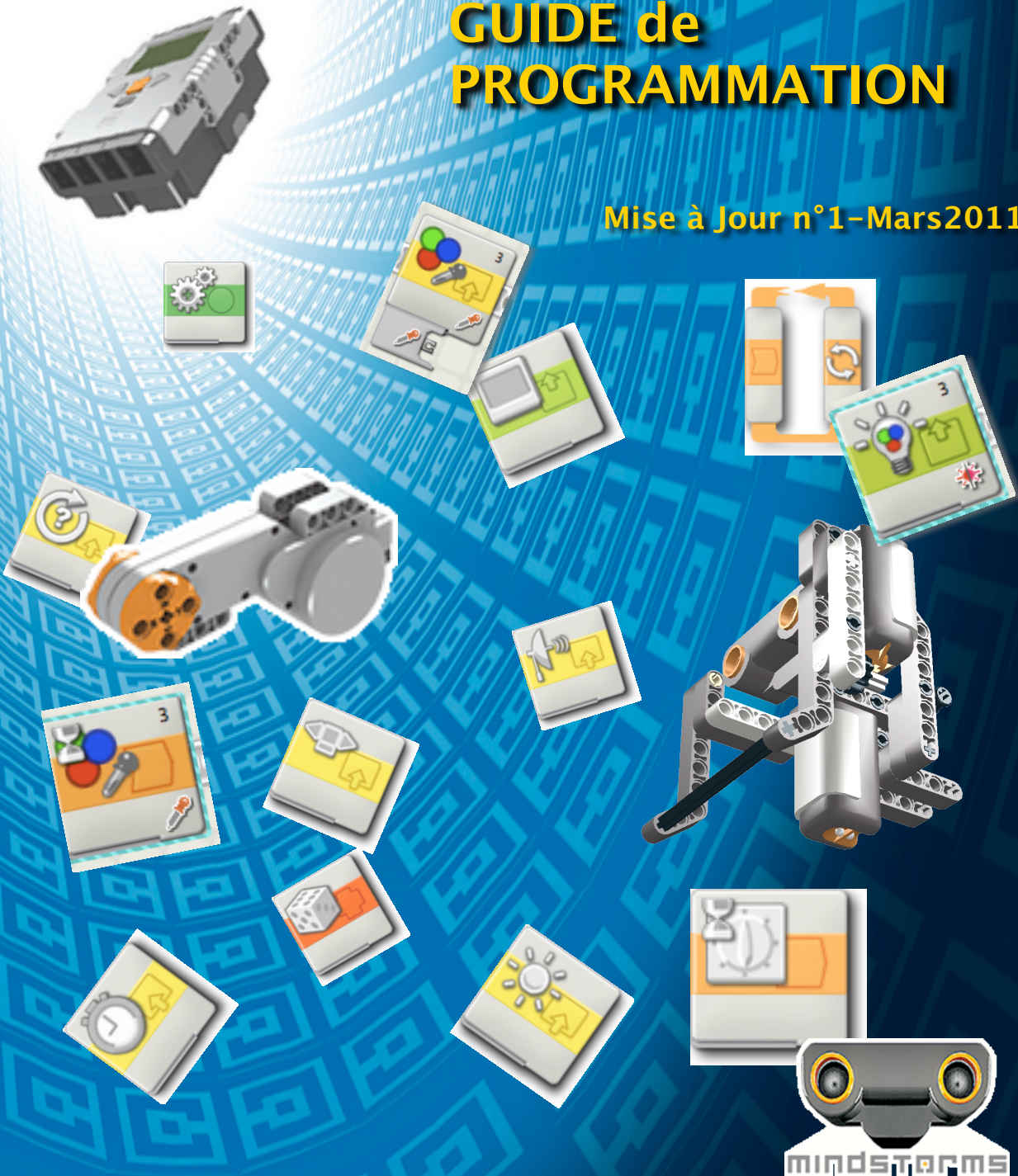


LEGO mindstorms
NXT 2.0

GUIDE de PROGRAMMATION

Mise à Jour n°1-Mars2011



Par Norbert Czaryski

<http://web.mac.com/roboleo/Roboleo/Accueil.html>

Table des matières

1 - Programmation NXT-G version 2.0	6
Introduction	6
Quelques mots sur la programmation	7
2 - Comparaison des kits	8
Les différences	8
Ce qui n'a pas changé	9
La brique NXT	9
Les capteurs	9
Les servomoteurs	9
Les câbles de liaison	9
Les pièces Technic	10
Les roues dentées	10
Les roues	10
Les poutres studless	10
Les pièces spéciales	10
Remarques	10
3 - L'ordinateur et le logiciel NXT-G2.	11
Configuration requise	11
Configuration PC:	11
Configuration Apple Macintosh:	11
Installation du logiciel	11
Nouveautés	12
Ce qui a changé	12
Nouvelles fonctions:	12
4 - Installer le microprogramme sur le NXT2	14
Quelques rappels	14
Le microprogramme	14
Première installation	14
Dépannage	15
Mise à jour du microprogramme	15
Assistant d'importation et d'exportation de blocs...(màj 1)	
1	16
5 - Mise à jour de la brique NXT1	20
Rappel	21
Les nouveaux blocs	21
Dans la palette commune	21
Dans la palette entière	22
6 - Compatibilité entre les 2 versions	23
Question 1: NXT2 exécute-t-il les programmes écrits avec NXT-G1?	23

Question 2: NXT1 exécute-t-il les programmes écrits ou lus avec NXT-G2?	23
Question 3: le logiciel NXT-G2 serait-t-il téléchargeable sans acheter le set NXT2.0 ?	23
CONCLUSION:	23

7 - Le capteur qui prends des couleurs... 25

Introduction	25
--------------	----

Test des fonctions	25
--------------------	----

A) choisir le menu TRY de la brique:	25
B) choisir le menu VIEW de la brique, puis REFLECTED LIGHT > PORT 3	26
C) Test avec un programme élémentaire:	26

8 - Les Nouveaux blocs 28

Les blocs bizarres	28
--------------------	----

Le bloc «Capteur de couleur»	28
------------------------------	----

Mode de détection de couleurs	29
-------------------------------	----

Mode de mesure de l'intensité lumineuse	30
---	----

Le bloc «Attendre»	32
--------------------	----

Le bloc «Commutation»	33
-----------------------	----

Le bloc «Boucle»	34
------------------	----

Le bloc «Lampe de couleur»	35
----------------------------	----

Le bloc «Constante»	36
---------------------	----

Le bloc «Connexion Bluetooth»	37
-------------------------------	----

Examen du bloc	38
----------------	----

Où et comment utiliser ce bloc?	39
---------------------------------	----

9 - Exemples de programmation 40

Le capteur de couleurs	40
------------------------	----

En mode Couleurs	40
------------------	----

En mode capteur photosensible	42
-------------------------------	----

En mode lampe de couleur	44
--------------------------	----

Connexion Bluetooth	44
---------------------	----

Connexion entre 2 NXT	44
-----------------------	----

Constantes et «Mon Bloc»	45
--------------------------	----

Rappels	46
---------	----

Créer un Mon Bloc avec des paramètres d'entrée	48
--	----

Utilisation d'un Mon Bloc avec un paramètre d'entrée	52
--	----

Concevoir son propre paramètre d'entrée	53
---	----

Changer le nom d'un paramètre	54
-------------------------------	----

Créer un Mon Bloc avec plusieurs paramètres d'entrée	56
--	----

Variables et «Mon Bloc»	57
-------------------------	----

Constante ou Variable?	57
------------------------	----

Définir des variables dans un «Mon Bloc»	58
--	----

Ré-écrire le contenu du «Mon Bloc»	58
------------------------------------	----

Partage des fils de données dans un Bloc «Commutateur»	59
--	----

Utiliser les variables du Mon Bloc à partir du programme principal	61
--	----

Réaliser une copie modifiée d'un «Mon Bloc»	62
---	----

Copier un «Mon Bloc»	62
----------------------	----

<i>Changer l'icône d'un Mon Bloc</i>	65
Créer des «Mon Blocs» complexes	66
<i>Utiliser un Mon Bloc à l'intérieur d'autres Mon Blocs</i>	66
<i>Utiliser des paramètres d'entrée à ,plusieurs reprises à l'intérieur d'un Mon Bloc</i>	66
<i>Créer un Mon Bloc avec un port de sortie de données</i>	67
<i>Utiliser une variable locale dans un Mon Bloc</i>	68
<i>Tester un Mon Bloc avant et après l'avoir créé</i>	69
Méthode pas-à-pas de création d'un «Mon Bloc» complexe	71
En guise de conclusion...	72
<i>Premier exemple: un compteur.</i>	73
<i>Deuxième exemple: la conduite d'un véhicule contrôlé par un capteur à ultrasons</i>	75
10 - Un grand classique Le «Suiveur de Ligne» (màj 1) 2	80
Un Suiveur de Ligne Simple	80
Un Suiveur de Ligne «Proportionnel»	82
<i>Auto-calibrage lumineux:</i>	82
<i>Le programme principal</i>	83
11 - La Palette personnalisée	86
Gestion de la palette personnalisée	87
Les «Mon Bloc» rompus	87
Création de Palette secondaires	88
12 - Les nouvelles fonctions	90
Créer un fichier compressé	90
Editeur d'images...	92
<i>Création d'une image libre</i>	92
<i>Importer une image</i>	95
Editeur de sons...	97
<i>Importer un fichier son</i>	97
<i>Enregistrer un fichier son à l'aide d'un microphone</i>	100
<i>Modifier un fichier son</i>	101
Commande à distance...	102
<i>Connexion entre le NXT et l'ordinateur</i>	102
<i>La Commande à distance</i>	104
<i>Configuration des moteurs</i>	105
<i>Utilisation en divers modes</i>	105
<i>Raccourci clavier</i>	105
Commande à distance par appareils mobiles (màj 1) 3	106
<i>Connexion avec un Smartphone</i>	106
<i>A partir d'un Android Phone</i>	106
<i>A partir d'un iPhone</i>	107
13 - L'enregistrement des données (màj 1) 4	109
Rappels:	109
<i>Quid un enregistreur de données?</i>	110
Débuter avec un enregistreur de données	110
Améliorer un simple enregistreur de données	113

Représentation graphique	114
Exemple avec un moteur	117
<i>Ecriture du programme:</i>	117
<i>Exécution du programme</i>	118
<i>Enregistrement des données (màj 1)</i>	118
<i>Autres points particuliers</i>	121
Conclusions	123
Mises-à-jour	124
Mise-à-jour n°1 - Mars 2011	124
Remerciements	125



1 – Programmation NXT-G version 2.0

Introduction

J'ai fait l'acquisition du nouveau set, après quelques hésitations et interrogations, et je me suis posé la question de savoir si cette version condamnait tout le travail accompli avec l'ancienne. Cette question souvent abordée par les utilisateurs du NXT risquait de compromettre leurs travaux aussi bien ceux du passé que ceux à venir

Pour éviter toute omission, je me suis interrogé sur la compatibilité entre ces deux versions, et il m'a semblé normal de m'assurer également de celle des matériels.

Toutes ces questions, je les avais en tête avant l'acquisition mais comme inconditionnel, j'ai fait confiance aux spécialistes de Lego © qui eux y avaient déjà réfléchi. Pas de doute, il y a continuité et compatibilité dans les deux versions.

Ce guide précise donc en plus des nouveautés liées à la programmation, comment franchir le pas et comment faire évoluer votre ancien kit #8527 vers le kit #8547.

Commençons donc par le début.

La programmation avec la version 2.0 est en tous points identique à la version précédente 1.0 et 1.1. Donc, quel que soit votre niveau de connaissances, je vous invite à reprendre le Guide NXT-G version 1 dans lequel vous trouverez toutes les informations de base.

Nous verrons par la suite en quoi consistent les différences qui dans l'ensemble apportent quelques améliorations et fonctions complémentaires. En effet, pas de révolution sauf une très nette amélioration du fonctionnement du logiciel NXT-G.



* Vous venez d'acquérir le Kit # 8547 version. 2.0 et vous découvrez la programmation des robots Mindstorms. Commencez par vous familiariser avec la programmation en vous référant au Guide NXT-G version 1 téléchargeable gratuitement.

* Vous disposez déjà du Kit # 8527 version 1.0 ou 1.1 et vous venez d'acquérir le Kit # 8547 version. 2.0. Vos connaissances sont acquises et le passage de l'une à l'autre version ne pose aucun problème. Mieux, rien n'est

perdu et tous vos anciens fichiers sont récupérables et utilisables dans les 2 versions en prenant quelques précautions.

Avant d'aller plus loin je précise les conventions que j'ai adoptées pour définir simplement les différentes versions:

kit # 8527 version 1: NXT1 Logiciel NXT-G1 (Microprogramme du NXT1: version 1.05)

Kit # 8547 version 2: NXT2 Logiciel NXT-G2 (Microprogramme du NXT2 : version 1.28)

Il faut aussi rappeler que Les logiciels NXT-G sont déjà installés sur vos ordinateurs ainsi que les microprogrammes sur les briques NXT.

Quelques mots sur la programmation

Est-ce compliqué de programmer à l'aide du logiciel NXT-G?

Pour des actions simples et limitées, la programmation en NXT-G vers 1.0, 1.1 ou 2.0 est en effet facile. Le logiciel a d'ailleurs été conçu sur une plateforme graphique faisant appel beaucoup plus au raisonnement logique et à la réflexion plutôt qu'à la connaissance d'un langage codé. Manipuler des objets sous forme de pictogrammes tels que les blocs, s'inscrit dans la «philosophie» du groupe Lego: celle de manipuler des objets virtuels comme les pièces composant les robots. C'est la volonté des concepteurs du système Mindstorms® NXT®.

De plus, ces blocs disposent de valeurs de défaut pour le paramétrage qui suffisent dans la plupart des cas; ne l'oublions pas qu'il s'agit aussi d'intéresser les enfants.

Mais, pour des actions combinées et ou enchaînées, c'est un peu plus délicat: les paramétrages, les boucles ou les commutateurs nous réservent parfois des surprises. Et si on ajoute des actions simultanées, c'est encore plus incertain, voire inattendu. Mais, tout le plaisir est là...

A l'intelligence de la brique NXT s'ajoute celle du servomoteur qui intègre son capteur tachymètre. Le traitement des informations fournies combinées à celles des autres capteurs offre d'immenses possibilités.

Les blocs de programmation sont en fait des routines, c'est à dire des petits modules réemployables. Pour les utilisateurs avertis, il est aisé d'ajouter des routines personnelles aux palettes existantes. La communauté des fans offre une grande quantité d'astuces et d'expériences profitables mises gracieusement à la disposition de tous. C'est en puisant dans ce réservoir que vous perfectionnerez votre savoir. Il existe par exemple un bloc de programmation intitulé "suiveur de ligne" conçu par un passionné qui remplace l'écriture d'un programme. Ce bloc peut être téléchargé et ajouté à la palette existante. C'est aussi le cas des moteurs PFS contrôlés par la brique NXT et un capteur InfraRouge.



2 – Comparaison des kits

Examinons les boîtes des 2 versions
Vous aurez déjà remarqué que les couleurs de fond sont très nettement différenciées: l'orange pour la version 1 et le bleu pour la version 2.

Et maintenant, le contenu des boîtes:
Pour le NXT1: 577 pièces
Pour le NXT2: 619 pièces

La boîte du NXT2 est plus grande que celle du NXT1.



La brique NXT: Dans les 2 cas, elle est identique. Les caractéristiques techniques sont les mêmes.

Les différences

Dans la version2, le capteur sonore a disparu. Il est remplacé par un 2ème capteur tactile. Les créateurs du kit ont pensé que le capteur sonore manquait d'intérêt, du moins pour les nouveaux robots proposés dans cette version.

Pour ma part, je pense que le capteur sonore est utile pour déclencher des actions. Par exemple pour faire démarrer un robot, alors qu'il est en attente. C'est très utile quand on

tient un APN ou une caméra. Il suffit de lancer un ordre vocal puissant. Cet exemple est valable surtout pour activer des actions simultanées de plusieurs robots. Il peut servir de déclencheur à l'occasion d'une sonnerie de portable, ou d'arrêt.

On peut imaginer un robot sensible au bruit n'intervenant que lorsque le niveau critique est atteint, etc...

Les puristes, pourraient regretter l'absence de ce capteur peut être à leurs yeux indispensable. Par contre la présence de 2 capteurs tactiles offre des nouvelles perspectives par combinaison des deux.

Le capteur photosensible monochrome est remplacé par un capteur de couleurs. Cette nouveauté élargit considérablement les performances et ouvre la voie à des choix multiples selon les couleurs

De plus, ce dernier assume 3 fonctions différentes:

- * Il distingue les couleurs: il est capable d'identifier 6 couleurs.
- * Il distingue la lumière de l'obscurité (fonction monochrome assumée par l'ancien capteur photosensible).
- * Il sert également de signal lumineux (3 couleurs différentes RVB), il joue le rôle de lampe de signalisation colorées à choix multiple.

A noter que ce capteur ne peut être acquis séparément sur le site de Lego® contrairement au capteur sonore, et qu'il représente la particularité de ce Kit.

Ce qui n'a pas changé

Dans l'ensemble toute la partie «robotique» conserve ses caractéristiques, ce qui ne désavantage aucune version. On retrouve cette volonté de maintenir une certaine continuité, qui permet, nous le verrons plus loin, une mise à niveau sans dépenses supplémentaire, sauf à acquérir de nouveaux capteurs.

La brique NXT

La brique est identique

Les capteurs

Le capteur de contact et le capteur à ultrasons sont identiques

Le capteur US, son cône de réaction est plutôt réduit; je dirais pour simplifier qu'il n'agit que dans une direction et détecte les objets perpendiculaires à cette direction. Si on souhaite percevoir un obstacle par terre et en l'air, en même temps, il faudrait 2 capteurs US, placés judicieusement pour éviter les interférences. Ou alors l'installer sur un support mobile (actionné par un moteur) qui balaie l'espace dans les 2 directions (réalisable, mais pas facile).

Le positionnement de ce capteur est important pour la précision.

Les servomoteurs

Les 3 servomoteurs sont identiques.

Les câbles de liaison

identiques en caractéristiques et en nombre.

Les pièces Technic

Les roues dentées

Elles sont très réduites pour le NXT2; on ne trouve pas de «turntable», ni de 40T (T pour Teeth-dents) ni de 24T, ni de 16T, ni de 8T. Pas de poulies 24. Seulement 2 de 36T conique, 2 de 20T conique et 2 de 12T conique. Enfin il n'y a pas de vis sans fin (worm). LEGO a privilégié l'utilisation du servomoteur en vitesse variable par programmation du capteur intégré. Ici, la priorité n'est pas donnée à la mécanique pure, mais plutôt à la programmation. Les renvois d'angles sont réalisés de préférence à l'aide de 'knobs', ce qui produit forcément quelques frustrations auprès des fans de boîtes de transmissions complexes.

Les roues

On retrouve des jantes identiques mais avec des pneus plats, diamètre réduit 4,2 cm environ pour le NXT2. Deux chenilles souples et crantées sont ajoutées.

Les poutres studless

Elles sont plus nombreuses. On remarquera peu de poutres de 15S (studs) mais un plus grand nombre de 13S, avec une répartition différente pour les autres dimensions.

Les pièces spéciales

Six éléments d'habillage, permettant un carénage plus soigné, ainsi que 3 éléments pour réaliser un «panier» destiné à canaliser les projectiles d'un lanceur de billes.

Remarques

1. LEGO n'a pas maintenu la vente des deux versions. La version ancienne ne peut donc s'acquérir qu'auprès des vendeurs la disposant en stock, ou sur des sites de vente par Internet.

Si j'avais un conseil à donner pour l'achat d'une version, je pencherais pour l'achat en premier de la version 2.0, complétée ensuite (pour ceux qui en auraient les moyens) par le kit de la version 1. C'est-à-dire un meilleur panachage de capteurs en plus d'un bon assortiment de pièces studless et autres.

C'est comme par hasard celle qui s'impose pour ceux qui découvrent le Mindstorms NXT®.

2. D'autre part, le site de LEGO propose un inventaire des pièces constituant les kits NXT 1 et NXT 2. Cette liste reprend aussi le kit "Education" pour avoir un ensemble très complet.

http://static.robotclub.ab.ca/pages/nxt/InventoryComparison/nxt_1_vs_2_vs_edu.html

Si vous possédez un # 8527, vous pouvez constituer par vous-même (par achat des pièces manquantes), le #8547 avec toutefois une réserve: Le capteur couleur ne peut pas être acquis séparément sur le site LEGO.

Reste bien entendu la solution HiTechnic qui propose depuis quelques temps déjà un capteur considéré plus performant, mais d'un prix conséquent.

Une étude très intéressante a été faite sur cette question par Philippe Hurbain.

<http://philohome.com/colcomp/cc.htm>



3 – L'ordinateur et le logiciel NXT-G2.

Configuration requise

Avant d'installer le logiciel NXT-G2, assurez-vous que votre ordinateur possède la configuration requise:

Configuration PC:

Processeur Intel® Pentium® ou compatible: 1,5 GHz ou plus rapide recommandé.

Windows XP Edition familiale ou professionnel avec Service Pack 2 ou ultérieur.

Windows Vista Service Pack 1 ou ultérieur.

Lecteur CD-ROM

512 Mo de RAM minimum

700 Mo d'espace disponible sur le disque dur

Affichage XGA (1024 x 768)

Un port USB disponible

Un dispositif Bluetooth ou un adaptateur compatible

Configuration Apple Macintosh:

Processeur Power PC® G3, G4, G5 ou Processeur Intel, 1,3 GHz ou plus rapide recommandé.

Apple Mac OS X v10.4, V10.5 ou V10.6 (Tiger, Leopard, Snow Leopard)

Lecteur DVD

512 Mo de RAM minimum

700 Mo d'espace disponible sur le disque dur

Affichage XGA (1024 x 768)

Un port USB disponible

Un dispositif Bluetooth ou un adaptateur compatible.

Installation du logiciel

Remarque: Il est recommandé de supprimer l'ancienne version 1.0 ou 1.1 avant d'installer la version 2.0.

Fermez tous les programmes ouverts

Insérez le CD-ROM dans le lecteur

Suivez les instructions qui s'affichent à l'écran.

Sélectionnez la langue désirée

Laissez le programme terminer l'installation.

Au terme de l'installation, cliquer sur «Terminer».

Vous pouvez à présent utiliser le logiciel NXT-G2.

Nouveautés

Dès l'ouverture nous constatons un petit changement dans la présentation de l'espace de travail. La couleur du fond est différente. C'est à peu près la seule différence visible.

Mais à l'usage, que constate-t-on de différent par rapport à l'ancienne version?

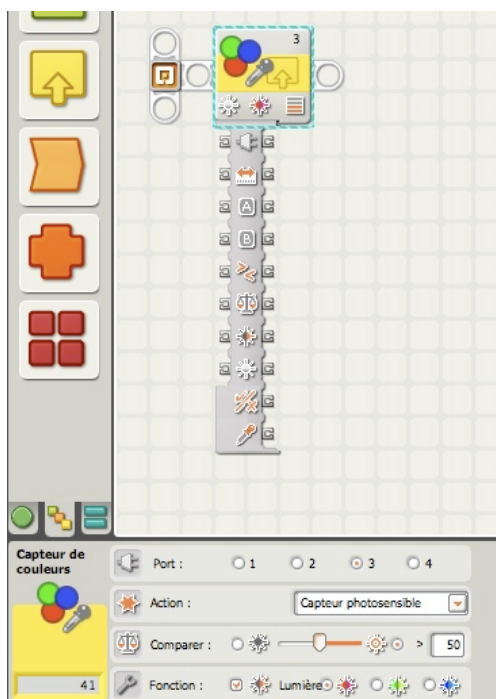
- Le programme tourne plus vite.
- Dans l'ensemble aucun changement sauf une refonte de certains blocs.
- Toutes les fonctions anciennes sont conservées et aucun bloc ancien n'est supprimé.
- Quelques nouveaux Blocs ont été ajoutés.

Ce qui a changé



Le nouveau Bloc «Capteur couleur» propose 3 fonctions distinctes:

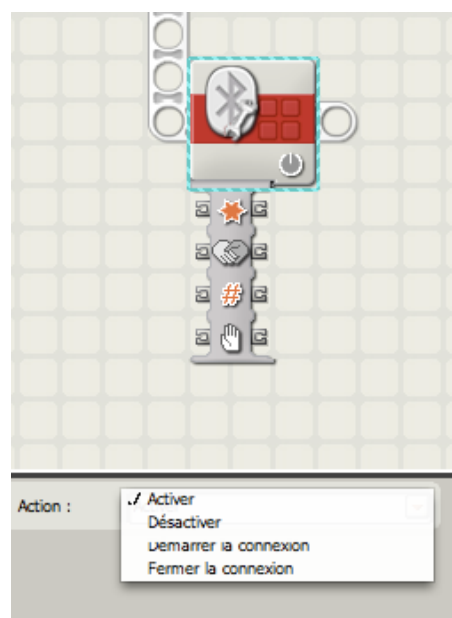
1. «Capteur de couleurs», capable d'identifier 6 couleurs,
2. «Capteur Photosensible», sur le rouge, capable de détecter des intensités lumineuses
3. Emetteur de lumière en 3 couleurs, rouge, vert et bleu, donnant plus de personnalité au robot.

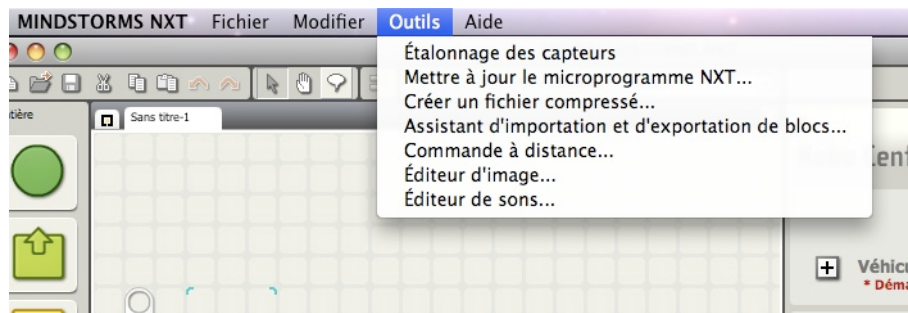


LE BLOC «Bluetooth Connexion» Qui sert à mettre en communication 2 ou plusieurs appareils sans passer par la brique NXT ni les manipulation par boutons.

Nouvelles fonctions:

Le menu "Outils" de la barre de menu offre de nouvelles possibilités:





Un éditeur de sons, utile pour fabriquer vos propres commandes dans la langue de votre choix, ou vos airs ou bruitages préférés.

Un éditeur d'images qui complétera la collection existante par vos images (photos, dessins...)

Une commande à distance directement à partir de votre ordinateur, et qui évite la télécommande à main.

Enfin, la création d'un fichier compressé que nous verrons plus loin.



4 – Installer le microprogramme sur le NXT2

Quelques rappels

Installer le logiciel NXT-G2 sur un ordinateur ne suffit pas.

Pour une première mise en route, il faut également installer un *microprogramme* sur la brique NXT.

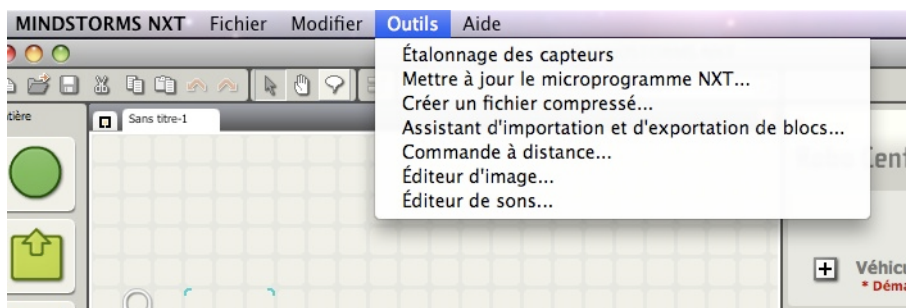
Le microprogramme

C'est un logiciel incorporé au NXT. Sans son microprogramme, le NXT est incapable de commander des moteurs, de recevoir les entrées des capteurs et, tout simplement, de fonctionner.



Il peut arriver, de temps à autre, que le Groupe LEGO® publie de nouvelles versions du microprogramme afin d'ajouter de nouvelles fonctionnalités ou de corriger des bogues. En mettant à jour le microprogramme de votre NXT, vous disposez des technologies les plus récentes et vous améliorez les performances et la fiabilité de votre NXT.

La dernière mise à jour concerne la version 1.29 pour PC et Mac. Elle peut se faire directement à partir du logiciel NXT2 menu «Outils» :



Ou bien par téléchargement gratuit sur le site Lego
<http://mindstorms.lego.com/en-us/support/files/default.aspx#Firmware>

Première installation

Vous n'avez pas de version NXT1 et venez d'acquérir votre Kit # 8545.

- 1) Insérer les piles ou la batterie dans votre NXT (piles rechargeables de préférence).
- 1) Allumez le NXT
- 2) Connectez le NXT à l'ordinateur à l'aide du câble USB (le NXT ne peut pas encore fonctionner sur bluetooth).
- 3) L'ordinateur identifie le NXT, il termine automatiquement l'installation du logiciel.

4) la version installée du microprogramme est la 1.28. Procéder par la suite à la mise à jour. Elle est expliquée plus bas ou dans le menu «Aide» du logiciel de programmation.

Dépannage

Si l'icône d'exécution cesse de tourner sur le petit écran, c'est que votre NXT est bloqué. Vous devez le réinitialiser.

Procédez ainsi:

1. Assurez-vous que le NXT est allumé.

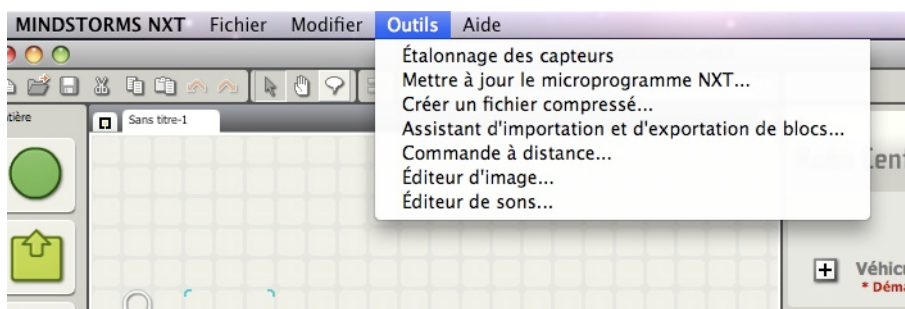
2. Appuyez sur le bouton de réinitialisation situé au dos du NXT, dans le trou LEGO® Technic situé dans le coin supérieur gauche. A l'aide d'un trombone déplié appuyez sur ce bouton.

3. Laisser ce bouton de réinitialisation appuyé pendant plus de 4 secondes.

A partir de ce moment, vous devez mettre à jour le microprogramme.

Mise à jour du microprogramme

Cette mise à jour se fait à partir de votre ordinateur et par le logiciel LEGO® Mindstorms® NXT. Le menu «Outils» permet de l'accomplir.

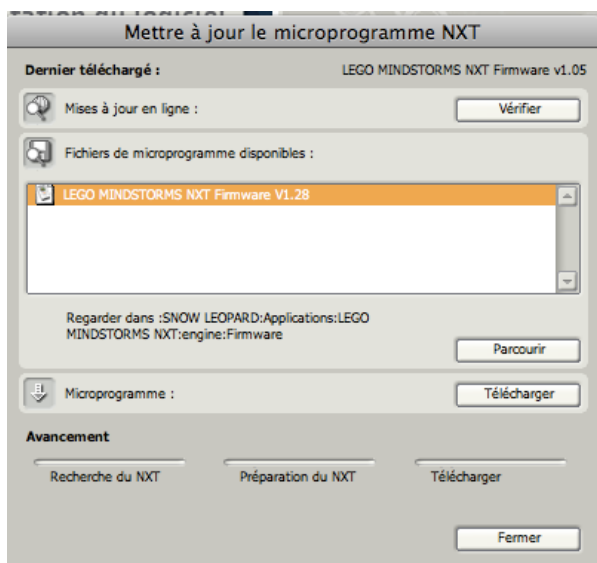


Rien n'est affiché sur l'écran du NXT après l'avoir allumé (juste un son de déclic indique la mise sous tension).

Assurez-vous que l'ordinateur et le NXT sont bien raccordés à l'aide du câble USB.

Choisir dans le menu «outils» > «Mettre à jour le programme NXT»,

Vous verrez apparaître la fenêtre suivante:



1. Cliquer sur le bouton «Vérifier» de la «Mise à jour en ligne». Vous êtes alors dirigé vers le site de Lego®Mindstorms® NXT.

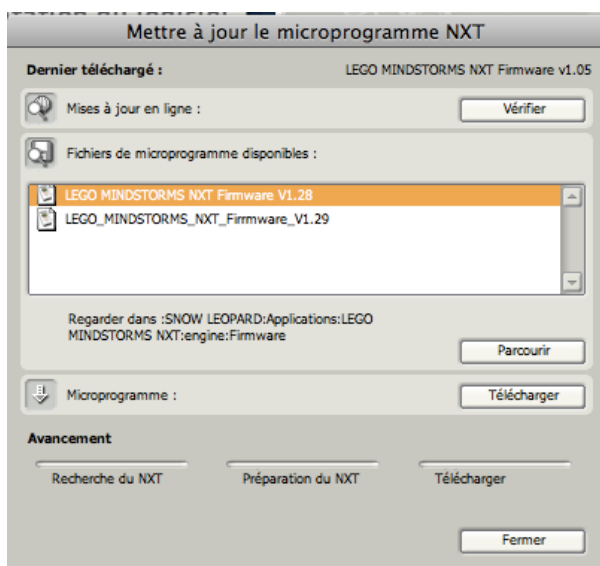
2. Sélectionnez le menu «Firmware et cliquez sur «Download. Le fichier est alors installé sur le bureau de votre PC.

3. Placez ce fichier

intitulé «LEGO_MINDSTORMS_NXT_Firmware_V1.29.rfw» dans le dossier \Program Files\LEGO Software\LEGO MINDSTORMS NXT\engine\Firmware,
Vous venez *seulement* d'installer dans votre ordinateur la dernière version du microprogramme.

4. Assurez-vous que votre NXT est bien connecté à votre PC et allumé, puis redémarrez le logiciel NXT-G.

5. Choisissez à nouveau dans le menu «*outils*» > «*Mettre à jour le programme NXT*», Dans la fenêtre «*fichiers de microprogramme disponibles*» vous verrez apparaître le nom de la dernière version.



Sélectionnez la nouvelle version puis «Cliquez sur «*télécharger*»

Dans la partie «*Avancement*» vous apercevrez la progression d'une barre de défilement qui contrôle le téléchargement en 3 étapes.

Appuyez sur «*Fermer*» une fois le travail terminé.

La nouvelle version du microprogramme est maintenant installé sur votre NXT.

Vous pouvez le vérifier en sélectionnant sur l'espace de travail l'affichage de la fenêtre du NXT.

Assistant d'importation et d'exportation de blocs...(màj 1) 1

Il existe un certain nombre de fabricants qui développent des capteurs et accessoires en complément de ceux fournis dans les sets LEGO #8527 ou #8547. On y trouve notamment des capteurs à Infra Rouge, gyroscopique, magnétique, optique, etc.. Des accessoires tels que multiplexeur sont également fabriqués pour accroître les capacités des capteurs d'origine et augmenter les possibilités de connexion.

Pour offrir à ces fabricants toutes sortes d'extensions possibles pouvant fonctionner dans la brique NXT, LEGO a prévu dans son logiciel NXT-G une fonction d'importation. Cette fonction est utilisable dans les 2 versions, 1.1 et 2.0.

Pour installer un capteur ne figurant pas dans le set ou un accessoire du type multiplexeur ou autre, il faut nécessairement:

1 - l'accessoire et ses câbles de liaison

2 - Le ou les blocs de programmation NXT-G conçu par le fabricant.

(Ces blocs sont généralement à télécharger sur le site du fabricant).

Ces blocs sont à installer dans la palette entière de votre programme NXT-G (choix "avancé"). Une fois installés, ils s'utilisent comme n'importe quel autre bloc.

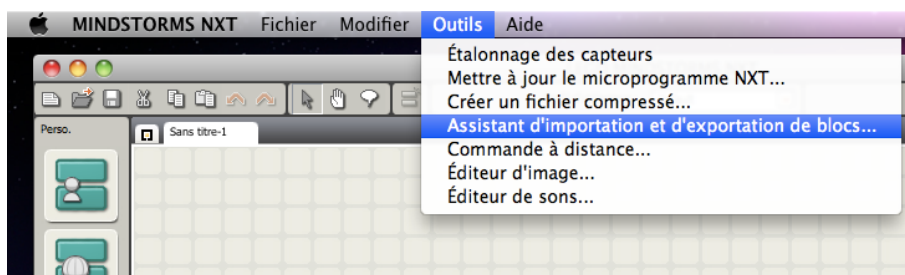
Installation d'accessoires additionnels dans la palette d'outils du NXT-G.

1 - Télécharger sur le site du fabricant le module compatible avec le logiciel NXT-G. L'installer sur le bureau de préférence.

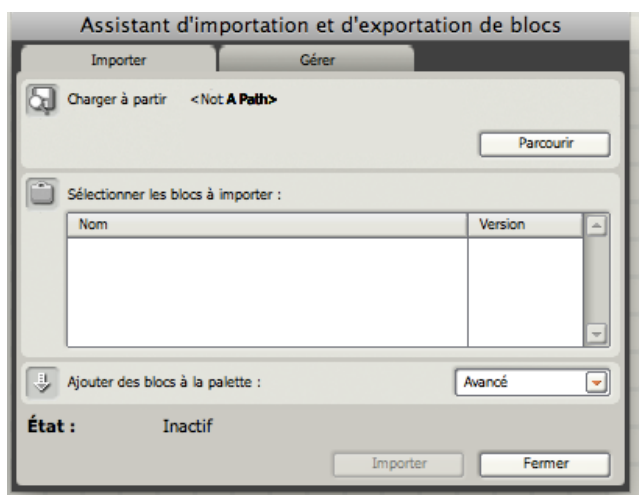
2 - Lancer le logiciel NXT-G et créer un nouveau programme.

3 - Sur la fenêtre de travail, choisir dans la barre de menus:

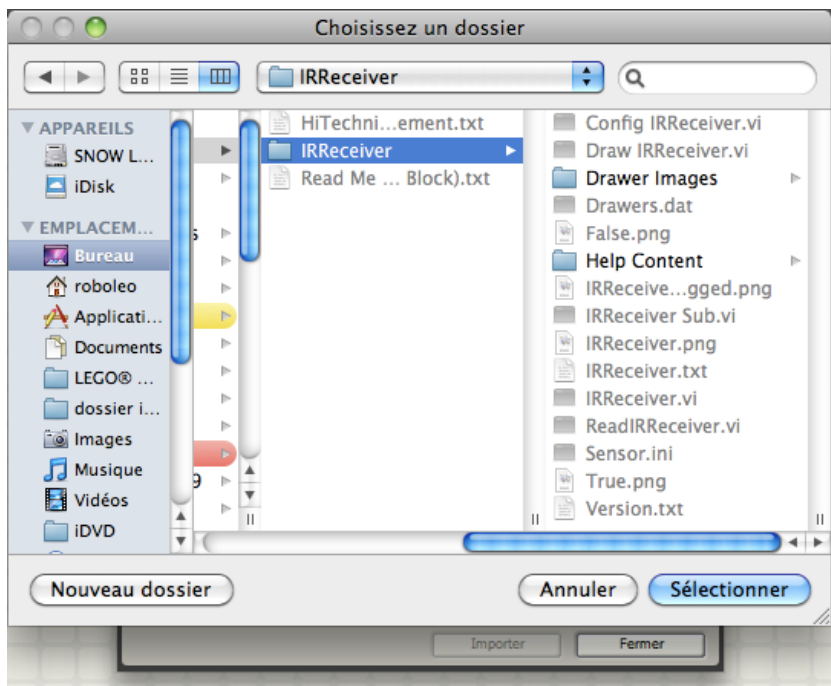
"Outils > Assistant d'importation et d'exportation de blocs..."



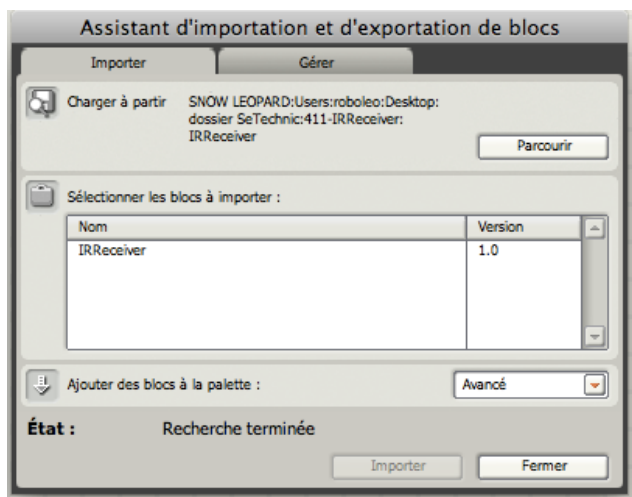
4 - La fenêtre suivante s'affiche:



5 - Cliquer sur le bouton "Parcourir"

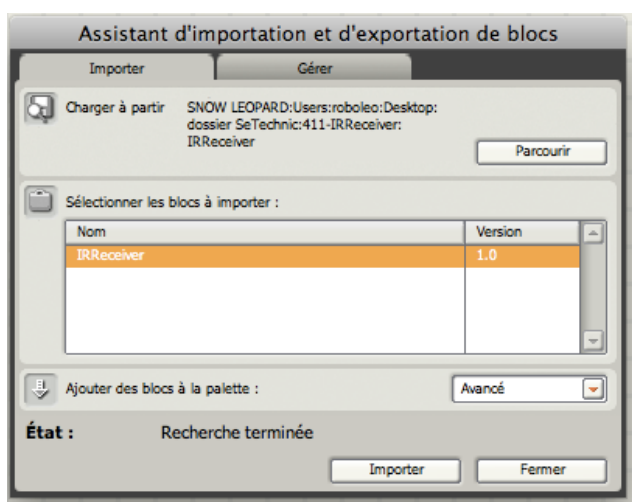


6 - Sélectionnez le dossier que vous avez téléchargé. Dans notre cas il s'agit du **DOSSIER IRReceiver**. Inutile de l'ouvrir. Cliquer sur le bouton "Sélectionner".

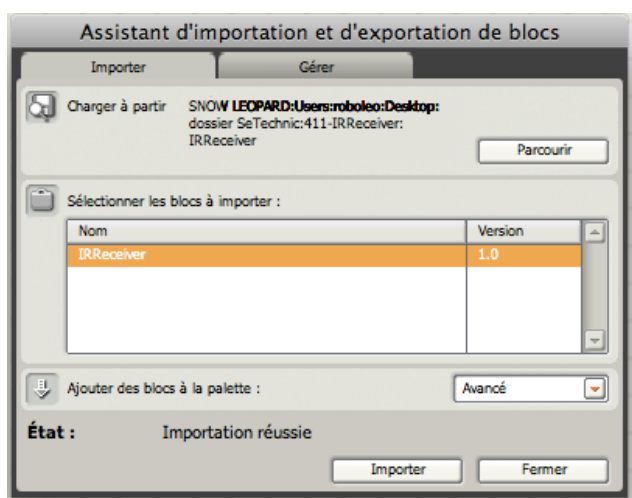


7 - Cette fenêtre s'affiche et le nom du dossier sélectionné apparaît. On peut lire dans la zone "Etat" > "Recherche terminée".

8 - Cliquer sur la ligne correspondant au bloc à importer, puis sur le bouton "Importer".



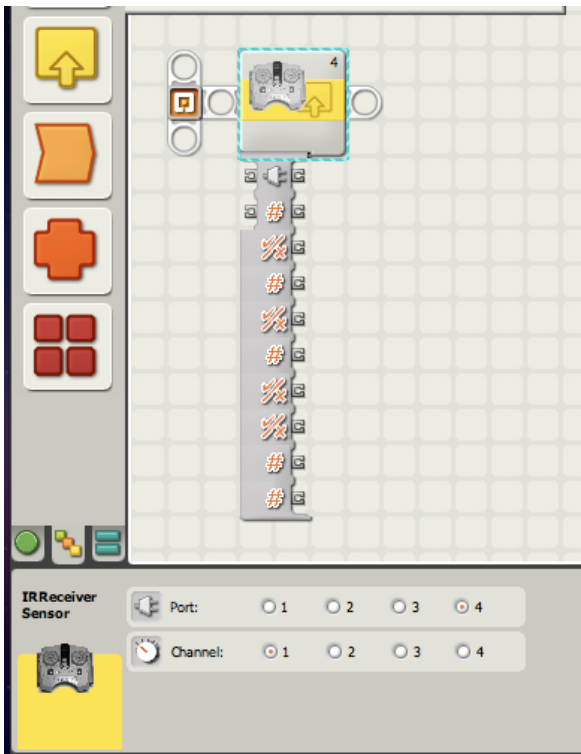
On peut maintenant lire dans la zone "Etat" > "Importation réussie".



L'importation est terminée et le bloc correspondant peut être utilisé dans le logiciel NXT-G. Le bloc **IRReceiver** est accessible dans la palette entière.



On peut donc l'utiliser comme n'importe quel autre bloc.





5 – Mise à jour de la brique NXT1

Vous avez supprimé l'ancienne version 1 et venez d'installer sur votre ordinateur le logiciel NXT-G2; tout s'est parfaitement déroulé.

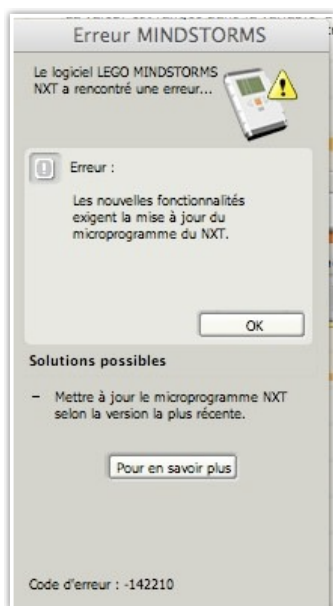
Vous souhaitez maintenant utiliser vos anciens programmes (vers 1.0 ou 1.1). Ouvrez un programme de votre choix et installez-le sur la fenêtre de travail.

Normalement il devrait apparaître sans difficulté. Vous avez à présent un programme version1 ouvert avec le logiciel version2. A peu de choses près il est identique à sa version d'origine.

Vous désirez à présent le télécharger comme d'habitude, sur le NXT.

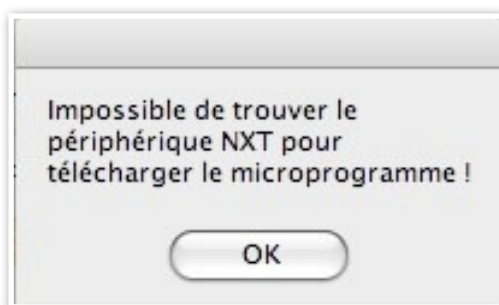
Allumez votre brique NXT puis connectez la à votre ordinateur, soit à l'aide du câble USB soit par Bluetooth.

Alors que vous tentez de le transférer par l'intermédiaire du Contrôleur, vous obtiendrez ce message d'Erreur



.Cela signifie qu'il faut mettre à jour sur votre brique NXT1 le microprogramme interne.

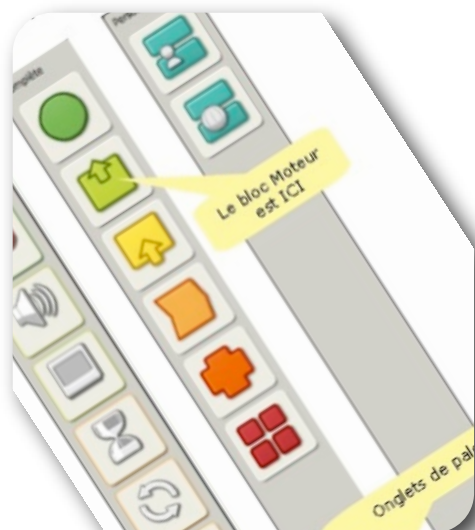
Pour ce faire, éteignez votre NXT1, puis raccordez la brique à votre ordinateur à l'aide d'un câble USB. Il n'est pas possible de télécharger ce microprogramme par Bluetooth, sinon le message suivant apparaît.



La mise à jour se fait alors normalement, à partir du logiciel NXT-G 2.0 comme une mise à jour classique.

Une fois le travail accompli, votre NXT1 s'est transformé en NXT2, il a été vidé de tous les anciens programmes NXT-G1, et il est prêt à recevoir les nouveaux programmes NXT-G2.

Recommandations: conservez vos anciens programmes NXT-G1 sur votre ordinateur, ils sont réutilisables.



Les palettes et les blocs

Rappel

Il est utile de rappeler l'usage des outils, et en particulier celui des «*Palettes*». Comme vous l'avez appris, Il en existe de trois types offrant des collections de blocs complémentaires. Elles sont accessibles en marge dans la bande verticale à gauche de l'espace de travail. Il suffit de sélectionner l'onglet correspondant.



Une première intitulée «*Palette commune*» permet la création rapide de programmes simples faisant appel à des blocs d'usage courant. C'est par définition la palette des débutants.

La deuxième palette est dite «*Palette entière*» parce qu'elle dispose plus de blocs que la palette commune et offre plus de possibilités; elle est donc à choisir pour la confection de programmes plus élaborés. L'onglet sélectionné dans la figure est celui de la palette entière.

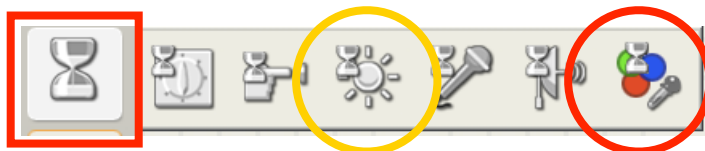
La dernière dite «*Palette personnalisée*» contient des blocs que vous avez créé et qui sont en fait des petits programmes dits '**Routine**' confectionnés à l'aide des blocs des autres palettes. Ces Routines sont réutilisables dans des programmes différents, particulièrement utiles pour des actions répétitives et raccourcissent considérablement la structure des programmes qui les utilisent. Cette palette est plutôt réservée à des programmeurs expérimentés.

Les nouveaux blocs

Dans la palette commune

Blocs «**Attendre**» on remarque la présence du bloc «**Capteur de couleurs**» avec son petit sablier.

Cette addition est normale puisque le capteur de couleurs est la nouveauté du kit. Nous verrons plus loin en quoi consistent ses caractéristiques.



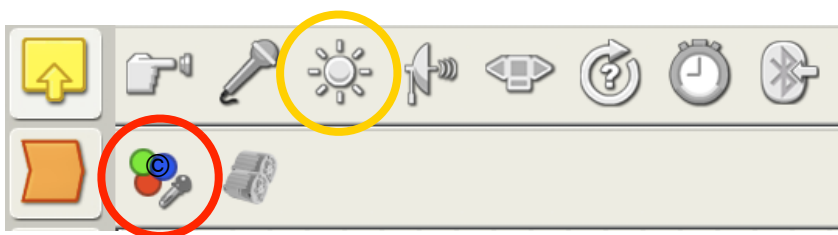
L'icône du bloc «**Capteur Photosensible**» est maintenue.

Dans la palette entière



Blocs «**Action**» on remarque là aussi la présence du bloc «**Lampe de couleurs**». A noter que les icônes sont différentes.

Cela sous-entend une nouvelle possibilité du capteur de couleurs. Il est en effet capable d'émettre des signaux lumineux de couleurs différentes.



Blocs «**Capteur**» : parmi les blocs existants, on observe l'ajout du bloc «**Capteur de couleurs**» qui est l'innovation de cette version 2.0. Le bloc «**Capteur Photosensible**» est maintenu.

Remarque: l'icône qui suit celle du capteur de couleurs est un ajout indépendant de la version originale. Elle provient d'un téléchargement pour des besoins particuliers.



Blocs «**Données**» : Un nouveau bloc «**Constante**» a été ajouté. Il ressemble fortement à un bloc «**Variable**» ; il se définit de

façon identique et conserve d'une manière définitive une valeur singulière (exemple $\text{Pi}=3,14116$, ou une expression comme OK).

Une constante ne change jamais avec le temps. Vous verrez qu'elle est très utile, par ce qu'elle évite de saisir à nouveau une même valeur ou texte, donc élimine les risques d'erreur de saisie.



6 – Compatibilité entre les 2 versions

Question 1: NXT2 exécute-t-il les programmes écrits avec NXT-G1?

Réponse: **OUI**

NXT2 "tourne" avec NXT-G1; il lit les programmes écrits dans la version ancienne, et les exécute, sous réserve d'examiner les

blocs signalés par "!"

Il communique en Bluetooth avec un ordinateur disposant de NXT-G1. Le téléchargement des programmes est possible.



Question 2: NXT1 exécute-t-il les programmes écrits ou lus avec NXT-G2?

Réponse: **NON**

Mais,

NXT1 signale la présence de nouvelles fonctionnalités et exige la Mise à Jour du nouveau microprogramme.

La mise à jour se fait à partir du NXT-G2.

Une fois cette mise à jour effectuée, Le NXT1 est vidé de tous ses anciens programmes et devient prêt à recevoir les

nouveaux programmes NXT-G2.

Le NXT1 est devenu un NXT2 et se comporte comme il est répondu à la question 1.

Les anciens programmes NXT-G1 archivés dans le disque dur de l'ordinateur sont utilisables sous réserve de quelques vérifications.

Cela signifie entre autre que vous devez conserver vos anciens programmes (mais cela est déjà archivé sur votre ordinateur), et au fur et à mesure de leur utilisation à l'aide du NXT-G2 ils sont actualisés. Par contre si vous souhaitez conserver les 2 versions et pour éviter toute confusion, il est souhaitable de les utiliser sur 2 ordinateurs différents.

Question 3: le logiciel NXT-G2 serait-il téléchargeable sans acheter le set NXT2.0 ?

Pour le téléchargement, je ne le pense pas, mais la vente du nouveau CD vers. 2.0 peut-être. La question devrait être posée à LEGO®.

CONCLUSION:

Le logiciel NXT2 est donc entièrement compatible avec NXT1.

Cela signifie que si vous achetez un NXT2, vous pouvez utiliser les programmes destinés à un NXT1 que vous trouverez sur divers sites, ou n'importe quels programmes que vous avez déjà écrits vous-même.

Cela signifie aussi que, si vous êtes l'heureux propriétaire d'un NXT1 et d'un NXT2, le logiciel NXT2 peut traiter la totalité (y compris le capteur son) et éventuellement d'autres actions non directement incluses dans NXT2.

Le logiciel NXT1 n'est généralement pas compatible avec des programmes écrits pour NXT2. Quelques programmes NXT2 très simples peuvent être chargés dans NXT1 puis compilés et téléchargés sur une brique NXT1, mais la plus part ne le pourront pas. Généralement, vous serez capables d'ouvrir et voir le contenu de programmes NXT2 sauvegardés dans le format .rbt normal uniquement en lecture, mais ne vous attendez pas à les utiliser.

NXT2 inclut aussi un nouveau format .rbtx sauvegardé grâce une nouvelle fonction "Pack-and-go" qui est très utile parce qu'elle inclut les «Mon Blocs», les fichiers images et sons (nécessaires aux programmes), les fichiers compressés. L'ensemble est ainsi nettement plus compact, mais ce format n'est absolument pas lisible par NXT1.

NXT1 et NXT2 peuvent communiquer entre eux par Bluetooth. Et si vous aviez vraiment quelque problème de compatibilité, il est facile de mettre à jour les deux briques en utilisant le microprogramme du NXT2, vers 1.29 . Il suffit de se connecter au logiciel NXT2 et télécharger le programme de mise à jour du microprogramme.

Enfin, si vous avez des kits de versions différentes, Il est préférable de mettre à jour toutes les briques et les ordinateurs au microprogramme NXT2 et au logiciel les plus récents, parce que les performances sont améliorées et la compatibilité fonctionne mieux dans ce sens (NXT2 peut lire les instructions du NXT1, mais non l'inverse).



7 – Le capteur qui prends des couleurs...

Introduction

Ce nouveau capteur dote les robots de la vue, tout comme le capteur photosensible de la version précédente, l'autre étant le capteur d'ultrasons. Mais ce qui fait son originalité c'est qu'en plus de la distinction de la lumière et l'obscurité, il permet aux robots de distinguer les couleurs.

Comme nous le verrons dans les blocs de programmation, il peut identifier 6 couleurs différentes ainsi que mesurer l'intensité d'un local et des surfaces colorées. En outre, sa conception dont on aperçoit les composants d'émission des 3 couleurs fondamentales (LED rouge, vert et bleu), permet de l'utiliser comme lampe de couleur.

Donc 3 usages possibles:

1. faire exécuter des tâches différentes selon la couleur détectée, comme par exemple un robot trieur d'objets colorés, ou réactif à une couleur spécifique comme faire changer de direction s'il détecte une tache verte. Il pourrait être utilisé pour diriger un robot vers un point donné en balisant à l'aide pastille colorées collées sur le sol du parcours. Par exemple, si c'est vert, aller tout droit; si c'est bleu, tourner à droite; si c'est rouge, tourner à gauche. On pourrait imaginer des tas de solutions sur cette idée
2. mesurer l'intensité lumineuse d'une lumière ambiante pour faire démarrer ou arrêter un robot (avancer dans la clarté et reculer dans l'obscurité), etc...
3. l'utiliser comme avertisseur de manoeuvre, grâce aux 3 couleurs de lampe: vert pour la marche avant, rouge pour la marche arrière et bleu pour une manutention.

Test des fonctions

Il est bon de s'assurer avant l'écriture du moindre programme que le capteur fonctionne correctement. Les tests peuvent-être réalisés de plusieurs manières:

Préparer le matériel:

Brancher un moteur sur le port B et le capteur sur le port 3

Allumer la brique NXT2.0

A) choisir le menu TRY de la brique:

1. test TRY COLOR

Diriger le capteur sur une surface claire

RUN

patienter un moment

le capteur lance la rotation du moteur à des vitesses variables.

Si le capteur est dirigé vers une surface sombre, le NXT émet un signal sonore et bloque le moteur.

Revenir vers une surface claire et patienter un moment pour relancer la rotation du moteur.

2. test TRY-LIGHT

La brique émet un bip sonore relativement faible associé à un son de fréquence aigüe.

B) choisir le menu VIEW de la brique, puis REFLECTED LIGHT > PORT 3

Affichage sur l'écran d'une fenêtre rectangulaire angles arrondis avec valeur 100%.

Le capteur est éteint et reste éteint.

Même manipulation avec le capteur allumé (voir plus loin comment procéder); même résultat.

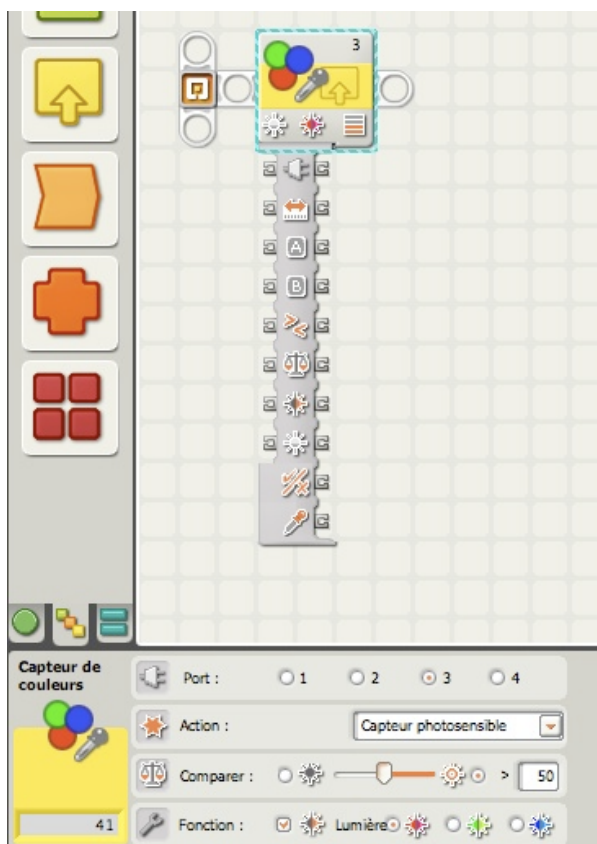
Il semblerait que ce menu ne fonctionne pas. L'observation d'un utilisateur (le bug viendrait du programme VIEW en lui même) est confirmée.

C) Test avec un programme élémentaire:

Connecter l'ordinateur à la brique (USB ou Bluetooth)

Créer un programme NXT-G ne comportant qu'un bloc CAPTEUR DE COULEURS

Paramétrer selon le tableau de configuration:



Observer la ligne Fonction. La case lumière est cochée; le capteur s'allume en couleur rouge (couleur de l'ancien capteur photosensible). Cette couleur est activée par le bouton de sélection devant l'icône correspondante.

Diriger le capteur vers une surface de référence (j'ai choisi une couleur claire vers le blanc).

On lit la valeur 41 dans la fenêtre sous l'icône. Cette valeur correspond au point de déclenchement qui peut remplacer la valeur 50 dans la ligne Comparer.

Maintenant, cliquer sur le bouton suivant *vert*. La lumière émise change de couleur et passe au vert. La valeur lue est dans ce cas pour le test 37.

Même manipulation avec le bouton suivant, et la couleur passe au bleu, avec une valeur de 33.

Conclusion:

Le capteur fonctionne avec 3 couleurs différentes que l'on peut choisir selon les circonstances.

Le capteur, en mode photosensible, fonctionne correctement puisqu'il fournit des valeurs différentes pour chaque couleur. De plus, pour une même couleur, en l'éloignant de la

surface, on obtient des valeurs différentes affichées dans la partie jaune du panneau de configuration.

C'est la preuve du bon fonctionnement du capteur.

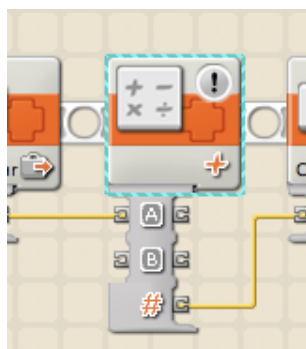


8 – Les Nouveaux blocs

Les blocs bizarres

Commençons par une particularité.

A l'ouverture d'un programme v1.0 ou v1.1 à l'aide du logiciel NXT-G2, vous constaterez que quelques blocs (comme le bloc mathématique ci-joint) présentent un (!)



sur l'icône. Cette indication signifie que les blocs concernés sont des versions «integer», c'est-à-dire des blocs qui ne prennent en compte que des valeurs numériques entières. Ces blocs ont la particularité de n'accepter aucune valeur décimale (on dit aussi virgule flottante). Et si vous tentez de taper des valeurs décimales dans les zones de saisie, vous constaterez que ces valeurs sont automatiquement arrondies en valeurs entières les plus proches. N'ayez aucune crainte, ils fonctionnent parfaitement avec la v2.0. Vous pourriez penser que nous n'avons pas besoin de procéder à des calculs faisant intervenir la virgule flottante. Détrompez-vous. Dans certains cas vous remarquerez que cet avantage simplifie la conception des programmes, supprime des calculs intermédiaires

et les allègent. C'est un avantage incontestable, et lorsque vous aurez acquis une bonne maîtrise de la programmation, vous en mesurerez les avantages.

Cette indication signifie aussi que les blocs ont été mis à jour dans la version v2.0 pour être transformés en blocs à virgule flottante. Mais vous remarquerez que ces blocs n'apparaissent sur aucune palette de la version v2.0.

Toutefois, si pour des raisons personnelles vous tenez à les voir figurer il faudra les ajouter à la palette. Pour le faire, créer simplement un fichier vide dans le répertoire BlockRegistry, situé dans le sous-répertoire de palette, avec le nom du bloc et .txt comme suffixe d'extension.

Ex : [LEGO MINDSTORMS directory] \engine\EditorVIs\BlockRegistry\Data\numeric Operations.txt

La liste de blocs disponibles est placée dans : [LEGO MINDSTORMS directory] \engine \vi.lib\LEGO\Blocks\

Si cela ne présente pour vous aucun intérêt, je vous conseille d'évitez de le faire.

Le bloc «Capteur de couleur»

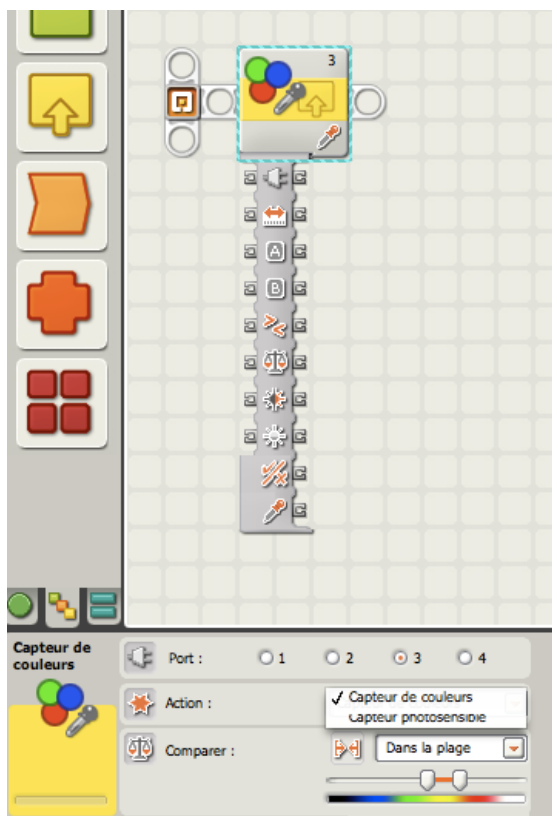
C'est la grande nouveauté de cette version.

Ce bloc «**Capteur de couleur**» situé dans la palette Entière dispose de deux modes : un mode de *détection des couleurs* et un mode de *mesure de l'intensité lumineuse*. A ce bloc sont associés des plots de données que l'on ne trouve pas en utilisant la palette Commune.

Remarque: ces deux modes s'utilisent avec un bloc «Attendre», un bloc «Commutation», ou un bloc «Boucle» pour générer un signal logique vrai/faux.

1.Mode de détection de couleurs

Dans la zone «Action», vous avez sélectionné «Capteur de couleurs».



Le nombre situé à l'angle haut droite indique le n° du port auquel votre NXT est connecté au capteur de couleur. Vous pouvez choisir un autre port dans le panneau de configuration.

L'icône en bas à droite indique le mode 'Capteur de couleur'. Dans la zone Action le menu déroulant permet de choisir le mode: ici Capteur de couleurs.

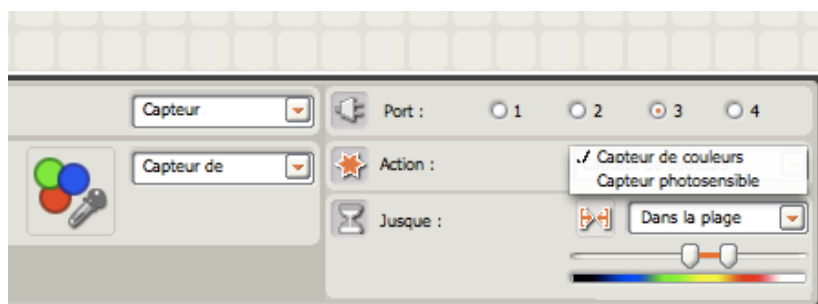
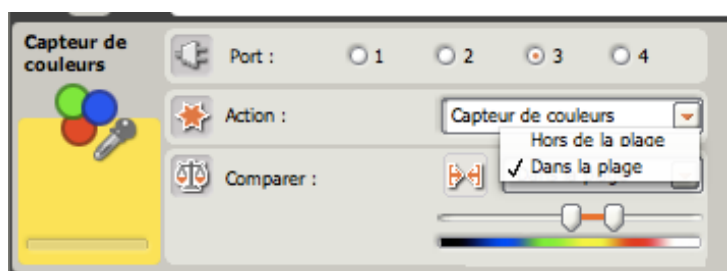
Le plot de données du bloc s'ouvre quand le bloc est placé dans l'espace de travail. Il permet de tirer des fils de données de la prise de sortie du bloc vers le plot de données d'un autre bloc. Pour plus d'informations, consultez le tableau ci-dessous.

Dans le menu déroulant de la zone «Comparer», choisissez 'Dans la plage' ou 'Hors de la plage'.

Choisissez 'Dans la plage' pour que le bloc s'active lorsque les couleurs sont dans la plage.

Choisissez 'Hors de la plage' pour que le bloc s'active lorsque les couleurs sont en dehors de la plage.

Utilisez les glissières gauche et droite pour délimiter la plage de couleurs définissant la valeur de déclenchement : noir, bleu, vert, jaune, rouge et blanc (vous avez 6 couleurs à votre disposition).



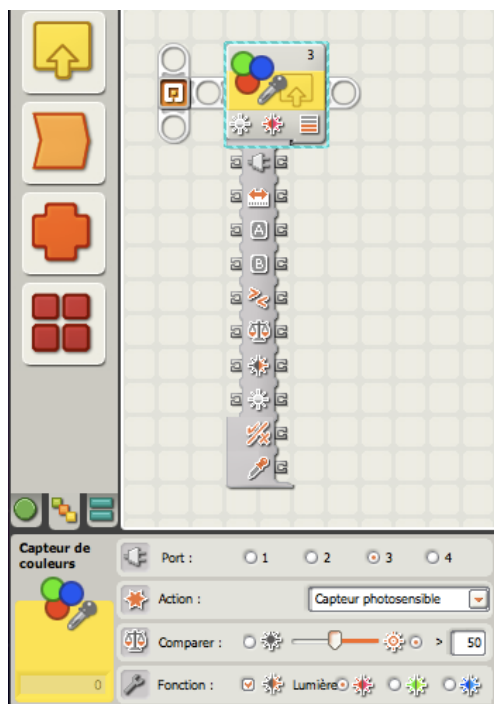
Choisissez 'Dans la plage' dans le menu déroulant du panneau de configuration pour générer un signal « vrai » ou 'Hors de la plage' pour générer un signal

« faux ».

Le réglage par défaut du bloc «**Capteur de couleurs**» est défini sur '*Dans la plage*' et pour détecter le *jaune*. Le signal « *vrai* » sera *jaune* et le signal « *faux* » sera d'une autre couleur.

Lorsque vous êtes en mode Capteur de couleur, la zone de rétroaction du panneau de configuration indique la couleur actuellement détectée. Pour recevoir une rétroaction, assurez-vous que le capteur photosensible est connecté au port approprié et que la communication a été établie avec le NXT.

2.Mode de mesure de l'intensité lumineuse



Dans la zone «Action», vous avez sélectionné «*Capteur photosensible*».

Le nombre dans le coin supérieur droit du bloc Capteur de couleurs indique quel port du NXT contrôlera le bloc. Si vous le souhaitez, vous pouvez choisir un autre port dans le panneau de configuration.

L'icône en bas à droite indique le mode «*Capteur photosensible*» et la couleur choisie dans la barre des fonctions : rouge, vert ou bleu. Cette icône indique à quel niveau le point de déclenchement est défini.

Le plot de données du bloc s'ouvre quand le bloc est placé dans l'espace de travail. Il permet de tirer un fil de données de la prise de sortie du bloc vers le plot de données d'un autre bloc. Pour plus d'informations, voir le tableau ci-dessous.

Vous pouvez définir le point de déclenchement à l'aide de la glissière ou en entrant un nombre dans la zone de saisie. Pour définir la plage (au-dessus ou en dessous du point de déclenchement) qui générera le signal « *vrai* », utilisez les cases d'option.

Par défaut, le bloc «*Capteur photosensible*» est paramétré pour qu'une luminosité supérieure à 50% génère un signal « *vrai* ». La case d'option à droite de la plage est activée et la glissière est réglée sur 50. Pour activer les parties « vraies » de la plage (définir les valeurs sous 50% comme étant « vraies »), activez la case d'option de gauche.

Lorsque vous êtes en mode «*Capteur photosensible*», la zone de rétroaction du panneau de configuration indique la valeur de luminosité (0-100) mesurée par le capteur. Pour recevoir une rétroaction, assurez-vous que le capteur photosensible est connecté au port approprié et que la communication a été établie avec le NXT.

caractéristiques des prises du plot de données du bloc Capteur de couleurs.

(Extrait de l'aide et support pour Lego® Mindstorms® NXT)

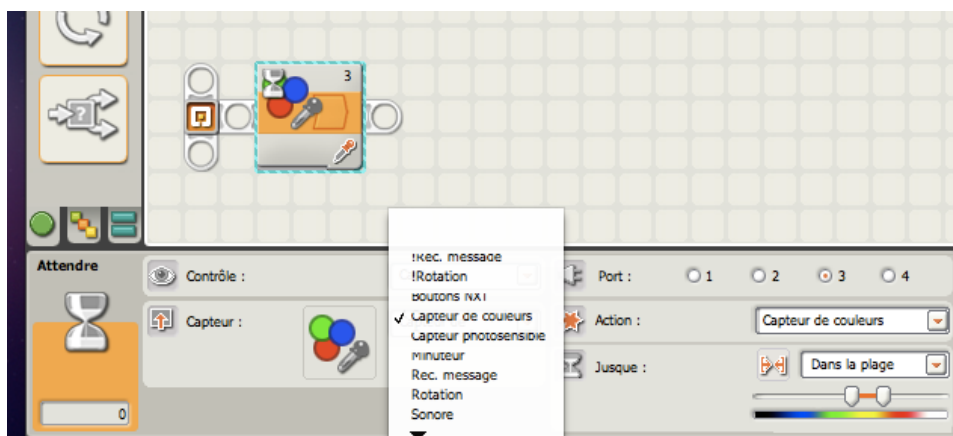
	Prise	Type de données	Plage admise	Signification des valeurs	Prise ignorée quand...
	Port	Numérique	1 - 4	Correspond au port d'entrée numéroté sur le NXT.	Jamais ignorée
	Plage	Logique	Vrai/Faux	Vrai = Dans la plage Faux = En dehors de la plage	En mode Capteur photosensible
	Plage de couleurs A	Numérique	0 - 6	0 = A gauche de Noir 1 = Entre Noir et Bleu 2 = Entre Bleu et Vert 3 = Entre Vert et Jaune 4 = Entre Jaune et Rouge 5 = Entre Rouge et Blanc 6 = A droit de Blanc	En mode Capteur photosensible
	Plage de couleurs B	Numérique	0 - 6	0 = A gauche de Noir 1 = Entre Noir et Bleu 2 = Entre Bleu et Vert 3 = Entre Vert et Jaune 4 = Entre Jaune et Rouge 5 = Entre Rouge et Blanc 6 = A droit de Blanc	En mode Capteur photosensible
	Supérieur/Inférieur	Logique	Vrai/Faux	Logique utilisée dans la comparaison : Vrai = Supérieur, Faux = Inférieur	En mode Capteur de couleur
	Point de déclenchement	Numérique	0 - 100	Valeur de référence de la comparaison	En mode Capteur de couleur
	Générer de la lumière	Logique	Vrai/Faux	Détermine si la LED du capteur est allumée ou éteinte.	En mode Capteur de couleur
	Couleur de lampe	Numérique	0-2	0 = Rouge 1 = Vert 2 = Bleu	En mode Capteur de couleur
	Oui / Non	Logique	Vrai/Faux	Résultat de la comparaison	Jamais ignorée

	Couleur détectée	Numérique	1-6	1 = Noir 2 = Bleu 3 = Vert 4 = Jaune 5 = Rouge 6 = Blanc	En mode Capteur photosensible
---	------------------	-----------	-----	---	-------------------------------------

Le bloc «Attendre»

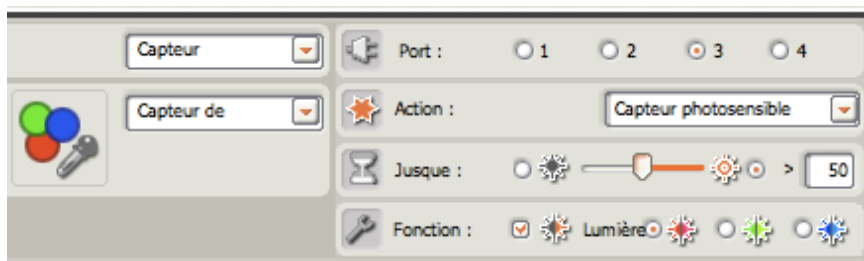
Disponible dans la palette commune ou entière, c'est une conséquence logique de cette nouveauté.

Dans la zone «capteur» le menu déroulant offre un capteur supplémentaire: le Capteur de couleurs. On remarque que le capteur photosensible figure toujours dans la liste, ce qui laisse supposer 2 choses: l'utilisation maintenue du capteur photosensible (si vous en possédez un) et la possibilité d'utiliser le capteur de couleurs en capteur photosensible.



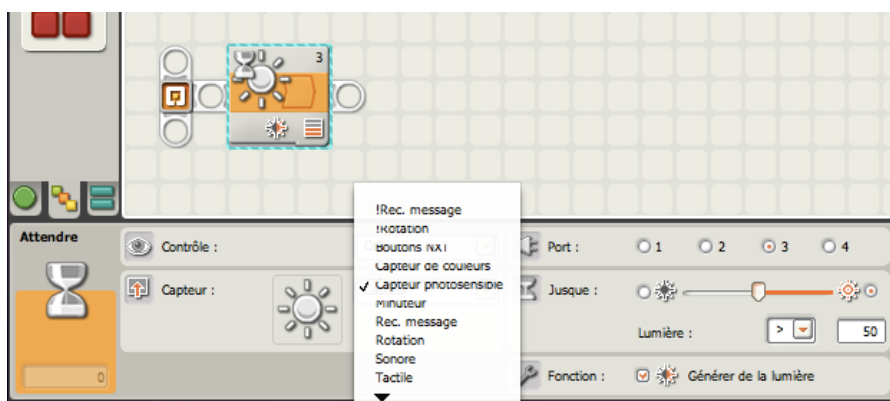
on notera dans la zone «action» la possibilité de ce choix.

* Si le capteur de couleurs est sélectionné, la condition d'attente dépend de la couleur à détecter. Il suffit d'utiliser les curseurs de réglage de la page, situés



au dessus du spectre de couleurs. Il est capable d'identifier jusqu'à 6 couleurs différentes.

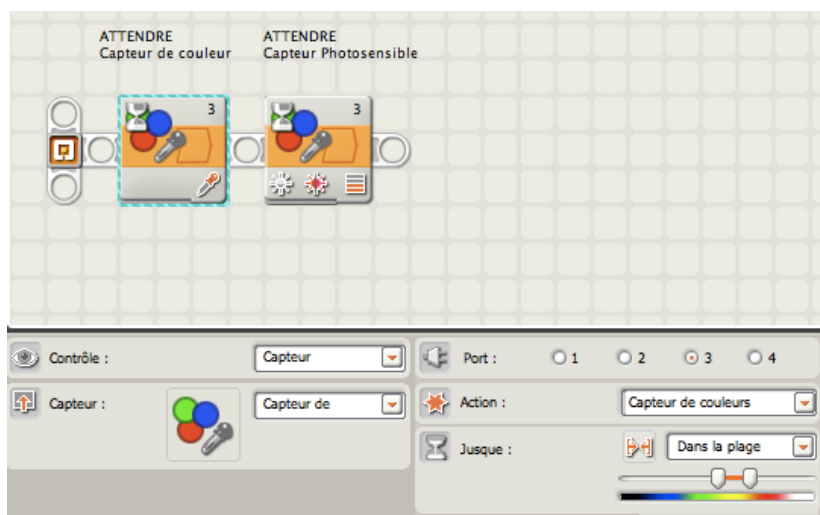
* Si le capteur photosensible est sélectionné, on retrouve une certaine ressemblance avec l'ancien bloc «Attendre, capteur photosensible». (A titre d'information, voici à quoi il ressemble). Il est utile de rappeler que cette fonction permet de relever l'intensité d'une source lumineuse, et aussi de mesurer l'intensité lumineuse de la lumière ambiante ou



de la lumière réfléchie.

La nouveauté se trouve dans la zone «Lumière». En effet, ce capteur a la possibilité d'émettre 3 couleurs différentes: rouge, vert ou bleu. On peut donc choisir la couleur de la lumière émise, et pour cela il suffit sélectionner le bouton d'option correspondant. En général, le rouge est la couleur choisie pour le capteur photosensible. L'utilisation d'une autre couleur de lumière peut générer des résultats différents

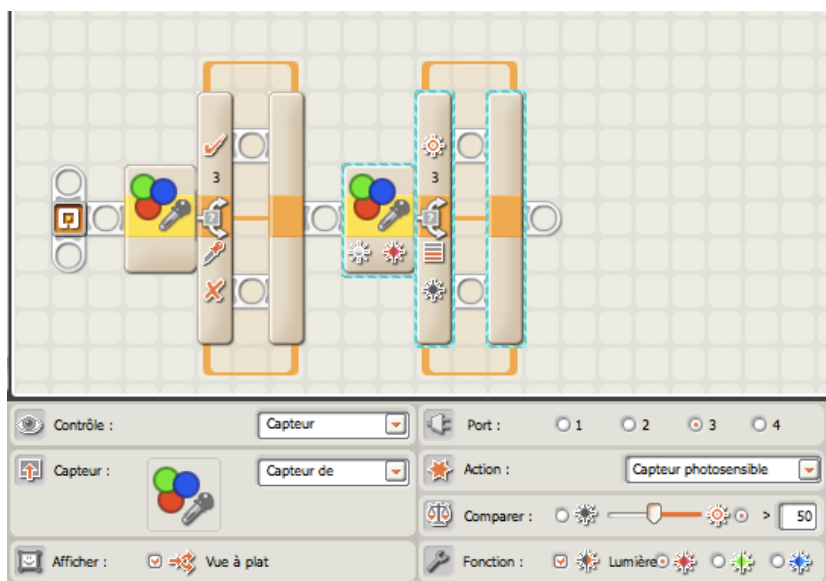
Nous disposons donc plus de souplesse pour le réglage du capteur selon les conditions d'ambiance et d'évolution du robot.



A titre de comparaison, examinez à nouveau les différences selon le type choisi afin de bien visualiser les petites icônes. Vous remarquerez que le sablier figure bien dans les 2 cas, et qu'en mode «capteur de couleurs» la plage choisie se situe dans la couleur jaune. Quant au mode «photosensible», la fonction lumière est activée en couleur rouge. Le niveau de luminosité est fixé pour une forte valeur (> 50), c'est-à-dire la clarté. Dans ce cas le robot ne réagira pas dans l'obscurité.

Remarque: ce bloc ne dispose pas de plots de données.

Le bloc «Commutation»

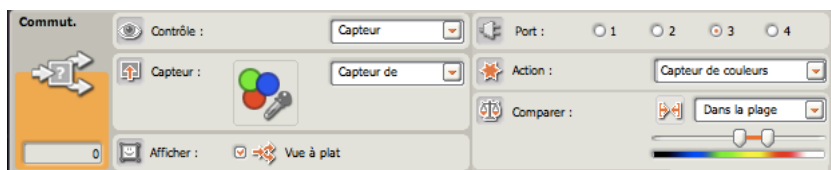


Comme le bloc «Attendre» il dispose de l'option capteur de couleurs avec les deux modes. Observez la ressemblance avec le bloc précédent; vous aurez compris le principe d'utilisation des couleurs.

✳ Dans le cas du capteur photosensible les petites icônes nous renseignent sur le mode.

Nous avons la possibilité de jouer sur l'allumage ou non, en cochant la case 'Lumière' dans la ligne

‘Fonction’, ainsi que sur ‘le bouton d’option des 3 couleurs de la lumière émise. On retrouve pour le reste, les configurations dans l’ensemble identiques à l’ancienne version.



* En mode ‘Capteur de couleurs’, Le bloc ne contient aucune petite icône significative. La commutation n’intervient qu’en fonction de la couleur choisie (6

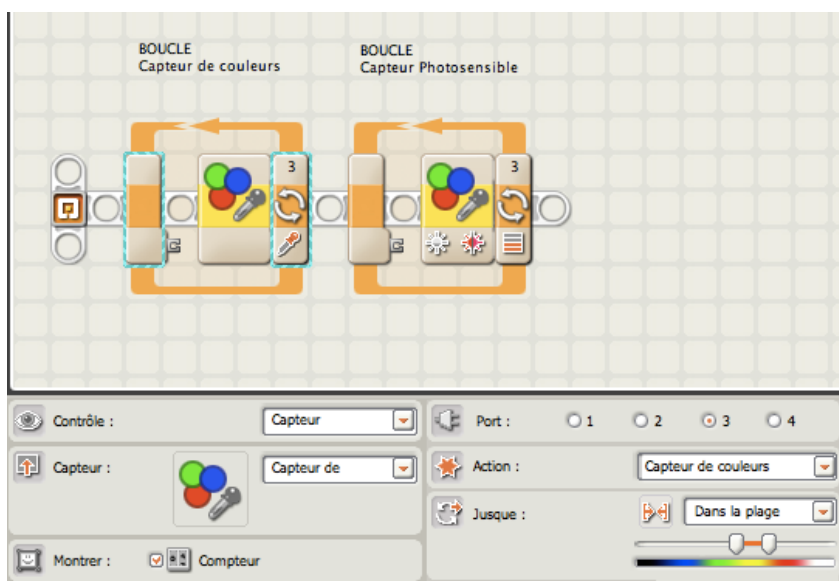
possibles), ce qui laisse supposer que le robot peut réagir d’une manière diversifiée selon les choix.(très intéressant pour détecter une piste ou une ‘porte’).

Remarque: ce bloc comme le précédent ne dispose pas de plots de données.

Le bloc «Boucle»

Comme les deux précédents, nous retrouvons les mêmes options. C’est maintenant chose acquise et il suffit d’utiliser ce bloc en veillant au bon choix du mode grâce aux petites icônes.

Tous les ajouts de blocs à l’intérieur du bloc Boucle ou tout déplacement se comportent



d’une manière inchangée par rapport à l’ancienne version 1.0 ou 1.1.

A noter ici que la case à cocher «compteur » est activée dans le panneau de configuration; une prise s’affiche alors qui permet d’utiliser le nombre de *boucles terminées* (sortie) pouvant servir d’entrée ailleurs dans le programme (si vous connectez un fil de données de la prise au plot de données d’un autre bloc).



Astuce: le capteur couleurs pourrait être utilisé pour diriger un robot vers un point donné en balisant à l’aide pastille colorées

collées sur le sol du parcours. Par exemple, si c’est vert, aller tout droit; si c’est bleu, tourner à droite; si c’est rouge, tourner à gauche. On pourrait imaginer des tas de solutions sur cette idée.

Le bloc «Lampe de couleur»

Dans la palette complète bloc d'action, nous avons à présent ce nouveau bloc qui contrôle la couleur émise par le capteur de couleurs, soit le rouge, le vert ou le bleu. Un bloc Lampe est requis pour allumer la lampe, et un autre pour l'éteindre.



Le chiffre en haut et à droite sur l'icône indique le port de raccordement au NXT. La petite icône en bas à droite indique si le bloc est configuré pour allumer ou éteindre une lampe et quelle couleur est activée. Elle disparaît en position «Off».

Dans le panneau de configuration, la zone action dispose de 2 boutons pour allumer ou éteindre. Le choix de la couleur se fait par sélection du bouton radio; par défaut, c'est le rouge qui est sélectionné.

Vous pouvez tirer des fils de données d'autres blocs vers le plot de données de ce bloc afin d'influer sur les propriétés du bloc Lampe de couleur.

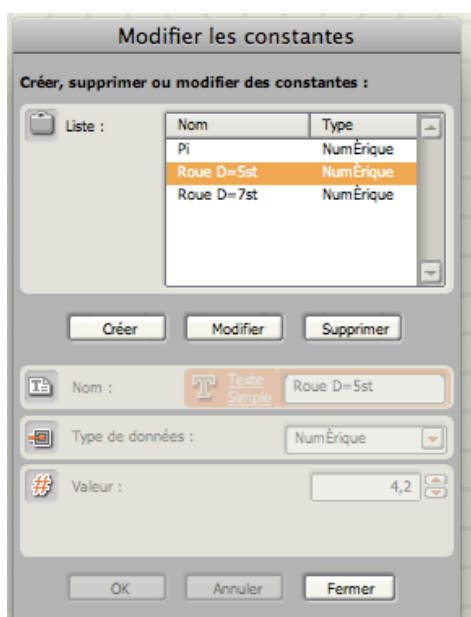
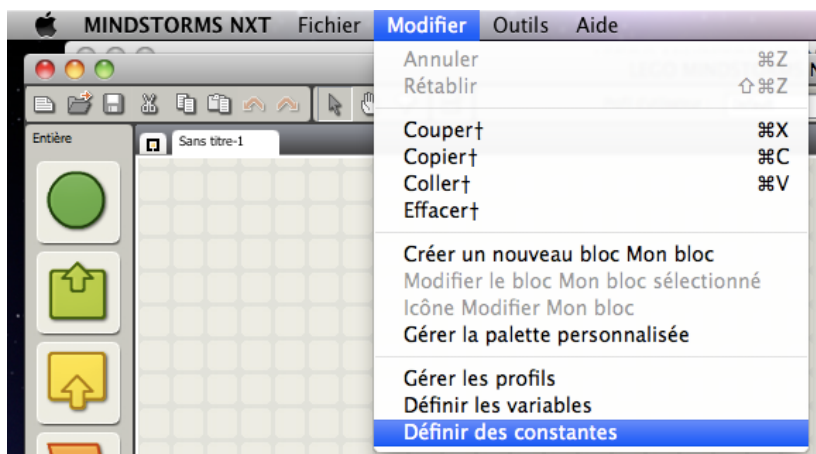
Le tableau ci-dessous présente les différentes caractéristiques des prises des plots de données du bloc

Caractéristiques des prises du plot de données du bloc Lampe de couleur. (Extrait de l'aide et support pour Lego® Mindstorms® NXT)

	Prise	Type de données	Plage admise	Signification des valeurs	Prise ignorée quand...
	Port	Numérique	1 - 4	Port physique auquel le capteur de couleur est connecté	Valeur < 1 ou > 4
	Action	Logique	On/Off	Vrai = Allumé, Faux = Éteint	Jamais
	Couleur de lampe	Numérique	0-2	0 = Rouge 1 = Vert 2 = Bleu	Action = Off ou Valeur < 0 ou > 2

Le bloc «Constante»

Ce bloc est situé dans la palette «Entière», sous-menu «Données».



Une constante peut être considérée comme un endroit où stocker une valeur dans la mémoire intégrée du NXT.

D'autres blocs de programmation peuvent lire la valeur de la constante en se connectant au bloc par des fils de données.

Ce bloc fonctionne comme un bloc Variable, mais sur une base globale. Si vous définissez une constante via «Définir des constantes» dans le menu «Modifier», vous pouvez l'utiliser avec tous vos programmes. Une constante ne change jamais avec le

temps, comme Pi (3,14...), qui est la constante mathématique correspondant au rapport de la circonférence d'un cercle sur son diamètre.

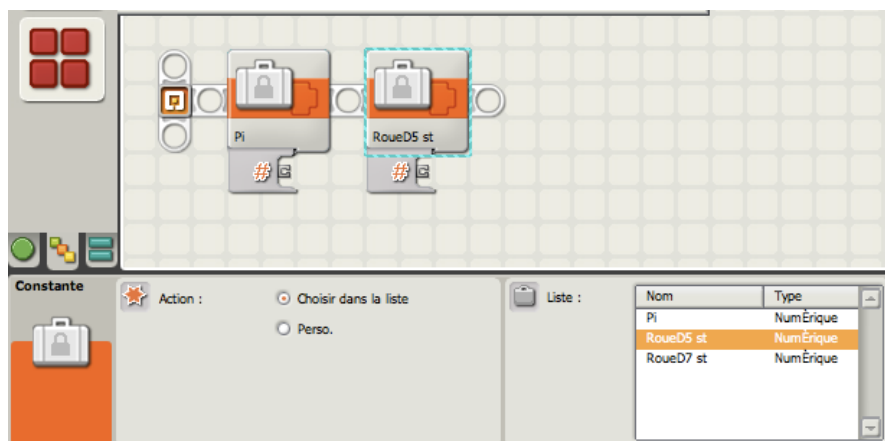
Vous avez le choix de créer une constante soit dans une liste permanente (utilisables pour tous les programmes), soit pour un programme en cours.

Le cas ci-dessous concerne le diamètres d'une roue exprimée en cm et dont le nom précise le nombre de studs. Très pratique pour calculer des distances à parcourir.

Dans le panneau de configuration (la zone Action) on remarque que le choix, à l'aide du bouton de sélection, peut aussi être personnel; dans ce cas, la constante ne concerne que le programme en cours et ne figure pas dans la liste.

Les constantes textuelles en liste sont aussi très utiles

pour l'affichage sur le petit écran du NXT. Il évite de saisir à plusieurs reprises un même texte et supprime par là le risque d'erreur. On peut citer comme exemple certains mots répétitifs comme juste, faux, oui, non, etc...



Partage d'informations entre un programme et un bloc Mon bloc

Pour partager des données entre un programme principal et un bloc Mon bloc (c.-à-d. entre deux fichiers .rbt), procédez comme suit :

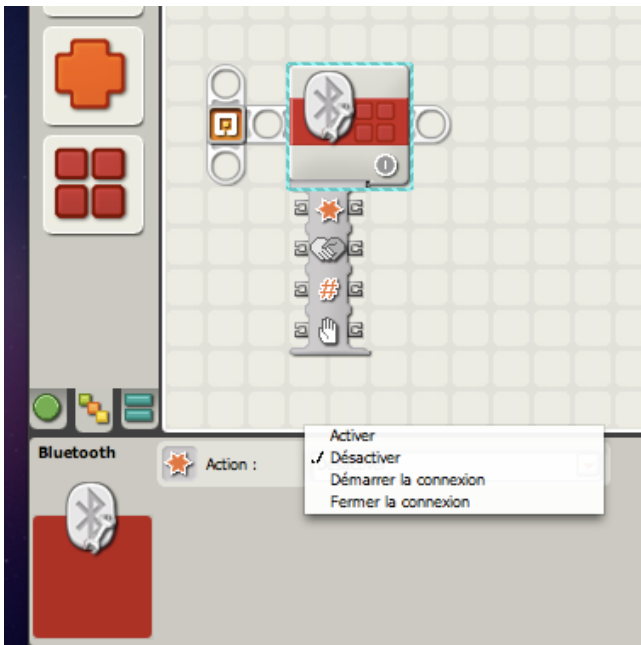
1. Définissez une constante du même nom et du même type de données dans le programme principal et dans le bloc Mon bloc.
2. Utilisez cette constante dans un bloc Constante du programme principal et dans le bloc Mon bloc.

Le bloc «Connexion Bluetooth»

Vous trouverez ce bloc dans la palette «Entière», sous-menu «Avancé».



Le bloc «**Connexion Bluetooth**» permet la connexion à un autre appareil Bluetooth, par exemple un autre NXT, un téléphone portable ou un ordinateur, avec lequel vous avez déjà établi une connexion (négocié la communication) et qui est déjà repris dans la liste des contacts.



La navigation sur le menu interne du NXT est considérable allégé, dans la mesure où ce bloc simplifie la mise en communication de 2 NXT par exemple.

Au lieu de travailler sur les petits écrans à l'aide des boutons, nous emploierons un bloc particulier, inutile donc de naviguer dans la fonction interne "Bluetooth", telle qu'elle a été développée dans la version 1.0 ou 1.1 du guide de programmation précédent. La recherche des appareils à l'aide d'un NXT n'est plus la seule solution. Mais cela sous entend que les 2 appareils sont reconnaissables à partir de votre PC et que le "jumelage" a été réalisé (voir comment procéder à partir du mode d'emploi de votre ordinateur). C'est aussi valable pour une connexion entre un NXT et un téléphone portable.

✱ Vous constaterez également que les petits symboles Bluetooth sur le coin gauche du petit écran sont toujours maintenus et permettent de vérifier de visu le bon fonctionnement de la liaison.

✱❖ Pourquoi maintenir une liaison entre 2 appareils? Rappelez-vous, je citais le cas d'un robot qui envoie un signal à un autre robot pour le déplacer, ou pour qu'il émette un son ou encore pour relever une valeur d'un capteur.

Parfois plusieurs robots peuvent évoluer à l'aide d'un même programme et s'envoient des signaux dans les deux sens; dans ce cas, choisissez l'un d'entre eux comme *maître*, et les autres (à concurrence de 3) seront configurés comme *esclaves*. Comme vous disposez de 4 valeurs possibles, les 3 briques NXT seront en communication avec une brique NXT maîtresse. Ainsi, un robot NXT pourra en contrôler jusqu'à 3.

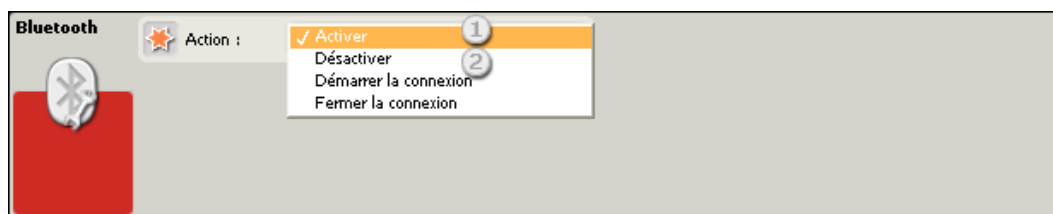
Examen du bloc

(Extrait de l'aide et support pour Lego® Mindstorms® NXT)



1. Cette icône indique le mode d'action du bloc de connexion : Activer, Désactiver, Démarrer la connexion ou Fermer la connexion

Activer/désactiver



1.

Choisissez Activer pour activer la fonction Bluetooth.

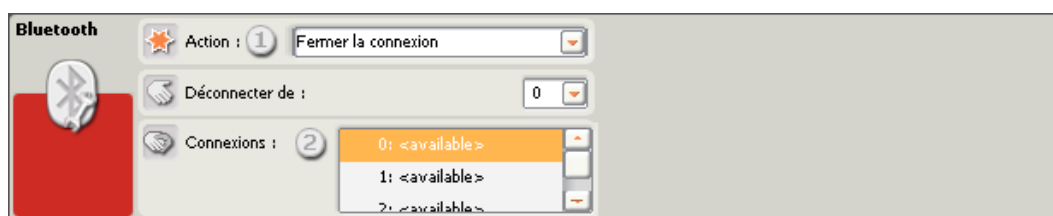
2. Choisissez Désactiver pour désactiver la fonction Bluetooth.

Démarrer la connexion



1. Choisissez Démarrer la connexion pour établir une connexion avec un autre appareil Bluetooth.
2. Lorsque le NXT est connecté à un ordinateur et activé, la liste des contacts est automatiquement complétée par la liste de contacts du NXT. Si le nom n'est pas repris dans la liste des contacts, saisissez le nom dans la zone de texte.
3. Choisissez un numéro de connexion entre 1 et 3 (le canal 0 est réservé au NXT maître ; une connexion au canal 0 doit être initiée par un autre appareil, par exemple votre ordinateur).

Fermer la connexion



1. Choisissez Fermer la connexion pour mettre fin à une connexion.
2. Choisissez le numéro de connexion à déconnecter dans le menu déroulant ou le nom de l'appareil Bluetooth dans la liste des connexions.

N.B. : Si la connexion 0 est attribuée à votre ordinateur et que vous la fermez, vous devrez rétablir la connexion dans la fenêtre NXT afin de pouvoir utiliser la fonction Bluetooth pour télécharger des programmes.

Où et comment utiliser ce bloc?

Ce bloc s'utilise d'abord dans le programme du robot «*maître*». C'est lui qui définit les liaisons avec les robots «*esclaves*». C'est donc dans le programme principal qu'il trouvera sa place.

Ensuite, il faut toujours songer à se déconnecter de l'ordinateur pour se raccorder à l'appareil esclave (rappelez-vous qu'un maître ne communique qu'avec un seul appareil Bluetooth à la fois). C'est en quelque sorte logique; pensez à une liaison filaire. Si un fil ne relie pas 2 appareils, il ne pourront pas communiquer.

Donc, commencer par fermer la connexion avec votre PC (un bloc), puis démarrer la connexion avec un autre appareil Bluetooth (un 2ème bloc).

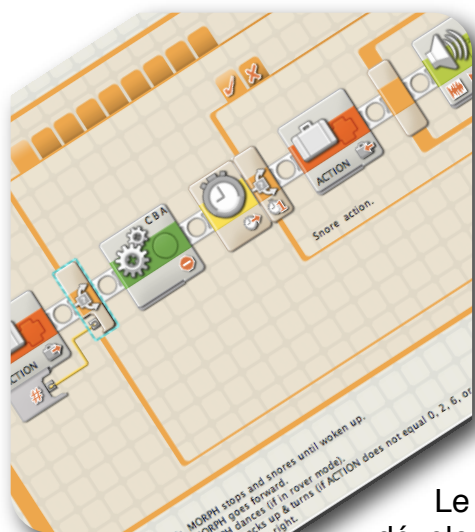
Alors que vous écrivez votre programme sur votre PC, je suppose que la liaison Bluetooth est établie entre lui et votre NXT (le maître). Dans ce 2ème bloc, au panneau de configuration, la fenêtre «Contact» affiche la liste des appareils reconnus (jumelés). Il suffit de choisir le nom de l'appareil qui sera l'esclave et le canal (n°) de connexion.

A partir de ce point, vous pouvez envoyer à l'esclave des messages et en recevoir (bloc «*envoyer message*» et bloc «*recevoir message*»), et lui ordonner des actions.

Une fois le travail de l'esclave accompli, pensez à fermer la connexion (3ème bloc), avec cet appareil, avant de poursuivre le programme.

Comme un maître peut-être connecté jusqu'à 3 appareils en même temps (un bloc pour chaque appareil), il ne pourra lancer des ordres que successivement à chaque appareil, et si des actions simultanées sont envisagées, il faudra les synchroniser par temporisation.

Remarque : Si vous disposez de plusieurs périphériques NXT interconnectés et si votre NXT maître est également connecté à plusieurs périphériques NXT, vous devez avoir au moins un délai d'une seconde dans le programme pour permettre à la radio Bluetooth de changer de canaux avant d'envoyer les messages.



9 – Exemples de programmation

Pour illustrer ces nouvelles dispositions, ces quelques exemples vous permettront d'élargir vos possibilités et vous donneront, je l'espère, de nouvelles idées de création dans votre manière de programmer.

Le paragraphe des «Mon Bloc» a été particulièrement développé, car, utilisé à bon escient, il clarifie la lecture et autorise l'écriture de programmes complexes. C'est un bon moyen pour acquérir de l'expérience et sortir des sentiers battus.

Ne l'oubliez jamais, le NXT est un ordinateur, faites-le donc travailler dès qu'il s'agit de déterminer des distances, puissances rotations, etc..Etablissez vos formules et limitez vos calculs Contentez-vous de récupérer les résultats pour les transmettre aux organes mobiles (par l'intermédiaire des moteurs).

Ce chapitre est inspiré notamment pour les «Mon Bloc» d'un texte de Dave Parker publié sur son site.

<http://www.nxtprograms.com/help/MyBlocks/tutorial.html>

Le capteur de couleurs

En mode Couleurs

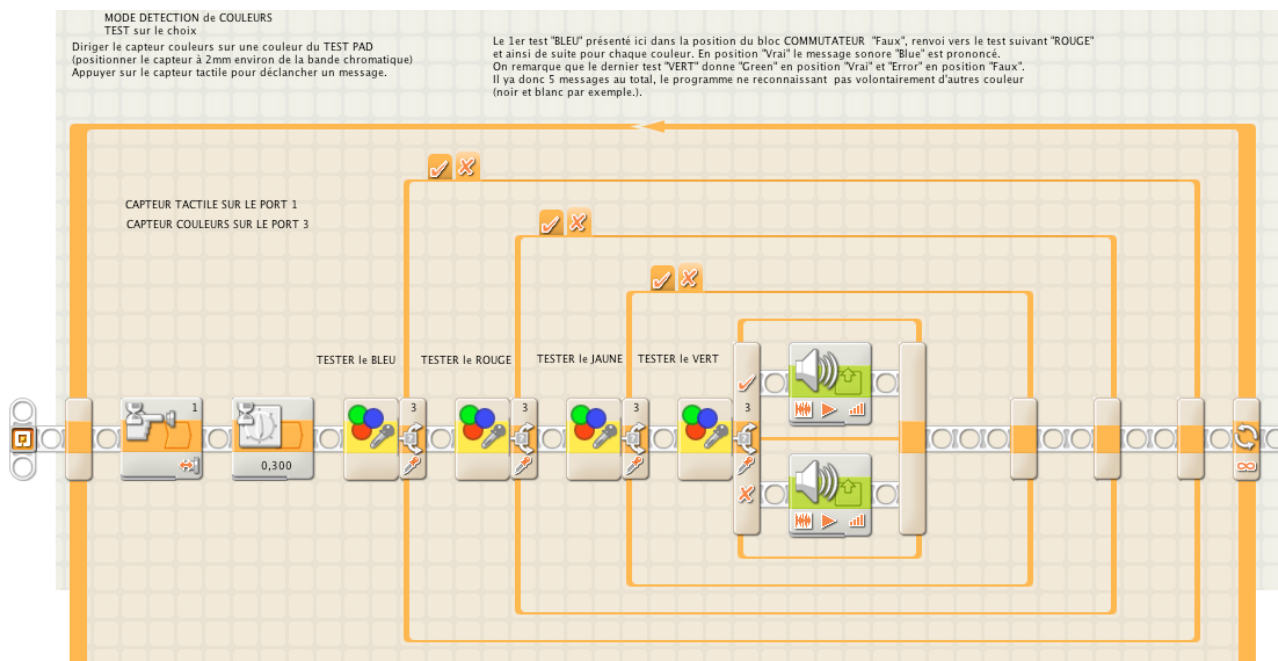
Premier exemple

Voici un programme qui réagit en fonction de la couleur sélectionnée.

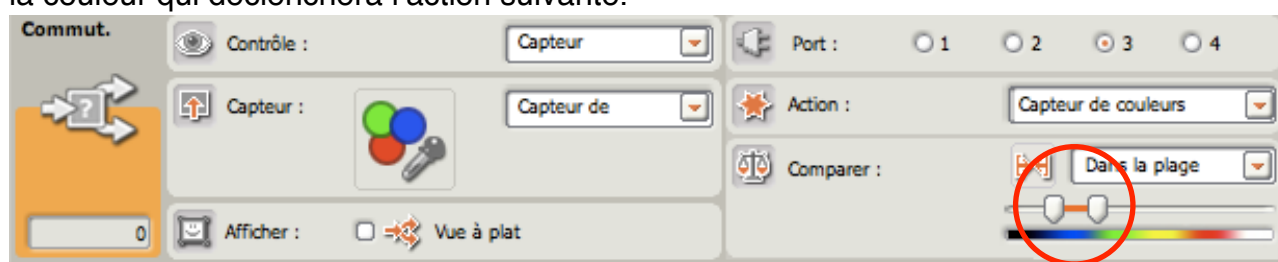
Pour cela, inutile de confectionner un robot; connectez simplement un capteur tactile sur votre NXT au port n°1, ainsi que le capteur Couleur au port n°3.

Pour le choix des couleurs, vous utiliserez le TEST PAD fourni dans le kit # 8547.

Lancez le programme installé sur le NXT. Dirigez ensuite le capteur sur la couleur de votre choix figurant sur la bande chromatique. Le capteur doit être très proche de la surface. Appuyez sur le bouton du capteur tactile et attendez une fraction de seconde. Si tout se passe correctement, vous entendrez un message sonore « BLUE » (en anglais) si vous avez choisi cette couleur. Recommencez le test avec une autre couleur. Vous devez obtenir le nom de la couleur sélectionnée. Si le capteur Couleurs est trop éloigné de la bande chromatique ou si vous sélectionnez une couleur non programmée (noir ou blanc), vous obtiendrez le message «ERROR»



Si on observe le panneau de configuration du bloc **Commutateur-bleu**, on remarque «*Dans la plage*» le positionnement des curseurs sur le bleu. C'est l'endroit où l'on choisit la couleur qui déclenchera l'action suivante.

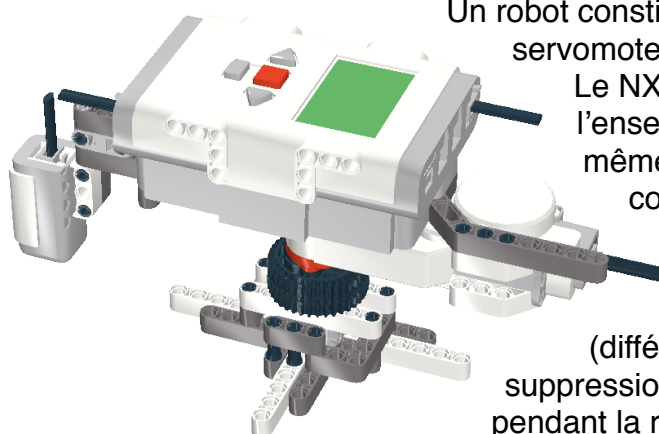


Le programme se présente comme une série de blocs «**Commutateur**» imbriqués, le tout dans une **Boucle** pour ne pas sortir du programme.

Le bloc **Commutateur-vert** vous permet de comprendre les actions qu'il est possible de déclencher; ici de simples messages sonores. Mais rien ne vous empêche d'actionner des moteurs ou d'arrêter une procédure. Par exemple, faire stopper un robot s'il détecte dans son parcours la couleur «rouge».

Deuxième exemple:

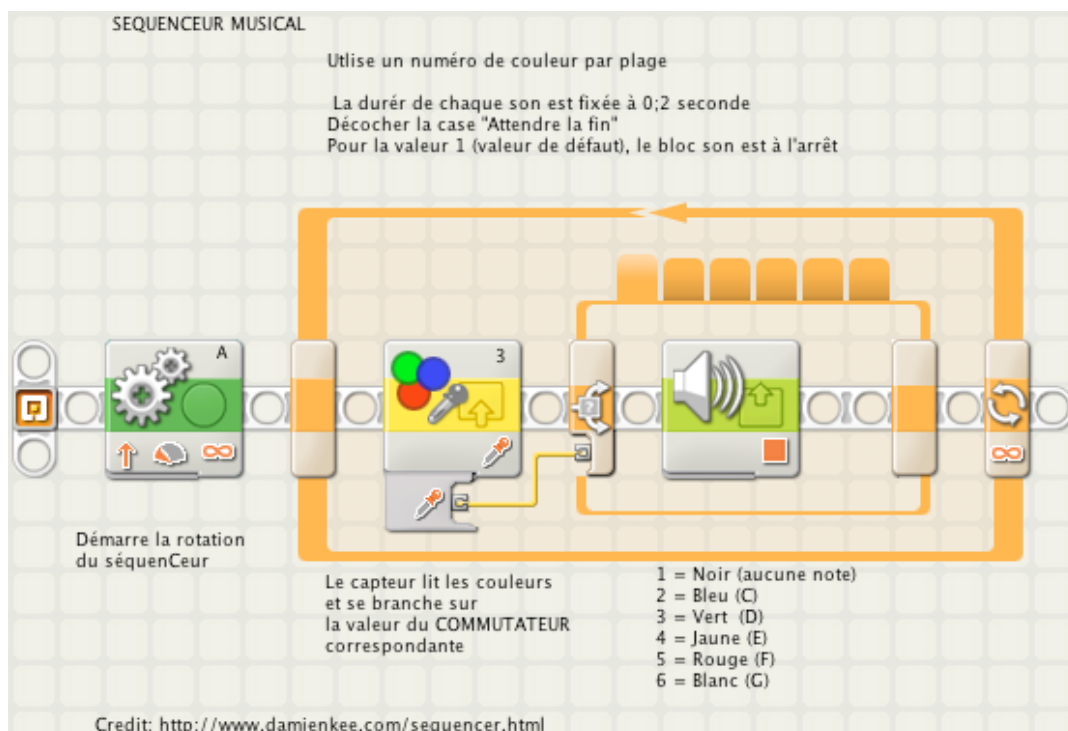
Il s'agit d'associer les couleurs aux sons.



Un robot constitué d'une brique NXT fixée sur un servomoteur, le tout reposant sur un pied fixe.

Le NXT est équipé d'un capteur de couleurs, l'ensemble est posé sur un plan et tourne sur lui-même en scrutant la surface d'appui d'une couleur proche du noir. Le capteur «lit» la couleur des réglettes colorées (6 couleurs) que l'on pose sur cette surface, et elle est traduite par une note (différente pour chaque couleur). L'ajout ou la suppression de réglettes de couleur différente pendant la rotation produit des séquences musicales variées.

Le programme est d'une simplicité déconcertante:



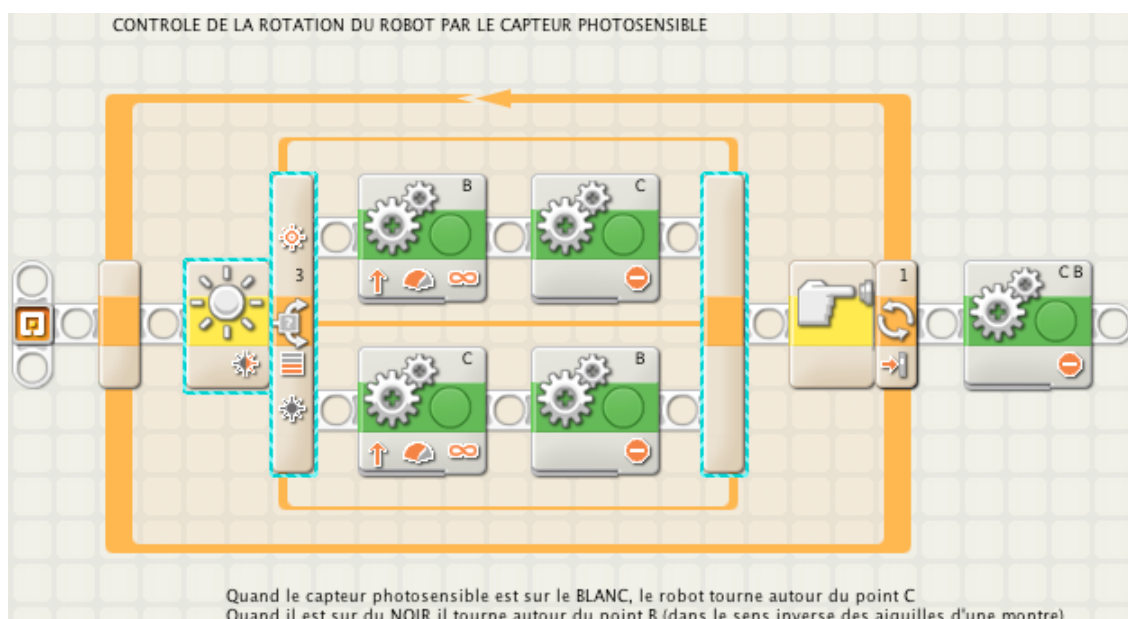
Un bloc **Déplacer** connecté au moteur A avec une puissance de 20 suivi d'une boucle **Commutateur** comportant 6 choix. Le choix 1 (défaut) ne produit aucun son. Les autres choix produisent des sons différents. Pour confectionner ce bloc, veuillez vous référer au Guide de Programmation Vers. 1.1 page 87 «Multiples décisions».

Ce Séquenceur est ludique et apprend à composer des suites répétitives de tonalités.
(Crédit à Damien Kee <http://www.damienkee.com/sequencer.html>).

En mode capteur photosensible

Voici un programme écrit avec la version 1.1 utilisant le capteur photosensible.

Un robot se déplace sur un sol comportant des zones noires et blanches. Son mouvement est contrôlé en fonction du niveau de luminosité détecté par son capteur. Il vire à droite ou à gauche selon sa position sur le NOIR ou le BLANC.

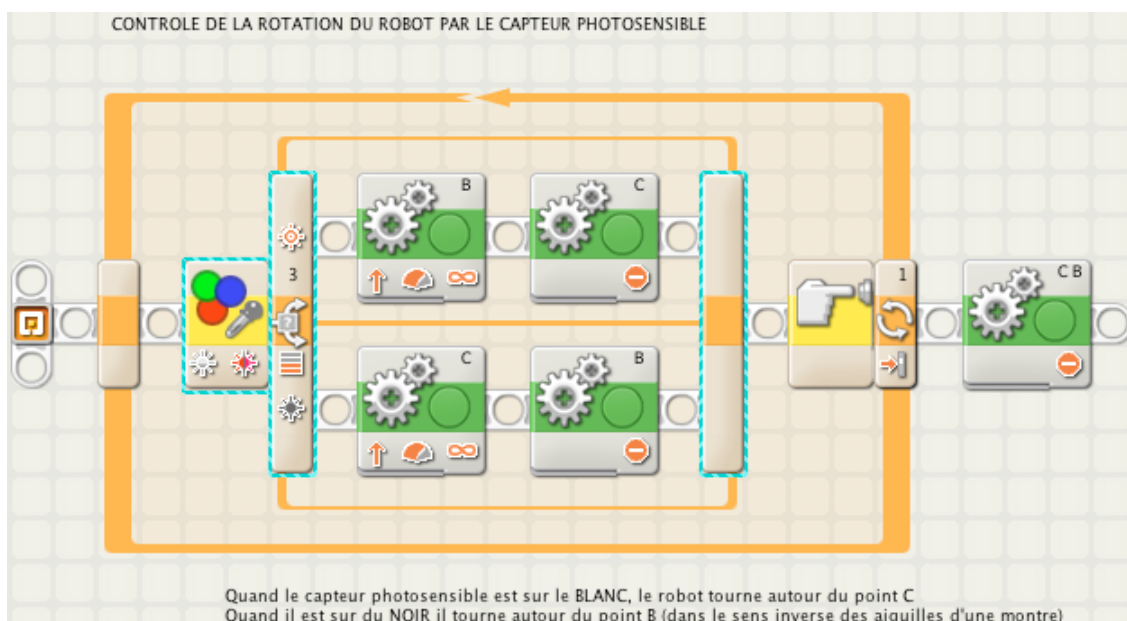


Le panneau de configuration se présente ainsi:

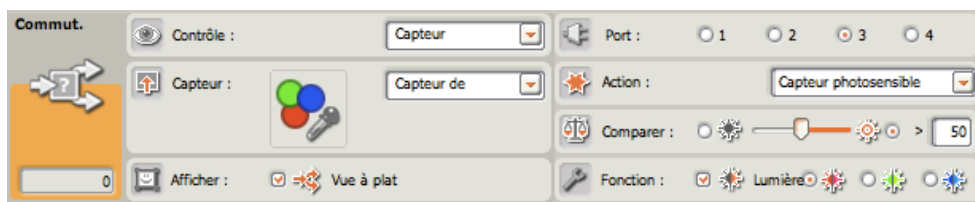


Ce programme aurait pu être écrit d'une manière identique à l'aide de la version NXT-G2.

Premier exemple: le voici réécrit en utilisant le capteur Couleurs:



Il est en tous points identique au précédent, si ce n'est que le bloc «Commutateur» est paramétré à partir du bloc «Capteur de Couleurs», et voici son panneau de configuration:



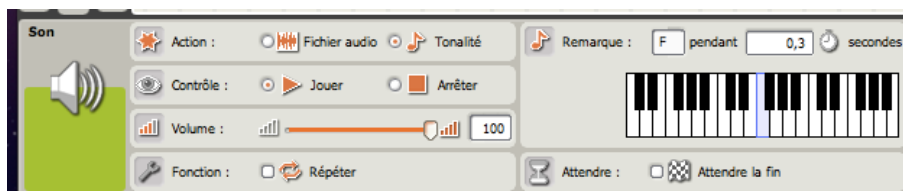
On remarque que la partie droite ressemble beaucoup au panneau de l'ancienne version. La seule différence réside dans le choix de 3 couleurs (normal), ce qui ajoute un petit plus si l'on cherche une sensibilité accrue à la luminosité, en fonction du lieu d'évolution du robot.

En mode lampe de couleur

Si vous souhaitez confectionner un avertisseur sonore et clignotant, voici un exemple assez simple à réaliser:



Il émet une couleur rouge, mais vous pouvez aussi bien choisir le vert ou le bleu.



Ce petit module peut-être intégré à l'intérieur d'un programme, ou même en faire un «Monbloc» à utiliser d'une manière générale.

crédit à Dave Parker

Connexion Bluetooth

Connexion entre 2 NXT

Dans cet exemple, il s'agit de mettre en communication deux briques NXT, et leur faire échanger des messages sonores et textuels.

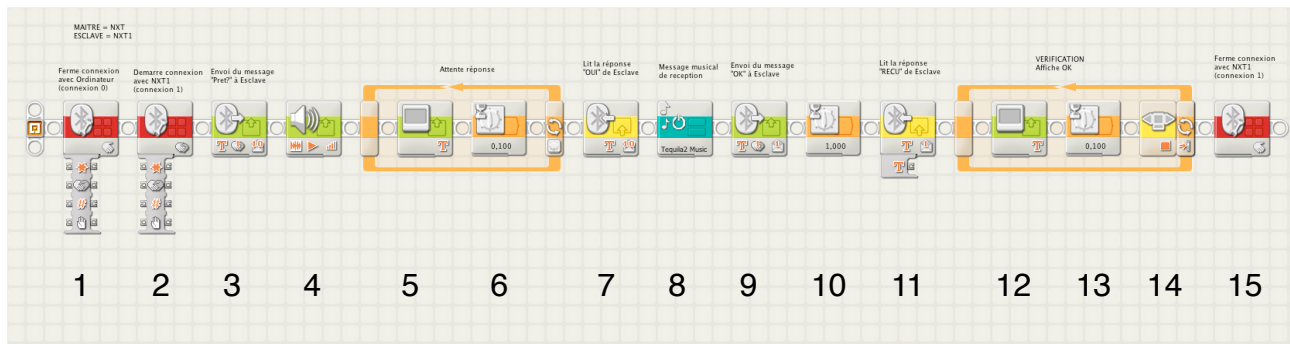
Pour que les 2 NXT puissent communiquer il faut que l'un d'eux joue le rôle de MAÎTRE (celui qui dirige et donne des ordres) et l'autre celui d'ESCLAVE (celui qui reçoit les ordres et les exécute). A partir du moment où ils communiquent, le Maître interrompt ses liens avec l'ordinateur, et se raccorde à l'Esclave qui doit déjà être en mesure de réagir. Cela signifie qu'un programme doit être écrit pour le Maître et un autre pour l'esclave. De plus, pour que l'esclave soit en mesure de recevoir les ordres, il doit être allumé et son programme lancé (logique, non?) avant de lancer celui du maître, d'où:

Règle: Avant de lancer le programme Maître, il faut lancer celui de l'esclave.

Le programme Maître:

Le maître se déconnecte du PC et se connecte à l'esclave (blocs **Bluetooth connexion** 1 & 2). Puis, il lui envoie un message textuel d'avertissement. (bloc **Envoyer un message** «prêt?», 3) et attend la réponse (blocs **Afficher** et **Attendre** une seconde, 5 & 6). Dès la réponse parvenue (Bloc **Recevoir un message** «oui», 7), le maître confirme par un petit air musical (un **Mon Bloc** 8), et renvoie un OK textuel (bloc **Envoyer un message** 9) puis attend la réponse finale de l'esclave (Bloc **Recevoir un message** 11). Une boucle d'affichage maintient l'affichage de l'accusé de réception jusqu'à ce qu'on appuie sur le

bouton rouge du NXT (12,13 & 14). A la sortie, le maître ferme la connexion avec l'esclave (bloc **Bluetooth connexion** 15), et le programme se termine.
Le maître aura utilisé 2 boîtes aux lettres.

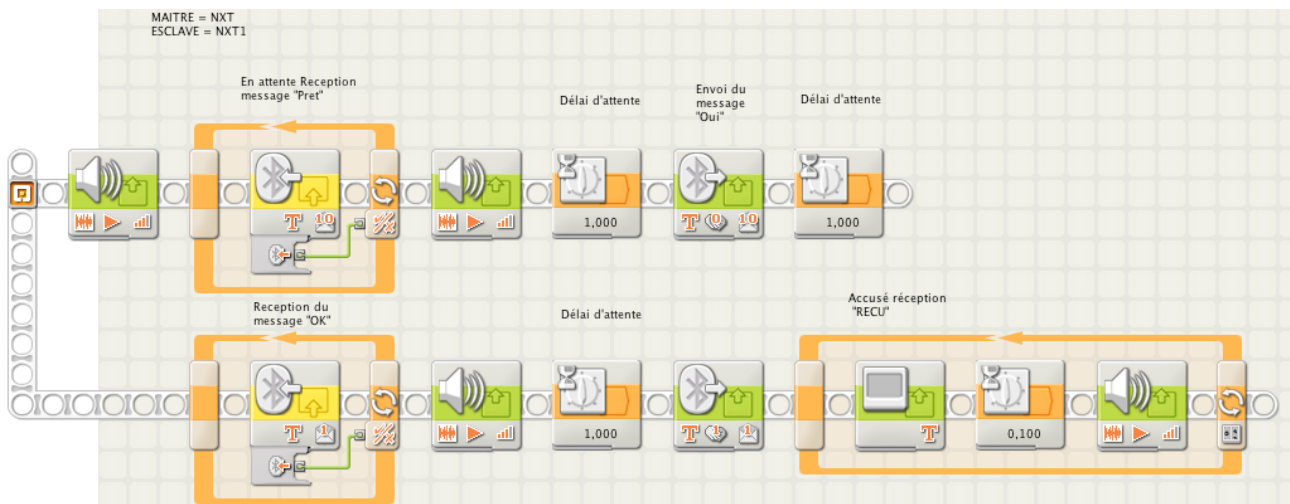


Le programme est téléchargeable ici:
files.me.com/roboleo/n4z9rn

Le programme esclave:

On remarque que ce programme contient 2 rayons de séquence.

Le premier ne sert qu'à maintenir en éveil les capacités de réception de l'esclave. Il attend en permanence le message d'avertissement du maître (Bloc **Recevoir un message** «Prêt?»). Dès que ce message lui parvient, il répond par «OK» (bloc **Envoyer un message**) dans la 1ère boîte aux lettres, ponctué par un message sonore (bloc **Son**). Le 2ème rayon de séquence contient les actions à accomplir. Ici, il s'agit, dès réception du 2ème «OK» (Bloc **Recevoir un message**, 2ème boîte aux lettres) de le signaler par un message sonore (bloc **Son**), et d'afficher la signature de l'esclave. Pour éviter toute confusion dans l'envoi et la réception des messages, on a utilisé deux boîtes aux lettres distinctes.



Le programme est téléchargeable ici
files.me.com/roboleo/cixfip

Constantes et «Mon Bloc»

Ce paragraphe est inspiré d'un texte de Dave Parker publié sur son site.

Pour illustrer convenablement l'usage des constantes et variables, nous serons amenés à traiter avec plus d'attention tout ce qui concerne les «Mon Bloc».

Je vous invite pour commencer à relire la leçon n°11 «Mes Blocs à Moi» de la version 1.1 du guide de programmation NXT-G, page 121.

La version NXT-G 2 apporte quelques améliorations que nous allons développer.

Rappels

1. Pourquoi utiliser un Mon Bloc?

La raison la plus commune d'utiliser un «Mon Bloc» est qu'elle vous permet de manipuler facilement une séquence de blocs d'une manière répétitive, sans avoir besoin de recréer à chaque fois cette séquence. En utilisant un «Mon Bloc», vous n'aurez qu'à le créer une seule fois et ensuite vous pourrez le réemployer autant de fois que nécessaire dans n'importe lequel de vos programmes. Il suffit d'insérer à chaque utilisation uniquement un simple «Mon Bloc» au lieu de recopier toute la séquence.

Un autre avantage des «Mon Bloc», est que la séquence de programme qui le compose est stockée une seule fois dans la mémoire du NXT, pour chaque programme qui l'utilise, même s'il l'utilise plusieurs fois. En vérité chaque fois que vous installez un «Mon Bloc» dans un programme, vous installez un pointeur qui dirige la suite vers le «Mon Bloc» stocké, puis revient vers ce programme initial. Mes «Mon Bloc» peuvent donc être une bonne façon d'économiser de la mémoire dans un NXT aux capacités limitées.

Comme autre avantage, on pourrait dire qu'il limite les erreurs de saisies, puisqu'il suffit de l'écrire une seule et unique fois. Si pour diverses raisons il fallait le corriger, tous les programmes faisant appel à lui seraient alors mis à jour.

Finalement, vous pourriez aussi utiliser Mes «Mon Bloc» simplement pour tronçonner un long programme en éléments plus facile d'emploi, même si vous n'envisagez de l'utiliser qu'une seule fois. Travailler avec des programmes plus courts les rend aussi plus compréhensibles.

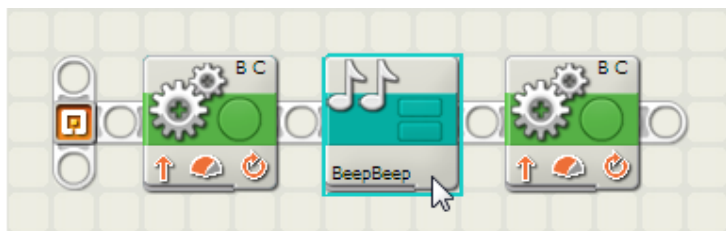
2. Créer un «Mon Bloc»

Pour commencer référez-vous à l'exemple donné page 128 figure 13 du guide NXT-1.1. Les pages précédentes vous ont déjà familiarisé avec les modalités de création. Nous supposons pour la suite que vous maîtrisez le sujet (sinon, référez à la page 124 du guide NXT-G vers 1.1.)

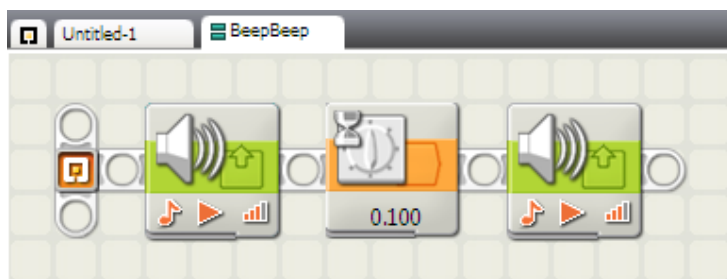
Remarque: après avoir créé un «Mon Bloc» dans un programme, il arrive parfois que la place occupée par les anciens blocs disparus, laissent apparaître un grand espace vide sur le rayon de séquence. Pour réduire cet espace, il suffit de positionner le curseur sur le rayon à proximité du bloc suivant, et de glisser la souris vers la gauche en maintenant le clic.

3. Afficher un Mon Bloc et modifier son contenu

Une fois que «Mon Bloc» *BeepBeep* a été créé, les blocs le composant ont disparu de la vue dans le programme original, comme visible ci-dessous.



Pour afficher le contenu d'un Mon Bloc ou modifier les blocs contenus à l'intérieur, vous pouvez double cliquer sur le «Mon Bloc», qui ouvrira son contenu dans une nouvelle fenêtre dans l'espace de travail, comme indiqué ci-après.



Cela ressemble à un autre programme NXT-G2 ouvert, sauf que l'étiquette BeepBeep est marquée de deux barres turquoise pour indiquer que le contenu est un «Mon Bloc» au lieu d'un programme normal.

Pour modifier le contenu d'un «Mon Bloc», il suffit d'afficher simplement le

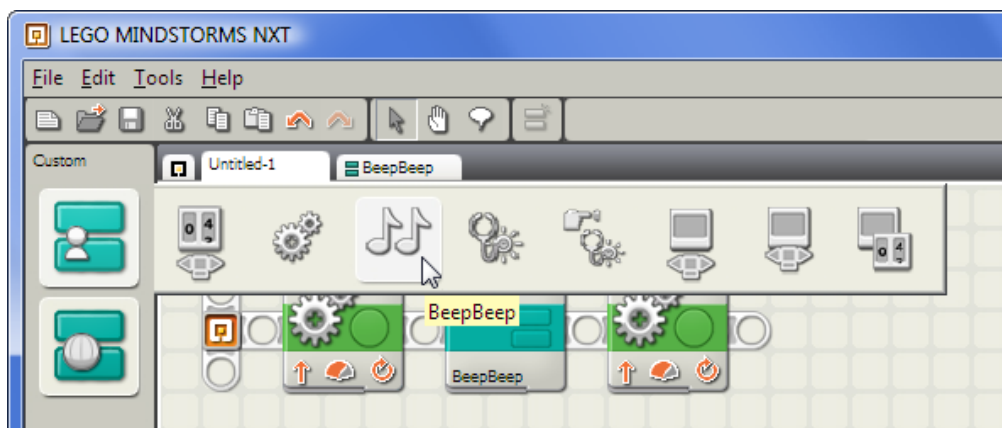
bloc marqué de l'étiquette correspondante, et de modifier le contenu comme dans un programme normal.

N'importe quel changement apporté au contenu affectera toutes les utilisations de «Mon Bloc» dans tous les programmes qui l'utilisent.

Remarque: Après une modification du contenu d'un «Mon Bloc», n'importe quel programme qui l'utilise doit être re-téléchargé sur le NXT pour incorporer ces modifications, sinon, le programme restera inchangé et utilisera le contenu ancien du «Mon Bloc». Une erreur commune après la correction d'un «Mon Bloc» est de télécharger le contenu du «Mon Bloc» lui-même au NXT au lieu du programme qui l'utilise

4. Insérer un Mon Bloc dans un programme

Après la création d'un «Mon Bloc», le logiciel NXT-G2 modifiera automatiquement le programme original pour l'utiliser comme nous l'avons vu ci-dessus. Dès maintenant vous pouvez réutiliser le «Mon Bloc» en l'insérant pour d'autres utilisations dans le même programme ou dans un autre programme.



Pour insérer une nouvelle utilisation d'un «Mon Bloc» dans un programme, choisissez la *Palette Personnalisée*. Le menu supérieur de blocs dans la Palette

Personnalisée contient des «Mon

Bloc» que vous avez créés.

La figure montre le bloc «BeepBeep» avec sept autres «Mon Bloc» créé par l'auteur. Ils sont classés automatiquement par ordre alphabétique.

Si vous utilisez des profils utilisateurs NXT-G multiples, chaque profil utilisateur contiendra sa propre liste de «Mon Bloc» qui ont été créés par ce profil utilisateur.

Ces «Mon Bloc» choisis dans la *Palette Personnalisée* peuvent être insérés dans un programme comme des blocs réguliers. Le programme modifié ci-dessus utilise maintenant deux fois le bloc «BeepBeep».



Comparé à l'alternative d'insérer directement dans le programme, et dans l'ordre, 2 fois les 3 blocs composant le

«*BeepBeep*», on comprend l'intérêt de cette formule. Le programme principal utilisant le bloc «*BeepBeep*» est plus court, plus facile de lire, plus facile à modifier et prend moins de place dans la mémoire du NXT.

Créer un Mon Bloc avec des paramètres d'entrée

Le bloc «*BeepBeep*» créé précédemment était très facile à concevoir et facile à utiliser, mais il est quelque peu limité dans son emploi. Il fournira un top sonore avec toujours les deux mêmes sons à chaque sollicitation.

C'est excellent pour quelques blocs, mais vous aurez plus souvent besoin de créer un «Mon Bloc» pouvant modifier son comportement chaque fois il est utilisé. Par exemple, chaque fois que vous choisissez le bloc **Déplacer** standard, vous pouvez spécifier une *Durée* différente, un *niveau de Puissance* différent, etc.

Ces aspects du bloc que vous pouvez modifier sont appelés «*paramètres*» (Plus spécifiquement, *paramètres d'entrée*, parce qu'ils fournissent des valeurs de données en entrée au bloc).

Il est possible de créer votre propre «Mon Bloc» contenant un ou plusieurs paramètres d'entrée qui modifient son comportement et le rendent plus flexible. C'est ce que nous allons apprendre.

1. Qu'appelle-t-on paramètre d'entrée?

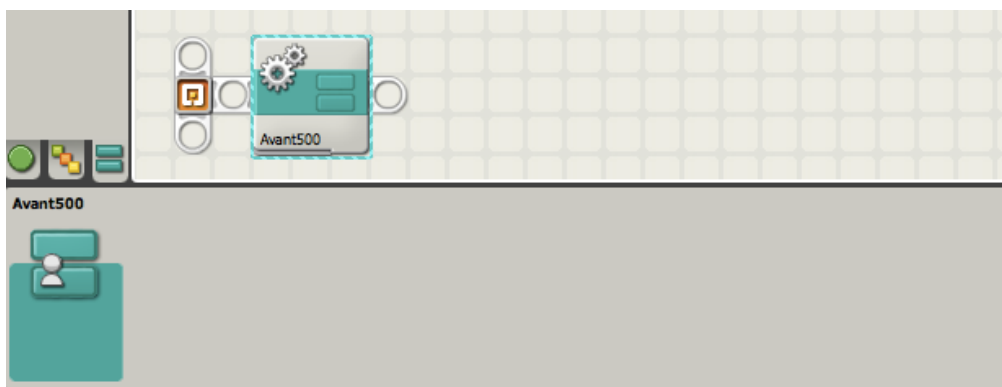
La plupart des blocs de NXT-G ont des paramètres de saisie incorporés. Par exemple, le bloc **Déplacer** ci-dessous a plusieurs paramètres disponibles dans le panneau de configuration du bloc sélectionné. Ces paramètres vous permettent de changer le comportement à chaque utilisation différente du bloc.



A titre d'exemple, créons un «Mon Bloc» en version simplifiée du bloc **Déplacer**.

Si vous choisissez la même démarche pour ce bloc que celle utilisée pour l'exemple précédent «*BeepBeep*» (un «Mon Bloc» peut consister en seulement un bloc si vous le décidez), vous obtiendrez un «Mon Bloc» sans aucun paramètre d'entrée. De plus, il utilisera toujours les valeurs des paramètres qui ont été entrées au moment de sa création. Par exemple, dans le panneau de configuration du bloc **Déplacer** ci-dessus la *Durée* a été fixée à 500 degrés, le niveau de puissance à 75, etc. La figure ci-après représente un «Mon Bloc» **Déplacer** résultant.

Parce que le «Mon Bloc» utilise en permanence une *Durée* de 500 degrés, il a été appelé **Avant500**. Puisqu'il n'y a aucun paramètre, le panneau de configuration affiché est vide.



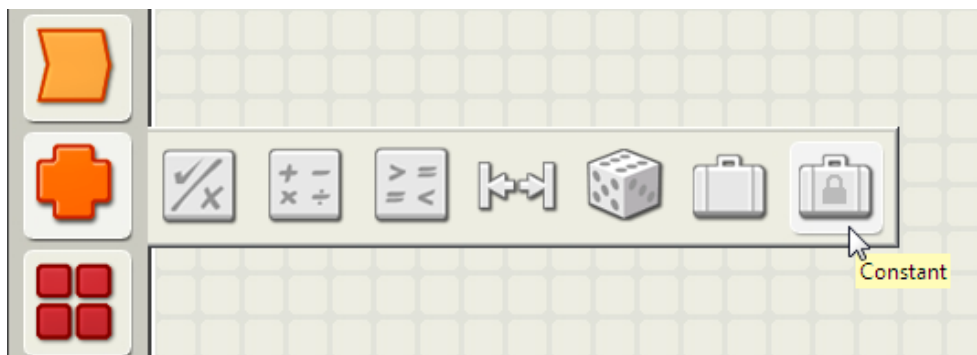
Cela pourrait être utile si vous aviez décidé de vous déplacer de cette quantité et à plusieurs reprises dans un programme. Mais il serait plus utile si ce programme, utilisant le «Mon Bloc» **Avant500**, pouvait exprimer la durée dans un paramètre, semblable au bloc **Déplacer** original.

2. Créer un Mon Bloc avec un paramètre d'entrée

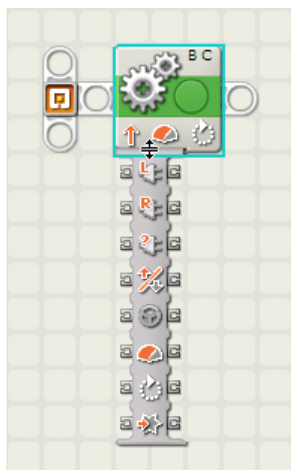
Nous allons maintenant apprendre à créer un «Mon Bloc» qui inclut un paramètre d'entrée de données. La réalisation d'un tel bloc est plus complexe qu'un simple «Mon Bloc» comme l'exemple *BeepBeep* et exige l'utilisation des plots de données, des fils de données ainsi que l'usage des blocs **Variable** et-ou **Constante**.

Soyez certains d'être familiers avec ces concepts avant de poursuivre. Si vos connaissances présentent quelques lacunes, je vous invite à revoir la leçon n° 4, Plots et fils de données (page 39), ainsi que la leçon n° 9, Variables...La valise fourre-tout, du guide de programmation NXT-G vers.1 ou 1.1.

A titre d'exemple d'un «Mon Bloc» avec un paramètre d'entrée de données, nous allons créer une nouvelle version d'un bloc **Déplacer simplifié** nommé **AvantDeg** qui inclura le paramètre *Degrés* (durée) qui permettra au programme de spécifier de combien de degrés se déplacer, au lieu de toujours utiliser 500 degrés comme dans le **Avant500** traité ci-dessus.

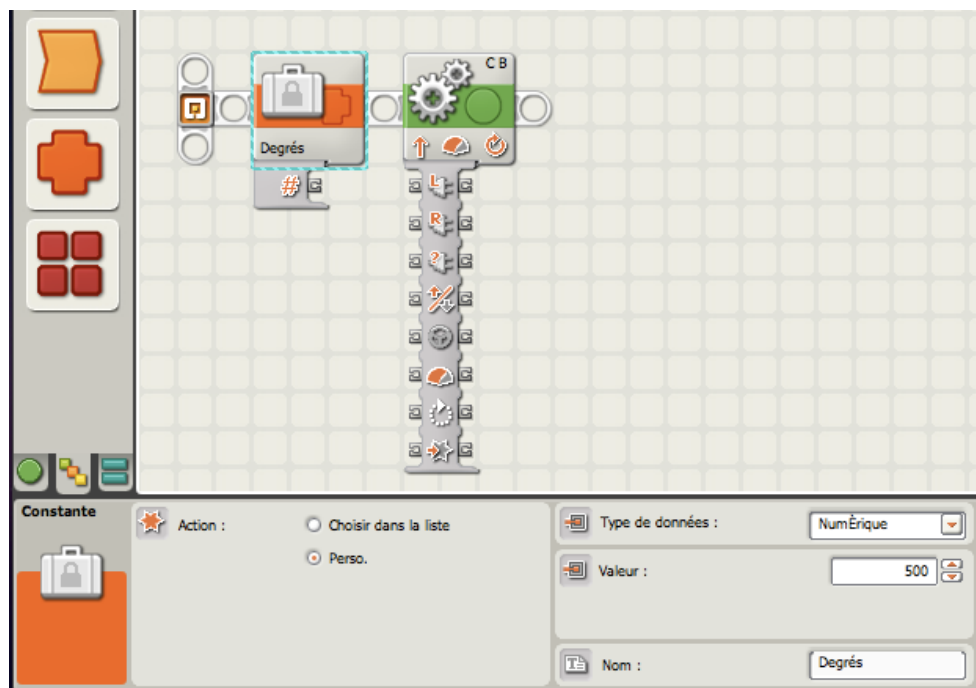


Pour créer un «Mon Bloc» avec un paramètre, il faut créer un fil de données raccordé à chaque paramètre des blocs auxquels vous voulez accéder. Nous voulons ici avoir accès au paramètre de *Durée* du bloc **Déplacer**, donc nous devons commencer par déployer les plots de données du bloc **Déplacer** comme indiqué ci-après.



Pour créer un fil de données raccordé au paramètre de «*Durée*» du bloc **Déplacer**, nous devons créer une *source* de données. Une manière facile de le faire avec le logiciel NXT-G 2.0 est d'utiliser le bloc **Constante**.

Choisissez le bloc **Constante** et positionnez-le devant le bloc **Déplacer**, comme indiqué ci-après.



Remarque : le bloc **Constante** a été ajouté au logiciel NXT-G 2.0. Comme cela a été dit plus haut, il fait parti des nouveautés.

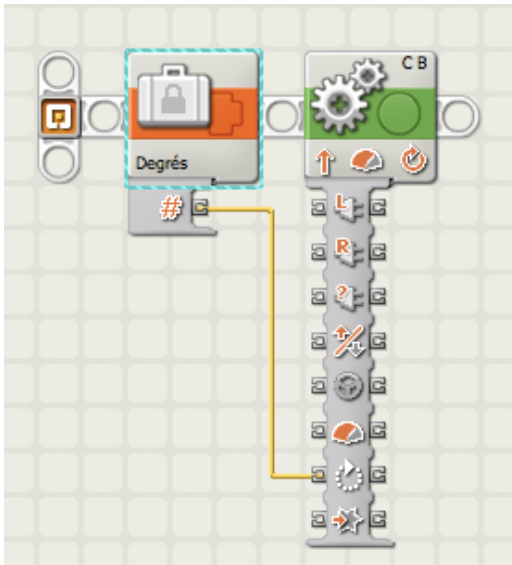
Si vous utilisez la version précédente NXT-G 1.0 ou 1.1 vous pouvez utiliser à la place, le bloc **Variable 'Lire'** Cependant, comme vous ne pouvez pas spécifier une valeur pour ce bloc **Variable 'Lire'**, il faut ajouter un autre bloc **Variable 'Écrire'** en amont dans le programme pour fixer la valeur qui sera utilisée comme constante.

Complétez le panneau de configuration ainsi:

Sélectionnez *Perso* pour l'*Action*, mettez ensuite *Numérique* pour le Type de Données, et pour valeur constante *500* dans le champ de saisie; enfin donnez un nom significatif (ici j'ai choisi "*Degrés*").

Cette valeur est utile pour l'évaluation avant de créer le «Mon Bloc» (mais le résultat du «Mon Bloc» aura un paramètre fixé pour cette valeur).

Une fois le *panneau de configuration* du bloc **Constante** correctement paramétré, tirez un fil de données entre les plots que vous souhaitez contrôler.



Ce programme à deux blocs exécutera les ordres de la même façon que le programme à un bloc réalisé plus haut. Il utilisera le bloc **Déplacer** pour avancer de 500 degrés.

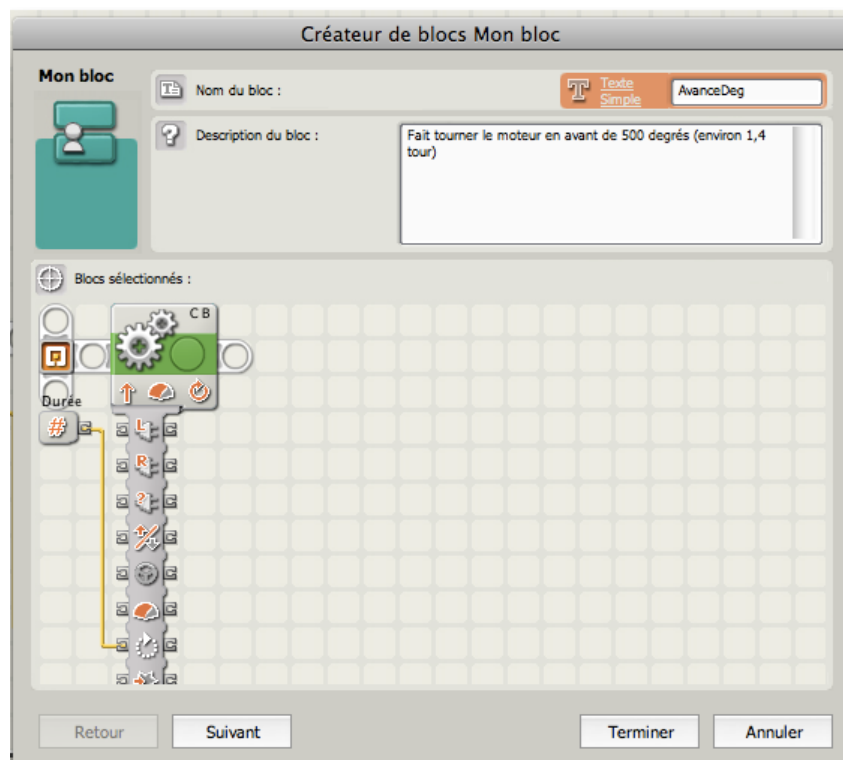
La différence est que nous avons maintenant «tiré» un fil de données vers le paramètre *Durée* à partir du bloc **Constante** dont la valeur a été fixée à 500 au lieu de mettre les 500 directement dans le bloc **Déplacer**.

Notez que tous les autres paramètres de ce bloc comme le Niveau de Puissance (Alimentation) sont définis directement sur le panneau de configuration du bloc **Déplacer** (incluant le champ de saisie «Durée» exprimée Degrés, et qui force le bloc de **Déplacer** à interpréter la valeur de 500 en degrés et non pas en secondes, en dépit de toute autre

indication figurant dans ce champ).

Il est temps maintenant de tester le programme pour s'assurer du résultat recherché.

Après avoir testé ce programme, vous pouvez le transformer en un «Mon Bloc». Pour créer ce «Mon Bloc», sélectionnez le bloc de **Déplacer**, mais pas le bloc **Constante** et procédez en cliquant sur le bouton *Créer mon Bloc* de la barre d'outils de l'espace de travail. Poursuivez selon les indications déjà connues jusqu'à la fin de la procédure. Le «Mon Bloc» sera appelé **AvanceDeg**.



Pendant le déroulement vous remarquerez dans la zone *Blocs sélectionnés*, la présence d'un plot «flottant» intitulé *Durée* relié au plot *Durée* du bloc **Déplacer** par un fil de données.

La création du «Mon Bloc» aboutira à un Mon Bloc avec un paramètre de saisie pour le bloc de **Déplacer** en l'occurrence *Durée*.

Règle : Quand une série de blocs est converti en un «Mon Bloc», un fil de données relié à une source de données située à l'extérieur du «Mon Bloc» (c'est-à-dire non incluse dans les blocs choisis) créera un paramètre de saisie pour ce fil de données dans le «Mon Bloc».

On pourrait penser que le «Mon Bloc» a été "coupé" du programme au moment de sa création et que les fils de données menant à l'extérieur des blocs «coupés» se traduisent en des «fins ouvertes» nécessitant des sources de données.

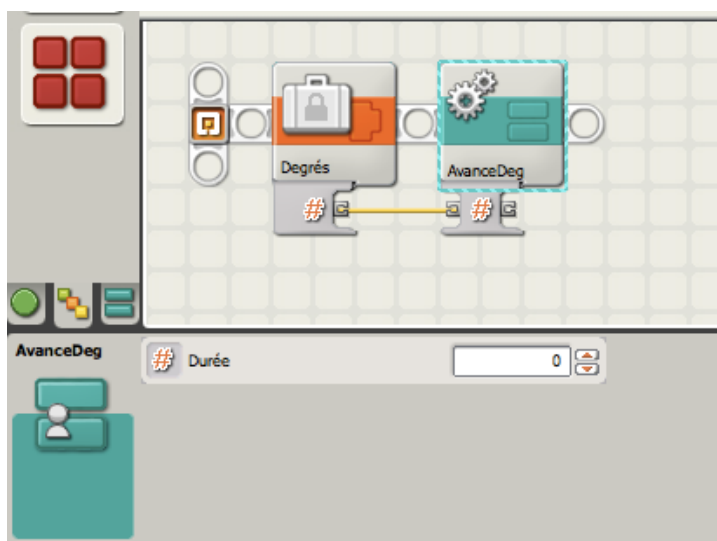
Ces fins ouvertes se traduisent en paramètres de saisie, où les sources de données seront spécifiées selon le programme qui utilise le «Mon Bloc».

Si nous avons intégré dans la sélection le bloc **Constante** pour créer le «Mon Bloc», nous aurions obtenu un «Mon Bloc» dont l'action aurait été de déplacer en avant de 500 degrés d'une manière permanente.

Le fait de sortir le bloc **Constante** permet de modifier éventuellement son contenu sans obligation de modifier tous les programmes qui l'utilisent. C'est une façon d'écrire un module de programmation universel, réutilisable en diverses occasions.

Utilisation d'un Mon Bloc avec un paramètre d'entrée

Le «Mon Bloc» ayant été crée, nous constatons que le programme initial incluant **AvanceDeg** se modifie automatiquement et se présente maintenant de la manière suivante:



Le bloc **Déplacer** (le seul bloc inclus dans le «Mon Bloc») a été remplacé par le « Mon Bloc **AvanceDeg**.

Le bloc **Constante** est toujours présent, il est maintenant relié par un fil de données au « Mon Bloc»

AvanceDeg. Sélectionner ce bloc fait apparaître le panneau de configuration; nous constatons que nous avons maintenant accès au paramètre de *Durée* du bloc **Déplacer** situé à l'intérieur du Mon Bloc.

Souvenez-vous, si vous désirez connaître les paramètres disponibles d'un «Mon Bloc», il suffit de le

sélectionner et son *panneau de configuration* affichera la liste des paramètres accessibles.

Comme pour les blocs des palettes du logiciel NXT-G, si une source de données est raccordée à un plot de données d'un «Mon Bloc», cette source de données sera utilisée, sinon c'est la valeur saisie au panneau de configuration du bloc; c'est en quelque sorte la valeur de défaut..

Ici, la valeur 500 est raccordée par le bloc **Constante Degrés**, elle s'impose donc à celle figurant dans le champ *Durée* du panneau de configuration (0). Le programme résultant fera toujours ce qu'il a fait auparavant (avance de 500 degrés). Si vous testez le

programme modifié après la création d'un Mon Bloc, il devrait faire la même chose que le programme utilisée pour créer le «Mon Bloc».

A présent que le «Mon Bloc» **AvanceDeg** est créé, vous pouvez supprimer le bloc **Constante Degrés**, et si vous le souhaitez, saisir directement la *Durée* sur son panneau de configuration. Vous pouvez également installer plusieurs blocs **AvanceDeg** dans votre programme à partir de la palette Personnalisée, et préciser leur *Durée* soit par les plots de données, soit par le panneau de configuration.

Arrivée à ce point, vous pourriez vous demander pourquoi créer un «Mon Bloc» comportant un simple bloc **Déplacer** avec un accès à un seul des paramètres?

D'une certaine manière, le bloc **AvanceDeg** semble moins utile que le bloc **Déplacer** lui-même, bloc unique disposant d'autres paramètres modifiables comme alimentation, etc..

Le bloc **AvanceDeg** n'est ici qu'un simple exemple, mais il se trouve parfois des situations où cela peut être utile. Par exemple, utiliser le bloc **AvanceDeg** peut rendre le programme plus lisible. Si ce dernier comporte un grand nombre de blocs **Déplacer**, vous pouvez désigner plus facilement celui qui *avance* ou *recule* ou encore choisir la *Durée* en degrés, sans avoir besoin de consulter le panneau de configuration. Choisir des «Mon Bloc» vous permet de lire les noms que vous avez associé à chaque bloc, et les rend plus significatifs pour ce qu'ils ont à accomplir.

De plus, vous pouvez considérer le fait qu'une modification à l'intérieur d'un «Mon Bloc» se répercutera sur l'ensemble des programmes qui l'utilisent.

Par exemple, si j'utilise le «Mon Bloc» **AvanceDeg** 10 fois dans un programme, ces 10 blocs utiliseront la même *Alimentation* se trouvant à l'intérieur du «Mon Bloc» (75 dans l'exemple ci-dessus). Aussi, si vous éditez (modifiez) le contenu du «Mon Bloc», et changez l'*Alimentation* du bloc **Déplacer** se trouvant à l'intérieur, les 10 blocs **AvanceDeg** utiliseront automatiquement la nouvelle valeur de *Alimentation*. Si vous aviez choisi 10 blocs classiques **Déplacer**, il aurait fallu les modifier un par un soit 10 fois.

Concevoir son propre paramètre d'entrée

Dans l'exemple **AvanceDeg** ci-dessus, un paramètre d'entrée a été créé pour ce «Mon Bloc», celui de *Durée*. Mais vous pouvez aussi concevoir vos propres paramètres en les personnalisant.

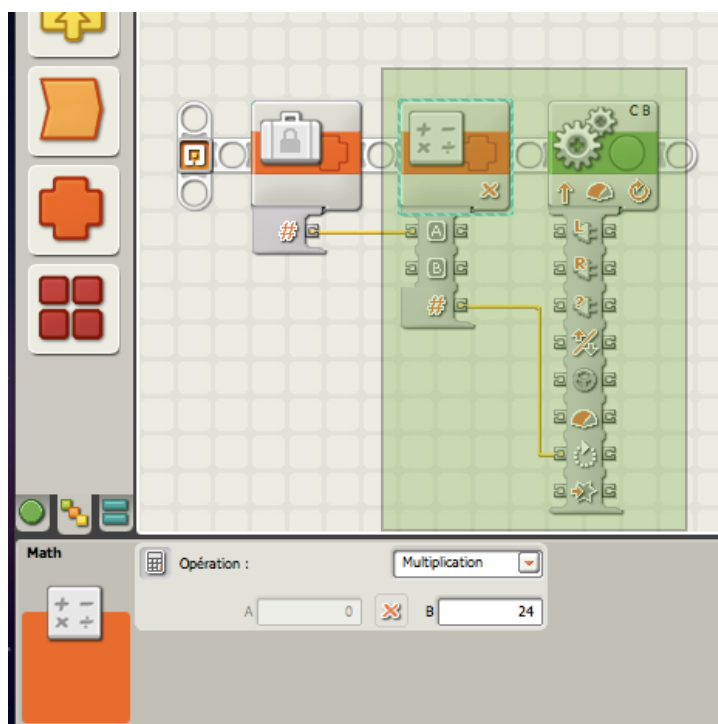
Reprenons l'exemple précédent mais réalisons le d'une manière un peu différente.

Nous l'appellerons **Avancecm** dans lequel le paramètre *Durée* (exprimé en degrés) sera remplacé par un autre paramètre sous le nom de *Distance cm* (il s'agit de la distance en cm que le robot doit parcourir). Nombreux sont les utilisateurs qui souhaitent exprimer les déplacements de leurs robots en inches ou en centimètres, plutôt qu'employer une valeur en degrés. Avec une astuce de calcul, nous pouvons créer un «Mon Bloc» qui fera ce travail à notre place.

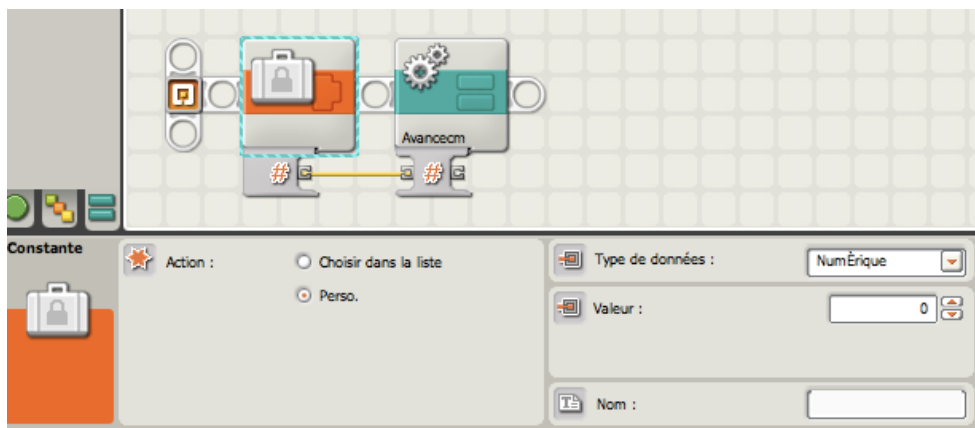
En faisant évoluer un robot sur roues équipées de pneus standard du kit # 8547 (43,2 x 22), on peut calculer la distance parcourue pour un tour de roue. (Diam.= 6 studs. Circ. = 18,85 st soit 15 cm env.) et déterminer le nombre de degrés correspondant à 1cm parcouru ($360/15 = 24$). Ce calcul est théorique, et correspond à un type de pneu. Il est différent pour un autre type, mais aussi dépend des engrenages, des frottements, etc.. Une autre façon consisterait à déterminer la distance en faisant évoluer le robot tel qu'indiqué dans le Guide de Programmation vers 1.0 et 1.1, leçon n° 13 Trucs et Astuces...page (149).

Voici à présent un programme qui à partir d'une distance en centimètres (provenant d'un bloc **Constante** intitulé *cm*) fournit au moteur le nombre de rotations exprimé en *degrés*.

On utilisera un bloc **Math**, *multiplication* avec la valeur 24 (24 degrés pour un cm parcouru).



En sélectionnant les blocs **Math** et **Déplacer** (mais pas le bloc **Constante**) on crée un «Mon Bloc».

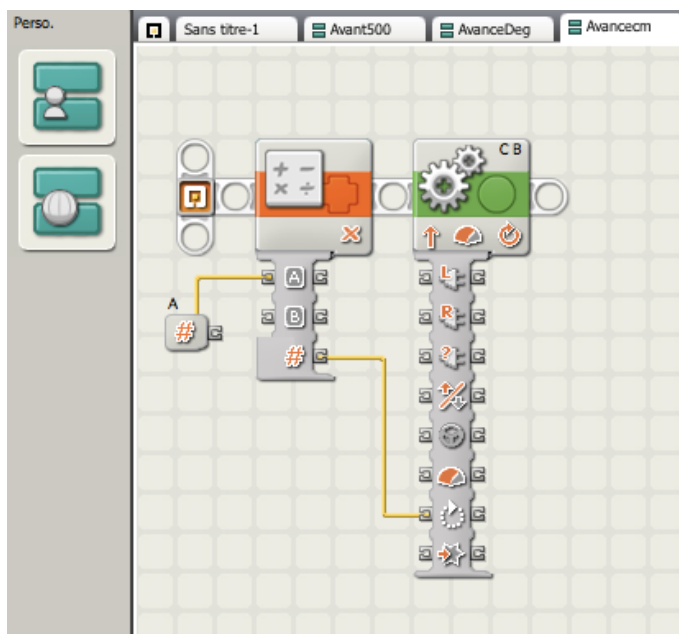


Ce «Mon Bloc» **Avancecm** dispose d'un paramètre de saisie qui permettra au programme l'utilisant, de lui fournir une distance en cm. Notez que le paramètre de saisie résulte du fil de données qui mène au port "A" du bloc **Math** quand celui a été "coupé" du bloc **Constante** pendant la création du «Mon Bloc».

Le fil menant au port de *Durée* du bloc **Déplacer** n'aboutit pas à un autre paramètre de saisie, puisque sa source de données (le Bloc **Math**) est à l'intérieur du Mon Bloc. C'est ce que nous voulons; la valeur du paramètre de saisie en cm entrant au bloc **Math**, est multipliée par 27 et est ensuite envoyée au bloc **Déplacer** converti en degrés.

Changer le nom d'un paramètre

Cliquez deux fois sur le «Mon Bloc» **Avancecm** pour ouvrir son contenu.



L'onglet **Avancecm** de la zone de travail contient les éléments du «Mon Bloc»: un bloc **Math** et un bloc **Déplacer** ainsi qu'un plot de connexion du paramètre spécial (contenant le symbole #) qui représente la source du paramètre de saisie.

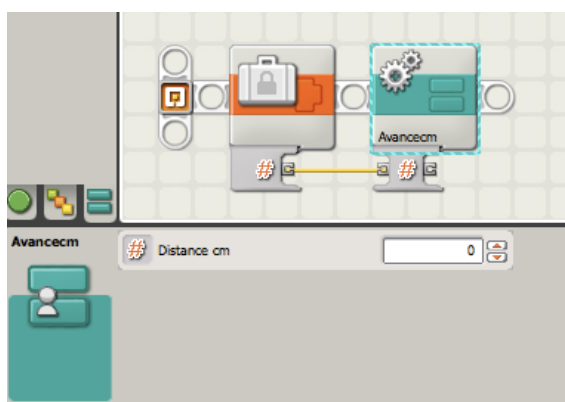
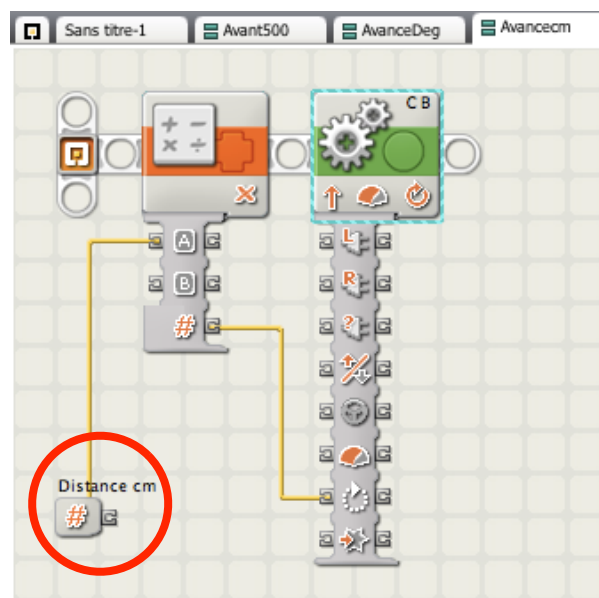
Par défaut, le logiciel NXT-G affecte au paramètre de saisie un nom en prenant celui du port de données auquel le fil source est raccordé, ainsi dans notre cas le paramètre est nommé "A", parce qu'il est raccordé au port "A" du bloc **Math**. C'est aussi le nom qui figurera dans le panneau de configuration du Mon Bloc. Vous pouvez changer le nom de ce paramètre pour le rendre plus significatif.

Voici comment procéder:

Double cliquez sur le nom du plot de connexion du paramètre, puis tapez par dessus le nom de votre choix.

Vous pouvez également déplacer ce plot en un endroit différent en cliquant sur l'icône et en le glissant vers une nouvelle position de votre choix.

Lorsque vous renommez un paramètre, le nouveau nom se retrouve également dans le panneau de configuration du «Mon Bloc». Mais pour le vérifier, vous devez au préalable supprimer l'ancien «Mon Bloc» et le remplacer par un nouveau (en prenant la précaution de reconstituer le réseau de fils de données en cas de «coupure»).

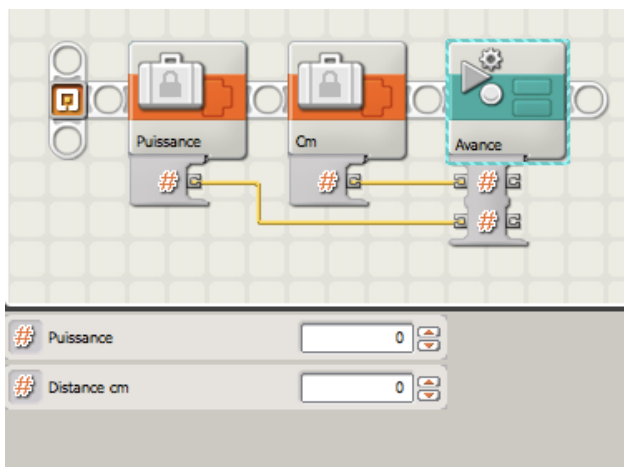


Créer un Mon Bloc avec plusieurs paramètres d'entrée

Les «Mon Bloc» **AvanceDeg** et **Avancecm** que nous avons créés n'ont seulement qu'un paramètre de saisie, mais comme vous l'avez probablement deviné, il est possible d'en

créer d'autres qui en disposent plus d'un, et c'est souvent utile.

Nous allons pour cela aborder un autre exemple qui est une variante du «Mon Bloc» **Avancecm**. Nous allons lui ajouter un autre paramètre de saisie pour contrôler le niveau de puissance.



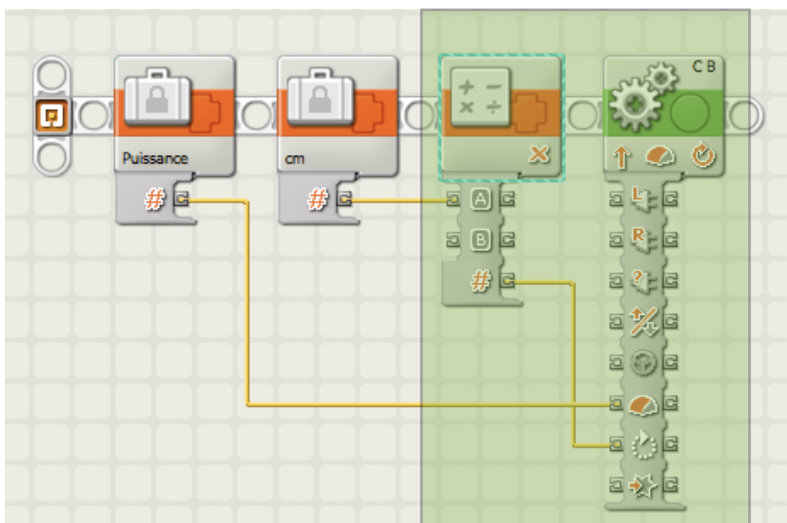
D'abord, il est important de signaler que vous ne pouvez pas ajouter un paramètre à un «Mon Bloc» existant. Vous devez obligatoirement en créer un nouveau. Vous pouvez cependant récupérer une partie de votre travail en copiant et collant éventuellement des blocs de l'ancien

«Mon Bloc», mais vous devez à nouveau passer par le processus de création du «Mon Bloc».

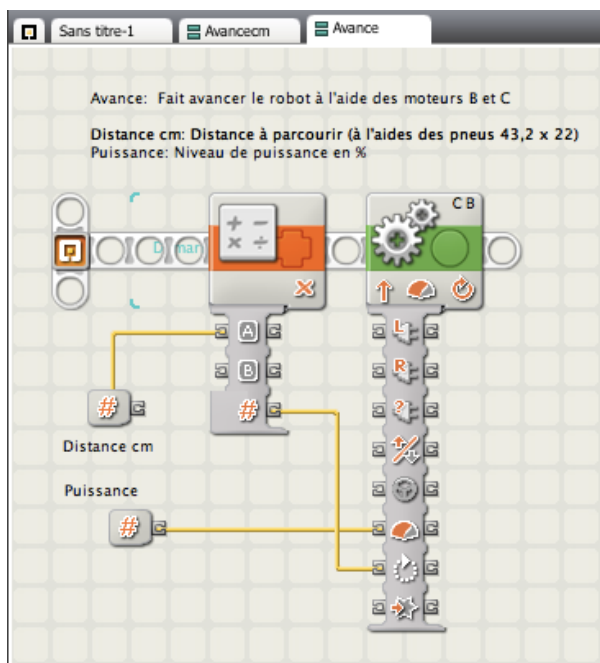
Remarque : Une fois un «Mon Bloc» créé, vous pouvez rebaptiser un paramètre de saisie, mais vous ne pouvez pas ajouter ou supprimer un paramètre supplémentaire, vous ne pouvez non plus changer le type (Nombre, Logique, ou Texte).

Avant de créer un «Mon Bloc» complexe, il est conseillé de réfléchir sur la nature et le nombre de paramètres à introduire. Pensez également à conserver une copie du programme source avant la création du «Mon Bloc». Cela facilite les révisions éventuelles.

Pour créer un «Mon Bloc» avec de multiples paramètres, commencez par créer un programme qui exécutera l'action recherchée en utilisant un bloc **Constante** indépendant avec son fil de données raccordé au(x) plot(s) de données du(des) paramètre(s) choisi(s). Ce programme ressemble au programme **Avancecm**, complété d'un autre bloc **Constante** intitulé *Puissance*, qui est raccordé au plot de données *Puissance* du bloc **Déplacer**.



La figure montre le programme, avec le bloc **Constante** *Puissance* positionné en tête du programme, pour que les blocs à inclure dans le «Mon Bloc» (seulement les blocs **Math** et **Déplacer**) puissent être côte à côte (la sélection coupera 2 fils).



Après avoir créé le «Mon Bloc», cette nouvelle version est appelée **Avance**. N'oubliez pas de modifier l'étiquette du plot «A» en «Distance cm» tel que nous l'avons vu plus haut.

Le programme source se trouve ainsi modifié. Vous pouvez constater que le «Mon Bloc» **Avance** dispose de 2 paramètres en entrées, toujours reliés par des fils de données aux 2 blocs **Constante** qui ont servi à la confection. Enfin et pour clore le sujet, n'oubliez pas de renseigner votre «Mon Bloc». Vous apercevez ici un commentaire en tête du programme. Utilisez pour cela l'*outil de commentaire* situé sur la barre d'outils, pour expliquer ce que fait ce «Mon Bloc». Si le menu *Aide* de la barre de menu du logiciel NXT-G2 vous permet de résoudre des problèmes pour les blocs courants, il n'existe aucune aide concernant

vos «Mon Bloc». Aussi, cette commodité vous rappellera (surtout après un long moment de non emploi) en quoi il est utile et de quoi il est constitué.

Documenter un «Mon Bloc» est aussi un moyen de simplifier la description d'un programme principal, puisque chacun d'eux dispose de sa propre documentation.

Variables et «Mon Bloc»

Constante ou Variable?

Comme nous l'avons vu, l'usage d'une **Constante** est un moyen permettant à un programme d'adresser des données à un «Mon Bloc». Un autre moyen est possible, à l'aide d'un bloc **Variable**.

Nous allons reprendre l'exemple précédent du programme **Avance** et nous utiliserons un bloc **Variable** pour lui adresser des données supplémentaires. L'usage de ce bloc donne plus de possibilités et de flexibilités.

Rappelons que le «Mon Bloc» **Avance** suppose que le robot du kit # 8547 utilise des roues équipées de pneus standards et que la distance parcourue en centimètres est convertie en degrés de rotation du moteur.

Mais si un robot utilise des chenilles au lieu de pneus (ou des pneus différents), la conversion ne sera plus adaptée. Notamment pour une base de robot «universelle» capable de se déplacer sur pneus ou chenilles.

C'est ce que nous allons faire à présent; nous allons écrire un «Mon Bloc» utilisable aussi bien pour un déplacement sur roues que sur chenilles.

Il faudra modifier le «Mon Bloc» **Avance** en ajoutant un troisième paramètre. Mais comme je l'ai dit, il doit être refait, puisque la modification n'est pas envisageable, mais cette fois-ci nous utiliserons une *variable*.

Pour transmettre des informations d'un point à l'autre à l'intérieur d'un programme, nous savons qu'une variable peut être placée en un endroit et lue ensuite dans un autre endroit. Une variable peut aussi être utilisée pour transmettre des informations d'un programme à un «Mon Bloc».

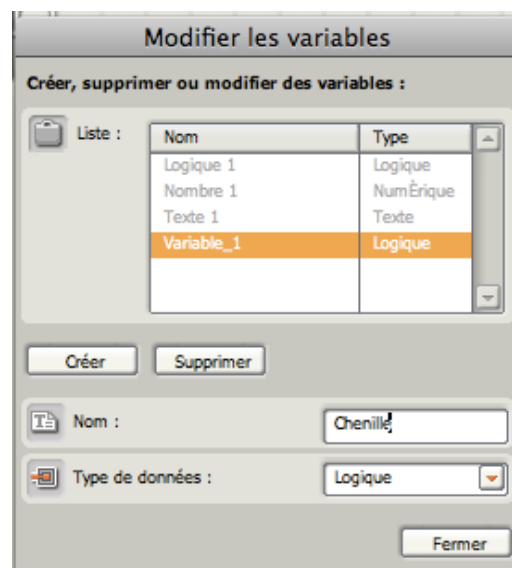
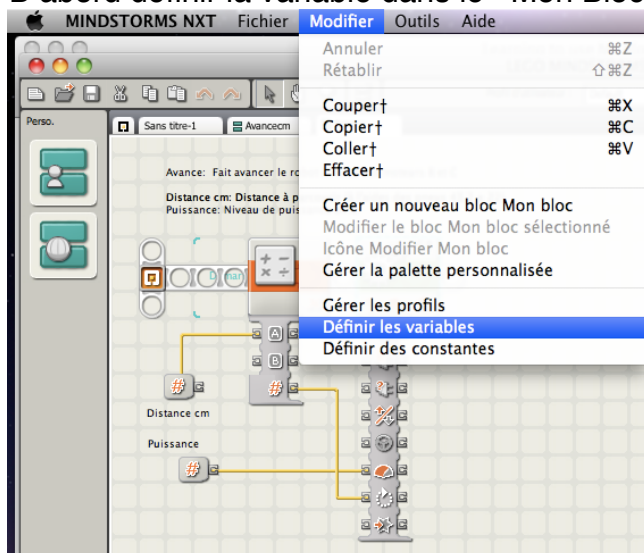
Pour compléter le «Mon bloc» **Avance** et permettre au robot de faire le bon choix (roues standards ou chenilles), nous utiliserons une variable de type **Logique** (la valeur est Vraie ou Fausse) intitulée *Chenille*. Cette variable aura la valeur **Vrai** si le robot se déplace sur chenilles, ou **Faux** dans le cas contraire. La valeur de la variable sera fixée en début du programme principal et le contenu du «Mon Bloc» **Avance** lira cette variable pour décider du choix. Si *Chenille* est **Faux** (le robot se déplace sur des pneus), chaque centimètre de parcours nécessitera 24 degrés de rotation du moteur. Par contre, si *Chenille* est **Vrai**, chaque centimètre exigera 36 degrés de rotation du moteur (déterminé par tests avec un robot équipé de chenilles).

Définir des variables dans un «Mon Bloc»

Contrairement à l'ajout d'un paramètre dans un «Mon Bloc», ajouter une **variable** ne nécessite pas une nouvelle création du «Mon Bloc».

Aussi, nous pouvons déployer le contenu du «Mon Bloc» **Avance** et faire tous les changements voulus.

D'abord définir la variable dans le «Mon Bloc» selon la méthode classique.

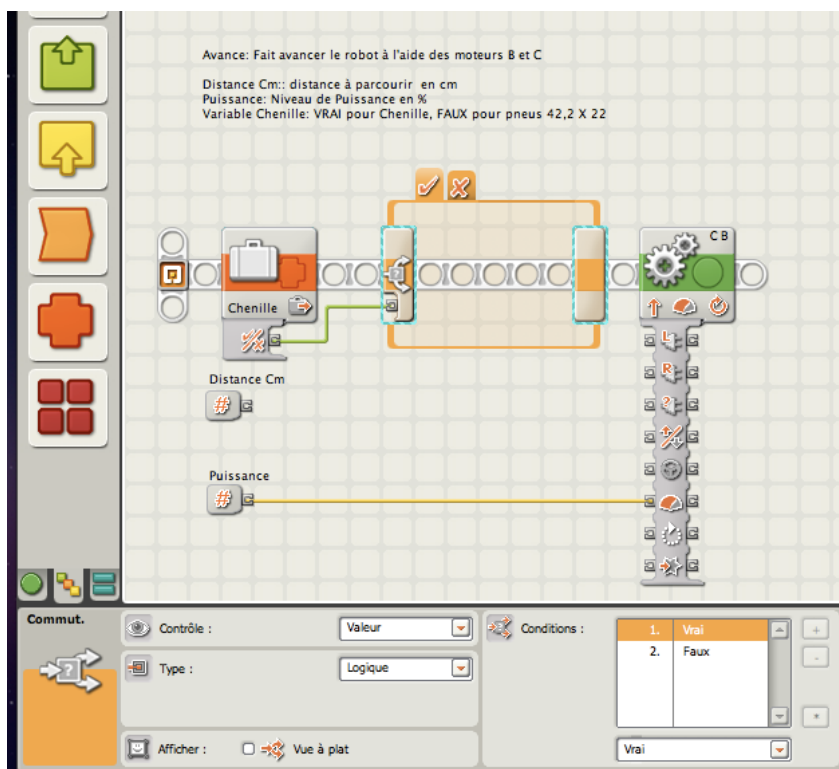


Ré-écrire le contenu du «Mon Bloc»

Après avoir défini **Variable** *Chenille* dans le «Mon Bloc» **Avance**, nous sommes prêts à modifier son contenu. Cela conduit à un changement substantiel dans la manière de composer le «Mon Bloc», aussi est-il utile de savoir qu'une fois écrit, vous pouvez aussi bien y apporter des modifications mineures que de le réécrire complètement.

En fait, les seules choses que vous ne pouvez pas changer sont le *nombre* de paramètres et leur *type*. Tout le reste peut être supprimé ou réécrit si nécessaire.

La figure ci-après montre le contenu du «Mon Bloc» après la suppression du bloc **Math**



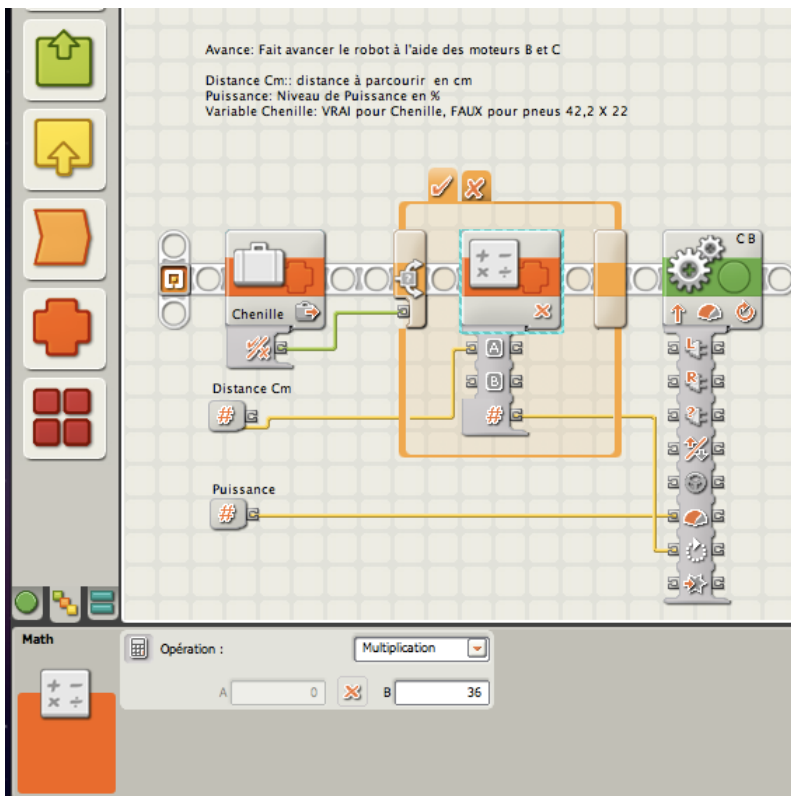
précédent et le remplacement par le bloc **Variable Chenille** raccordé à un bloc **Commutateur** pour introduire la notion de choix. Le commentaire d'en-tête du bloc a lui aussi été modifié pour tenir compte de la **variable Chenille**.

Notez que les deux plots de connexion de paramètre ne peuvent pas être supprimés, mais vous pouvez supprimer les fils de données assurant les liens, pour les reconnecter ensuite aux nouveaux blocs introduits. C'est ce que nous allons faire maintenant.

Partage des fils de données dans un Bloc «Commutateur»

Pour fixer le choix de la **Variable Chenille** nous utiliserons la technique de fils de données à l'intérieur d'un bloc **Commutateur**. Bien que mal adaptée aux «Mon Bloc», cette technique est particulièrement utile lorsqu'il s'agit de conditions pour l'utilisation d'un paramètre.

Un bloc **Commutateur Logique** sera utilisé pour tester la **Variable Chenille** et décidera s'il faut multiplier le paramètre *Distance Cm* par 24 pour des pneus (si la condition est **Faux**) ou par 36 si la condition est **Vrai**. La figure suivante montre la première partie du bloc **Commutateur** (*Chenille* est **Vrai**) relié au bloc **Math** (multiplication par 36). Notez que le bloc **Commutateur** doit avoir l'option «vue à plat» décochée dans la zone «Afficher» pour visualiser les fils de données du **commutateur** vers l'extérieur pour la suite du programme.



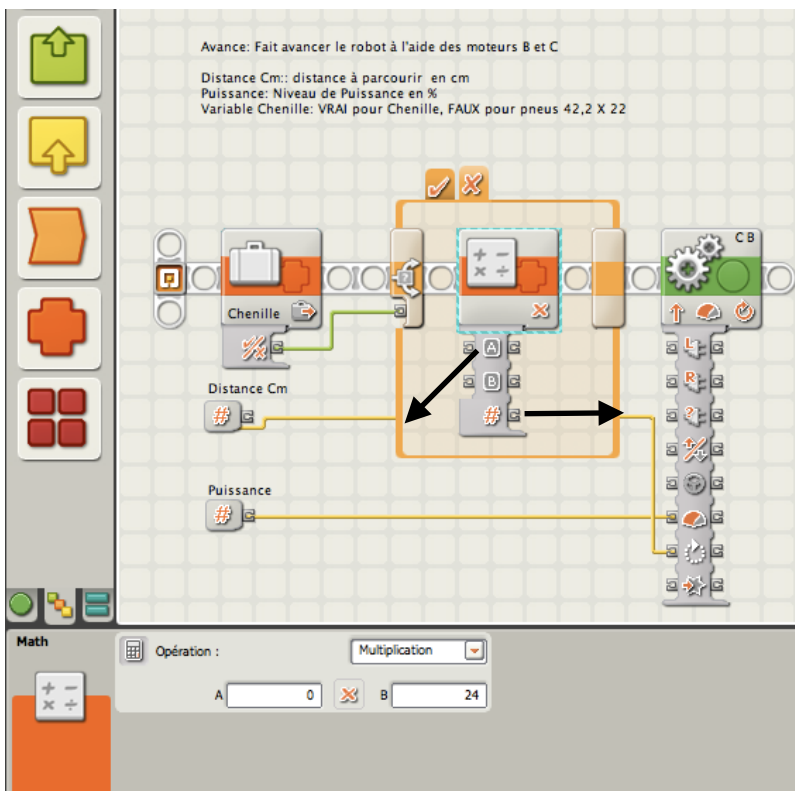
Cette figure nous montre Le bloc **Commutateur** dans la condition **VRAI** avec un bloc **Math multiplication** par 36.

La figure qui suit nous montre le bloc **Commutateur** après avoir sélectionné la condition **Faux**, contenant un autre bloc **Math** (multiplication par 24 au lieu de 36).

Vous remarquerez que les fils de données à l'intérieur du bloc **Commutateur** ont disparu. C'est normal puisque seule la condition **Vrai** a été « câblée ». Il faut donc procéder à cette même opération pour le 2ème bloc **Math** et se raccorder aux deux points de connexion cachés qui ont été créés, à la lisière du bloc **Commutateur**.

Pour tracer ces liens, « tirez » un

fil à partir du plot **A** bloc **Math** vers la lisière du bloc **Commutateur** au contact du fil allant vers le plot *Distance Cm*. Faites de même pour le plot **#** du bloc **Math** vers la lisière du bloc **Commutateur** pour un raccord vers le plot *Durée* du bloc **Déplacer**.



Pour résumer, quand il y a plusieurs conditions, il faut travailler à l'intérieur du bloc **Commutateur**, et se raccorder à la lisière. Si vous tentez le raccord vers le plot du paramètre, vous risquez d'obtenir un fil coupé.

Le résultat final consiste en deux blocs **Math** (un dans chaque condition du bloc **Commutateur**); tous les deux aboutissant au même plot d'entrée du paramètre *Distance cm* (toutefois, une seule condition sera exécutée selon la **variable Chenille**), et la valeur résultante du bloc **Math** choisi sera transmise au port *Durée* du bloc **Déplacer**.

Utiliser les variables du Mon Bloc à partir du programme principal

Une fois le contenu du «Mon Bloc» **Avance** modifié, nous pouvons à présent rectifier le programme principal. Nous l'appellerons **Test Avance** pour éviter toute confusion.

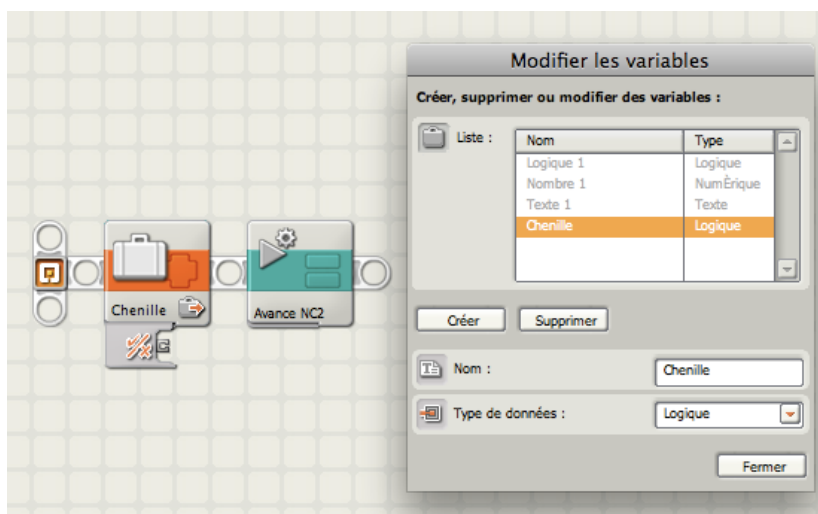
Si le «Mon Bloc» contient la **Variable** *Chenille*, le programme principal qui l'utilisera, lui, ne le contient pas. Pour s'en assurer, il suffit de l'ouvrir et de vérifier la liste des variables à disposition. Et si cette variable n'existe pas, le robot ne pourra pas exécuter les instructions.

Il faut donc la définir à nouveau pour l'ajouter au programme principal.

Ce travail doit être fait au préalable, en respectant scrupuleusement l'orthographe de son nom, et de la même façon qu'il a été défini dans le «Mon Bloc», y compris les lettres majuscules.

Remarque: Chaque programme et chaque «Mon Bloc» dispose de sa propre liste de variables utilisées à l'intérieur de chacun. Toutefois, si une variable dans un programme a le *même* nom qu'une variable d'un «Mon Bloc» utilisée dans ce programme, les deux variables disposeront de la même adresse de stockage, et pourront dans ce cas s'envoyer respectivement des données. Dans d'autres langages de programmation on parlerait de variable «globale», et dans ce sens, toutes les variables définies en NXT-G sont globales

Le programme principal se présente ainsi:



Création d'une **Variable** avec un bloc *Chenille* logique à placer en début de programme.

La variable est paramétrée en valeur **VRAI** pour indiquer que le robot se déplace sur chenille et non pas sur roues. Un exemple plus complet utilisant deux fois le «Mon Bloc» **Avance NC2** avec un top sonore intercalé précise l'intérêt de cette méthode: il suffit de paramétrer sur **VRAI** une seule fois la **variable** *Chenille*, et tous les blocs

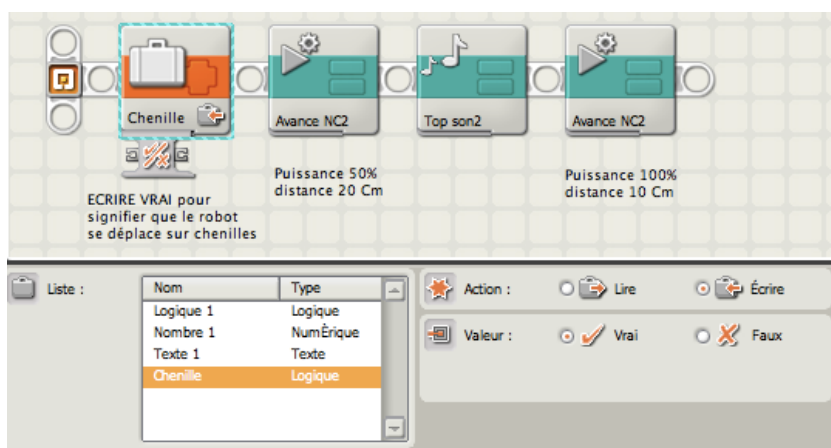
suivants utiliseront cette valeur.

Remarque: Si aucune valeur n'est écrite dans la variable avant qu'elle ne soit lue, si elle est du type *logique* elle aura la valeur **FAUX**; une variable de type *Numérique* sera égale à **zéro** et une variable *Texte* sera **vide**.

Il est recommandé de noter (en commentaire) la valeur choisie pour la variable, avant de la lire, afin d'éviter toute confusion et ne pas l'oublier dans la suite des opérations.

Si plus tard nous décidons de modifier la construction du robot en remplaçant les chenilles par des roues, il suffira de modifier la **variable** *Chenille* en début de programme et fixer sa valeur sur **FAUX** au lieu de **VRAI**, pour que tous les «Mon Bloc» **Avance NC2** soient automatiquement adaptés

Noter que, si nous avons décidé d'utiliser un 3ème paramètre pour le «Mon Bloc» **Avance NC2** pour envoyer les données de *Chenille* au lieu d'utiliser une *variable*, chacun des



«Mon Bloc» aurait nécessité une correction.i

Choisir une variable au lieu d'un paramètre est ici préférable.

En général, si la valeur d'une donnée attendue par un «Mon Bloc» est constante pendant la durée du programme, une variable peut-être choisie à la place d'un paramètre en entrée.

Remarque: Les variables peuvent aussi être utilisées pour adresser des données à partir d'un «Mon Bloc» vers le programme qui l'utilise. Un résultat calculé par un «Mon Bloc» qui est stocké dans une variable peut être lu par le programme principal ou par d'autres «Mon Bloc» du programme.

Réaliser une copie modifiée d'un «Mon Bloc»

Le «Mon Bloc» **Avance NC2** que nous venons de réaliser permet d'une manière pratique de faire avancer un robot, qu'il soit construit sur chenilles ou sur roues. Mais qu'en est-il si nous voulons le faire reculer?

Le bloc **Déplacer** dispose d'un paramètre *direction* qui autorise justement le changement du sens de déplacement. Mais notre «Mon Bloc», lui, ne peut pas assurer cette fonction. Nous pourrions faire une autre version d'un «Mon Bloc» incluant un autre paramètre pour préciser le sens de la *direction*.

Mais une autre solution consisterait à créer un «Mon Bloc» différent intitulé **Recule** pour faire agir le robot en sens inverse. Cette solution aurait le mérite de rendre le programme plus lisible, puisque il suffirait de lire tout simplement le nom du «Mon Bloc» dans le programme pour comprendre le sens du déplacement.

Ce qui suit, explique comment réaliser une copie modifiée d'un «Mon Bloc» existant ayant un nom différent, très pratique dans une situation de ce type, aussi bien que dans d'autres multiples situations (essayez à titre d'exemple des variations sur un bloc).

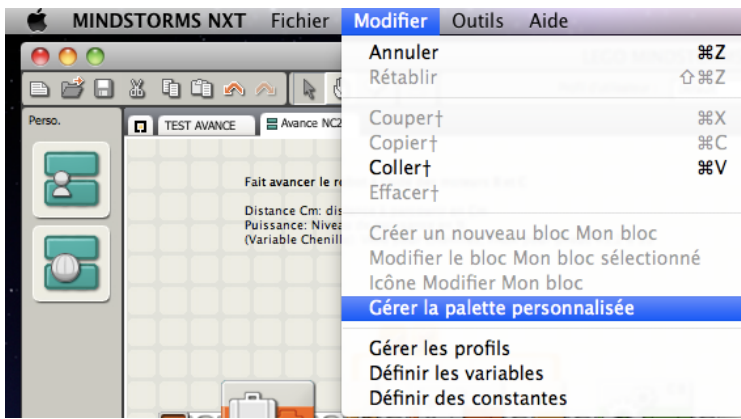
Copier un «Mon Bloc»

Pour créer un «Mon Bloc» **Recule**, vous remarquerez qu'il est strictement identique au «Mon Bloc» **Avance NC2**, sauf la direction du déplacement. Aussi, nous ferons une copie de ce dernier, procéderons au changement de direction pour enfin le renommer en «Mon Bloc» **Recule**.

Pour copier, procéder ainsi:

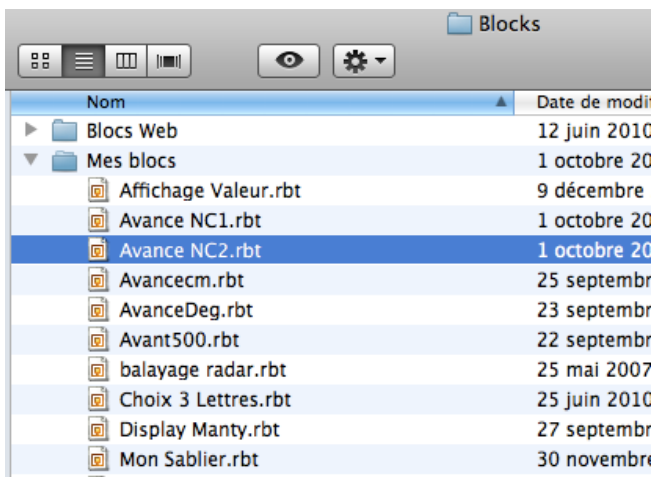
Remarque: une autre solution serait d'utiliser la commande Enregistrer sous pour créer une copie affectée d'un nouveau nom.

(Ce qui suit a été réalisé sur iMac OS X, sous Windows, les fenêtres sont différentes mais le principe est le même).

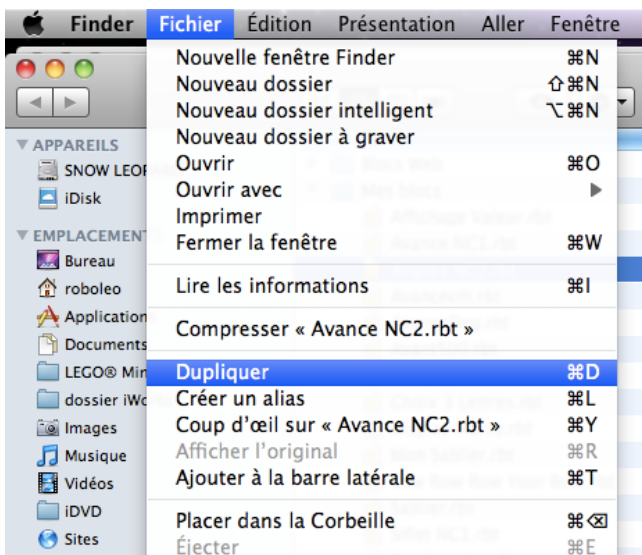


1. Choisir «*Modifier*» dans le menu principal, puis «*Gérer la palette personnalisée*»

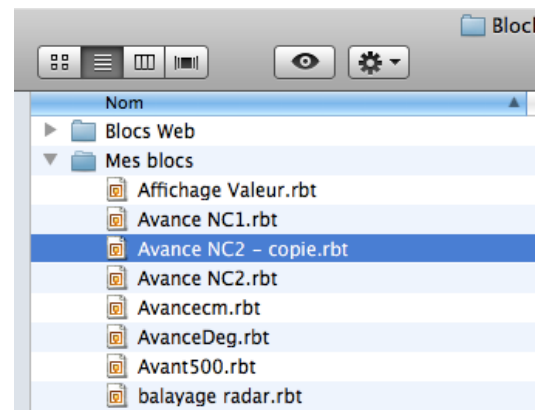
2. Une fenêtre s'ouvre contenant le dossier «Mes blocs». Sélectionner «Avance NC2».



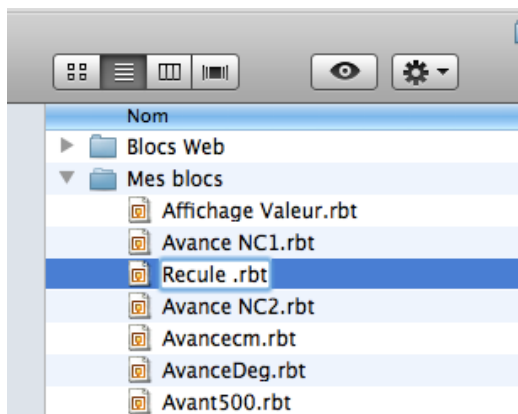
3. Dans la barre de menu, choisir «Fichier» puis «Dupliquer».



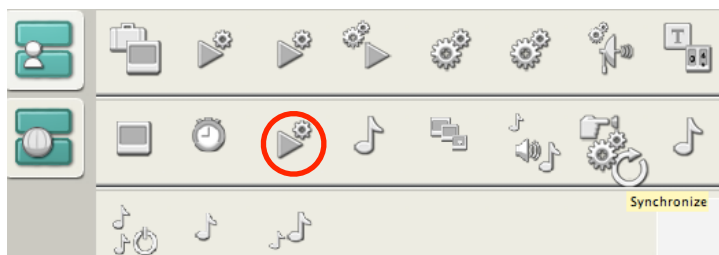
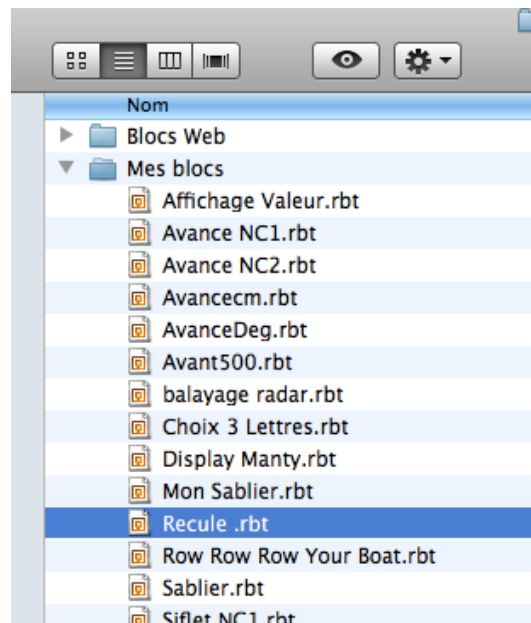
4. Dans la fenêtre précédente apparaît le nom de la copie qui s'est ajoutée à suite de l'original.



5. Il suffit maintenant de le renommer. Une fois le nom corrigé, il se place automatiquement dans l'ordre alphabétique.



6. Une fois ce travail terminé, vous constaterez qu'il s'est ajouté à la palette personnalisée. A noter que les icônes sont dans l'ordre alphabétique des «Mon Bloc».



7. Il ne reste plus qu'à modifier son contenu.

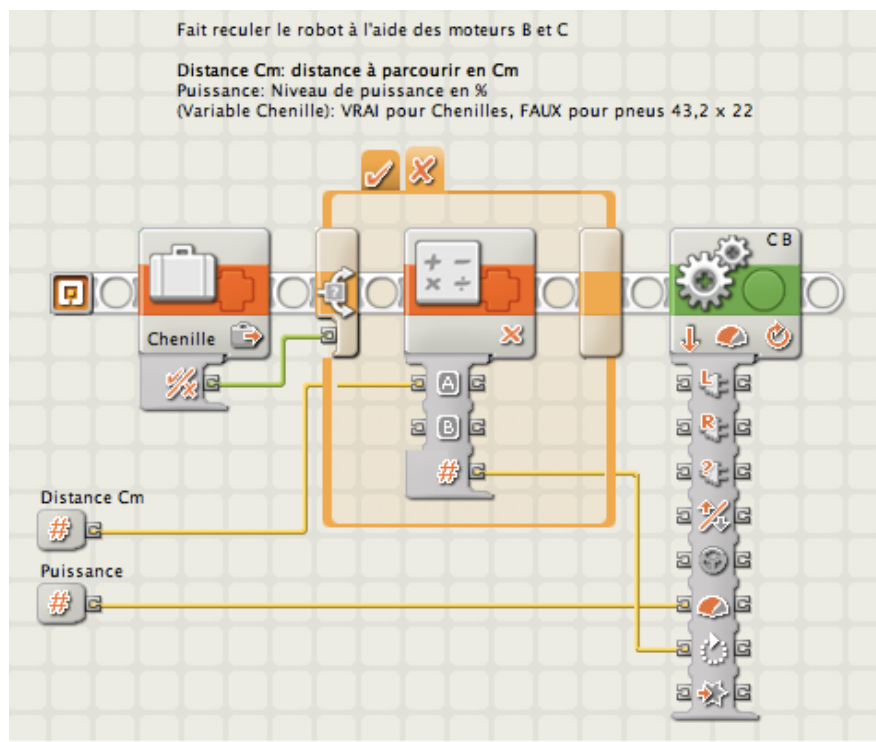
Ouvrir un nouveau programme «Sans Titre1» et installez un «Mon Bloc»

Recule.

Double-cliquez sur l'icône pour l'ouvrir.

Apportez les modifications suivantes:
Dans le bloc **Déplacer**, changez la

Direction. Modifiez le commentaire pour obtenir le résultat suivant:



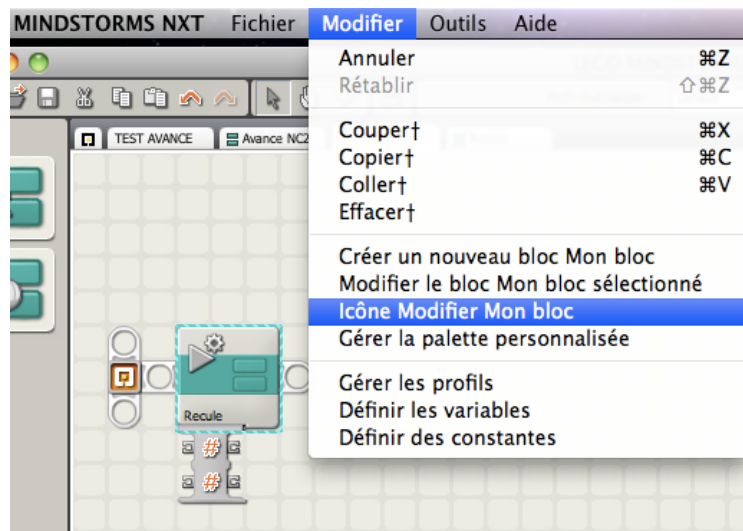
N'oubliez pas enfin de sauvegarder le fichier.

Changer l'icône d'un Mon Bloc

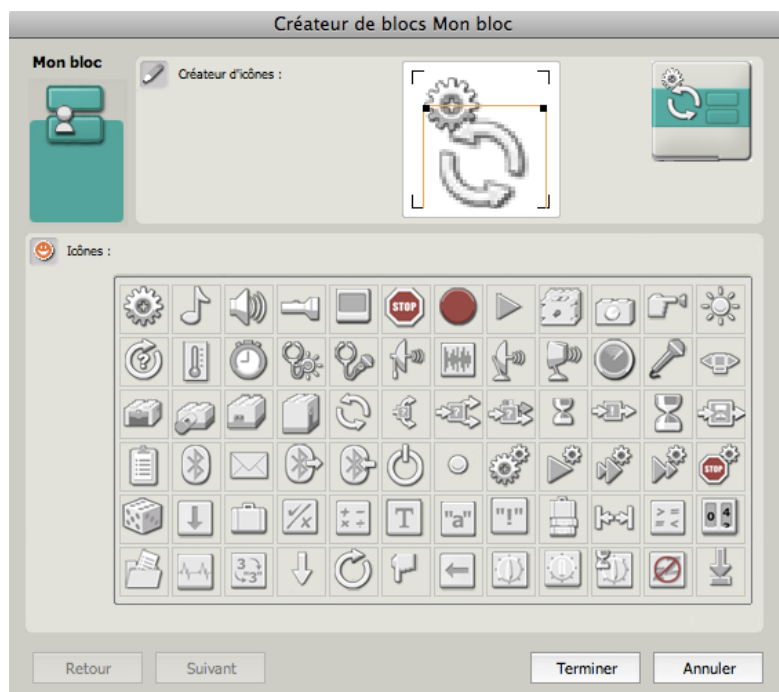
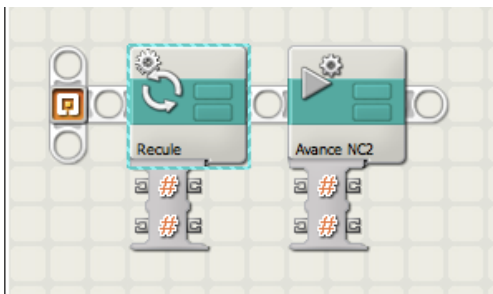
Les «Mon Bloc» **Avance NC2** et **Reculé** sont représentés par la même icône, ce qui pourrait conduire à confusion, malgré leur nom différencié. Dans certains cas et pour éviter ce genre d'inconvénient, vous souhaiteriez changer l'aspect de ces icônes. Cela est possible et voici comment procéder.

D'abord, sachez que vous ne pouvez pas faire cette modification à l'intérieur du contenu du «Mon Bloc». Il faut donc sélectionner ce «Mon Bloc» à l'intérieur d'un programme qui l'utilise.

Dans la barre de menu, choisir «*Modifier*» puis «*Icône Modifier Mon bloc*»



Cette commande vous conduit directement à la 2ème étape du «*créateur de blocs Mon bloc*» qui met à votre disposition les outils permettant de recomposer le graphique. Une fois cette opération terminée, elle s'appliquera à toutes les nouvelles utilisations de ce «Mon Bloc».



A titre d'exemple, voici à quoi ressemble cette transformation; les deux «Mon Bloc» sont bien différenciés, par leur nom et par leur icône.

Créer des «Mon Blocs» complexes

Nous allons maintenant présenter 2 exemples de «Mon Bloc» développant quelques techniques pouvant être employées pour l'élaboration de «Mon Bloc» plus complexes. Nous allons d'abord composer un «Mon Bloc» **AllerRetour** construit sur les «Mon Bloc» que nous avons précédemment créés pour avancer un robot à une distance définie, puis le faire revenir à son point de départ. Ces mouvements seront ponctués par des tops sonores avant chaque mouvement.

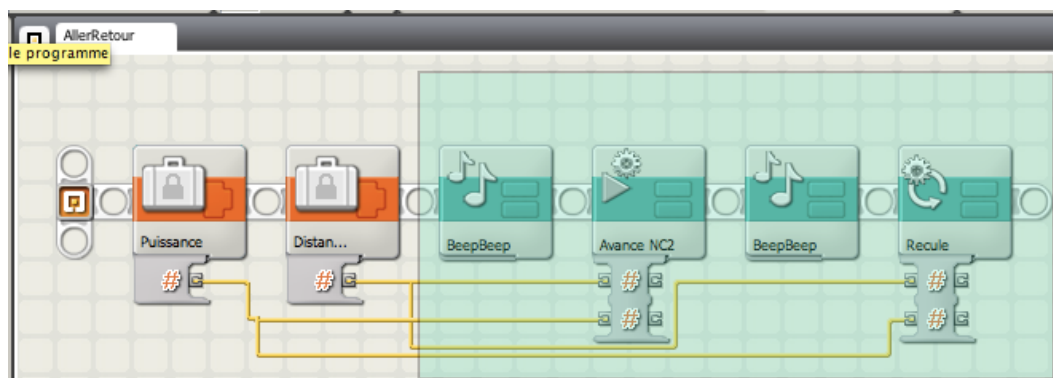
Le 2ème exemple, plus complexe est intitulé **NbreLimit**. Il testera et modifiera une *valeur Numérique* pour s'assurer qu'elle se trouve dans une plage donnée. Ce «Mon Bloc» **NbreLimit** est suffisamment complexe et requiert une bonne organisation ainsi que des tests de contrôle.

Enfin en conclusion une méthode comprenant une liste d'étapes pouvant servir de processus pas-à-pas, à utiliser comme guide pour la création de «Mon Bloc» plus complexes.

Utiliser un Mon Bloc à l'intérieur d'autres Mon Blocs

Pour écrire un «Mon Bloc» **AllerRetour**, nous pouvons remarquer que les trois actions envisagées pour ce «Mon Bloc» ont déjà été développées précédemment. Il se trouve fort heureusement, qu'un «Mon Bloc» peut en contenir d'autres, le «Mon Bloc» **AllerRetour** sera donc facile à réaliser à partir de l'utilisation de ces trois blocs.

Voici le programme utilisé pour créer le «Mon Bloc» **AllerRetour**.



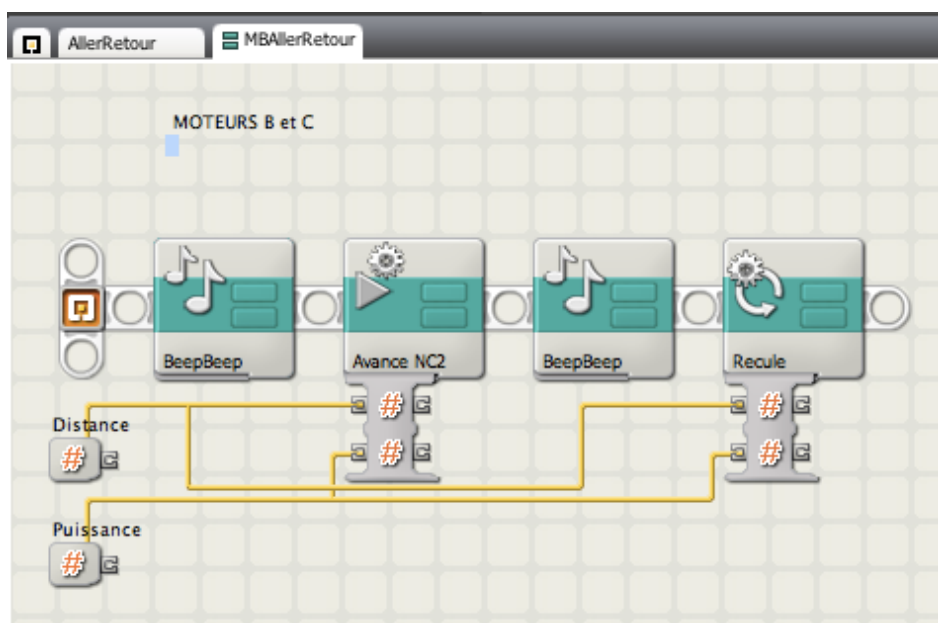
Utiliser des paramètres d'entrée à ,plusieurs reprises à l'intérieur d'un Mon Bloc

Comme indiqué dans la Figure ci-dessus, les paramètres d'entrée Puissance et Distance cm sont utilisés à l'intérieur du «Mon Bloc» par les deux «Mon Blocs» **Avance NC2** et **Recul**. Quand un paramètre de saisie est utilisé plus d'une fois à l'intérieur d'un «Mon Bloc», il est important que toutes les utilisations du paramètre soient raccordées au même bloc **Constante** (ou d'autre source de données) dans le programme qui l'utilise. pour créer le «Mon Bloc». C'est l'assurance d'une source unique de valeur de donnée pour l'ensemble des utilisations.

Rappelez-vous qu'en choisissant les blocs pour composer le «Mon Bloc», ils doivent tous être contigus, donc les blocs **Constante** dans la Figure ci dessus, doivent être placés en début de programme.

En observant la sélection des blocs dans la Figure ci-dessus qui sera coupée pour former le «Mon Bloc», vous remarquerez que quatre fils traversent la frontière de la sélection.

Toutefois, ces quatre fils représentent seulement deux fils de données différents, parce que certains fils se joignent à la même source de données.



Il aura suffi d'insérer à partir de la *Palette Personnalisée* les «Mon Bloc» précédemment créés.

Comme le «Mon Bloc» **AllerRetour** aura deux paramètres de saisie, (*Distance* et *Puissance*), deux blocs **Constante** seront utilisés comme sources de données de ces valeurs avant la création du «Mon Bloc».

On remarquera que chaque paramètre d'entrée est utilisé 2 fois.

Créer un Mon Bloc avec un port de sortie de données

Nous allons maintenant travailler sur des «Mon Bloc» qui disposent d'une sortie de données à partir de leurs plots de données. Cela permettra aux «Mon Bloc» de fournir des valeurs de résultat au programme qui l'utilise. Ces «Mon Bloc» demanderont un peu plus d'attention, sans pour cela engendrer de complications.

Le «Mon Bloc» **NbreLimit** acceptera un paramètre de saisie *Nombre* avec deux autres paramètres d'entrée *Min* et *Max*. Il testera la valeur de *Nombre* pour savoir si elle se trouve dans la plage *Min* à *Max*, et produira une valeur de sortie qui se trouve dans cette plage. Le résultat de **NbreLimite** peut être envisagé de la manière suivante:

Si *Nombre* est supérieur à *Max*, le résultat sera **Max**

Si *Nombre* est inférieur à *Min*, le résultat sera **Min**

Si *Nombre* est compris entre *Min* et *Max*, le résultat sera **Nombre**.

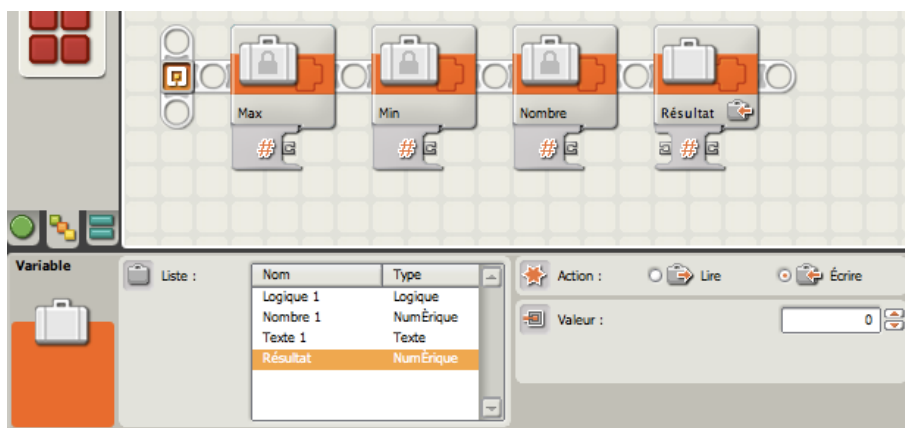
Nous allons donc commencer par installer 3 blocs **Constante** qui contiendront les données source des 3 paramètres et que nous appellerons *Min*, *Max* et *Nombre*.

Pour le résultat en sortie, nous avons besoin d'un endroit pour raccorder le fil de données à l'extérieur du «Mon Bloc» futur. Créé d'une manière identique à un port d'entrée mais inversé, un port de sortie sera nécessaire pour chaque fil de données ayant leur origine à l'intérieur de la sélection (du «Mon Bloc» futur) et dirigé vers l'extérieur de cette sélection.

Règle: Pour une sortie de données, un plot de sortie est nécessaire pour chaque fil de données ayant son origine à l'intérieur de la sélection constituant le futur «Mon Bloc». Ce fil sera dirigé vers l'extérieur ou après les blocs sélectionnés.

La manière la plus simple de créer cette connexion est d'utiliser un bloc **Variable Ecrire** qui recevra le résultat en sortie. Plus clairement, utiliser la commande *Modifier > Définir les variables* dans la barre de menu du logiciel NXT-G, pour créer une variable avec un nom significatif. Placer ce bloc **Variable Ecrire** à la fin de la sélection destinée au futur «Mon Bloc».

Nous allons donc commencer la préparation du «Mon Bloc» **NbreLimit** ainsi:



Utiliser une variable locale dans un Mon Bloc

Dans le cas d'une formule complexe pour résoudre le problème ci-dessus du *Nombre* basé sur la plage *Max > Min*, il est parfois utile de s'aider de variables temporaires pour stocker et modifier des résultats intermédiaires avant de produire le résultat final.

Si nous avons une variable temporaire intitulée *ResultatTemp*, nous pourrions écrire pour **NbreLimite** cette formule en utilisant le pseudo-code suivant:

1. Sélectionner **Ecrire** dans la variable **ResultatTemp** pour récupérer la valeur de *Nombre*
2. Comparer *Nombre* avec *Max*, et si *Nombre* est plus grand que *Max*, écrire dans *ResultatTemp* la valeur *Max*
3. Comparer *Nombre* avec *Min*, et si *Nombre* est plus petit que *Min*, écrire dans *ResultatTemp* la valeur *Min*
4. Relier la variable *ResultatTemp* à un fil de sortie de données.

La variable temporaire *ResultatTemp* est dans cette séquence utilisée à l'intérieur du «Mon Bloc» et n'a aucune utilité, ni avant ni après le travail accompli par le «Mon Bloc». Mais, il existe un problème avec cette variable temporaire à l'intérieur du «Mon Bloc». Si elle a le même nom qu'une autre variable utilisée dans le programme principal qui contient ce «Mon Bloc», cette variable temporaire s'imposera dans le programme principal ce qui pourrait créer un bug. Si certains langages de programmation peuvent créer des variables locales pour éviter ce genre d'inconvénient, ce n'est pas le cas pour le logiciel NXT-G. Aussi, nous emploierons une convention pour définir le nom de ces variables.

Remarque: Toutes les variables dans le logiciel NXT-G sont «globales», cela signifie que deux variables de même nom se réfèrent à la même adresse de stockage. On ne peut pas définir une variable «locale» garantissant le contenu d'un seul «Mon Bloc».

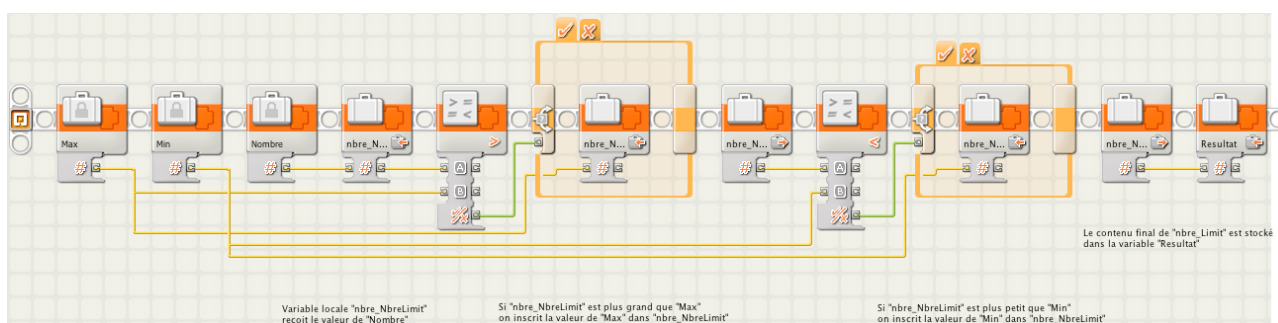
Nous avons utilisé le nom de *ResultatTemp* qui est malheureusement source de confusion dans sa désignation.

Il est préférable d'adopter les principes suivants:

1. Le nom des **variables temporaires** dans un «Mon Bloc» commence par une lettre *minuscule* et suivi du nom du «Mon Bloc», séparés par le caractère underscore (" _")
 2. Le nom des **variables normales** commence par une lettre *majuscule*.
- Cette convention évitera toute interférence avec les variables normales d'un autre programme ou d'un autre «Mon Bloc».

Dans notre cas il pourrait s'intituler: **nbre_NbreLimit**.

Voici un exemple de programme pouvant servir à la création du «Mon Bloc» **NbreLimit** et utilisant la variable temporaire *nbre_NbreLimit*. Vous remarquerez que le nom de la variable temporaire a été tronquée sur l'icône, mais elle figure dans sa globalité dans la fenêtre *Modifier les variables*.

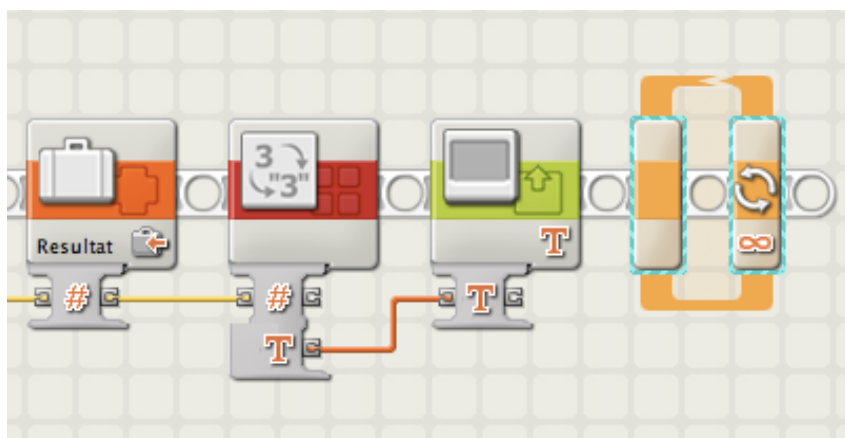


Comme nous l'avons vu dans l'exemple précédent **AllerRetour**, tous les paramètres d'entrée de même nom sont reliés par des fils de données à un même bloc **Constante**. Dans le cas d'un «Mon Bloc» imposant par sa longueur, il est parfois difficile de «tirer» convenablement des fils entre les plots. C'est le cas dans ce dernier exemple, notamment pour tirer un fil d'entrée de paramètre à partir du bloc **Constante Min**. Pour palier à cet inconvénient, vous pouvez introduire une autre variable temporaire contenant une copie de la valeur du paramètre. Dans ce cas vous utiliserez un bloc **Variable Ecrire** pour copier cette valeur au début de la sélection, puis utiliser de multiples blocs **Variable Lire** pour accéder au contenu de cette variable temporaire à l'intérieur du programme.

Remarque: Ce programme comprend quelques commentaires pour en faciliter la compréhension. Il est souhaitable d'attendre la création du «Mon Bloc» avant d'ajouter ces commentaires, car cela évite de les repositionner.

Tester un Mon Bloc avant et après l'avoir créé

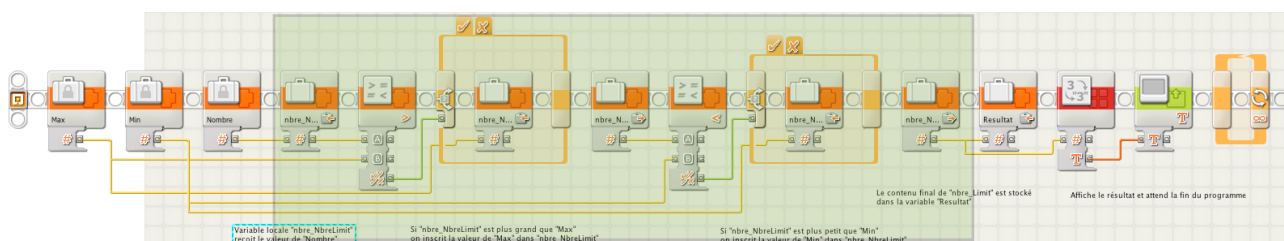
Le programme que nous venons de créer est plus complexe que ceux étudiés précédemment, et il est aussi plus difficile à tester parce qu'il ne produit aucun mouvement ni son ou affichage visible sur le robot. Pour tester ce programme, nous devons être en mesure de «voir» à la fin, la valeur de la variable *Resultat*.



Ces quelques blocs additionnels peuvent être employés pour constater le résultat.

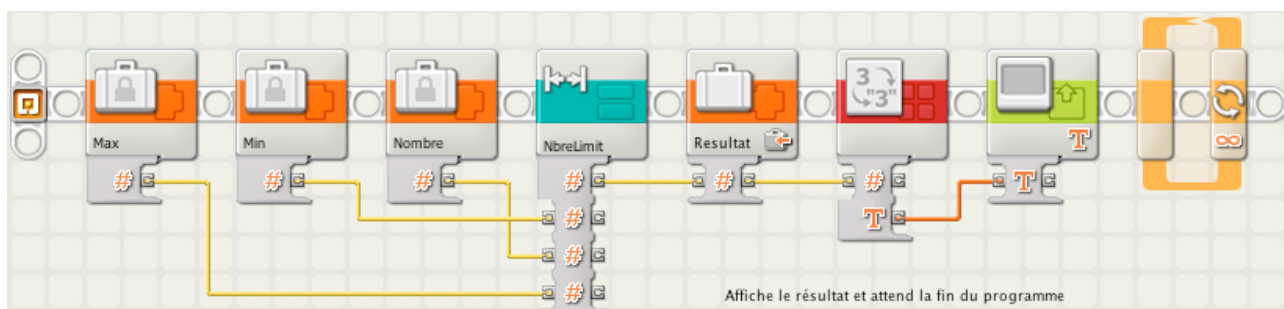
La **variable** *Resultat* est convertie en *texte* et affichée sur l'écran du NXT. Pour maintenir cet affichage, il suffit d'ajouter un bloc **Boucle toujours** qui mettra fin au programme en appuyant sur la touche *annulation*.

Ces blocs seront ajoutés en fin de programme *avant* de créer le «Mon Bloc» **NbreLimit**, puis testé à plusieurs reprises en modifiant les valeurs des constantes *Max*, *Min* et *Nombre* en début du programme. L'affichage permet de s'assurer des résultats attendus. Il est préférable de procéder à ces tests avant la création du «Mon Bloc», parce qu'une fois créé, ce «Mon Bloc» ne sera pas aisément modifiable.



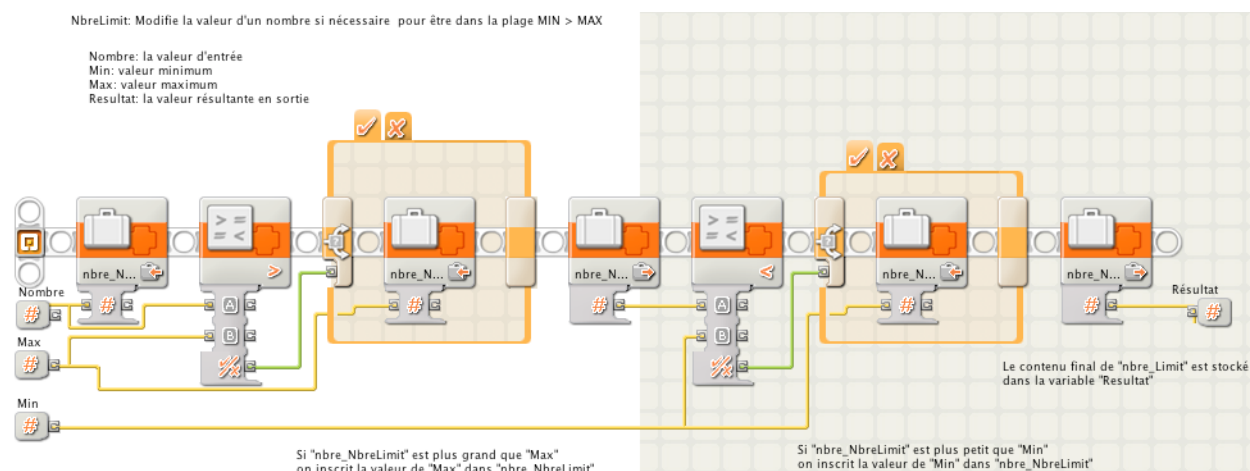
Remarque: juste avant de créer un «Mon Bloc», il est important d'enregistrer le programme initial afin d'y revenir en cas de fausse manoeuvre. Utiliser pour cela la commande «*Enregistrer sous*» et différenciez les noms des versions. Par exemple sauvegardez le programme ci-dessus sous le nom de «**NbreLimitTest1**», puis, après avoir créé le «Mon Bloc» **NbreLimit**, «*enregistrer sous*» **NbreLimitTest2**, etc..

Pour créer le «Mon Bloc» **NbreLimit** à partir du programme complet, sélectionnez les blocs après les blocs **Constante** en début et avant le bloc qui écrit le résultat en sortie dans la variable **Resultat**. Cette sélection se traduit par par 3 données en entrées qui deviennent les 3 paramètres d'entrée et une sortie en fin de la sélection. Le programme se transforme ainsi:

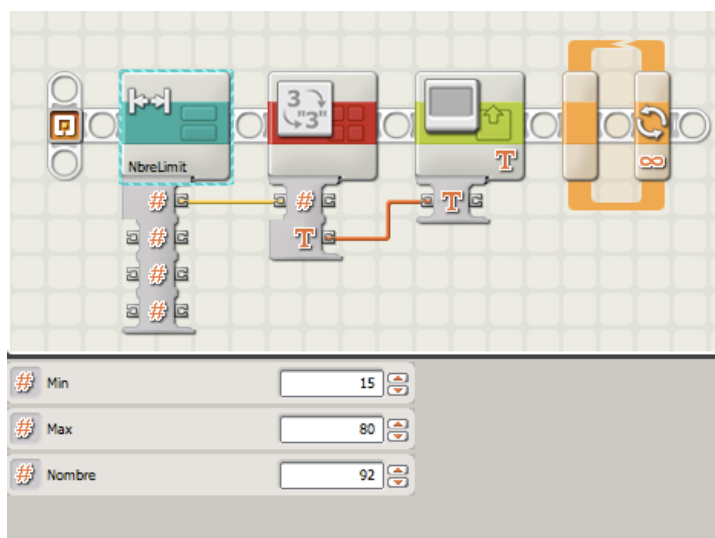


Notez que les blocs de test sont présents à la suite du «Mon Bloc» **NbreLimit**. Vous pouvez tester à nouveau ce programme pour vous assurer du résultat, qui doit être identique.

Le contenu du «Mon Bloc» **NbreLimit** se présente ainsi:



Les paramètres ont été renommés et un commentaire a été ajouté en tête. Notez que le plot de sortie résultant à la même représentation que les plots d'entrée et qu'il peut être également renommé.



A titre de test final du «Mon Bloc» **NbreLimit**, vous pouvez créer un nouveau programme contenant ce «Mon Bloc» relié directement aux blocs de test pour essayer différentes configurations, et en entrant directement les valeurs dans le panneau de configuration.

Méthode pas-à-pas de création d'un «Mon Bloc» complexe

Comme nous l'avons vu avec le «Mon Bloc» **NbreLimit**, il y a plusieurs étapes à franchir dans la création d'un «Mon Bloc» complexe, ainsi que des moments cruciaux pendant le déroulement pour sauvegarder et tester le travail.

Ce qui suit résume pas à pas cette approche que vous pourriez utiliser comme un guide dans la création et les essais d'un «Mon Bloc» complexe.

1. Définir les paramètres, des plots de données en sortie et les variables que le «Mon Bloc» utilisera pour communiquer avec les programmes qui les contiendront. On ne peut changer ni le nombre ni le type de paramètres ni les plots après la création d'un «Mon Bloc». Prendre donc le temps de bien réfléchir sur le choix de ces éléments.
2. Créer un nouveau programme qui sera utilisé pour tester et créer le «Mon Bloc». Appelons-le **TestProg**.
3. Dans le programme **TestProg**, insérer en début du programme un bloc **Constante** Personnalisé pour chaque paramètre en entrée et un bloc **Variable Écrire** pour chaque port de données en sortie à la fin du programme.
4. Définir n'importe quelles variables provisoires à utiliser dans le «Mon Bloc» d'une manière conventionnelle afin d'éviter tout conflit de nom avec d'autres variables. A titre d'exemple et par convention, nommer les variables provisoires en commençant par une lettre minuscule accolé au nom du «Mon Bloc», et séparé par le caractère underline ("_"). Garder une majuscule de début pour les variables normales.
5. Créer le contenu du «Mon Bloc» dans **TestProg** comme une séquence de blocs entre les paramètre de saisie des blocs **Constante** et les ports de sortie des blocs **Variable**.
6. S'assurer que si un paramètre de saisie est utilisé à plusieurs reprises, tous les fils de données sont reliés au même plot de données menant au bloc **Constante** original. Si un fil de données devient trop long ou complexe, employer une variable provisoire intermédiaire pour stocker la valeur d'un paramètre de saisie.
7. Ajouter des blocs de test au début ou à la fin du **TestProg** pour vérifier les résultats attendus.
8. Exécuter plusieurs variantes d'essais, en modifiant les valeurs d'entrée dans les blocs **Constante**, pour s'assurer des bonnes réponses dans tous les cas. Faire toutes modifications nécessaires au **TestProg**.
9. Faire une copie de **TestProg** à l'aide de la commande «Enregistrer sous». Vous aurez ainsi une sauvegarde avant de créer les «Mon Bloc».
10. Choisissez l'ordre des blocs qui composeront le contenu de «Mon Bloc». Cela devrait inclure tous les blocs compris entre les blocs **Constante** et les blocs **Variable Écrire** et ne devrait pas inclure les blocs ajoutés pour des tests supplémentaires.
11. Cliquer «Créer mon Bloc» sur la barre d'outils et utiliser le «créateur de bloc «Mon Bloc» pour préciser un nom et choisir une icône.
12. Vérifier que le **TestProg** modifié se présente correctement après la substitution de la séquence par le «Mon Bloc», en particulier les fils de données pour les paramètres d'entrée et de sortie. Procéder à des essais complémentaires si nécessaire.
13. Double cliquer sur le «Mon Bloc» du **TestProg** pour ouvrir son contenu. Renommer les paramètres d'entrée et de sortie, si on le souhaite; ajouter un commentaire général en tête accompagné d'autres précisions sur le contenu. Repositionner. éventuellement tous les fils de données. Si certains sont illisibles, les supprimer et les réinstaller.
14. Sauvegarder le «Mon Bloc» à l'aide de la commande «Enregistrer».
15. Créer un nouveau programme pour un test final du «Mon Bloc». Insérer une nouvelle copie à partir de la Palette personnalisée et ajouter tous les blocs nécessaires à une série complète d'essais supplémentaires.

En guise de conclusion...

Comme vous avez pu le constater, l'utilisation des «Mon Bloc» simplifie et clarifie considérablement la programmation en NXT-G. Mais cela demande aussi une certaine rigueur dans le raisonnement et de la méthode pour la mise en forme.

Pour terminer en beauté ce chapitre, je vous propose 2 exemples qui vous seront utiles pour l'évolution de vos futurs robots, le premier se contentera de compter en manipulant les boutons du NXT, le second interviendra sur les moteurs.

Premier exemple: un compteur.

Enoncé du problème: il s'agit d'afficher sur l'écran du NXT une valeur qui croît ou décroît en appuyant sur les touches «Flèche Droite» ou «Flèche Gauche». La valeur choisie est confirmée en appuyant sur le bouton rouge de «Validation». Cette valeur pourra par la suite être reprise à des fins diverses (choix d'une vitesse ou d'un moteur, choix d'un menu, choix d'une couleur ou une distance, etc.).

Nous procéderons par étape.

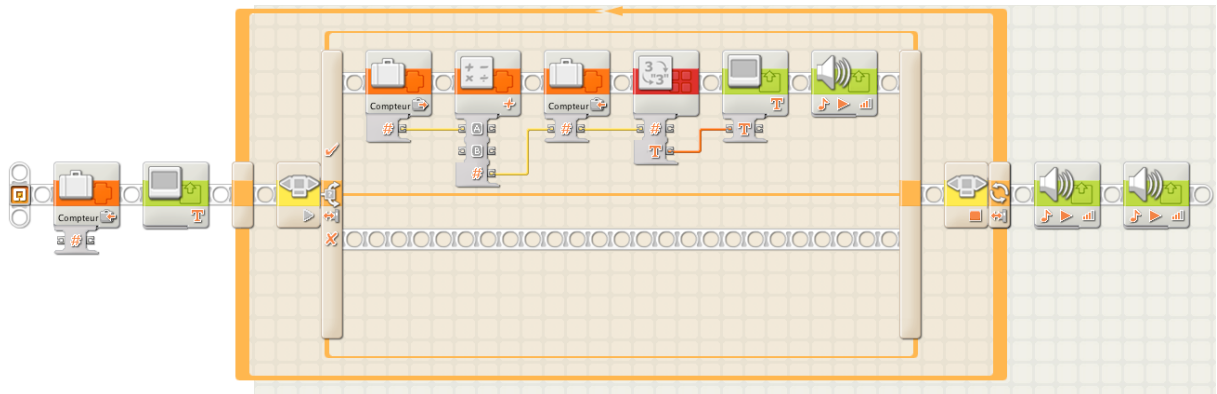
1. Ecrire le premier programme capable de répondre à une partie du problème: ici faire afficher une valeur en utilisant seulement la Flèche Droite.

On utilisera un bloc **Variable numérique** intitulée *Compteur* qui sera lue au départ. rappelez-vous qu'une variable lue pour la première fois sans être au préalable écrite a zéro comme valeur de défaut.

Pour faire croître cette valeur d'une unité à chaque heurt du bouton, il faut 2 choses:

- * Il faut attendre que cette action soit déclenchée grâce à un bloc **Commutateur** (comprenant le module de comptage, activé par le bouton «Flèche Droite») inclus dans un **Boucle** qui maintient cette attente (dont la sortie est commandée par la touche «Validation» suivi de tops sonores).
- * Une fois l'action déclenchée, ajouter une unité dans la Variable *Compteur*, grâce à un bloc **Math** puis attendre à nouveau la suite des opérations.

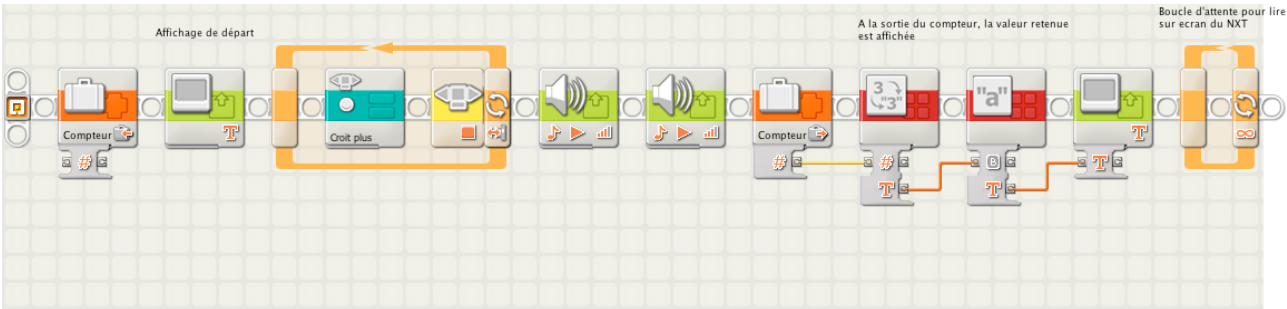
Pour bien visualiser le processus, nous ajouterons quelques blocs d'affichage (au départ et à l'intérieur du module de comptage) et de sons pour ponctuer la progression.



On remarque qu'il ne se passe rien tant qu'on n'agit pas sur les boutons.

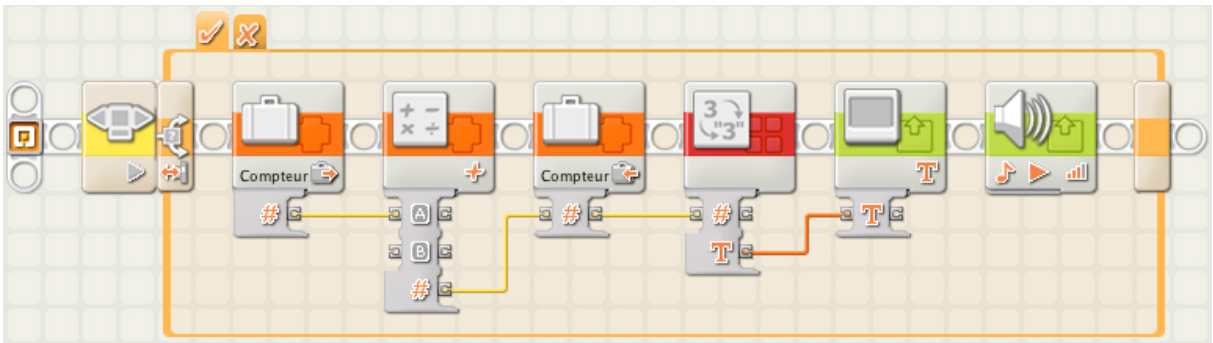
Ce programme est enregistré sous le nom de **ProgTest**.

2. Enregistrer sous... à nouveau ce programme en lui donnant le nom de **ProgTest1**. **Ce programme sera modifié** (on conserve la version précédente).
3. Sélectionner le bloc **Commutateur** et créer un «Mon Bloc» **Croît plus**. le programme **ProgTest1** est modifié de la sorte, après avoir ajouté quelques blocs en fin de séquence pour expliciter le résultat enregistré.

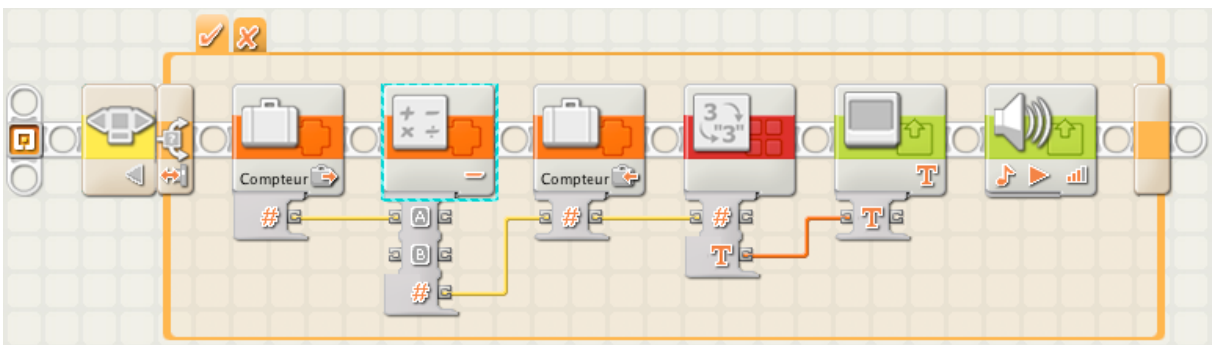


Aucun fil de données n'apparaît en sortie le la boucle, ce qui est normal puisque le comptage est interne. Le Bloc **Variable Lire Compteur** dans la suite de la boucle ne fait que reprendre la dernière valeur validée pour l'afficher. Tester le programme.

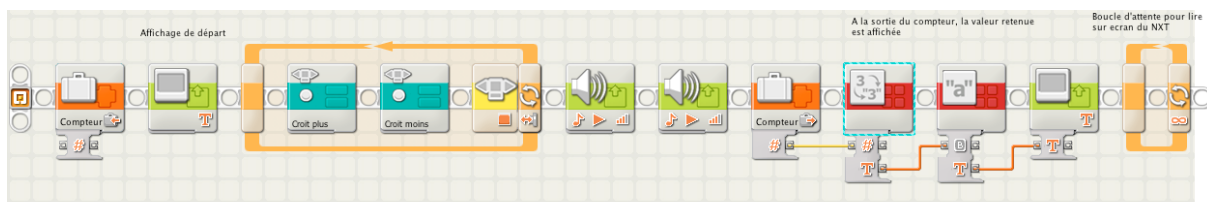
4. Enregistrer ce programme **ProgTest1** pour le sauvegarder.
5. Enregistrer sous... à nouveau ce programme en lui donnant le nom de **ProgTest2**.
Ce programme sera modifié.
6. Ouvrir le «Mon Bloc» **Croit plus** en double cliquant sur son icône.



7. Enregistrer sous... ce «Mon Bloc» en lui donnant le titre de **Croît moins. Ce «Mon Bloc» sera modifié pour être transformé en moins.**
8. Procéder aux modifications en choisissant de bouton *Flèche Gauche* dans le bloc **Commutateur** ainsi que le bloc **Math** qui devient - (moins, soustraction).



9. Enregistrer ce «Mon Bloc» **Croit moins** pour le sauvegarder. Il vient s'ajouter automatiquement dans la *Palette Personnalisée > Mes blocs*.
10. Revenir sur le programme **ProgTest2**.et procéder à l'ajout du «Mon Bloc» **Croit moins**.



Tester le programme.

11. Enregistrer le programme **ProgTest2** dans sa version (presque) définitive. Je dis presque, parce que je serais tenté de créer un «Mon Bloc» **Affichage** à la suite de la boucle pour clarifier encore plus le résultat. Mais là, je vous laisse faire; je pense que vous avez compris le mécanisme.

Deuxième exemple: la conduite d'un véhicule contrôlé par un capteur à ultrasons



Pour que ce programme soit vérifiable, il est nécessaire de l'installer sur un véhicule équipé en conséquence (à construire) et disposant des caractéristiques suivantes:

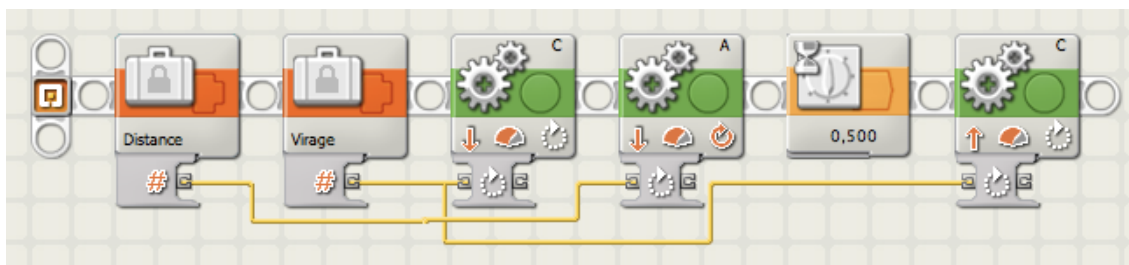
- la propulsion est fournie par un moteur A entraînant deux roues à l'arrière
- la direction est assurée par un moteur C activant la barre de direction
- Le capteur à ultrasons est solidaire des roues directrices de manière à suivre le mouvement des roues.

J'ai construit ce véhicule pour l'exemple, vous pouvez vous inspirer de projets déjà réalisés à

trouver sur le site de Mindstorms Lego par exemple. (<http://mindstorms.lego.com>).

Enoncé du problème: il s'agit de faire avancer le véhicule jusqu'à ce qu'il détecte un obstacle situé à 30 cm du capteur. Lorsque cette distance est vérifiée, le véhicule s'arrête, braque les roues directrices soit à droite, soit à gauche (mouvement aléatoire), fait une marche arrière suffisante, revient au neutre, et repart dans une autre direction.

1. Première étape, définir le module de base:



Les blocs **Constante Distance** et **Virage** produiront les valeurs paramètres suffisantes pour une bonne manoeuvre (pour l'instant fixées aux valeurs indiquées). Le bloc **Déplacer C** produit une rotation inversée de 45 degrés (virage à gauche) du moteur C. Le bloc **Déplacer A** produit deux rotation inversée soit 720 degrés (recul) du moteur A. Le bloc **Attendre** 0,500 seconde laisse le temps au moteur C de

revenir à sa position initiale. Enfin le bloc **Déplacer C** produit une rotation normale de 45 degrés pour faire revenir le moteur C en position de départ.

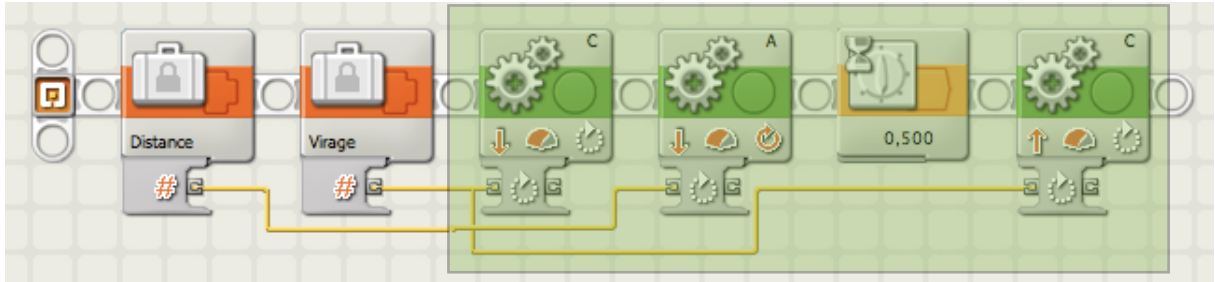
Les fils de données vont nous servir à définir les 2 paramètres *Distance* et *Virage* exprimés en degrés.

Enregistrer ce programme sous le nom **VirageTest**.

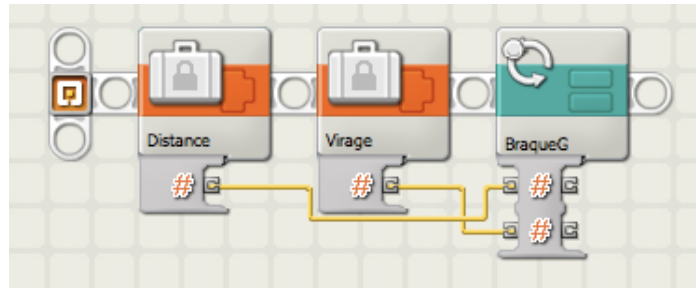
2. Enregistrer **VirageTest** sous un nouveau nom **VirageTest1**.

Ce programme sera modifié (on conserve la version précédente).

3. Sélectionner la partie qui sera un «Mon Bloc» puis créer le «Mon Bloc»**BraqueG**.



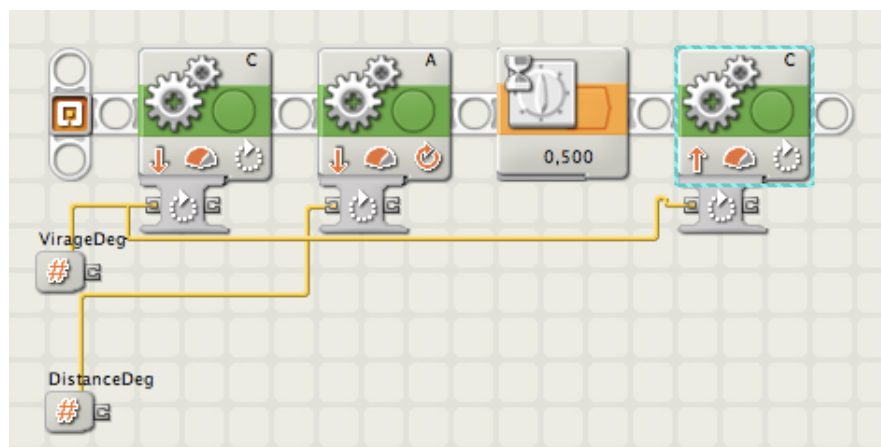
VirageTest1 devient ce programme. Enregistrer le pour le sauvegarder.



4. Enregistrer sous...à nouveau ce programme en lui donnant le nom de **VirageTest2**.

Ce programme sera modifié.

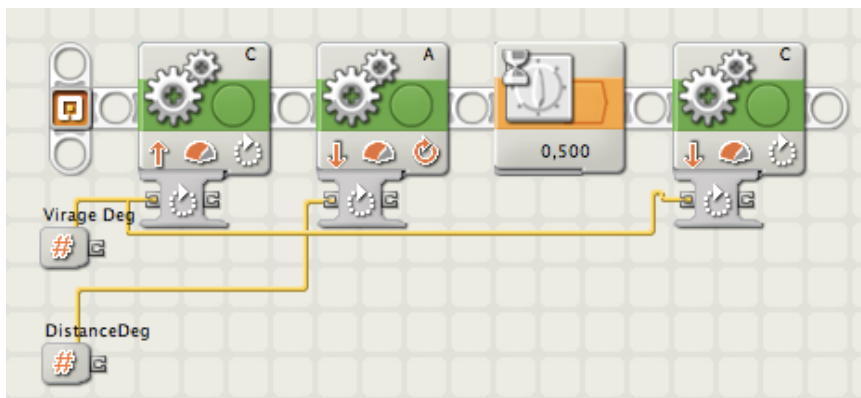
5. Ouvrir le «Mon Bloc» **BraqueG** en double cliquant sur son icône. Modifier les étiquettes des plots de paramètres.



Nous allons en faire une copie pour créer son miroir D

6. enregistrer sous... ce «Mon Bloc» en lui donnant le titre de **BraqueD**. **Ce «Mon Bloc» sera modifié pour être transformé en Droite.**

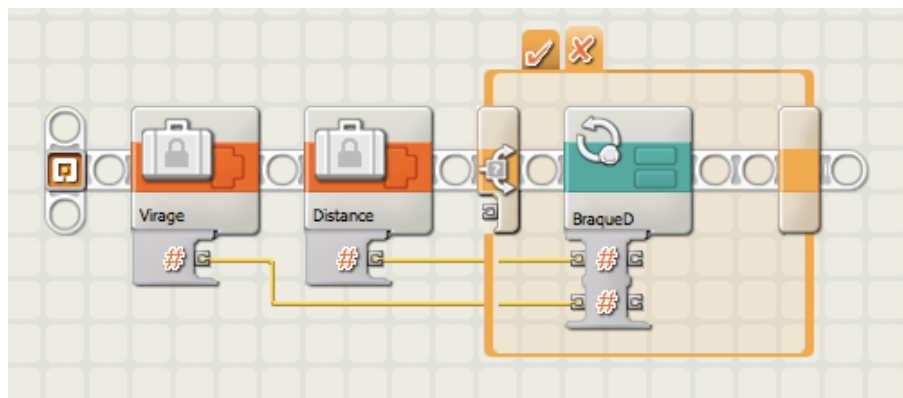
7. Modifier les étiquettes des plots de paramètres ainsi que les sens de rotation des 2 blocs **Déplacer C**. Songer également à modifier l'icône.



Enregistrer ce «Mon Bloc» **BraqueD** pour le sauvegarder. Il vient s'ajouter automatiquement dans la *Palette Personnalisée > Mes blocs*

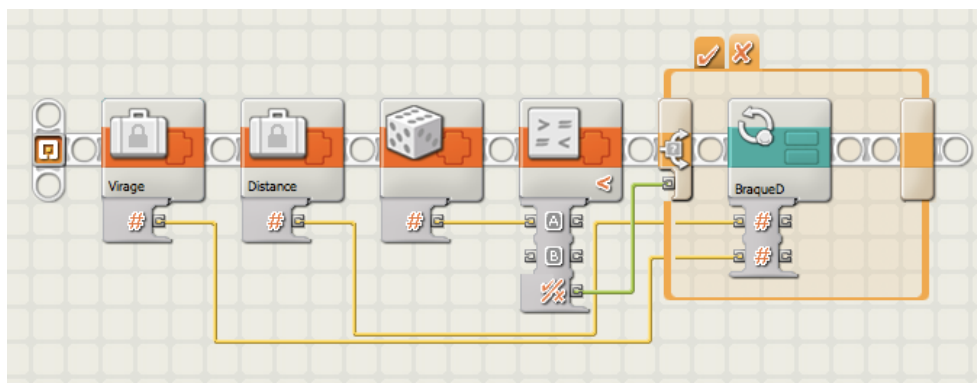
Nous disposons maintenant des 2 «Mon Bloc» **BraqueG** et **BraqueD**.

8. Revenir sur le programme **VirageTest2**.et enregistrer le sous...**VirageTest3**, puis procéder à l'ajout d'un bloc **Commutateur valeur Logique**, et placer les «Mon Bloc» **BraqueG** et **BraqueD** dans chaque séquence de condition. Brancher également les fils de données aux blocs **Constante**.



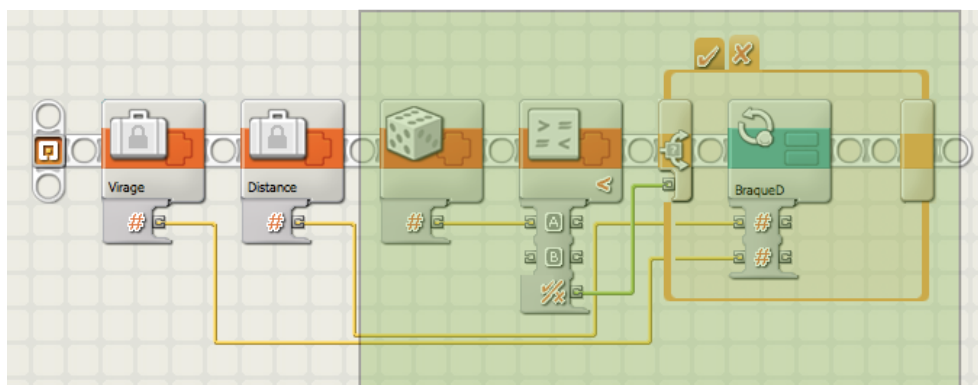
Enregistrer de programme.

9. Enregistrer sous...ce programme en **VirageTest4**
Ce programme sera modifié
10. Ajouter au programme **VirageTest4** les blocs **Aléatoire** et **Comparer** pour créer la commutation aléatoire. Brancher à nouveau les fils de données.

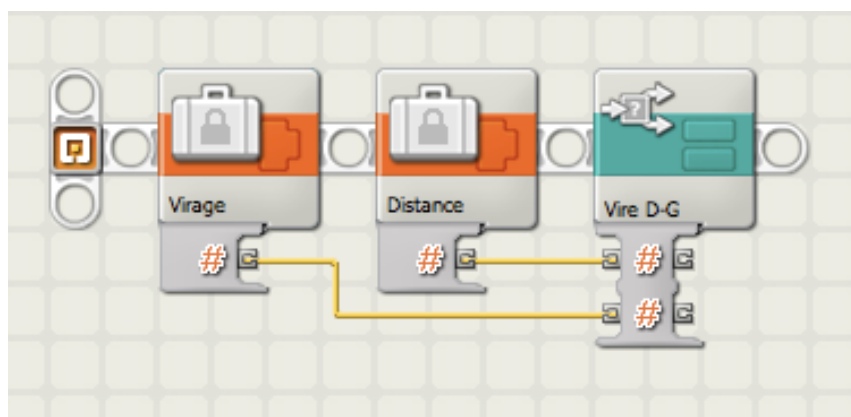


Enregistrer ce programme après tests.

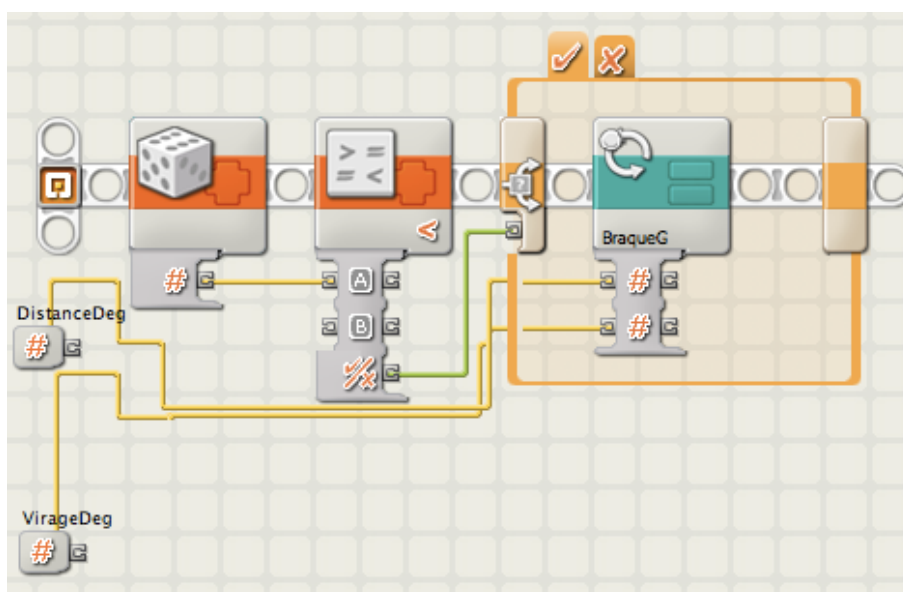
11. Enregistrer sous...ce programme en **VirageTest5**
Ce programme sera modifié
12. Sélectionner les 3 derniers blocs pour créer un «Mon Bloc» intitulé **Vire D-G**.



Le programme **VirageTest5** se présente sous cette forme:

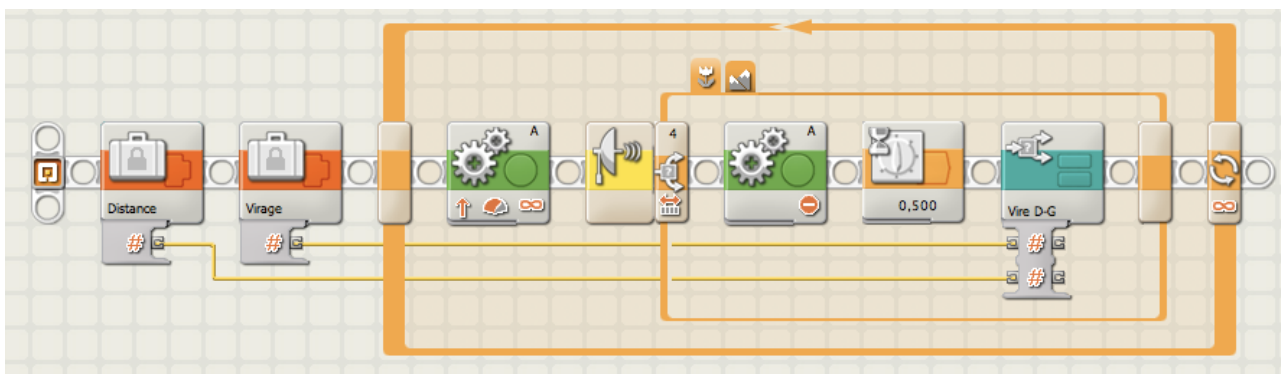


13. Enregistrer le programme **VirageTest5**.
Double-cliquer sur l'icône **Vire D-G** pour ouvrir le «Mon Bloc» qui se présente ainsi:
Modifier les étiquettes des paramètres et observer les fils de données, puis enregistrer ce «Mon Bloc» **Vire G-D**.

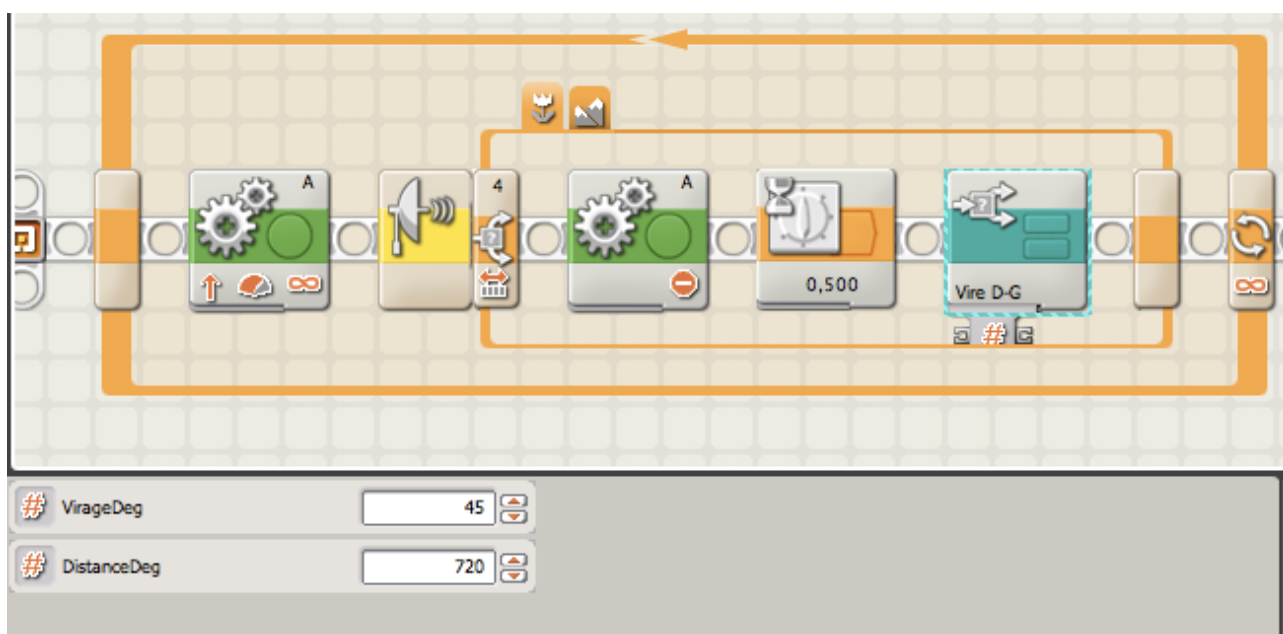


14. Enregistrer sous... le programme **VirageTest5** en **VirageTest6**
Ce programme sera modifié.
15. Les modifications à apporter sont les suivantes:

Dans une **boucle pour toujours** (pour assurer la continuité des actions), placer un bloc **Déplacer A Illimité** pour que le véhicule avance en permanence. Le bloc **Commutateur Capteur US** placé à la suite, scrute la distance d'un obstacle situé à l'avant. Tant que la distance est supérieure à 30 cm, la direction est bloquée en position neutre, l'avance se poursuit (séquence de condition Bloc **Déplacer C Stop**). Si la distance devient inférieure à 30 cm, c'est alors que le test de virage intervient. C'est la partie visible du programme. Le véhicule s'arrête puis réalise les manoeuvres signalées en début de l'exemple, d'une manière aléatoire, c'est-à-dire braquage à droite ou à gauche, puis reprend son parcours dans une autre direction.



Les 2 **constante** de début peuvent être supprimées pour faire des tests; il suffit de saisir les valeurs directement dans le panneau de configuration du «Mon Bloc» **Vire D-G**.



Ces valeurs dépendent de la construction du véhicule, et des rapports d'engrenages. Ils sont à déterminer par des tests.

En résumé et pour la bonne compréhension de l'exemple, le programme **VirageTest6** est celui à conserver; il est utilisable pour n'importe quel véhicule construit selon les caractéristiques fixées en début du §; tous les autres ne servent qu'à reprendre éventuellement le travail à l'endroit d'un bug. Les 3 «Mon Bloc» **BraqueG**, **BraqueD** et **Vire D-G**, se retrouvent dans la *Palette personnalisée «Mes Blocs»* et sont de ce fait réutilisables pour d'autres programmes.



10 – Un grand classique Le «Suiveur de Ligne» (màj 1) 2

Il existe une multitude de programmes écrits pour des robots, capables de suivre un tracé au sol, quel que soit sa forme. Chacun a ses propres astuces, et si vous n'avez encore jamais essayé, je vous propose d'en découvrir deux.

Le premier est une approche simple, qui permet au robot de se déplacer en "zigzag". Le second est un peu plus complexe et fait appel à la notion "Proportionnelle", avec un résultat plus fluide et un déplacement plus rapide.

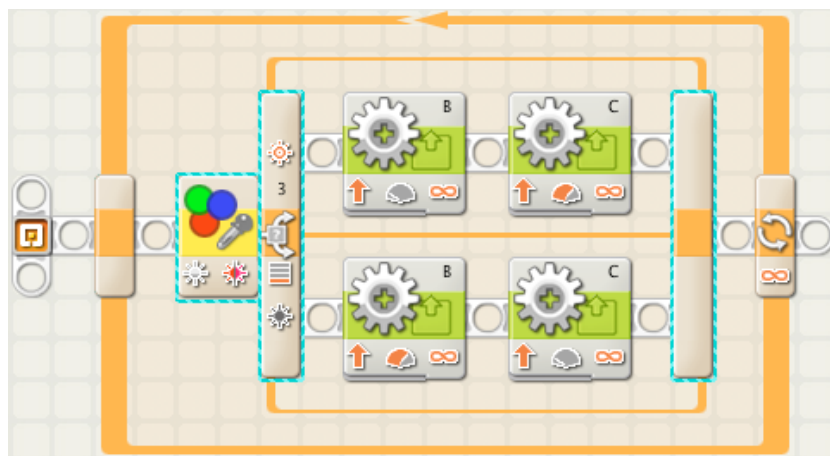
Pour débuter Vous allez construire un robot à 2 roues directrices avant et une roue folle arrière. Vous disposerez le capteur de couleurs à l'avant entre les roues directrices, placé le plus bas possible à environ 3mm du sol. Si vous n'avez pas d'idée, vous pouvez réaliser le modèle ci-dessus de Dave Parker, accessible sur son site:

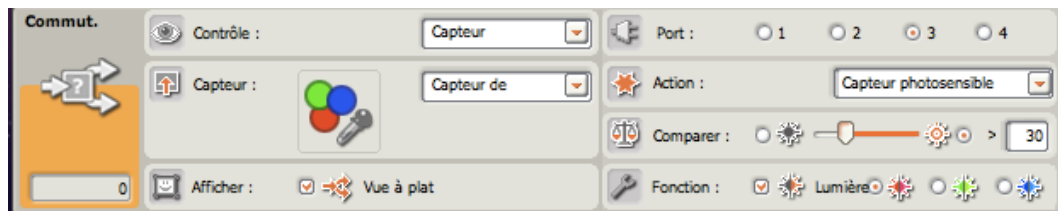
http://www.nxtprograms.com/NXT2/line_follower/index.html

Pour tester le parcours, vous utiliserez le Test Pad fourni dans le set # 8547. Une fois le programme mis au point, vous pourrez tracer votre propre circuit dans toutes les formes imaginables en évitant les virages trop serrés dits en «épingle à cheveu».

Un Suiveur de Ligne Simple

Voici un programme simple Suiveur de Ligne à "Deux Etats", faisant appel au Capteur de Couleurs utilisé en mode Photosensible.





Dans une **Boucle** "Pour toujours", on trouve un Bloc **Commutateur** contrôlé par le **Capteur de Couleurs** en mode **Photosensible**. Le capteur mesure le niveau lumineux et le compare à une valeur seuil: ici fixée à 30.

Ce seuil est calculé approximativement comme étant la demie somme des valeurs lues du minimum (milieu de la ligne noire), et du maximum (hors la ligne, sur un sol de teinte claire). On pourrait écrire un programme pour déterminer ces valeurs et même l'afficher pour un éventuel contrôle.

Nous verrons dans le cas suivant comment écrire un tel programme à partir d'un Mon Bloc de calibrage.

Le seuil étant la moyenne de ces deux nombres, le robot cherchera le bord à droite de la ligne, le capteur se positionnant à cheval, pour moitié sur la ligne et moitié sur le sol. Il utilisera ses deux moteurs B et C dont les alimentations (Puissance) sont fixées respectivement à 10 et 50 dans la première condition, ou inversés 50 et 10 dans l'autre condition du bloc **Commutateur**.

Si la valeur lue est plus grande que le seuil, le robot se trouve en dehors à droite de la ligne; il réagira par une correction à gauche en accélérant le moteur C et en ralentissant le moteur B. Si la valeur lue est plus petite que le seuil, le robot s'est trop éloigné du bord et se rapproche du centre de la ligne. Il corrigera sa trajectoire dans l'autre sens, en accélérant le moteur B et en ralentissant le moteur C.

Les relevés du capteur et les corrections de direction sont reproduites rapidement dans la boucle (plusieurs fois par seconde).

Trois choses dans ce programme peuvent être facilement modifiées pour adapter votre robot à de meilleures conditions de conduite :

1. le niveau lumineux-seuil : le Bloc Capteur de Couleurs "Compare" la valeur relevée à la moyenne Min et Max (valeurs à modifier). Cela corrige le programme en fonction de la luminosité réelle relevée sur la ligne d'évolution.
2. la vitesse maximale: dans les deux blocs "rapides" **Moteur** (réglés ici sur 50), Le niveau de Puissance utilisé détermine la vitesse maximum du robot. Si cette vitesse est trop élevée, le robot ratera les virages serrées.
3. les virages serrés: dans les deux blocs "lents" du Moteur (réglés ici sur 10), Le niveau de Puissance utilisé détermine la manière dont le robot agit pour accomplir les virages. Plus la Puissance est réduite, et plus le robot pourra prendre des virages plus serrés, alors que plus la puissance est élevée, et plus il pourra accélérer la conduite sur les portions rectilignes.

Notez que "Illimité" est utilisé comme Durée pour tous les blocs **Moteur**.

Cela permet aux blocs **Moteur** d'assurer la fluidité du mouvement pour l'étape suivante sans provoquer de secousses. Cela permet aussi au Capteur de Couleurs de procéder à

des relevés aussi rapidement que possible, afin que la direction réagisse de même aux variations sur la ligne.

En jouant sur ces valeurs par approximations et tests successifs, chaque robot peut disposer d'un réglage approprié.

Un Suiveur de Ligne «Proportionnel»

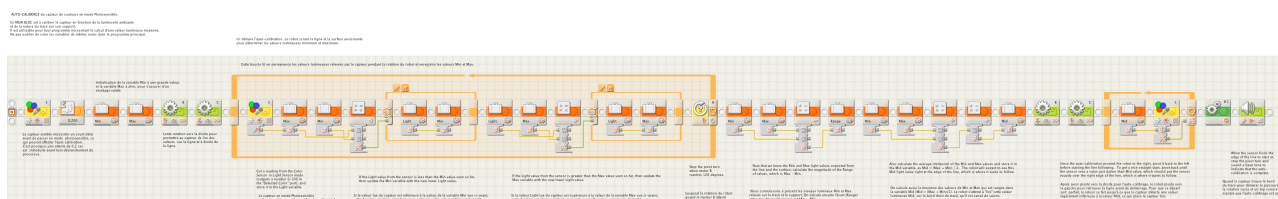
La méthode que nous allons suivre s'inspire d'un PID simplifié (voir leçon n° 14 Un contrôleur PID pour Robots - Guide de programmation NXT-G version 1.1, chapitre 2 page 168) et que nous allons développer.

Je l'ai appelé "Suiveur de ligne Proportionnel" et il comporte un paramétrage automatique du capteur de couleurs en mode Photosensible.

Le robot est sensé démarrer avec le capteur positionné dans l'axe de la ligne. Il balayera la ligne et la surface sur la droite de la ligne pour auto-calibrer au départ les niveaux lumineux ambiants; il suivra ensuite, le bord droit du tracé.

Auto-calibrage lumineux:

La partie d'auto-calibrage du programme est en fait un MonBloc intitulé *Auto-calib_Photo*, où le robot scrutera la ligne et la surface voisine pour déterminer les valeurs de luminosité minimale et maximale.



(Voir plus bas pour le téléchargement).

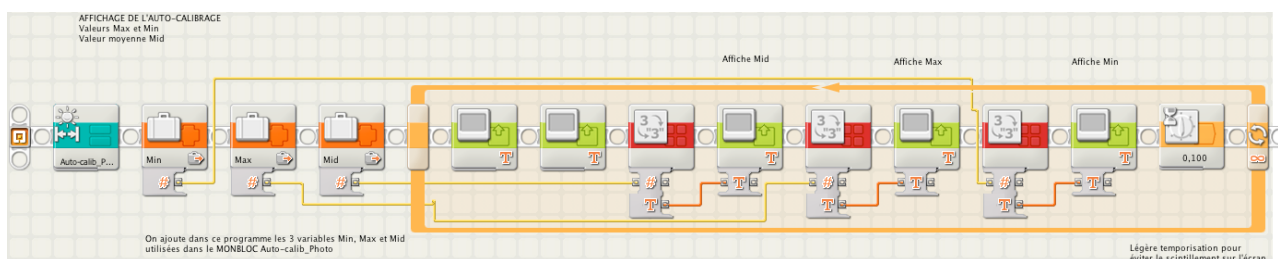
Remarque: la lampe du capteur photosensible semble avoir un petit retard avant l'allumage, qui pourrait affecter le calibrage auto. C'est pourquoi une petite temporisation de 0.2 seconde lui est accordée en début du MonBloc.

- * 1 - Le robot commence par une rotation lente à droite afin que le capteur puisse parcourir l'épaisseur de la ligne et une partie à l'extérieur droite hors la ligne. On utilise 3 variables, **Min**, **Max** et **Light**, pour jouer le rôle de «contenants» pendant les relevés successifs.
- * 2 - Le Capteur de Couleurs en mode Capteur Photosensible fournit une lecture (une valeur comprise entre 0-100 provenant du plot "Couleur Détectée") qui est stockée dans la variable **Light**.
Light contiendra la valeur instantanée du capteur qui sera comparée en permanence aux valeurs Min et Max, fixées arbitrairement en tête du MonBloc. En initialisation, la variable **Min** aura une grande valeur (9999), et la variable **Max** sera égale à zéro (0) de façon à ce que le balayage fournisse toujours une valeur.
- * 3 - Si la valeur **Light** du capteur est inférieure à la valeur **Min** (9999), alors on remplace la valeur de la variable **Min** par cette plus basse valeur de **Light**.
- * 4 - Si la valeur **Light** du capteur est supérieure à la valeur **Max** (0), alors on remplace la valeur de la variable **Max** avec cette plus haute valeur de **Light**.

- * 5 - La rotation du robot s'arrête quand le moteur B atteint 120 degrés.
 - * 6 - Maintenant que nous connaissons les valeurs **Min** et **Max** relevées sur la ligne et la surface, nous calculons l'**écart** des valeurs dans la plage **Max - Min**.
 - * 7 - Calculons aussi la moyenne des valeurs **Min** et **Max** et stockons-la dans la variable **Mid** ($\text{Mid} = (\text{Max} + \text{Min}) / 2$).
- Le robot "comprendra" que cette moyenne de la valeur lumineuse calculée se situe sur le bord droit de la ligne, qui est précisément ce qu'il cherche à suivre.
- * 8 - Puisque l'auto-calibrage a fait pivoter le robot à droite, il revient à gauche pour se retrouver au départ de la ligne.
- Pour obtenir un bon démarrage, il pivote en sens inverse jusqu'à ce que le capteur «lise» une valeur juste plus basse que la moyenne **Mid**, ce qui devrait mettre le capteur exactement à cheval, sur le bord de la ligne à suivre.
- * 9 - Quand le capteur trouve le bord de la ligne pour démarrer la trajectoire, le robot s'arrête et envoie un signal sonore pour indiquer que l'auto-calibrage est terminé.

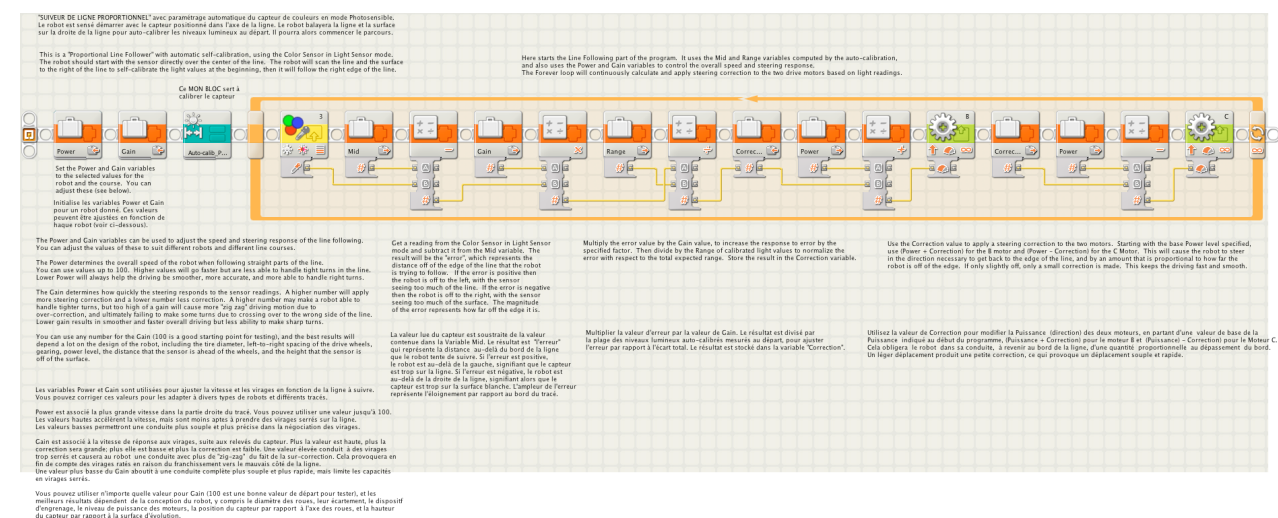
Si vous souhaitez connaître ces valeurs, il suffit d'écrire un petit programme indépendant, qui utilisera ce MonBloc avec un affichage des valeurs.

A titre d'exemple, il pourrait ressembler à ceci:



Ce programme est intitulé **Test Auto-calib Lum** (Voir plus bas pour le téléchargement).

Le programme principal



Il est intitulé **SmoothFollowNC**. Il nécessite quelques explications. (voir plus bas pour le téléchargement).

Vous remarquerez en tête deux variables **Power** et **Gain**, qu'il faut initialiser.

Ces variables **Power** et **Gain** sont utilisées pour ajuster la vitesse et la direction du suiveur de ligne.

Mettez dans ces variables les valeurs les mieux adaptées à votre robot et son parcours. Les valeurs qui sont ici retenues correspondent au robot que j'ai réalisé pour cette démonstration. En rectifiant ces valeurs vous les adapterez à des robots et à des parcours de ligne différents.

Vous pouvez fixer une valeur arbitraire pour **Gain** (100 est une bonne valeur de départ pour tester), et les meilleurs résultats dépendent de la conception du robot, y compris le diamètre des roues, leur écartement, le dispositif d'engrenage, le niveau de puissance des moteurs, la position du capteur par rapport à l'axe des roues, et la hauteur du capteur par rapport à la surface d'évolution. Ne soyez pas surpris des résultats au premier essai, il faudra les corriger.

Power détermine la plus grande vitesse du robot quand il suit des parties rectilignes du tracé. Vous pouvez utiliser des valeurs jusqu'à 100. Les valeurs hautes augmenteront la vitesse, mais elles rendront plus difficile la négociation des virages serrés. Les valeurs basses donneront toujours une conduite plus douce, plus précise et fourniront une meilleure approche des virages.

Juste après ces blocs Variables, commence la partie du programme du suiveur de Ligne. Il utilise les variables **Mid** et **Gain** calculées par l'auto-calibrage et aussi utilise les variables **Power** et **Gain** pour contrôler la vitesse et les corrections de direction.

La boucle "Pour toujours" calculera continuellement et appliquera la correction de direction aux deux moteurs de propulsion basées sur les lectures des valeurs lumineuses.

La valeur de la variable **Correction** modifiera la *Puissance* (direction) des deux moteurs, en partant d'une valeur de base **Power** indiqué au début du programme, (**Power + Correction**) pour le moteur B et (**Power - Correction**) pour le Moteur C.

Cela obligera le robot dans sa conduite, à revenir sur le bord de la ligne, d'une quantité *proportionnelle* au dépassement du bord. Un léger déplacement produit une petite correction, ce qui provoque un déplacement souple et rapide.

Pour faciliter la détermination de ces deux variables, je vous propose la formule simple permettant de les calculer par approches successives.

$$\text{Correction} = \frac{\text{Gain} * (\text{Light} - \text{Mid})}{\text{Range}}$$

L'expression (Light-Mid) est en fait l'erreur du PID, et le rapport [(Light-Mid)/Range] est un coefficient proportionnel K_{light} qui s'applique sur Gain.

Avec:

Light = niveau lumineux variable relevé en permanence par le capteur qui suit la ligne.

Mid est déjà calculé par le MonBloc, ainsi que **Range**.

Exemple avec un robot sur roues 56x26. :

J'ai utilisé ici un tableur pour faciliter les calculs, dans le contexte de la démonstration.

MIN	0
MAX	57
MID	28,5
RANGE	57
LIGHT	50
PUISSANCE	35
GAIN	55
CORRECTION	21
Moteur B	14
Moteur C	56

Min et **Max** sont les valeurs relevées pour calculer **Mid** et que l'on peut trouver en utilisant le programme *Test Auto-calib Lum.rbt*.

J'ai supposé que le capteur avait débordé sur la droite, avec une valeur Light relevée de 50

En modifiant les valeurs de **Power** (Puissance) et **Gain**, le tableur calcule les puissances correspondantes des moteurs B et C dans ce contexte.

L'objectif est de trouver la plus faible **Correction** avec un meilleur **Power** possible

Des tests successifs en modifiant ces deux valeurs, et seulement ces deux dans le programme principal permettent de vérifier le déroulement des parcours et d'en choisir le meilleur.

Note: le robot utilisé est différent de celui proposé par Dave PARKER dans la mesure où la conception est inversé. Cela explique que le parcours se fait dans le sens des aiguilles d'une montre (bord extérieur du tracé), donc bord gauche.

Pour le téléchargement des programmes

Programme principal (avec son MonBloc): files.me.com/roboleo/yxrk6/

Affichage de l'Auto calibrage: files.me.com/roboleo/linyrk

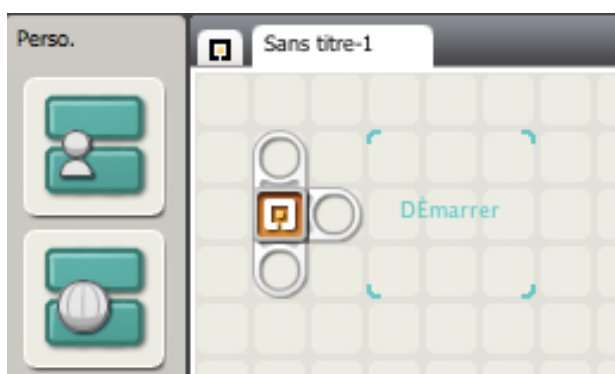


11 – La Palette personnalisée

Vous avez utilisé d'une manière progressive les Palettes *commune* et *entière*. C'est dans ces palettes que résident les blocs standards constituant les programmes les plus courants. Les mécanismes d'écriture et de composition vous sont familiers. Vos programmes fonctionnent correctement, mais vous êtes insatisfait quant à l'usage répétitif de certaines parties, la lecture et la bonne compréhension de l'ensemble, ainsi que leur encombrement, bref, vous êtes tenté de réduire, de contracter le tout. Vous commencez à réfléchir à une méthode de simplification et de meilleure lisibilité.

Vous êtes mûr pour utiliser la *Palette personnalisée*.

C'est dans cette palette que vous installerez tous vos travaux personnels, à commencer par les «Mon Bloc» (cela, vous le savez déjà), mais aussi les dossiers de rangement qui vous permettront de classer et de rechercher aisément ces «Mon Bloc».



Comme vous le constatez, cette palette est pratiquement vide - du moins au départ. Elle comprend deux sous-menus: **Mes blocs** et **téléchargements Web**.

Si vous survolez les deux icônes représentatives à l'aide de la souris, aucun déploiement se produit, sauf l'infobulle de leurs noms.

Ces deux sous-menus sont en fait deux dossiers dans lesquels vous rangerez automatiquement vos fichiers. Ils sont pour

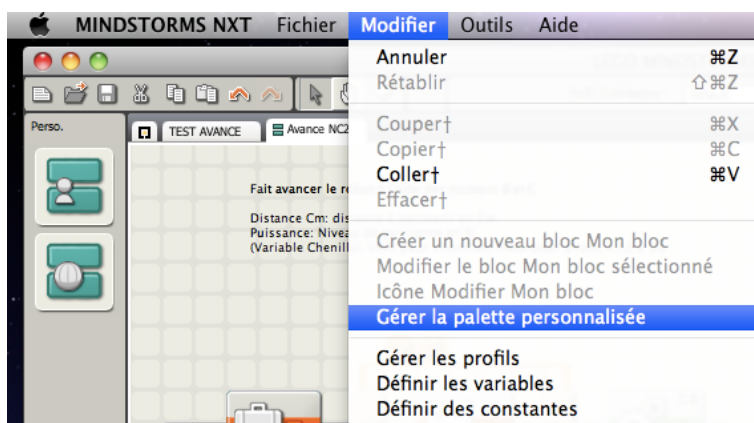
l'instant vides.

Nous avons vu au chapitre 9 - **Exemples de programmation**, § *Constante et Mon Bloc* pourquoi et comment créer un «Mon Bloc». Nous savons qu'il constitue un moyen pratique pour réduire l'encombrement des programmes et favoriser leur compréhension.

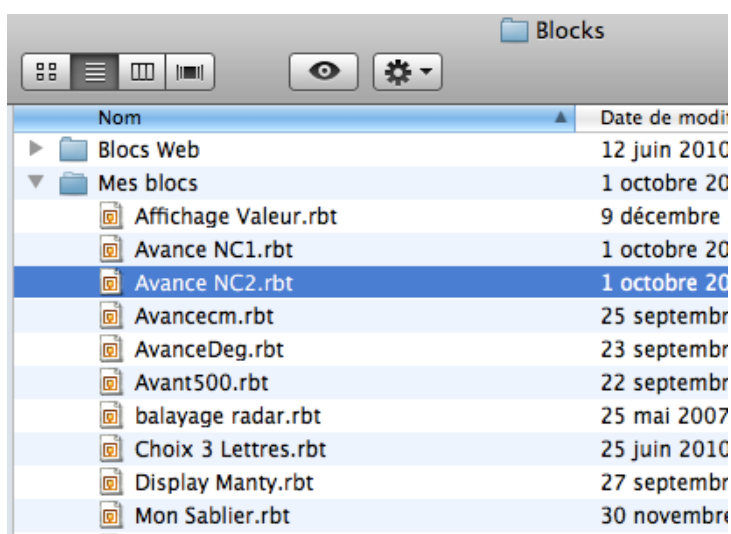
Chaque fois qu'un «Mon Bloc» est créé, il est automatiquement stocké dans le sous-menu **Mes blocs**. Il est également accessible grâce au déploiement du sous menu.

Il est évident que le sous-menu comportera au fur et à mesure un nombre croissant de «Mon Bloc» qu'il faudra peut-être organiser.

Gestion de la palette personnalisée



Comme nous l'avons vu au chapitre 9 - **Exemples de programmes** Vous pouvez copier, ajouter et supprimer des «Mon Bloc» de la *Palette personnalisée* en choisissant la commande *Gérer la palette personnalisée* du menu *Modifier*. Vous pouvez également modifier le nom de ces «Mon Bloc» pour en faire des versions supplémentaires et retrouver trace des modifications successives, etc..



v

Vous pouvez le vérifier par la commande qui ouvre un dossier **Blocks** contenant les deux sous-dossiers.

tous les «Mon Bloc» sont automatiquement regroupés dans le sous-dossier **Mes blocs**. Vous avez également remarqué la présence des deux icônes dans la bande verticale d'affichage de la fenêtre de travail. Si vous souhaitez envoyer un «Mon Bloc» à l'un de vos amis, effectuez une copie à joindre à un message e-mail. Si un ami vous envoie un «Mon

Bloc», choisissez la commande *Gérer la palette personnalisée* puis déplacer le nouveau «Mon Bloc» dans le dossier **Mes blocs**.

Les «Mon Bloc» rompus



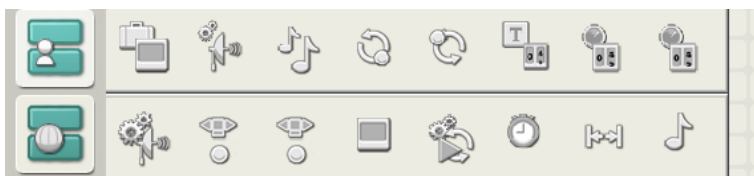
Un programme NXT-G qui utilise un «Mon Bloc» (sélectionné à partir de la Palette personnalisée), n'inclus pas en vérité le contenu de ce «Mon Bloc» à l'intérieur du programme principal. Il n'installe seulement qu'un «pointeur» avec le nom du «Mon Bloc» ainsi que quelques informations comme l'adresse de rangement. Quand on ouvre le programme, le logiciel recherche l'adresse de ce fichier «Mon Bloc» en utilisant ces informations. S'il ne trouve pas cette adresse (parce que le fichier a été supprimé ou renommé par exemple), le logiciel affichera un «Mon Bloc» *rompu* au lieu du véritable «Mon Bloc», et il ne sera pas possible de télécharger le programme vers le NXT.

La Figure ci-jointe, montre un exemple de *bloc rompu*. Créé pour un programme utilisant le «Mon Bloc» **Affichage valeur**. Le programme principal a été fermé, puis le «Mon Bloc» supprimé (à partir de la gestion de la Palette personnalisée) et le

programme principal réouvert. Le logiciel sait que le «Mon Bloc» a pour nom **Affiche valeur**, mais il est incapable de le retrouver. C'est pour cette raison que le bloc rompu est affiché avec le nom correspondant.

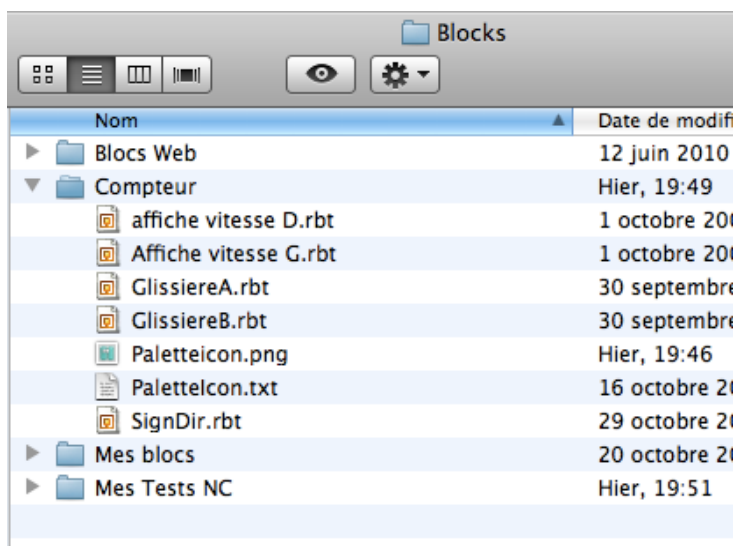
Tous les «Mon Bloc» rompus d'un programme doivent être supprimés et remplacés par une nouvelle insertion du «Mon Bloc» rectifié; ou bien réinstaller le «Mon Bloc» manquant avant l'utilisation du programme.

Création de Palette secondaires



Au fil du temps, vous disposerez d'un nombre important de «Mon Bloc». Le sous-menu **Mes blocs** se trouvera encombré par une longue liste et il deviendra difficile de retrouver le ou les «Mon Bloc»

utiles à un programme en cours. Se pose alors la question du regroupement et notamment celui de tous les «Mon Bloc» d'un projet particulier. Le logiciel NXT-G offre heureusement la possibilité de créer des sous-menus supplémentaires dans la Palette personnalisée dans lesquels vous pouvez regrouper tous vos «Mon Bloc».



Pour créer un sous-menu dans la *Palette Personnalisée*, ouvrir le dossier **Blocks** à partir de la commande *Gérer la palette personnalisée* du menu *Modifier* (voir ci-dessus).

Dans ce dossier, les deux sous-dossiers **Mes Blocs** et **Blocs Web** correspondent aux deux menus standard de la *Palette Personnalisée*.

Créez tout simplement un autre sous-dossier au même niveau que le sous-dossier **Mes blocs**. Donnez lui le nom de votre choix et déplacer n'importe quel fichier «Mon Bloc» (.rbt)

du dossier **Mes blocs** dans le nouveau dossier. Dans la figure ci-dessus deux sous-menus ont été créés: **Compteur** et **Mes Tests NC**.

Pour personnaliser l'apparence de chaque icône des sous-menus, il faut installer deux fichiers dans le dossier du sous-menu. L'un est du type .png (Portable Network Graphics) contenant l'image personnalisée de l'icône d'une dimension de 45 x 45 pixels, et l'autre du type .txt contenant le nom de l'infobulle.

De plus ces deux fichiers auront les mêmes noms dans chaque sous-menu:

Palettelcon.png et *Palettelcon.txt*. visibles dans la figure ci-dessus.

Au cas où ces deux fichiers n'étaient pas joints, l'apparence serait celle-ci:



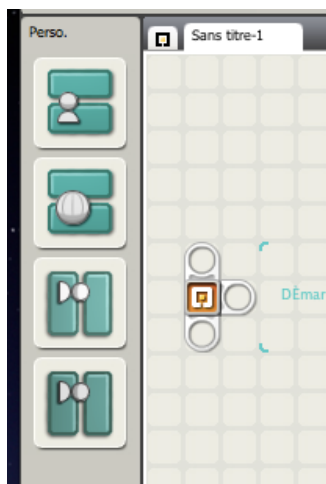
Par défaut, toutes les icônes sont identiques à celle du **Mes blocs**, y compris l'infobulle. Cela est extrêmement gênant et conduit à des confusions. Il vaut mieux personnaliser l'image et préciser les infobulles.

Pour le texte, il est très facile d'y procéder. Il suffit de l'enregistrer à l'aide d'un quelconque éditeur, texte tout simple constitué d'une seule expression, dans notre cas, *Compteur* et *Mes Tests NC*.

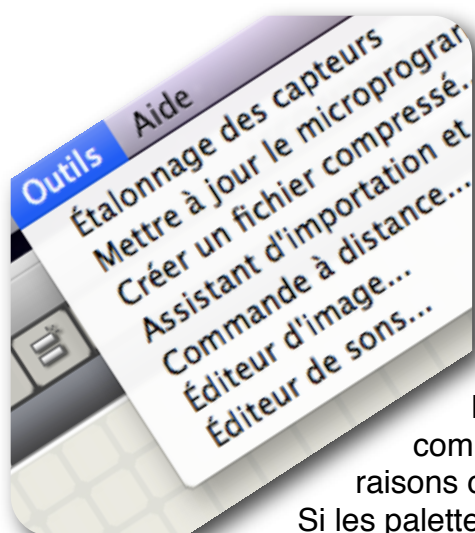
Pour l'icône, c'est un peu différent. Si vous savez utiliser un logiciel de dessin, il suffit de dimensionner une fenêtre à 45 x 45 pixels, et de la remplir par une composition de votre cru (Sous Windows avec l'application *Paint* par exemple). N'oubliez pas de sauvegarder au format .png.

Si vous ne savez pas dessiner, ou si vous ne disposez pas de logiciel de dessin, il suffit de copier une des icônes du logiciel NXT-G. Vous les trouverez dans le dossier, *Applications/LEGO Mindstorms NXT/Engine/Palette/Blocks*. Faites-en une copie (attention de ne pas la supprimer ou la déplacer) et modifiez-la à votre convenance.

Pour ma part j'ai choisi *My Blocks.png*, je lui ai fait subir une rotation pour obtenir cet aspect, mais j'aurais pu choisir une image plus significative... Et bien entendu, les infobulles limitent les erreurs.



Remarque: le fichier *Palettelcon.txt* d'une Palette secondaire est ignoré sauf si le fichier *Palettelcon.png* est également présent. Pour personnaliser un sous-menu, il faut également installer une image de l'icône.



12 – Les nouvelles fonctions

La barre de menu du logiciel NXT-G2 contient une commande qui met à disposition des Outils. Pour quelle raison ce logiciel nécessiterait-il d'outils ?

Si les palettes offrent déjà beaucoup de possibilités, les programmeurs avertis seront tentés d'expérimenter toutes les ressources du Mindstorms NXT en conjonction avec d'autres matériels de la marque Lego ou de constructeurs agréés par la marque. On peut citer par exemple le système PFS avec ses moteurs et télécommande, des capteurs compatibles fabriqués par HITECH, et même certaines applications de fans mises à disposition de la communauté.

De plus, la mise à jour du logiciel et du microprogramme s'avèrera nécessaire, ainsi que l'ajout de nouveaux blocs d'origines diverses. Bref, compléter sa bibliothèque pour en faire un large éventail de possibilités et surtout l'envie de personnaliser ses programmes incitera le projeteur averti à aller toujours plus loin.

Quatre nouvelles fonction ont été introduite; la première simplifie la circulation et la distribution des fichiers grâce à un mode de stockage compressé. Deux d'entre elles offrent un moyen d'ajouter des sons et images aux programmes en plus de tous ceux existants; enfin la dernière exploite le système de liaison Bluetooth et propose de commander les robots directement à partir de l'ordinateur, ce qui constitue un excellent moyen de tester le bon fonctionnement des robots.

Créer un fichier compressé

Cette commande accessible par la barre de menu *Outils* constitue une nouvelle façon de sauvegarder un programme NXT-G accompagné des «Mon Bloc» et fichiers sons et graphiques utilisés dans ce programme principal.

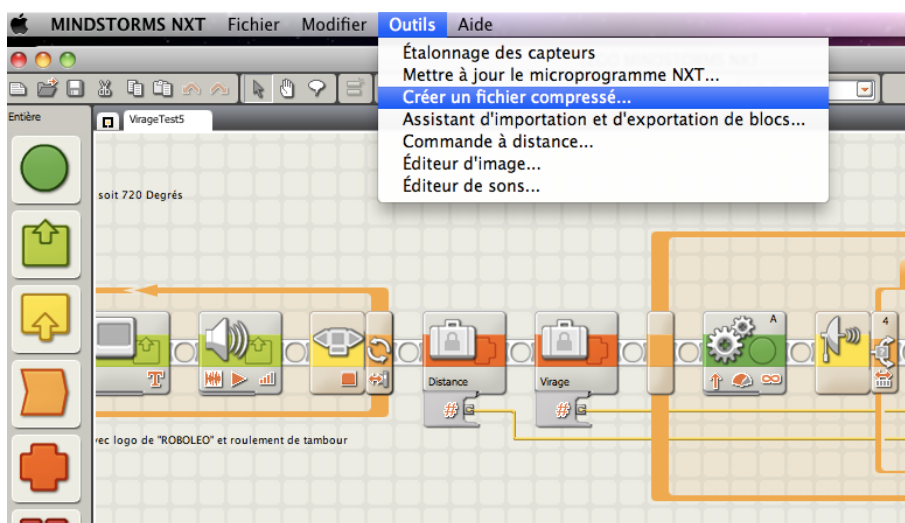
Sauvegarder un programme principal à l'aide de la commande classique «*Enregistrer*» ou «*Enregistrer sous*» ne fait qu'enregistrer ce programme sans aucune référence aux «Mon Bloc», fichiers sons et images utilisés, et le programme ne fonctionnera sur un autre ordinateur que si ces fichiers additionnels sont déjà présents. Dans le cas contraire, il faut également se donner la peine de les transférer et de les installer convenablement.

Utiliser la commande «*Créer un fichier compressé*» constitue un moyen aisé et plus fiable pour partager des programmes avec d'autres utilisateurs sur d'autres ordinateurs.

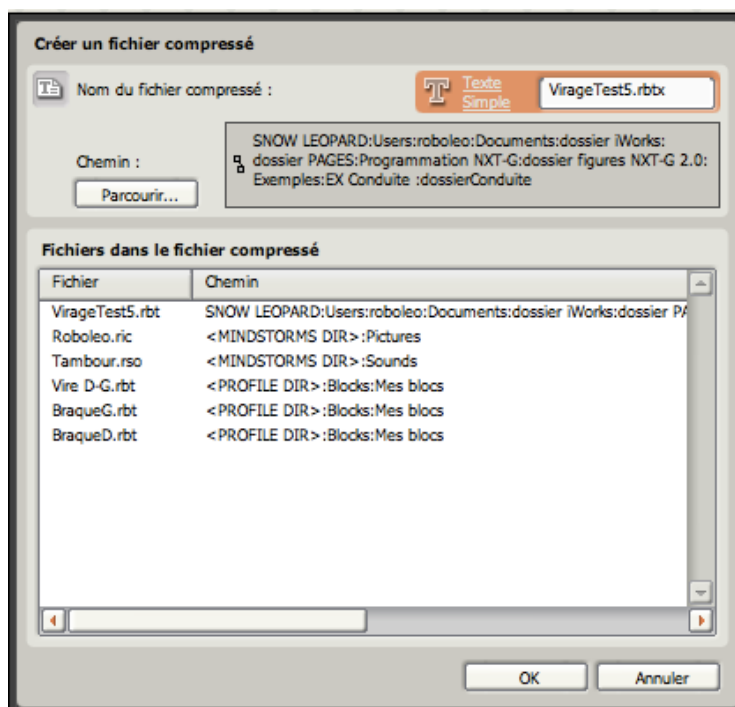
Remarque: Cette commande n'est pas disponible dans les versions antérieures. De plus, ces fichiers sont enregistrés dans un format de stockage (avec le suffixe **.rbtx**) non lisible sur l'ancienne version 1.1.

Les deux figures ci-après présentent le programme précédent **VirageTest5** auquel a été ajouté en tête un module de présentation et d'attente. Ce module affiche sur l'écran du NXT le logo ROBOLEO accompagné d'un son (roulement de tambour). Pour les besoins

de la démonstration, un fichier *image* et un fichier *son* ont été créés. Ils seront détaillés dans les paragraphes qui suivent.



Commencer par choisir dans la barre de menu > «*Outil*» > «*Créer un fichier compressé*».



Dans la fenêtre «*fichiers dans le fichier compressé*» sont regroupés tous les fichiers qui ont été utilisés par le programme **VirageTest5**.

En tête le fichier principal suivi d'un fichier *image* (terminaison **.ric**) et d'un fichier *son* (terminaison **.rso**). Ces deux fichiers ne figurent pas dans la liste standard du NXT-G. C'est tout l'intérêt de cette compression, car grâce à cette opération, il est possible de «distribuer» à d'autres ordinateurs ces fichiers personnalisés.

La liste est complétée des noms des trois «Mon Bloc» associés au programme principal. Dans la fenêtre de saisie du haut, le nom du fichier compressé est proposé par défaut, suivi de la terminaison **.rbtx**. Ce nom peut être modifié mais non la terminaison. Dans la zone suivante s'affiche le chemin de défaut où sera enregistré le fichier compressé. Ce chemin peut être autre si vous le souhaitez.

Appuyer sur bouton **OK** de la boîte de dialogue pour confirmer la création et la sauvegarde de ce «Paquet».

Le programme est sauvegardé dans un format différent que celui normalement utilisé (**.rbt**). Le logiciel NXT-G appelle ce format une *Distribution*, parce qu'il est conçu pour distribuer un programme et ses composantes à d'autres ordinateurs.

Remarques: le fichier au format **.rbtx** a été compressé, ce qui se traduit par une réduction significative de son poids en octets. C'est aussi valable pour un simple programme ne comportant aucun autre fichier.

En outre, la compression utilisée est du type zip standard. Si vous renommez un fichier compressé («Paquet» ayant la terminaison **.rbtx**) par l'extension **.zip**, alors vous pouvez l'ouvrir avec n'importe quel extracteur zip (fourni avec la plus part des systèmes d'exploitation d'ordinateurs), et avoir accès aux fichiers individuels composant le Paquet (même si vous ne disposez pas du logiciel NXT 2.0).

Editeur d'images...

L'écran du NXT peut afficher des illustrations pour appuyer une action ou une présentation. Ces illustrations peuvent se combinent facilement à du texte, et/ou s'animer. C'est par le bloc **Afficher** que le choix est possible, au travers d'une liste standard préétablie forcément limitée, insuffisante pour ceux qui recherchent une personnalisation ou des effets spéciaux.

Ajouter par soi-même d'autres images, était dans l'ancienne version difficile voire impossible. Avec la version 2.0, cela devient envisageable grâce à un éditeur d'images inclus dans le logiciel (barre de menu > «Outils»).



Il faut rappeler que cette liste utilise le format du type **.ric**, format peu courant; elle n'accepte pas d'autres types tels que .jpg, .png, .bmp. Il fallait donc trouver un moyen, soit pour créer directement de nouvelles images, soit pour en importer à partir de sources extérieures.

Création d'une image libre

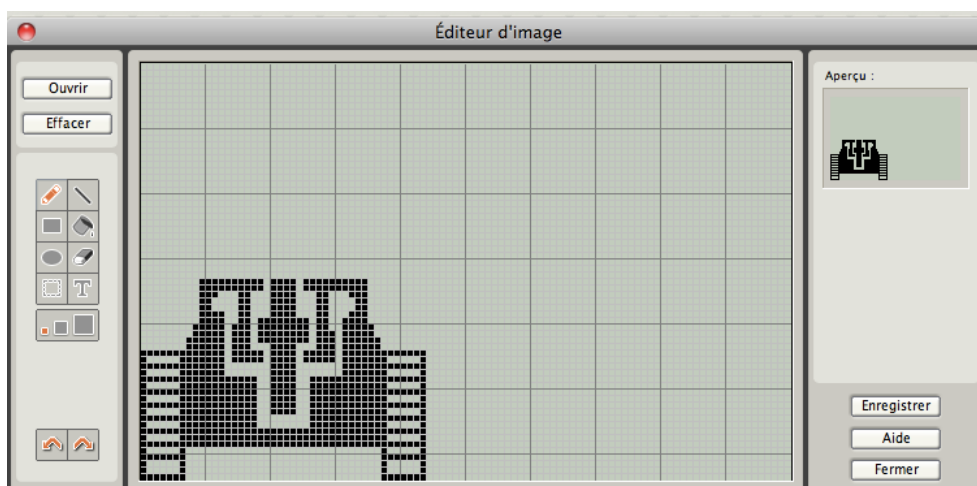
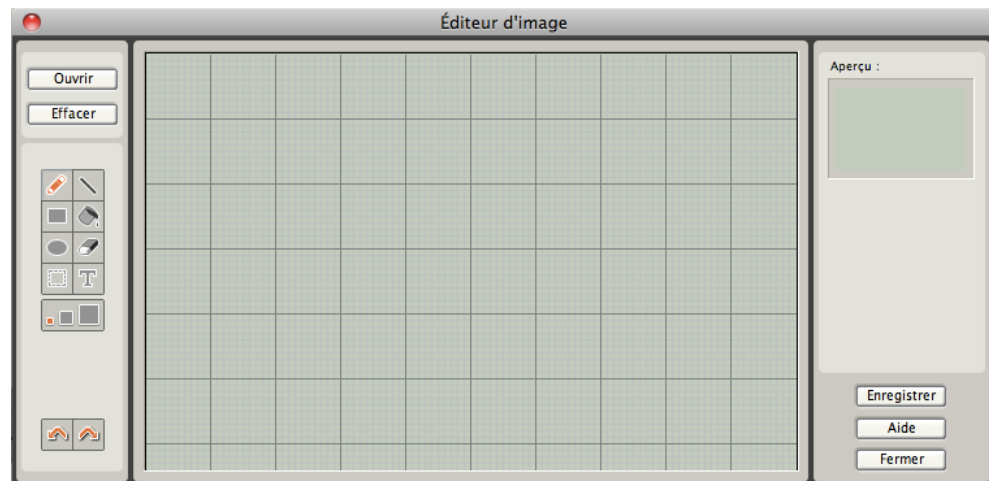
L'éditeur d'image est constitué de trois zones: la zone centrale dite fenêtre d'édition, est celle dans laquelle les images seront créées ou modifiées. Elle est complétée par une bande verticale à gauche comportant une palette d'outils de dessin, et une bande verticale à droite pour l'Aperçu et l'enregistrement.

La fenêtre d'édition comporte le même nombre de pixels que l'écran du NXT. Par conséquent, les images que vous éditez ou créez sont toujours ajustées au NXT. La fenêtre Aperçu permet de voir l'image telle qu'elle sera affichée à l'écran du NXT.

Pour mémoire, l'écran LCD du NXT a une résolution horizontale de 100 pixels et une résolution verticale de 64 pixels.

Les outils mis à disposition:

- * Dessiner à main levée
 - * Dessiner des rectangles
 - * Dessiner des ellipses
 - * Faire une sélection
 - * Dessiner des lignes droites
 - * Remplir
 - * Effacer des portions de l'image
 - * Saisir du texte
 - * Choisir l'épaisseur de trait
 - * Annuler ou rétablir les étapes d'édition
- Un exemple de dessin libre donnerait ceci:

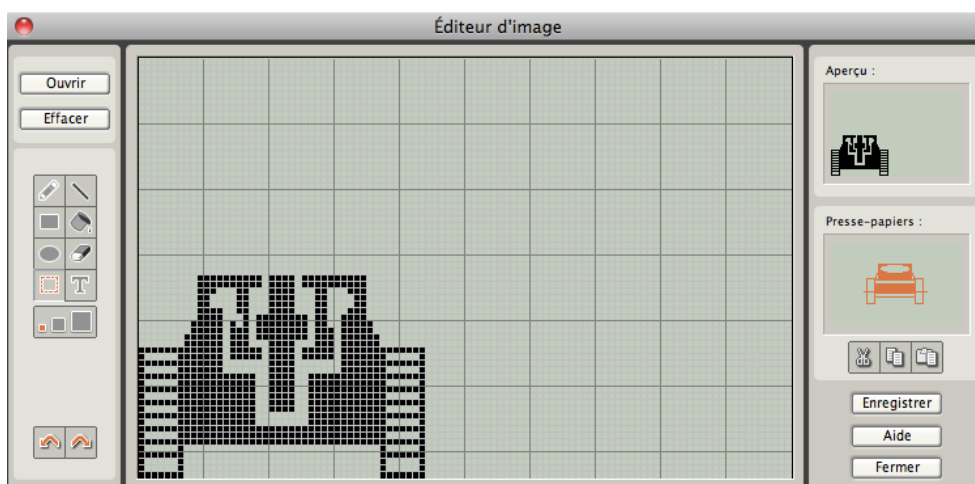


La fenêtre aperçu affiche un dessin réalisé à l'aide des outils de la palette.

Si l'outil «*sélection*» est activé, il ouvre, sous la fenêtre «*Aperçu*», une nouvelle fenêtre d'affichage de même dimension: c'est le «*presse-papiers*» avec ses trois outils (*couper*, *coller* et *copier*). Cet outil permet de tracer un rectangle de sélection dans la zone de travail.

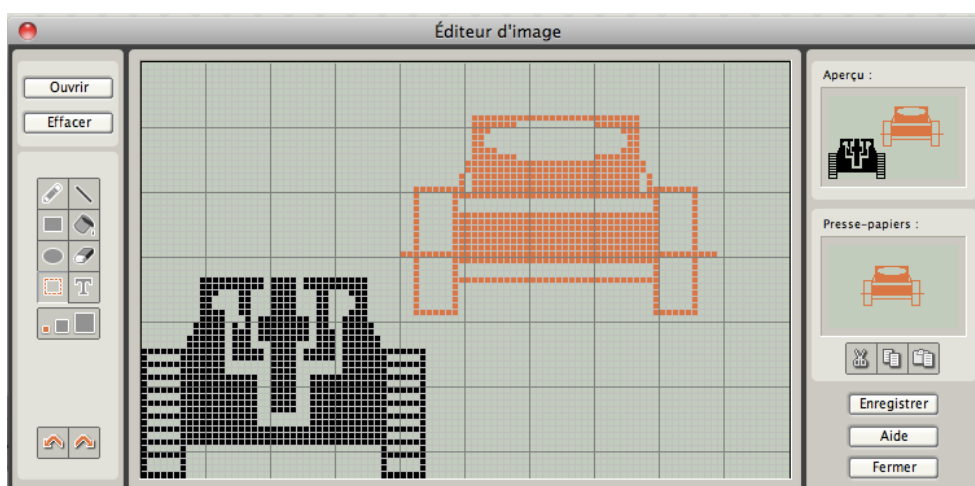
Tout ce qui se trouve inclus dans ce rectangle est sélectionné et change de couleur en orange (visible également dans la fenêtre aperçu). Si l'outil *copier* est ensuite activé, la partie sélectionnée se retrouve dans le «*presse-papiers*». Si je continue en activant l'outil *coller*, cette partie est ajoutée dans la zone de travail. Voilà un bon moyen de dupliquer une partie du dessin principal.

Tout ce qui est enregistré dans le «*presse-papiers*» demeure jusqu'à ce qu'une nouvelle **sélection** l'efface et le remplace, (dans notre cas, c'est l'avant-dernière version du dessin



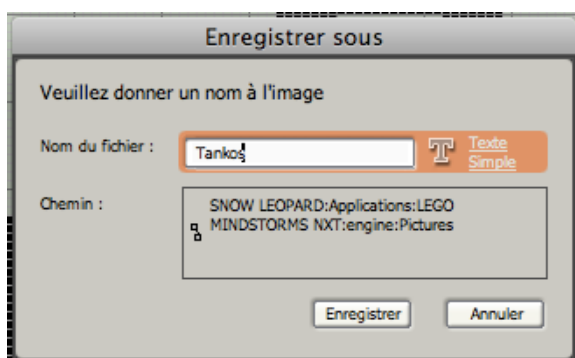
qui a été sélectionnée, différente du dessin final).

On remarque qu'elle figure en orange. La commande «Copier» a placé l'image du presse-papiers dans la fenêtre d'édition.



A l'aide de la souris cette sélection peut être déplacée à un autre endroit de la zone. Un clic en dehors de la sélection fige le dessin qui apparaît alors en noir. Il est toujours possible d'annuler cet ajout à l'aide de l'outil «*Annuler ou rétablir les étapes d'édition*». Activer cet outil ne vide pas le «*presse-papiers*».

Pour l'enregistrer, appuyez sur le bouton correspondant.



Donnez un nom au fichier du type **.ric** qui sera placé dans le dossier contenant les autres images (*fichier>illustration*). Il se trouvera dans l'ordre alphabétique.

Remarque: un dessin peut se combiner à un texte (un logo par exemple). Il est figé lorsque sa couleur rouge passe au noir. En cas d'erreur, il est possible de revenir en arrière à l'aide de l'outil «*Annuler ou rétablir les étapes d'édition*».

En résumé, utilisez la commande *Coller* pour

coller l'image du presse-papiers dans la fenêtre d'édition. Cliquez ensuite sur l'image collée et faites-la glisser sur la position souhaitée. Les images collées peuvent être déplacées lorsqu'elles sont oranges. Une fois qu'elles sont noires, elles ne peuvent plus être déplacées dans la fenêtre d'édition.

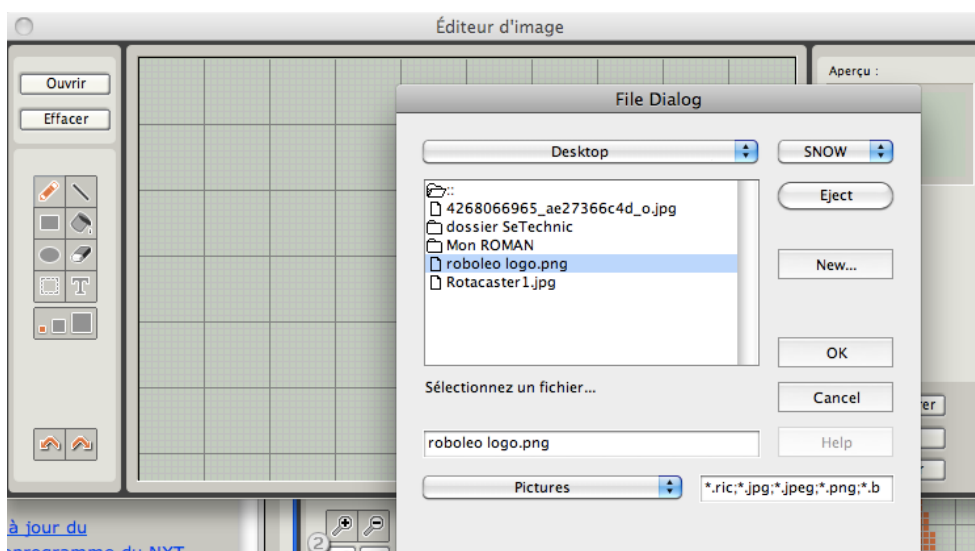
Importer une image

A titre d'exemple, nous allons importer un fichier image .png pour en faire un logo.

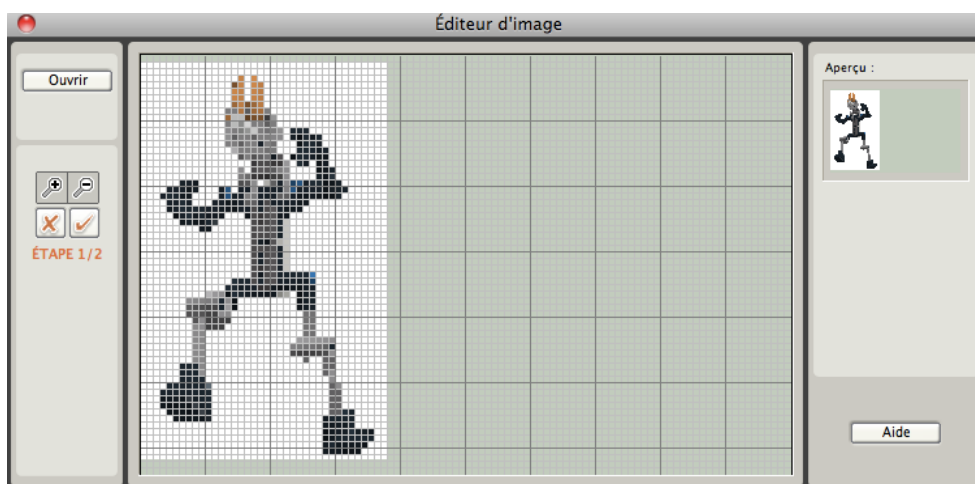


L'original:

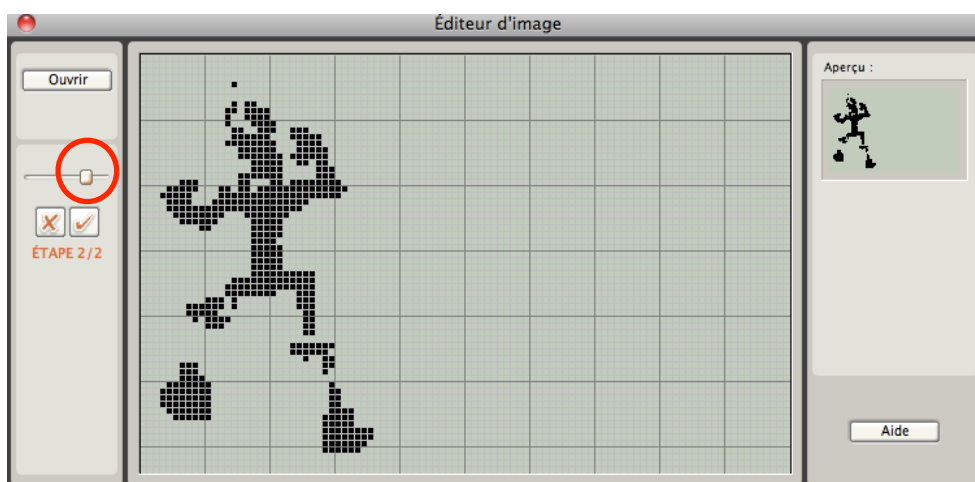
- * Choisissez l'image à éditer en parcourant votre ordinateur. Par défaut, le programme ouvre le dossier *Pictures* du logiciel NXT-G. Dans notre cas l'image est sur le bureau.



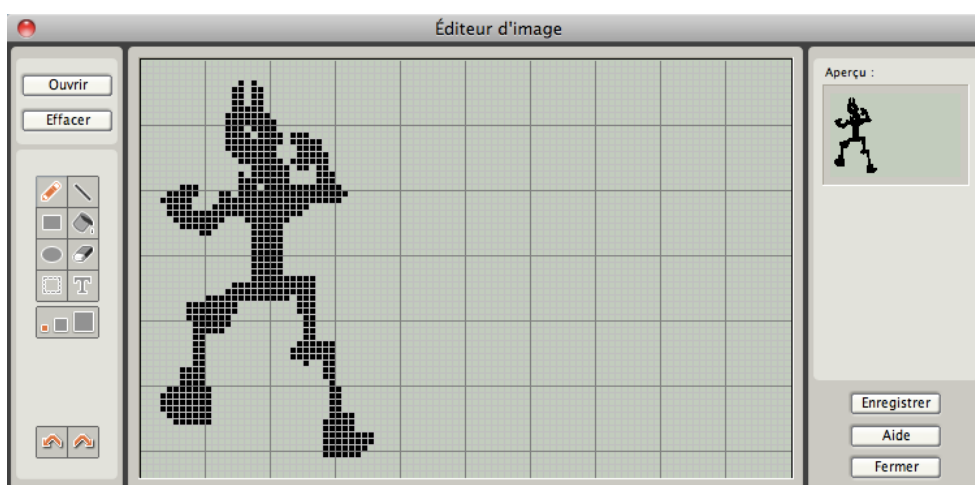
- * Recadrez la section de l'image à utiliser en (dé)zoomant et faites défiler l'image en cliquant dessus, puis en la faisant glisser.



- * Définissez le niveau de détail de l'affichage en utilisant la glissière de limite.



- * Complétez le motif, si nécessaire, en utilisant les outils de dessin. La fenêtre *Aperçu* permet de voir l'image telle qu'elle sera affichée à l'écran du NXT.



- * Enregistrer l'image comme précédemment. L'importation des images personnelles est idéale pour compléter votre panoplie. Vous avez le choix entre photos, logos, pictogrammes, motifs décoratifs, reproductions, etc..

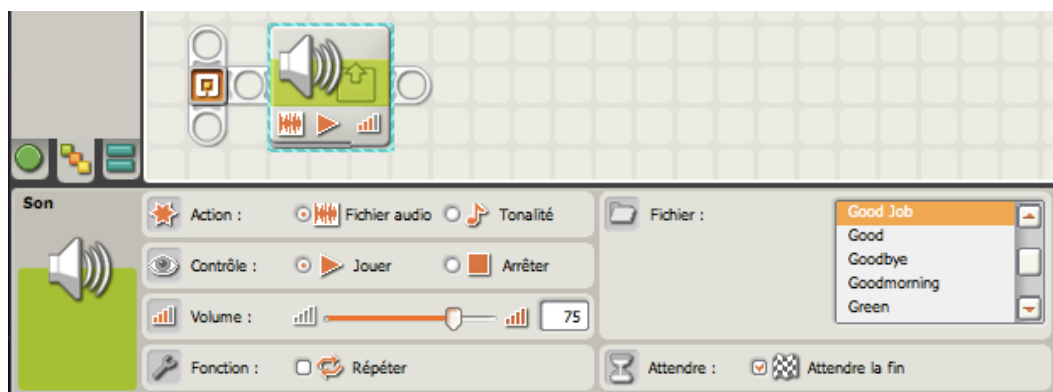
La seule contrainte réside dans le type de fichier; prenez donc le soin de vérifier qu'ils sont bien acceptés par le logiciel NXT-G.

Editeur de sons...

Le bloc **Son** permet d'entendre un fichier audio ou une note seule.
L'éditeur de sons ne s'intéresse seulement qu'au fichier audio.

Comme cela est visible dans le panneau de configuration, la fenêtre *fichier* permet de sélectionner un fichier audio dans la liste mise à disposition. Si vous estimez cette liste incomplète, l'éditeur de sons va vous permettre de modifier les fichiers audio existants ou d'en créer de nouveaux.

Il y a plusieurs raisons de vouloir créer de nouveaux fichiers audio. D'abord adapter le vocabulaire à votre propre langue. La liste standard comporte des expressions en langue anglaise. Il est agréable d'entendre vos robots parler votre propre langue, et par voie de conséquence, souhaitable d'enrichir le vocabulaire par l'ajout d'expressions faisant défaut. Ensuite, les bruitages existants sont peu nombreux, d'où le désir d'ajouter les vôtres. Vous pourriez aussi ajouter des «jingles» ou sonneries préférées, etc.. Comme vous pouvez le constater, la liste est longue...



Rappels: si vous activez l'option « *Attendre la fin* » du panneau de configuration, la lecture du fichier audio sera effectuée entièrement avant que le programme passe au bloc suivant. Si cette option est désactivée, la lecture du fichier audio se poursuivra pendant l'exécution du bloc suivant du programme. L'option « Répéter » permet de lire le fichier audio en boucle.

Vous avez donc à votre disposition plusieurs moyens d'enrichir votre bibliothèque de sons et de les exploiter selon les effets recherchés.

L'éditeur de sons est un outil qui vous permet d'enregistrer et de modifier vos fichiers **son**. Les fichiers **son** modifiés peuvent être enregistrés sur votre ordinateur, puis utilisés à l'aide du bloc **Son** comme tout autre fichier **son** figurant dans la liste. Vous pouvez également modifier des sons existants.

Vous pouvez importer des fichiers de grande taille dans l'éditeur de sons, mais seulement un tronçon ne dépassant pas cinq secondes (environ 40 ko) peut être enregistré. Le chargement de gros fichiers peut demander du temps.

Importer un fichier son

L'éditeur de son n'accepte que les fichiers avec l'extension **.rso**, **.mp3**, ou **.wav**. Une fois édités, ils sont enregistrés au format **.rso**. Il faudra donc les convertir dans l'un de ces trois

formats s'ils sont d'un autre type. Cette conversion est à faire avant d'utiliser l'éditeur de sons.

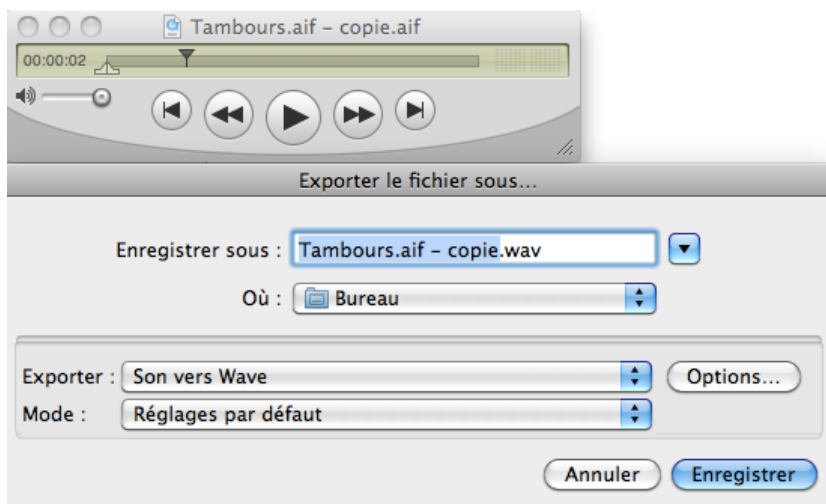
Nous allons prendre un exemple à partir d'un fichier à terminaison **.wav** aussi bien employé sur PC que sur Mac.

Le fichier choisi contient plusieurs roulements de tambour et sa terminaison est **.aif**, format qu'il faudra transformer en **.wav**.

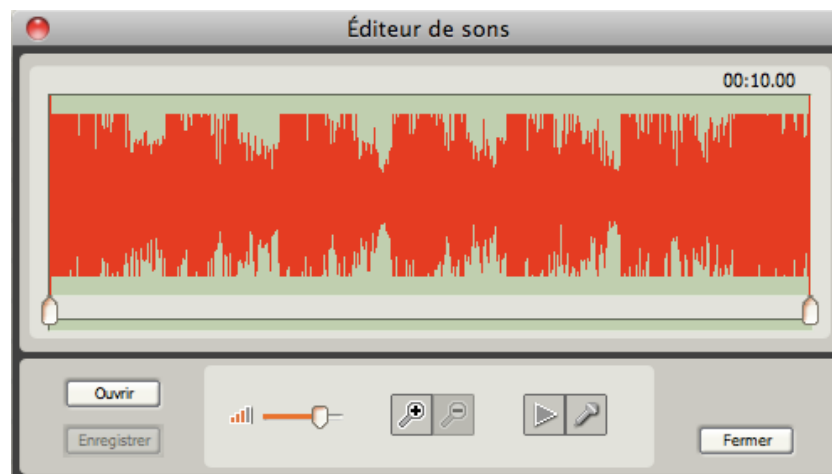
1. transformation du format:

J'ai utilisé le logiciel Quicktime Player 7 Pro pour accomplir ce travail.

Menu > Fichier > Exporter. Le fichier est sur le bureau



2. Ouvrir le fichier Tambours à l'aide de l'éditeur de son

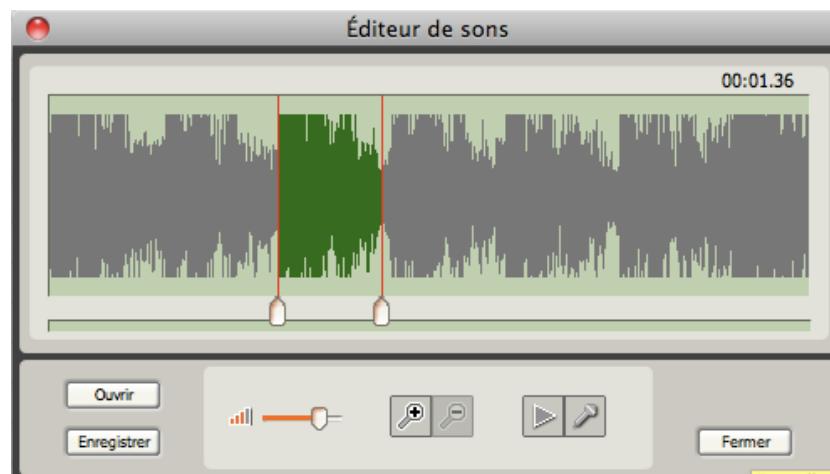


Le fichier a une longueur de 10 secondes comprenant une série de roulements. Nous allons choisir qu'un seul. Il est inutile d'en faire plus car si on souhaite entendre une série de plusieurs roulements, il suffira de mettre le bloc **Son** dans une boucle contrôlée.

3. Sélectionner le tronçon à enregistrer à l'aide des deux glissières

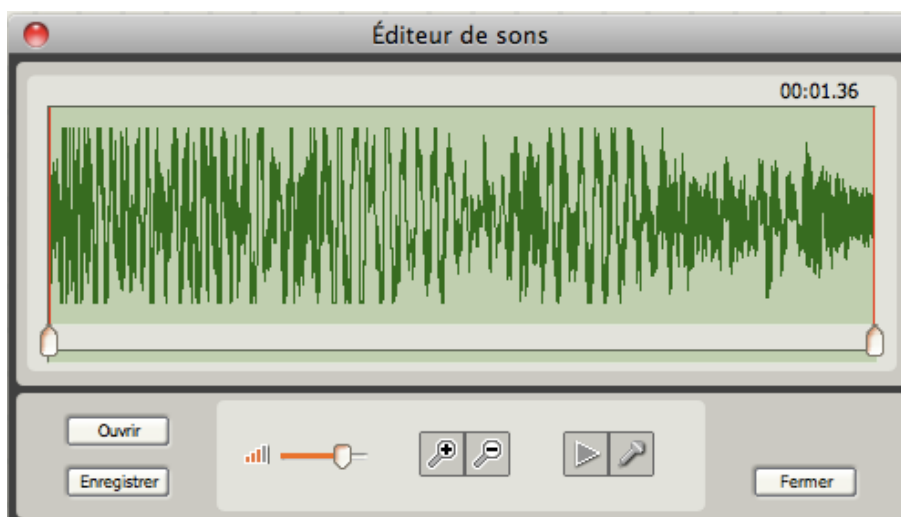
Utilisez ces glissières pour définir les limites gauche et droite du fichier son. L'onde sonore est affichée en rouge entre les glissières si le fichier son est trop volumineux. Elle devient verte lorsque la longueur de l'onde sonore entre les glissières est d'environ 5 secondes ; le bouton de sauvegarde est alors activé.

La partie sélectionnée représente ici 1.36 seconde.



4. Régler et écouter avec précision la partie sélectionnée

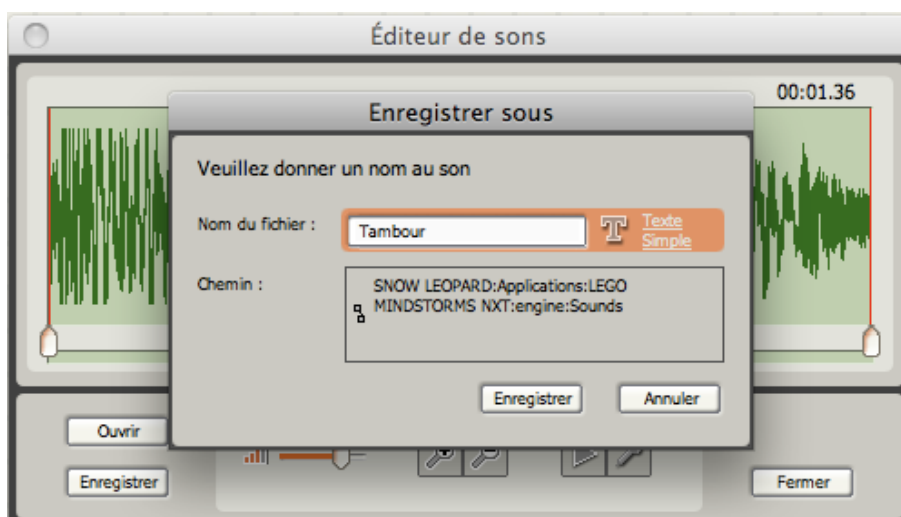
Le Zoom+ affiche la sélection sur toute la fenêtre entre les glissières pour plus de détail, ce qui permet éventuellement un réglage précis.



Le bouton > permet l'écoute de la sélection. Si elle ne convient pas, il suffit d'appuyer sur le Zoom- et on se retrouve dans la situation précédente.

5. Enregistrer la sélection

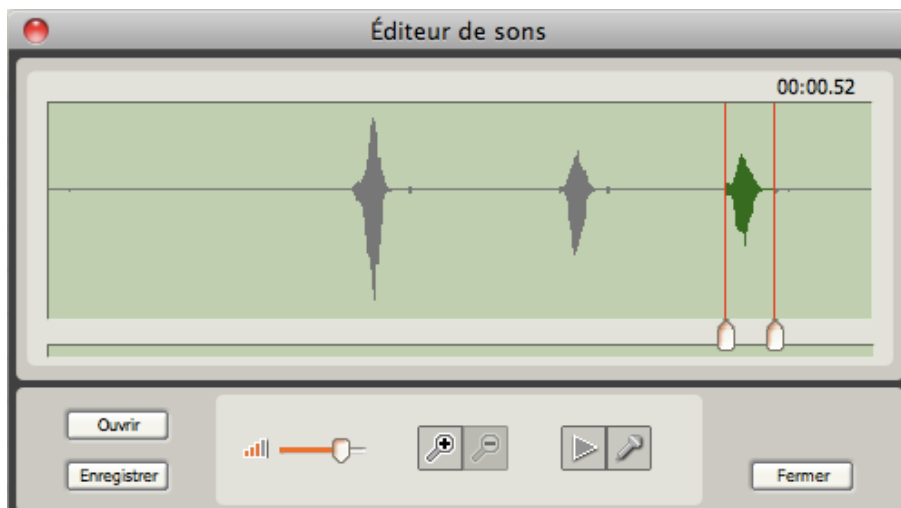
Saisir le nom du fichier qui sera placé dans le dossier *Sounds*. Le chemin est ici imposé.



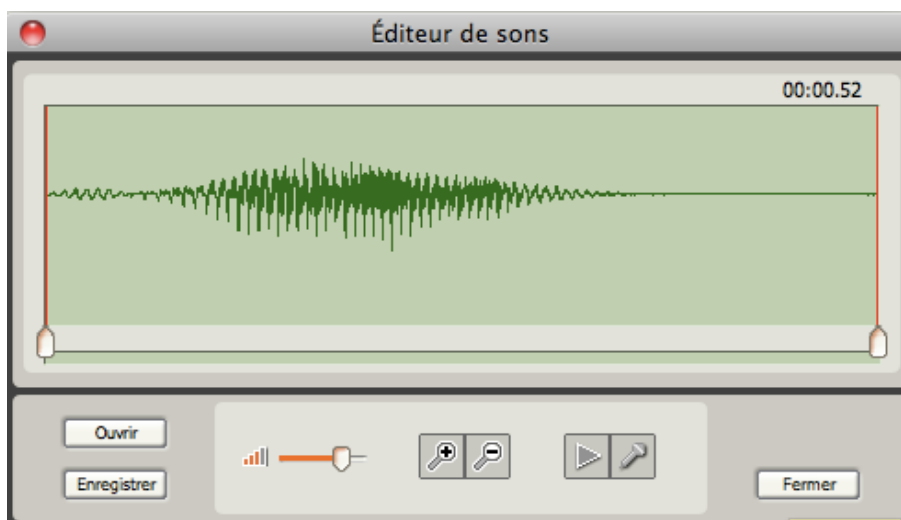
Vous pouvez alors vérifier que le fichier Tambour se trouve ajouté dans la liste.

Enregistrer un fichier son à l'aide d'un microphone

J'ai enregistré à l'aide de mon microphone intégré et à trois reprises le mot «Droite» après avoir appuyé sur le bouton micro. Puis j'ai choisi la dernière prise en utilisant les glissières, de la même façon que précédemment.

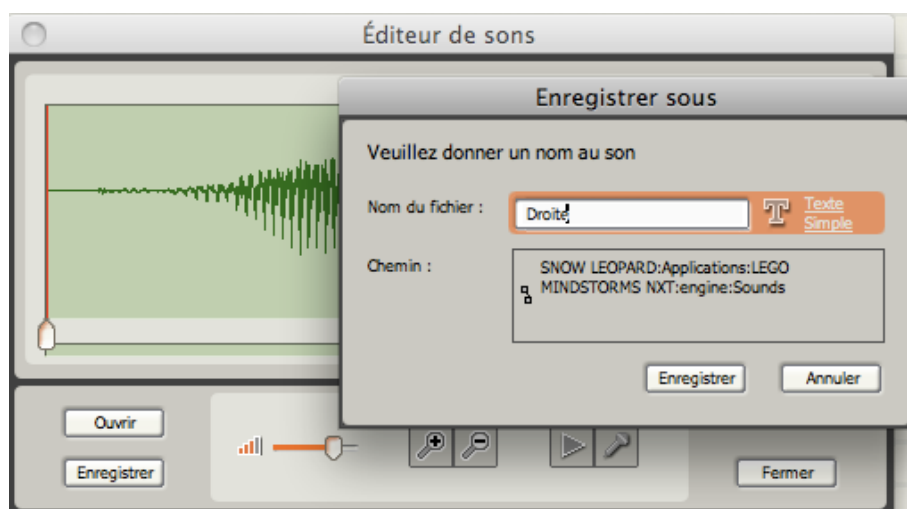


En appuyant sur le Zoom+, la partie sélectionnée s'affiche sur toute la largeur de la fenêtre. Le bouton > permet l'écoute de la sélection.



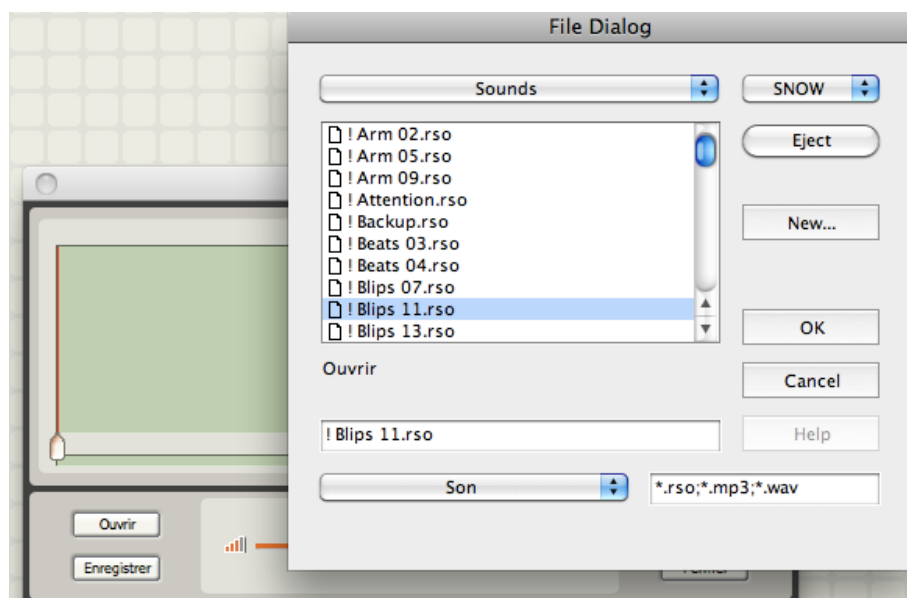
Si cette prise ne me convient pas, en appuyant sur Zoom- plusieurs fois si nécessaire, je reviens en arrière pour afficher la totalité de mon enregistrement. Je peux à nouveau sélectionner une autre prise et recommencer le travail.

Une fois satisfait du résultat, il ne me reste plus qu'à enregistrer le fichier en lui donnant un nom. Il sera placé dans le même dossier *Sounds* et ajouté à la liste.

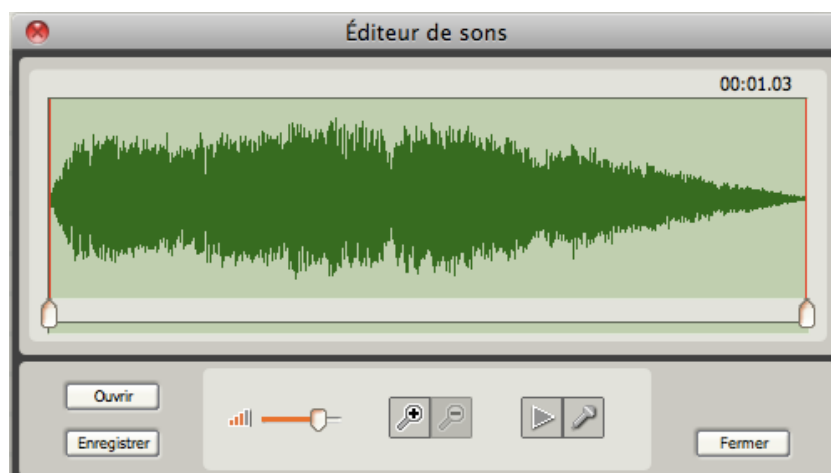


Modifier un fichier son

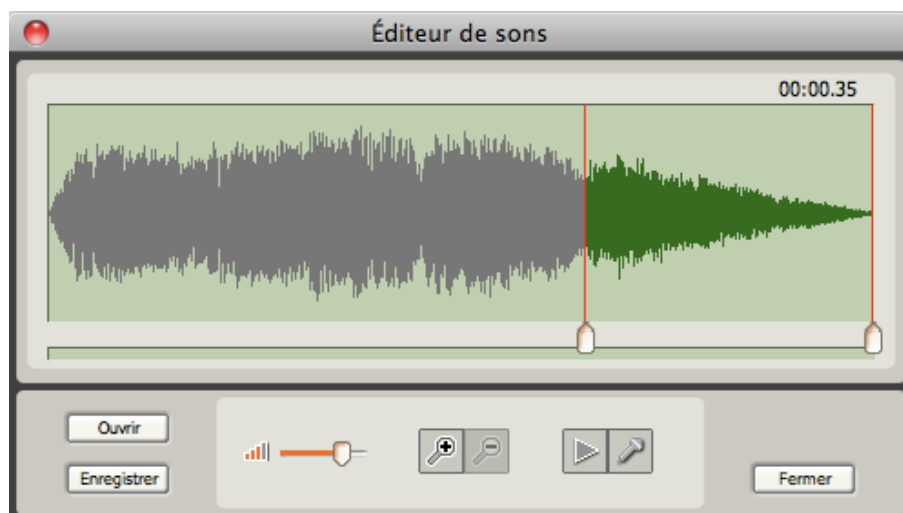
A partir d'un son existant dans le dossier *Sounds*, j'ai sélectionné **!Blips 11.rso** pour en extraire une partie et en faire un bruitage (un choc métallique).



Le bruitage de ce fichier dure 1,03 seconde. La partie de l'onde sonore qui m'intéresse se trouve en fin de forme triangulaire.



En déplaçant une glissière je ne garde que la fin et j'écoute le résultat.



Je rectifie la position de la glissière jusqu'à trouver le meilleur effet. Je suis satisfait et j'enregistre comme précédemment en donnant comme nom de fichier «*métallique*». Il trouve sa place dans le dossier *Sounds* et s'ajoute à la liste.

En conclusion et avec un peu d'imagination, il est possible de se fabriquer une infinité d'effets sonores à partir de la liste standard et en ajoutant des fichiers sons glanés sur le net. Enregistrer enfin ses mots ou expressions. personnalisera ses créations. C'est facile et rapide, alors n'hésitez pas...

Commande à distance...

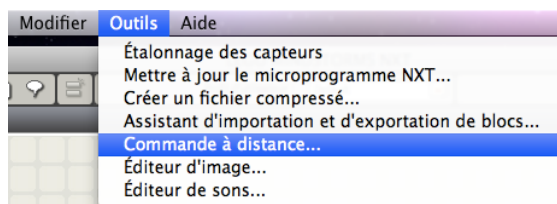
La commande à distance est idéale pour contrôler un véhicule NXT ou d'autres modèles utilisant deux moteurs pour le déplacement et un moteur pour les actions. Elle permet de conduire à vue le NXT, contrôler sa vitesse et lui faire exécuter une action directement à partir de l'ordinateur sans créer de programme; elle permet également de tester un modèle sur le plan mécanique et vérifier que les organes fonctionnent normalement, c'est très pratique, ça évite d'écrire des routines de test. Cette commande faisait défaut dans l'ancienne version NXT-G 1.

Il faut tout de même signaler que cette possibilité est restrictive et demande que les modèles soient construits en suivant quelques règles. Tous les moteurs ne sont pas entièrement et librement contrôlables.

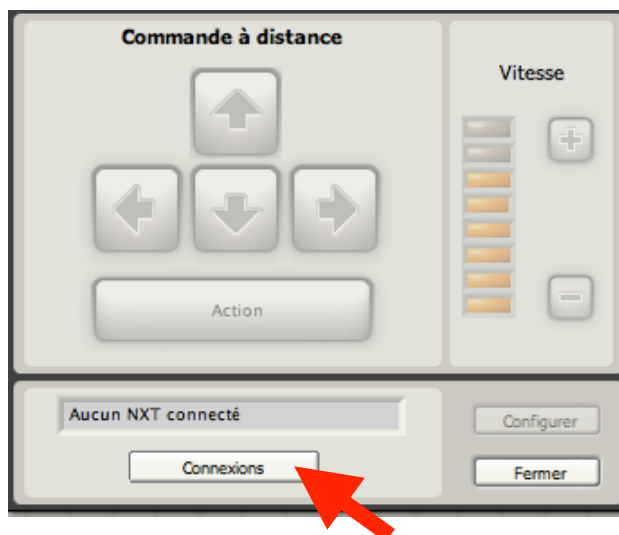
Connexion entre le NXT et l'ordinateur



Connectez directement votre ordinateur à un NXT soit par un câble USB soit par une connexion sans fil Bluetooth. La solution Bluetooth offre une plus grande liberté de mouvement, mais elle nécessite l'équipement de la technologie dans l'ordinateur ou l'acquisition d'une Clé électronique (réf. 9847 Bluetooth Dongle). Lancez le logiciel NXT-G et ouvrez un fichier sans titre. Sélectionnez «*Commande à distance...*»



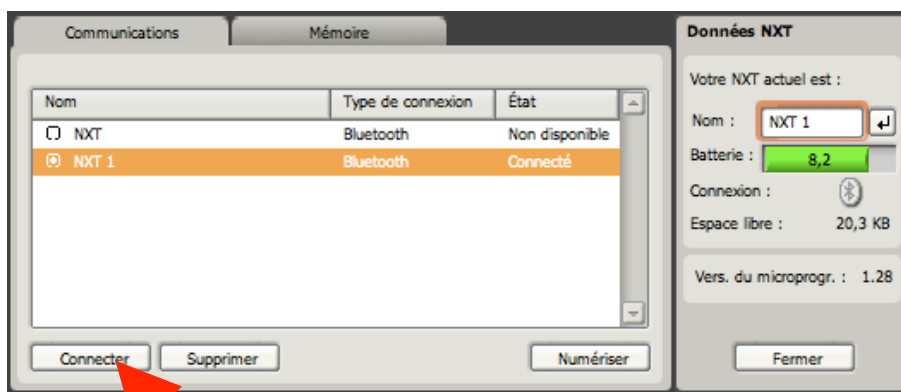
La fenêtre suivante s'ouvre:



On remarque qu'aucun NXT est connecté.

Allumez votre NXT puis appuyez sur le bouton «*Connexions*»

Vient alors en superposition la *fenêtre de Communication*. Sélectionnez le NXT à commander, puis appuyez sur le bouton «*Connecter*».

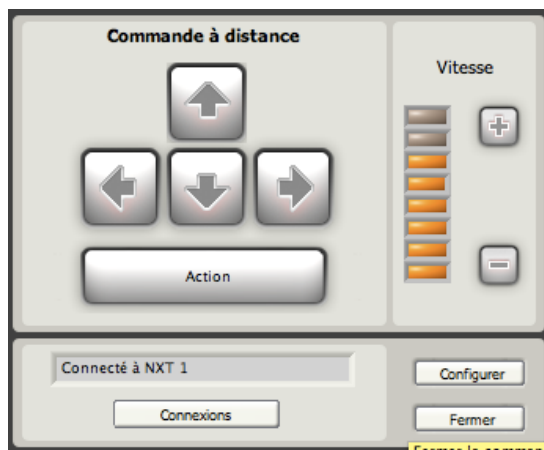


Appuyez sur le
Votre NXT est

bouton «*Fermer*» une fois les Données NXT affichées..
maintenant en communication avec l'ordinateur, et la fenêtre de

commande à distance devient opérationnelle.

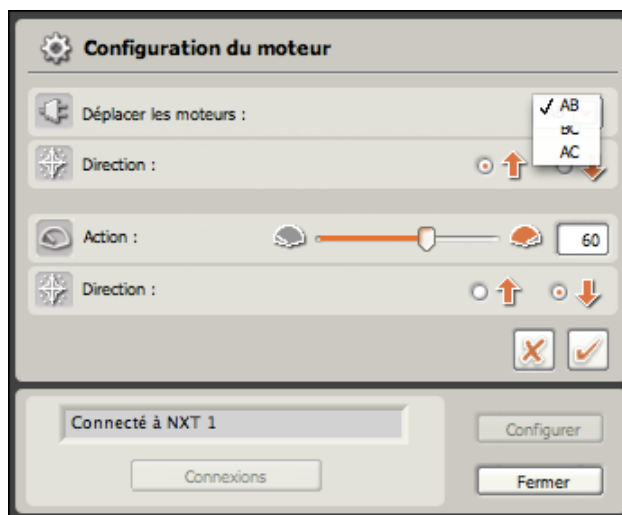
La Commande à distance



Le dispositif est organisé de la manière suivante:

A cette fenêtre sont associés deux moteurs pour avancer et diriger le modèle. Ils sont contrôlés par les boutons *flèches*. Par défaut ces moteurs sont raccordés aux ports B et C. Un moteur (le 3ème libre) est réservé pour une action, contrôlé par le bouton «*Action*».

1. Dirigez le modèle en utilisant les touches fléchées du clavier ou le clic souris sur les boutons de direction à l'écran. Le pilotage avec le clavier permet d'appuyer simultanément sur deux boutons, ce qui fait tourner le NXT progressivement et effectuer des cercles. Par exemple, appuyer simultanément sur les flèches avant et droite fait tourner progressivement le NXT sur la droite. Le pilotage avec la souris donne pour résultat des virages à angle droit.
2. Commandez le moteur d'action en utilisant la barre d'espace du clavier ou le clic souris sur le bouton *Action*. Ce bouton ne permet pas d'inverser le sens de rotation du moteur (Il n'y a pas deux boutons *flèche action*).
3. Réglez la vitesse de déplacement du NXT en appuyant sur les touches + (plus) et – (moins) du clavier. Vous pouvez aussi cliquer sur les boutons d'augmentation et de réduction de la vitesse dans le fenêtre de la commande à distance ou simplement cliquer sur l'un des huit indicateurs de vitesse. Vous pouvez également modifier la vitesse, tout en dirigeant le NXT, en utilisant le clavier. La vitesse du 3ème moteur n'est pas réglable pendant sa rotation.



Configuration des moteurs

1. Choisissez les deux moteurs qui vont faire avancer le NXT. Les moteurs seront synchronisés et tourneront en marche avant ou en marche arrière à la même puissance. Les moteurs B et C sont sélectionnés par défaut.
2. Choisissez la direction des moteurs.
3. La glissière permet de définir le niveau de puissance [0-100%] du moteur d'action.
4. Choisissez la direction du moteur d'action.
5. Cliquez sur «*Accepter les modifications*» (x ou v) pour *appliquer les modifications effectuées* ou sur *Annuler les modifications* pour conserver les propriétés que vous utilisiez déjà.

Appuyez sur le bouton «Fermer» une fois la configuration terminée.

La fenêtre commande à distance se referme également. Elle demeurera inchangée lorsque la commande à distance est rappelée.

Utilisation en divers modes

Si votre modèle exécute un programme en liaison Bluetooth, vous pouvez «prendre la main» pour le contrôler à distance.

Dans certains cas, par exemple, s'il devient inaccessible ou incontrôlable, il suffit d'arrêter le programme à l'aide du bouton «contrôleur» sur l'écran du PC, et dans le menu «Outils» choisir la «Commande à distance». C'est une manière de récupérer un modèle qui est allé se loger dans un endroit difficile d'accès.

Raccourci clavier

Vous pouvez utiliser les raccourcis clavier suivants avec la commande à distance du NXT.

<i>Raccourcis clavier</i>	<i>Action</i>
+	Augmenter la vitesse
-	Réduire la vitesse
A	Augmenter la vitesse
Z	Réduire la vitesse
1	Régler la vitesse sur le niveau 1
2	Régler la vitesse sur le niveau 2
3	Régler la vitesse sur le niveau 3
4	Régler la vitesse sur le niveau 4
5	Régler la vitesse sur le niveau 5
6	Régler la vitesse sur le niveau 6

7	Régler la vitesse sur le niveau 7
8	Régler la vitesse sur le niveau 8
Barre d'espace	Activer le moteur d'action

Une démonstration sur la manière de conduire à distance est visible sur cette vidéo:

<http://youtu.be/CA9DnLeV7Ac>

Comme vous le constaterez, c'est propre, sans fils et facilement transportable...

Ici, le contrôle du 3ème moteur n'a pas été utilisé.

Commande à distance par appareils mobiles (màj 1) 3

Connexion avec un Smartphone

Les téléphones portables de nouvelle génération sont à présents susceptibles d'assurer dans des conditions confortables et rapides des liaisons avec la brique NXT. Ils sont capables de jouer le rôle d'une télécommande à partir de petites applications gratuites ou à faible coût téléchargeables directement sur le SmartPhone.

A partir d'un Android Phone

Les téléphones mobiles Android sont le résultat de la collaboration entre plusieurs constructeurs (Motorola, Sony Ericson, Nokia, Acer, Samsung, AG, HTC, etc.) et Google™; ils sont équipés de la version du système d'exploitation Android™, et doté des meilleures applications Google™.

Lego MinDroid est une application qui vous permet de transformer votre Android Phone en télécommande pour votre Robot **Mindstorms NXT**.

<http://mindstorms.lego.com/en-us/News/ReadMore/Default.aspx?id=227417>

Lego Mindroid permet de:

- Créer une connexion sans fil avec votre robot
- Diriger à distance votre robot
- Activer à distance d'autres moteurs d'actions

Cette application mise à jour (vers 1.1) est disponible gratuitement pour les téléphones sous plateforme Android 2.1 et plus.

Elle exige l'acquisition du set # 8547 Lego Mindstorms NXT 2.0 et permet:

- Son installation sur une carte SD (pour Android 2.2).
(Nouveau modèle ajouté: Robogator)
- Une pré-installation du programme sur le NXT via le Smartphone
- Un lancement automatiquement du programme en début de l'appel
- Une activation du moteur d'action par une longue pression sur le bouton.
- Un programme à lancer peut-être sélectionné à partir du logiciel NXT-G ou du logiciel. leJOS NXJ

A signaler que cette application ne prend pas en compte le contrôle des capteurs ni la gestion des données.

Pour télécharger:

<https://github.com/NXT/LEGO-MINDSTORMS-MINDdroid>

A partir d'un iPhone

Il s'agit d'une appiTunes intitulée **iNXT Remote**.

[Par Daniel Siemer](#)

(Ouvrir *iTunes* pour acheter et télécharger les apps).

Cette application nécessite un ordinateur avec une connexion au réseau et une liaison bluetooth pour contrôler les commandes du NXT. Il exige aussi un set # 8547 Lego Mindstorms NXT pour réaliser un robot.

Une connexion directe par Bluetooth n'est actuellement pas possible, même dans la version OS 3.0, en raison des restrictions imposées par Apple sur tout le matériel (hardware) et sur ces nouvelles applications devant être approuvées par la firme. Si Apple ouvrait l'accès nécessaire pour dialoguer directement avec des dispositifs existants, comme le NXT, cette fonction sera volontiers ajoutée.

iNXT Remote (contrôle à distance) est un contrôleur pour un robot Lego Mindstorms NXT. Il peut contrôler des moteurs, scruter et enregistrer des valeurs des capteurs, enregistrer, télécharger et supprimer des fichiers du NXT et les exécuter.

Il dispose à la fois du contrôle directs des moteurs, aussi bien celui du système du véhicule qui utilise le contact, que celui de la direction différentielle.

Il vous permet de vous connecter à distance via Internet pour contrôler votre robot, ou lire des données de ses capteurs. Il dispose dans le serveur Web d'une fonction pour ajouter et télécharger des fichiers stockés sur le téléphone via un navigateur.

iNXT Remote 2.0 est compatible avec toutes les fonctions du set original Lego Mindstorms NXT 2.0, y compris le Capteur de Couleurs. Cette dernière version a été entièrement réécrite pour faciliter son usage.

Mac OSX Serveur (Requiert OSX 10.5+, USB ou Bluetooth)

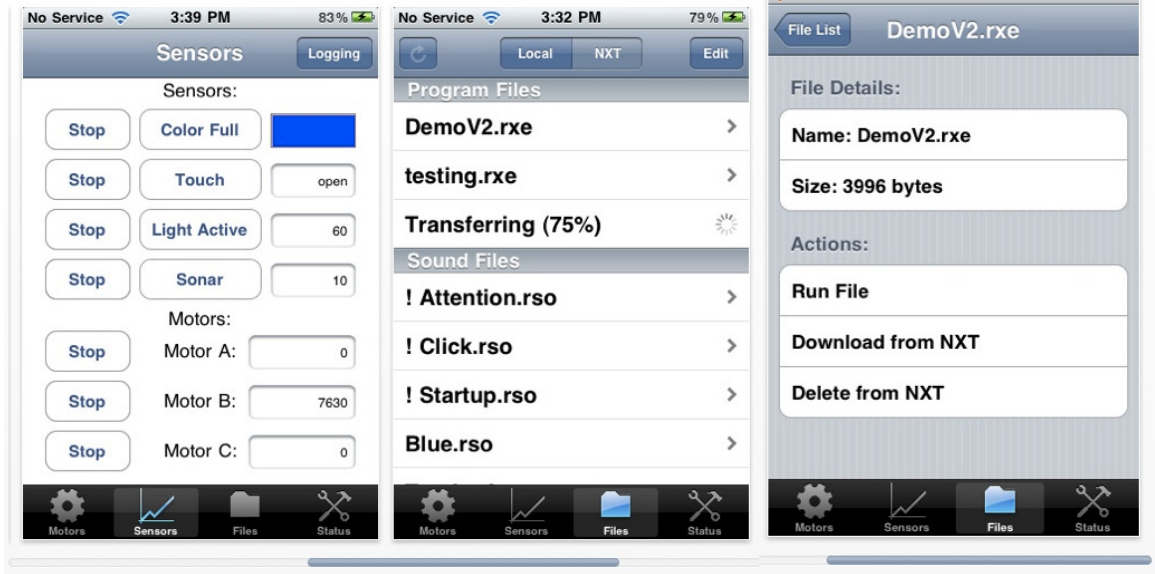
<http://code.google.com/p/nxtbluetoothbridge>

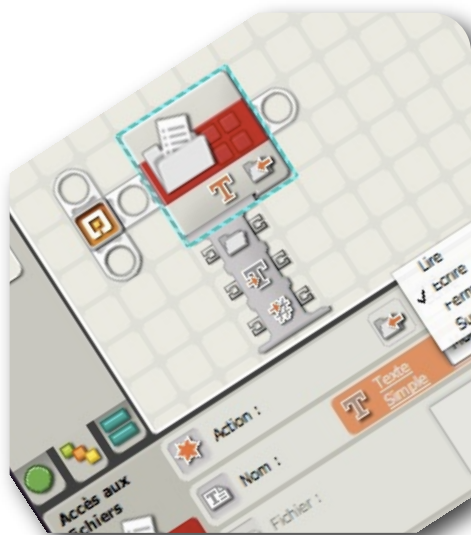
Windows serveur (3ème partie, sans garanties, devrait fonctionner sur Windows XP+ avec bluetooth)

<http://www.norgesgade14.dk/networkserver.php>



Captures d'écran iPhone





13 – L'enregistrement des données (màj 1) 4

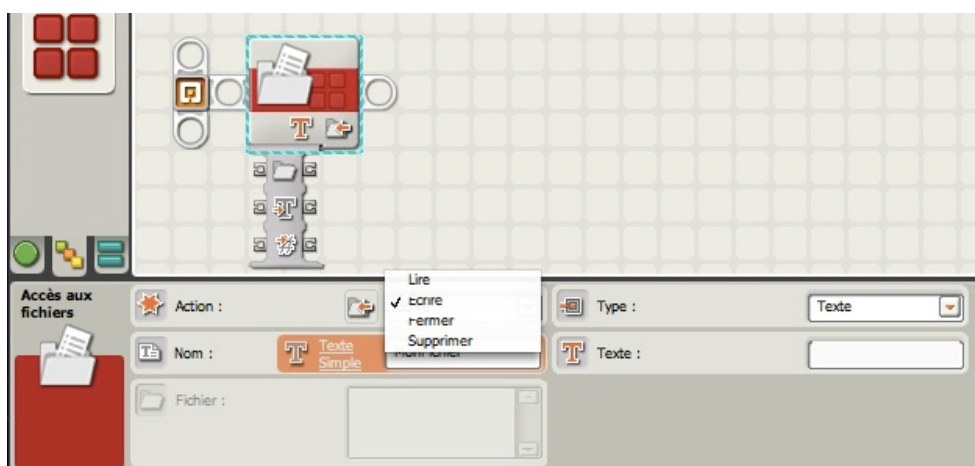
Les esprits curieux ont de multiples occasions pour explorer le monde qui s'offre à eux. Certains souhaitent approfondir leurs expériences et exploiter les informations recueillies. La grande difficulté réside dans les outils spécialisés de traitement de ces informations.

La brique NXT est un micro-contrôleur qu'il faut utiliser pleinement. C'est pourquoi dialoguer avec un NXT dans ce domaine peut à la fois être instructif et divertissant. Il est en effet facile d'emploi et suffisamment puissant grâce à ses capteurs, le tout à un coût abordable. Il est réutilisable à l'infini, c'est pour cela qu'il faut sans cesse l'exploiter. C'est aussi un bon moyen pour stimuler l'imagination et élargir ses connaissances sans pour cela fréquenter des laboratoires sophistiqués.

Rappels:

Nous allons aborder un aspect particulier de l'emploi des données au travers de la manipulation des fichiers.

Vous trouverez dans le Guide de programmation NXT-G Vers. 1.1 leçon n°13 Trucs et astuces... (page 154) une première utilisation du bloc **Accès aux fichiers** qui traite du sujet.



Il existe une autre façon de l'utiliser dans le domaine de l'expérimentation. Non seulement ce bloc est capable d'enregistrer des données, mais il est aussi capable de les restituer à un ordinateur pour les traiter d'une manière plus scientifique.

Quid un enregistreur de données?

L'idée de base d'un enregistrement de données (datalogging en anglais) consiste à réaliser une expérience, en s'aidant d'un ordinateur pour effectuer un travail fastidieux, à savoir noter à votre place une grande quantité d'informations.

Par exemple, si vous voulez connaître le changement de température extérieure pendant le cours d'une journée entière, vous pourriez vous installer à proximité d'un thermomètre, et enregistrer soigneusement toutes les cinq minutes sur une feuille de papier la température, et reporter plus tard les résultats sous forme de graphique... mais cela se traduirait par une énorme quantité de travail. Dans d'autres cas vous souhaiteriez étudier des conditions quelque peu dangereuses ou inaccessibles (la lumière s'éteint-elle vraiment dans le réfrigérateur quand vous fermez la porte ?), ou comprendre ce qui arrive dans un processus trop rapide manuellement pour l'enregistrer (à quelle vitesse une ampoule incandescente s'allume-t-elle ?).

Dans tous ces cas vous pouvez utiliser un micro-contrôleur comme la brique NXT pour automatiser le processus.

En fait, on trouve dans le commerce un grand nombre de capteurs simples et d'enregistreurs de données, pour des besoins environnementaux naturels ou industriels, mais ils ont tendance à être très spécifiques (l'enregistrement à partir d'un capteur spécifique ou deux seulement) ou plutôt chers (plusieurs centaines à plusieurs milliers d'Euros). Avec la brique NXT, c'est flexible, relativement peu coûteux et très facile.

Vous trouverez beaucoup d'ouvrages spécialisés et des sites Web pour vous renseigner sur la fabrication de vos propres capteurs personnalisés assez faciles à interfacer.

Quelques exemples du monde réel, comme la surveillance de la température et l'humidité d'une vitrine de musée, ou le niveau d'étiage d'une rivière pendant un orage.

L'enregistrement de données a aussi été utilisé pour surveiller les accélérations expérimentales des montagnes russes (un équipement de « simulacre d'essai » est envoyé sur le parcours avant qu'un humain ne l'emprunte). On pourrait citer aussi Les accélérations et la consommation de carburant d'un camion pendant un long périple. L'une des utilisations les plus connues est probablement la « boîte noire ». C'est essentiellement un enregistreur de tous les systèmes critiques de l'avion : les positions des surfaces à contrôler, la vitesse relative, les accélérations et souvent beaucoup d'autres éléments. Tout cela peut être utilisé pour reconstituer les conditions pendant un vol (une variante serait le dispositif de contrôle de la Navette spatiale Columbia, qui a fourni un nombre considérable de mesures après la catastrophe quand il a été retrouvé au sol, en grande partie intact.

Débuter avec un enregistreur de données

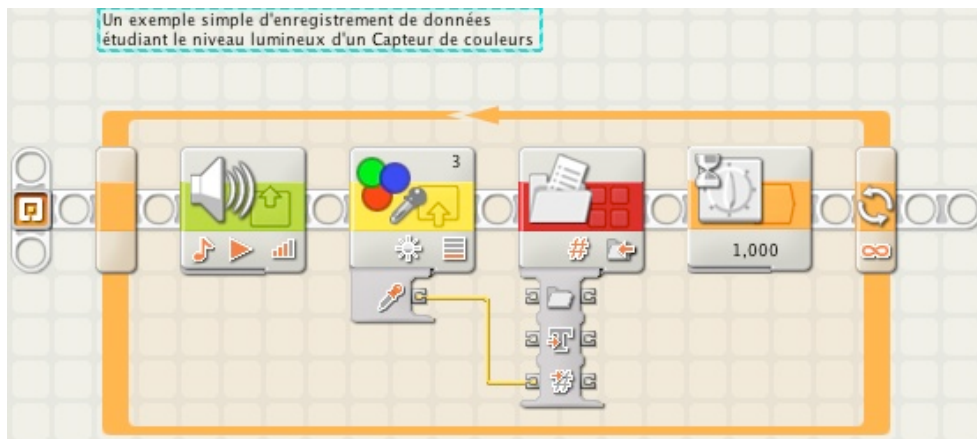
Pour commencer avec un exemple simple, nous allons étudier le niveau lumineux dans une pièce pendant une période donnée.

Par exemple, quand vous allumez une ampoule électrique, il arrive que les démarrages ne sont pas immédiats et qu'elle s'éclaire ensuite progressivement dans le temps. Vous pouvez le voir, mais pour déterminer le temps nécessaire pour atteindre la "pleine luminosité", ou pour noter l'augmentation de la luminosité pendant les premières minutes, la mesure est plus difficile à effectuer - d'où l'intérêt de la pratique d'enregistrement des données.

Si vous deviez le faire "à la main", vous pourriez vous contenter de relever et noter sur une feuille de papier les valeurs annoncées par le capteur lumineux chaque seconde (ou plus rapidement, si vous le pouvez), puis après quelques minutes, arrêter et tracer graphiquement les résultats.

La brique NXT peut faire la même chose, en inscrivant les informations dans un fichier stocké dans sa mémoire. Donc, ce que nous recherchons, c'est juste un programme qui répète en boucle (aussi longtemps que nous le désirons), la lecture de la valeur du capteur lumineux et le transfert de ce résultat dans un bloc **Accès aux Fichiers**. Voici un programme qui lit la valeur ambiante d'un capteur lumineux une fois par seconde, et qui écrit les résultats dans un fichier nommé "DEnr.txt"

Fig.1

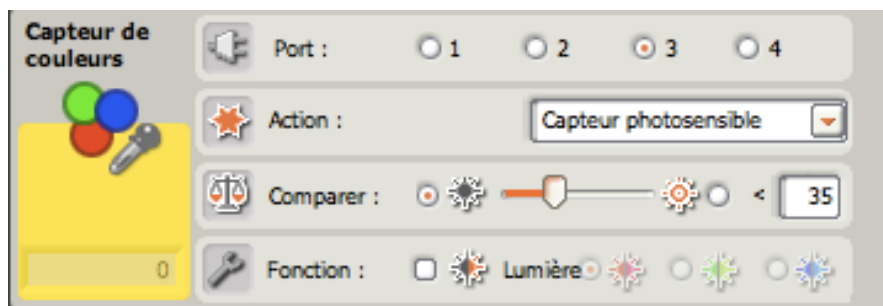


Un top sonore ponctue chaque enregistrement pour mieux suivre le déroulement de l'expérience.

Les panneaux de configuration sont les suivants:

Le capteur de couleurs est en mode Capteur photosensible, la rouge est la valeur de défaut retenue, et la case d'option Lumière a été décochée.

Fig.2



Le fichier de données a pour nom "DataEnr" et l'action "Ecrire" a été sélectionnée. Il sera enregistré avec la terminaison .txt.

Fig.3

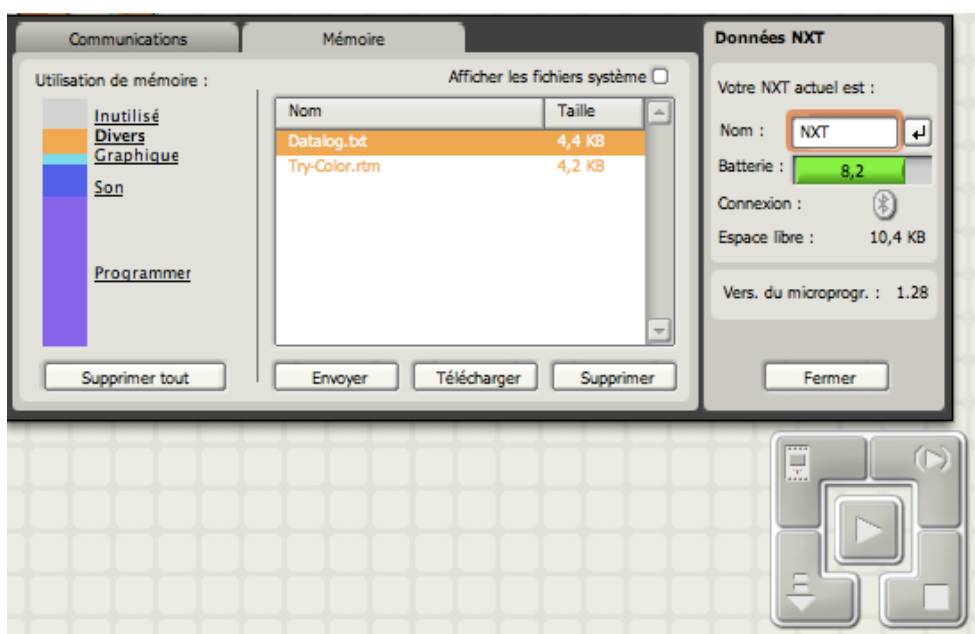


Raccordez un capteur de couleurs au port n°3 du NXT à l'aide d'un câble de connexion. Le capteur est tenu à la main.

Lancez ce programme dans une pièce sombre puis allumez après quelques secondes (le capteur est éloigné de la lampe), rapprochez rapidement le capteur lumineux vers la source lumineuse, enfin éloignez-le un peu de cette source. Maintenez sa position fixement pendant deux secondes, enfin arrêtez le programme en appuyant sur le bouton gris "Annuler" (la **Boucle** étant réglée sur "pour toujours", c'est la seule façon d'arrêter le programme).

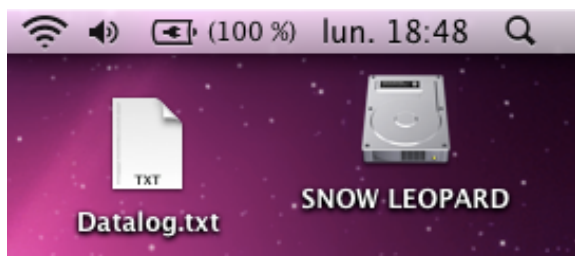
Après l'exécution du programme, connectez le NXT à l'ordinateur par le câble USB ou par Bluetooth (si ce n'est pas le cas), et ouvrez la fenêtre NXT par le contrôleur. Sélectionnez l'onglet "Mémoire" puis pour avoir accès à l' "Utilisation de la Mémoire". Dans la liste de la partie à gauche, sélectionnez "Divers". Dans la fenêtre "Afficher les fichiers système" une liste surgira comprenant le nom du fichier nouvellement créé (le nom qui nous intéresse est ici différent) suivi de la terminaison **.txt**.

Fig.4



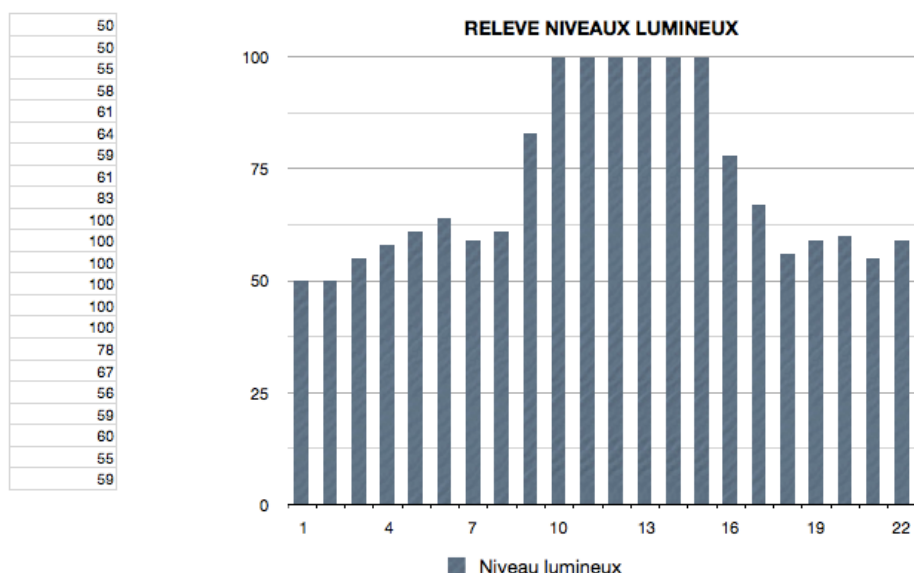
Cliquez sur le bouton "Envoyer". Cette action transfère une copie de ce fichier à votre ordinateur.

Fig.5



Si vous ouvrez ce fichier dans un logiciel de *traitement de texte*, vous verrez une série de nombres (une par ligne), et si vous l'ouvrez dans un *tableur* (Excel étant le plus connu), vous pouvez convertir ces valeurs et voir exactement ce que le capteur a enregistré pour chaque mesure sous forme graphique.

Fig.6



Améliorer un simple enregistreur de données

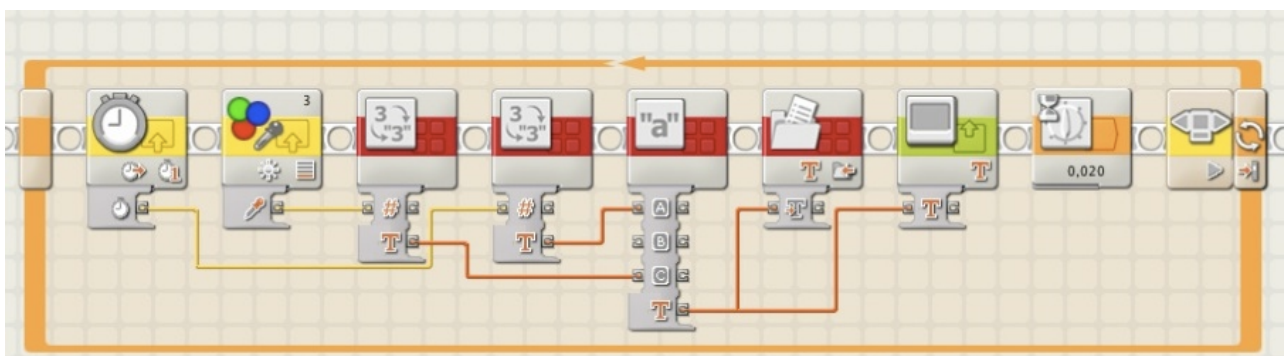
C'est l'idée, mais on pourrait faire mieux.

D'abord il serait bon que le NXT nous précise le moment de chaque relevé, de telle sorte que par la suite, nous n'ayons pas à calculer, ni nous rappeler exactement les intervalles choisis. Nous pourrions le faire en lisant un chronomètre et en associant cette lecture avec celle du capteur lumineux.

Ces deux lectures seraient regroupées sur une ligne pour être interprétées séparément. Mais pour cela, nous avons besoin d'un séparateur pour éviter la confusion.

Il faudra ensuite convertir les deux valeurs numériques en mode texte et les séparer par une virgule (le séparateur) entre elles avant l'écriture dans le fichier.

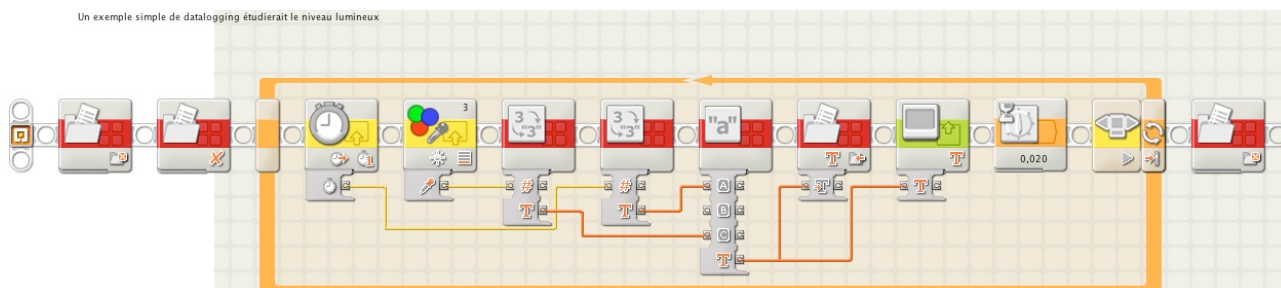
Fig.7



C'est le même programme que la fig.1 auquel on a ajouté un bloc **Minuteur** (qui mesure le temps écoulé en millièmes de seconde), et les sorties des deux blocs convertis en texte dans chacun des blocs **Nombre en texte**. Ces deux textes sont ensuite réunis dans le bloc **Texte** en une chaîne de caractères comprenant un séparateur *virgule* (,) en position **B**, séparateur saisi dans le panneau de configuration. Le résultat est ensuite relié par un fil de données vers le bloc **Accès aux fichiers** en mode *texte* (ne pas oublier de faire la modification) et vers le bloc **Afficher** pour visualiser les relevés.

Si nous lançons le programme de la Fig.1 une deuxième fois, nous constatons que les nouvelles données sont enregistrées à la suite des anciennes. Aussi pour cette nouvelle version Fig.7, nous allons prendre une précaution, celle de mettre deux blocs au début du programme, pour fermer et effacer le fichier que nous allons utiliser (pour démarrer avec une feuille propre), et pour garder seulement ces données, un troisième bloc pour fermer le fichier avant de quitter le programme (c'est une bonne habitude à prendre).

Fig.8



On aperçoit ici les deux blocs **Accès aux fichiers** avec "Datalog" comme nom de fichier, *fermer* et *effacer*, et en fin de programme un bloc *fermer*.

Et comme le NXT fait tout le travail, nous pourrions aussi bien relever les niveaux lumineux beaucoup plus fréquemment. C'est pourquoi le bloc **Attendre** a été modifié pour un temps de 20 millisecondes (ou 0,02 sec) entre chaque mesure (c'est considérable, mais c'est le NXT qui fait tout le travail). Enfin pour sortir du programme nous utiliserons la touche *flèche droite* du NXT.

Représentation graphique

Faisons une petite parenthèse pour expliquer le comportement des fichiers et leur transposition sous forme graphique.

Vous avez remarqué que leur terminaison est en **.txt** ce qui signifie qu'il est enregistré en mode texte. A chaque cycle de mesures, nous avons deux valeurs qu'il est impératif de séparer par un signe particulier. Si cette précaution n'est pas respectée, les valeurs se retrouvent réunies en une seule et n'a donc plus aucune signification. Un cycle représente en fait une succession de valeurs écrites sur une ligne. Les valeurs du cycle suivant sont reportées sur la ligne suivante, et ainsi de suite jusqu'à la fin du processus.

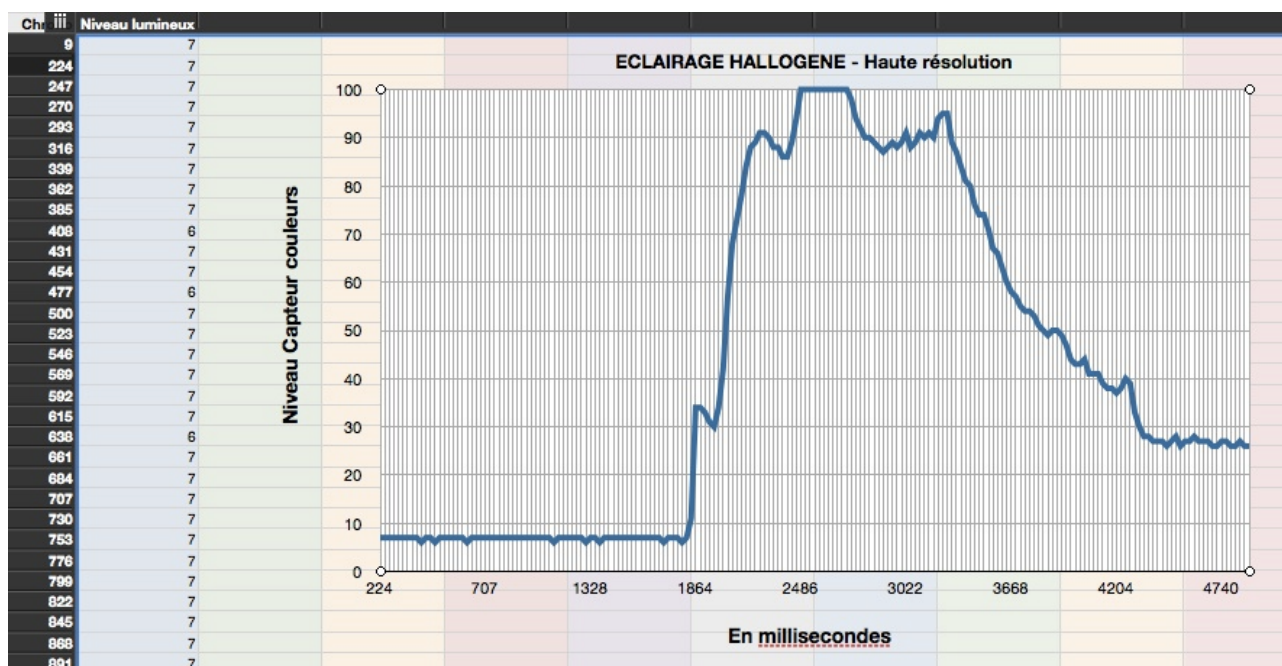
Lorsque vous ouvrez ce fichier à l'aide d'un logiciel de *traitement de texte* vous obtenez une colonne de chiffres contenant des virgules. Si vous souhaitez les faire apparaître en colonnes *distinctes* pour les exploiter séparément il faut choisir un *tableur*.

Selon le *tableur* utilisé, il faudra passer par une importation du fichier *texte* vers le *tableur*. Pour que ce dernier fasse la séparation en colonnes, prenez la précaution de modifier la terminaison du fichier **.txt** en **.csv** (Comma Separated Values) qui interprète la virgule comme un séparateur. Le *tableur* exécutera automatiquement la séparation en colonnes dès qu'il sera ouvert.

1. Lancez ce programme dans une pièce sombre (avec un éclairage halogène éteint), puis allumez au bout de 5 secondes environ, le capteur lumineux du NXT pointé vers la source lumineuse, et maintenu pendant 10 secondes. Arrêtez le programme en appuyant sur la touche *flèche droite*.
2. Connectez le NXT à votre ordinateur, choisissez le fichier "Datalog" et téléchargez-le sur votre PC.

3. Ouvrez le fichier "Datalog" à l'aide d'un tableur (Excel ou n'importe quel autre fera l'affaire. J'ai utilisé Numbers)
4. Le résultat se présente sous forme de données en deux colonnes, la première étant celle des lectures du minuteur, la deuxième correspondant aux valeurs du capteur lumineux.
5. Choisissez la fonction graphique dans la barre d'outils, le type de représentation et suivez les indications.
6. Personnalisez la première représentation brute du graphique en y ajoutant les titres, l'échelle, intervalles etc..

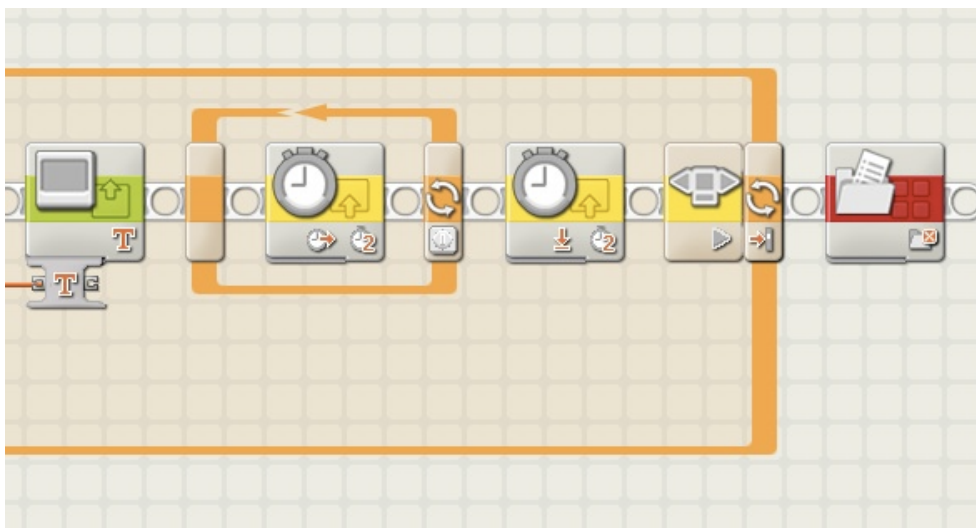
Fig.9



Très intéressant! Je peux maintenant "voir" clairement un faible éclairage, et constater à quelle vitesse se produit l'allumage. Mais il y a quelques problèmes. D'abord si je détermine la durée du cycle (temps écoulé) entre deux relevés du chronomètre, il se situe autour de 23 millisecondes, et non pas 20 comme cela a été spécifié dans le programme (et parfois plus dans de rares occasions). La raison est très simple: alors que le bloc **Attendre** est paramétré sur 20 m/s, d'autres facteurs interviennent dans la boucle (comme la lecture du capteur, la conversion en mode texte, l'écriture dans le fichier) de sorte qu'une itération de la boucle prend plus de temps qu'un simple bloc **Attendre**. Ce que nous pouvons faire de mieux pour se rapprocher de notre objectif, serait de remplacer ce bloc par une boucle intérieure qui se termine quand un deuxième chronomètre dépasse la limite fixée (en l'occurrence 20 m/s), puis juste après, ce 2ème chronomètre est remis à zéro pour accomplir une nouvelle boucle de relevés.

Fig.10

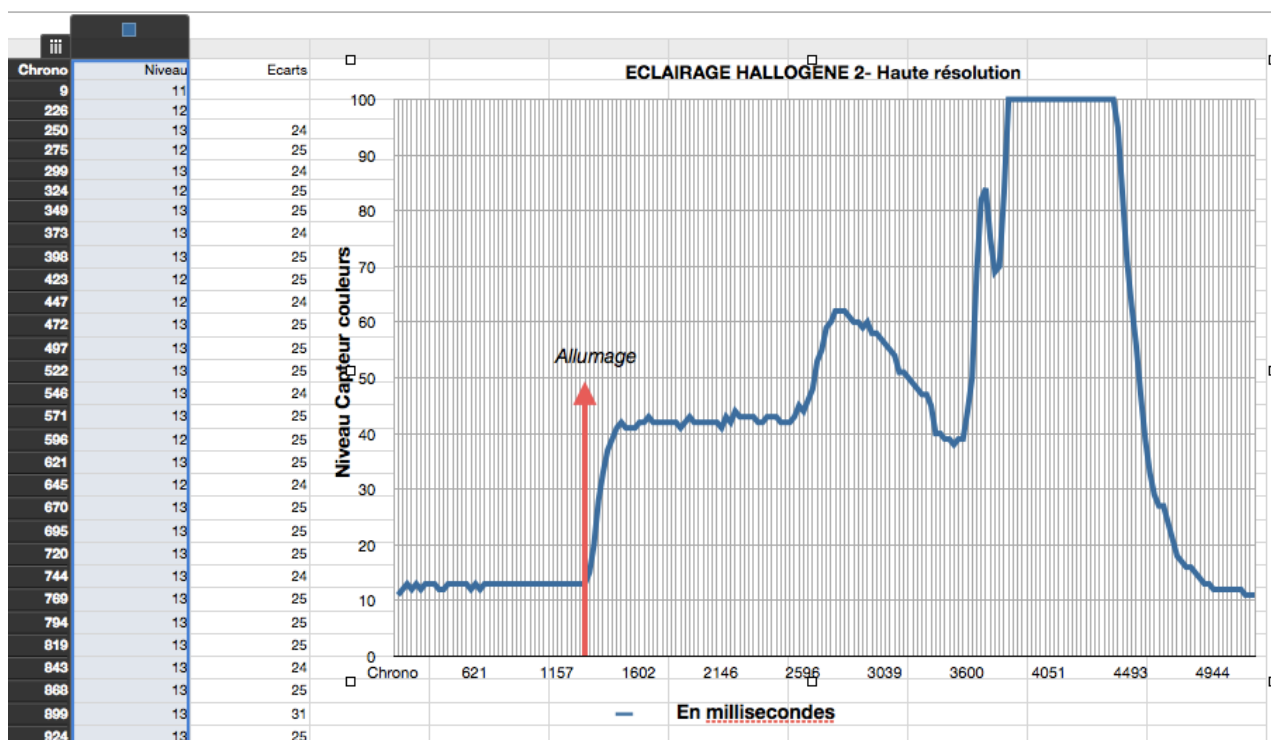
La fin du programme ressemblerait à ceci:



Une boucle "paramètre" pour durer 20 m/sec contient le bloc **Minuteur** n°2, suivie par le même bloc **Minuteur** remis à zéro.

Cela ne fonctionne pas mieux (l'intervalle normal entre deux enregistrements est d'environ 25 millisecondes).

Fig.11



En examinant les données, la plupart du temps les durées entre lectures sont plus grandes que celles attendues (en particulier si le programme enregistre un grand nombre de données). Ces sortes de timing imprévisibles imposent des contraintes aux données dans bien des cas. Ces "pauses" parasites résultent du mode de fonctionnement du NXT qui n'écrit pas immédiatement toutes les informations en mémoire. Il attend en fait d'avoir "suffisamment" de texte à écrire puis procède à l'écriture en une seule fois. Les longues pauses éventuelles du NXT résultent de l'obligation de déplacer le fichier tout entier (copie

dans une nouvelle zone de mémoire) parce qu'au fur et à mesure de son grossissement il ne dispose plus assez de place.

C'est un bon sujet de réflexion pour ceux qui recherchent la précision à de grandes vitesses. Ne vous fiez pas aveuglément aux performances du NXT et songez toujours à vérifier les résultats attendus et apporter éventuellement les corrections nécessaires.

Exemple avec un moteur

Il s'agit dans ce cas de faire tourner un moteur raccordé au NXT (par le port A) et d'enregistrer sa vitesse de rotation en fonction de l'augmentation de puissance jusqu'à son maximum. La vitesse est évaluée en tours par minutes (TPM), le moteur étant supposé tourner librement sans aucune contrainte mécanique.

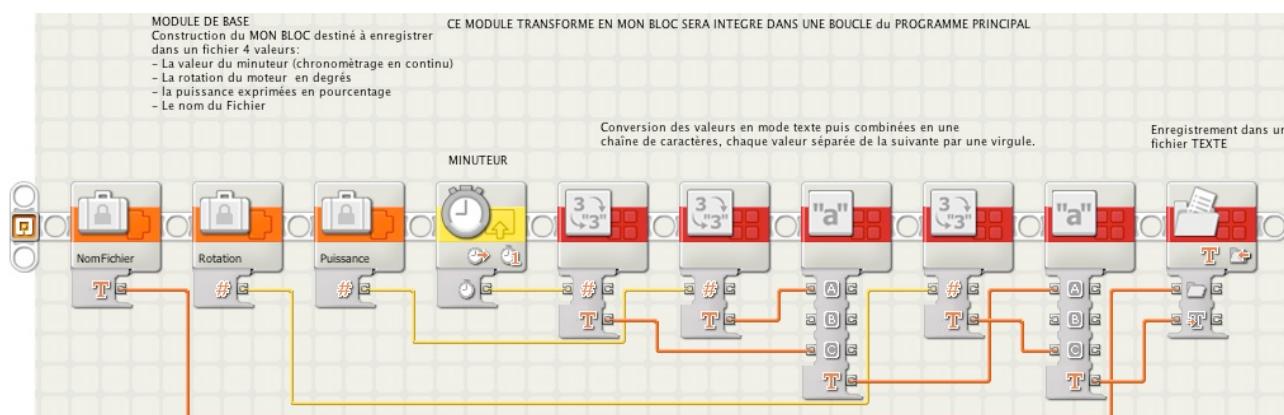
Ecriture du programme:

Module de base (pour la confection du **Mon Bloc**.

Nous avons besoin de 3 paramètres (configurés par 3 blocs **Constante**) pour composer le module de base: 2 valeurs Numériques qui seront produites par le moteur, et une valeur Texte pour le nom du fichier qui contiendra ces valeurs. Des fils de données transmettront les valeurs attendues. Un 4ème paramètre, interne au module puisque produit par le minuteur, est le chronométrage. Il n'a donc pas besoin d'un bloc **Constante**.

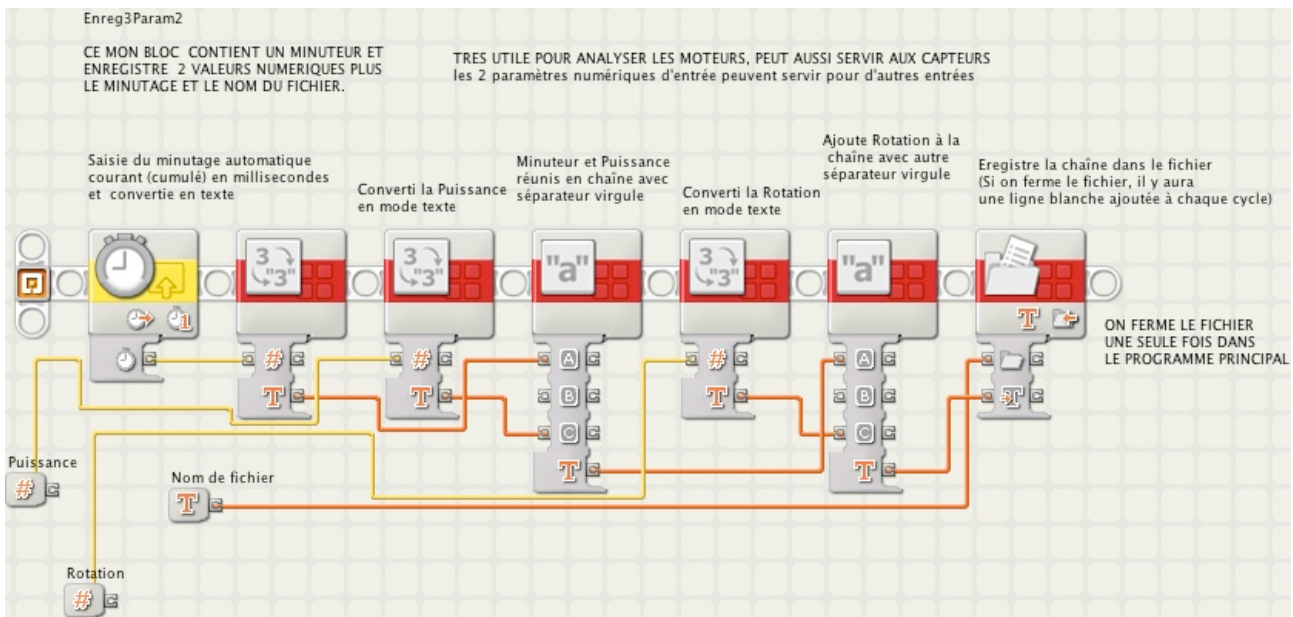
Par extension, ce module pourrait s'utiliser pour d'autres sources, comme les capteurs (son, lumineux ou ultrasons), température, etc..

Fig.12

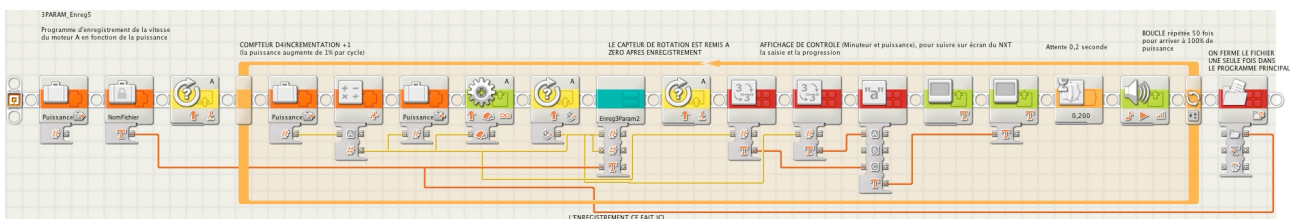


Ce module de base sert à la création d'un Mon Bloc intitulé Enreg3Param2.

Fig.13



Et le programme se présente ainsi:
Fig.14



Le nom du fichier provient d'un bloc **Constante**. La *Puissance* (en %) part de zéro et augmente de 1% à chaque cycle; la **Boucle** est répétée 100 fois. Le capteur de rotation est remis à zéro (réinitialisation) après l'enregistrement des données pour saisir uniquement le nombre de degrés à chaque augmentation de puissance. L'affichage sur l'écran de la brique NXT n'est ici que pour contrôler de visu l'exécution du programme. Pour que l'enregistrement se fasse correctement, une attente de 0,2 seconde est ajoutée en fin de cycle. La fermeture du fichier n'a lieu qu'une seule fois, à la fin du programme principal.

Exécution du programme

Lancer le programme. Le moteur se met en mouvement a une vitesse croissante; l'écran affiche les valeurs de la Puissance et la rotation en degrés. La fin du cycle est ponctué d'un top sonore. Arrivé à son terme, le fichier se ferme et le moteur s'arrête progressivement.

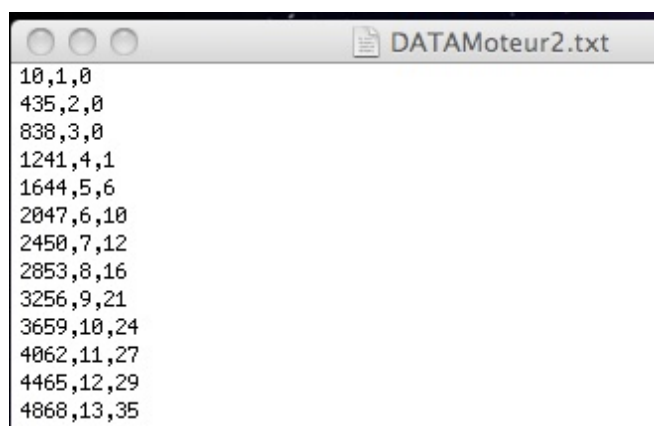
Enregistrement des données (màj 1)

Après l'exécution du programme, connectez le NXT à l'ordinateur par le câble USB ou par Bluetooth (si ce n'est pas le cas), et ouvrez la fenêtre NXT par le contrôleur. Sélectionnez l'onglet "Mémoire" puis pour avoir accès à l' "Utilisation de la Mémoire". Dans la liste de la partie à gauche, sélectionnez "Divers". Dans la fenêtre "Afficher les fichiers système" une liste surgira comprenant le nom du fichier nouvellement créé, ici **DATAMoteur2** suivi de la terminaison **.txt**.

Cliquez sur le bouton "*Envoyer*". Cette action transfère une copie de ce fichier à votre ordinateur.

Si on ouvre ce fichier à l'aide d'un logiciel de *traitement de texte*, on obtient une liste semblable à ceci:

Fig.15



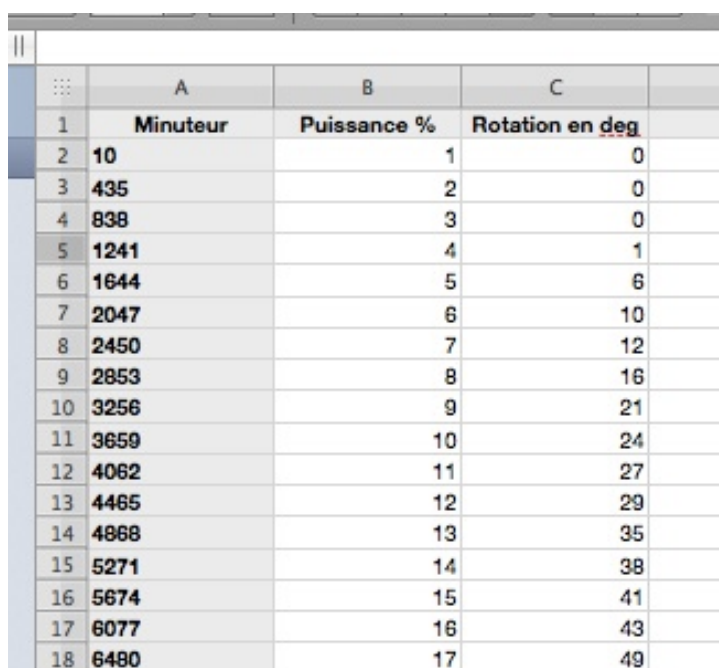
```

10,1,0
435,2,0
838,3,0
1241,4,1
1644,5,6
2047,6,10
2450,7,12
2853,8,16
3256,9,21
3659,10,24
4062,11,27
4465,12,29
4868,13,35

```

Changez la terminaison du fichier en .csv, ce qui donne **DATAMoteur2.csv**, puis ouvrez le à l'aide d'un logiciel *tableur*.

Fig.16



	A	B	C	
1	Minuteur	Puissance %	Rotation en deg	
2	10	1	0	
3	435	2	0	
4	838	3	0	
5	1241	4	1	
6	1644	5	6	
7	2047	6	10	
8	2450	7	12	
9	2853	8	16	
10	3256	9	21	
11	3659	10	24	
12	4062	11	27	
13	4465	12	29	
14	4868	13	35	
15	5271	14	38	
16	5674	15	41	
17	6077	16	43	
18	6480	17	49	

Le fichier a été ouvert sous Numbers (ou Excel si vous en disposez) et des colonnes de calculs intermédiaires ont été ajoutées pour les conversions.

Fig.17

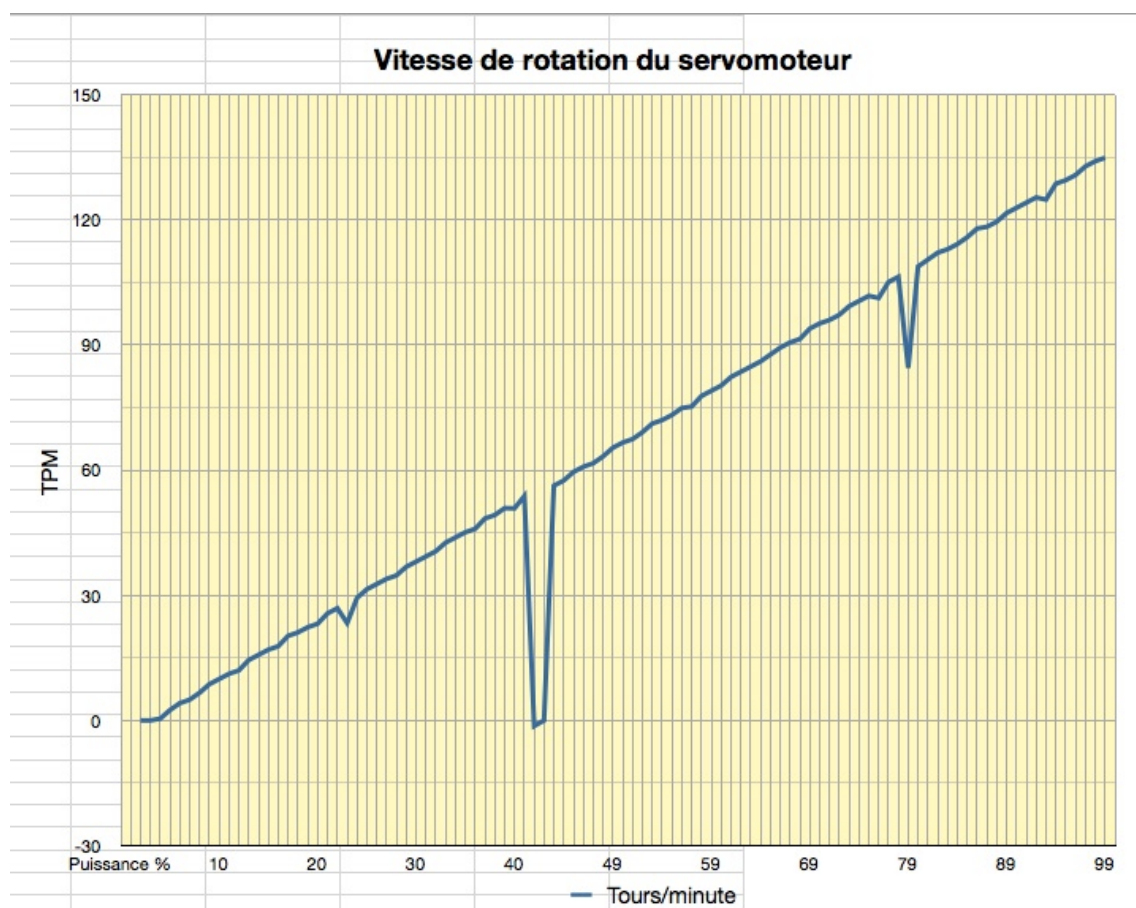
	A	B	C	D	E
1	Minuteur	Puissance %	Rotation en deg	Intervalle	Tours/minute
2	10	1	0	425	
3	435	2	0	403	0
4	838	3	0	403	0
5	1241	4	1	403	0,4
6	1644	5	6	403	2
7	2047	6	10	403	4
8	2450	7	12	403	5,0
9	2853	8	16	403	7
10	3256	9	21	403	9
11	3659	10	24	403	9,9
12	4062	11	27	403	11
13	4465	12	29	403	12
14	4868	13	35	403	14,5
15	5271	14	38	403	16
16	5674	15	41	403	17
17	6077	16	43	403	17,8
18	6480	17	49	403	20
19	6883	18	51	403	21
20	7286	19	54	403	22,3

La formule retenue pour calculer les rotation en TPM (tours par Minute) est la suivante:
 $TPM = (Rotation \text{ deg} * 1000 * 60) / (360 * Intervalle)$

J'ai préféré calculer sur le tableur ces conversions plutôt que dans le programme NXT-G, pour ne pas allonger les délais d'enregistrement de chaque cycle; Rien cependant n'empêche de calculer à l'intérieur du programme.

- Représentation graphique
- Il ne reste plus qu'à tracer la courbe à l'aide de l'outil graphique en suivant les instructions et en personnalisant le graphique.

Fig.18



On remarque trois anomalies: la première et la troisième sont liées à un intervalle de temps plus grand que l'intervalle courant, et la seconde à un cycle pour lequel il n'y a pas eu d'enregistrement. Deux explications possibles: soit un freinage intempestif du moteur, soit un délai supplémentaire requis par le NXT pour enregistrer les données.

Autres points particuliers

* Enregistrements de longue durées

Il y a encore certaines petites choses dont il faut se souvenir quand on aborde l'enregistrement des données sur le NXT. La première est la quantité de données à enregistrer et la nécessité de maintenir le NXT en veille pendant tout le déroulement de l'enregistrement.

Le NXT est généralement configuré pour s'éteindre si aucun bouton n'est sollicité au bout d'un temps donné. Si par exemple il doit s'éteindre au bout de 10 minutes, et si vous lancez un programme d'enregistrement sans surveillance qui doit durer une heure, vous trouverez le NXT éteint et vous n'aurez enregistré dans le fichier que 10 minutes de données.

Il y a deux façons de régler le problème. La première est de naviguer sur le petit écran du NXT, sur le menu système, et de régler le mode Sleep (veille) sur "never" (jamais). Cela fonctionnera, mais si jamais vous oubliez d'éteindre le NXT en fin d'enregistrement, vous risquez de vous retrouver avec des batteries à plat.

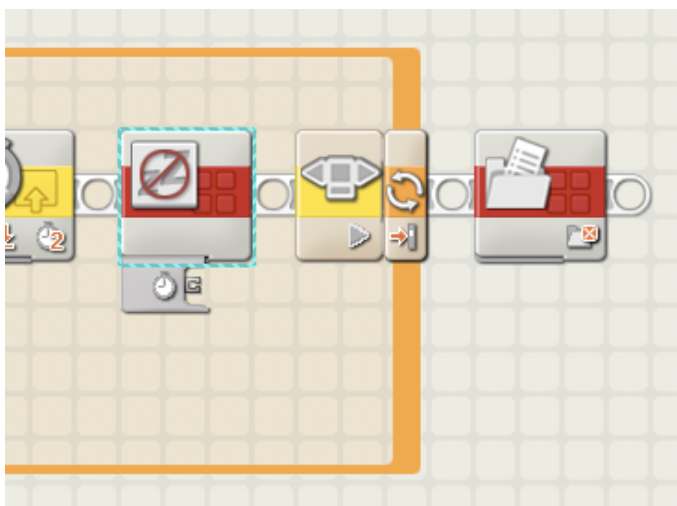
Une deuxième solution est d'utiliser le bloc **Maintenir en vie**.

Fig.19



C'est un bloc qui neutralise le délai de veille du NXT et qui fonctionne comme un presse-bouton - une façon de dire au NXT de rester éveillé (et de continuer le programme d'enregistrement des données). Tout ce qu'il faut faire est de s'assurer que le programme exécute de temps en temps un bloc **Maintenir en vie** aussi longtemps qu'il se déroule; par exemple, à la fin d'une **Boucle** principale, comme ci-après:

Fig.20



Cette solution a un autre avantage. Tant que le programme est en cours, il maintiendra le NXT "en éveil"... Mais une fois le programme terminé (après avoir parcouru le temps fixé), le NXT est libre d'agir et de se "couper" en conservant sa source d'alimentation. Cela est très pratique quand l'alimentation se fait à l'aide de piles et si vous ne voulez pas les solliciter à plusieurs reprises sur des projets à long terme.

Cela nous amène à évoquer les problèmes de la source d'énergie:

Prendre soin de l'usure des piles, c'est

bien; mais pour des projets de longues durées, les piles ordinaires sont insuffisantes (ou bien vous vous lasserez des rechargements ou achats répétés). Si vous possédez la batterie LEGO Pile au lithium polymère avec prise CC (capacité 2100 mAH-durée de charge 4 à 5 heures environ), et si vous travaillez en local fermé, vous pouvez brancher directement le transformateur requis pour le chargement (10 V CC 8887) à une prise de courant et fonctionner en prise directe. De cette façon, vous pouvez enregistrer des données pendant des jours, voire des semaines d'une seule traite, sans aucune dépense pour des piles. De plus, cette batterie est légère et ses performances sont stables à basse température.

Si vous ne disposez pas de la batterie Li-ion rechargeable, des piles neuves peuvent fonctionner pendant très longtemps, à condition de ne pas entraîner un moteur en permanence. Plus l'indication en mAH (milliAmpère-heure) est grande, plus la durée des piles sera grande; 24 heures et parfois plus.

* Multiples fichiers de données

Nous avons vu que le nom du fichier était raccordé à un fil de données, ce qui implique l'idée du changement de fichier d'un enregistrement de données à un autre, par exemple permettre à un programme de créer différents fichiers comme "DataFich1", "DataFich2", etc., au lieu d'écraser le contenu du même fichier. Ce qu'il faut, pour l'utilisateur, c'est un moyen commode pour préciser le nom du fichier (tout simplement en ajoutant un chiffre à la fin du nom grâce aux touches flèches droite et gauche du NXT). De sorte que, pendant un déplacement sans ordinateur, il puisse enregistrer de multiples fichiers et les traiter tous ensemble une fois revenu à son poste de travail.

Conclusions

Pourquoi utiliser le NXT?

Sérieusement, il existe des tas d'enregistreurs de données étonnants fabriqués par des entreprises spécialisées, capables de faire cela et même plus. Les appareils commerciaux sont capables d'enregistrer plus rapidement et plus longtemps qu'un NXT. De plus, ils peuvent stocker un plus grand nombre de points avant de saturer la mémoire. Les articles sont parfois vendus avec des logiciels de traitement, sont plus petits et coûtent moins cher qu'un NXT. Alors, pourquoi un NXT?

D'abord, le NXT procure une flexibilité qu'il est rare de trouver dans un kit bon marché. Non seulement peut-il se connecter à une aussi large variété de capteurs et enregistrer sur plusieurs fils de données, mais par dessus tout, c'est le degré de contrôle que vous avez sur lui et la façon dont il enregistre les choses.

Quel que soit le mode d'enregistrement, vous pouvez programmer - vous êtes sans aucune manière limité par les contraintes imposées par un fabricant. Pour un enregistrement "stupide" (lectures de séries aux intervalles prédéfinis), n'importe quel enregistreur immédiatement disponible peut être utilisé. Mais avec le NXT, un ordinateur sous votre contrôle, vous pouvez faire des enregistrements "intelligents". Enregistrer quand les valeurs changent rapidement, ou seulement quand certains capteurs se situent dans des plages données. Eventuellement contrôler les moteurs, les points lumineux ou d'autres sorties selon la situation présente. Aucun enregistreur du commerce n'a d'équivalence et certainement pas au prix du NXT.

Deuxièmement, du point de vue de l'enseignement, le NXT est un outil qui est "presque complètement transparent" pour l'étudiant. Parce ce dernier doit comprendre ce que font exactement les capteurs. Devrions-nous mesurer la valeur instantanée, ou la moyenne ? Ou peut-être voulons-nous enregistrer des valeurs maximales et minimales ? Une fois les données obtenues, que signifient-elles ? Comment convertissons-nous les lectures de capteurs en "des unités réelles" comme des accélérations en mètres par seconde au carré, ou les niveaux lumineux comparés à une certaines normes connues ? Quelle est la résolution ou la précision des données et en quoi sont-elles différentes de la tolérance ?

Tout ces questions sont critiques... et souvent dissimulées parce que l'étudiant n'est jamais tenu de leur faire face, ou pire n'a aucune occasion de manipuler (les appareils du commerce enregistre-t-ils les lectures moyennes, ou les lectures instantanées ? Ou quelque chose d'autre ? L'utilisateur ne le sait souvent pas lui-même). Avec le NXT vous avez un outil qui est "ouvert" et qui n'existe pas dans le domaine commercial. Le NXT n'est pas destiné à produire seulement quelques données pour l'analyse - c'est une façon de comprendre comment obtenir ces données. Et dans un cadre éducatif c'est beaucoup plus important que "saisir les chiffres".

Crédits à Brian Davis

<http://www.teamhassenplug.org/NXT/>



Mises-à-jour

Mise-à-jour n°1 - Mars 2011

4 - Installer le microprogramme sur le NXT2

Assistant d'importation et d'exportation de blocs...([màj 1](#))

10 - Un grand classique: Le «Suiveur de Ligne» ([màj 1](#))

12 - Les nouvelles fonctions

Commande à distance par appareils mobiles ([màj 1](#))

13 - L'enregistrement des données ([màj 1](#))



Remerciements

Ce guide est l'aboutissement d'un travail de recherche, de rédaction, de sélection de programmes et d'articles et communications diverses, de tests sur modèles construits pour l'occasion.

Il est loin d'être complet, mais il n'a d'autre ambition que d'éclairer les adeptes du Mindstorms NXT© et utilisateurs du logiciel NXT-G, dans la conception et l'écriture de leurs programmes. Ce logiciel est digne d'intérêt dans ce qu'il contient de ressources et d'astuces, dans sa structure et le mode de raisonnement qui en découle.

La communauté des AFOL's ne s'est pas trompée.

j'observe avec un grand intérêt l'évolution de cette discipline qui se glisse dans les foyers.

Je tiens tout particulièrement à remercier

Les rédacteurs du site: <http://us.mindstorms.lego.com/en-us/default.aspx>

Les rédacteurs de l'«Aide et support pour LEGO® MINDSTORMS® NXT»

La communauté de tous ceux qui partagent leurs projets:

<http://us.mindstorms.lego.com/en-us/Community/NXTLog/DisplayProjectList.aspx>

Tous les collaborateurs du blog The NXT STEP

http://thenxtstep.blogspot.com/2010_10_01_archive.html

Et plus spécialement,

Team Hassenplug

<http://www.teamhassenplug.org/NXT/>

Dave Parker

<http://www.nxtprograms.com/help/MyBlocks/tutorial.html>

Damien Kee

<http://www.damienkee.com/sequencer.html>).

Matthias Paul Scholz

<http://mynxt.matthiaspaulscholz.eu/index.html>

Philippe E. Hurbain

<http://philohome.com/index.htm>

Que tous les autres me pardonnent de ne pas les citer...

Je cite :

« Les fans de LEGO Mindstorms NXT© sont parmi les plus joueurs, les plus inventifs, et les plus authentiques personnes que vous puissiez rencontrer. Leurs perspectives sont pratiques, leur art est exquis. »