

**DOSSIER D'INITIATION A LA**

**PROGRAMMATION GRAPHIQUE**

Présentation :

Il s'agit d'un robot Mbot du commerce.

Il est doté d'une carte électronique programmable. Celle-ci peut être programmé grâce à un ordinateur soit par fil ( port USB ), soit par Wifi.

On utilisera le logiciel « Mblock » pour créer nos programmes. Le langage graphique utilisé est le scratch.



**SOMMAIRE**

**PRÉSENTATION DU MATÉRIEL** ….................................................................... p1

**LOCALISER LES CAPTEURS ET ACTIONNEURS** …....................................... p2

**L'ACCUEIL DU LOGICIEL DE PROGRAMMATION** …........................................ p3

**LA MÉTHODE POUR CRÉER LE PROGRAMME** …............................................ p4

**VOS PREMIERS PAS DANS LA PROGRAMMATION** …...................................... p5

## RAPPEL DU CONTENU DES BIBLIOTHÈQUES COURAMMENT UTILISÉES ... p6

**LA DÉMARCHE POUR ENVOYER LE PROGRAMME DANS LE ROBOT** …...... p7

**LES PROGRAMMES A RÉALISER 1 / 3** …....................................................... p8

**LES PROGRAMMES A RÉALISER 2 / 3** …....................................................... p9

**LES PROGRAMMES A RÉALISER 3 / 3** …....................................................... p10

**PRÉSENTATION DU MATÉRIEL**

Le robot mBot interagit avec son environnement en fonction du programme qu'on lui implante. Pour cela, il est capable de collecter des informations grâce à ses capteurs et de réaliser des actions grâce à ses actionneurs.

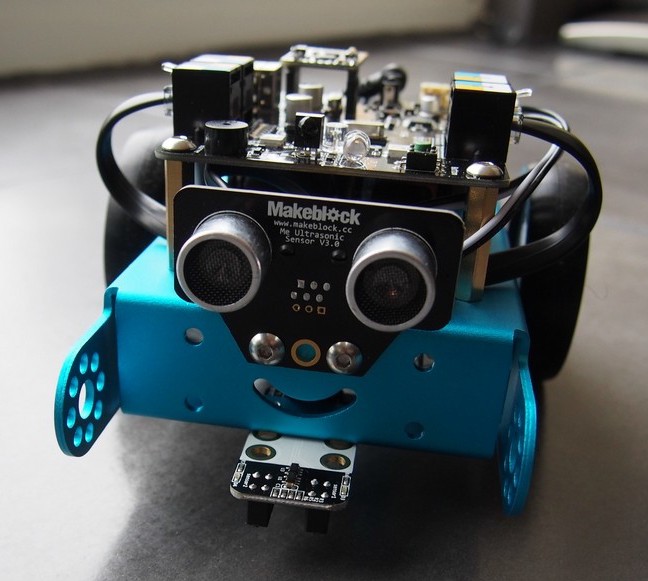
#### Actions et actionneurs :

* le robot est capable de **se déplacer** : il est équipé de deux moteurs indépendants qui pilotent chacun une roue motrice.
* il peut **émettre des sons** grâce à un buzzer.
* il peut **émettre de la lumière** grâce à 2 DEL RGB dont la couleur est paramétrable.

#### Boutons et capteurs :

Pour interagir avec son environnement et y recueillir des informations, on retrouve sur le robot :

* un module Wifi qui permet de recevoir les ordres émis par l'ordinateur.
* un capteur de luminosité qui le renseigne sur la luminosité ambiante.
* un module à ultrasons qui lui permet de « voir » les obstacles à l'avant et d'en connaître la distance.
* un module de suivi de ligne au sol.
* un bouton paramétrable ( on peut donc choisir la fonction qu'on veut pour ce bouton )
* un bouton de mise sous tension ON / OFF.

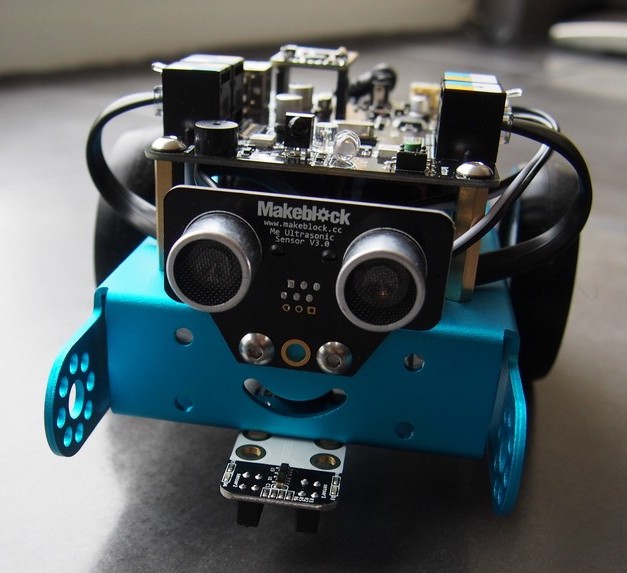
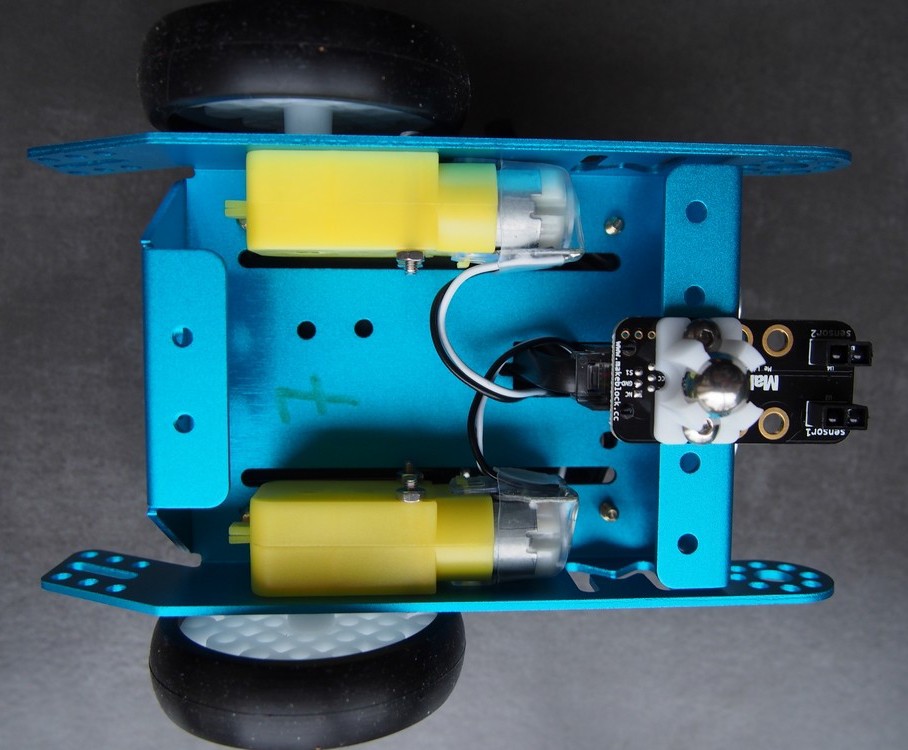
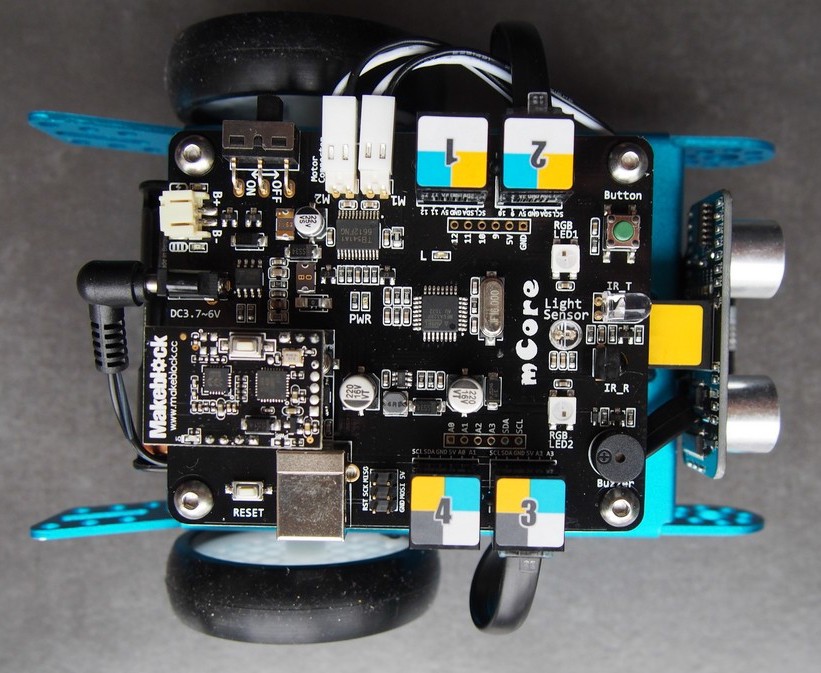


1

**LOCALISER LES CAPTEURS ET ACTIONNEURS**

Bouton de mise sous tension ON / OFF

LED RGB1



bouton

Capteur de luminosité

LED RGB2

Buzzer

Module wifi ou bluetooth

Moteur de la roue droite

Module de suivi de ligne

Moteur de la roue gauche

Module à ultrasons

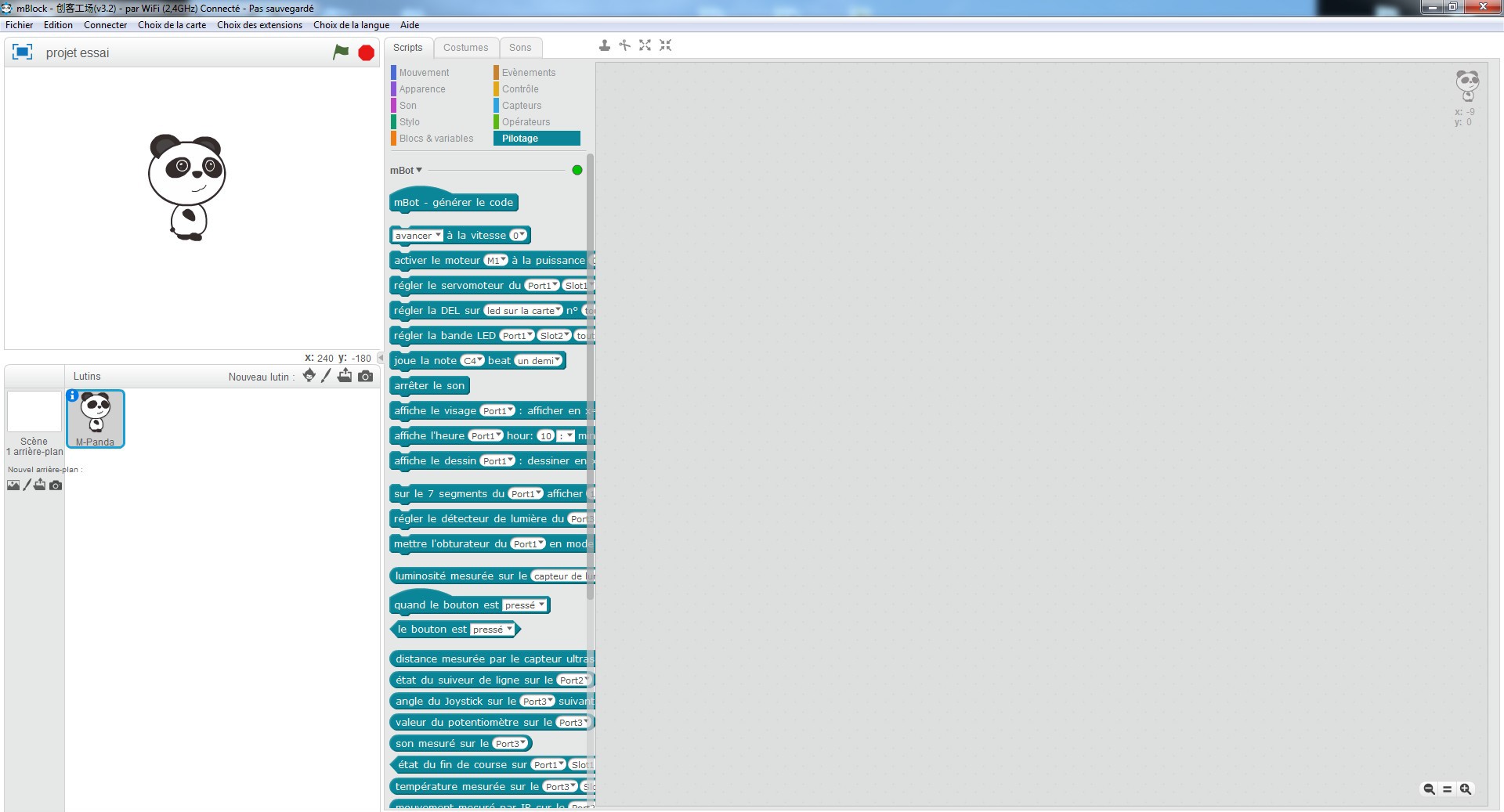
Module de suivi de ligne

2

**L'ACCUEIL DU LOGICIEL DE PROGRAMMATION**

* 1. Lancez le logiciel mBlock qui se trouve dans : Démarrer / programmes / technologie / mBlock
  2. Dans « choix de la carte », sélectionnez mBot puis dans « choix des extensions », sélectionnez Makeblock.

Vous trouverez ici les instructions de l'onglet sélectionné. Ici, on voit celles de « Pilotage ».



Zone dans laquelle

vous allez saisir votre programme

Vous trouverez ici les différentes bibliothèques d'instructions pour concevoir vos programmes. En ce qui nous concerne, nous utiliserons essentiellement :

* + - Evènements
    - Contrôle
    - Opérateurs
    - Pilotage
  1. Pour créer le programme, il suffit de glisser les éléments de la bibliothèque dans la zone de création du programme. Attention, il faut parfois les déposer avec précision pour que cela s'imbrique.

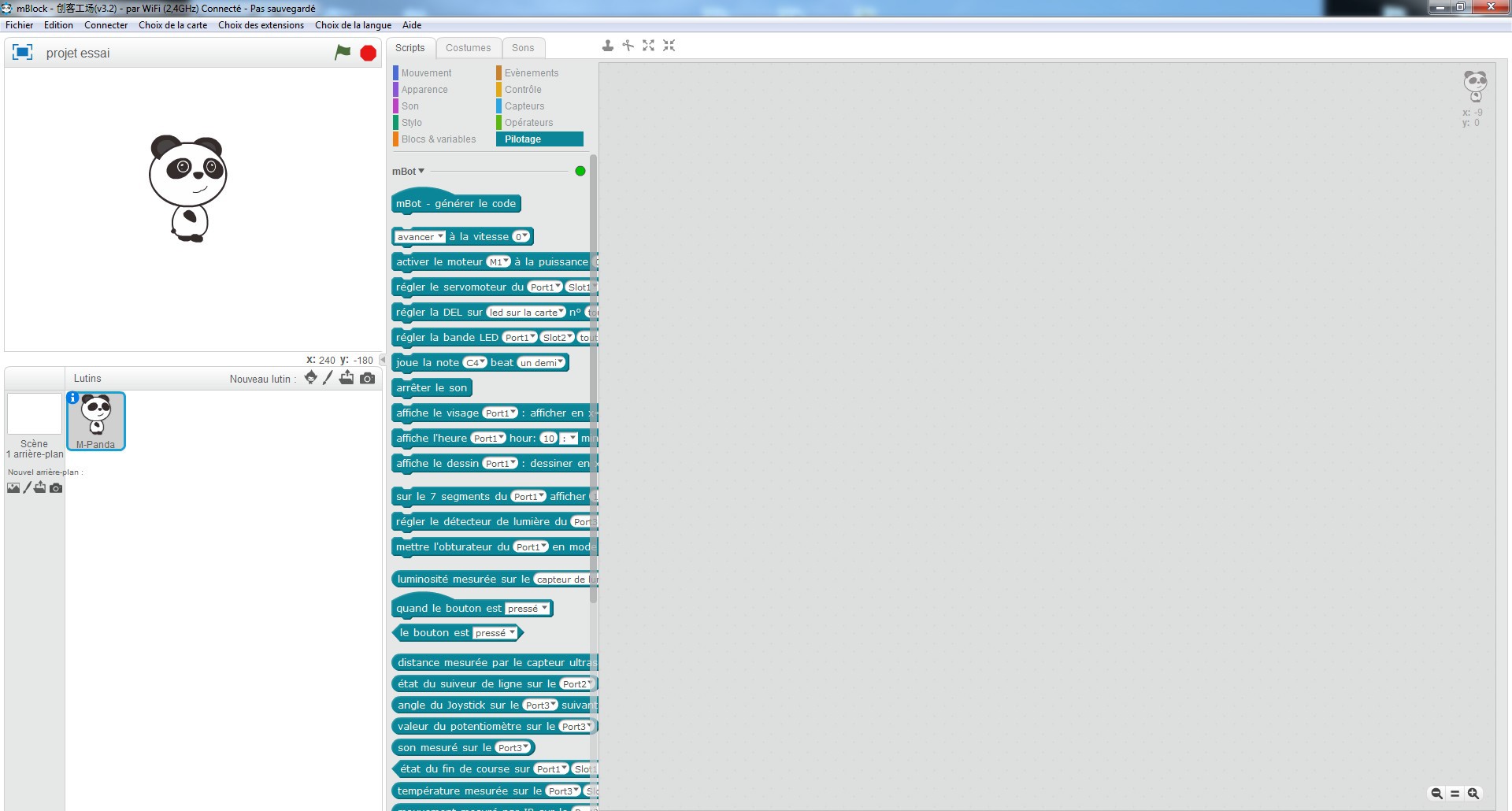
3

**LA MÉTHODE POUR CRÉER LE PROGRAMME**

1. Je sélectionne la bonne bibliothèque d'instruction



1. Une fois les différentes instructions affichées, je choisis celle qui m'intéresse et la glisse dans la zone de droite.



#### Cliquer – glisser - déposer

**Pour que les blocs s'imbriquent correctement, il faut être précis en les positionnant.**

**!**

**SUPPRIMER UNE INSTRUCTION :**

**!**

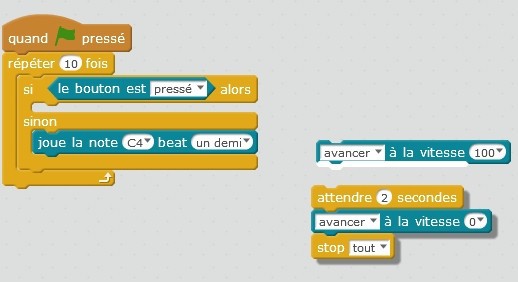
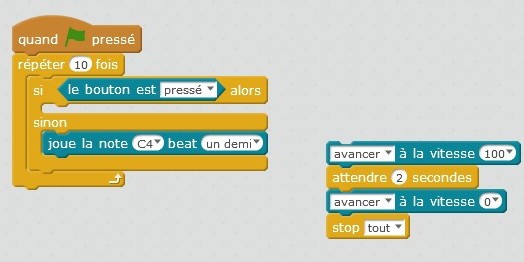
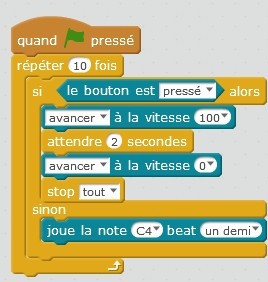
### Pour supprimer une instruction non souhaitée, il suffit de la re-glisser dans la bibliothèque à gauche. Attention, il faut parfois procéder par étapes car les instructions imbriquées viennent ensemble.

Exemple : pour supprimer le bloc « avancer à la vitesse 100»

**1**- je saisis le bloc par sa première instruction et le je le sors du programme **2**- je saisis « attendre 2 secondes » et je le dissocie des autres instructions pour isoler l'instruction que je veux effacer.

A

**3**- je peux maintenant supprimer l'instruction « avancer à la vitesse 100 »



A

A

4

**VOS PREMIERS PAS DANS LA PROGRAMMATION**

1. Votre programme commencera par :

Si vous souhaitez que le robot enregistre votre programme pour l'exécuter ensuite. ( on pourra mettre « attendre jusqu'à – le bouton est pressé » pour que le programme démarre à l'appui sur le bouton du robot. )

1. Vous devez ensuite choisir la structure de base de votre programme. Dans un premier temps, nous utiliserons deux types de structures :

 La **boucle infinie**

OU

 La **boucle avec compteur** (qui s'exécute un nombre de fois déterminé )

1. Ensuite vous pouvez enchaîner les instructions dans l'ordre souhaité.

**EXEMPLES DE PROGRAMMES SIMPLES :**

Le robot doit se déplacer en avant à la vitesse 100 pendant 3 secondes puis s'arrêter.

**Réalisez ce programme, téléversez-le dans le robot en suivant les instructions page 7 et testez-le !**

Le robot doit se déplacer en avant à la vitesse 100 pendant 3 secondes puis s'arrêter 3 secondes. Le programme se répète 2 fois.

**Complétez ce programme et testez-le sur le robot.**



**!**

**REMARQUE : les actionneurs ne s'arrêtent pas tout seul. Il faut TOUJOURS leur donner l'ordre de s'arrêter (à l'exception du buzzer car on détermine la durée de la note jouée )**

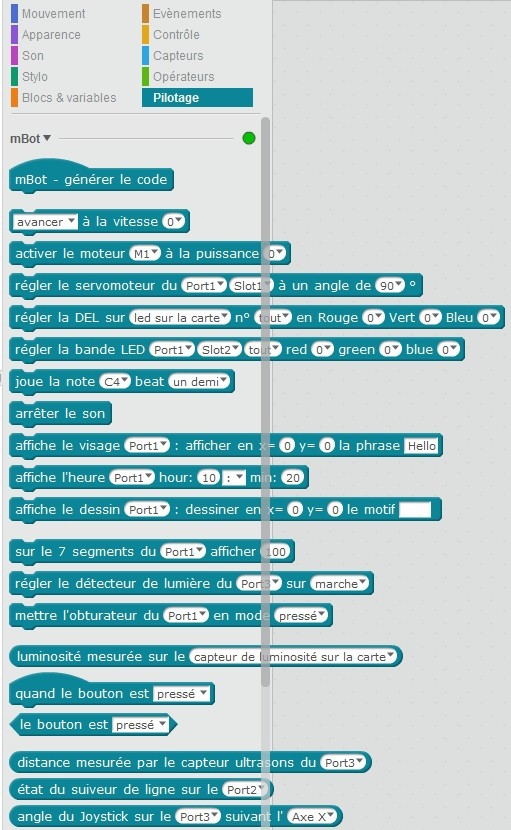
5

**RAPPEL DU CONTENU DES BIBLIOTHÈQUES COURAMMENT UTILISÉES.**

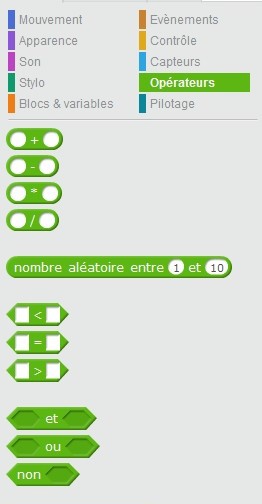
Le marqueur indique que vous aurez à utiliser cette commande dans les programmes que vous devrez créer.

# Bibliothèque PILOTAGE

Bibliothèque CONTRÔLE



Bibliothèque OPÉRATEURS



6

**LA DÉMARCHE POUR ENVOYER LE PROGRAMME DANS LE ROBOT**

Vous allez créer tous les programmes proposés dans l'ordre.

1. Poser le robot sur un support, branchez-le à l'ordinateur et vérifiez sur le robot que le commutateur est bien sur « **ON** ».
2. Allez dans l'onglet « **Connecter** », « **par port série** », choisissez le bon port COM puis assurez-vous que l'instruction « **connecter** » est bien activée.
3. Une fois le programme prêt à être testé, cliquez sur « édition », « mode arduino ». A ce moment-là, à droite apparaît votre programme en langage C. Vous n'avez plus qu'à cliquer sur

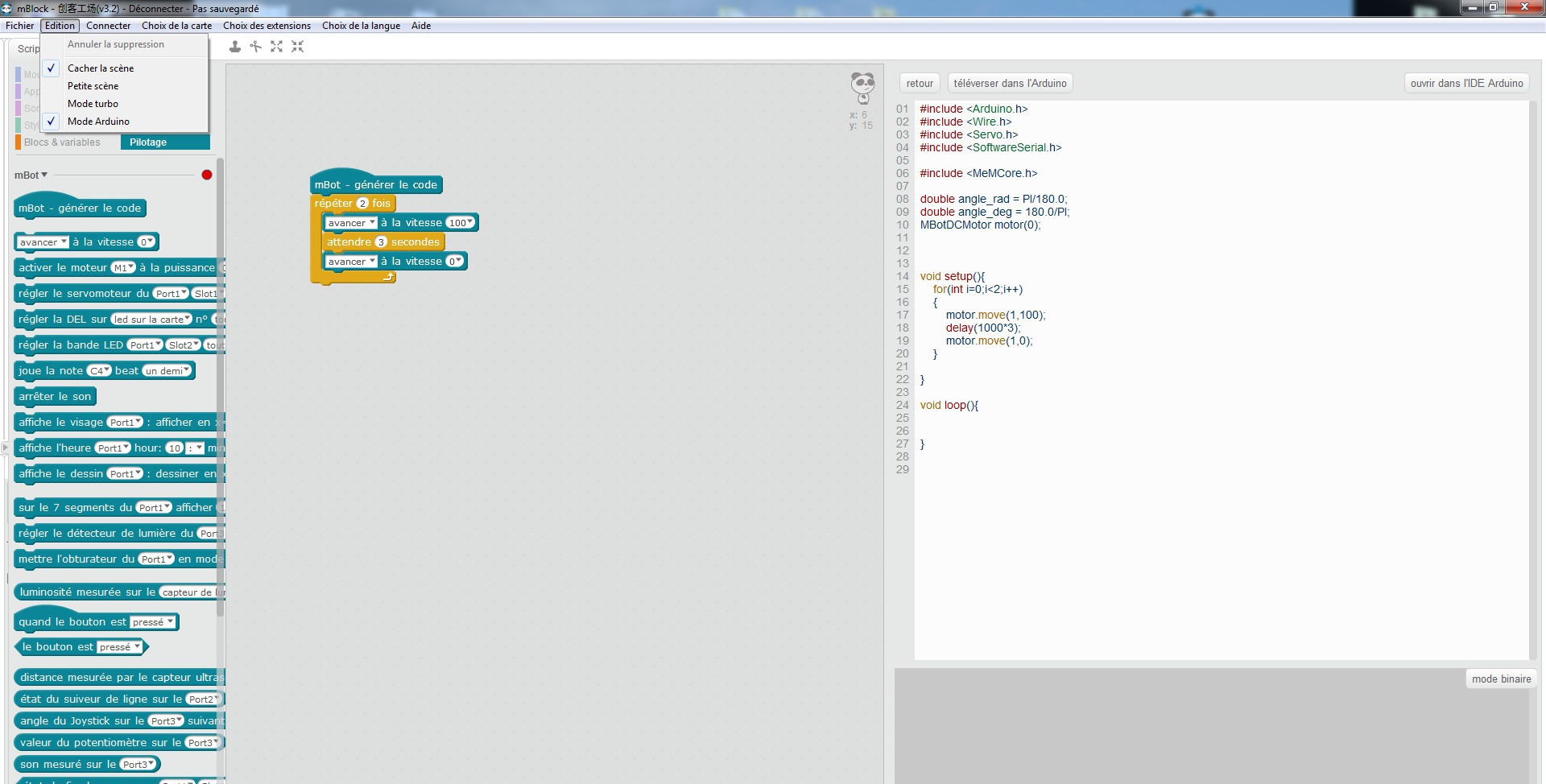
#### « téléverser dans l'arduino».

1. Un message s'affichera pour vous prévenir que le téléversement est terminé.

**!**

**ATTENTION :** une fois le téléversement terminé, le robot exécute le programme ! Tenez le en l'air pour qu'il ne saute pas de la table !

Édition / mode arduino



Téléverser dans l'arduino

7

**LES PROGRAMMES A RÉALISER 1 / 3**

**Programme n°1 : l'objectif est de manipuler tous les actionneurs**

Il ne s'exécutera qu'une seule fois.

Le robot avance durant 3 secondes à la vitesse 100.

Puis il s'arrête ( on peut activer chaque moteur à la vitesse 0 ).

Les Led s'allument en vert pendant 2 secondes puis s'éteignent. Le robot émet alors le son G7 durée « tout ».

#### UNE FOIS LE PROGRAMME CRÉÉ, PENSEZ BIEN A L'ENREGISTRER !

**Programme n°2 : faire fonctionner les 2 moteurs indépendamment**

Il ne s'exécutera qu'une seule fois.

Le robot se déplacera avec un moteur à la vitesse 150 et l'autre à la vitesse 50, le tout durant 4 secondes. Ensuite les moteurs s'arrêtent.

Normalement, le robot fait un rond.

**Programme n°3 : Utiliser une boucle, une condition et le module à ultrasons de détection d'obstacles**



Le but est que le robot se déplace en ligne droite avec ses lumières vertes

allumées. Il doit s'arrêter et s'allumer en rouge quand il arrive à 30 cm d'un obstacle. Si on supprime l'obstacle, il repart.

C'est une **boucle infinie** donc tout commencera par:

**Le principe :**

Jusqu'à ce qu'un obstacle soit à moins de 30 cm, le robot avance à la vitesse 100 avec ses Leds allumées en vert ( on mettra une petite temporisation de 0,1s à la fin, c'est -à-dire « attendre 0,1s » ).

**?**

Quand il détecte un obstacle à moins de 30 cm, il doit

s'arrêter et allumer ses Leds en rouge puis attendre une seconde.

**Astuce 1 :** il va falloir aller chercher le comparateur **<** dans « opérateurs » et y coller la distance mesurée que vous trouverez dans « pilotage ».

**Astuce 2 :** il faudra utiliser « **répéter jusqu'à** » que vous trouverez dans l'onglet « contrôle.





ici, on indiquera la valeur de la distance limite en cm.

8

**LES PROGRAMMES A RÉALISER 2 / 3**

## Programme n°4 : l'objectif est de faire fonctionner le robot à la télécommande avec les flèches

Il faut que le robot avance à 100 avec la flèche du haut, recule à 100 avec la flèche du bas, tourne à droite avec la flèche de droite et à gauche avec la flèche de gauche. (pour tourner, on mettra un moteur à 100 et l'autre à

-100 ). Si rien n'est pressé, le robot est à l'arrêt.

Remarque : l'instruction concernant les boutons de la télécommande se trouve parmi les dernières instructions de la bibliothèque « pilotage ».

**Astuce 1** : utiliser des conditions si / alors / sinon **imbriquées.**

## Programme n°5 : Utiliser le capteur de luminosité

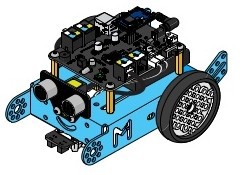
Il s'exécutera en continu.

Le robot allume ses 2 Leds en vert 255 si la luminosité est inférieure à 250 ( on mettra une temporisation de 0,2 s ) , sinon elles sont éteintes.

**Astuce** : on utilisera le même comparateur **<** que dans le programme 3 mais cette fois-ci on ira chercher la valeur de la luminosité mesurée.

## Programme n°6 : Robot suiveur de ligne

On utilisera la piste fournie avec le robot pour mettre en œuvre ce programme. Il s'agit d'un circuit noir sur fond blanc.



En fonction de ce que voit le robot, il renvoie une valeur qui peut aller de 0 à 3.

La Led de droite renvoie la valeur 1 si elle quitte la piste noire ou 0 si elle est au- dessus du noir.

La Led de gauche renvoie la valeur 2 si elle quitte la piste noire ou 0 si elle est au- dessus du noir.

Il existera donc 4 situations possibles :

* le robot est sur la piste.
* le robot est un peu trop à gauche.
* le robot est un peu trop à droite.
* le robot n'est plus du tout sur la piste.

Réalisez un programme qui permet au robot de faire des tours de piste en suivant la ligne noire, aussi vite que possible.

Astuce : il faudra utiliser des « si » imbriqués.

9

**LES PROGRAMMES A RÉALISER 3 / 3**

## Programme n°7 :

## Le robot doit suivre une ligne et allumer ses phares lorsqu’il passe sous un tunnel.

## Programme n°8 :

Le robot doit suivre une ligne, s’il rencontre un obstacle, il l’évite et reprend sa route

10