

INITIATION à la programmation en ROBOTIQUE

ARDUBLOCK : la programmation graphique (Arduino compatible)

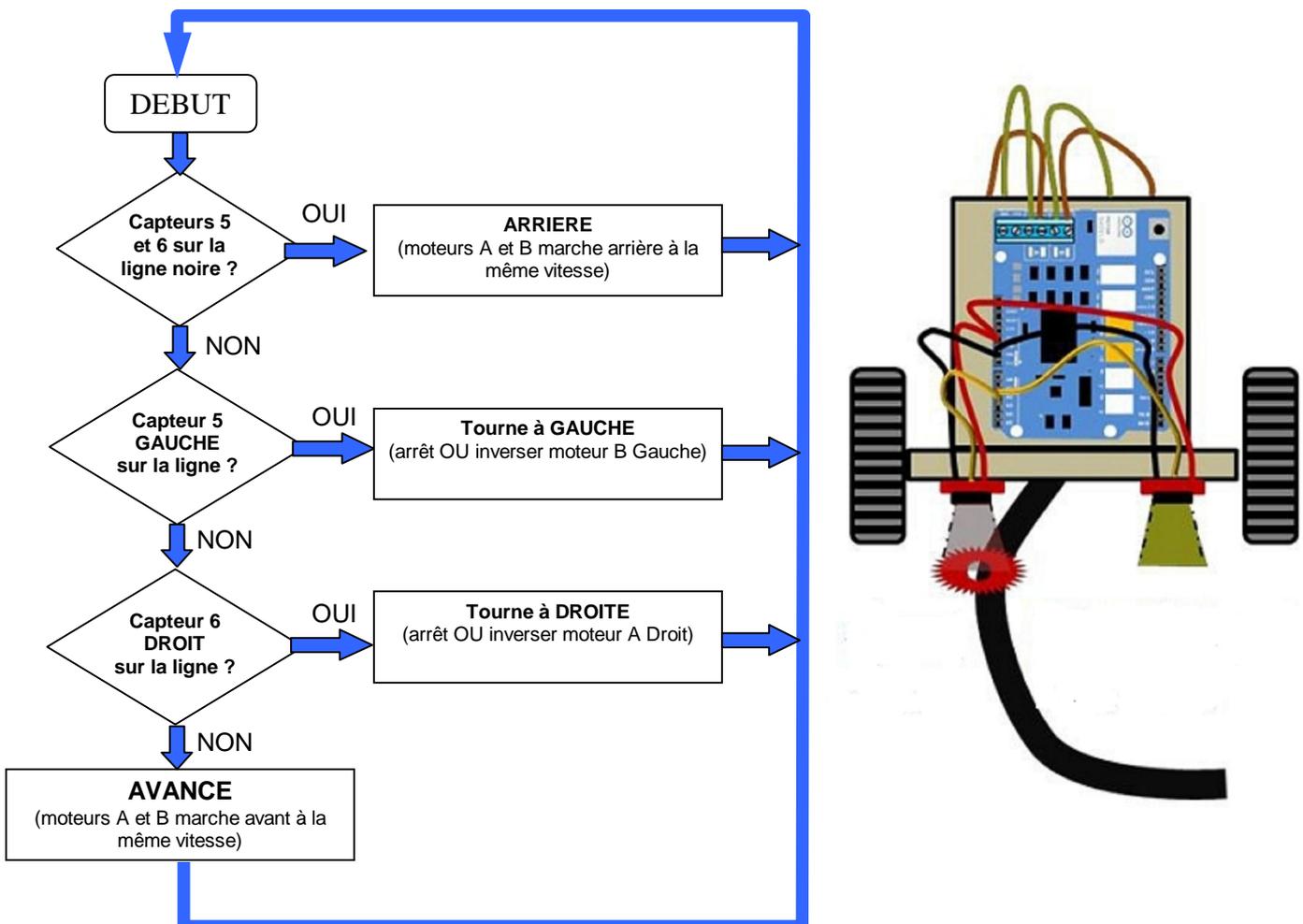


Le Robot suiveur de ligne équipé de :

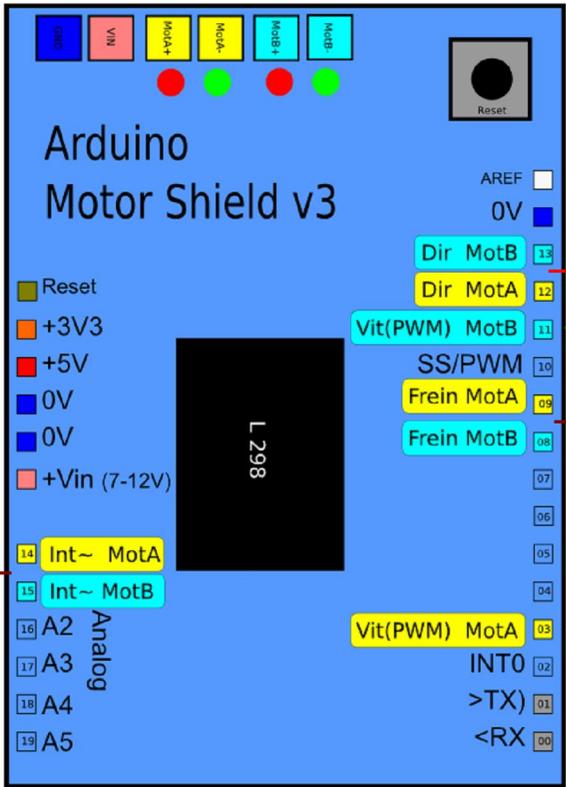
- 2 moteurs à courant continu
- 1 carte Arduino UNO R3
- 1 Motor Shield R3 : carte de contrôle du sens de rotation des moteurs (« double pont en H »)
- 2 capteurs infrarouges (sensors) pour détecter la ligne noire

Logigramme de programmation du Robot suiveur de ligne*

* Les capteurs 5 et 6 sont des détecteurs Infrarouges permettant de détecter une ligne noire sur un fond clair



FONCTIONNEMENT DE LA CARTE DE PILOTAGE DES DEUX MOTEURS DE ROUES

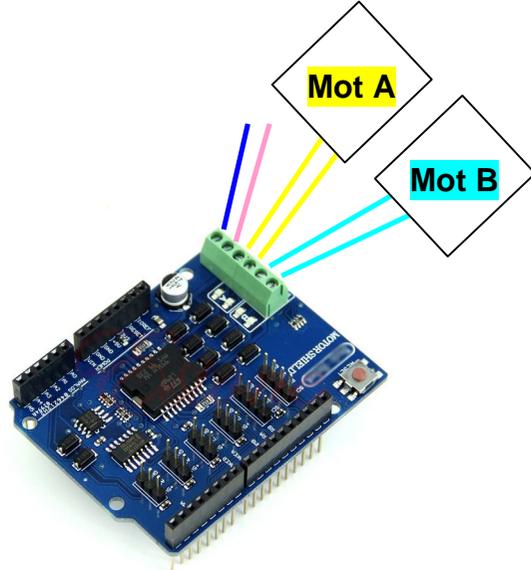
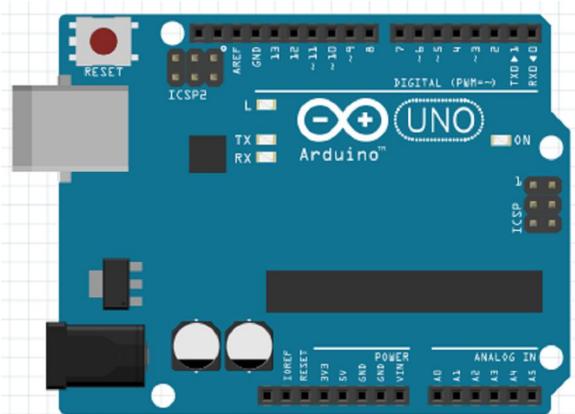


Int Mot :
Mesure de l'**Intensité** des moteurs A et B

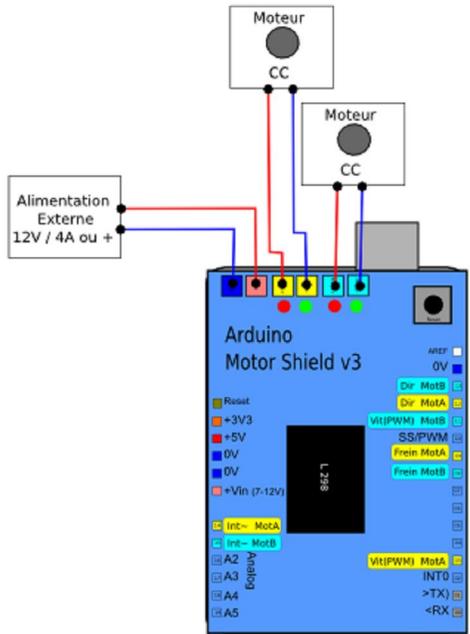
Dir Mot : la broche de **SENS** fixe le sens de rotation (**HIGH** dans un sens, **LOW** dans l'autre sens)

Frein Mot : la broche de **Frein** stoppe le moteur : **Inactif** sur niveau **LOW**, **Actif** sur niveau **HIGH**

Vit Mot : la broche de **Vitesse** (PWM) fixe la vitesse de rotation du moteur (vitesse Maxi = 255)

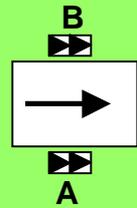


CABLAGE

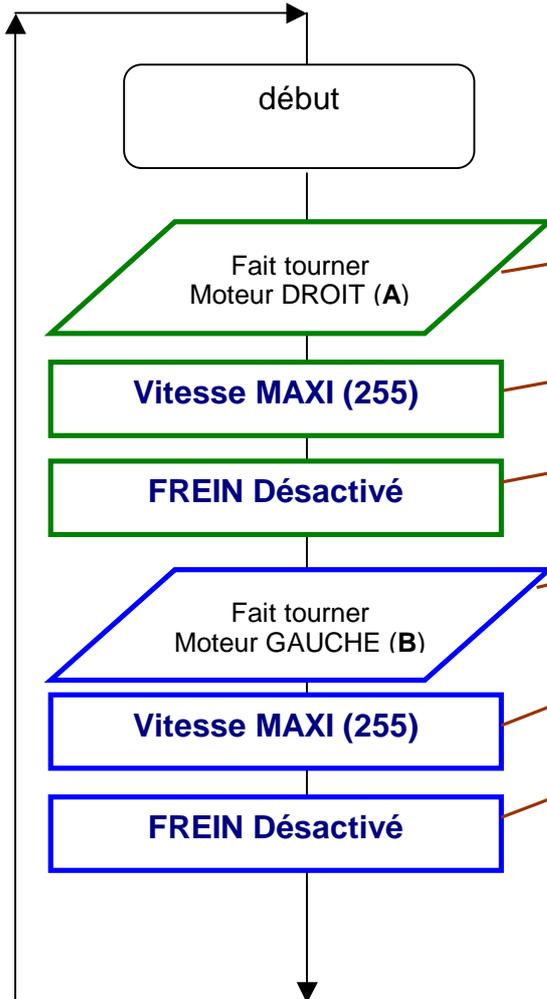


Programmation Ardublock : PARTIE 1

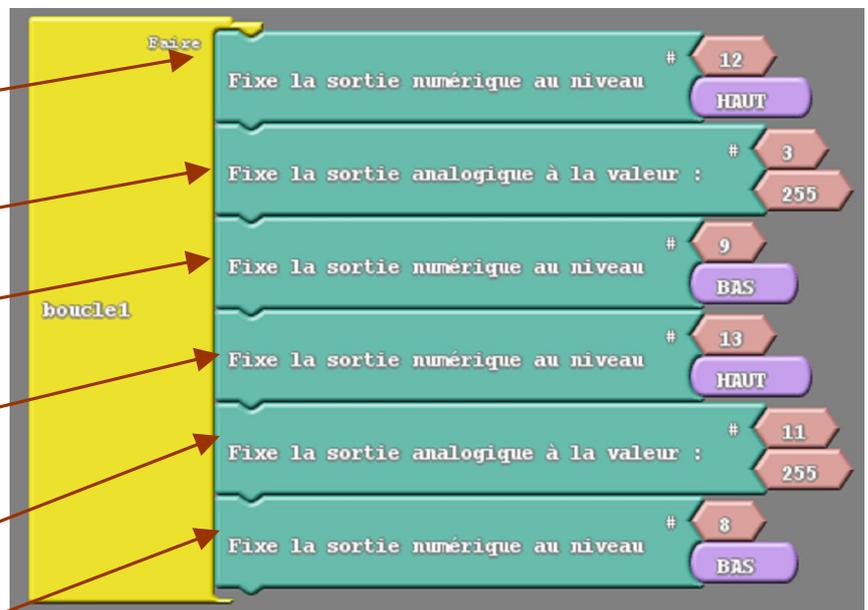
Robot AVANCE (tout droit)
*Faire tourner deux Moteurs
A et B en AVANT*



Logigramme



Programmation Ardublock



réaliser le programme avec Ardublock

- 1- démarrer le logiciel Arduino
- 2- dans la barre d'outils « Outils » puis « Ardublock »
- 3- réaliser le programme ci-dessus
- 4- transférer le programme en cliquant sur en haut de l'écran.

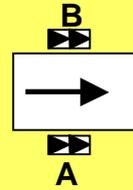
Téléverser vers l'Arduino

Fonctionnement observé : le programme est exécuté en boucle donc le robot avance tout droit (sans jamais s'arrêter).

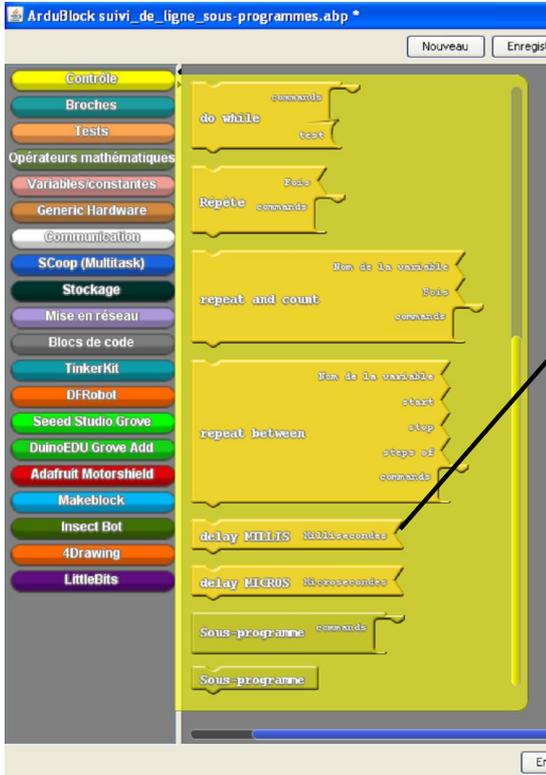
👉 Dans la partie suivante tu vas apprendre à programmer la durée des actions.

Programmation Ardublock : PARTIE 2

Robot **AVANCE** pendant **5 secondes** puis **s'ARRETE** 2 secondes



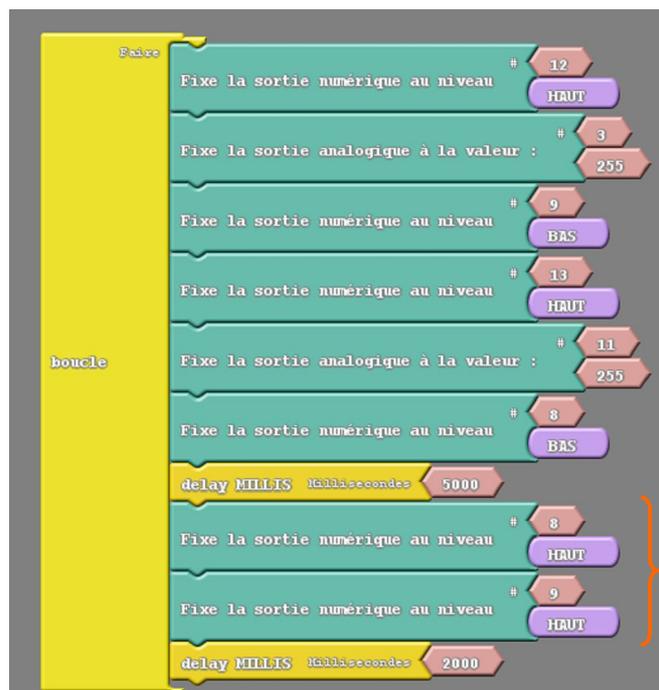
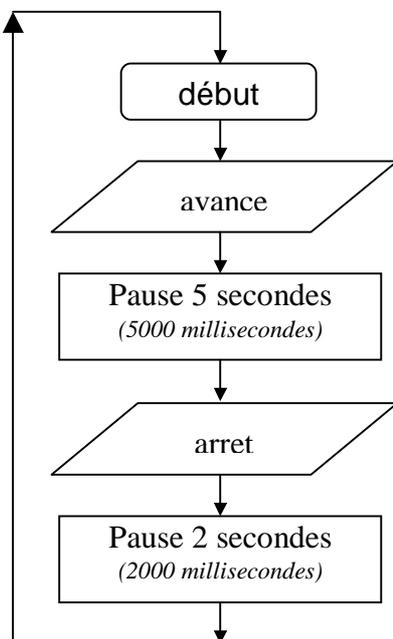
1- Afin de limiter la durée d'une action dans le temps il faut utiliser la fonction « Délais » :



Pour indiquer la durée il faut saisir une valeur numérique correspondante en millisecondes sur l'onglet **rose**
ex : 5 secondes = 5000 millisecondes

2- Tu vas modifier le programme de la partie 1 afin que le robot avance pendant 5 secondes puis s'arrête pendant 2 secondes et répète ainsi de suite le programme en boucle. Transférer le programme modifié dans la carte Arduino en cliquant sur « Transférer » :

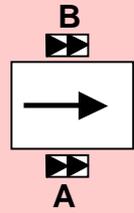
Logigramme



Arrêter le robot
= activer **frein MOT A** et **frein MOT B**
= mettre les sorties numériques 8 et 9 au niveau HAUT

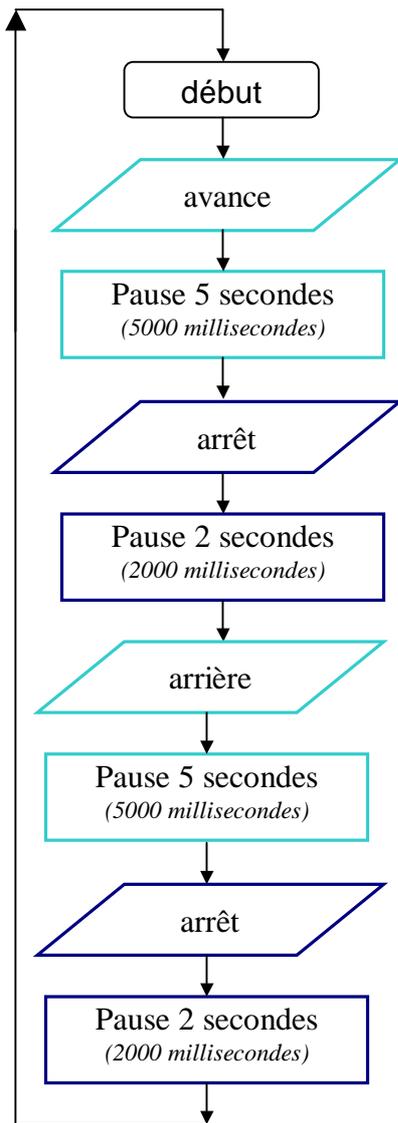
Fonctionnement observé : le robot avance pendant 5 sec. puis s'arrête 2sec. Ce programme est exécuté en boucle.

Programmer une succession de plusieurs déplacements

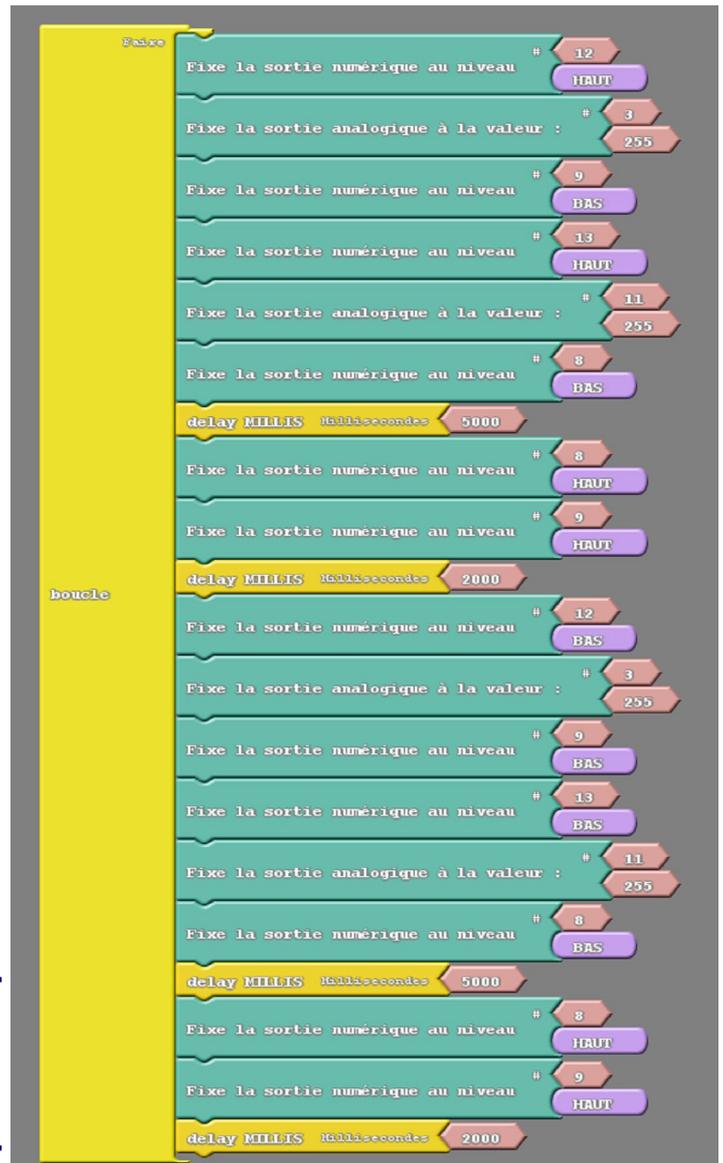


On va programmer ici les déplacements suivants :
 robot avance tout droit pendant 5 secondes / puis s'arrête 2 sec. / puis recule (arrière)
 pendant 5 sec. / puis s'arrête 2 sec. / et répète en boucle ces déplacements.

Logigramme



Programmation Ardublock

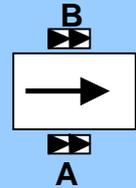


Dans la programmation Ardublock (ci-dessus à droite) on observe une multiplication des blocks qui génère un programme de plus en plus long.

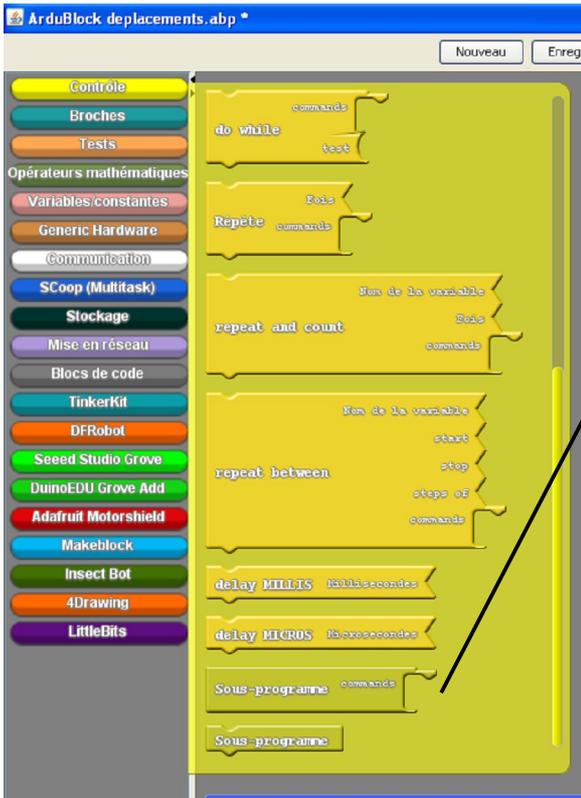
☞ Dans la partie suivante tu vas apprendre à créer des **sous-programmes** ce qui permettra de raccourcir et simplifier la programmation avec Ardublock

Programmation Ardublock : PARTIE 4

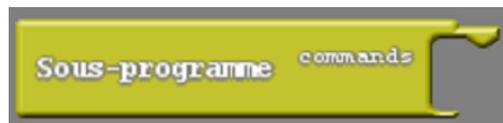
Créer des sous-programmes et programmer les déplacements du robot



Afin de simplifier la programmation nous allons utiliser des sous-programmes. Un sous programme regroupe plusieurs actions qui seront exécutées en utilisant un seul block qui porte le même nom que le sous programme correspondant.
Exemple :



La fonction sous-programme « **Sous-programme** » se trouve dans les blocks de contrôle (jaune)

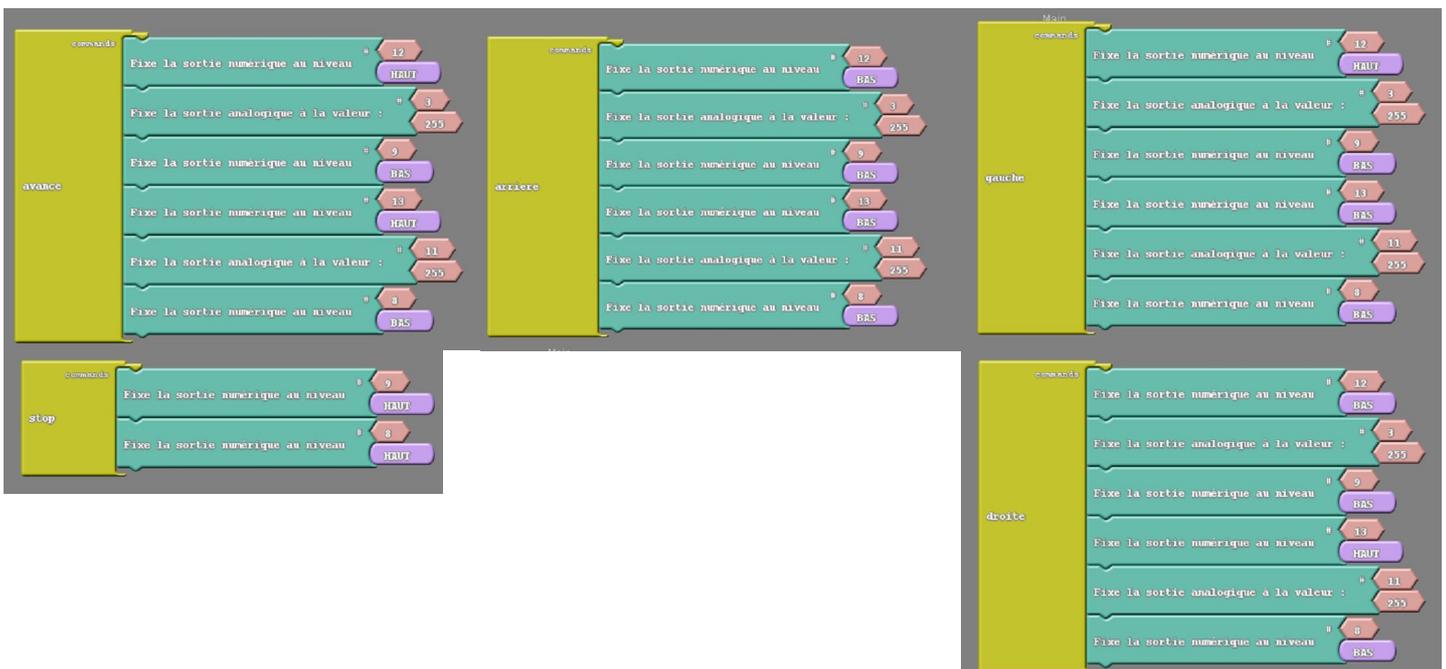


Pour programmer les déplacements d'un robot à 2 roues motrices il est plus simple de créer d'abord un sous-programme par type de déplacement : avant / arrière / droite / gauche / arrêt.

Pour créer un sous-programme :
Il faut associer les actions correspondantes au déplacement (en vert) et écrire le nom du déplacement dans le block Sous-programme.

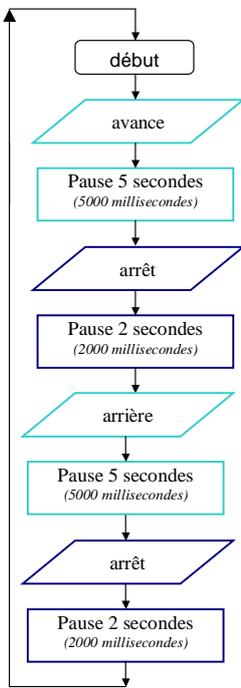
Il faut créer les sous-programmes un par un et les placer cote à cote en haut à droite de la fenêtre du logiciel Ardublock.

Voir ci-dessous :



Il sera ainsi beaucoup plus simple de programmer les déplacements du robot car il suffira de remplacer les différentes lignes de commande (blocks) par un seul block Sous-programme « avance », « arrière », « droite » etc.

Logigramme



Programmation Ardublock classique



Programmation Ardublock avec sous-programmes



TRAVAIL A FAIRE :

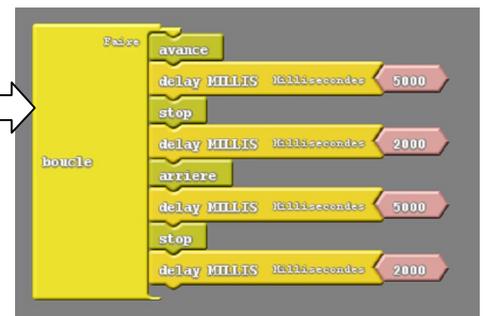
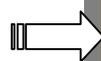
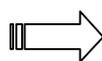
1- Pour aller plus vite tu vas OUVRIR le fichier Ardublock « déplacements » en cliquant sur en haut la fenêtre du logiciel Ardublock (voir les consignes de l'activité pour retrouver l'emplacement du fichier sur le réseau du collègue).

2- Une fois ouvert le fichier contient uniquement les sous-programmes de déplacement du robot (avance – arrière – gauche – droite – stop)

3- Tu vas utiliser la fonction « **boucle** » puis la fonction « **Sous-programme** » .

4-Il faut maintenant remplacer le texte « **Sous-programme** » et écrire à la place le nom du sous programme désiré par exemple « **avance** ».

(attention il faut écrire exactement le même texte que dans le sous-programme sinon il ne sera pas reconnu et pas exécuté)

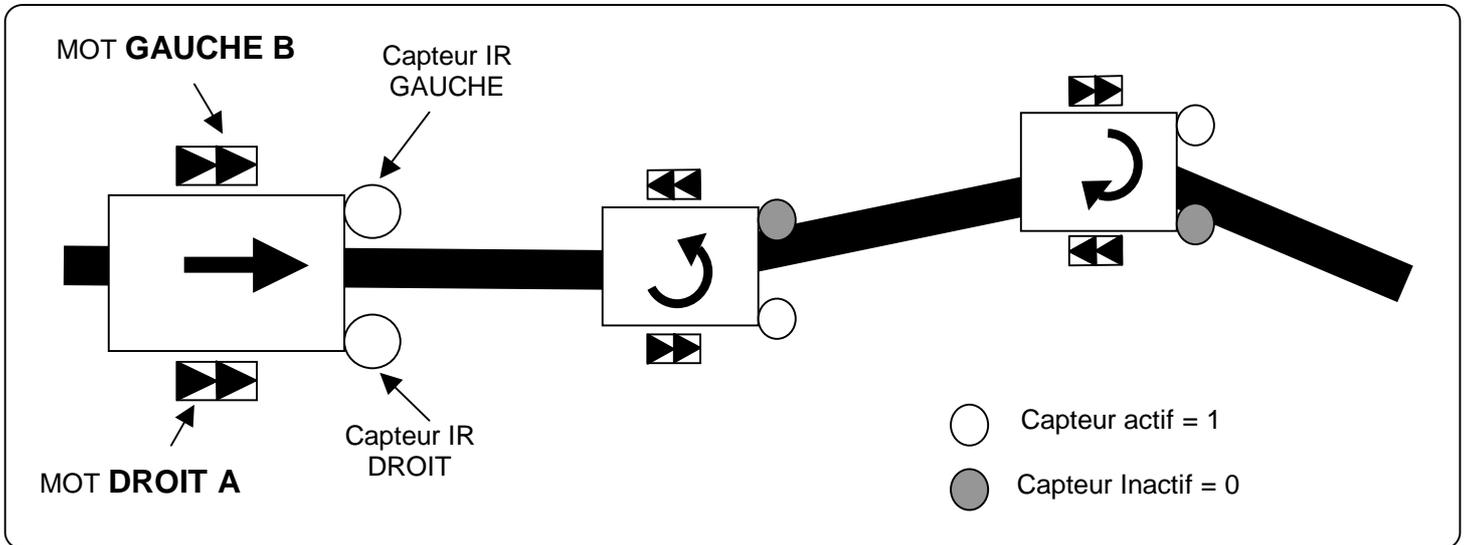
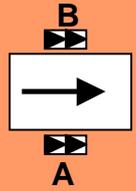


Transférer le programme en cliquant sur « Téléverser vers l'Arduino » »

Fonctionnement observé : le robot avance pendant 5 sec / s'arrête 2 sec / puis recule 5 sec / et s'arrête 2 sec / et recommence en boucle ce fonctionnement

☞ le programme est beaucoup plus court et compréhensible

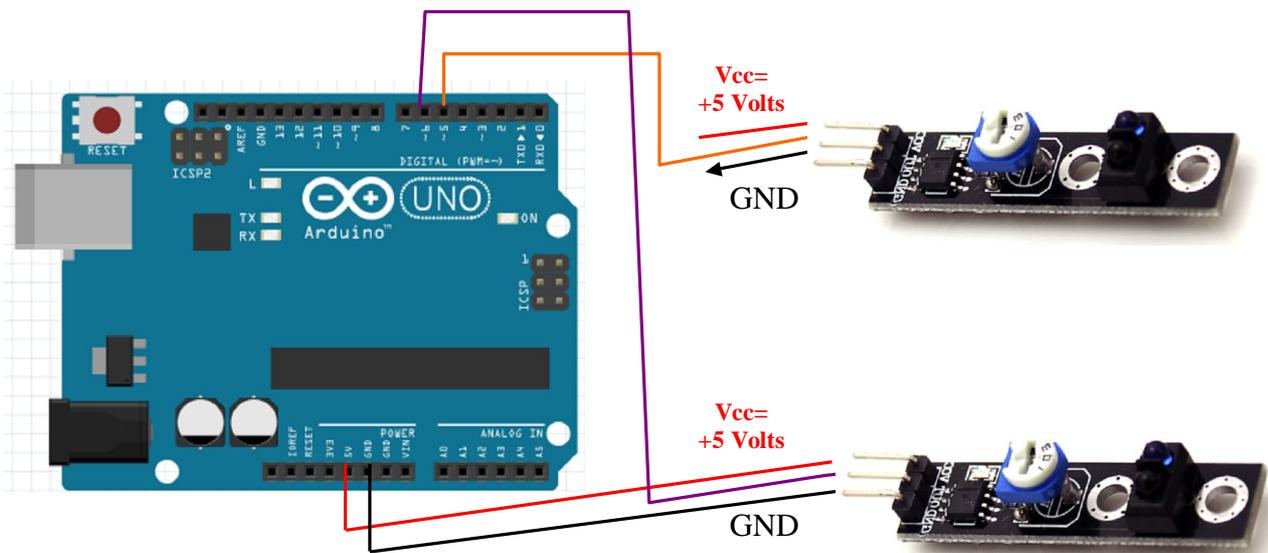
Branchement des détecteurs de ligne à Infrarouges



Réaliser le câblage des deux capteurs de ligne comme indiqué ci-dessous :

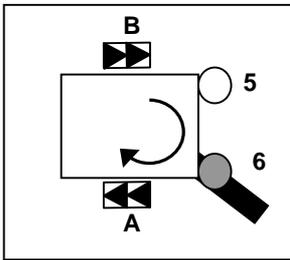
- **Vcc** relié au **+ 5Volts** de l'Arduino
- **GND** relié au **GND** de l'Arduino
- **Signal** relié à l'Entrée Numérique **5** ou **6** de l'Arduino (**PIN 5** ou **PIN6**)

Capteur **GAUCHE** PIN5

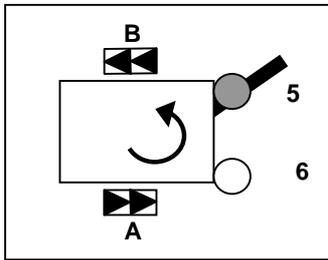


Capteur **DROIT** PIN6

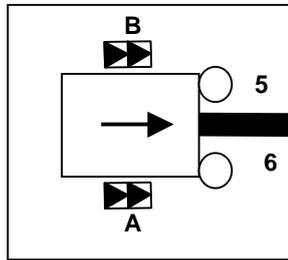
Logique de programmation des capteurs de suivi de ligne



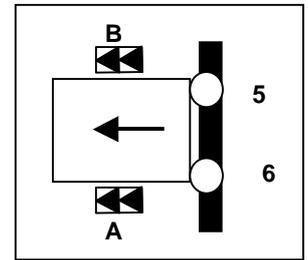
Si le capteur Droit détecte la ligne alors tourner à Droite



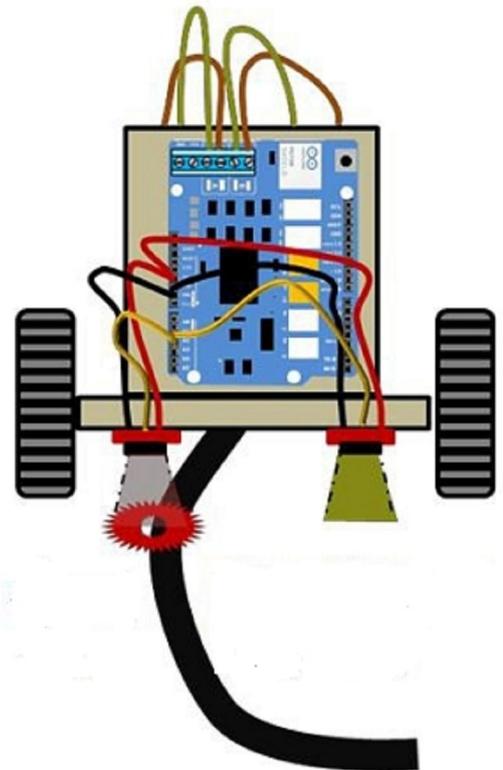
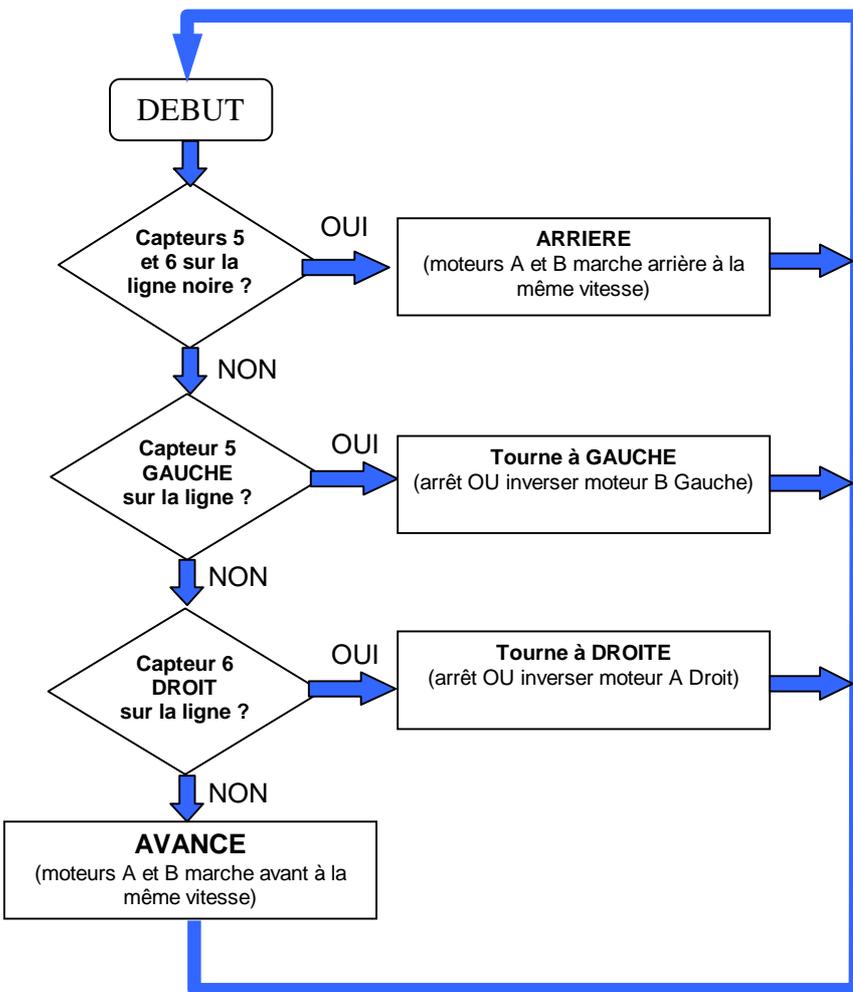
Si le capteur Gauche détecte la ligne alors tourner à Gauche



Si aucun capteur sur la ligne alors AVANCER tout droit



Si les deux capteurs sont sur la ligne alors RECULE



Programmation du robot suiveur de ligne

Découverte des nouvelles fonctions qui seront utilisées pour la programmation du robot suiveur de ligne

Exécute en boucle



La condition Si



La condition Si ...Sinon.....



Les Sous-Programmes

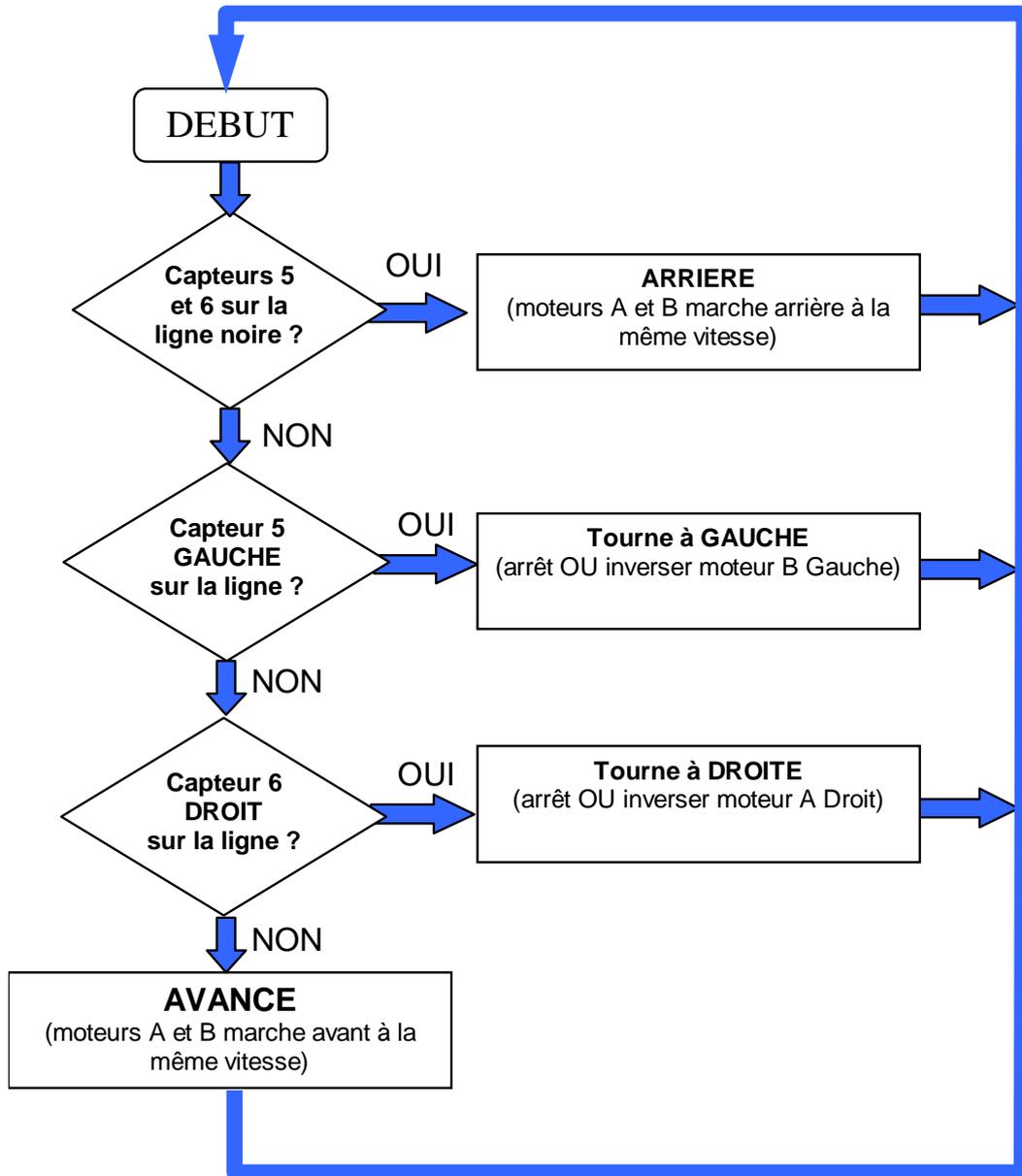


Les fonctions Logiques ET et NON

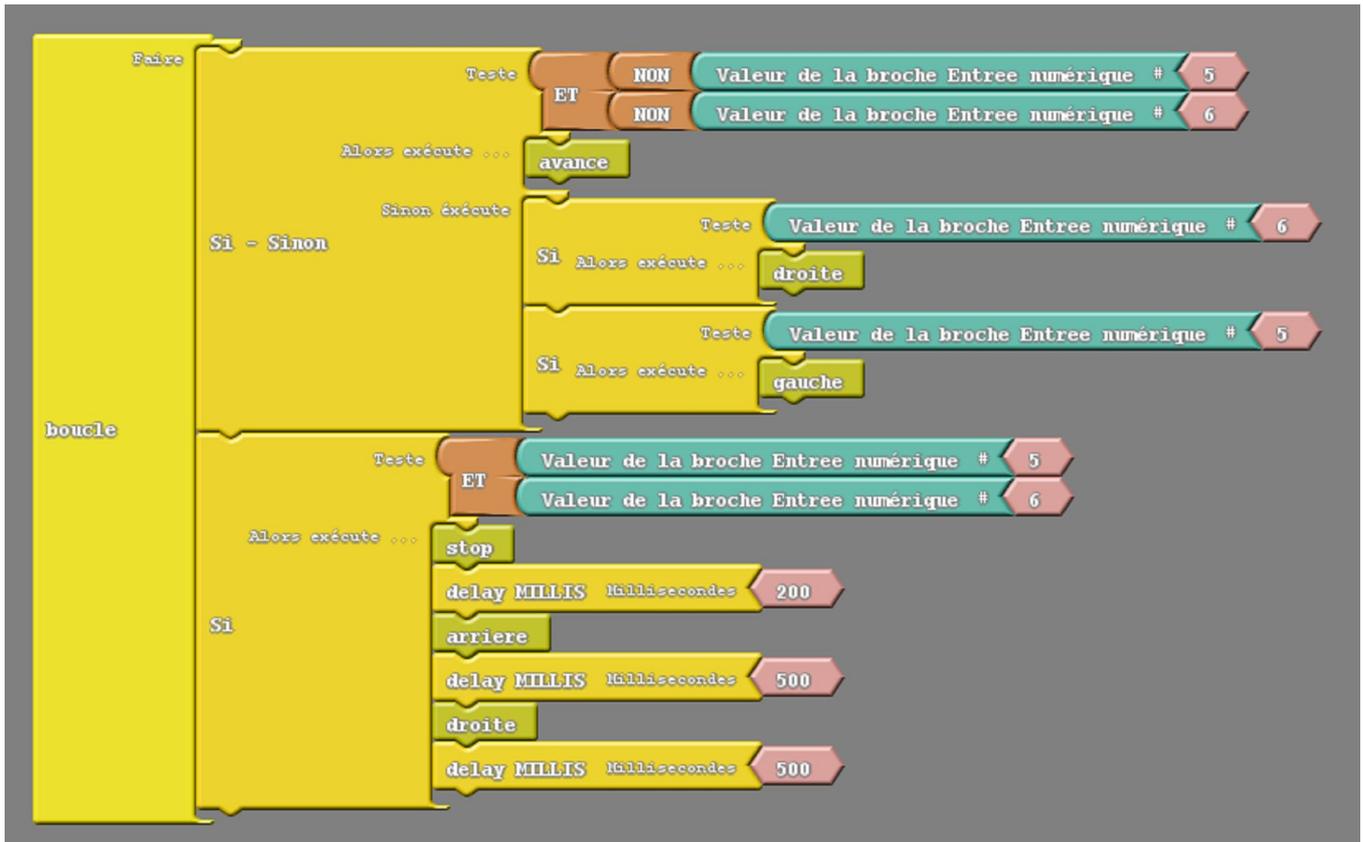


Programmation du robot suiveur de ligne

Logigramme

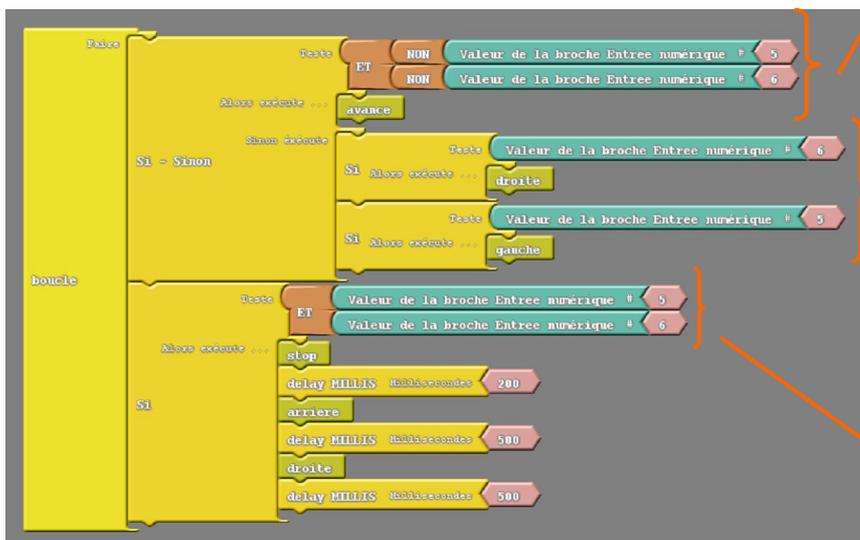


Programmation Ardublock



Réaliser le programme ci-dessus et le transférer dans la carte Arduino en cliquant sur « Téléverser vers l'Arduino »

ANNEXE 1



Si les capteurs 5 **ET** 6 sont sur la surface blanche **Alors** exécute l'action « **avance** »

Sinon teste si la capteur 6 (droit) est sur la ligne noire :
Alors exécute virage à **droite**

Sinon teste si la capteur 5 (gauche) est sur la ligne noire :
Alors exécute virage à **gauche**

Si les capteurs 5 **ET** 6 sont sur la ligne noire :
Alors exécute l'action marche « arriere » puis « demi-tour par la droite »



Suivre une ligne et s'arrêter devant un obstacle.

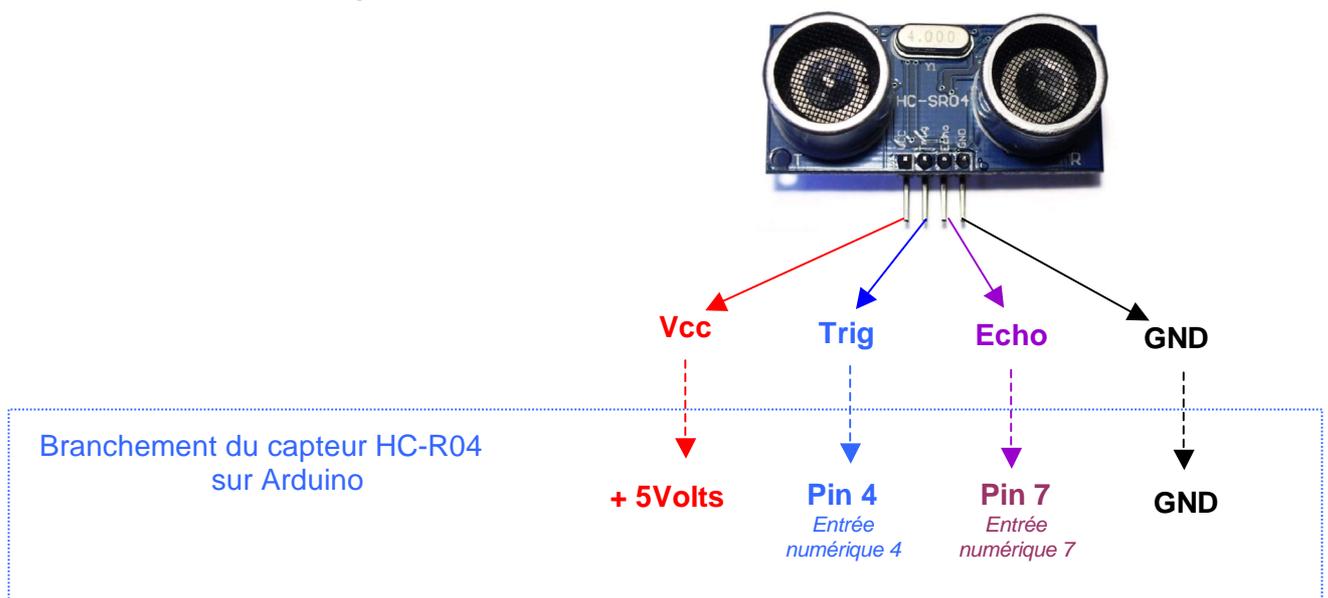
Brancher et Tester un détecteur d'obstacle HC-SR04

1- Principe de fonctionnement :

Le cylindre Emetteur T (Transmitter) envoi des ultrasons, l'autre cylindre R (Recepteur) récupère ceux qui reviennent suite à la collision avec un objet. En fonction du temps que l'onde aura mise pour revenir, nous pourrons savoir la distance qu'elle a parcourue grâce à la formule :

$$\text{Distance} = [(\text{durée du niveau Haut}) \times (\text{vitesse du son : } 340\text{m/s})] / 2$$

2- Branchement du capteur



3- Test du fonctionnement du capteur HC-SR04

3-1 : programmation graphique

afin de tester le bon fonctionnement du capteur d'obstacle, voici comment créer un petit programme graphique permettant de mesurer la distance entre le capteur et un objet et d'afficher cette distance en centimètres sur le Moniteur Série. Téléversez le programme vers 'Arduino puis Afficher le Moniteur Série . Le Moniteur Série affiche à l'écran en centimètres les distances mesurées ...

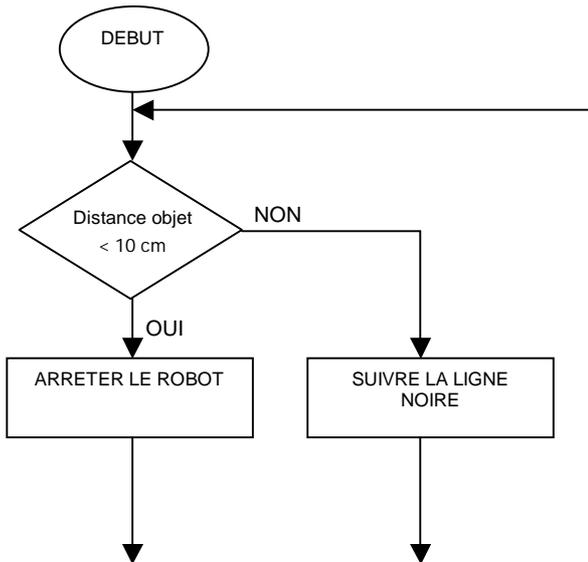




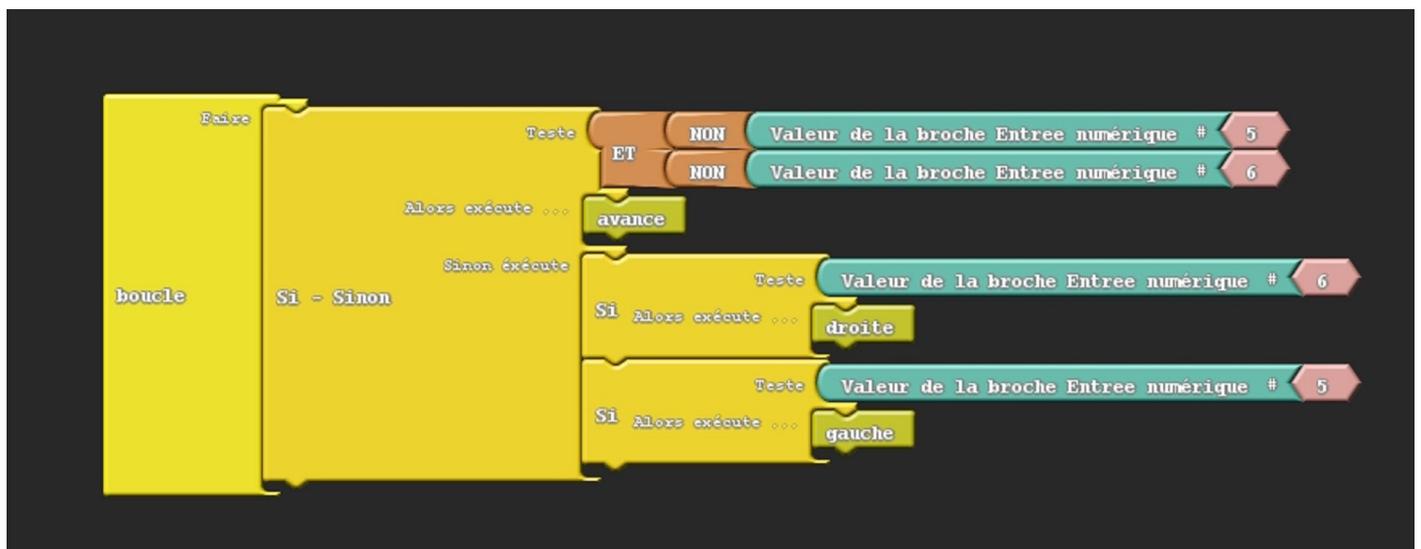
Suivre une ligne et s'arrêter devant un obstacle.

Programmer le robot utilisant le HC-SR04

- 4- Modifier le programme de suivi de ligne afin que le Robot suive la ligne noire et s'arrête à 10 cm avant l'obstacle détecté. Voici le logigramme de fonctionnement du Robot Suiveur de ligne qui doit s'arrêter à 10 cm devant la boîte en carton lorsqu'elle sera détectée par le HC-SR04 (capteur à Ultra sons).



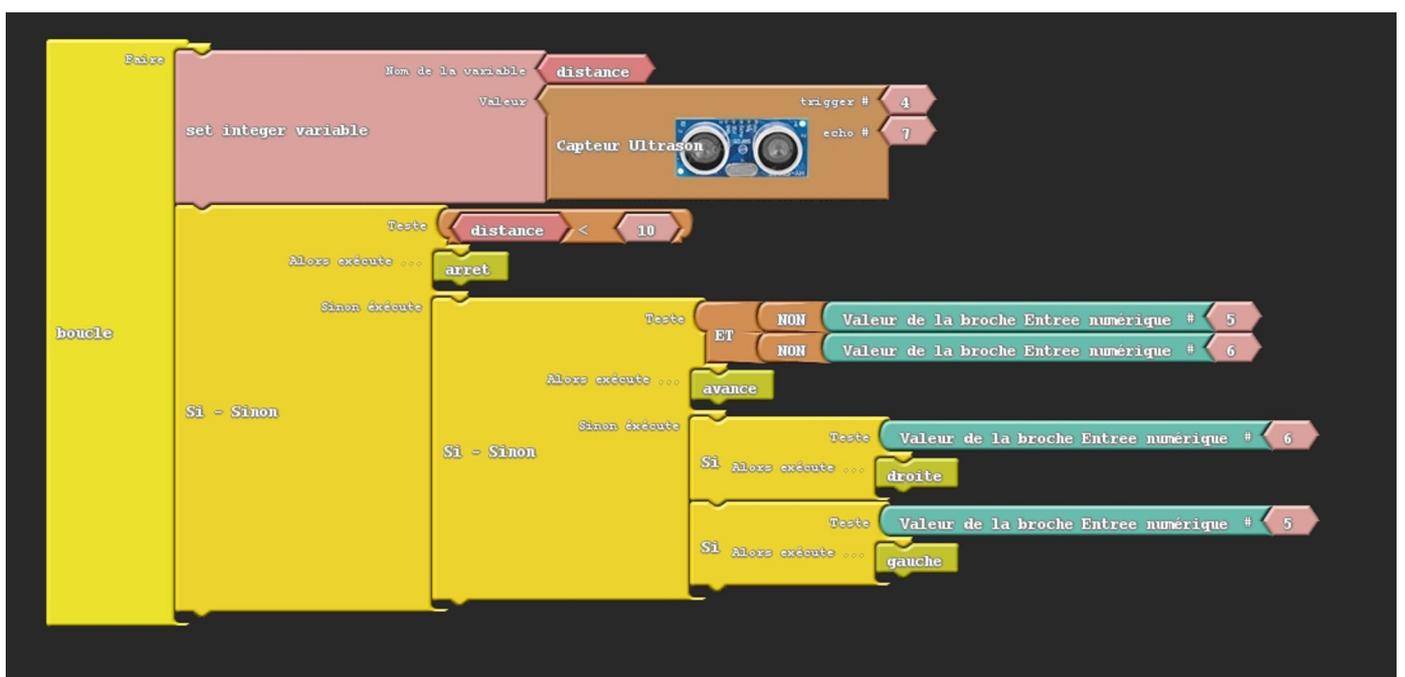
Pour commencer vous devez partir du programme permettant de suivre la ligne noire, voir ci-dessous : (appel : 5 et 6 sont les capteurs Infrarouges de suivi de ligne)



Ensuite il faudra créer la variable « distance » et indiquer que la « valeur » de cette variable correspond au signal mesuré par le capteur Ultrason.



Ensuite sous la variable « distance » il faudra tester si la distance est inférieure à 10 cm, dans ce cas il faudra arrêter le robot.....Sinon il faudra exécuter le programme de suivi, voir ci-dessous :



Téléversez ce programme dans l'Arduino, le robot doit s'arrêter s'il détecte un obstacle placé à moins de 10 cm, sinon il suivra la ligne noire.

Comment optimiser le programme (ou algorithme) afin de fluidifier les déplacements du robot