.

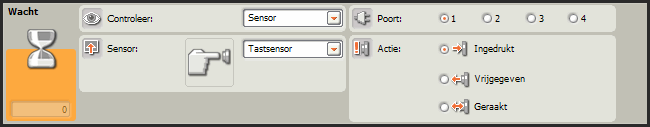


1. Glisse un bloc "attendre" derrière le bloc "déplacer" dans l’éditeur visuel



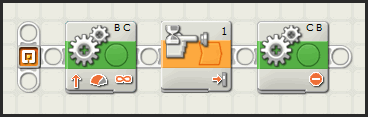
Le cycle d’attente se déclenche dès que le robot avance... Ce cycle se poursuit tant que le capteur sensitif n’a pas été déclenché.

1. Configure le bloc "attendre" du capteur sensitif.

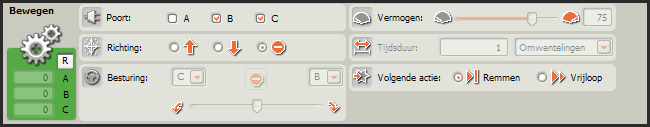


1. Glisse un bloc "déplacer" derrière le bloc "attendre" dans l’éditeur visuel

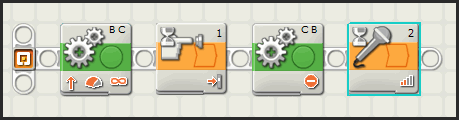
Dès que l’interrupteur du capteur est enfoncé, le cycle de garde est interrompu et le programme se poursuit. Le bloc suivant permet le freinage et l’arrêt (quasi immédiat) du robot.



1. Configure le second bloc "déplacer" avec un ‘freinage immédiat’.

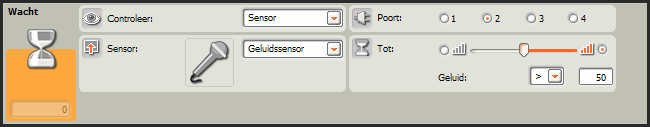


1. Glisse un bloc "attendre" derrière le bloc "déplacer" dans l’éditeur visuel



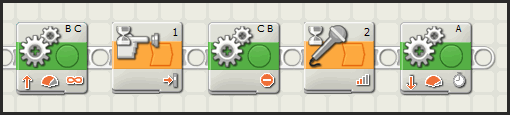
Le robot attend puis entame un nouveau cycle d’attente. Ce cycle est dirigé par le capteur auditif.

1. Configure le bloc ‘attendre’ en induisant un déclenchement du capteur auditif avec un niveau de bruit supérieur à 50db.

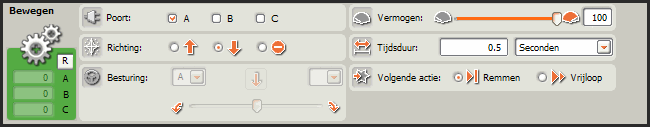


Le cycle de garde est interrompu dès que le capteur auditif détecte un niveau de bruit 50% supérieur au niveau de bruit maximum normal de l’environnement donné.

1. Glisse un bloc "déplacer" derrière le bloc "attendre " dans l’éditeur visuel



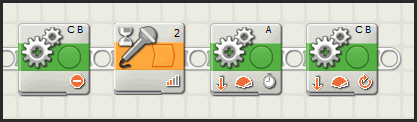
1. Configure le troisième bloc "déplacer" afin d’induire la fermeture de la ‘pince de préemption’. Le troisième servomoteur actionne la pince. La fermeture de la pince est provoquée par la rotation inversée du moteur pendant une demi-seconde. La puissance maximale est requise pour induire cette action dans les plus brefs délais.



Paramètre ½ seconde comme valeur et ne précise pas de ‘rotation’ ou de ‘degré’ comme unité.

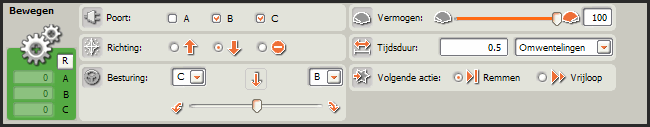
Que se passerait-il si tu faisais usage de ces unités?

1. Glisse un bloc "déplacer" derrière le troisième bloc "déplacer" dans l’éditeur visuel



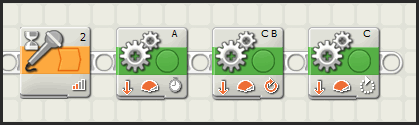
Le robot effectue un mouvement de recul afin de disposer de l’espace suffisant pour pouvoir tourner

1. Configure le quatrième bloc "déplacer" afin d’induire le point de recul du robot (roule en arrière). Un déplacement de 0,5 rotations est suffisant pour permettre une rotation sûre

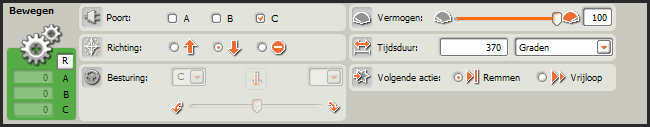


1. Glisse un bloc "déplacer" derrière le quatrième bloc "déplacer" dans l’éditeur visuel

Le robot tourne selon l’axe d’une première roue qui sera provoquée par commande. Une seconde roue sera ensuite provoquée dans le sens inverse afin de poursuivre le mouvement de rotation du robot.



1. Configure le cinquième bloc "déplacer" afin d’induire un mouvement inverse du servomoteur et spécifie une durée de 370°

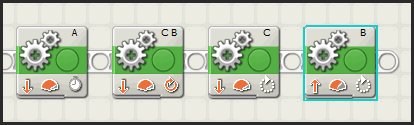


Une rotation de 370° du moteur induit un mouvement de rotation du robot d’environ 90°!!!

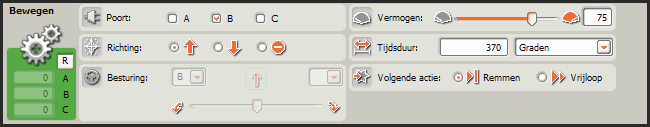
1. Glisse un bloc "déplacer" derrière le cinquième bloc "déplacer" dans l’éditeur visuel

Le même processus est appliqué à une seconde roue en provoquant une rotation inversée.

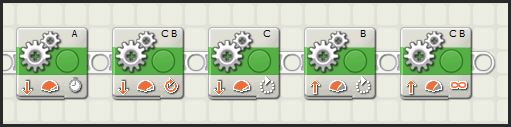
Ce faisant, le robot aura effectué une rotation de 180°.



1. Configure le sixième bloc "déplacer" afin d’induire un mouvement inverse du servomoteur et spécifie une durée de 370°

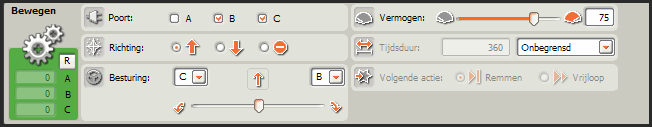


1. Glisse un bloc "déplacer" derrière le sixième bloc "déplacer" dans l’éditeur visuel

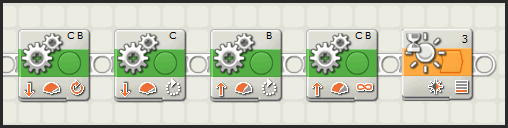


Le robot roule en direction du point de départ mais sans induire de limite spécifique à ce déplacement. Le but est d’induire l’arrêt du robot dès qu’il aura détecté la ligne noire de contour du ‘playground’. La position de la ligne sera interprétée par le robot via son capteur visuel (capteur de lumière)

1. Configure le bloc "déplacer" afin d’induire un mouvement vers l’avant sans limite fixée

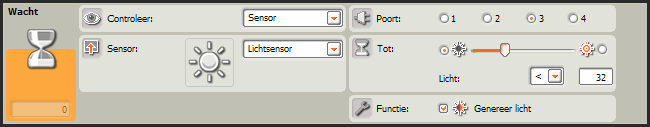


1. Glisse un bloc "attendre" derrière le septième bloc "déplacer" dans l’éditeur visuel

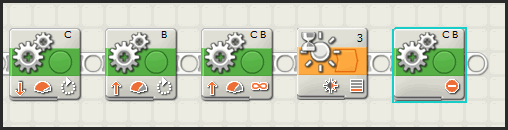


Dès que le robot se met en mouvement, un cycle de garde est induit. Ce cycle est géré au travers de la configuration du capteur de lumière. Le capteur génère de la lumière et mesure le pourcentage de lumière retourné. Si ce pourcentage tombe sous un niveau de 32%, le cycle de garde est interrompu.

1. Configure le bloc ‘attendre’ en induisant un déclenchement du capteur visuel dès que le niveau de lumière réfléchie est inférieur à 32%. Selon le contexte environnant, cette valeur pourra être adaptée afin de procurer une réponse optimale du robot.

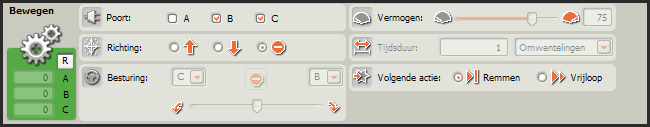


1. Glisse un bloc "déplacer" derrière le bloc "attendre " dans l’éditeur visuel

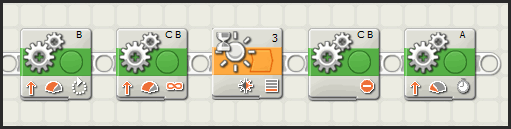


Arrête le robot!

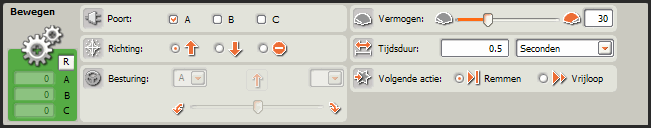
1. Configure le bloc "déplacer" afin d’induire l’arrêt du robot



1. Glisse un bloc "déplacer" derrière le septième bloc "déplacer" dans l’éditeur visuel



1. Configure le bloc "déplacer" afin d’induire l’ouverture de la pince. Le servomoteur A doit pour ce faire être actionné pendant une demi-seconde. La puissance est donc a diminuée à concurrence de 30%.



La pince est commandée par le servomoteur A. Les autres servomoteurs qui commandent les roues ne doivent donc pas être sélectionnés! La pince revient dans sa position / configuration intiale.

1. Connecte le robot au PC via le câble USB et allume le (position ‘on’).
2. Charge le programme dans le robot
3. Déconnecte le robot du PC
4. Place le robot dans le ‘playground’ de test
5. Exécute le programme

Pense aussi à sauvegarder le programme sur le disque dur du PC!!

Si le comportement du robot n'est pas celui espéré, c’est que le programme chargé par le stagiaire est erroné. Une version correcte du programme (version de démo) sera dans ce cas chargée sur le robot du stagiaire, en remplacement de la version erronée. Le même nom de programme pourra être utilisé.

## Quatrième mission

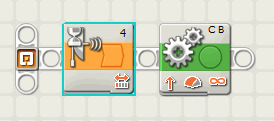
### Aperçu – Objectif du programme

* Le robot n’avance que si la balle est située ou placée à une distance de moins de 50 cm.
* Dès qu’il détecte la présence de la balle, le robot avance jusqu’à la balle et attend.
* Lorsqu’il perçoit un niveau sonore donné, il s’empare de la balle
* Le robot tourne de 180° sans heurter la pince retenant la balle.
* Le robot le robot roule jusqu’à la ligne de délimitation du ‘playground’ et libère la balle une fois ce périmètre atteint.

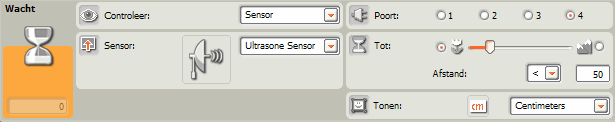
### Instructions

Ce programme étant très similaire à celui décrit pour la mission 3, les instructions ne sont pas ici décrites étape par étape. Ce programme débute toutefois autrement que dans la mission 3.

Au lieu de débuter par un bloc "déplacer", le programme débute ici directement avec un bloc "attendre". Le cycle de garde est conditionné par le capteur ‘ultrasonique’. Si le capteur ultrasonique est configurer afin de détecter des objets à une distance de 50 cm, dès lors qu’un objet est détecté, le cycle s’interrompt.



Tant qu’aucun objet n’est disposé à moins de 50cm du robot, celui reste immobile



Si la balle est déposée devant le robot, le robot avance jusqu’à sa position. C’est alors le capteur sensitif qui prend le relais, en induisant l’arrêt du robot dès lors qu’il a atteint (rencontré) la position de la balle. La suite de la mission peut être calquée sur les instructions décrites dans la mission 3.

Pense aussi à sauvegarder le programme sur le disque dur du PC!!

Si le comportement du robot n'est pas celui espéré, c’est que le programme chargé par le stagiaire est erroné. Une version correcte du programme (version de démo) sera dans ce cas chargée sur le robot du stagiaire, en remplacement de la version erronée. Le même nom de programme pourra être utilisé.

## Mission optionnelle

Utiliser les blocs de programmation, fonctionnalités et bonnes pratiques (techniques) mises à disposition dans le software pour écrire son propre programme que le robot sera à même d’interpréter afin d’effectuer, par la suite, une série contrôlée des mouvements.

Des techniques complémentaires pourront éventuellement être associées aux blocs fonctionnels afin de développer le potentiel de fonctionnalités du programme.

Ci-après, décrivons quelques outils complémentaires utiles en programmation

* Répétition
* Activités parallèles



LE BUT EST QU’A LA FIN DE LA PREMIERE LESSON, UN DEMO DE CE QUE DOIT PERMETTRE DE REALISER LE PROGRAMME PARAME SOIT DONNEE, APRES CHARGEMENT DU PROGRAMME DANS UN ROBOT DANS UN ROBOT

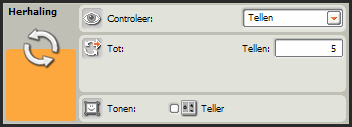
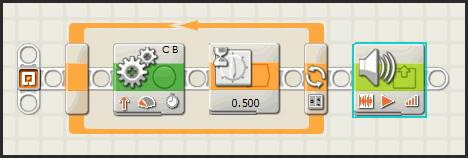
IL FAUT DONC PREVOIR EN AVANCE UNE SAUVEGARDE DE CHAQUE PROGRAMME SUR LE DISQUE.

### Répétition

Le bloc “répétition” peut être intégré dans un programme par glissement vers l'éditeur visuel et intégration adéquat dans le schéma du programme élaboré. La répétition est provoquée par adaptation de la valeur d’un compteur, une durée, ou une modification apportée au niveau d’un capteur. Une description plus complète et détaillée de ces principes est fournie dans le guide utilisateur (et en fin de document, en NL)

Ci-après est illustré un exemple de programme utilisant un bloc “répétition” configuré avec un compteur imposant cinq répétitions. Tous les blocs qui ont été insérés dans la répétition seront répétés cinq fois. En l’occurrence, le robot va effectuer cinq fois une marche arrière d’une durée d’une demi-seconde et, entre chaque répétition, le robot effectuera une pause d’une demi-seconde. Après la fin du cycle de répétitions, un signal auditif sera envoyé et le programme sera terminé.

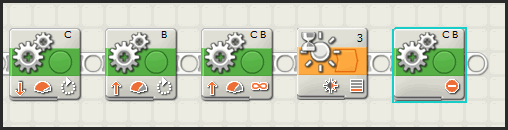
La configuration du bloc “répétition” est ici basée sur un simple compteur.

### Activités parallèles

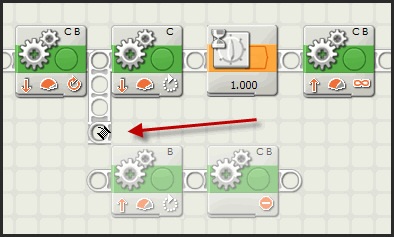
Jusqu'à présent, tous les programmes présentés se structuraient selon un schéma typiquement séquentiel, les blocs étant exécutés les uns après les autres, selon l’ordre imposé. Il est cependant possible de rendre fonctionnel et appliquer simultanément divers cheminements.

Le programme suivant présente un programme permettant de faire faire un mouvement rotatif au robot. Au lieu de donner un impulse à deux roues séparément et successivement, les actions sur les deux roues seront provoquées simultanément. Il est impossible de réaliser cette action avec un seul bloc fonctionnel puisque ce mouvement implique de donner à deux servomoteurs une commande impliquant un mouvement contraire. L’une devra tourner de 370 degrés dans un sens, l’autre devra tourner de 370 degrés dans le sens contraire.



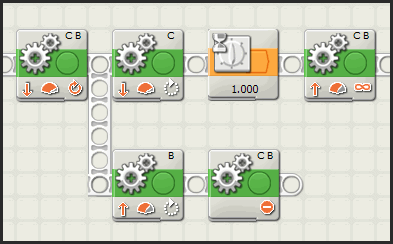
Le bloc "Déplacer" impactant le servomoteur connecté à l'output B provoque est déplacé (glisser) en dehors du flux. Une ombre blanche couvre le bloc et indique que le bloc extrait du flux n’est plus relié au flux primaire.

Le curseur de la souris est placé au point de connexion du bloc "déplacer" et la touche ‘SHIFT’ est enfoncée. Le curseur se modifie en une forme de ‘bobine’ et garde cette visualisation aussi longtemps que ‘SHIFT’ reste enfoncé.



Tandis que la touche ‘SHIFT’ est enfoncée, nous glissons le curseur pour qu'un nouveau connecteur apparaisse. Nous relions ce connecteur avec le composant déconnecté.

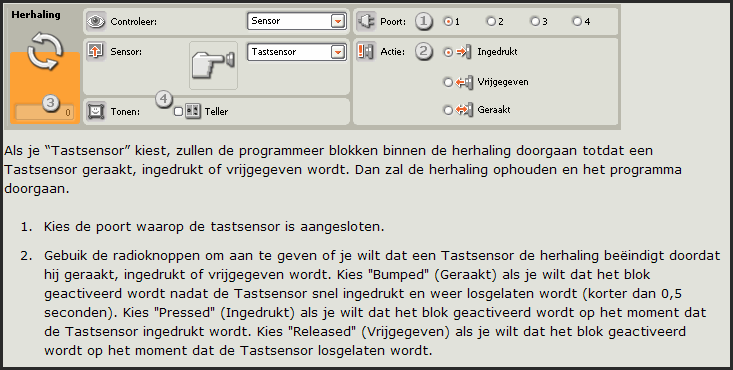
Nous obtenons finalement un cheminement parallèle. Au nœud de connexion, chaque sentier est traité comme un égal. Chaque cheminement parallèle est effectué simultanément.



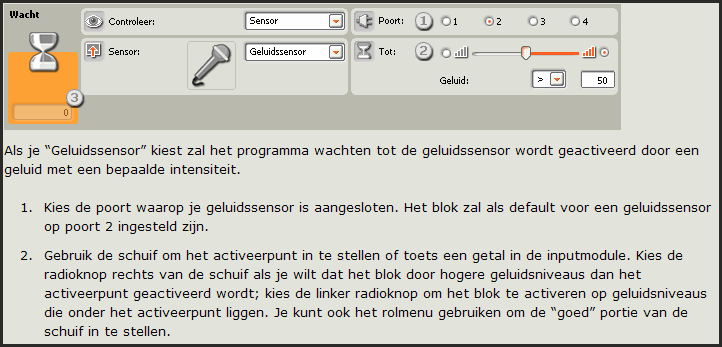
Dans ce programme, les deux servomoteurs sont actionnés simultanément mais avec une configuration imposant une rotation en direction contraire : le robot tourne sur son axe, les moteurs sont ensuite arrêtés et après une seconde le programme poursuit son exécution en suivant le cheminement unique. Le cheminement primaire parallèle est mené à son terme.

# Referentiemateriaal

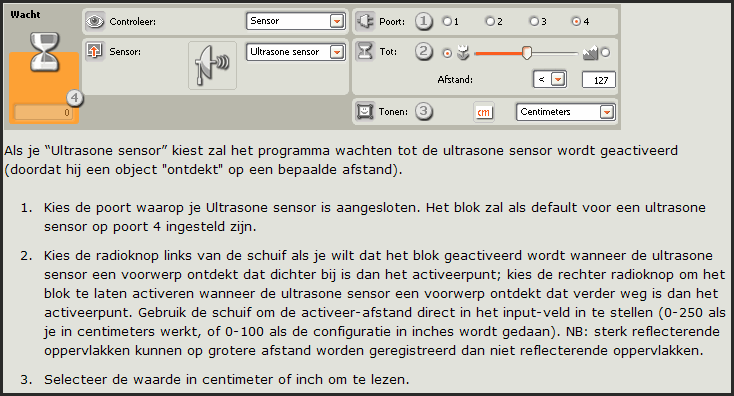
## De tastsensor



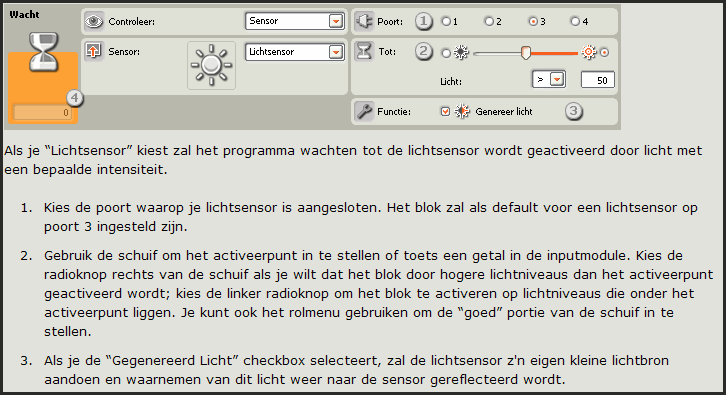
## De geluidssensor



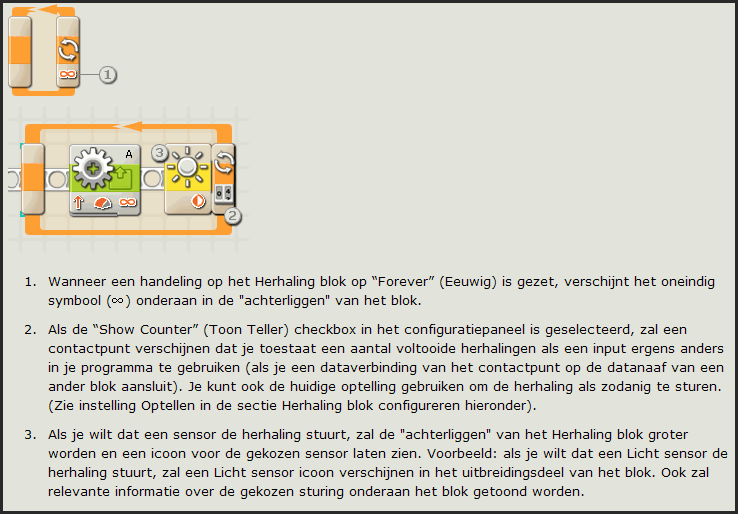
## De ultrasone sensor

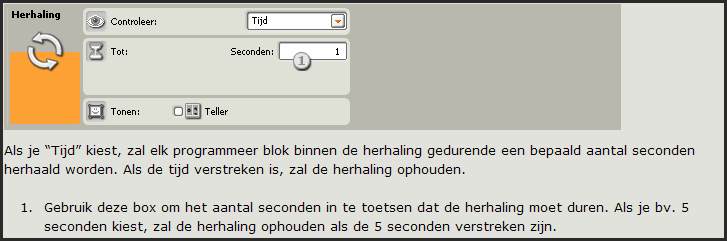


## De lichtsensor



## De herhaling





Het is mogelijk om een sensor te gebruiken die als een ‘wachtpost’ zal dienen voor de herhaling: de herhaling wordt doorlopen totdat een sensor een welbepaalde waarde inleest. De volgende zijn voorbeelden voor geluid of tast, maar ook andere sensoren kunnen gebruikt worden.

