

*Journée académique Maths/sciences*

**Algorithmique et programmation à l’aide de « mBot »**

Logiciel associé : mBlock (Scratch)

**Sommaire**

Page 2 : Introduction

Page 3 : Connection via clé Wifi

Page 4 : Premiers pas dans la programmation avec mBlock (Scratch)

Page 5 : Exemple1 (mouvement uniforme/ proportionnalité)

Page 6-7 : Exemple 2 (mouvement accéléré/ralenti/proportionnalité/suite géométrique)

Page 8 : Capteur ultrason

Page 9 : Capteur de luminosité

**Introduction**

Ce document offre aux enseignants de mathématiques et de sciences physiques et chimiques la possibilité de découvrir un outil pédagogique (robot mBot) afin de mobiliser une partie du nouvel enseignement d’algorithmique et de programmation.

**mBlock** est un logiciel de programmation graphique par bloc libre et gratuit. Il constitue une extension robotique de Scratch pour piloter les robots mBot-s. L’ensemble des commandes pour piloter le robot sont réunies dans la rubrique « pilotage ». Ce logiciel est utilisable sous Windows, Mac et Linux.

Le robot **mBot** peut être connecté à un terminal (ordinateur, tablette ou smartphone) de trois façons différentes :

|  |  |  |
| --- | --- | --- |
| Via | Avantages | Inconvénients |
| Cable USB | * Téléverser rapidement un programme dans le circuit du robot. | * Pour faire déplacer le robot, on est limité par la longueur du fil USB. |
| Bluetooth en utilisant le module Bluetooth | * Utiliser une tablette ou un téléphone. | * Interférences des signaux entre robots dans un contexte de classe. * Perte du signal au-delà d’environ 5 mètres. * On ne peut pas utiliser un ordinateur sans « Bluetooth ». |
| Wifi (module Wifi et clé Wifi fournis avec le robot) | * Pas d’interférence. * La portée du Wifi est importante (testé jusqu’à 10 mètres.) | …. |

**En classe, on privilégie la connectique « Wifi ». Il n’est peut-être pas nécessaire d’acheter le module Bluetooth si le robot est équipé d’un module Wifi.**

**Connection via la clé Wifi**

Pour tester ce robot en utilisant la clé Wifi, commencer par :

**Etape 1** : allumer le robot (bouton ON/ OFF) situé sur le côté latéral gauche du robot ;

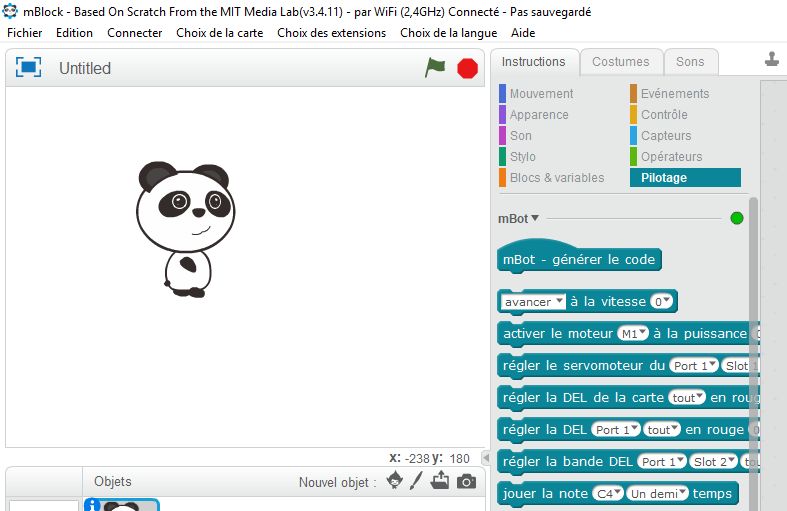
**Etape 2** : à l’aide d’un tournevis appuyer sur le bouton du module Wifi ;

Bouton du Module Wifi

Bouton de mise en marche (on/off)

**Etape 3 :** connecter la clé Wifi à l’ordinateur si la LED bleue du robot s’allume en continu, cela veut dire que celui-ci est connecté à la clé ;

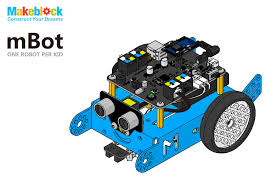
**Etape 4 :** Lancer le logiciel **mBlock** puis connecter celui-ci au robot comme l’indique le schéma ci-dessous.



Onglet « Connecter » pour connecter le robot en choisissant Wifi

Lorsque la connexion est réussie la « pastille » devient verte.

L’indication « **connecté** » apparait en haut de la page.

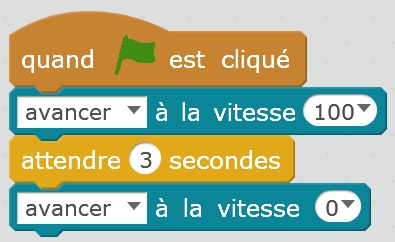


**Vos premiers pas dans la programmation avec mBlock**

Si vous souhaitez piloter le robot depuis l'ordinateur. Dans ce cas, il démarrera lorsque vous appuierez sur le drapeau vert situé sur l'écran d'accueil.

Pour réaliser un programme, il suffit de **glisser les instructions** des éléments de la bibliothèque **dans la zone de création** du programme.

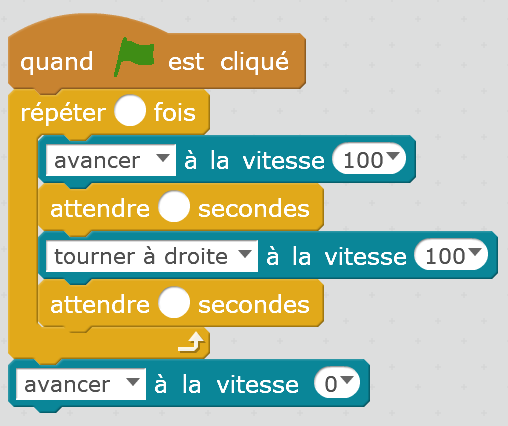
**EXEMPLE D'UN PROGRAMME SIMPLE :**



Quand l'icône départ est appuyée, le robot doit se déplacer en avant à la vitesse 100 pendant 3 secondes puis s'arrêter.

Attention : l’instruction « stop tout » arrête les programmes, mais pas les actionneurs du robot.

Pour l’arrêter il faut utiliser l’instruction « avancer à la vitesse 0 ».



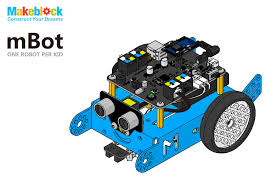
**A VOUS MAINTENANT :**

Quand l'icône départ est appuyée, le robot doit se déplacer en avant à la vitesse 100 pendant 3 secondes puis tourne à droite à la vitesse 100 pendant 0.9 seconde. Le programme se répète 4 fois puis s’arrête.

**Compléter et réaliser le programme ci-contre.**

**Remarque :** Les actionneurs ne s’arrêtent pas tout seul. Il faut toujours leur donner l’ordre de s’arrêter.

Dans le premier exemple si on ne met pas en place l’instruction « **stop tout** » ou « **avancer à la vitesse de 0** ». Le robot continuera à avancer de manière continue.



**Exemples de programmes à réaliser pour manipuler les actionneurs et les capteurs**

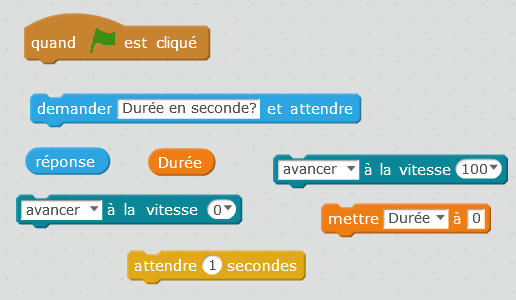
**Programme 1** (mouvement uniforme, situation linéaire)

**But**

Réaliser un programme qui permet au robot de se déplacer à une vitesse constante pendant une durée que l’utilisateur entre au clavier.

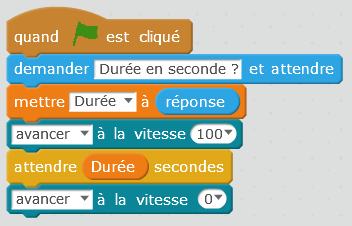
**Proposition :**

Ci-contre, des éléments nécessaires et à imbriquer correctement.



**Remarques**:

* Durée est une variable ici (à créer).
* Le robot n’avance qu’à partir d’une vitesse supérieure strictement à 50.



**Corrigé :**

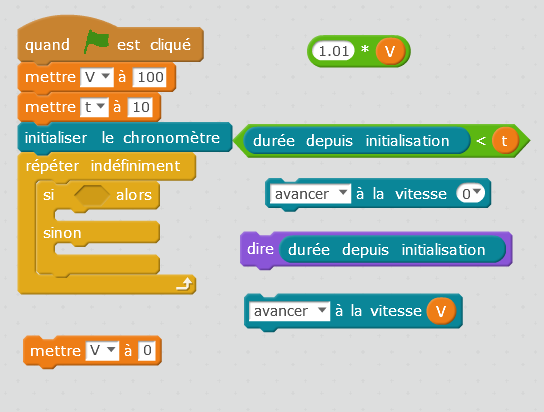
C’est un programme qui permet d’étudier la distance parcourue par le robot en fonction de la durée, en seconde.

**Programme 2 :** Mouvement accéléré pendant une durée donnée, situation non-linéaire, suite géométrique

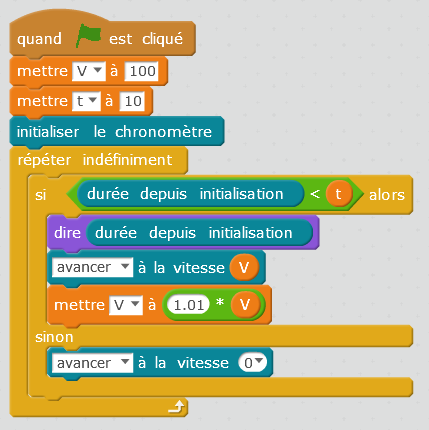
**But**

Réaliser un programme qui permet de déplacer le robot de plus en plus vite pendant une durée qu’on précisera avant de lancer le programme.

**Proposition**

* Mettre par exemple la vitesse initiale à *V = 100 mm/s* ;
* mettre par exemple la durée initiale *t = 10 s* ;
* initialiser le chronomètre ;
* répéter indéfiniment les instructions suivantes :
* si la durée depuis l’initialisation est en dessous de la durée *t =10s*
* avancer à la vitesse *V*;
* augmenter *V* de *1%* après chaque boucle *;*
* dire la durée depuis l’initialisation ;
* sinon, avancer à la vitesse 0.

**Remarque** : **V** et **t** sont des variables ici (à créer).

**Corrigé 1 :**

**Dans ce cas, la vitesse augmente de manière exponentielle.**

**Idée de TP** : étudier la distance parcourue en fonction de la durée.

On s’aperçoit que la distance n’est pas proportionnelle à la durée et que la vitesse moyenne augmente.

**Corrigé 2**



**Ce deuxième programme nous permet d’enregistrer la durée et la vitesse au moment où les moteurs changent de vitesse dans deux listes distinctes.**

Les données peuvent être exportées vers un tableur (Excel, Geogebra …) pour les exploiter.

Remarque :

On peut créer les listes à l’aide de la bibliothèque « **Blocs et Variables** »

**Programme 3 : Mouvement ralenti pendant une durée donnée, suite géométrique …**

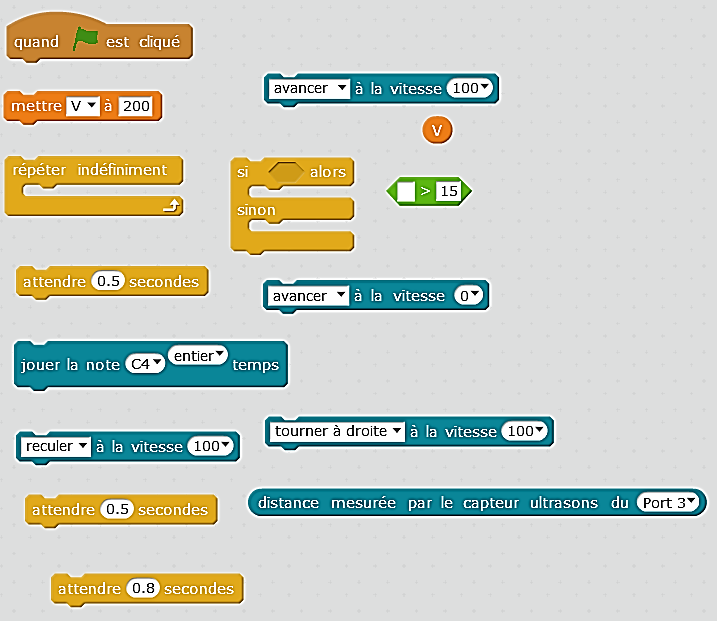
Le robot est animé par un mouvement ralenti si on remplace 1.01 par 0.99 par exemple.

Il est judicieux dans ce cas de mettre la vitesse initiale à 255 (vitesse maximale du robot).

Et si on remplace « avancer » dans le programme précédent par « tourner à droite » ou « tourner à gauche ». En jouant sur le taux d’accroissement de la vitesse, le robot réalise un mouvement circulaire uniforme (1), accéléré (1.01) ou ralenti (0.99) sur lui-même.

**Programme 4 : capteur ultrason**

Le robot avance. S'il détecte un obstacle devant lui, il recule, tourne à droite et reprend son mouvement.



**Proposition**

* Régler la vitesse à 200 (utiliser une variable)
* Réaliser une boucle infinie
* Si la distance mesurée est > 15

Avancer à la vitesse V=200.

* Sinon :

Arrêter pendant 0.5s

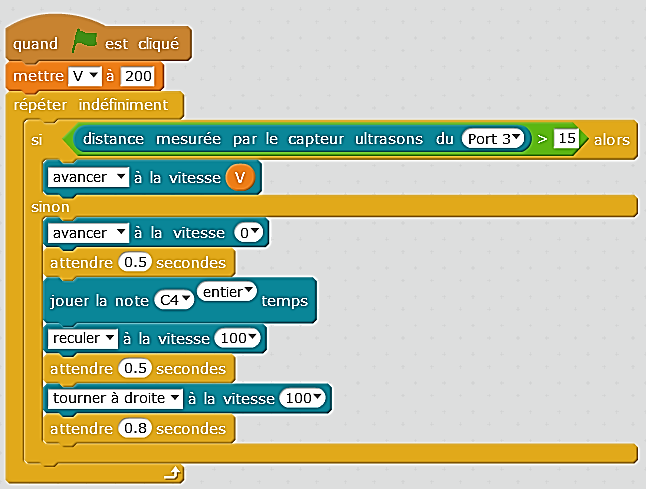
Jouer la note C4

Reculer à la vitesse 100 pendant 0.5s.

Tourner à droite pendant 0.8s à la vitesse 100.

**« Programme éclaté »**

**Corrigé**



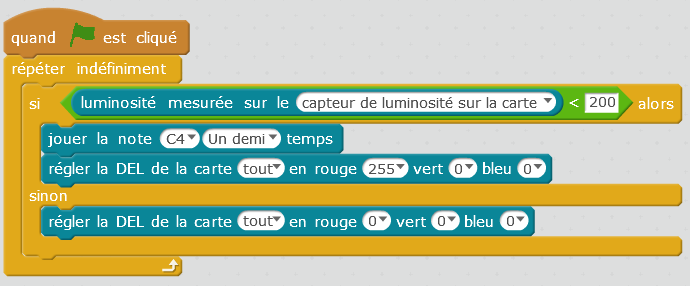
**Programme n°5** : Capteur de luminosité

Utiliser le capteur de luminosité pour simuler une alarme.

**Proposition**

* Réaliser une boucle infinie
* Utiliser la condition **si alors/sinon**
* Une condition **Si** (il se passe ceci) **alors**
* Ici l'événement déclencheur est la baisse de la lumière mesurée sur la carte en dessous de 200 lux
* Jouer la note de l'alarme sur le buzzer et allumer les DEL-s de la carte en rouge
* **Sinon** éteindre les DEL-s.

**Remarque** : régler d’abord la DEL de la carte sur R=0%, V=0% et B=0%

**Corrigé**

On peut également enregistrer les mesures dans une liste à l’aide du programme ci-dessous.

