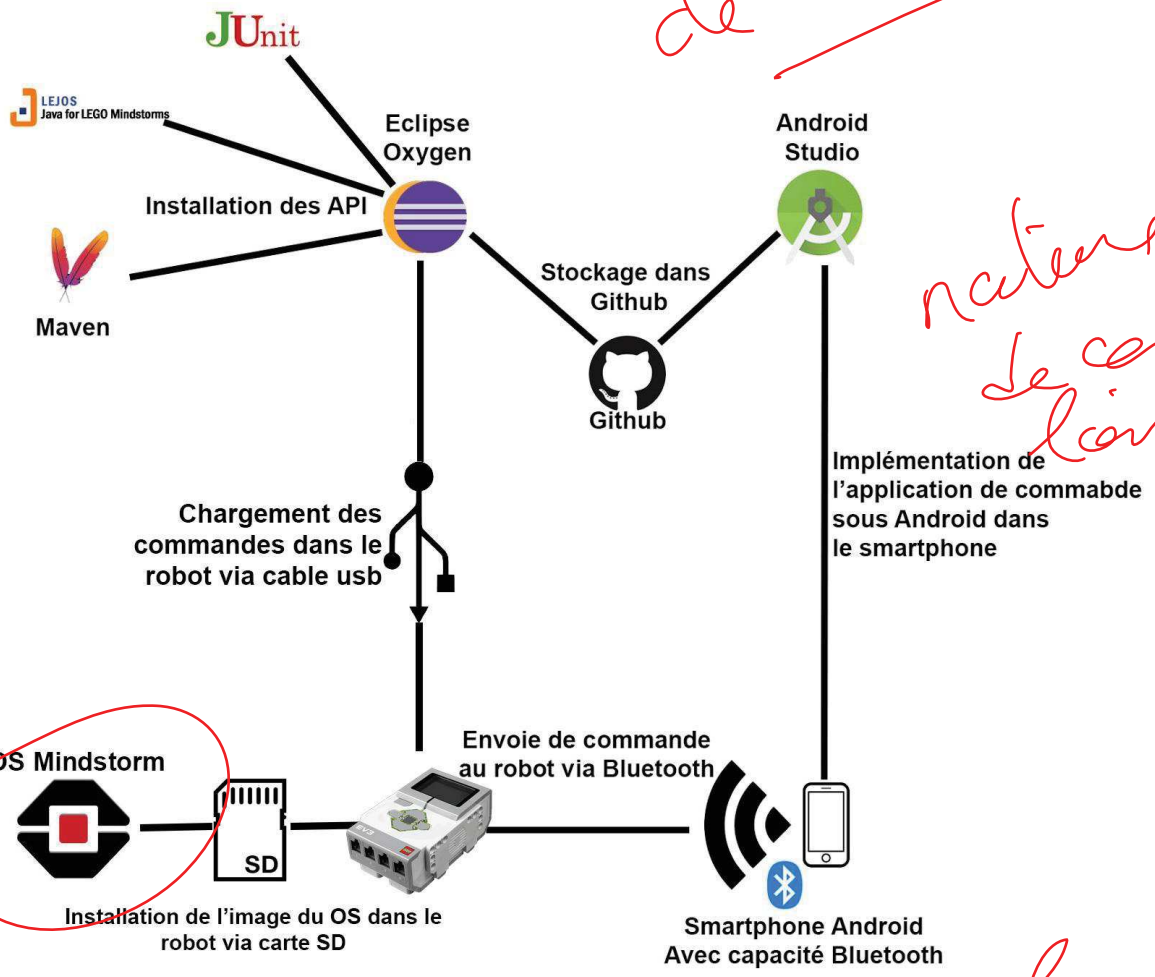
4. Liaison entre outils de développement

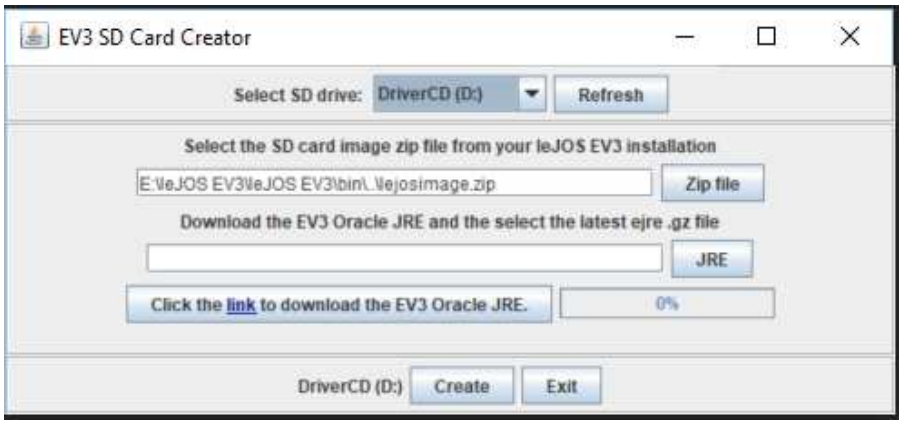


5. Manuel d'installation

Pour pouvoir développer un programme sur le robot lejos EV3, deux possibilités s'offraient à nous, la première était d'installer le programme "LEGO MINDSTORMS Education EV3" fourni par Lego, qui nous permet de faire de la programmation visuelle, en assemblant des composants. La seconde possibilité était de développer notre programme en Java. Pour des raisons de familiarité avec l'environnement, nous avons choisi d'utiliser le Java .

Nous avons ajouté à notre IDE, Eclipse, le plugin "lejos-ev3-plug", disponible sur son marketplace, qui va nous permettre d'utiliser l'api de lejos ev3, contenant tous les packages, classes et interfaces nécessaires à la manipulation du robot. Pour installer le plugin, nous devons aller dans le menu "windows > preferences " d'Eclipse, et d'y glisser le plugin depuis le marketplace.

Une fois notre environnement de travail installé et configuré, nous devons mettre dans une carte SD de deux giga octets minimum, l'image , à l'aide du logiciel "leJOS EV3 0.9.1-beta" qui va monter l'image dans la carte SD. Le logiciel requiert un jdk, et un jre, dans notre cas, nous avons utilisé comme environnement la version 1.7.0_80 du jdk. Une fois installé, le logiciel va lancer un programme "EV3 DS Card Creator".



Pour avoir le "jre" il faut aller sur le lien donné, et choisir le jre "Oracle Java SE Embedded version 7 Update 60 " puisqu'il est compatible avec la version du jdk que nous avons utilisé (le robot n'arrivait pas à utiliser convenablement la version du jre la plus récente). Une fois l'image montée dans la carte SD, nous l'avons mis dans le robot que nous avons allumé par la suite. Celui-ci débute automatiquement l'installation de l'image, qui prend une vingtaine de minutes.

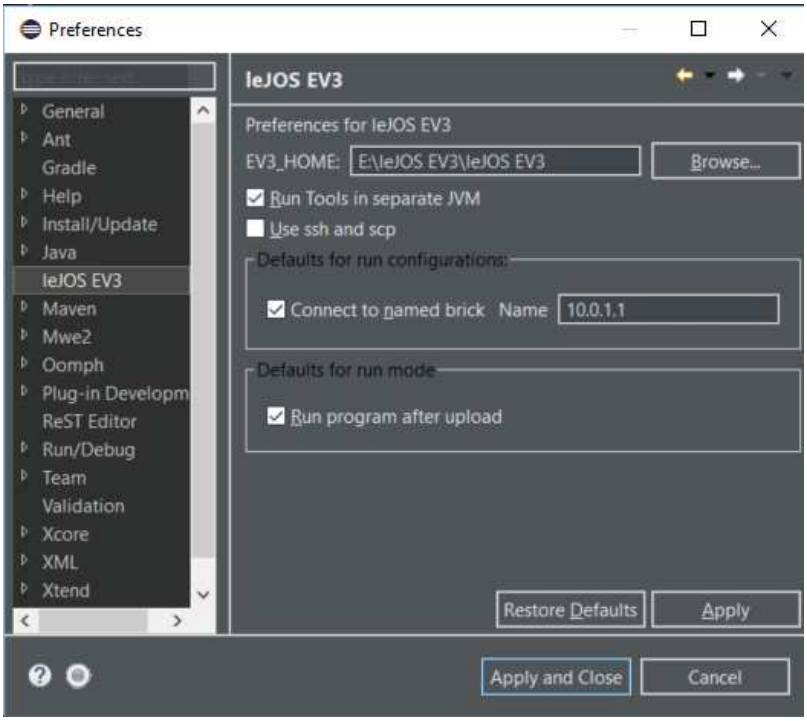
Une fois l'installation terminée, on arrive sur le menu suivant sur la boîte Mindstorm :

Menu d'accueil de Lejos

indiquer ce qu'il faut faire quand ça ne marche pas.

Nous avons aussi décidé que la connexion entre l'ordinateur et le robot, se fera via un câble ethernet USB pour l'envoie des programmes, et pas en mettant le programme dans la carte SD.
Nous avons trouvé qu'utiliser le câble fourni était plus simple, et l'exécution sur le robot se faisait très rapidement, alors que si nous avions choisi de mettre le programme dans la carte SD, le robot aurait recommencé le processus d'installation de l'image à chaque fois qu'on voudrait exécuter notre code, ce qui est très long.

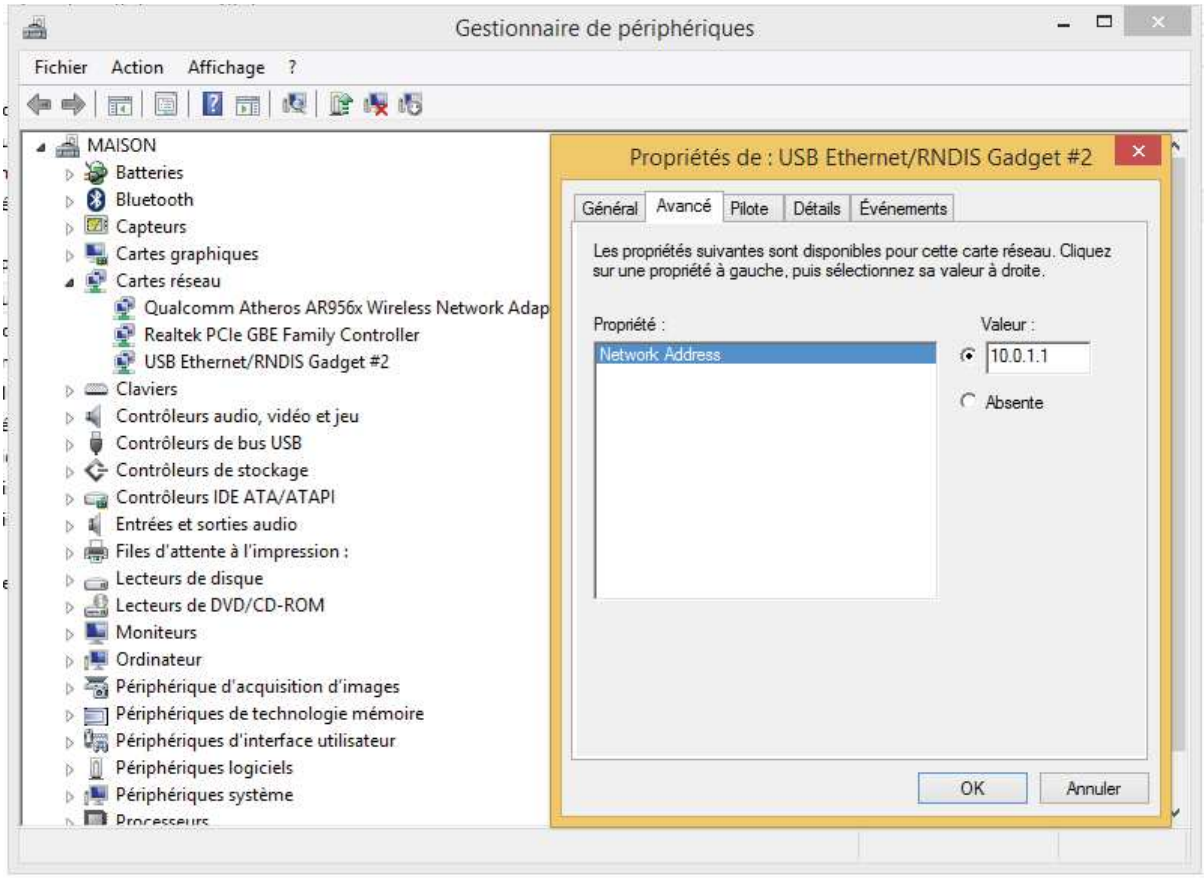
La connexion USB ethernet avec le robot est une tâche délicate, et a représenté la plus grande difficulté de la phase d'installation, il existe plusieurs drivers, et logiciels permettant d'établir la connexion, nous avons commencé par renseigner le plugin avec l'adresse IP du robot que l'on peut récupérer directement depuis l'écran principal du robot, dans notre cas, l'adresse IP est "10.0.1.1", via le menu "preferences > leJOS EV3" de l'IDE.



Configuration du plug in avec l'adresse IP du robot

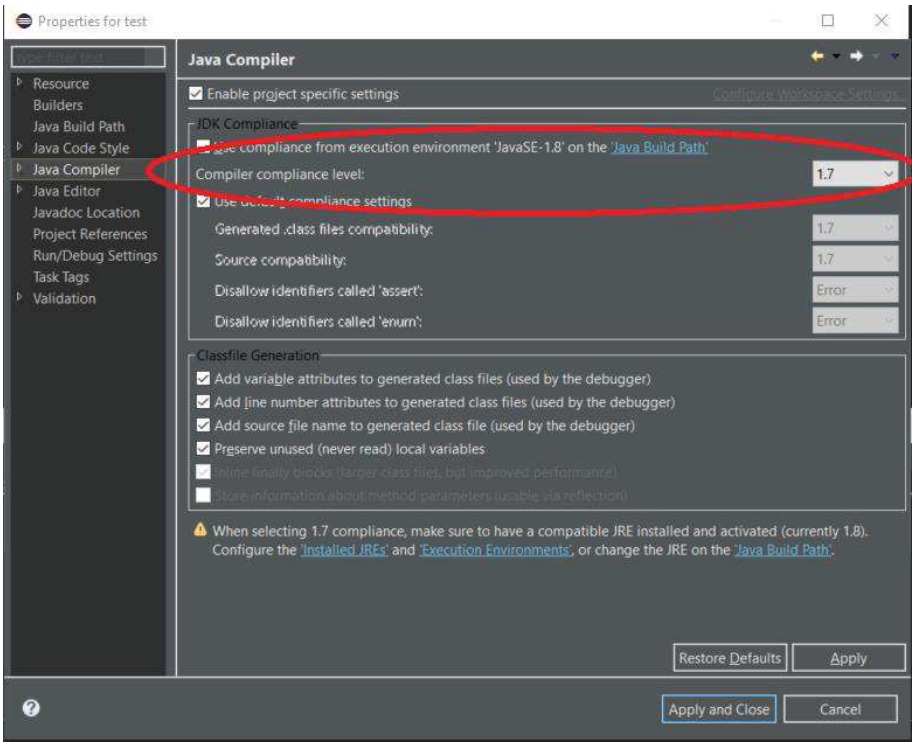
La première solution a été d'utiliser le logiciel "Open Roberta USB Program", qui permet de se connecter à un robot sans forcément renseigner son adresse IP dans le logiciel, à partir du moment où il est relié avec le robot via un câble USB . Cependant, notre machine ne détectait pas le robot. Cette solution a donc été abandonnée au profit d'une autre qui consistait à installer un driver spécifique pour ce genre de connexion.

Nous avons dû installer le driver "USB Ethernet/RNDIS Gadget " (à noter qu'il existe une version différente selon les systèmes d'exploitation) , ce driver permet d'établir une connexion entre l'ordinateur et une machine, donc nous avons renseigner l'adresse IP du robot dans la configuration du driver à partir du "gestionnaire de périphériques" dans la section "carte réseau ".



Configuration du driver

Pour créer un nouveau projet, nous devons nous rendre dans l'IDE, puis créer un projet en allant sur "file > new > project > other > LeJOS EV3 Project ". Une fois le projet créé, nous avons créé un package avec un nom arbitraire, dans lequel nous avons mis notre classe principal avec notre programme "main" qui affichera "Hello World". Pour pouvoir lancer le programme, ce dernier doit être compilé avec la même version de jdk que nous avons mis dans la carte SD (dans notre cas jdk 1.7), nous devons donc faire un clic droit sur "projet > propriété > Java Compiler" , et mettre la version du compilateur a 1.7 .



Changement de la version de java utilisé

Ensuite nous avons lancé le programme en faisant un clic droit sur le fichier “main” puis “Run as” et enfin “Lejos EV3 Program”. L'exécution de ce programme “Hello World” donne le résultat suivant :

Exécution du programme principal

Même après le débranchement du câble USB ethernet, on pourra retrouver le programme compilé sous forme d'un fichier "jar" , dans la section "programs" du menu d'accueil.

Liens utiles pour l'installation :

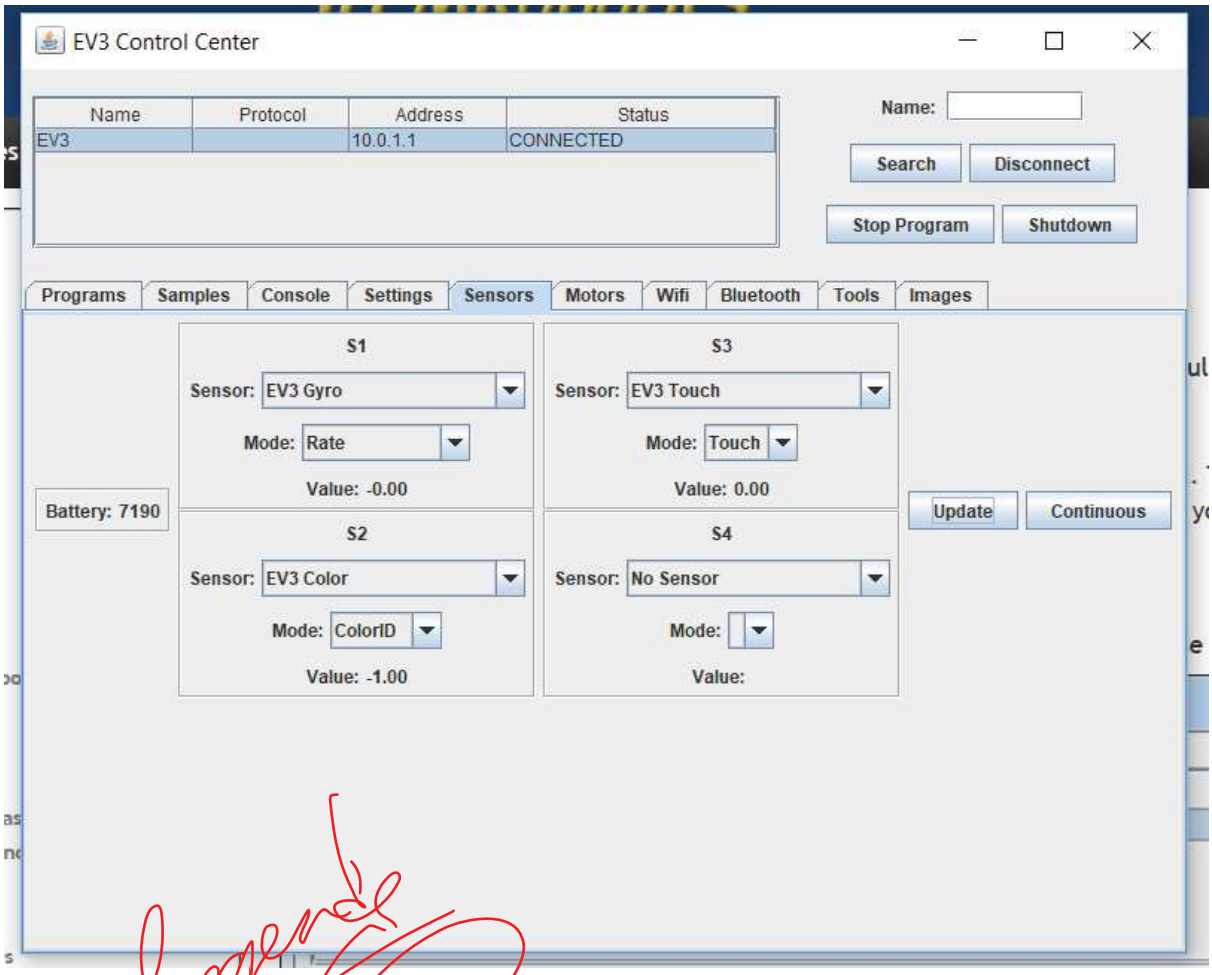
Installation d'eclipse : https://www.eclipse.org/downloads/
Installation du plugin : https://marketplace.eclipse.org/content/lejos-ev3-plug
Installation de leJOS EV3 0.9.1-beta : https://sourceforge.net/projects/ev3.lejos.p/files/
Driver "USB Ethernet/RNDIS Gadget : http://www.aplu.ch/home/ev3_inst.html

6. Prototype

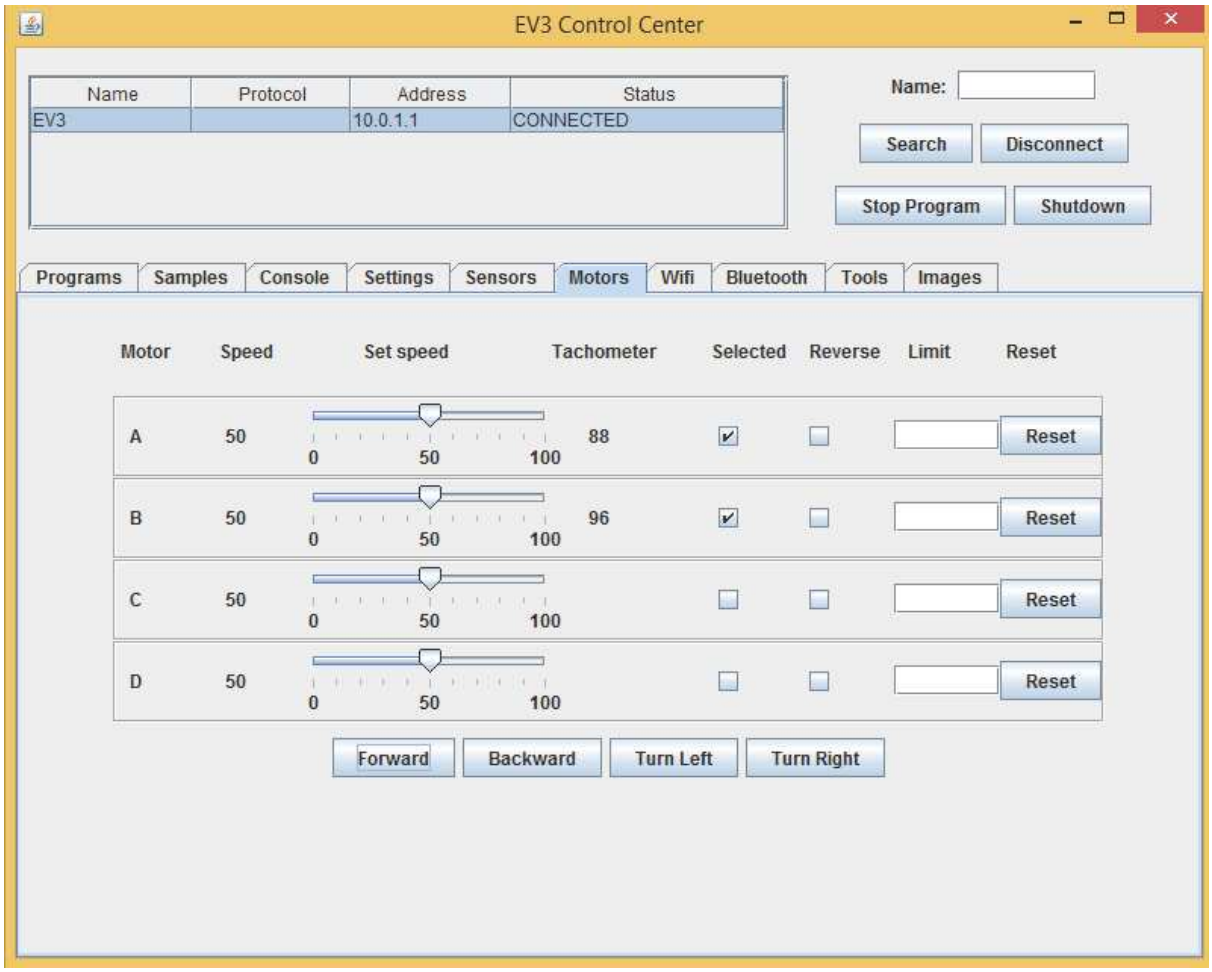
Dans cette section, nous allons explorer les composants du robot, et les dispositifs associés (capteurS, actionneurs), celui-ci possède plusieurs capteurs :

- Capteur gyroscopique : détermine, et change la direction de déplacement du robot.
- Capteur couleur : indique la couleur de l'objet.
- Capteur de distance UltraSonic.
- Capteur de proximité : indique si un obstacle est proche

Pour faire fonctionner les capteurs, il faut, après avoir allumé le robot, lancer "EV3 Control Center", et associer dans l'onglet "Sensors" chaque capteur à son port. Sur le robot, les capteurs possèdent les ports de 1 à 4, et les moteurs possèdent les ports A, B, C, D.

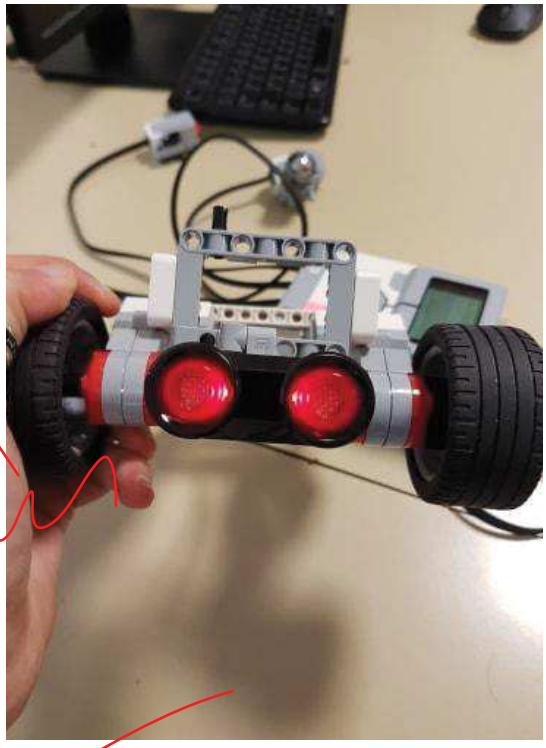


Paramétrage des capteurs du robot



Paramétrage des moteurs du robot

Une fois que le contrôleur est mis à jour et que les capteurs sont configurés, ces derniers s'allument.



Capteur de distance UltraSonic

Capteur couleur.

Nous avons choisi de tester uniquement le fonctionnement du capteur couleur, celui-ci doit être proche de la surface (environ 1 centimètre), et possède également plusieurs modes, selon le résultat que l'on souhaite obtenir :

- ColorID
- Red
- RGB
- Ambiant

Pour effectuer nos tests, nous avons récupéré un programme directement depuis les liens suivant :

http://stemrobotics.cs.pdx.edu/node/5200?root=4196
https://github.com/stemrobotics/EV3-Exercises/blob/master/ColorSensor.java
https://github.com/stemrobotics/EV3-Exercises/blob/master/Lcd.java

Une fois le programme exécuté, on peut voir que le résultat affiché sur l'écran change à chaque fois que la couleur devant le capteur change.

7. Conclusion

Afin de fixer notre environnement de déploiement et de développement, nous avons choisi comme langage de programmation java et android. Grâce aux outils d'android, il est possible pour nous de transformer notre smartphone en télécommande. Grâce aux outils java, nous pouvons implémenter les fonctionnements du robot en simulant le langage LeJos. Suite à cela nous avons vérifié l'installation de notre environnement de développement en connectant l'EV3 et ses dispositifs associés. Il en ressort que la connexion est un succès puisque le message "hello world" utilisé pour la tester, s'affiche sur le terminal de l'EV3. Nous pouvons donc passer à la phase suivante du projet qui est la réalisation d'un dossier de conception générale.