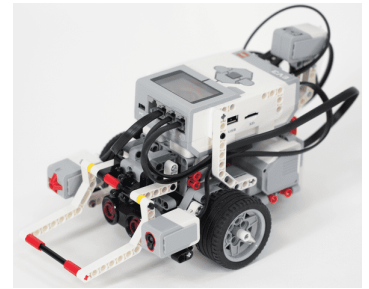


Objectifs :

- ⇒ Savoir définir ce qu'est la robotique du côté informatique
- ⇒ Manipuler des capteurs et des actionneurs de robot
- ⇒ Programmer des tâches simples sur un robot



I - Qu'est que la robotique ?

La robotique est l'ensemble des techniques permettant la conception et la réalisation de machines automatiques ou de robots.

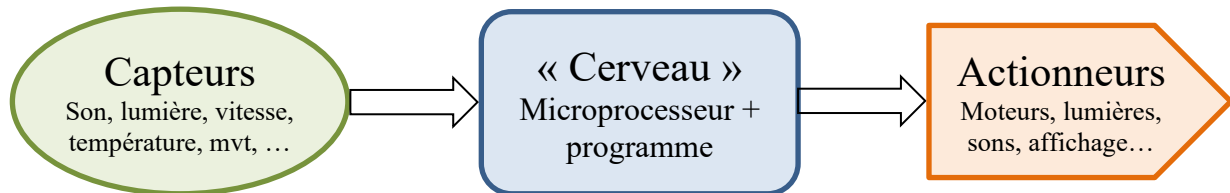
Mais qu'est-ce qu'un robot ? Il existe de nombreuses définitions possibles. On peut en retenir une assez simple :

Un robot est une machine chargée d'effectuer une ou plusieurs tâches de manière autonome.

Certains incluent dans la définition de robot des dispositifs purement logiciels (des programmes), mais on attribue plutôt le terme de « bots » à ces programmes automatiques.

Le mot-clé est ici l'autonomie : le robot dispose d'un programme suffisamment sophistiqué pour lui permettre d'accomplir seul la tâche qui lui est dévolue.

Les robots actuels peuvent être décomposés en trois parties :



Ce sont les capteurs qui renseignent le robot sur son environnement extérieur et les actionneurs qui lui permettent d'agir afin de se déplacer ou d'accomplir sa tâche. On peut considérer que les capteurs sont les « entrées » du programme du robot et les actionneurs ses « sorties ».

Depuis de nombreuses années, les robots ont envahi l'industrie pour effectuer des tâches complexes et répétitives avec une précision constante, mais on en trouve de plus en plus dans des domaines plus variés comme le domestique, le médical, le spatial ou le militaire avec de nombreuses questions soulevées tant sur le plan technique qu'éthique.

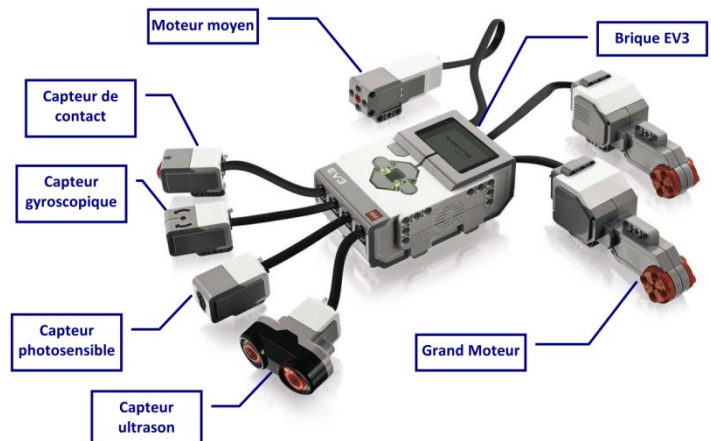
Voir <https://fr.wikipedia.org/wiki/Robot> ou <https://www.futura-sciences.com/tech/dossiers/robotique-initier-former-robotique-889/page/3/>

II - Programmation d'un robot

1) Présentation du système LEGO MINDSTORM ev3

Nous allons maintenant essayer de programmer un petit robot réalisé avec des LEGO MINDSTORM. Le cerveau du système est la brique intelligente EV3. Elle intègre un microprocesseur, 4 ports d'entrées pour y connecter des capteurs, 4 ports de sorties pour y connecter des actionneurs (ici des moteurs) ainsi qu'une connectivité Bluetooth et USB. L'ensemble est géré par un système d'exploitation linux par-dessus lequel LEGO a disposé une interface propriétaire permettant le transfert et la création de programme ainsi que quelques opérations de base.

Nous verrons dans cette séance la programmation du robot en python via l'EDI Visual Studio Code.



Pour programmer le robot en python, il est nécessaire de disposer d'un interpréteur python qui s'exécute directement sur la brique. Pour ce faire, on utilise une carte SD contenant le système [ev3dev](#) sur laquelle le système va booter et qui contient un interpréteur [micropython](#) (une version de python optimisée pour les microcontrôleurs) ainsi que les outils permettant au PC de communiquer avec la brique ev3.

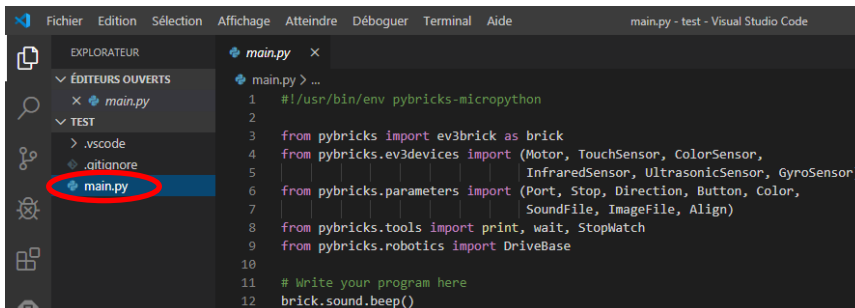
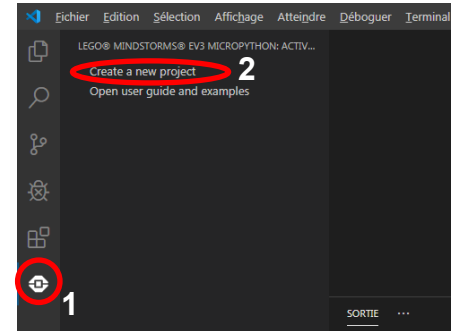
2) Prise en main

a. Logiciel VSCode



Lancer le logiciel VSCode. Sur l'écran d'accueil, dans la barre de gauche, cliquer sur la dernière icône « LEGO MINDSTORM ev3 Micropython », sélectionner « Create a new project »,

Donner un nom au projet : « Test » par exemple, puis sélectionner le fichier main.py qui contient le programme principal :



On remarque la première ligne qui commence par « #! ».

C'est un [shebang](#) qui permet au système de la brique ev3 d'exécuter le programme en utilisant l'interpréteur micropython. Il ne faut surtout pas la toucher et toujours la laisser au tout début du programme.

Viennent ensuite plusieurs lignes d'import qui servent à rendre disponible pour le programme toutes les fonctions spécifiques au système ev3. On a accès à la documentation complète de toutes ces fonctions via le bouton en bas de la barre latérale gauche, puis option « Open user guide and examples » (ou directement en ligne sur <https://pybricks.github.io/ev3-micropython/>).

Ce premier programme ne contient qu'une seule instruction (à la ligne 12) qui permet de produire un son.

b. Connexion à la brique ev3 et exécution d'un programme

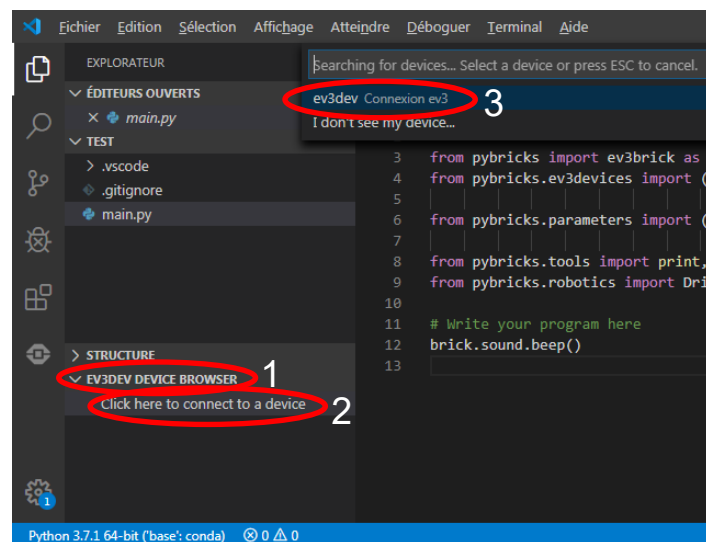
Nous allons maintenant transférer le programme dans l'ev3 et l'exécuter.

Commencer par démarrer la brique ev3 en appuyant sur le bouton central (le démarrage de la brique prend environ 2 minutes pour démarrer).

Connecter ensuite le câble USB entre la brique ev3 (connecteur mini-USB marqué « PC ») et un port USB en façade du PC.

Dans VSCode, cliquer ensuite tout en bas du volet de gauche sur « EV3DEV DEVICE BROWSER », puis sur « Click here to connect to a device » et enfin sur la fenêtre qui s'ouvre en haut sur « ev3dev Connexion ev3 ».

On obtient alors d'abord un rond jaune sous EV3 DEVICE BROWSER, puis il devient vert pour signaler que la connexion est établie.



Les DEL lumineuses du clavier de l'ev3 deviennent également verte.

Pour transférer et exécuter notre programme sur la brique, il nous suffit maintenant d'appuyer sur la touche F5 (ou menu Débugger/Démarrer le débogage). L'opération est plus longue qu'un programme python classique car VSCode doit d'abord téléverser le script python dans l'ev3, procéder à quelques réglages, puis demander à la brique ev3 l'exécution du fichier. Celle-ci lance l'interpréteur micropython et enfin le programme est exécuté. Exécuter le programme et vérifier que la brique émet bien le « bip » attendu.

3) Premier programme

Maintenant que nous savons utiliser VSCode pour écrire du code, le transférer vers l'ev3 et l'exécuter, on cherche à écrire un premier programme de déplacement simple du robot.

Question 1 :

- 1) Démarrer un nouveau projet (appelé « Premier_programme ») et écrire un programme qui fait avancer le robot en ligne droite à vitesse réduite (100 mm/s) jusqu'à ce qu'il se trouve à moins de 15 cm d'un obstacle (après quoi le programme s'arrête). Tester et valider le programme.
- 2) Améliorer le programme précédant en faisant faire un demi-tour sur place au robot après s'être arrêté devant l'obstacle.
- 3) Modifier encore le programme pour que le robot répète les opérations précédentes (avant/obstacle/demi-tour) pendant 20 secondes, après quoi le robot émet un bip et s'arrête.

4) Radar de recul

Question 2 :

- 1) Ecrire en pseudo-code l'algorithme d'un radar de recul utilisant le télémètre ultrason du robot et répondant au cahier des charges suivant :
 - Le radar émet des bips réguliers ayant une fréquence de 300 Hz et durant 30 ms chacun
 - Le radar doit émettre des bips uniquement si l'obstacle se trouve à moins de 50 cm
 - Les bips doivent être d'autant plus rapprochés que l'obstacle est proche
- 2) Démarrer un nouveau projet (appelé « Radar_de_recul ») et écrire un programme répondant au cahier des charges. Le programme doit fonctionner pendant 20 secondes puis s'arrêter.

5) Utilisation du capteur de couleur

On va utiliser un capteur supplémentaire : le capteur de couleur et modifier le comportement du robot en fonction de la couleur du sol.

Question 3 :

Ecrire un nouveau programme (« Capteur_de_couleur_sol ») qui fait avancer le robot jusqu'à ce qu'il rencontre un obstacle devant lui. Si le sol est de couleur bleue à ce moment-là, le robot tourne à droite à 90° et reprend son chemin. Si le sol est rouge, il s'arrête définitivement et émet trois bips. Si le sol est d'une autre couleur, il fait demi-tour puis continue à avancer.
Modifier ensuite le programme pour que le robot s'arrête au plus tard au bout de 20 secondes, même s'il n'a pas rencontré de sol rouge (mais sans les bips dans ce cas).

6) Robot trieur

Dans cette dernière partie, on va se servir du bras du robot qui permet de faire descendre le bras articulé devant lui et ainsi de « capturer » ou de repousser une caisse qui se trouverait devant lui. La commande de ce bras se fait via le moteur A (le petit moteur). On peut le déplacer d'un angle d'environ +80° pour le descendre ou -80° pour le monter.

Le scénario du robot est le suivant : On a disposé en ligne droite tous les 20 cm environ une série de cubes de couleur. Le but du robot est de trier les caisses en fonction de leurs couleurs (voir programme, puis figure ci-après).

Pour que le robot puisse détecter la couleur des caisses, il faut déplacer le capteur de couleur pour qu'il « éclaire » les caisses devant le robot comme dans la photo ci-contre.

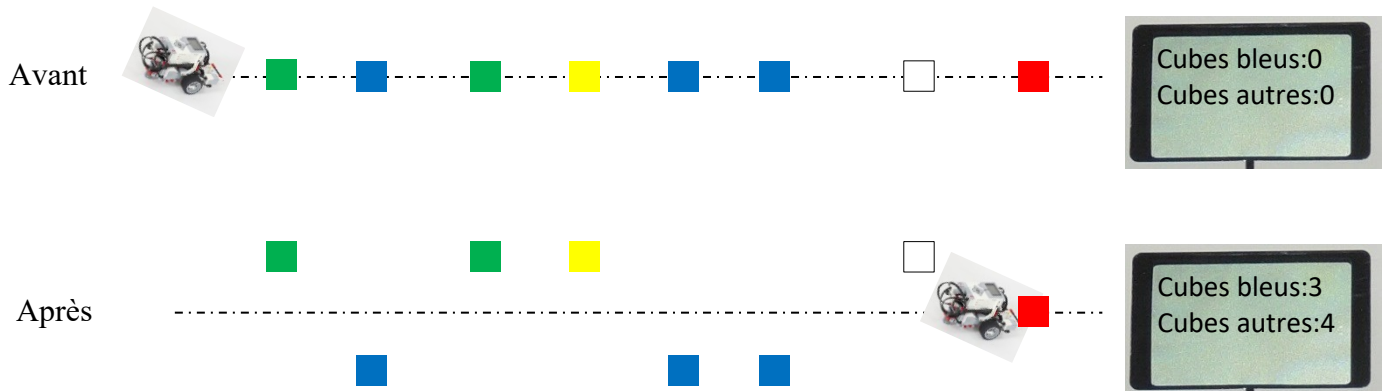


Question 4 :

1) Ecrire un programme « Robot trieur » qui fasse avancer le robot jusqu'à ce qu'il rencontre un cube.

- ⇒ Si ce cube est rouge, le programme s'arrête.
- ⇒ Si ce cube est bleu, le robot le capture et le met à droite de sa ligne de déplacement, puis reprend sa progression.
- ⇒ Si ce cube est d'une autre couleur, il le place à gauche et reprend son déplacement.

2) Améliorer le programme pour qu'il compte et affiche en permanence le nombre de cubes bleu traités et le nombre de cubes d'une autre couleur que bleu ou rouge.



Pour aller plus loin, quelques ressources internet :

http://www.lepoint.fr/high-tech-internet/les-robots-humanoides-sont-deja-parmi-nous-24-02-2016-2020682_47.php

<https://www.futura-sciences.com/tech/actualites/robotique-sophia-robot-veut-detruire-humanite-62142/>

<https://www.youtube.com/watch?v=p647ezfEows>

Blog robotique : <https://humanoides.fr/>, notamment <https://humanoides.fr/robocup-football-france/>