

# Algorithme de communication dans un ensemble de Lego Mindstorms NXT 2.0

Encadreur : HOUDA Keltoum , Etudiants : GUERMIT Radja – KLAOUA Wafa

Université Kasdi Merbah Ouargla

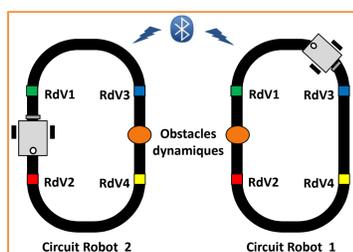
[houda.keltoum@univ-ouargla.dz](mailto:houda.keltoum@univ-ouargla.dz) ; [gr.radja@yahoo.com](mailto:gr.radja@yahoo.com) ; [klaoua.wafa@gmail.com](mailto:klaoua.wafa@gmail.com)

## Résumé.

Ce projet consiste à implémenter un algorithme de communication entre deux robots de type Lego NXT A l'aide du langage Java et de la technologie leJOS. Un capteur de couleur est utilisé pour le suivi de ligne ainsi qu'un capteur ultra-sonore pour la détection d'obstacle dynamique. En fin la technologie Bluetooth pour réaliser la communication entre les robots. Mots clefs : Communication Bluetooth, Lego , NXT, Robot Maître, Robot Esclave, Java, Suivi de ligne, Point de rendez-vous.

## Objectif (Projet global).

Le projet global consiste à réaliser une plateforme de robotique mobile modulaire et ouverte. Comportant deux (2) robots mobiles Autonomes identiques. Ces robots sont de types **Lego Mindstorms NXT 2.0**. Pouvant se déplacer dans un environnement dynamique entre plusieurs points de Rendez-vous. Et ceci en réalisant des tâches et en s'échangeant des messages sur ces points de Rendez-vous.



Le projet comporte deux aspects :

- Un aspect pratique avec la réalisation d'une plate-forme de robotique mobile comportant deux robots mobiles autonomes identiques. Ces robots sont de types **Lego Mindstorms NXT 2.0**. ils se déplacent chacun sur un circuit fermé de couleur noire et un fond blanc entre des points de rendez-vous et échangeant des messages lorsque le rendez-vous entre deux robots à lieu effectivement.
- Un aspect liée à la modélisation de l'ordonnancement des rendez-vous entre robots dans lequel les problèmes de compétition et de coopération sont traités ; la compétition entre les robots est liée au conflits générés par la gestion des ressources limitées alors que la coopération est liée à la tâche commune qu'ils doivent réaliser.

## Présentation de la gamme Lego Mindstorms NXT 2.0

### Interface NXT

La brique NXT est le cerveau du robot LEGO® MINDSTORMS. Cette brique LEGO contrôlée par ordinateur dote le robot d'un comportement programmable et intelligent et lui permet de prendre des décisions. Elle propose les entrées / sorties suivantes :

#### Ports de sortie

Trois ports de sorties A, B, C permettant de connecter et contrôler les moteurs .

#### Port USB

Un port USB permet le branchement à un ordinateur pour programmer la brique.

#### Haut-parleur

Lorsque des sons sont inclus dans un programme, vous pouvez les entendre par le haut-parleur.

#### Boutons NXT

Bouton orange : Activation/Entrée. Flèches gris clair : Navigation, gauche et droite. Gris foncé : Effacer/Retour.

#### Ports d'entrées

Quatre ports d'entrées notés 1, 2, 3 et 4 permettant le branchement de différents types de capteurs.

#### Bluetooth

La brique dispose d'une connectivité sans fil Bluetooth permettant l'ajout de programmes et de fichiers sans passer par le câble USB. Ainsi que la communication entre plusieurs NXT.

#### Ecran LCD

Celui-ci permet l'affichage du menu de la brique (lancement d'un programme enregistré sans l'intervention d'un ordinateur), et permet aux programmes l'affichage de textes, ou d'images.

## Modules d'entrées et de sorties de la brique NXT.

### Modules d'entrées :

Le Mindstorms NXT dispose de nombreux capteurs permettant aux robots une meilleure perception de son environnement.

### Modules de sorties :

Il n'existe qu'un seul module de sortie compatible, les servomoteurs fournis avec la brique NXT. Ces moteurs permettent les déplacements du robot.

#### NXT

Le NXT est une brique LEGO intelligente, qui constitue le cerveau d'un robot MINDSTORMS.

#### Servomoteurs interactifs

Permettent au robot d'effectuer des mouvements précis.

#### Capteurs tactiles

Permettent au robot de ressentir son environnement et d'y réagir.

#### Capteur d'ultrasons

Permet au robot de détecter les couleurs et la lumière. Peut aussi servir de lampe de couleur.

## Conception de la plateforme de robotique mobile.

Les robots conçus sont des robots autonomes à configuration différentielles. Ils comportent deux roues commandées indépendamment, une roue libre placée à l'arrière du robot pour assurer sa stabilité. Cette plate-forme est très simple à commander, puisqu'il suffit de spécifier les vitesses des deux roues, et permet de plus au robot de tourner sur place, chacun des robots est équipé :

- D'un cerveau comportant un **microprocesseur** 32bits ARM7d/Atmel.
- De deux **servo-moteurs** permettant la propulsion du robot.
- D'un **capteur couleur** permettant au robot d'effectuer un suivi de ligne et de détecter les différents couleurs des points de rendez-vous qui sont le : Vert – Rouge – Bleu – Jaune.
- D'un **capteur ultrasonore** permettant au robot de détecter d'éventuels obstacles dynamiques sur sa trajectoire, et de s'arrêter jusqu'à libération de celle-ci .



## L'application.

Notre application est constituée de deux robots mobiles autonomes, chaque robot se déplace sur un circuit fermé de couleur noire sur un fond blanc, ce circuit comporte quatre (04) points de rendez-vous de couleurs différentes (Vert – Rouge – Bleu – Jaune). L'ordre de visite des points de rendez-vous de chaque robot respecte le tableau suivant :

Robot 1			Robot 2		
Données du site	Données reçus	Comparaison	Données du site	Données reçus	Comparaison
45	12	Supériorité	12	45	Supériorité
106	45	Supériorité	45	106	Supériorité
25	189	Infériorité	189	25	Infériorité
98	98	Egalité	98	98	Egalité

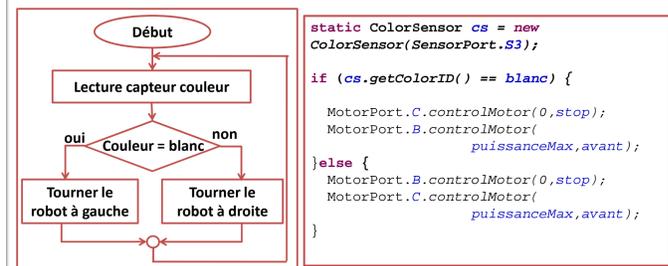
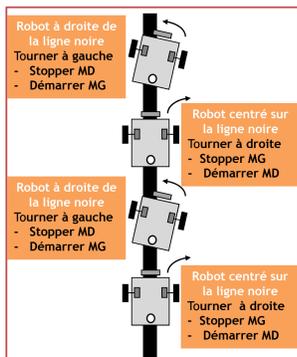
## Algorithmes employés : Suivi de ligne noire.

Cette première fonction à pour objectif d'utiliser le capteur couleur du robot pour permettre le suivi de la ligne noire.

Le capteur couleur est utilisé avec la fonction **CapteurCouleur.getColorID()** qui renvoie le numéro de la couleur détectée.

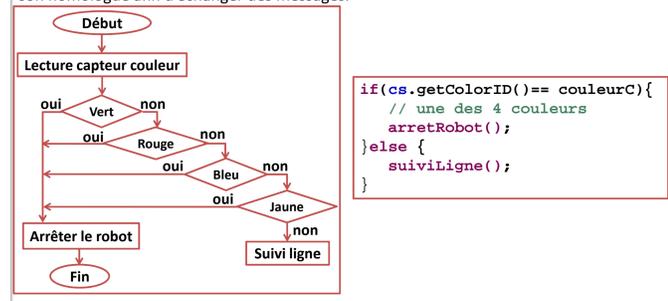
Le principe de l'algorithme est le suivant :

- Initialement, le robot sera placé sur le centre de la ligne noire.
  - Le robot va avancer en pivotant d'un côté ou de l'autre.
  - Lorsque le robot détecte la couleur noir, il avance en tournant vers la droite .
  - De ce fait, ce léger virage va déplacer le capteur couleur vers la couleur blanche.
  - Lorsque le robot détecte la couleur blanche, il avance en tournant vers la gauche, ainsi,
  - le robot va revenir vers l'axe de la ligne noir.
- C'est le cœur de notre algorithme, il va suffire qu'à faire effectuer ce cycle au robot de manière permanente.



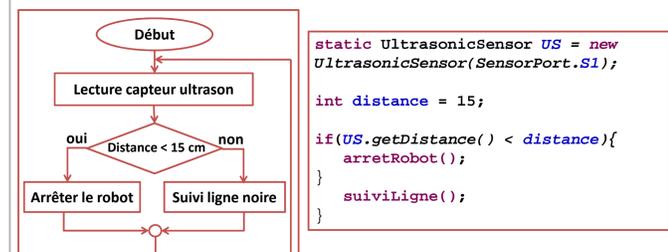
## Algorithmes employés : Détection de points de rendez-vous.

Pour détecter les points de rendez-vous qui sont représentés par des carrés de couleurs différentes (Vert – Rouge – Bleu – Jaune), nous utilisons le capteur couleur qui peut lire six différentes couleurs. Le capteur nous renvoie le numéro de la couleur détectée, si l'une des couleurs mentionnées ci-dessus est détectée, le robot s'arrête et se met en attente de son homologue afin d'échanger des messages.



## Algorithmes employés : Détection d'obstacle.

Pour détecter un éventuel obstacle dynamique, le robot utilise son capteur ultrasonore. La fonction **CapteurUltrason.getDistance()** renvoie la distance en cm de l'objet qui est en face du capteur. L'organigramme suivant présente la détection d'obstacle dynamique.



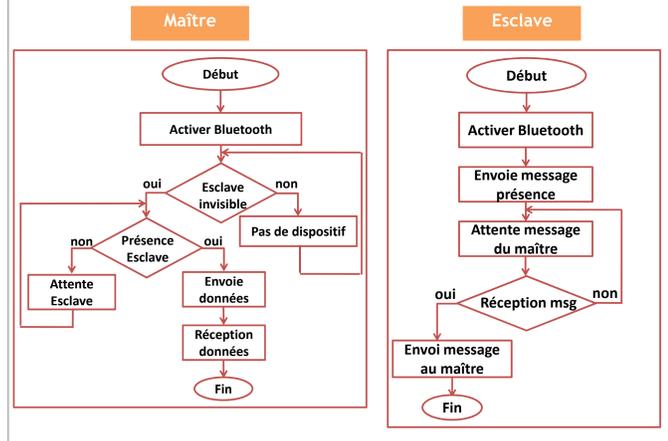
## Algorithmes employés : Communication Bluetooth.

La communication entre les deux robots est établie en utilisant la technologie Bluetooth entre robot Maître et robot Esclave. Il y a deux scénarios à considérer :

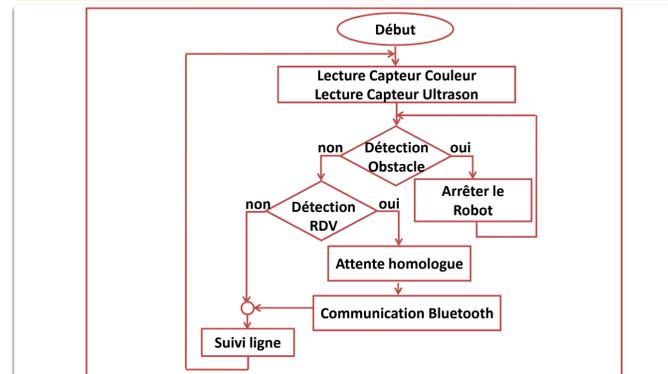
**Scénario 1 :** Le robot maître arrive au point de RDV avant le robot Esclave

**Scénario 2 :** Le robot maître arrive au point de RDV après le robot Esclave.

Les organigrammes suivants montrent le protocole de communication entre Maître et Esclave.



## Organigramme global.



## Discussion des résultats.

Le but du projet est de permettre à deux robots mobiles autonomes d'effectuer une tâche commune complexe en coordonnant et en communiquant leurs informations récupérées tout au long de leurs parcours en passant par des points de rendez-vous prédéfinies. La communication entre les deux robots est établie en utilisant la technologie Bluetooth entre robot Maître et robot Esclave.

L'un des problèmes rencontrés est le cas où le robot Maître arrive sur le point de rendez-vous avant l'arrivée du robot Esclave.

Dans le cas générale, pour établir un échange de donnée entre maître et esclave par Bluetooth, l'esclave doit être en attente du maître sinon il y a échec de l'exécution du programme embarqué sur le robot maître.

Pour remédier à ce problème, nous avons forcé le maître à rétablir une connexion avec l'esclave tant que ce dernier n'est pas disponible, c'est-à-dire, mettre le maître en attente de l'esclave.

En pratique, le suivi de ligne que nous avons effectué est un peu saccadé, toute fois, pour pouvoir faire un déplacement plus lisse, il faut utiliser un système de régulation.

Par contre, pour la détection des points de rendez-vous, le robots réussi à détecter l'une des quatre couleurs définissons les points de rendez-vous et s'arrêter immédiatement. Pour la détection d'obstacles dynamiques, le robot réussi à s'arrêter à la distance programmé qui est de 15 cm avec une erreur de 2 à 3 cm.

## Conclusion.

Ce projet intitulé « **Algorithme de communication dans un ensemble de Lego Mindstorms NXT 2.0** » nous a tout d'abord à nous initier à la robotique. De plus, la plateforme Lego NXT est un réel plaisir à utiliser et à programmer. Elle propose différents types de capteurs ainsi que de nombreux langages de programmation.

Pour ce projet, nous avons donc eu le choix des technologies employées. Nous avons choisi le capteur couleur pour le suivi de ligne et la détection des points de rendez-vous. L'utilisation du capteur ultrasonore nous a permis la détection d'éventuels obstacles dynamiques.

Et enfin, nous avons utilisé la technologie Bluetooth disponible sur les robots afin d'établir un échange de données pour permettre d'effectuer un objectif commun entre les robots. Nous avons entamé la partie pratique de notre projet début février et nous sommes à 50 % de son avancement.

La prochaine partie à faire consiste à implémenter un algorithme auto-stabilisant de visite des points de rendez-vous dans le cas de panne temporaire de l'un des robots. Ainsi qu'établir une communication Bluetooth avec l'ordinateur pour permettre un affichage de l'évolution des visites et de l'échange des données.

## Bibliographie.

- [1] Fabien Buda, Pierre Gaucher, « Résolution d'un labyrinthe à l'aide d'un robot Lego Mindstorms », Projet robotique, École Polytechnique de l'Université de Tours. 2013
- [2] Houda Keltoum, « Modélisation des systèmes complexes parallèles Application : ROBOTS mobiles autonomes évoluant dans un environnement parsemé d'obstacles mobiles ». Mémoire Magister, Université Badji Mokhtar Annaba 2006.
- [3] [www.lejos.org/](http://www.lejos.org/) Site sur le langage de programmation Lejos.
- [4] <http://www.sitedunxt.fr/faireDuSuiviDeLigne>