

Connaissances associées : Piloter un système connecté localement ou à distance. Modifier ou paramétrer le fonctionnement d'un objet communicant.

Comment programmer un robot ?

Prénom :

Classe :

Comment programmer un robot pour qu'il suive une ligne ?

Le robot **mBot** peut se déplacer en suivant une ligne noire au sol. Pour assurer cette fonction, il dispose à l'avant d'un module **suiveur de ligne**, composé de deux **capteurs optiques**.

POUR CE PROJET, IL FAUT : un robot mBot, 2 capteurs optiques, le logiciel mBlock, un parcours tracé en noir.

Vidéo « [suivre une ligne](#) »



Capteurs optiques



Capteur gauche

Capteur droit

Analyser le comportement du robot

Indique, dans le tableau, la position des deux capteurs optiques, droite et gauche, lorsque le robot se déplace.

Le robot avance	Le robot tourne à gauche	Le robot tourne à droite	Le robot recule
Le capteur optique droit est sur la ligne noire.	Le capteur optique droit	Le capteur optique droit	Le capteur optique droit
Le capteur optique gauche est sur la ligne noire.	Le capteur optique gauche	Le capteur optique gauche	Le capteur optique gauche

Modifier, compléter, écrire un algorithme

Selon la position des deux capteurs, le module suiveur de ligne renvoie une valeur au programme qui définit son état :

- si les deux capteurs détectent la ligne noire, l'état du module suiveur de ligne est à 0;
- si le capteur droit ne détecte plus la ligne noire, l'état du module suiveur de ligne est à 1;
- si le capteur gauche ne détecte plus la ligne noire, l'état du module suiveur de ligne est à 2.

En dehors de ces valeurs reçues par le module suiveur de ligne, le robot recule.

Compléter les tests des algorithmes 2 et 3 qui permettent au robot de se remettre dans l'axe de la ligne noire.

Algorithme 1

Si l'état du suiveur de ligne = 0 **alors**
Faire avancer le robot

Algorithme 2

Si l'état du suiveur de ligne = ... **alors**
.....

Algorithme 3

Si l'état du suiveur de ligne = ... **alors**
.....

Dire ce que fait le robot si l'état du module suiveur de ligne est différent de 0, 1 ou 2.

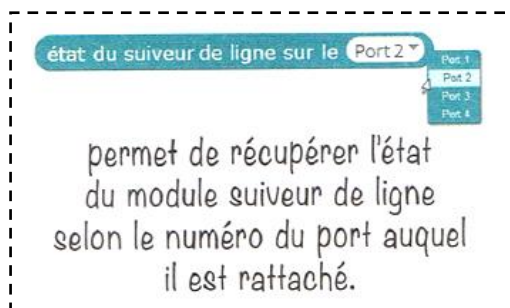
Ecrire un programme

Programmer les capteurs

La vitesse du robot est affectée à la variable *Vitesse*. L'état du module suiveur de ligne est affecté à la variable *Etat*.

- **Lancer** le logiciel mBlock et **ouvrir** le fichier **Suiveur 1**.
- **Donner** le numéro du port qui récupère l'état des deux capteurs du module suiveur de ligne.

- **Compléter** le programme, à partir des trois algorithmes de l'étape 2 et des éléments ci-dessous.



Pour **implanter** un programme dans mBot consulter le fichier « **Robot Implanter** »

- **Allumer** le robot et **implanter** le programme
- **Tester** le programme. **Noter** le problème rencontré et **rechercher** une solution pour que le programme teste en permanence l'état des deux capteurs.

Mettre au point et exécuter un programme

- **Insérer** la boucle « **répéter indéfiniment** » qui permettra au robot de suivre la ligne noire.
- **Allumer** le robot et **implanter** le programme.
- **Tester** le programme.



DEFI

- **Lancer** le logiciel mBlock et **ouvrir** le fichier **Suiveur 2**.
- **Proposer** une amélioration du programme pour que le robot s'arrête 20 cm avant de percuter un obstacle. **Utiliser** le module détecteur d'obstacle.



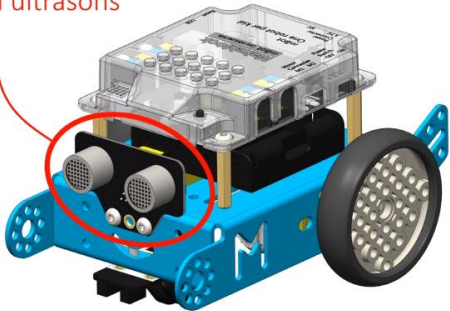
Comment programmer robot pour qu'il surveille un espace ?

Le robot mBot peut surveiller un espace et effectuer des rondes de manière **aléatoire**. Grâce à un **capteur à ultrasons**, il peut éviter les obstacles. Ce capteur envoie au programme, via le **port** 1, 2, 3 ou 4 de la carte mère, la distance qui le sépare d'un obstacle, il peut ainsi modifier son chemin.

POUR CE PROJET, IL FAUT : un robot mBot, un capteur à ultrasons, le logiciel mBlock.

Vidéo « [surveillance](#) »

Capteur à ultrasons



Analyser le comportement d'un robot

Donner le comportement du robot suivant les situations dans lesquelles il pourrait se retrouver durant son parcours aléatoire.

L'obstacle se situe entre 20 et 25 cm	L'obstacle est à plus de 25 cm	L'obstacle est à moins de 15 cm	L'obstacle se situe entre 15 et 19 cm
	Le robot avance		

Modifier, compléter, écrire un algorithme

À l'aide du tableau précédent, **compléter** l'algorithme qui permet au robot d'éviter les obstacles.

Algorithme

Répéter indéfiniment

Si la distance de l'obstacle est inférieure à 26 cm **alors**

Si la distance de l'obstacle est inférieure à 15 cm **alors**

.....

Sinon

Si la distance à l'obstacle**alors**

.....

Sinon

.....

Sinon

Faire avancer le robot

Écrire un programme

Détecter un obstacle et l'éviter

Le robot se déplace à la vitesse | de « 100 » pendant une demi-seconde. Le capteur à ultrasons est relié à un des quatre ports.

- **Compléter**, à l'aide de l'algorithme de l'étape 2, les cases blanches de la séquence d'instruction ci-contre pour que le robot puisse détecter et éviter les obstacles.

- **Lancer** le logiciel mBlock. **Ouvrir** le fichier **Surveillance1**

- A l'aide des blocs d'instruction de la rubrique

- « **pilotage** », **compléter** le programme.

- Donner le numéro du port utilisé pour le capteur à ultrasons.

.....

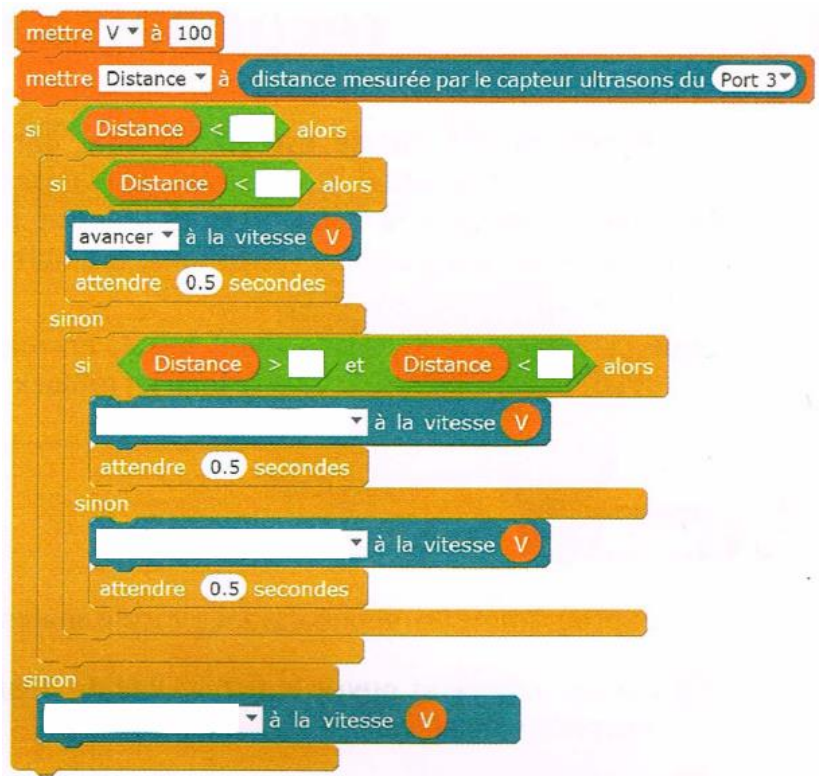
- **Allumer** le robot et lui **implanter** le programme de **Surveillance1**.

- **Tester** le programme en utilisant un obstacle. **Dire** ce que l'on constate.

.....

.....

.....



Mettre au point et exécuter un programme

Se déplacer de manière autonome

Lorsque le robot a fini de reculer, on souhaite qu'il tourne au hasard à droite ou à gauche.

Pour cela, on réalise un tirage aléatoire d'un nombre compris entre 1 et 10. Si ce nombre aléatoire est inférieur à 6 alors le robot tourne à droite, sinon il tourne à gauche.

- **Lancer** le logiciel mBlock. **Ouvrir** le fichier **Surveillance2**.

- Introduire dans le script la boucle « **répéter indéfiniment** » et le bloc d'instruction « **nombre aléatoire entre** » pour que le robot puisse se déplacer de manière autonome et imprévisible.

- Allumer le robot et lui implanter le programme **Surveillance2**.



DEFI

Lorsque le robot mBot détecte un obstacle, il doit pouvoir se positionner perpendiculairement à celui-ci.

Programmer son orientation à 90° lorsqu'il rencontre un obstacle.