

Robot mBot – Niveau 1, séance 1

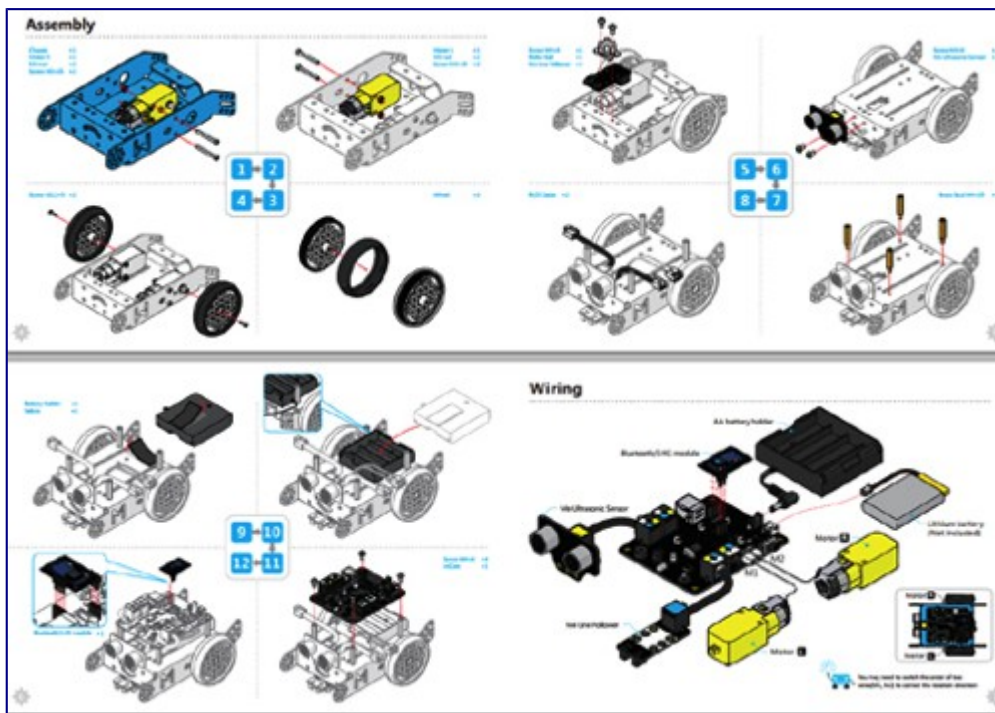
Ce module se veut ludique, utilise le robot mBot et permet de compléter l'apprentissage de scratch et principalement l'utilisation de capteurs d'environnement.

Découverte du robot mBot

Le robot

Le robot « mBot » est conçu sur la base d'une carte arduino UNO-328 (base ATmega328). Cette carte est associée à un contrôleur de moteur spécialisé qui va nous permettre de commander les 2 moteurs du robot.

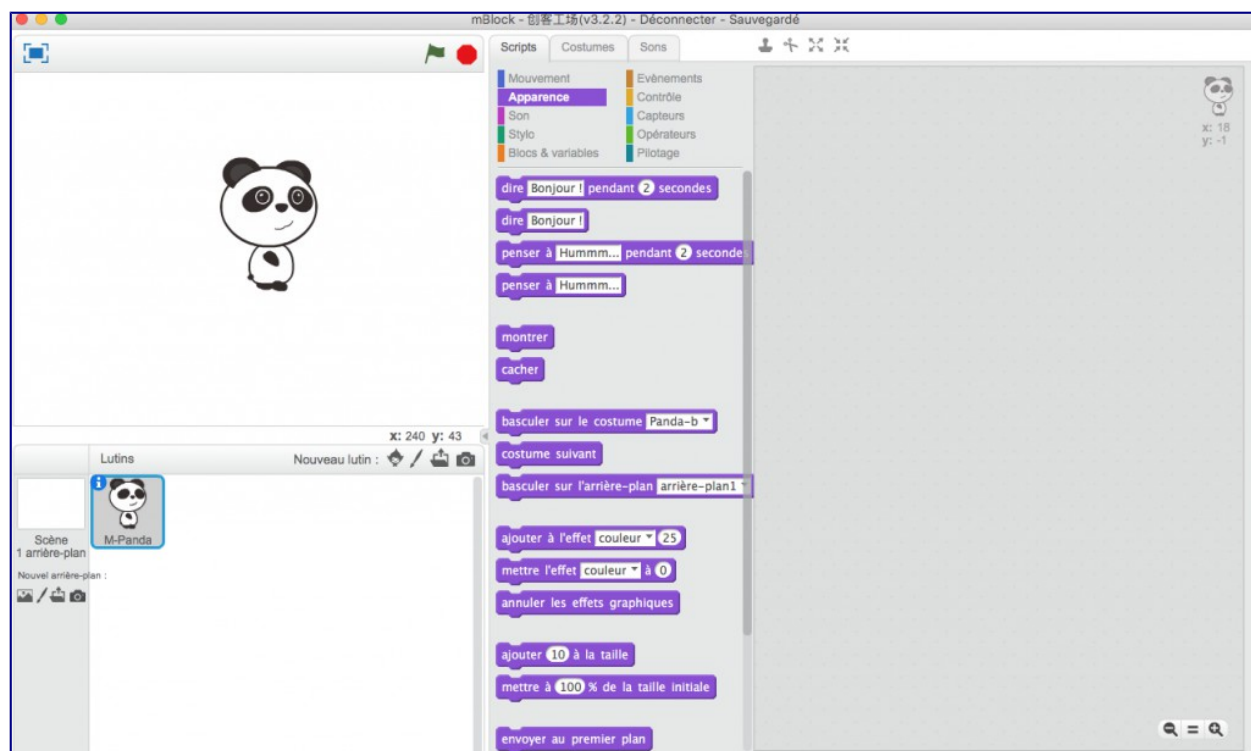
Schéma de montage:



L'environnement de développement

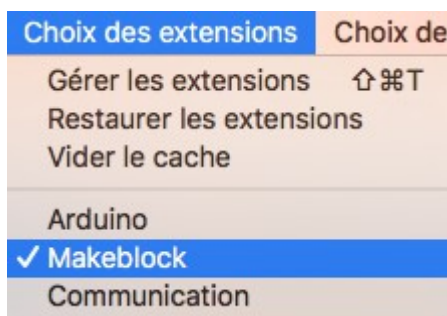
Dans ce second module les élèves utiliseront le langage scratch pour piloter le robot. Cependant une **version spéciale de scratch appelée mblock est nécessaire** pour programmer les déplacements, accéder aux capteurs spécifiques du robot, envoyer les commandes vers le robot lors de l'exécution du programme.

L'environnement mblock



Pour les utilisateur de scratch pas d'inquiétude, les concepts restent les mêmes, et l'environnement parait tout de suite familier.

Tout d'abord nous allons dire à l'environnement que nous allons programmer un robot mBot. Pour cela nous allons aller dans le menu, choix de la carte et sélectionner mBot. Et choix des extensions makeblock



Identification des commandes du robot:

L'ensemble des instructions de programmation du robot est situé dans l'onglet de commande pilotage.

Exercice 1:

Le premier exercice va nous permettre de se familiariser avec la création et le lancement du programme sur le robot.

But:

- Apprendre à écrire un programme pour le robot
- Apprendre à télécharger le programme dans le robot
- Faire avancer le robot à la vitesse 50 lorsque je la touche **A** de la télécommande est pressée.
- Stopper le robot lorsque la touche **C** de la télécommande est pressée.
- Faire reculer le robot à la vitesse 25 lorsque la touche **flèche vers le bas** est pressée.

Ecrire le programme:

Les instructions permettant de commander le robot se trouvent dans le **menu Pilotage**.

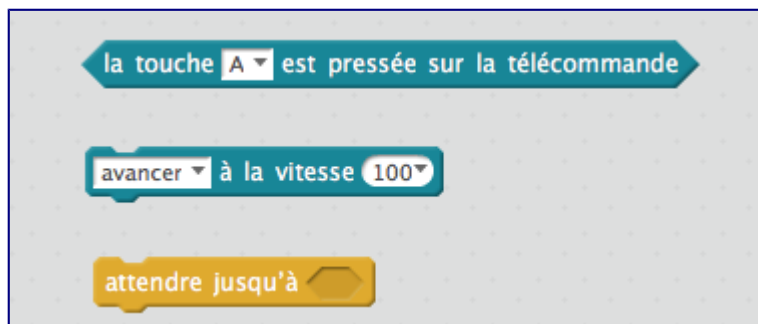
Les autres menus (Blocs & variables, Contrôle, Opérateurs), permettent de réaliser la logique du programme.

Tous les programmes que l'on voudra télécharger dans le robot commencent par



1. Rechercher dans la liste des commandes du menu pilotage la commande qui permet de détecter l'appui sur une touche de la télécommande
2. Rechercher dans la liste des commandes du menu pilotage la commande qui permet de faire avancer le robot
3. Comment faire pour que le robot ne démarre pas tout de suite?
4. Comment allons nous faire pour arrêter le robot?

Les commandes:



L'exercice se traduit par:

Quand la touche **A** de la télécommande est **pressée**

Alors

Avancer à la vitesse 50

Quand la touche C de la télécommande est pressée

Alors

Avancer à la vitesse 0

Quand la touche flèche bas est pressée

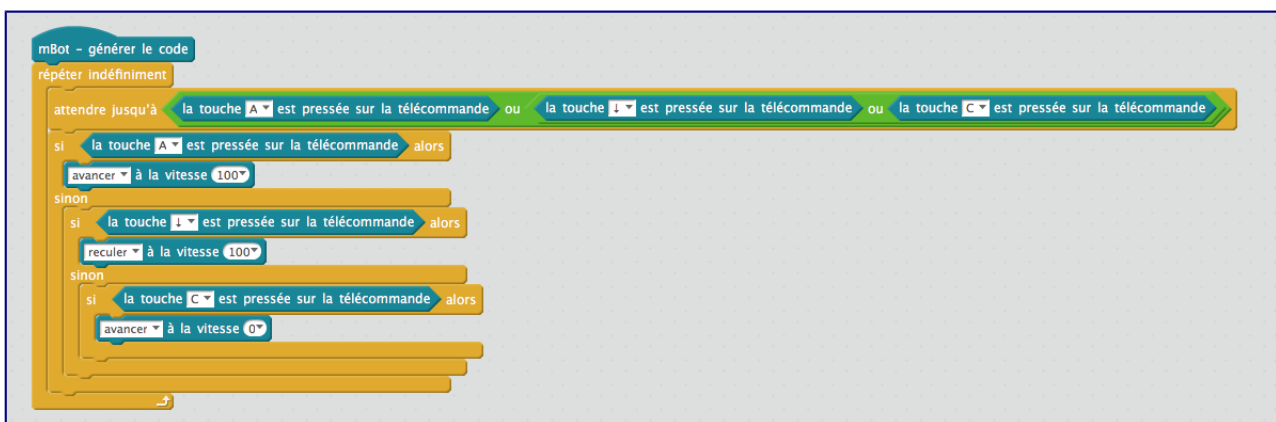
Alors

Reculer à la vitesse 50

A vous de jouer:

Les instructions scratch correspondantes à écrire dans l'onglet script (programme) sont:

Un programme avec toutes les instructions en ligne:



ou plus judicieusement, puisque l'environnement le permet, le programme avec traitement en parallèle:

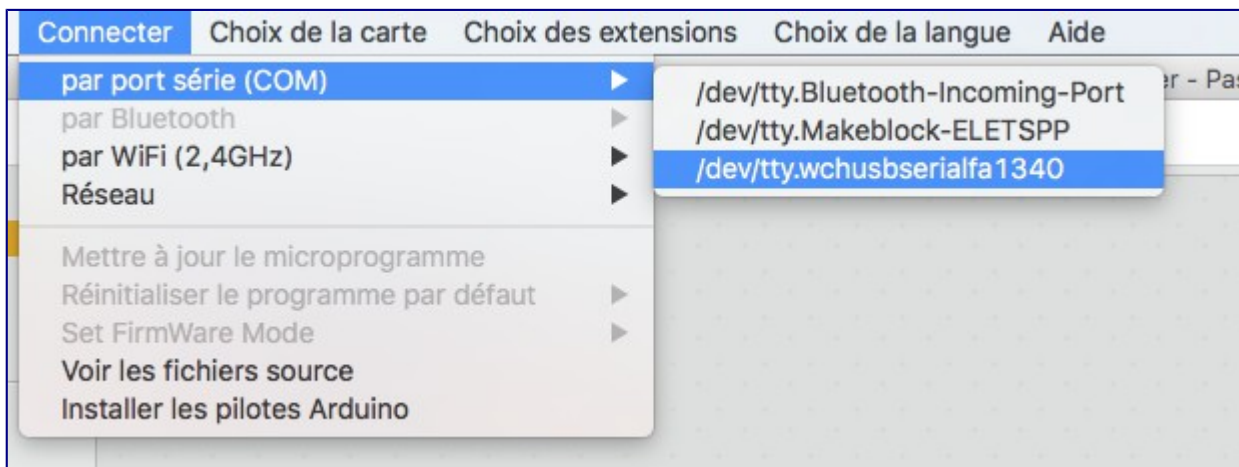


Pour tester le programme nous allons devoir connecter le robot avec l'ordinateur. Cette connexion se fait par le cable USB.

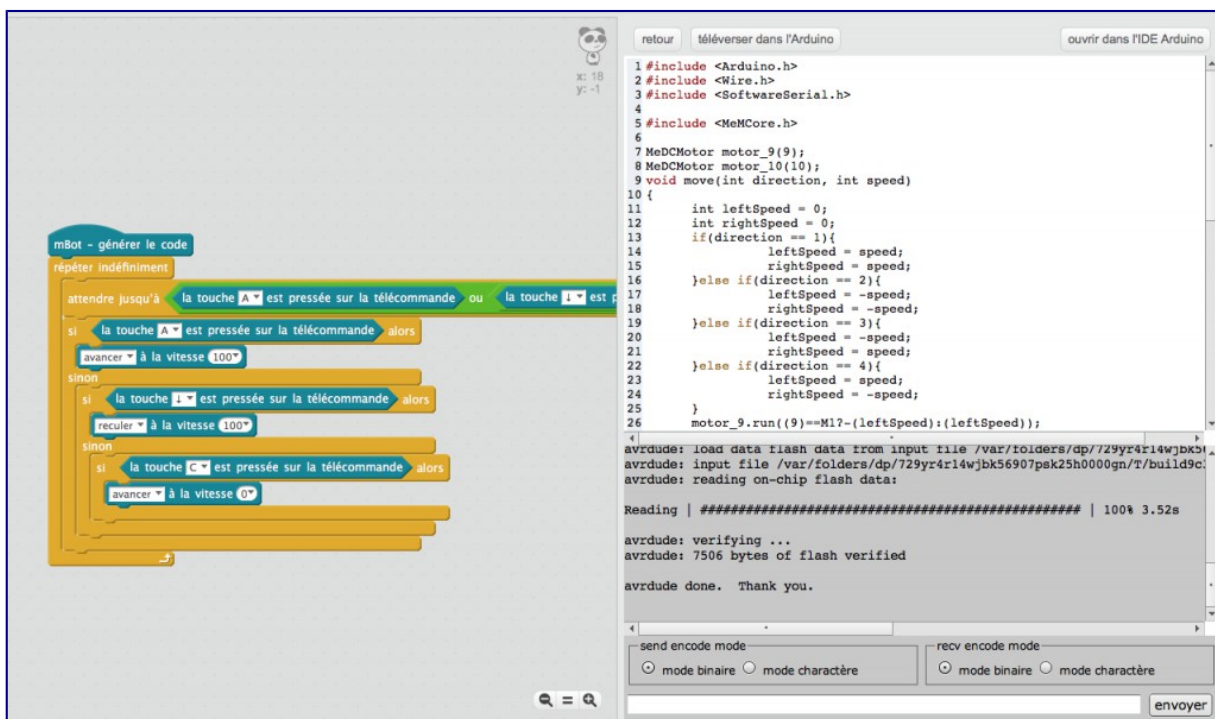
Le programme va être téléchargé dans le robot, il va se lancer automatiquement.

Marche à suivre:

- Allumer le robot (interrupteur on/off, situé au dessus de la roue arrière gauche)
- Connecter le robot et l'ordinateur (Connecter-> par port Serie COM -> /dev:



- Passer en mode arduino (Edition-> Mode Arduino), la fenêtre programme s'ouvre.



- Sélectionner téléverser dans l'Arduino
- Attendez que la fenêtre Téléversement fini



Pour aller plus loin:

Modifier le programme pour que le robot émette un son (Bip Bip) quand il recule comme les véhicules de chantier.

L'exercice se traduit par:

En *italique* ce qui doit être ajouté au programme précédent:

Quand la touche a de la télécommande est pressée

Alors

Avancer à la vitesse 50

Quand la touche c de la télécommande est pressée

Alors

Avancer à la vitesse 0

Quand la touche flèche bas est pressée

Alors

Reculer à la vitesse 50

Mettre variable RobotRecul à 1

Répéter tant que variable RobotRecul = 1

Emettre son (C5 beat 1 qart)

Attendre 0.2 seconde

Fin tant que

Quand la touche C de la télécommande est pressée

Alors

Reculer à la vitesse 0

Mettre variable RobotRecul à 0

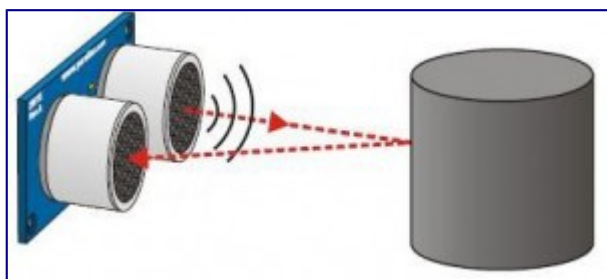
Exercice supplémentaire : Modifier le programme pour que le robot tourne à droite (en avançant) lorsque l'on appuie sur la flèche droite et à gauche lorsque l'on appuie sur la flèche gauche de la télécommande.

Robot mBot – Niveau 1, séances 2 et 3 – Capteur de proximité

Fonctionnement du capteur de proximité

Le capteur que nous allons utiliser est un capteur de distance à ultrason. Il fonctionne avec une partie émetteur d'onde et une partie récepteur d'onde.

Cette onde va se déplacer dans l'air (comme le son) et être renvoyé par un obstacle. Comme on connaît la vitesse du son, il est possible de déterminer la distance de l'obstacle en mesurant le temps écoulé entre l'envoi de l'ultrason et la réception de l'écho.



Scratch va effectuer pour nous ce calcul de distance en fonction des informations qu'il va recevoir du capteur.

Exercice:

L'exercice consiste à afficher sur l'écran de l'ordinateur la distance que mesure le capteur. Pour vérifier le bon fonctionnement du capteur, approcher et reculer votre main. Que constatez vous?

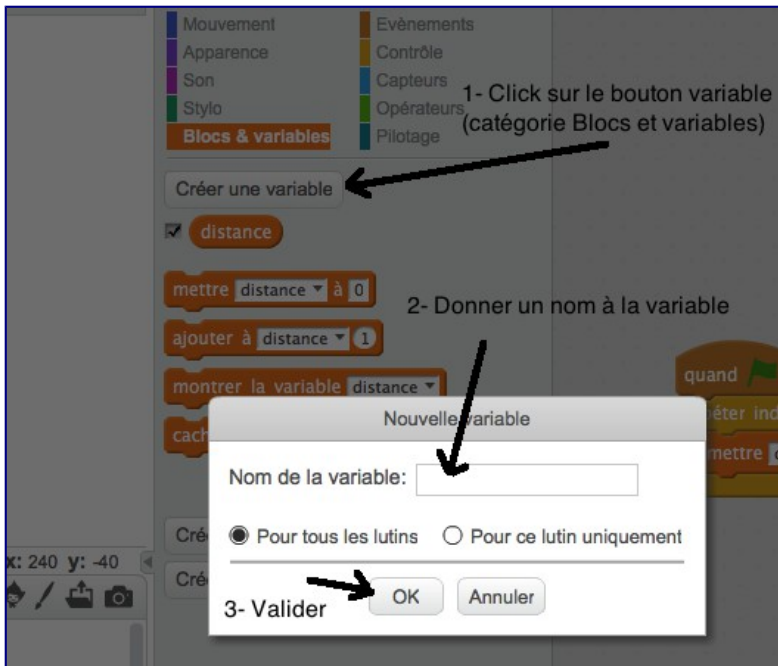
Aide:

Qu'est ce qu'une variable en informatique:

- L'ordinateur possède de la mémoire, c'est un peu comme la boîte à outils de papa, il y a toutes sortes de petites cases où l'on peut ranger des choses. Papa pour se retrouver à coller des petites étiquettes sur toutes ces cases avec le nom de ce qu'il y a dedans.
- Pour nous la variable est une petite case de l'ordinateur où l'on peut ranger des choses (des données), par exemple l'âge d'une personne, ou son prénom. Pour nous faciliter la tâche pour retrouver ce qu'il y a dans cette variable, on prendra l'habitude de lui donner un nom qui représente ce que l'on met dedans.

Exemple:

Pour créer la variable qui représente le nom de notre personnage sous scratch on procède de la manière suivante:



Nom de notre variable: Prénom du personnage.

L'exercice pour afficher les mesures du capteur se traduit par:

Créer la variable Distance

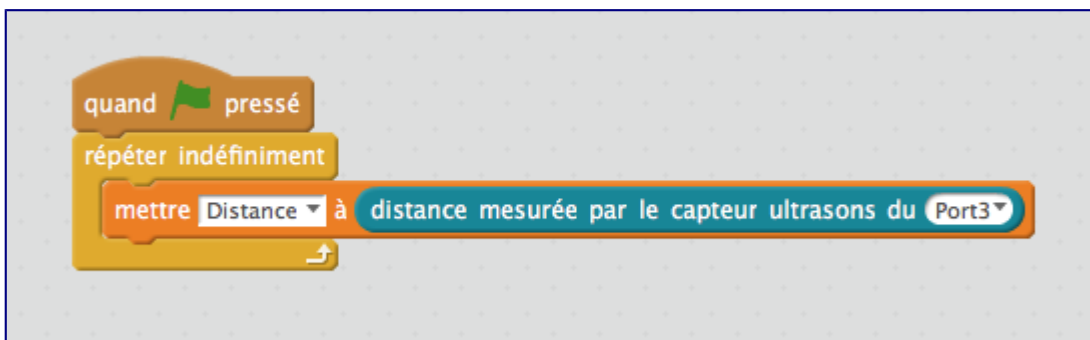
Quand la touche **drapeau** est **pressée**

Répéter indéfiniment

Mettre la variable **Distance** à distance mesurée par le capteur ultrason

Fin Répéter indéfiniment

Les instructions Mblock correspondantes, à écrire dans l'onglet script (programme) sont:



Pour tester ce programme nous allons utiliser la connexion USB entre le robot et l'ordinateur.

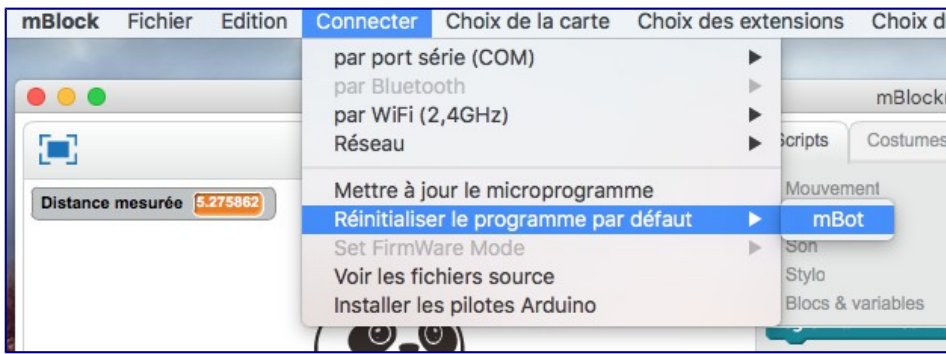
1. Connecter le cordon USB au robot et à l'ordinateur.
2. Allumer le robot
3. Etablir la connexion informatique entre l'ordinateur et le robot, dans le menu Connecter → par port série com

Lorsque le robot est connecté il émet des petits Bips.

Lancer alors le programme, que remarquez vous?



Si le programme ne démarre pas, recharger le programme par défaut du robot qui permet d'exécuter le programme à travers la connexion USB



Pour aller plus loin:

- Lorsque la distance entre le capteur et votre main est inférieure à 30 cm, faire émettre par le robot des bips et si la distance est inférieure à 15 cm émettre un bip continu. Voilà comment fonctionne les détecteurs de distance et aide au stationnement de la voiture de vos parents !

L'exercice se traduit par:

Créer la variable Distance

Quand la touche **drapeau** est **pressée**

Répéter indéfiniment

Mettre la variable **Distance** à distance mesurée par le capteur ultrason

Si variable **RobotRecul** < 30 **et** **RobotRecul** > 15

Alors

Émettre son (C5 beat 1 quart)

Attendre 0.2 seconde

Fin Alors

Si variable **RobotRecul** < 15

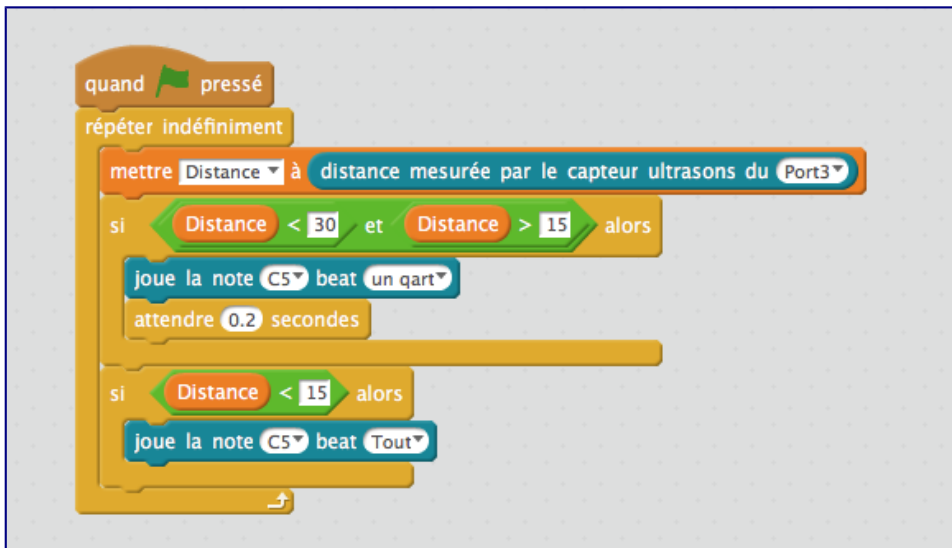
Alors

Émettre son (C5 beat tout)

Fin Alors

Fin Répéter indéfiniment

Les instructions scratch correspondantes, à écrire dans l'onglet script (programme) sont:



Maintenant, coupez cette fonction avec la fonction avancer du robot:

- Lorsque le robot se rapproche à moins de 50 cm de l'obstacle, ralentir sa vitesse à 50.
- Si la distance est inférieure à 30 cm, stopper le robot.

L'exercice se traduit par:

Créer la variable Distance

Créer la variable Vitesse

Répéter indéfiniment

Si la touche **a de la télécommande** est **pressée**

Répéter jusqu'à touche **c de la télécommande** est **pressée**

Mettre la variable **Distance** à distance mesurée par le capteur ultrason

Mettre la variable **Vitesse** à 100

Si variable **Distance** < 50

Alors

Mettre la variable **Vitesse** à 50

Fin Alors

Si variable **Distance** < 50

Alors

Mettre la variable **Vitesse** à 0

Fin Alors

Avancer à la vitesse **Vitesse**

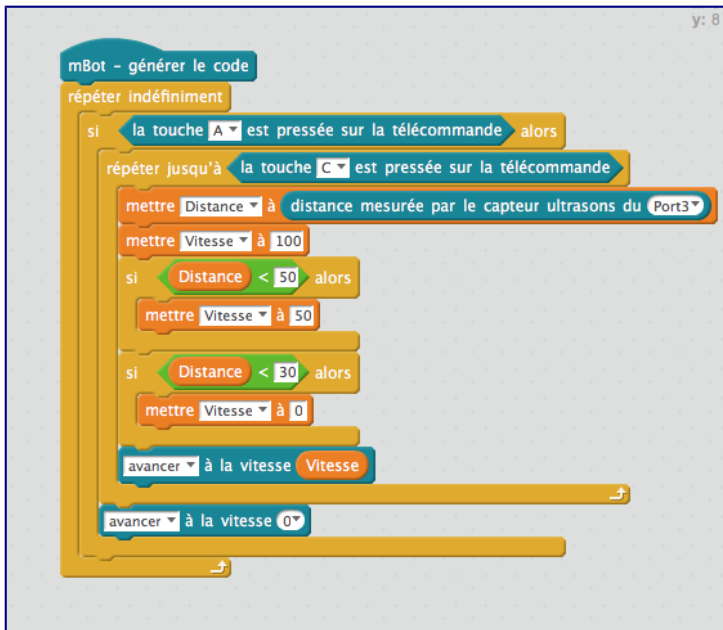
Fin Répéter jusqu'à

Avancer à la vitesse 0

Fin Si

Fin Répéter indéfiniment

Les instructions scratch correspondantes à écrire dans l'onglet script (programme) sont:



Télécharger le programme dans le robot.

Robot mBot – Niveau 1, séances 4 et 5 – Capteur suivi de ligne

Fonctionnement du capteur de suivi de ligne

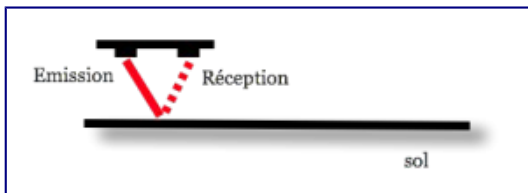
Le capteur que nous allons utiliser est un capteur de contraste. Il est constitué d'une LED émettrice et d'un phototransistor.

Le capteur du robot:



La LED émettrice envoie une lumière infrarouge que le sol réfléchit en direction du phototransistor qui capte ainsi la quantité de lumière en retour.

Schéma de principe:

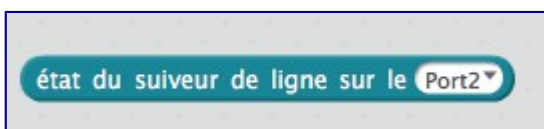


Comme les couleurs foncées réfléchissent moins bien la lumière que les couleurs claires, le capteur peut ainsi définir s'il se trouve au dessus d'une couleur claire (lorsqu'il reçoit beaucoup de lumière), il prend alors la valeur 1, ou s'il se trouve au dessus d'une couleur foncée (lorsqu'il reçoit peu de lumière), il prend alors la valeur 0 (zéro).

Notre robot possède deux capteurs de suivi de ligne à l'avant. Ils sont situés l'un à coté de l'autre, l'un à droite du centre du robot, l'autre à gauche.

Mblock met à disposition du programmeur une instruction qui permet de lire la valeur des deux capteurs en même temps.

Il s'agit de l'instruction « état du suiveur de ligne sur port2 »



- Lorsque les **deux capteurs détectent une couleur claire** la valeur état suiveur est à **3**.
- Lorsque le **capteur de droite détecte une couleur claire** et le **capteur de gauche détecte une couleur foncée** la valeur état suiveur est à **1**.
- Lorsque le **capteur de droite détecte une couleur foncée** et le **capteur de gauche détecte une couleur claire** la valeur état suiveur est à **2**.
- Lorsque les **deux capteurs détectent une couleur foncée** la valeur état suiveur est à **0**.

Exercice:

L'exercice consiste à afficher sur l'écran de l'ordinateur la valeur état du suiveur. Pour vérifier le bon fonctionnement du capteur, approcher du capteur une feuille sur laquelle il y a un gros trait noir. Que constatez vous?

L'exercice se traduit par:

Créer la variable **EtatSuiveur**

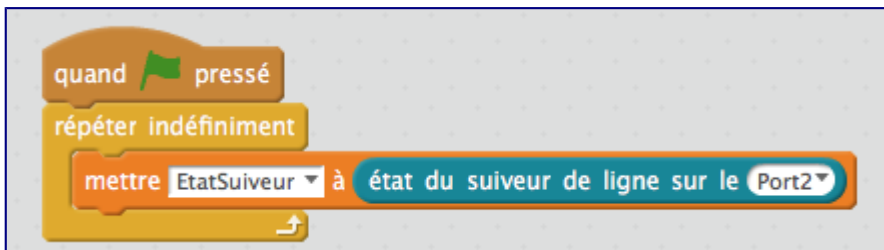
Quand la touche **drapeau** est **pressée**

Répéter indéfiniment

Mettre la variable **EtatSuiveur** à état du suiveur de ligne

Fin Répéter indéfiniment

Les instructions scratch correspondantes, à écrire dans l'onglet script (programme) sont:



Pour tester ce programme nous allons utiliser la connexion USB entre le robot et l'ordinateur.

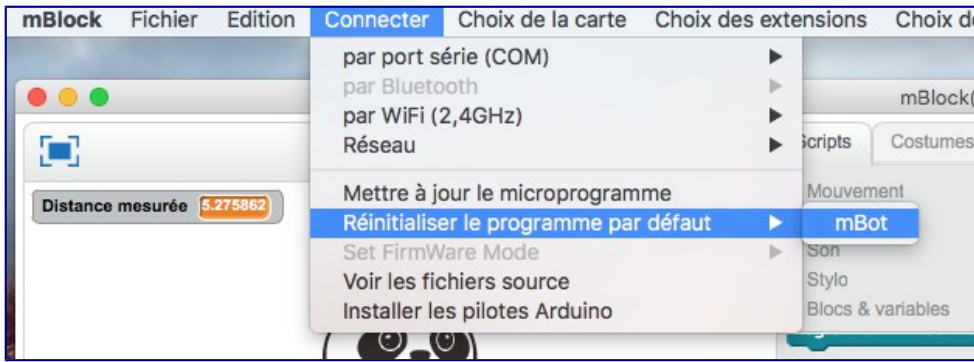
1. Connecter le cordon USB au robot et à l'ordinateur.
2. Allumer le robot
3. Etablir la connexion informatique entre l'ordinateur et le robot, dans le menu Connecter → par port série com

Lorsque le robot est connecté il émet des petits Bips.

Lancer alors le programme, que remarquez vous?



Si le programme ne démarre pas recharger le programme par défaut du robot qui permet d'exécuter le programme à travers la connexion USB



Pour aller plus loin, maintenant, coupez cette fonction avec la fonction avancer du robot:

Ecrire les scripts correspondants aux deux cas suivants:

- Lorsque le robot arrive sur une ligne foncée, il s'arrête
- Puis, lorsque le robot a franchi 3 lignes foncées, il s'arrête

Le cas 1 se traduit par:

Attendre l'appui sur la **touche A** de la **télécommande**

Créer la variable **EtatSuiveur**

Quand la touche **drapeau** est **pressée**

Mettre la variable **EtatSuiveur** à état du suiveur de ligne

Répéter jusqu'à variable **EtatSuiveur** est **égale à 0**

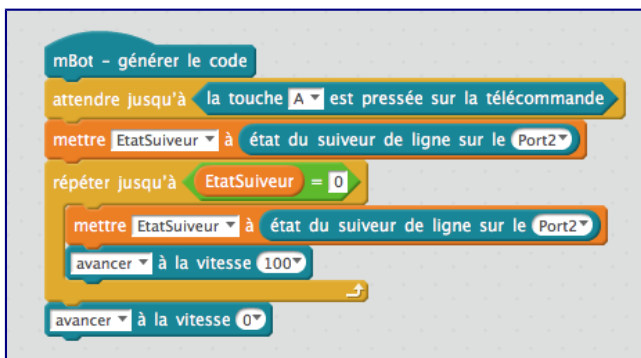
Mettre la variable **EtatSuiveur** à état du suiveur de ligne

Avancer

Fin Répéter tant que

Arrêter le Robot

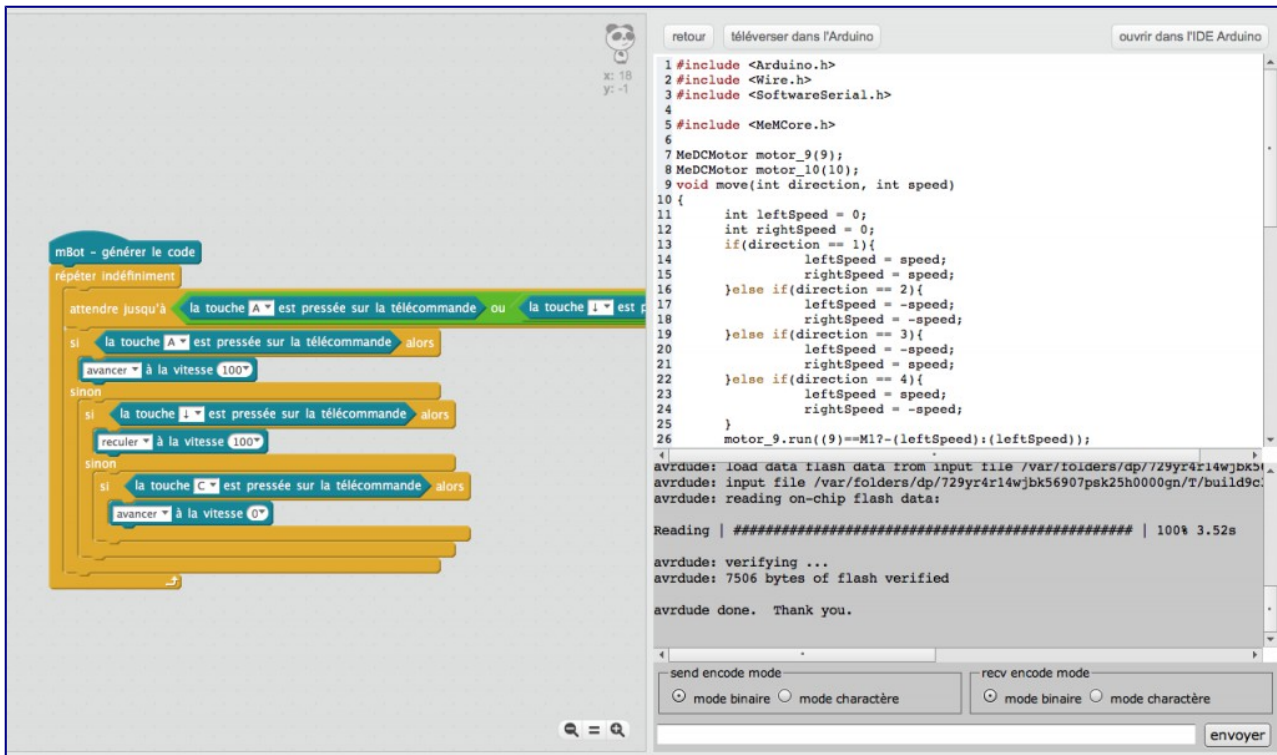
Les instructions Mblock correspondantes, à écrire dans l'onglet script (programme) sont:



Télécharger le script vers le robot:

Marche à suivre:

- Allumer le robot
- Connecter le robot et l'ordinateur (Connecter-> par port Serie COM -> /dev:
- Passer en mode arduino (Edition-> Mode Arduino), la fenêtre programme s'ouvre.



- Sélectionner téléverser dans l'Arduino
- Attendez que la fenêtre Téléversement fini



Le cas 2 se traduit par:

Créer la variable EtatSuiveur

Créer la variable NombreDeLignesFranchies

Quand la touche drapeau est pressée

Mettre la variable NombreDeLignesFranchies à 0

Répéter jusqu'à NombreDeLignesFranchies est égale 3

Mettre la variable EtatSuiveur à état du suiveur de ligne

Avancer

Si variable EtatSuiveur est égale 0

répéter jusqu'à variable EtatSuiveur est égale 3

Mettre la variable EtatSuiveur à état du suiveur de ligne

Fin Répéter jusqu'à

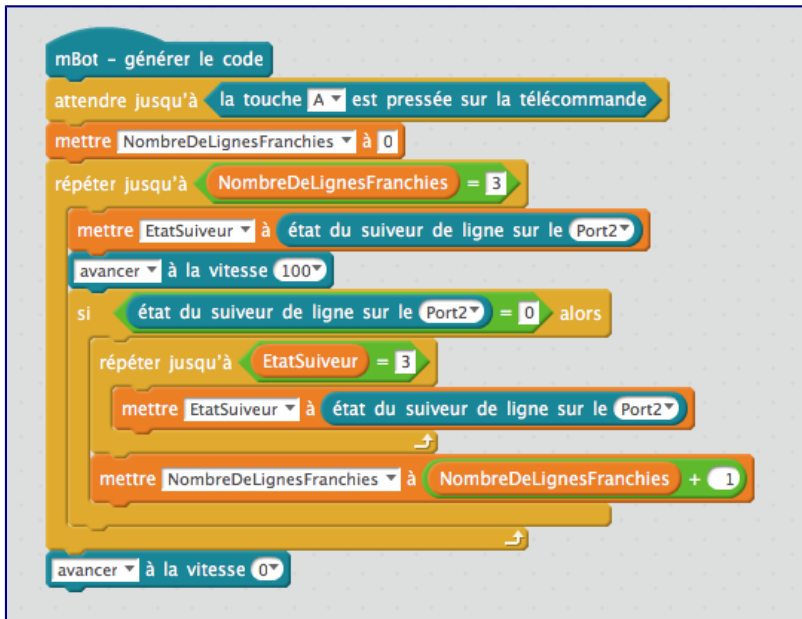
NombreDeLignesFranchies = NombreDeLignesFranchies +1

Fin Si

Fin Répéter jusqu'à

Arrêter le Robot

Les instructions scratch correspondantes, à écrire dans l'onglet script (programme) sont:



Télécharger le script vers le robot (voir la démarche à suivre ci dessus)

Question supplémentaire, comment faire pour relancer ce meme traitement à chaque fois que l'on appuie sur la toute A de la télécommande?

POUR ALLER PLUS LOIN ...

Exercice: Robot prisonnier

L'exercice consiste à déposer le robot sur une surface claire. On délimite cette surface à l'aide d'une ligne de couleur foncée. Le but de l'exercice est de conserver le robot à l'intérieur de la surface délimitée. La vitesse de déplacement du robot est libre.

Créer la variable EtatSuiveur

Quand la touche drapeau est pressée

Avancer

Répéter indéfiniment

Mettre la variable **EtatSuiveur** à état du suiveur de ligne

Si la variable **EtatSuiveur** égale **2**

Alors

Tourner à gauche

Répéter jusqu'à ce que la variable **EtatSuiveur** égale **3**

Mettre la variable **EtatSuiveur** à état du suiveur de ligne

Fin Répéter jusqu'à

Fin Alors

Si la variable **EtatSuiveur** est inférieur à **2**

Alors

Tourner à droite

Répéter jusqu'à ce que la variable **EtatSuiveur** égale **3**

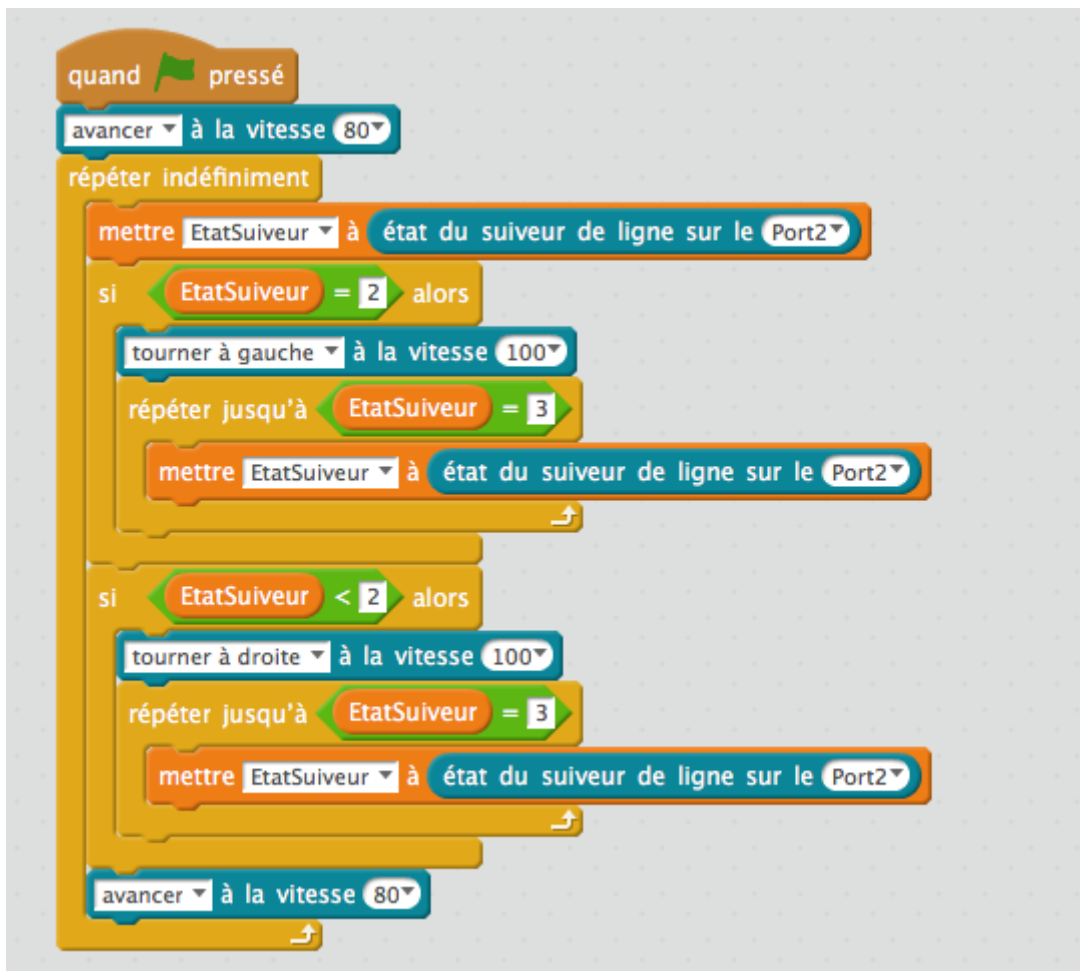
Mettre la variable **EtatSuiveur** à état du suiveur de ligne

Fin Répéter jusqu'à

Fin Alors

Fin Répéter indéfiniment

Les instructions scratch correspondantes, à écrire dans l'onglet script (programme) sont:



Pour aller plus loin, si le robot sort parfois de la zone trouver un moyen de sécuriser les déplacements pour éviter les sorties du robot.

Exercice: 24h du Mans

L'exercice consiste à déposer le robot sur le circuit. La vitesse de déplacement du robot est libre, cependant, le but est de réaliser le meilleur temps sur un tour de circuit ... tout en évitant les sorties de piste.

Créer la variable EtatSuiveur

Créer la variable Vitesse

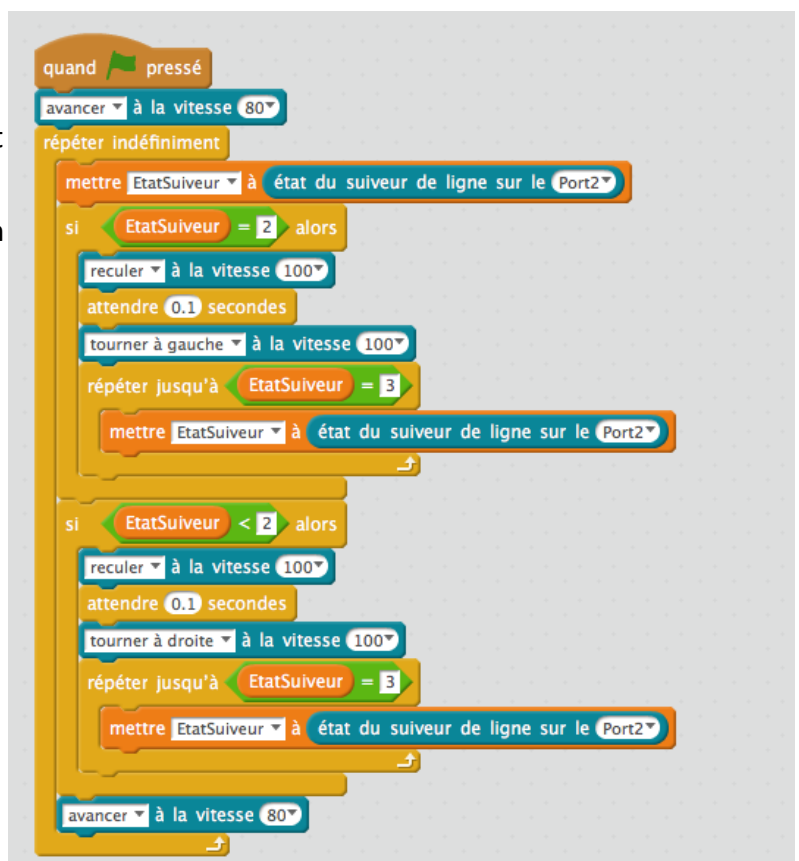
Quand la touche drapeau est pressée

Mettre la variable Vitesse à 255

Avancer à la vitesse Vitesse

Répéter indéfiniment

Mettre la variable EtatSuiveur à état du suiveur de ligne



Si la variable **EtatSuiveur** égale **2**

Alors

Tourner à droite

Répéter jusqu'à ce que la variable **EtatSuiveur** égale **0**

Mettre la variable **EtatSuiveur** à état du suiveur de ligne

Fin Répéter jusqu'à

Fin Alors

Si la variable **EtatSuiveur** égale **1**

Alors

Tourner à gauche

Répéter jusqu'à ce que la variable **EtatSuiveur** égale **0**

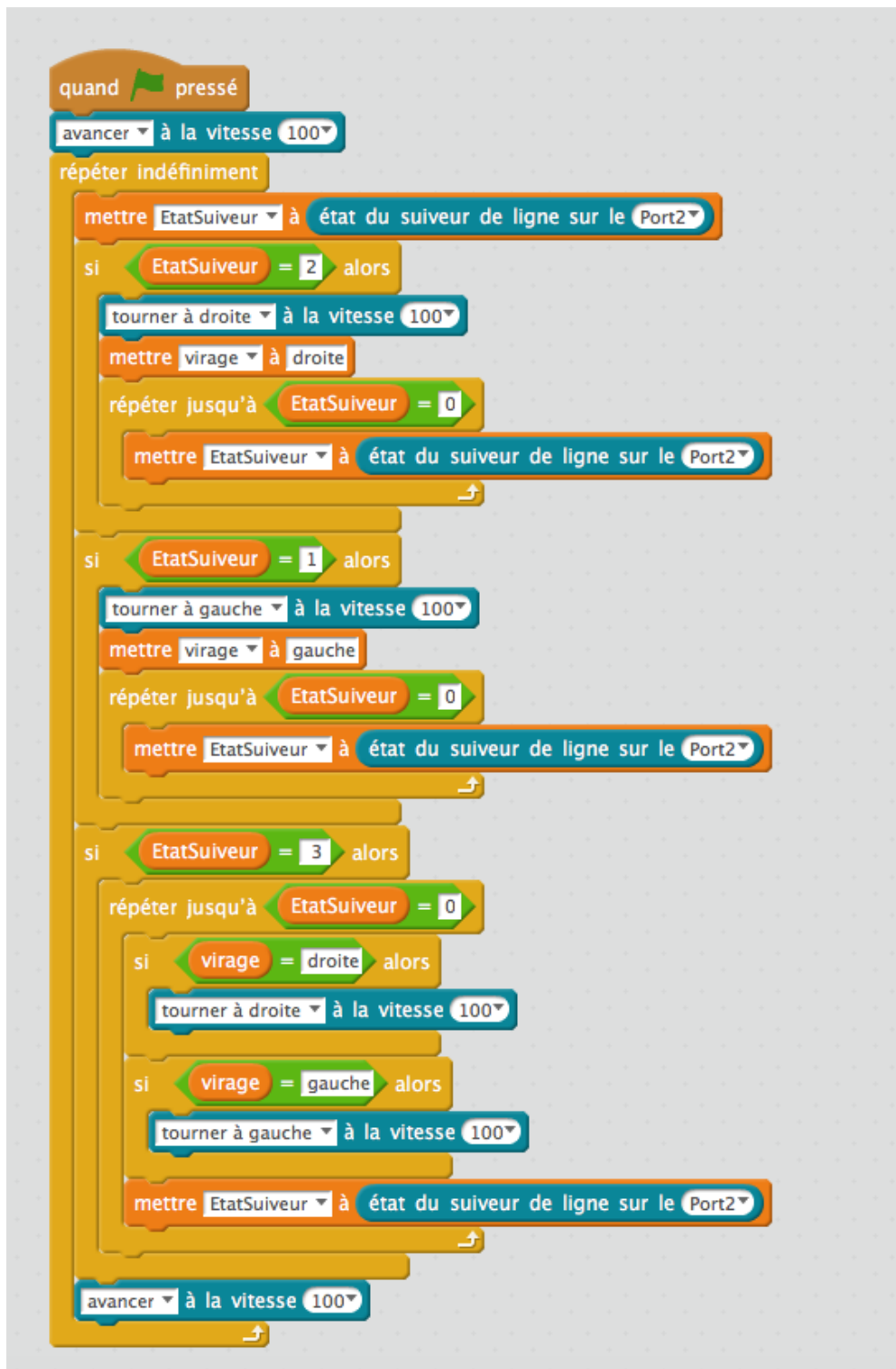
Mettre la variable **EtatSuiveur** à état du suiveur de ligne

Fin Répéter jusqu'à

Fin Alors

Fin Répéter indéfiniment

Les instructions Mblock correspondantes, à écrire dans l'onglet script (programme) sont:



En cas de sortie de piste (capteur ayant pour valeur 3), le programme choisit ici de poursuivre le virage dans le sens du dernier changement de direction. Malheureusement ce choix n'est pas infallible, à vous d'améliorer cet algorithme pour rester sur la piste quelque soit la situation rencontrée.